The Educational Electronic Patient Dossier for Technical Medicine

Design Report

Jasper Grimberg, Nijs van der Laan, Sanne Lohuis, Hossein Rajabali Nejad, Ümid Akbarov, and Brian Patrick van Oers

EEMCS, University of Twente, 7500 AE Enschede, The Netherlands

Abstract. In Technical Medicine (TG), students work with an Electronic Patient Record (EPD). Like in a hospital, they can open a patient file, request diagnostic procedures, and view medical images, such as CT and MRI scans. Teachers use the system to run realistic patient cases and respond to the requests made by the students.

The new system allows teachers to automatically run these cases when tutors are unavailable. The system can automatically approve and reject requests. When the students requested all the needed procedures, and when the next stage is unlocked, they advance to the next stage. Based on the results, the students have to re-evaluate their initial assumptions and continue their diagnosis, and possibly treatment, of the patient. If they didn't order all the needed procedures in time, they will still get the results for that stage and automatically progress to the next stage. The system will then mark that stage as incomplete.

Furthermore, the system allows students to view the medical images directly in the browser through a DICOM viewer. In addition, the entire dossier can be exported to PDF with additional information, such as missed deadlines, missed and rejected procedures, and when and by whom actions were performed, such as approving and rejecting requests.

Keywords: Educational Electronic Patient Dossier (eEPD) · Technical Medicine (TG)

Table of Contents

1 Introduction 4
2 Domain Research 4
3 Methodology 5
3.1 Initial planning 5
3.2 Project Goal and Goal Hierarchy 6
3.3 Sub-goals 7
3.4 Stakeholder Wishes 8
3.5 Requirements Specification 9
Functional Requirements 9
Non-functional Requirements 12
3.6 Research Questions 13
3.7 Design Log 14
3.8 Risk Analysis 14
Risk Register 15
4 Design 19
4.1 From Problems to Design Options 19
4.2 Scoring the options 19
4.3 From scored options to design decisions 20
4.4 Backend Design 20
Backend Class Diagram 20
Login and Accounts 22
Requesting Workflow 23
File support 24
Automation 26
Technical Architecture 26
4.5 Web-App Design 29
Students 29
Procedure Requests and Notes 30
Tutor Dashboard 31
Group Creation 31
User Management 33
Case Creation 33
Examinations, Case Progression, and Results 34
Overview and PDF Export 37
Technical Architecture (Frontend) 39
5 Testing 41
5.1 Front-end 41
5.2 Back-end 41
5.3 Security and Penetration Testing 42
Results of passive penetration testing 42
Results of active penetration testing 43

6 Results 44
6.1 Achievement of the Project Goals 44
6.2 Requirements 45
6.3 Research Questions 46
6.4 Overall Result 47
7 Discussion & Future Work 48
7.1 Transfer to the UT 48
7.2 Known Issues 48
7.3 Future Additions 48
8 User Manuals 49
9 Contributions 50
10 AI Disclaimer 50
11 Acknowledgments 50
A Wireframes 51
B Stakeholder Analysis 54
C SWOT Analysis 57
D FMEA 58
E Design Decisions (DEC) 68
F Option Scoring Matrix 89
G Design Log 92
H Student Manual 112
I Tutor Manual 119
J Teacher Manual 127

1 Introduction

For the Design Project of Module 11 of the Technical Computer Science (TCS) Bachelor's program at the University of Twente (UT), we designed a new Educational Electronic Patient Dossier (eEPD) for the Technical Medicine (TG) program, at the same university.

We started with initial requirements gathering and contacted the right people (supervisor, client, and examination board chair) to get confirmation that it was perfectly reasonable to re-purpose an old project of Brian in order to accelerate the process of building the web-server and, therefore, spend our time on actually improving the final product. Afterwards, we closely worked with our client to make the requirements and client wishes as clear as possible while also implementing the basic parts of our code. During development, our clientele were contacted each weak to show our progress and get back valuable feedback.

During this project, we learned to make maximum use of our skills that we obtained here at the university, including making use of our: network/connections, design knowledge, programming knowledge, hardware knowledge, team-working skills, and security experience. We also learned new skills thanks to our supervisor, particularly systematic design. With the guidance of our supervisor we made strong use of Failure Mode and Effects Analysis (FMEA) and Risk Analysis during this project. Which we used in our decision making with regards to the requirements laid out.

Our final product resulted in a maintainable and user-friendly eEPD that more closely follows how an actual EPD work with integrated stage-based learning. The system has the ability to automate the review process if needed, display DICOM medical images on the website itself and export the results of the students to a clean looking PDF.

2 Domain Research

Electronic Patient Records (or Elektronisch Patiënten Dossiers in Dutch) are commonly used in the medical world. Their purpose is give all relevant and authorized healthcare professionals access to a patient's data. The digitization of this data also has some additional benefits, such as standardization of the data, the automating of routine processes, and the ability to give the patient easy access to the data.[5]

To give students some experience with such a system, the Technical Medicine program at the University of Twente had previously created an educational version of an EPD. In this web application, students can find information about their assigned theoretical patients, request examinations based on their conclusions, and view the results of those examinations once the request has been approved by a teacher or teaching assistant.

The application was written in PHP and though the language is a little archaic, the application was used for many years, educating hundreds of students. Throughout its use, different problems arose. Some of these could be fixed on a code level by the maintainer, such as when there were no teaching assistants available, so correct examination requests needed to be approved automatically. But other issues were too foundational to the application's structure for the maintainer to fix, such as the lack of a

user-friendly way of creating and editing patient cases that the students could work on. This required the maintainer to manually add the cases through the code, which created a lot of hassle for him.

Lastly, the structure of the previous EPD's codebase was also lacking. Practically all files related to the actual code lived in the project's root folder, with little guidance beyond the filenames regarding structure and no code comments. The maintainer was able to learn the inner workings of the system over time, but anyone else had inherited this project, they would have had to put in a lot of work.

3 Methodology

The design of the EPD was developed using an Agile approach; iteratively throughout the project. Stakeholder needs and project requirements were used to design criteria, after which possible solutions were discussed and evaluated during team meetings. Decisions were revised when new requirements, technical constraints, stakeholder feedback, or other risks required changes in the design. The subsections below describe step by step that process we adhered by. Together they explain how the design shaped itself, rather than just how it looks like.

3.1 Initial planning

Before the module started, projects were put online on Canvas. We had to form groups, be sociable enough to introduce ourselves to each other, and then pick a few projects that peaked our interests.

We chose "TG: Build a new Electronic Patient Record for Technical Medicine", a project offered by Dr. Jordy van Zandwijk, the study program manager of Technical Medicine at the UT. Moreover, we needed to decide on a supervisor. We chose Dr. Farzad Vazinram, a man of admirable intelligence, strong work ethic, and above all, willingness to mentor students. Thereafter, we needed to contact our client and our potential supervisor as soon as possible.

Finally, we made an initial planning that roughly outlined the workload per week for the duration of the project (10 weeks). The planning from week 3 onward are representative of our initial planning in week 2, after we gathered the initial requirements:

Table 1: Initial project planning per week.

Week	Dates	Planned activities
1	31-08 - 04-09	Form a team, get to know each other and select a project. Attend the introductory lecture. Contact the client for an introductory meeting and initial requirement gathering. Find a supervisor. Set up communication, progress tracking, code hosting, document hosting and the design-report platform.
2	07-09 - 11-09	Meet the client and gather the initial requirements and additional resources. Set up the web server and database.

Week	Dates	Planned activities
3	14-09 - 18-09	Initial Design & diagrams.
4	21-09 - 25-09	Start implementation.
5	28-09 - 02-10	Implementation and client feedback.
6	05-10 - 09-10	Implementation and teacher feedback.
7	12-10 - 16-10	Implementation of UT login, and feedback from Maroeska and Lennart.
8	19-10 - 23-10	Final touches on the project.
9	26-10 - 30-10	Contingencies.
10	02-11 - 06-11	Presentation, poster, and handover of the project to the client.

Notes on tooling. For communication, progress tracking, code and document hosting we considered WhatsApp, Trello, UT GitLab, Google Drive and Overleaf against Discord, GitHub, OneDrive/Proton Drive and Google Docs. We chose the first set: WhatsApp for day-to-day communication, Trello for progress tracking, UT GitLab for code and version control, Google Drive for shared documents, and Overleaf for the design report.

Notes on the testbed. The web server and database were set up as Apache Tomcat and PostgreSQL on a Raspberry Pi 4. This was chosen because it is familiar (an older project could be reused), open source and low cost (the Pi was already available), and because the Pi can keep running while a team member's laptop is in use, being updated or shut down.

3.2 Project Goal and Goal Hierarchy

Main goal (MG-01). The main goal of this project is to develop a maintainable educational Electronic Patient Dossier (EPD) that personnel of the Technical Medicine programme can operate without continuous dependence on student developers. The system should support the educational workflows of students, tutors, and teachers.

This main goal forms the basis for the project goals, requirements, and design decisions. The project goals describe what the system should achieve, while the functional and non-functional requirements specify how these outcomes can be realised and verified.

Table 2: Project goals derived from the main goal.

Goal	Description	Related needs
G-01	Staff can manage cases, groups, stages, and content without changing source code.	SG-07, SG-10, SG-11, SG-17
G-02	The system can be maintained, deployed, documented, and transferred to Technical Medicine support without depending on the original developers.	SG-18

Goal	Description	Related needs
G-03	Students, tutors, and teachers can complete their educational tasks clearly and efficiently.	SG-01-SG-05, SG-09, , SG-12 - SG-15
G-04	Data is stored securely, remains consistent, and can be recovered or deleted according to the applicable policies.	SG-16, SG-19
G-05	The system supports progressive learning, configurable automation, and medical examination results, including medical imaging.	SG-06, SG-08

3.3 Sub-goals

The sub-goals describe the outcomes needed to achieve the project goals in Table 2. Each sub-goal is linked to its parent project goal.

Table 3: Sub-goals and their parent project goals.

SG	Goal	Description
SG-01	G-03	Navigation should be efficient and scale well as more cases and categories are added.
SG-02	G-03	The tutor dashboard should support efficient work across multiple groups and cases.
SG-03	G-03	Requests and notes should be easy to access without unnecessarily interrupting the user's workflow.
SG-04	G-03	Group management should be clear and efficient, including the prevention of duplicate assignments.
SG-05	G-03	The overview should provide accessible and printable information about student progress.
SG-06	G-05	Students should be able to select examinations efficiently, while invalid requests are identified during tutor review.
SG-07	G-01	Authorized staff should be able to manage modules, teachers, and permissions.
SG-08	G-05	Case progression should be controlled through configurable stages, deadlines, and approval rules.
SG-09	G-03	The interface should remain consistent and use appropriate Dutch medical terminology.
SG-10	G-01	Case creation and configuration should be clear and manageable by authorized staff.
SG-11	G-01	Cases should be reusable across groups and modules without unnecessary duplication.
SG-12	G-03	Patient information should be clearly separated from actions and examination requests.

SG	Goal	Description
SG-13	G-03	Examination results, including medical images, should be readable and interactive.
SG-14	G-03	Students and teaching staff should be able to see relevant progress and deadline information.
SG-15	G-03	Teachers should be able to view the complete case, including information not yet released to students.
SG-16	G-04	Case data and user actions should remain consistent and persist correctly across sessions.
SG-17	G-01	Authorized staff should be able to manage educational content independently while the system enforces appropriate access control.
SG-18	G-02	The system should be maintainable, documented, deployable, and transferable to Technical Medicine support.
SG-19	G-04	Data should remain secure, recoverable, and subject to the agreed retention policy.

3.4 Stakeholder Wishes

The stakeholder wishes cover additional features and expectations discussed during the project. Not all of these were required or implemented, but they were considered when defining the requirements and making design decisions.

Most wishes came from discussions with the client and teaching staff. Some relate to the student interface, although these were not necessarily collected directly from students, however previous years feedback of the old system was taken into account as student feedback.

Table 4 lists the stakeholder wishes and their related requirements.

Table 4: Stakeholder wishes identified during the project.

Code	Stakeholder	Wish
W-01	Teacher / Admin	Create and edit patient cases directly in the system without relying on a programmer.
W-02	Teacher / Admin	Control student progression by locking stages behind deadlines or manual actions.
W-03	Teacher / Admin	View the complete patient case, including information not yet released to students.
W-04	Teacher / Admin	Clearly identify when a student or group has missed a deadline.
W-05	Teacher / Admin	Prevent duplicate student assignments when adding students to groups.
W-06	Tutor	Review requests and student reasoning without constantly switching between screens.
W-07	Tutor	View examination results separately rather than as one large block of text.

Code	Stakeholder	Wish
W-08	Tutor	Navigate directly to the relevant request from a notification.
W-09	Tutor	See who performed an action and when it occurred in the activity history.
W-10	Student	Select multiple individual blood values when requesting laboratory examinations.
W-11	Student	Open examination results through a clear and clickable interface.
W-12	Student	View the GP referral as part of the patient information.
W-13	Client	Use appropriate Dutch medical terminology throughout the interface, including terms such as “stadium” and “aanvullend onderzoek”.
W-14	Client	Generate professional PDF documents that follow the Technical Medicine visual identity.
W-15	Teacher/Admin	Automate all cases in a module all at once.

3.5 Requirements Specification

The requirements define what the EPD should do to meet the project goals and sub-goals. They were established through discussions with the client and other stakeholders, while also taking into account limitations identified during development.

Functional requirements cover what the system must be able to do, while non-functional requirements address aspects such as maintainability, security, performance, and deployment.

The requirements are prioritized using the MoSCoW method: *Must* (essential), *Should* (important but not essential), *Could* (desirable if time permits), and *Won't* (outside the project scope). These priorities indicate the importance of each requirement, regardless of whether it has been implemented.

Each requirement has an identifier that is also used in the design decisions, risk analysis, and evaluation, allowing decisions and findings to be linked back to the original requirements.

Functional Requirements

Table 5: Functional requirements and their traceability to sub-goals and stakeholder wishes.

Code	Role	Prio.	Requirement	SG	W
FR-01	Student	Must	Log in using a University of Twente account through the configured authentication mechanism.	SG-17	—
FR-02	Student	Must	View patient information for an assigned case, including the GP referral.	SG-12	W-12

Code	Role	Prio.	Requirement	SG	W
FR-03	Student	Must	Create and update notes, including clinical reasoning and differential diagnoses.	SG-03, SG-16	—
FR-04	Student	Must	Request additional diagnostic examinations.	SG-03, SG-06	—
FR-05	Student	Must	Provide motivation for each diagnostic request.	SG-06	—
FR-06	Student	Must	View diagnostic results made available after requests through a clear, interactive interface.	SG-13	W-07, W-11
FR-07	Student	Must	View supported medical images, including CT and thorax images, within the EPD.	SG-13	—
FR-08	Student	Must	View the history of student actions and clinical reasoning, including attribution and timestamps.	SG-05, SG-16	W-09
FR-09	Student	Must	Access only patient cases assigned to the student's group.	SG-17, SG-19	—
FR-10	Student	Must	Prevent viewing or modifying information belonging to other groups.	SG-17, SG-19	—
FR-11	Student	Must	Prevent students from modifying case configuration, official results, tutor decisions, or administrative information.	SG-17, SG-19	—
FR-12	Tutor	Must	View assigned student groups and patient cases through a navigable dashboard.	SG-01, SG-02	W-06
FR-13	Tutor	Must	Review diagnostic requests, including motivation and reasoning, with direct navigation from notifications.	SG-02, SG-03	W-06, W-08
FR-14	Tutor	Must	Approve or reject diagnostic requests and provide a message.	SG-02	W-06
FR-15	Tutor	Must	Provide examination results, reports, images, or other information to student groups.	SG-13	W-07
FR-16	Tutor	Must	Follow progress and clinical reasoning of assigned groups, including deadline information and activity timestamps.	SG-05, SG-14, SG-16	W-04, W-09
FR-17	Tutor	Must	Access only assigned student groups and patient cases.	SG-17, SG-19	—
FR-18	Tutor	Must	Restrict approval of requests and release of results to authorized tutors and administrators.	SG-07, SG-17	—
FR-19	Admin	Must	Create, edit, copy, configure, and manage patient cases.	SG-07, SG-10, SG-11	W-01
FR-20	Admin	Must	Create and manage student groups.	SG-04	—
FR-21	Admin	Must	Assign one or more patient cases to a student group.	SG-11	—
FR-22	Admin	Must	Assign tutors and students to groups or cases while preventing duplicate student assignments.	SG-04	W-05
FR-23	Admin	Must	Configure deadlines, stages, and milestones within a patient case.	SG-08, SG-14	W-02, W-04

Code	Role	Prio.	Requirement	SG	W
FR-24	Admin	Must	Manage patient information, examination results, documents, images, and files.	SG-10, SG-13	—
FR-25	Admin	Must	Configure which actions are performed automatically and which require tutor approval.	SG-08	W-02
FR-26	Admin	Must	Have access to the permissions required to manage all user roles.	SG-07, SG-17	—
FR-27	Admin	Should	Delete data with appropriate authorization and confirmation.	SG-19	—
FR-28	Admin	Should	Copy and reuse modules across cases and groups.	SG-11	—
FR-29	System	Must	Generate a printable overview of a group's work at the end of a case, using the agreed visual identity.	SG-05	W-14
FR-30	System	Must	Include clinical reasoning, requests, motivations, and results in the generated overview.	SG-05	W-14
FR-31	System	Must	Allow authorized staff to manage cases and educational content without changing source code; hard-coded procedure categories remain a documented limitation.	SG-07, SG-17	W-01
FR-32	System	Must	Allow case workflows and settings to be modified through configuration without changing source code.	SG-10, SG-17	W-01, W-02
FR-33	System	Must	Authenticate users through the University of Twente identity provider without storing UT passwords locally.	SG-17, SG-19	—
FR-34	System	Must	Prevent unauthorized session use and expire inactive sessions after a configured timeout.	SG-17, SG-19	—
FR-35	System	Must	Encrypt communication between users and the EPD.	SG-19	—
FR-36	System	Must	Securely store case information, user data, reports, and uploaded files.	SG-19	—
FR-37	System	Must	Restrict access to images, scans, reports, and cases according to user permissions.	SG-17, SG-19	—
FR-38	System	Must	Validate user input and uploaded content.	SG-16, SG-19	—
FR-39	System	Must	Enforce different permissions for each user role.	SG-07, SG-17	—
FR-40	Admin	Must	Allow a super-administrator to delete stored data according to the approved retention policy, including the proposed two-year period.	SG-19	—
FR-41	Admin	Must	Allow authorized teachers to view the complete patient case, including information not yet released to students.	SG-15	W-03
FR-42	Student	Must	Allow students to select individual blood values within a laboratory examination request.	SG-06	W-10

Code	Role	Prio.	Requirement	SG	W
FR-43	Admin	Must	Prevent duplicate procedure categories within the same stage when duplicates would cause requests to be handled incorrectly.	SG-08, SG-10	—
FR-44	Tutor	Must	Highlight requests outside the current stage or case and prevent approval of invalid requests.	SG-06, SG-08	—
FR-45	Tutor	Should	Highlight valid and invalid items individually within combined laboratory requests while allowing tutors to review the complete request.	SG-06	—
FR-46	System	Should	Display Dutch medical terminology agreed upon with the client.	SG-09	W-13
—	Student	Could	Provide students legal documents/info they might encounter in a real EPD in the program		
—	System	Could	Have a bright light pink and blue UI following the colors of the “Technishe geneeskunde” study		
—	System	Could	Be available for all days of the week		
—	System	Could	Provide UI color palates for the color blind		
—	System	Could	Provide text to speech		
—	System	Could	Have tablet and phone versions		
—	System	Could	Have emissions as low as possible		
—	System	Could	Provide statistics on the emissions made by the software		
—	System	Won’t	Automatically score the students on their task		
—	System	Won’t	Provide different colors for Medical data		
—	System	Won’t	Provide the ability to change text size		

Non-functional Requirements The non-functional requirements describe the quality attributes and operational constraints of the system. They cover maintainability, deployment, performance, security, privacy, and long-term operation.

Each requirement references the sub-goals it supports and includes a proposed method of verification. Targets that have not yet been agreed upon with the client are identified as provisional. Listing a requirement does not imply that it has already been implemented or verified.

Table 6: Non-functional requirements and their traceability to sub-goals and stakeholder wishes.

Code	Requirement	Prio.	Verification	SG	W
NFR-01	A Technical Medicine maintainer must be able to deploy the system using documented instructions without assistance from the original developers.	Must	Independent deployment test	SG-18	—

Code	Requirement	Prio.	Verification	SG	W
NFR-02	Documentation must cover architecture, installation, configuration, database structure, backups, recovery, and troubleshooting.	Must	Documentation review	SG-18	—
NFR-03	Authorized staff must be able to modify ordinary case content and stage settings without changing source code; hard-coded procedure categories remain a documented limitation.	Must	Configuration tests	SG-10, SG-17	W-01
NFR-04	The system should function correctly in the desktop browsers and versions agreed upon with the client.	Should	Cross-browser tests	SG-09	—
NFR-05	Representative pages should load within 6 seconds under an agreed test workload. This target is provisional.	Should	Performance measurements	SG-01, SG-02	—
NFR-06	All protected resources, including medical images and exported documents, must enforce server-side authorization.	Must	Authorization tests	SG-17, SG-19	—
NFR-07	Inactive sessions must expire after a configured timeout, and University of Twente passwords must not be stored locally.	Must	Authentication and session tests	SG-17, SG-19	—
NFR-08	Stored data must be removable according to the approved retention policy, including the proposed two-year retention period and applicable exceptions.	Must	Retention and deletion tests	SG-19	—
NFR-09	The database and uploaded files must be recoverable from backups within recovery targets agreed upon with the client.	Must	Backup restoration test	SG-16, SG-19	—
NFR-10	Sensitive educational and patient-like data must be minimized or anonymized according to the agreed privacy policy.	Must	Privacy and data review	SG-19	—
NFR-11	The system should remain maintainable for an intended service life of five years, with documented dependencies and update procedures. This target is provisional.	Should	Maintainer review	SG-18	—
NFR-12	Hosting responsibilities and required availability during teaching activities must be documented and agreed upon with the client.	Must	Operational acceptance review	SG-18	—

3.6 Research Questions

During the project, several design questions came up regarding the interfaces, case management, security, and system architecture. These were used to guide the design choices made during development.

To answer these questions, we compared different options, tested implementations, and discussed our choices with the client and teaching staff. The results are discussed later in this report.

Table 7: Research questions.

RQ	Research question
RQ-01	How can the interfaces support the workflows of students, tutors, and teachers while keeping navigation and information access straightforward?
RQ-02	How can patient cases, groups, and educational content be configured and reused without making administration difficult for teaching staff?
RQ-03	How can the system enforce stage progression and examination-request rules while supporting the educational workflow?
RQ-04	How can the system manage access permissions, data consistency, and retention while allowing users to perform their assigned tasks?
RQ-05	How can the system architecture and deployment process support maintenance and handover to Technical Medicine staff?

3.7 Design Log

A complete chronological Design Log is provided in Appendix G. The design log records significant design decisions and revisions made during the project. It was reconstructed from the documented requirements meetings, project meetings, and prototype feedback. Where the available documentation does not show that a final decision was made, the item is explicitly marked as provisional or open. The log therefore distinguishes stakeholder input and proposed changes from confirmed design decisions.

3.8 Risk Analysis

We analysed possible risks in both the system and the educational workflow to help decide between design options. This included a SWOT analysis (Appendix C), a stakeholder analysis (Appendix B), and a Failure Mode and Effects Analysis (FMEA).

In the FMEA, we listed possible failure modes, their consequences, and their possible causes. Each failure mode was scored based on severity (S), occurrence (O), and detectability (D). The Risk Priority Number (RPN) was calculated as:

$$RPN = S \times O \times D \tag{1}$$

Each factor was scored from 1 to 10. A higher severity score means a more serious consequence, while a higher occurrence score means the failure is more likely to happen. A higher detectability score means the failure is harder to detect before it causes problems.

The scoring criteria are shown in Table 8. The scores are estimates used to compare risks, not measured probabilities.

When a failure mode affected a design decision, we referenced the related DEC or design-log entry. Some decisions reduced a risk without removing it completely, so the remaining risks were kept in the analysis.

Table 8: Scoring criteria for the Failure Mode and Effects Analysis.

Score	Severity (S)	Occurrence (O)	Detectability (D)
1–2	Little or no impact on functionality or workflow	Unlikely to occur during normal use	Almost always detected before affecting users
3–4	Minor disruption with an easy workaround	Occurs occasionally under specific conditions	Usually detected during normal use or testing
5–6	Disrupts a user task or educational workflow	Could occur during regular use	May go unnoticed without specific checks
7–8	Major disruption, loss of functionality, or incorrect data	Likely to occur under certain conditions	Difficult to detect before affecting users
9–10	Critical security, privacy, data-loss, or workflow failure	Occurs frequently or is almost unavoidable	Very unlikely to be detected before affecting users

Table 9 shows three examples from the FMEA. FM-20 addresses teachers not being able to view the full patient case. FM-23 covers differences between the mock API and the production backend, which led us to remove the mock API setup before preparing the final product. FM-67 concerns laboratory requests that contain both valid and invalid items. We added visual indicators to distinguish these items, but whether tutors should be allowed to approve mixed requests still needs to be decided with the client.

The full FMEA, including all failure modes and their links to the design decisions and design-log entries, is provided in Appendix D. The table is kept in the appendix because its length would interrupt the main text; the discussion above shows a short what it contains and how to read it.

Risk Register The risk register combines related failure modes from the FMEA and includes broader project risks, such as delays and integration problems. It also records risks identified while comparing design options.

Each risk is linked to the relevant failure modes, requirements, or options where applicable. The register describes what could go wrong, the possible consequences, and how we planned to reduce the risk.

Table 9: Representative rows from the FMEA, showing the structure of the full table in Appendix D.

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log
FM-20	Teacher overview	Teacher cannot see the full case	Cannot coach or assess the group	8	7	4	224	DEC-49 (full-case overview); D17
FM-23	Mock mode	Mock diverges from backend	Services pass in mock, fail in production	6	4	3	72	DEC-87; D59 (mock setup removed for final product)
FM-67	Mixed laboratory requests	Request contains both in-case and out-of-case procedures	Blanket accept/reject may treat valid and invalid items alike	6	5	5	150	DEC-85; D63 (highlight in-case and out-of-case items)

Table 10: Risk register with sources, consequences, and proposed controls.

ID	Source type	Risk	Related references	Potential consequence	Response / control
RISK-01	FMEA	Navigation and layout do not adequately support user tasks.	FM-08–FM-15, FM-17, FM-64	Users may overlook information or spend unnecessary time navigating between screens.	Review representative workflows with users and test navigation between groups, cases, and requests.
RISK-02	FMEA	Examination requests or stage progression behave incorrectly.	FM-01–FM-07, FM-21, FM-65–FM-67	Students may bypass learning stages, receive incorrect results, or have invalid requests approved.	Validate stage and case membership; test duplicate procedures and mixed laboratory requests.
RISK-03	FMEA	Deadlines and teacher oversight are insufficiently visible.	FM-18–FM-20	Teaching staff may miss overdue work or lack information needed to guide students.	Test deadline visibility and full-case access with representative teacher workflows.

ID	Source type	Risk	Related references	Potential consequence	Response / control
RISK-04	FMEA	Activity history or request-result associations are incomplete.	FM-16, FM-42–FM-47	Clinical reasoning records may become misleading or difficult to reconstruct, and previously approved results may become inaccessible after a procedure is updated.	Preserve request identifiers and associations between requests and results. Test whether approved results remain accessible when case procedures are edited.
RISK-05	FMEA	Group, module, case, or form state becomes inconsistent.	FM-06, FM-27–FM-33	Data may be assigned incorrectly or changes may be lost when switching between screens.	Validate form state and test creation, editing, switching, and assignment workflows.
RISK-06	FMEA	Frontend and backend behaviour or data contracts diverge.	FM-22–FM-26, FM-55, FM-59	Features may work in development but fail when integrated with the real backend.	Test against the real backend and verify authentication, data formats, and procedure categories.
RISK-07	FMEA	Exported dossiers differ from the information shown in the system.	FM-35–FM-37, FM-40–FM-41, FM-53–FM-54, FM-56, FM-58	Exported records may be incomplete, outdated, or difficult to read.	Compare generated PDFs with the corresponding case information and test supported log entries.
RISK-08	FMEA	Files or exported data are accessed by unauthorized users.	FM-38, FM-86–FM-87; FR-09, FR-10, FR-37, FR-39	Medical files or information belonging to another group may be disclosed.	Enforce server-side authorization, restrict file-system access, and test access across groups and roles.

ID	Source type	Risk	Related references	Potential consequence	Response / control
RISK-09	FMEA	Medical images or examination results are unavailable, unreadable, or incorrectly combined.	FM-07, FM-52, FM-60–FM-61, FM-91	Students may receive incorrect images or miss diagnostic information needed for the case.	Test representative DICOM files, including bulk uploads of separate files with matching resolutions, and verify the resulting image groups.
RISK-10	Project	Remaining implementation or architectural rework exceeds the available project timeframe	FM-34, FM-62–FM-63, FM-79	Required functionality may remain incomplete, or existing features may regress near delivery	Freeze scope for the delivery period; prioritise mandatory FRs; defer high-rework changes unless explicitly re-approved.
RISK-11	Project	Late requirement changes increase the project scope.	FM-34, FM-62 (related context)	Rework may reduce the time available for integration and validation.	Record change requests, assess their impact, and agree priorities with the client.
RISK-12	Project	External dependencies or environment assumptions delay integration.	FM-22–FM-24, FM-59 (related context)	Backend, authentication, or deployment issues may block progress.	Verify integration assumptions early and document unresolved dependencies.
RISK-13	Project	Insufficient time remains for final testing.	FM-22, FM-38, FM-40, FM-56 (related context)	Permission, integration, or export defects may remain undetected at delivery.	Reserve time for end-to-end testing and prioritize security and data-integrity checks.
RISK-14	Option analysis	Exported learning records may be modified after generation.	—	Edited copies may no longer match the original system-generated dossier.	Prefer a fixed-layout export and retain the authoritative record within the system.

ID	Source type	Risk	Related references	Potential consequence	Response / control
RISK-15	FMEA	File storage becomes inconsistent or server resources are exhausted.	FM-88–FM-90	Incomplete or unregistered files may remain on the server, while excessive uploads may reduce system availability.	Remove files after failed operations and enforce upload size limits.

4 Design

During development, we made design choices based on the requirements, feedback from stakeholders, and problems found during implementation. Where multiple solutions were possible, we compared the available options before choosing an approach.

Not every decision came from the FMEA. Some were made to meet specific requirements or deal with technical limitations.

4.1 From Problems to Design Options

We considered different solutions when a requirement could be implemented in multiple ways or when a problem was identified during development.

For example, FM-64 concerned the amount of information shown on the tutor dashboard. We considered a split-screen layout, pop-ups, and separate pages for groups and cases. We chose separate pages to keep the interface manageable while allowing tutors to navigate between related information.

Other decisions involved teacher access to patient cases, deadlines, examination requests, and medical results.

The options and their comparisons are recorded in the Options tables. For rejected options, we documented why they were not chosen, referring to relevant risks, requirements, or technical constraints where applicable.

4.2 Scoring the options

We scored each option against the solution metrics in Table 11. The metrics are scored on a consistent 1–5 scale (1 = poor fit, 5 = strong fit), and each metric has a weight that reflects how important it is for this project. The weighted total of an option is $\sum_i (\text{score}_i \times \text{weight}_i)$, and the same scale and weights were used for all options that were meaningfully comparable.

The weights in Table 11 were used as the default for comparing design options. When a metric did not apply or a comparison needed different priorities, we documented the adjusted criteria and weights.

The full scoring results can be found in Appendix F. The Options and DEC tables show which alternatives were considered, which were chosen, and why the others were rejected.

Table 11: Solution metrics used to compare design alternatives.

Code	Metric	Weight
SM-01	Feasibility within the available development time	5
SM-02	Maintainability and ease of future modification	5
SM-03	Usability for the intended users and tasks	4
SM-04	Accessibility of the interface and interactions	3
SM-05	Sustainability and resource efficiency	1
SM-06	Relative implementation and operational cost	5
SM-07	Scalability and ability to support future changes	3
SM-08	Testability and ease of verifying correct behaviour	4
SM-09	Consistency with the existing architecture and interface	3

4.3 From scored options to design decisions

After comparing the design options, we recorded the selected solution for each design area in the DEC table (Appendix E). Each entry includes the chosen option, the rejected alternatives, and the reasons behind the decision.

Where relevant, decisions are linked to failure modes in the FMEA and changes recorded in the design log. We also kept the rejected options and their reasons so that the alternatives can still be reviewed.

The following sections describe the backend design (Section 4.4) and web-app design (Section 4.5), including the options considered and the decisions made.

4.4 Backend Design

Backend Class Diagram Figure 1 shows backend design used in the begining this project. The model separates user roles, the educational structure, and the entries that form a group's electronic patient dossier.

The role model is cumulative. A `Student` stores its identity, note, and assigned group and can submit a diagnostic `Request` with clinical reasoning. A `Tutor` adds the ability to review such requests, while a `Teacher` adds administrative operations for creating and editing modules, groups, and patient cases and for assigning students and tutors. This hierarchy gives the teacher role the capabilities of the lower roles without duplicating their methods.

The educational structure is centred on `Module`, `Group`, and `PatientCase`. A module contains the groups and patient cases belonging to an educational period and specifies whether responses may be automated. A group belongs to one module, contains students and supervisors, and works on one patient case. It also maintains an ordered dossier log. A patient case stores the case description and patient characteristics and manages the diagnostic results that can be released to a group.

The dossier workflow is represented by `Request`, `Review`, and `Result`. These share the `Recordable` abstraction, which provides a name, origin, and timestamp. A request records the requested test, the student's reasoning, and the group that submitted it. A

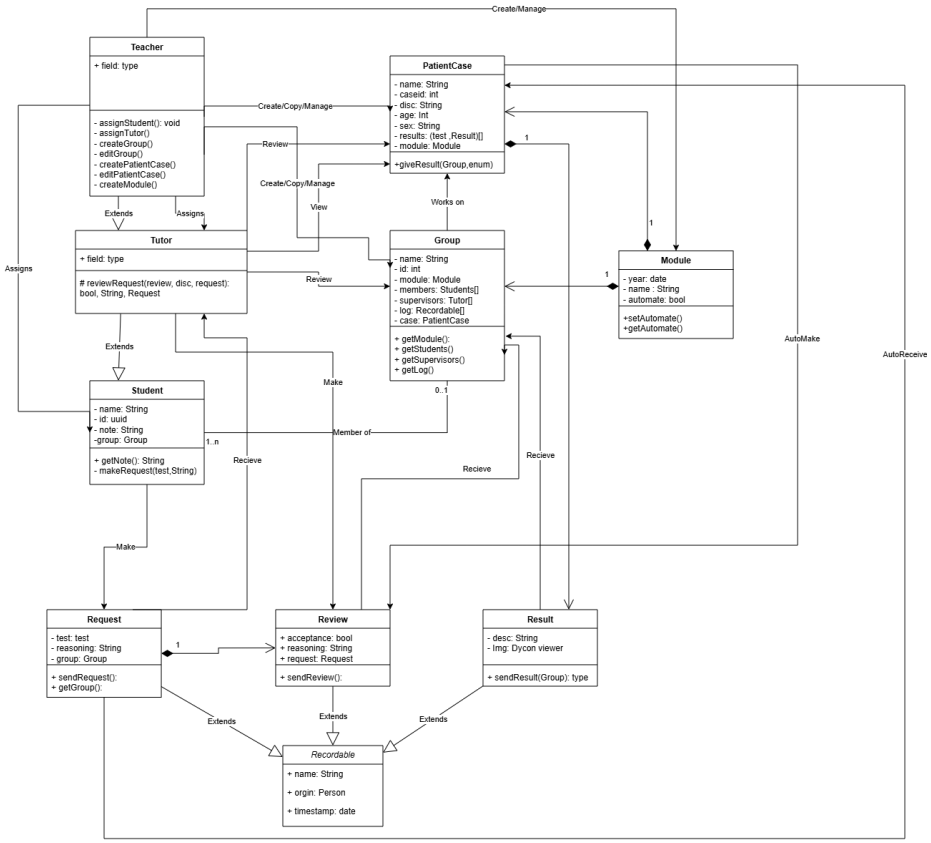


Fig. 1: Backend class diagram.

review records a tutor's acceptance decision and feedback and retains a reference to the original request. A result stores its description and an optional medical-image viewer reference. Sending any of these records adds it once to the relevant group's log, creating a chronological and traceable overview of the students' diagnostic process.

Login and Accounts The backend needs to represent different kinds of user accounts while still sharing the common account data and behavior between them. Both students and teachers need the same core account information, such as name, email, and authentication data, but they also have role-specific properties and permissions. Because of this, the system needs an account model that supports shared functionality while still allowing specialization where needed.

Table 12: Options for authentication and account hierarchy.

OPT	Area	Description	Why not	Status
OPT-85	Development identity provider	Create our own login system	Would require implementing and maintaining authentication ourselves, adding security and testing work before the application can be tested (FM-83, FM-92, RISK-13).	Rejected
OPT-86	Development identity provider	Run Keycloak locally as an OIDC provider with controlled test accounts	—	**Chosen**
OPT-118	Development identity provider	Connect directly to UT's Microsoft authentication during development	Requires university access and app registration before the full login flow can be tested independently. Microsoft remains the intended production provider (FM-92, RISK-12).	Rejected
OPT-119	Microsoft migration	Adapt the existing configurable OIDC integration to connect directly to UT's Microsoft authentication	—	**Chosen**
OPT-120	Microsoft migration	Replace the existing OIDC integration with a separate Microsoft-specific login implementation	Requires replacing and retesting the existing authentication integration, adding implementation work during handover (FM-93, RISK-10).	Rejected
OPT-121	Microsoft migration	Keep Keycloak in production as an identity broker connected to UT's Microsoft authentication	Technically viable, but adds a separately operated identity service with its own configuration, database, backups, and updates (FM-93, NFR-01, NFR-11, RISK-12).	Rejected

OPT	Area	Description	Why not	Status
OPT-87	Account hierarchy	Subclass hierarchy	—	**Chosen**
OPT-88	Account hierarchy	Individual account classes	Would be less consistent with repeated functions.	Rejected

Keycloak was selected as the development identity provider (OPT-86, DEC-102). It allows the team to create test accounts and test the OIDC login flow without depending on access to the university's authentication services (FM-92). Developing a separate login system (OPT-85) would add authentication and security work, while connecting directly to UT's Microsoft service during development (OPT-118) would depend on university access and app registration being available. Microsoft remains the intended production provider; the Keycloak choice applies to development and does not replace the university-login requirement (FR-01, FR-33).

For the production handover, the selected approach is to adapt the existing configurable OIDC integration to connect directly to UT's Microsoft authentication (OPT-119, DEC-105). UT must register the application and provide the approved tenant and connection settings. The issuer, client ID, client credentials, redirect URI, and scopes are supplied through environment variables. The migration also requires checking token exchange, identity-claim mapping, and logout against Microsoft's behaviour; changing the configuration alone does not establish compatibility (FM-93). In particular, the current email-based account linking and expectation of an `email_verified` claim must be reviewed with UT before production use. Application roles remain controlled by the application database. Login, account linking, logout, and access for each role must be tested before the switch. This is a handover plan and has not yet been validated against UT's Microsoft tenant.

We chose a subclass hierarchy because it keeps the shared account fields in one base class while still allowing tutors and teacher accounts to extend it with their own specific data and behavior. This makes the code easier to maintain and reduces duplication. It also fits the rest of the backend design, since role-specific logic can be handled through inheritance instead of being spread across separate unrelated classes.

Using individual account classes was rejected because it would duplicate shared logic and make updates harder to manage. It would also make the account model less consistent, since common functionality would need to be repeated in multiple places.

Requesting Workflow The backend must support two request-related actions: students sending procedure requests and teachers reviewing those requests. Each request needs to be logged so the group's actions can later be shown in a timeline. The system also needs a way to store the request status together with the request data, while keeping the implementation maintainable and the database queries efficient. This is done to give the teacher the ability to revise and review and alter it later if needed. In addition, if a module is automated, the backend must be able to handle requests automatically without teacher review. More on this in the automation section.

Table 13: Options for request handling.

OPT	Area	Description	Why not	Status
OPT-80	Request Handling	Attach request status to log	Complicates code by requiring more and bigger database queries and more filtering	Rejected
OPT-81	Request Handling	Attach request status to linking table	—	Chosen
Chosen				
OPT-83	Request Handling	Completely new data object	Log and process data always needed when requesting and reviewing. Makes DB and code unnecessarily bigger	Rejected
OPT-84	Request Handling	Attach request status to group	Would limit the group to only 1 request until the request is handled.	Rejected

We chose for a linking table between logs and procedures because it keeps the request status tied to the request without overloading the log entity itself (OPT-81). This makes the design cleaner and avoids the extra queries that would be needed if the status of a request were stored directly on the log. It also keeps the request-related data together in one place, which improves maintainability and makes it easier to work with both the log and the procedure when handling requests.(DEC-90)

A separate request data object was rejected because it would still require fetching the log and procedure separately, while also adding unnecessary extra structure to the backend. (OPT-83) Attaching the status to the group was also rejected, since a group can have multiple requests at the same time and this solution would block that.(OPT-84) Finally, the log-based solution was rejected because it would make the database queries more complex and less efficient. (OPT-80)

File support The backend uses a dedicated `FileRoute` for uploading, retrieving, and deleting medical image files (OPT-108, DEC-97). Keeping file operations in a separate route centralizes authentication and authorization checks and prevents the front-end from accessing the underlying file system directly. Tutors and administrators can upload and delete files, while students can only retrieve files belonging to procedures and stages they are authorized to access.

File paths are restricted to a configured storage directory (OPT-110, DEC-98). Incomplete uploads and files that fail database registration are removed (OPT-112, DEC-99). A maximum upload size is enforced to prevent excessive resource usage (OPT-114, DEC-100).

DICOM Processing A DICOM examination may consist of multiple files representing individual slices of the same scan. The backend groups these files according to their `SeriesInstanceUID` (OPT-116, DEC-101). Slices belonging to a compatible series are ordered and combined into a multi-frame DICOM object when requested by the

frontend. The implementation checks relevant image properties before merging and rejects incompatible series rather than producing an invalid image. This lets the original slices remain separately stored while providing the frontend with a coherent image stack for display in the Cornerstone3D viewer.

known limitation: the backend currently groups them based on image resolution. This means that if dcm files belonging to different series may be grouped together if their resolutions match. this does not apply to multi-set files which contain multiple image sets.

Table 14: Options for backend file handling.

OPT	Area	Description	Why not	Status
OPT-108	File access	Use a dedicated `FileRoute` with authentication and authorization checks.	—	chosen
OPT-109	File access	Serve stored files directly through static file URLs.	Does not provide the required per-user and per-group authorization checks, potentially exposing medical images to unauthorized users (FM-86, RISK-08).	rejected
OPT-110	File-system access	Resolve and normalize file paths against a configured storage root.	—	chosen
OPT-111	File-system access	Use supplied file paths without validating their location against the storage root.	Manipulated paths could access files outside the intended storage directory (FM-87, RISK-08).	rejected
OPT-112	File consistency	Remove incomplete uploads and stored files whose database registration fails.	—	chosen
OPT-113	File consistency	Leave stored files unchanged when upload or database registration fails.	Failed operations could leave incomplete or unregistered files on the server, causing inconsistencies between the database and file storage (FM-88, FM-89).	rejected
OPT-114	Upload limits	Enforce a maximum file upload size through the `FileRoute`.	—	chosen
OPT-115	Upload limits	Allow uploads without an application-level file size limit.	Excessively large files could exhaust server resources and reduce system availability (FM-90).	rejected

OPT	Area	Description	Why not	Status
OPT-116	DICOM processing	Group files by `SeriesInstanceUID` and validate image compatibility before combining slices.	—	chosen
OPT-117	DICOM processing	Combine uploaded DICOM slices without checking series identity and image compatibility.	Incompatible slices could produce unreadable or incorrect medical images (FM-91, RISK-09).	rejected

Automation The backend also needed to support automated handling of requests in modules where a tutors review is not available. This means the system must be able to decide whether a request is accepted or rejected automatically and, if accepted, continue with the next stage of the case when appropriate. The system also needs to check for deadlines and start dates. When a teacher creates a case they have the option to add start dates and deadlines to a stage. When a deadline is implemented the system need to advance every group that is still behind a stage after the deadline and stop groups from advance before the start date

After a request is created by a student, the backend checks whether the module is automated. If it is, the request is processed automatically and either accepted or rejected without tutor review. This was implemented by checking the automation flag in the Module object, since the module is already available when handling the request and because teachers may want all cases in a module to be automated together (DEC-94). Checking if the given answer is correct is done by storing and retrieving the correct procedure in the DB. The correct procedures are decided by a Teacher during case creation. The automated response given back to the student clarifies whether the request was rejected or accepted and states that this review was an automatic review.

If a request is accepted , either automatically or by a tutor, the backend also checks whether the group advances to the next stage of the case. This is done by taking all correct procedures of the stage the group is currently in and the accepted requested procedures the group has made in that stage. Then the group will advance one stage in the case and be notified of the stage advancement,if the accepted requested procedures contain all correct procedures and the start date for the next stage has been passed. (DEC-95) Deadlines work somewhat similarly. In the code there is a scheduler that checks if a deadline has been passed every 2 hours. If a deadline has passed for a stage since last check the system checks every group that's working on that case and sees if they are in that stage. If a group still is in that stage, they are immediately send to next stage and are notified of the stage advancement. That they missed a deadline is also logged and stored for the PDF export.

Technical Architecture The technical architecture was divided into two main decisions: the database and the backend framework. The alternatives for each were compared based on the project's requirements, available development time, and the team's experience.

Database The database structure is shown in Figure 2.

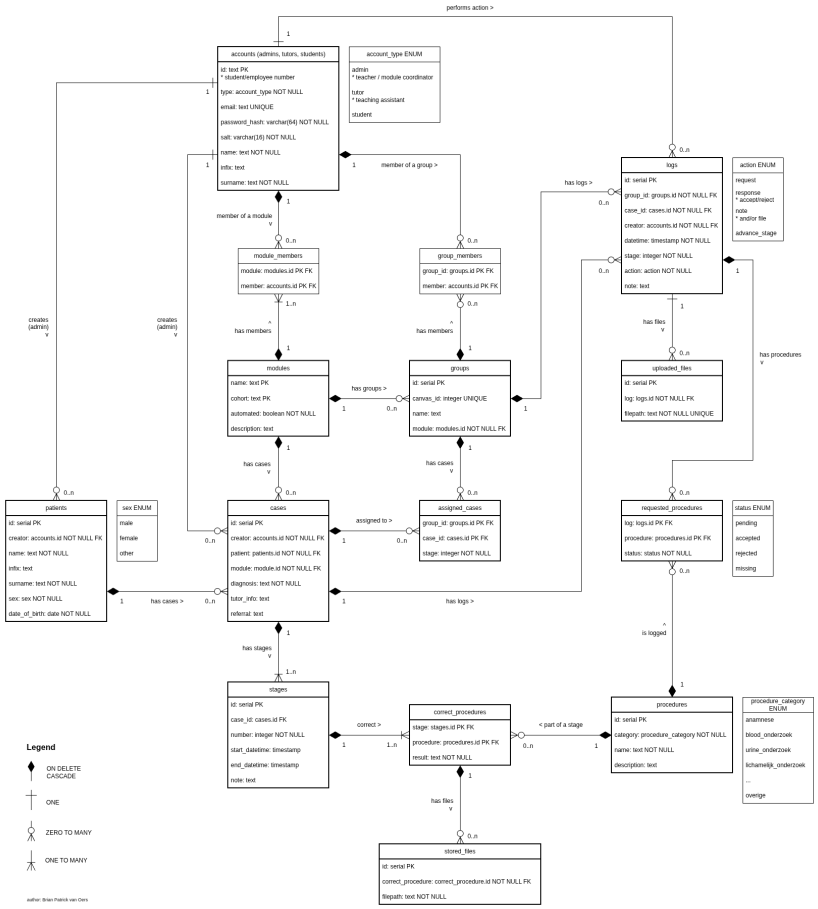


Fig. 2: Database diagram.

The database alternatives are listed in Table 15. PostgreSQL was selected because the team was already familiar with it, reducing the time needed for implementation. The decision is documented in DEC-91.

Table 15: Options for the database.

OPT	Area	Description	Why not	Status
OPT-89	Database	PostgreSQL	—	chosen

OPT	Area	Description	Why not	Status
OPT-90	Database	MySQL	Less familiar to the team than PostgreSQL, increasing implementation time (RISK-10).	rejected
OPT-91	Database	SQLite	Less familiar to the team, while PostgreSQL already met our requirements (RISK-10).	rejected
OPT-92	Database	Microsoft SQL Server	PostgreSQL was already familiar to the team, while SQL Server would require additional setup and evaluation of edition limitations (RISK-10, RISK-12).	rejected
OPT-93	Database	Oracle Database (SQL)	Offered no clear advantage over PostgreSQL and would require additional familiarisation and setup (RISK-10, RISK-12).	rejected
OPT-94	Database	MongoDB (NoSQL)	Its document-oriented model offered no clear advantage for the relational data in the EPD (RISK-10). ACID properties [1]. A reported security vulnerability was also considered [2].	rejected
OPT-95	Database	CouchDB (NoSQL)	Its document-oriented model offered no clear advantage for the relational data in the EPD (RISK-10). referenced ACID properties [1].	rejected

Backend Framework The backend framework alternatives are listed in Table 16. Jakarta EE was selected primarily because of the team's existing experience with Java and the framework. This reduced the amount of time needed to learn a new technology. The decision is documented in DEC-92.

An email server was also considered but was not implemented because of the additional development time required.

Table 16: Options for the backend framework.

OPT	Area	Description	Why not	Status
OPT-96	Programming Language / Framework (server)	Jakarta EE (Java)	—	chosen

OPT	Area	Description	Why not	Status
OPT-97	Programming Language / Framework (server)	Spring Boot (Java)	Less familiar to the team than Jakarta EE, increasing implementation time (RISK-10).	rejected
OPT-98	Programming Language / Framework (server)	Flask (Python)	Required switching to a less familiar language and framework without a clear benefit for this project (RISK-10).	rejected
OPT-99	Programming Language / Framework (server)	Django (Python)	Less familiar to the team and would require additional time to learn and integrate (RISK-10, RISK-12).	rejected
OPT-100	Programming Language / Framework (server)	Laravel (PHP)	Required a less familiar language and framework, increasing implementation time (RISK-10). Reported PHP vulnerabilities were also considered [6,7].	rejected
OPT-101	Programming Language / Framework (server)	.NET (C#)	Less familiar to the team than Java and Jakarta EE, increasing implementation time (RISK-10).	rejected

4.5 Web-App Design

by N. van der Laan & J. Grimberg

The Web-App is the part of the EPD that the student, tutor and teacher actually work with, so the design was driven by the workflow of those roles rather than the backend specifications. We designed it using `Balsamiq` wireframes, which let us put a screen in front of the client quickly and change it on the spot when the feedback asked for it. These were used to agree on the layout and interactions before any code was written. the final choices are described below

For each major design area, we considered several options before deciding which one to use. These options are listed in the OPT tables below, including the reason why certain options were rejected. The final decisions are recorded in the the DEC table in Appendix E, together with the related requirements, stakeholder wishes, and risks. The chosen options are highlighted in green. The wireframes can be found in Appendix A

Students The student interface was designed around a two-bar layout (see Figures 3 and 4 in Appendix A): a collapsible navigation bar on the left for modules, patients, requested procedures and export, and a collapsible bar on the right for a short summary of the current patient, requesting procedures and adding notes. This lets a student reach every function from any page they are working in. The vertical expandable bar (OPT-03) was chosen because students mostly work on laptops, which often do not have the screen space for many menu items; a horizontal bar (OPT-01) takes too much of that

space, a fixed vertical bar (OPT-02) cannot grow with the information structure, and the circle-to-ring menu (OPT-04) is too unfamiliar to be discoverable. This decision is recorded as DEC-01 in Appendix E.

Table 17: Options for navigation and layout.

OPT	Area	Description	Why not	Status
OPT-01	Menu's	Horizontal bars	Consumes scarce screen space, increasing the chance that users miss content or need extra navigation (RISK-01)	Rejected
OPT-02	Menu's	Vertical bars	Cannot scale to many sub-categories; users may lose access to navigation choices as the information structure grows (RISK-01)	Rejected
OPT-03	Menu's	Vertical expandable bar	—	**Chosen**
OPT-04	Menu's	Expandable circle-to-ring	Unfamiliar interaction can make navigation hard to discover and increase user error (RISK-01)	Rejected
OPT-05	Tutor dashboard	Split screen	Too much information remains visible on one page, so tutors can lose overview and struggle to find the relevant group or case (RISK-01)	Rejected
OPT-06	Tutor dashboard	Separate pages with click-through navigation to the selected group/case	—	**Chosen**
OPT-07	Tutor dashboard	Pop-up	Adds interaction state and can obscure the dashboard context, increasing missed or inconsistent tutor actions (RISK-01)	Rejected
OPT-08	Requests & notes	Separate page	Requires switching away from the current case view, interrupting the student's workflow and adding navigation steps while working on a case (RISK-01)	Rejected
OPT-09	Requests & notes	Pop-up	—	**Chosen**
OPT-10	Requests & notes	Splitscreen	Divides limited viewport space, making requests and notes harder to read together (RISK-01)	Rejected

Procedure Requests and Notes The student needed to be able to request procedures and take notes while working on a case. To make the workflow as easy as possible,

constant access to both functions was considered optimal. Based on the assumption that most students work on laptops, a pop-up (OPT-09) was chosen over a separate page (OPT-08) and a splitscreen (OPT-10): a separate page interrupts the case view, while a splitscreen divides the limited viewport. The relevant options are evaluated in Table 17 (OPT-08–OPT-09). To mitigate the failures and risks of pop-ups, the technical limitations of when the pop-up closes and when the user can leave it were defined explicitly in the implementation. This decision is recorded as DEC-03 in Appendix E.

DEC-03 uses OPT-09 for the student request and note pop-up. DEC-14 concerns the tutor review page, which follows the separate-page tutor navigation in OPT-06. The student request input and tutor review therefore use different interfaces, as documented in Appendix E.

Tutor Dashboard The tutor dashboard (Figure 5 in Appendix A) provides everything a tutor needs at the click of a button: the module they are working on, the groups they have been assigned, the ability to accept or reject student requests, and the log of each group. The tutor only needs to see three main things: the group/module they are looking at, the request list, and the log. This means the user does not need much work space, and their main objective is to quickly navigate between groups. That is why the left side of the screen is used for navigation and the right side for both the giving notes to groups and their respective logs. The relevant options are evaluated in Table 17 (OPT-05–OPT-07), and the decision is recorded as DEC-02 in Appendix E.

Group Creation An admin needs to be able to assign people, tutors, and cases to groups (see Figure 6 in Appendix A). This can be a time-consuming task and should be made as easy to use as possible for the user. Because the main priority is to quickly assign groups, drag & drop within a split screen was chosen. This allows the user to quickly adjust groups and create them. This is done after a teacher creates a module and imports a group CSV file from canvas allowing for quick population of the groups while only needing minimal manual adjustments. This decision is recorded as DEC-15 in Appendix E.

Table 18: Options for group management and admin features.

OPT	Area	Description	Why not	Status
OPT-11	Grouping	Drag & drop split screen	—	**Chosen**
OPT-12	Grouping	Create forms for groups	More manual steps increase setup time and make group changes harder to maintain within the project scope (RISK-05, RISK-10)	Rejected
OPT-13	Grouping	Split screen (no D&D)	Requires more manual selection and movement, increasing setup effort and the chance of assignment mistakes (RISK-05)	Rejected

OPT	Area	Description	Why not	Status
OPT-14	Grouping	Student-focussed grouping	Adds implementation and validation work that threatens completion of mandatory functionality within the deadline (RISK-10, RISK-11)	Rejected
OPT-59	Procedure selection within a stage	Prevent selecting the same procedure type more than once in one stage	—	**Chosen**
OPT-60	Tutor request validation	Highlight requests outside the current stage or case and disable Accept	—	**Chosen**
OPT-61	Tutor request validation	Allow acceptance despite stage/case mismatch	Can approve an invalid request and produce results inconsistent with the configured case (RISK-02)	Rejected
OPT-62	Mixed laboratory requests	Block the whole request if any item is outside the case	Could block valid in-case procedures before a mixed-request policy is agreed (RISK-02)	Rejected
OPT-63	Mixed laboratory requests	Keep Accept available and highlight in-case versus out-of-case items	—	**Chosen**
OPT-106	Group Management	Import student-group assignments from a CSV file to populate multiple groups at once.	—	chosen
OPT-107	Group Management	Manually assign students to groups through the interface.	Requires repetitive work for teachers and increases the risk of incorrect student-group assignments.	rejected

For the procedure-selection and request-validation choices, we prevent the same procedure type being selected twice in one stage (OPT-59) so a stage stays a clear set of examinations; this is recorded as DEC-23 in Appendix E. We also highlight requests outside the current stage or case and disable Accept (OPT-60), because allowing acceptance anyway (OPT-61) could produce results that do not match the stage of a case; this is recorded as DEC-84 in Appendix E. For mixed laboratory requests we keep Accept available and highlight in-case versus out-of-case items (OPT-63), since blocking the whole request (OPT-62) could stop valid in-case procedures from being approved; this is recorded as DEC-85 in Appendix E, which is explicitly marked as a decision pending a teaching-policy clarification. In addition the ability to import a group CSV file from canvas was added to spare teachers time to populate groups in the system(OPT-106) this is to reduce the strain on the teacher to manually populate all the groups (DEC-96).

User Management The user management screen shows the accounts connected to the selected module in a table containing each person's name, ID, email address, and role. Administrators can search by name, ID, or email address and change an account's role from student to tutor or vice versa. Role changes remain pending until the administrator selects Save, after which the application asks for confirmation before applying them.

Role changes were initially added because the imported CSV data did not distinguish tutors from students. Although that is no longer an issue, we still considers the feature useful to retain. The pending-save pattern was chosen primarily because it was straightforward to implement. (OPT-68, DEC-104).

Table 19: Options for changing user roles.

OPT	Area	Description	Why not	Status
OPT-68	Role changes	Stage role edits and apply them together using an explicit Save action	—	**Chosen**
OPT-69	Role changes	Apply each role change immediately when its dropdown is changed	Individual changes take effect before the administrator can review the full set	Rejected
OPT-70	Role changes	Open a separate edit dialog for each account and save changes one at a time	Adds steps when an administrator needs to change several accounts	Rejected

Case Creation An admin needs to be able to create and edit new cases and patients (see Figure 7 in Appendix A). This must include all possible procedures a student can request, in addition to their answers. The product also needs to accept pictures and DICOM files. DICOM is a file type used by DICOM viewers that a user can scroll through and is commonly used in the medical field. The case also needs to have stages, so a student can work through a case in a defined manner and cannot request procedures that are not yet available or appropriate.

The fastest way to create a case would be uploading a file and reading it, but this is not possible within the scope of the project. In addition, one of the client's main priorities is a system that works without the need for an IT person. Automatically importing cases would also introduce another part of the system that staff might need technical help to maintain or troubleshoot. This is something we wanted to avoid. That is why case creation is left to the user and the scope of what they can add is made as broad as possible. The decision to combine case creation and editing into one page is recorded as DEC-21 in Appendix E, and the case start state (Stage 1 with an empty procedure list) is recorded as DEC-35 in Appendix E.

The case form collects the patient's name, sex, and date of birth, along with the case diagnosis and referral. It also provides a separate tutor-information field, which is labeled as visible only to tutors. Administrators can add and remove stages and add

or remove procedures within each stage. The form starts with the first stage. Patient details and the diagnosis are required before the case can be saved, and saving requires confirmation.

Examinations, Case Progression, and Results Examination requests and results are central to how students use the application. We considered different ways to select procedures, show case progression, and display results.

Table 20: Options for examinations, progression, and results.

OPT	Area	Description	Why not	Status
OPT-15	Examination selection	Flat dropdown menu	Hard to find exam in long list without categories (RISK-09)	Rejected
OPT-16	Examination selection	Categorized selection menu (main category with subcategories)	—	**Chosen**
OPT-17	Examination selection	Checkboxes	A large checkbox list becomes visually dense and increases the chance of selecting the wrong tests or overlooking required ones (RISK-02, RISK-09)	Rejected
OPT-18	Overview progress	Acceptance timeline	—	**Chosen**
OPT-19	Overview progress	Progress stage tree	Did not clearly improve the look of the exported document; also not enough time to implement properly (RISK-10)	Rejected
OPT-20	Overview progress	Point/grade system	Imposes an evaluative score rather than showing progress, conflicting with the non-grading goal and risking misleading feedback to students (RQ-05, RISK-01)	Rejected
OPT-21	Overview progress	Copy of the log	Raw logs obscure progress patterns and make it harder for tutors to identify the next action or intervene in time (RISK-03, RISK-04)	Rejected
OPT-35	File attachment	Attach files to procedure groups	Group-level ownership can leave files ambiguous or orphaned, making it unclear which procedure a file supports (RISK-05, RISK-09)	Rejected

OPT	Area	Description	Why not	Status
OPT-54	Log actions	Single `note` action	Encoding progression and missed deadlines as generic notes can hide system state and prevent reliable deadline/progression handling (RISK-02, RISK-03, RISK-04)	Rejected
OPT-65	Log actions	Add enums for `advance_procedure` and `deadline_missed`	—	**Chosen**
OPT-55	DICOM viewer	Download only	Forces students to leave the learning workflow and rely on external software, increasing the chance that imaging is unavailable or unreadable during the task (RISK-09)	Rejected
OPT-56	Procedure categories	Render categories and subcategories from server data (users can add new ones at runtime)	Requires broad frontend and backend changes late in the project, risking contract defects, regressions, and insufficient time for integration tests (RISK-06, RISK-10, RISK-12, RISK-13)	Rejected
OPT-57	Displaying procedure results	Show a compact list displaying the case stage and result type; selecting an item opens a dialog with the result	—	**Chosen**
OPT-71	Displaying procedure results	Display all results directly on the page	Multiple results, particularly images, can make the page cluttered and leave images too small to inspect	Rejected
OPT-72	Displaying procedure results	Show results in expandable accordion rows	Multiple expanded results can clutter the page, while images may have insufficient viewing space	Rejected
OPT-73	Displaying procedure results	Open each result on a separate page	Requires additional navigation between results	Rejected
OPT-74	DICOM rendering technology	Use Cornerstone3D libraries to load and display DICOM images in the browser	—	**Chosen**
OPT-75	DICOM rendering technology	Embed a full DICOM viewer platform, such as OHIF	A broader viewer platform may add more functionality and integration complexity than this application needs	Rejected

OPT	Area	Description	Why not	Status
OPT-76	DICOM rendering technology	Build a custom DICOM loading and rendering implementation	Implementing and maintaining image decoding and viewer behavior would require specialized development work	Rejected
OPT-77	DICOM viewer controls	Use mouse gestures for pan, window and level adjustment, zoom, and frame scrolling, with an on-demand help control	—	**Chosen**
OPT-78	DICOM viewer controls	Provide persistent on-screen buttons or sliders for the viewer controls	A persistent control panel may take space from the image viewport	Rejected
OPT-79	DICOM viewer controls	Use mouse gestures without an in-viewer explanation	Users may not discover which gestures control the image	Rejected

The decisions for these options can be found in Appendix E. The categorized examination menu was chosen in DEC-36, while DEC-37 and DEC-38 cover the progress overview and results view. Other decisions cover imaging support (DEC-54), bloodwork selection (DEC-55), procedure categories (DEC-16), file uploads (DEC-58), and displaying multiple results (DEC-59).

One important change was how progression and missed deadlines are recorded. Initially, these could have been stored as regular notes (OPT-54). However, this would make it difficult to distinguish between an actual action and something a student or tutor had written. We therefore introduced separate `advance_procedure` and `deadline_missed` log entries (OPT-65), as recorded in DEC-57.

The results view uses a compact list that displays the case stage and result type without displaying every result at once. Selecting an item opens its result in a dialog. This keeps the list easy to scan and gives images more room for inspection. Showing every result inline or in expanded accordion rows could clutter the page and make images too small, while a separate page for each result would add navigation steps. This choice is recorded in DEC-59 and design-log entry D58.

The frontend uses Cornerstone3D to load and display DICOM files in the browser. The application uses its core, DICOM image loader, and tools packages for rendering and image interaction. Downloading the file for use in external software remains a fallback if the viewer cannot display it.

The viewer uses mouse gestures for image interaction, with a help control that reveals the available controls when requested. The left mouse button pans the image, the middle mouse button adjusts window and level, the right mouse button zooms, and the scroll wheel moves through image frames. Keeping the instructions available on demand avoids a permanently visible control panel while allowing students to discover the gestures.

Overview and PDF Export At the end of a case, students need to be able to export an overview of their work. We considered several options for generating this document, how it should look, and how students could preview it before exporting.

Table 21: Options for the overview and PDF export.

OPT	Area	Description	Why not	Status
OPT-22	Overview file format	PDF	—	**Chosen**
OPT-23	Overview file format	DOCX	Students can edit the exported record afterwards, and the layout may differ between applications (RISK-07, RISK-14).	Rejected
OPT-24	Overview file format	TXT	Students can edit the exported record, while the loss of images and formatting would make parts of the dossier harder to interpret (RISK-07, RISK-14).	Rejected
OPT-36	PDF generation	Generate the PDF in the frontend from case logs	—	**Chosen**
OPT-37	PDF generation	Generate the PDF through a backend endpoint	Requires additional implementation and integration work when the frontend already has the necessary data (RISK-10, RISK-12, RISK-13).	Rejected
OPT-38	PDF preview	Read-only HTML/SVG preview	—	**Chosen**
OPT-39	PDF preview	Browser's built-in PDF viewer	Built-in annotation tools make it harder to keep the preview separate from an editable copy (RISK-14).	Rejected
OPT-40	Preview/PDF consistency	Use the same layout operations for the preview and PDF	—	**Chosen**
OPT-41	Preview/PDF consistency	Generate the preview and PDF separately	The two layouts could differ, leaving students with a preview that does not match the exported document (RISK-07, RISK-13).	Rejected
OPT-42	Export scope	Include only the selected group's logs	—	**Chosen**
OPT-43	Export scope	Include all logs returned by the endpoint	Could include information from other groups in the exported dossier (RISK-08).	Rejected
OPT-44	Request/respc layout	Combine each request and its response into one block	—	**Chosen**

OPT	Area	Description	Why not	Status
OPT-45	Request/response layout	Show requests and responses as separate entries	Makes it harder to see which response belongs to a request (RISK-04, RISK-07).	Rejected
OPT-46	PDF generation	Use a separate backend PDF endpoint	Duplicates PDF generation work and adds another integration point (RISK-07, RISK-10, RISK-12).	Rejected
OPT-47	PDF preview	Use the browser's built-in viewer	Offers less control over the read-only preview and exposes annotation tools (RISK-14).	Rejected
OPT-48	Response linking	Use an explicit request ID with a fallback	—	**Chosen**
OPT-49	Response linking	Match responses using heuristics only	Could link a response to the wrong request, making the activity history unreliable (RISK-04, RISK-07).	Rejected
OPT-50	Request headings	Show procedure type, name, and decision	—	**Chosen**
OPT-51	Request headings	Use generic or numeric headings	Makes it harder to identify the procedure when reading the export (RISK-04, RISK-07).	Rejected
OPT-52	Export styling	Reuse TG styling and Univers fonts	—	**Chosen**
OPT-53	Export styling	Create a separate export-specific theme	Adds styling work and another dependency that could cause problems during export (RISK-07, RISK-10, RISK-12).	Rejected

The decisions for the export and overview features are recorded in Appendix E. Some of these choices had a direct impact on how the final PDF looks and how students use it, so they are explained below.

For the progress overview, we chose an acceptance timeline (DEC-61, OPT-18). It shows which requests have been accepted without the extra complexity of a progress tree. The export itself is a PDF (DEC-62, OPT-22) instead of DOCX or TXT, because the export is meant as a fixed record of the student's work. Editable formats would allow students to change the record after the fact, which conflicts with the requirement that the export reflects the actual activity trail (FR-30). Although PDF files can still be modified using external tools, they are less straightforward to edit than DOCX or TXT files. PDF was therefore chosen to provide a consistent layout and discourage changes to the exported record, rather than to prevent editing entirely. We generate this PDF on the frontend (DEC-63, OPT-36) rather than through a backend endpoint, mainly to avoid duplicating work that the frontend already has access to, which would add unnecessary complexity for the delivery deadline.

The preview and the exported PDF should look the same, so students know what they are handing in. For this reason, we chose a read-only preview (DEC-64, OPT-38) that

uses the same layout as the PDF (DEC-65, OPT-40). If both were generated separately, differences could appear between the preview and the final document.

The export only includes the logs of the selected group (DEC-66, OPT-42), preventing information from other groups from appearing in the document. We also decided to keep requests and their responses together (DEC-67, OPT-44). This makes it easier for teachers to see why a student requested something and what the tutor decided, without having to search for the corresponding response.

For the styling, we reused the existing TG design (DEC-70, OPT-52). Creating a separate theme would have required additional work without providing much benefit. The remaining decisions (DEC-71 to DEC-74) cover the layout, authorship, separation between motivation and response, and grouping by stadium.

Technical Architecture (Frontend) Besides the design of the individual screens, we also had to make decisions about how the frontend would be built and how it would communicate with the backend.

Table 22: Options for the frontend technical architecture.

OPT	Area	Description	Why not	Status
OPT-25	Examination categories	Load categories dynamically from the database	Requires changes to both frontend and backend, with insufficient time for implementation and testing (RISK-06, RISK-10, RISK-12, RISK-13).	Rejected
OPT-26	Frontend framework	Vanilla JavaScript	Would require more manual work for component reuse and state management, increasing maintenance effort (RISK-10).	Rejected
OPT-27	Frontend framework	WordPress	Not suited to the required custom workflows and would introduce additional integration work (RISK-06, RISK-12).	Rejected
OPT-28	Deployment	Single WAR file	Does not fit the selected frontend build and deployment setup (RISK-12).	Rejected
OPT-29	Frontend data types	Separate DTOs for each screen	Duplicates data definitions, which could become inconsistent with the API contracts (RISK-06).	Rejected
OPT-30	Mock API	Multiple configuration flags	Different screens could use different data sources, making integration problems harder to identify (RISK-06, RISK-12).	Rejected
OPT-31	Pop-up handling	Custom `PopupProvider` and `usePopup`	—	**Chosen**

OPT	Area	Description	Why not	Status
OPT-32	Delete confirmation	Reusable `RemoveButton` with confirmation pop-up	—	**Chosen**
OPT-33	Admin and tutor views	Duplicate the tutor interface for administrators	Would require maintaining two similar interfaces and applying fixes in both (RISK-01, RISK-05).	Rejected
OPT-34	Procedure categories	Load categories dynamically from the backend	Requires changes to the existing category handling and additional integration testing (RISK-06, RISK-10, RISK-13).	Rejected
OPT-58	Frontend framework	React	—	**Chosen**
OPT-82	Pop-up handling	Native browser dialogs (`window.confirm` and `alert`)	Do not match the application's styling and are less flexible to test (RISK-10).	Rejected
OPT-64	Delete confirmation	Native browser `confirm()`	Provides less control over the confirmation interface and its behaviour (RISK-10).	Rejected
OPT-66	Final mock API setup	Keep the mock API in the delivered application	Leaves an unused alternative data path that future maintainers would need to understand (RISK-06).	Rejected
OPT-67	Final mock API setup	Remove the mock API and use the real backend	—	**Chosen**

We chose React for the frontend (DEC-79, OPT-58), mainly because its component structure made it easier to reuse parts of the application and keep the code consistent. This also fits the client's wish for a system that can be maintained after the project.

For pop-ups and confirmation messages, we created a custom `PopupProvider` (DEC-11, OPT-31) instead of using the browser's default alerts. This allowed us to use the same style and behaviour throughout the application. We also reused the tutor environment for administrators through an `AdminViewSwitch` (DEC-12), rather than creating a separate screen that would need to be maintained.

Another decision was to keep the procedure categories fixed in the code instead of loading them dynamically from the backend (DEC-16). Although dynamic categories would allow more flexibility, implementing this would have required changes to both the frontend and backend. There was not enough time left in the project to make and properly test these changes.

During development, we used a mock API to work on the frontend before the backend was ready. Once the real backend was available, we removed the mock setup (DEC-87, OPT-67). Keeping both would have made the code harder to maintain and could cause differences between the mock and actual backend behaviour.

Finally, we created a reusable `RemoveButton` (DEC-32, OPT-32), which asks for confirmation before deleting something. This keeps deletion behaviour consistent across the application.

5 Testing

5.1 Front-end

For the front-end, we used **Jest** and **React Testing Library** to set up automated tests. These tests mainly focused on functionality where errors might not be immediately noticeable during manual testing. This included the API layer, authentication, backend service calls, pop-ups, form conversions for cases and procedures, and the logic used in the request and review screens. By mocking the backend, we tested how the frontend responds to different situations without depending on the actual backend. For example, we tested whether requests made in the wrong stage were blocked and whether mixed lab requests correctly highlighted individual items.

In addition to the automated tests, we manually tested the system from the perspectives of teachers, tutors, and students. This helped us check complete user flows, the layout of the interfaces, and how different parts of the system worked together. It was especially useful for finding issues that were not covered by the automated tests.

We also regularly tested the design with the client and teachers throughout the project. During feedback sessions and hands-on testing, we could see whether the system worked as they expected and identify parts that needed to be changed. This feedback was used to improve the design as development continued. The frontend was also tested in most common desktop browsers to check whether the main workflows and interface elements behaved consistently across them.

5.2 Back-end

The backend was tested using an automated JUnit test suite covering individual components, backend workflows, API contracts, and endpoint behaviour. A substantial part of the test suite directly targets the routes and application logic. These tests verify both the responses returned by backend operations and the resulting system state, such as whether data is correctly stored or modified in the database. They cover, among other things, procedure requests, logs, case and group management and the data structures exchanged between the front-end and back-end.

Authentication and authorization received additional testing. Valid users were able to log in, while invalid identities and tokens were rejected. Role restrictions were also tested, for example to verify that a student cannot perform administrative actions such as modifying groups. Login, logout, session handling, token expiration, and restrictions on access to protected information were also tested.

File handling tests cover valid uploads as well as interrupted uploads. Failure cases verify that partially stored files are removed when necessary and that unauthorized students cannot upload files requiring tutor or teacher permissions. DICOM processing was additionally tested for the grouping of image slices.

Further tests cover persistence and data-contract behaviour, including log persistence, account lookup, CSV import parsing, group data structures, and consistency between front-end JSON representations and backend models.

Session handling was also tested, including token expiration,inactivity-based sessiontimeouts, and access to protected resources after a session had expired.

5.3 Security and Penetration Testing

The EPD system was penetration-tested to assess if there exist any vulnerabilities which could be exploited by a malicious user. We used the OWASP Application Security Verification Standard (ASVS) 5.0 [4] and the OWASP Web Security Testing Guide (WSTG) 4.2 [3] to test this, focusing on security controls relevant to the EPD. To ensure a consistent and repeatable analysis, the system was deployed on Docker. The testing was primarily focused on controls relevant to the EPD, which consists of: authentication, authorization, session management, API security, input validation, file handling, and access to protected resources.

Automated dynamic application security testing was performed using OWASP ZAP [8]. The application was crawled and inspected using passive analysis, followed by active testing of the test deployment.

Manual penetration testing was also conducted. This included trying to send invalid data formats to the back-end, which included modifying case, group, request, result, and file identifiers. We also directly invoked tutor and teacher operations using a student account. Furthermore, we tried accessing endpoints without authentication and testing session and logout behavior.

Results of passive penetration testing No high-risk vulnerabilities were found, after conducting the OWASP ZAP passive assessment. In total, from the analysis, we found two medium-risk, five low-risk, and three informational findings.

The two medium-risk findings concerned missing HTTP security headers. A Content Security Policy (CSP) was not configured, reducing the browser-level protection against attacks such as cross-site scripting and content injection. ZAP also reported missing anti-clickjacking protection, as neither an appropriate `frame-ancestors` CSP directive nor an `X-Frame-Options` header was present.

The low-risk findings concerned additional browser security headers: the Cross-Origin-Embedder-Policy, Cross-Origin-Opener-Policy, Cross-Origin-Resource-Policy, Permissions-Policy, and `X-Content-Type-Options` headers were missing or not configured. These findings primarily concern security hardening of the web-server responses rather than the application's educational workflow or role-based authorization logic.

The automated tests did not identify and high-risk vulnerabilities, but they covered endpoints and test cases during testing. Additional security testing would need to be performed before deployment.

Table 23: OWASP ZAP passive scan findings

Finding	**Risk**	**Instances**
Content Security Policy (CSP) Header Not Set	Medium	Systemic
Missing Anti-clickjacking Header	Medium	2
Cross-Origin-Embedder-Policy Header Missing or Invalid	Low	2
Cross-Origin-Opener-Policy Header Missing or Invalid	Low	2
Cross-Origin-Resource-Policy Header Missing or Invalid	Low	5
Permissions Policy Header Not Set	Low	Systemic
X-Content-Type-Options Header Missing	Low	Systemic
Modern Web Application	Informational	1
Non-Storable Content	Informational	1
Storable and Cacheable Content	Informational	Systemic

Results of active penetration testing Conducting the OWASP ZAP active assessment did not identify any high-risk vulnerabilities. The generated report contained two medium-risk, five low-risk, and three informational findings, corresponding to the HTTP security configuration issues identified during passive testing. No additional vulnerability categories were reported with the exception of an outlier on a 5xx response.

The assessment also recorded that 82% of HTTP responses returned a 4xx status code. This is because many bad, invalid requests are sent to the server, to try to achieve a system failure. Additionally, 1 % of HTTP responses returned a 5xx server-error status code. The cause of these errors was not established, and they do not necessarily indicate an exploitable vulnerability. This however does not guarantee an attack was blocked.

Overall, the assessment did not reveal any high-risk vulnerabilities, while the previously identified HTTP security configuration issues remain to be addressed. By nature, the active automated tests tend to have a limited coverage which also means that the results cannot be considered a comprehensive verification of the application's security.

The authentication and role-based permission operations have to be tested manually for verification, and no users were able to access unauthorized operations (relative to their roles).

Table 24: OWASP ZAP active scan report findings

Finding	Risk	Instances
Content Security Policy (CSP) Header Not Set	Medium	Systemic
Missing Anti-clickjacking Header	Medium	2
Cross-Origin-Embedder-Policy Header Missing or Invalid	Low	2
Cross-Origin-Opener-Policy Header Missing or Invalid	Low	2
Cross-Origin-Resource-Policy Header Missing or Invalid	Low	2
Permissions Policy Header Not Set	Low	Systemic
X-Content-Type-Options Header Missing	Low	2
Modern Web Application	Informational	1
Storable and Cacheable Content	Informational	Systemic
User Agent Fuzzer	Informational	Systemic

Table 25: Comparison of OWASP ZAP scan statistics

Metric	Passive	Active
Discovered endpoints	19	14
GET endpoints	100%	100%
HTTP 2xx responses	42%	13%
HTTP 3xx responses	4%	3%
HTTP 4xx responses	52%	82%
HTTP 5xx responses	Not reported	1%
Network failures	Not reported	1%
Slow responses	Not reported	1%
Logged ZAP warnings	1	30

6 Results

6.1 Achievement of the Project Goals

The project resulted in a functional educational Electronic Patient Dossier that supports the core workflows of students, tutors, and teachers. Most of the main functional objectives were implemented, including configurable patient cases and stages, procedure requests and reviews, automation, medical imaging, and dossier export. The main project goal was therefore largely achieved, although several aspects related to long-term operation, deployment, and maintainability still require further verification and development.

G-01 — Configurable administration.

This goal was largely achieved. Authorized teachers can manage modules, patient cases, groups, stages, users and other educational content through the application without modifying the source code. A remaining limitation is that procedure categories are still defined statically and cannot yet be configured dynamically.

G-02 — Maintainability and handover.

This goal was partially achieved. The system uses a separated React frontend, Jakarta EE backend, PostgreSQL database, and external authentication mechanism. Technical documentation, deployment instructions, and separate user manuals were also created. However, independent deployment and long-term maintenance by Technical Medicine staff have not yet been fully verified.

G-03 — Educational workflows.

This goal was largely achieved. Students can work through assigned patient cases, maintain notes and clinical reasoning, request diagnostic procedures, and view released results. Tutors can review requests, provide feedback and results, and monitor group progress. Teachers can manage the educational structure, including modules, groups, cases, stages, users, and deadlines.

G-04 — Data integrity and security.

This goal was partially achieved. Authentication, authorization, role restrictions, session handling, and file handling were implemented and tested. The penetration tests did not identify any high-risk vulnerabilities, but several HTTP security headers still need to be configured. Some data consistency issues also remain, alongside the implementation of automatic retention, backups, and recovery.

G-05 — Progressive workflow and medical results.

This goal was largely achieved. Patient cases can be divided into configurable stages, procedure requests can be reviewed manually or handled automatically, and the system supports progression between stages. Diagnostic results and medical images can also be provided through the EPD, including support for DICOM images.

6.2 Requirements

In the Table 26 below, we have sorted the requirements that we decided on for this project by in what part they have been implemented. Any requirements that are not fully implemented, will be discussed in this section.

Table 26: Progress on the Requirements of the EPD.

Status	**Requirements**
Fully Implemented	FR-02, FR-04, FR-05, FR-06, FR-07, FR-08, FR-09, FR-10, FR-11, FR-12, FR-14, FR-15, FR-16, FR-17, FR-18, FR-19, FR-20, FR-21, FR-23, FR-24, FR-26, FR-27, FR-28, FR-29, FR-30, FR-31, FR-32, FR-34, FR-36, FR-37, FR-38, FR-39, FR-41, FR-42, FR-43, FR-44, FR-45, FR-46, NFR-03, NFR-04, NFR-06, NFR-07, NFR-10, NFR-11, NFR-12
Partially Implemented	FR-01, FR-03, FR-13, FR-22, FR-25, FR-33, FR-35, NFR-01, NFR-02
Unverified / Remaining	FR-40, NFR-05, NFR-08, NFR-09

Partially achieved requirements.

- **FR-01.** Since this required full access to the University of Twente account system, we could not fully implement this. Instead, we used a modular design and a temporary authentication module so that the UT login system can be swapped in with relative ease.
- **FR-03.** In the end, we opted to make the notes in the dossier non-editable, so that the system kept integrity. However, students can create additional notes explaining mistakes in previous ones.
- **FR-13.** The foundation for an email-notification system was laid, but during implementation we decided to not fully implement it due to time constraints.
- **FR-22.** There is a known bug allowing teachers to duplicate student across multiple groups.
- **FR-25.** Automation is decided on a module level, rather than for individual actions. This was decided in conversation with the teachers that will use the EPD.
- **FR-33.** The University of Twente identity provider will need to be configured during the handover period, though the groundwork for it has been laid.
- **FR-35.** No encryption is done within the EPD codebase itself. However, once the system is deployed on a UT server with an SSL Certificate, this requirement will be met.
- **NFR-01.** Deployment instructions have been prepared, but an independent deployment by Technical Medicine staff has not yet been tested. This will need to be verified during the handover.
- **NFR-02.** There is no documentation regarding backups and recovery, since no systems were put in place for this within the EPD.

Unverified or remaining requirements.

- **FR-40.** In conversation with the client, we decided against a super-admin role, since only a handful of people would get access to the base admin role. This left little reason for a role solely to manage those few.
- **NFR-08.** retention and automatic deletion still needs to be investigated and implemented. we did not have the time to investigate the data retention laws of the exact data we are holding in accordance to the GDPR.
- **NFR-09.** Due to time constraints, we were not able to implement a backup and recovery system within the EPD.
- **NFR-05.** Page loading times were not measured under a defined workload. Although the application was tested manually, the proposed six-second target has not been formally verified.

6.3 Research Questions

RQ-01 — User interfaces and workflows.
How can the interfaces support the workflows of students, tutors, and teachers while keeping navigation and information access straightforward?

The interfaces were separated by user role, so students, tutors, and teachers see the functionality relevant to their tasks. Navigation is placed on the left, the main content in

the centre, and actions and history on the right where applicable. Tutor pages were also split into separate views to avoid showing too much information at once. The layout was adjusted based on feedback from the client and teachers.

RQ-02 — Configuration and reuse of educational content.
How can patient cases, groups, and educational content be configured and reused without making administration difficult for teaching staff?

Patient cases are stored separately from modules, allowing teachers to reuse existing cases without changing the originals. Teachers can manage modules, stages, groups, and deadlines through the interface. Student groups can also be imported using CSV files, reducing the need to enter this information manually.

RQ-03 — Stage progression and examination requests.
How can the system enforce stage progression and examination-request rules while supporting the educational workflow?

The EPD uses stages and deadlines to control progression through patient cases. Examination requests can be reviewed by tutors or handled automatically, depending on the module settings. Requests made in the wrong stage are highlighted, and tutors cannot approve them. For laboratory requests, individual items that do not belong to the case are highlighted, but tutors can still approve the request. Whether this should be allowed remains to be decided with the client. This allows the system to enforce stage restrictions while leaving tutors some flexibility when reviewing laboratory requests.

RQ-04 — Permissions, data consistency, and retention.
How can the system manage access permissions, data consistency, and retention while allowing users to perform their assigned tasks?

Authentication and role-based authorization restrict access to different parts of the EPD. The backend also checks permissions for stored medical files, while validation and cleanup during file operations help prevent inconsistent data. These controls were included in testing, but still need to be verified in the final deployment environment. Automatic data retention and backup recovery have not yet been implemented.

RQ-05 — Maintainability and handover.
How can the system architecture and deployment process support maintenance and handover to Technical Medicine staff?

The application separates the React frontend, Jakarta EE backend, and PostgreSQL database, allowing the components to be developed and maintained separately. Technical documentation, deployment instructions, and user manuals were prepared for the handover. Members of the project team will also assist Technical Medicine staff with the remaining deployment work. Whether their staff can maintain the system independently still needs to be verified.

6.4 Overall Result

The project resulted in a working educational EPD that implements most of the requirements agreed upon with the client. The main workflows for students, tutors, and teachers have been implemented and tested. However, some known issues remain, and the system still needs to be deployed on the University of Twente infrastructure and tested in its final environment before it can be used in education. These remaining tasks are discussed in the following section.

7 Discussion & Future Work

7.1 Transfer to the UT

Though the application was designed with the University's infrastructure in mind, it will still take some work to get the system up and running for the students to use. After discussing this with the client and the maintainer of the current EPD, we decided that a number of our team members will do some additional work to get our EPD ready for use. The goal is that it is fully set up by the end of the third quartile (in may), so it can be used in TG's module 8, in the fourth quartile of this academic year.

In that time, the application needs to be transferred to the UT servers, where it needs to be made accessible by URL. The current implementation uses Keycloak for authentication (OPT-86, DEC-102). Before deployment, the authentication setup will need to be adapted to support the University's Microsoft Single Sign-On system, allowing students to log in using their university accounts. Lastly, the system will be validated and tested more thoroughly than what was reasonable for within the scope of our project.

Hosting responsibilities and the required availability during teaching activities were discussed and agreed upon with the client. The remaining work concerns deploying and validating the system in the University of Twente environment.

7.2 Known Issues

Beyond the general setup and planned alterations, there are also some issues that were found while testing the application that could not be fixed before the deadline of this project. The chance that similar issues will be found during the validation of the system is very high, so this list is not exhaustive. These issues need to be resolved before the application can be used in education. Further problems may also be identified during final testing and validation. This list includes:

- When multiple .dcm images are uploaded, it is not possible for the server to distinguish whether they belong to different series unless they have different resolutions.
- The backend endpoints for uploading images only accept Dicom files, though other image types (such as .jpg or .png) should also be accepted.
- Student are currently not able to upload files as part of their procedure request.
- There is currently no good way of adding teachers to the database, and this implementation should work with the Single Sign On system that will be implemented.
- When dragging students from group to group on the Group Overview page, they are duplicated rather than moved.
- Updating a procedure in an existing case can cause previously approved requests for that procedure to disappear from the student's view, preventing access to their results.

7.3 Future Additions

Besides problems, we also found a lot opportunities of how the application could be improved upon in the future. Depending on the wishes of the client and available

resources, some of these features might be implemented before the launch of the system. Or, this list could one day be used to guide another project group on considerations to be made when the next iteration of this EPD is made. Valuable additions include:

- Making the procedure categories editable, rather than hardcoded in the database schema, allowing teachers to change them as they see fit and as new procedures are adding or archaic ones removed.
- Allowing tutors and teachers to manually progress a group to the next stage in the case that no deadline for a stage was set, but a decision was made to move them along anyway.
- Expanding on the simple email system that is currently part of the application, to give detailed and timely information to tutors and students.
- Adding a dark mode option to improve readability and user preference.
- Supporting different languages, rather than only Dutch, to prepare student for working with such systems in international settings.
- Improvements upon the automation system of the application, allowing for a more detailed automatic evaluation beyond just checking whether the selected procedure is correct, but possibly also analyzing the reasoning behind the request.
- Automatic backups of the database and archival or deletion of information to always remain in line with the applicable retention policy.
- Examination mode, by expanding on the current deadline system with more frequent checks for stage progression, for select cases only from start to finish. Furthermore, there should be an option to give a student extra time in case of technical difficulties.

8 User Manuals

by J. Grimberg

The system has three different user roles, and each role works with the EPD in a different way: students work through a patient case, tutors review requests and guide groups, and teachers create and configure modules, cases and groups. A single combined manual would mix instructions for three very different tasks and make it harder for each role to find what it needs. We therefore wrote three separate manuals, one per role:

- the **student manual** (Appendix H), covering navigation, viewing patient information, requesting examinations, writing notes, and exporting the dossier;
- the **tutor manual** (Appendix I), covering the tutor dashboard, reviewing requests, releasing results and following a group's progress;
- the **teacher manual** (Appendix J), covering modules, cases, groups, deadlines and the automatic handling of requests.

The manuals are written in Dutch rather than English. The reason is that the end users are Dutch-speaking students, tutors and teachers at the University of Twente, and the EPD itself uses Dutch medical terminology (for example *stadium* instead of *fase*, and *aanvullend onderzoek* instead of *onderzoek aanvragen*; W-13). A manual that used English terms for the interface would not match what the user sees on screen, and assume all student would understand the english termonology when they follow a dutch study. Writing the manuals in Dutch keeps the instructions consistent with the interface and with the terminology the client asked for.

9 Contributions

Table 27 summarizes the individual contributions of each member of the team.

Table 27: Individual contributions of the team members.

Name	**Contribution**
Hossein	Backend design and implementation, documentation, presentations, drafting poster, DICOM functionality and penetration testing
Jasper	Frontend wireframes, frontend admin implementations. Student/Teacher/Tutor manual creation. G/SG/FM/DEC/OPT/RISK/SM/RQ/SG tables creation and setup.
Brian	Initial contact with client. Finding a supervisor. Database design. Backend interaction with the database. Setting up a testbed with a database and webserver on a Raspberry Pi, using PostgreSQL and Apache Tomcat.
Umid	Backend class diagram design, initial backend scaffold, initial backend design.
Sanne	Backend class diagram design, backend design, backend Implementation, request handling, automation, connecting front and back ends, Poster design.
Nijs	User flow diagrams, frontend structure design, frontend student implementations, frontend tutor implementation.

10 AI Disclaimer

During the preparation of this work, we used ChatGPT, Claude and Github Copilot to assist in coding, structuring and refining author-provided content, Grammarly to rephrase and improve grammar, punctuation and readability and to assist in populating and structuring some tables and to keep track of relevant codes. After using these tools, we thoroughly reviewed and edited the content as needed, taking full responsibility for the final outcome.

11 Acknowledgments

This design project has been partially funded by the DMB group of the EEMCS department at the University of Twente in the Netherlands. Furthermore, we would like to thank our supervisor, dr. Farzad Vazinram, for mentoring us and for his dedication to this project.

References

1. Feuerstein, S., Pribyl, B.: Oracle pl/sql Programming. " O'Reilly Media, Inc." (2005)

2. Jackson, S.: Zlib compressed protocol header length confusion may allow memory read (2025), https://www.cve.org/CVERecord?id=CVE-2025-14847
3. OWASP Foundation: Owasp web security testing guide, version 4.2. https://wstg.owasp.org/v4.2/ (2020)
4. OWASP Foundation: Owasp application security verification standard 5.0.0. https://asvs.owasp.org/ (2025), stable release, published 30 May 2025
5. Peckham, D.: Electronic patient records, past, present and future. Paediatric Respiratory Reviews (2016)
6. PHP Group: CVE-2026-109618 (2026), https://nvd.nist.gov/vuln/detail/cve-2026-109618
7. PHP Group: CVE-2026-109619 (2026), https://nvd.nist.gov/vuln/detail/cve-2026-109619
8. ZAP Development Team: Zap baseline scan. https://www.zaproxy.org/docs/docker/baseline-scan/ (2026)

A Wireframes

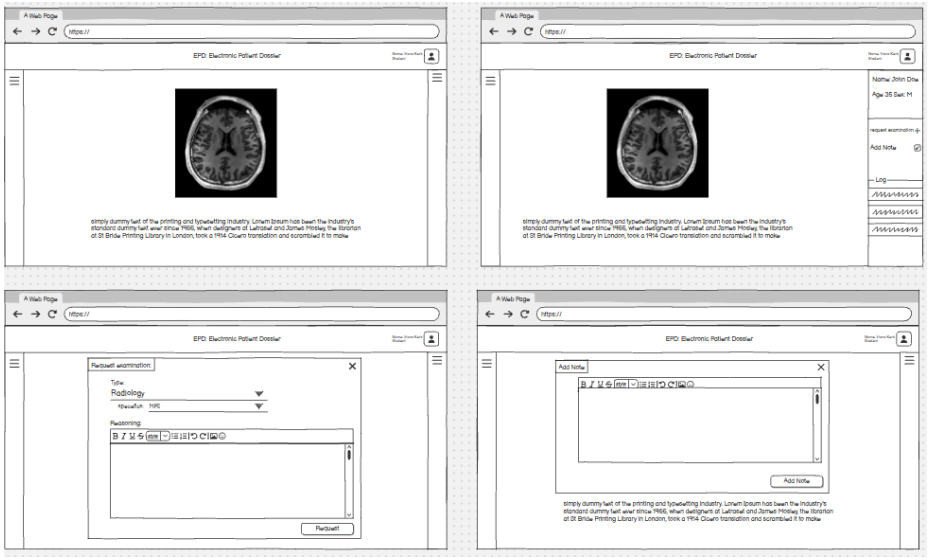


Fig. 3: Initial student wireframe (design draft).

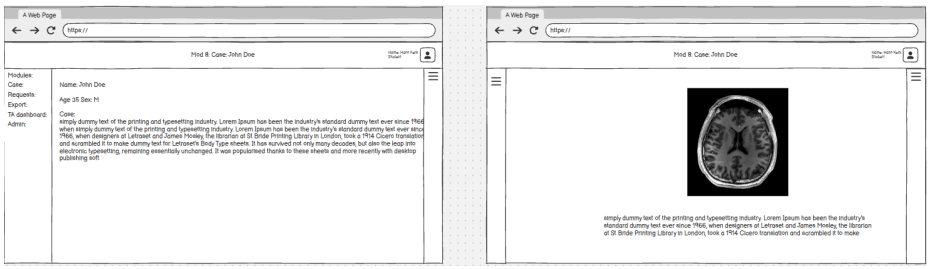


Fig. 4: Initial student wireframe: left, full case; right, specific examination (design draft).

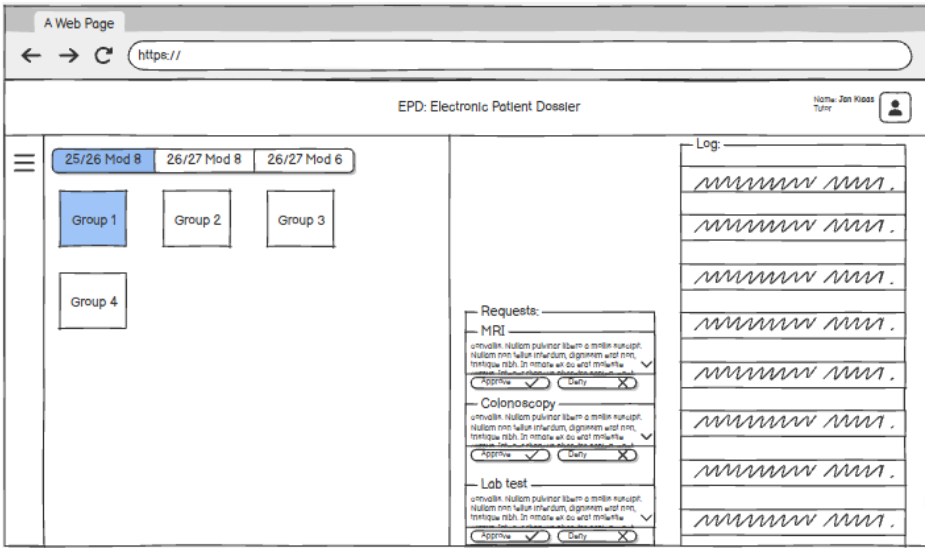


Fig. 5: Initial tutor dashboard wireframe (design draft).

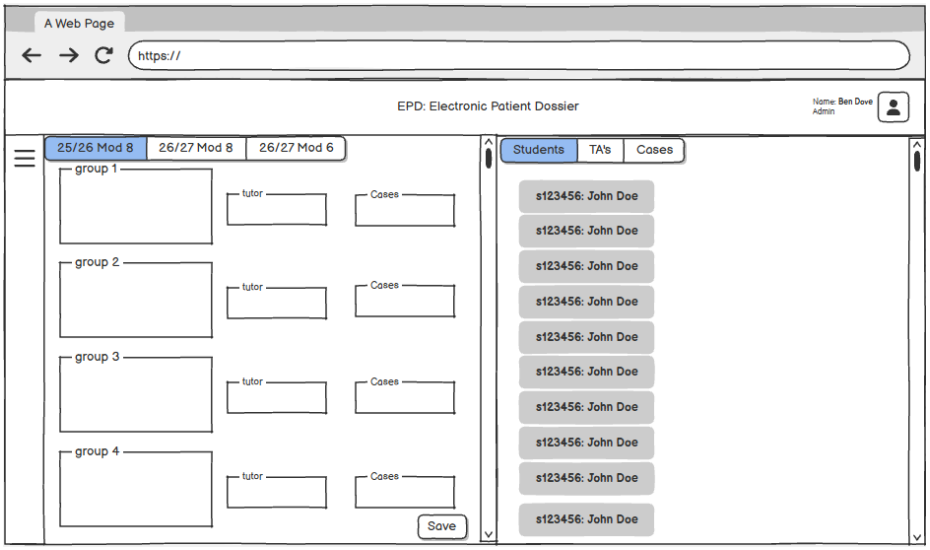


Fig. 6: Initial group creation wireframe (design draft).

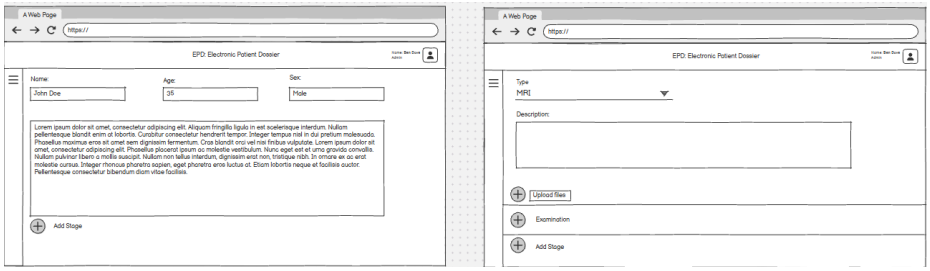


Fig. 7: Initial case creation wireframe (design draft).

B Stakeholder Analysis

The stakeholder analysis identifies the groups that affect, use, operate, or may later benefit from the EPD. For each stakeholder, both their interest in the system and their ability to influence the project were considered. The current scope is the Bachelor Technical Medicine program; possible later use in the Master's program was treated as a future extension rather than a current requirement.

Table 28: Stakeholder analysis of the EPD project.

Stakeholder	Role and main needs	Power / interest	Engagement and design impact
Technical Medicine programme / client	Client and intended owner of the EPD. Needs a maintainable educational system that supports patient cases, diagnostic requests, results, phases, group work, and long-term use without dependence on the original development team.	**High / High**	Manage closely. Requirements were gathered directly through meetings and prototype demonstrations. Client feedback drove changes to case structure, phases, procedures, reporting, and administrative functionality.
Bachelor Technical Medicine students	Primary current end users. Need clear patient information, the ability to request examinations and explain their reasoning, take notes, view released results, and progress through cases without seeing information that has not yet been released.	**Medium / High**	Keep informed and validate usability. The student workflow, group-based access control, diagnostic-request process, logging, result release, and separation between patient information and actions were designed around their case-based learning process.
Tutors / teaching assistants	Supervise groups and review student decisions. Need quick access to assigned groups and cases, pending requests, student reasoning, approval/rejection controls, result release, and progress history.	**Medium–High / High**	Manage closely during workflow design. Their needs shaped the tutor dashboard, request-review process, group overview, notifications, and the chronological case log.

Stakeholder	Role and main needs	Power / interest	Engagement and design impact
Teachers / module coordinators / administrators	Configure and oversee the educational workflow. Need to create and edit modules, cases, groups, procedures, phases, and deadlines; assign students and tutors; and see the complete case, including information still hidden from students.	**High / High**	Manage closely. Teacher feedback led directly to changes involving phases and deadlines, full-case visibility, missed-deadline alerts, configurable examinations, clearer result handling, and integration of information previously kept outside the EPD.
Master Technical Medicine students (potential future users)	Possible future users if the EPD is extended beyond the Bachelor's programme. Their exact workflow is outside the current project scope, but the system should avoid unnecessary assumptions that would make later extension difficult.	**Low–Medium / Medium**	Monitor as a future stakeholder. Extensibility was considered in design decisions such as allowing a group to be associated with multiple cases instead of permanently hard-coding a one-group–one-case model.
Technical Medicine support / future maintainers	Will operate and maintain the system after handover. Need understandable architecture, documentation, reliable storage and deployment, backups, and the ability to change educational content without editing source code.	**Medium–High / High**	Keep closely involved in handover. Maintainability influenced the modular architecture and the decision to expose routine case, module, and group configuration through the application instead of requiring intervention from the original developers.
University of Twente / ICT environment	Provides the institutional environment and infrastructure in which the system is intended to operate. Needs appropriate authentication, authorization, secure communication, data protection, and compatibility with university systems.	**High / Medium**	Keep satisfied. UT authentication, role-based authorization, session handling, HTTPS, restricted data access, and infrastructure constraints were included in the requirements and risk analysis.

Stakeholder	Role and main needs	Power / interest	Engagement and design impact
Project team	Designers and developers responsible for delivering the system within the module timeframe. Need a technically feasible solution that can be implemented, tested, documented, and handed over with the available time and expertise.	**High / High**	Manage closely internally. Technical feasibility, previous experience, implementation effort, stakeholder feedback, decision tables, FMEA, and the Design Log were used to compare and justify design choices.

C SWOT Analysis

To evaluate the current position of the proposed EPD system, a SWOT analysis was performed. Strengths and weaknesses concern characteristics internal to the project, while opportunities and threats originate primarily from the environment in which the system is intended to operate.

Table 29: SWOT analysis of the EPD project.

Strengths	**Weaknesses**
• The system is specifically designed around the workflow of the Technical Medicine programme rather than being a generic EPD. • Direct access to the client, supervisors and other stakeholders in the university allows requirements and design decisions to be validated. • The backend has a modular domain model separating users, groups, patient cases, requests, reviews and results. • The team already has experience with several selected technologies, including Java and PostgreSQL. • Existing project components can be reused and adapted, reducing the amount of basic functionality that must be developed from scratch.	• The system is developed within the limited duration of a university module, restricting development and evaluation time. • Some requirements and design decisions are still being refined during development. • The development team has limited experience operating a production system containing medical or medically related information. • The current Raspberry Pi setup is primarily a test environment and may not accurately represent the final production infrastructure yet. • The system is specifically adapted to the University of Twente's specifications rather than a universal EDP.
Opportunities	**Threats**
• The system could replace or substantially improve the existing EPD used within the Technical Medicine programme. • Integration with UT authentication could provide users with an existing and familiar account system. • Integration of a DICOM viewer could make patient cases more realistic by incorporating medical imaging directly into the workflow. • The architecture could later support additional modules, patient cases or educational programmes. • Parts of the diagnostic-request workflow could potentially be automated, reducing repetitive work for tutors.	• Security or privacy problems could prevent deployment or reduce trust in the system. • Changes to UT infrastructure, authentication systems or external services could require modifications to the application. • Stakeholder requirements may change as the prototype is evaluated by actual users. • Additional requested functionality could increase the project scope beyond what can realistically be implemented during the module. • Long-term use depends on maintainability and transfer of the system after our development team leaves.

D FMEA

The FMEA lists the specific ways a process, feature or implementation can fail. Each row describes one failure mode (FM) or problem: what could go wrong, what the effect would be, and how serious, likely and detectable it is, scored as severity (S), occurrence (O) and detectability (D). We calculate the Risk Priority Number as $S \times O \times D$ and use it to decide which problem to look into first.

An FM is one specific failure scenario, not automatically a separate project risk. Where one or more failure modes point to a broader threat that we need to keep an eye on, we link them to a RISK entry in Table 10. Risks that are not properly covered by a single failure mode are added to that table directly.

The FMEA is also linked to the design decisions. Where a decision addresses a failure mode, the corresponding DEC is referenced. If a design-log entry records how the issue developed, its D-ID is included as well. Failure modes without a completed mitigation remain in the table without a DEC reference.

Table 30: Failure Mode and Effects Analysis (FMEA) with related design decisions and controls.

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-01	Requesting exams	Student requests an examination that is not available in the case	Student receives no result but can learn from the inappropriate request	3	6	3	54	DEC-42 (Allow requests beyond the prepared case); D06
FM-02	Blood work request	Student can't select which value to test	Wrong/incomplete test, confusion	4	5	3	60	DEC-55 (Selection list); D18
FM-03	Case progression	Student moves to next phase without required steps	Learning objectives skipped	7	6	3	126	DEC-39 (Phase locking); D05
FM-04	Case progression	Student blocked because phase not available	Frustration, stalled progress	5	5	3	75	DEC-40 (Soft deadlines); D05
FM-05	Case creation	Teacher can't control which exams available	Wrong exam set per case	5	6	4	120	DEC-23 (Teacher selects exams); D06
FM-06	Case–module mapping	Case not linked to correct module	Student sees wrong case	6	5	3	90	DEC-18 (Multiple modules); D11
FM-07	Imaging module	CT scans / thorax photos not supported	Imaging learning goal not met	5	4	3	60	DEC-54 (Support CT + thorax); D18

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-08	Screen layout	Patient info mixed with actions	Confusing UI, slow navigation	5	8	3	120	DEC-06 (Separate info from actions); D15
FM-09	Screen layout	Patient info not visible at a glance	Teacher/student misses context	4	7	3	84	DEC-07 (Patient vignette); D15
FM-10	Screen layout	“Onderzoek aanvragen” label unclear	User doesn’t find block	3	7	3	63	DEC-45, DEC-53 (request entry wording and visibility); D12
FM-11	Results view	All results one big text block	Hard to read, info missed	5	7	3	105	DEC-37 (Tabs/buttons); D16
FM-12	Results view	Results not clickable	Poor interaction	3	7	3	63	DEC-38 (Clickable results); D16
FM-13	Results view	Supplementary exams coupled to patient info	Confusing structure	4	6	3	72	DEC-46 (Decouple exams); D15
FM-14	Referral display	GP referral not on screen	Missing base context	4	6	3	72	DEC-41 (GP referral base element); D15
FM-15	Tab structure	Too many tabs	Overwhelming UI	4	6	3	72	DEC-08 (One tab); D15
FM-16	Logging	Student actions are not recorded correctly	Activity history is incomplete, making it difficult to reconstruct the group’s work	6	5	4	120	— (Logging completeness requires verification); D17
FM-17	Tutor navigation	Pop-up interrupts tutor flow	Tutor loses overview of pending tasks	5	6	3	90	DEC-14, DEC-44 (tutor review page and notification navigation); D12
FM-18	Deadline monitoring	Missed deadline hidden in small text	Teacher misses it	7	6	3	126	DEC-48 (Prominent deadlines); D17
FM-19	Deadline monitoring	No alert when group misses deadline	No intervention possible	7	6	3	126	DEC-48 (prominent missed-deadline flag); D17
FM-20	Teacher overview	Teacher can’t see full case	Can’t coach or assess	8	7	4	224	DEC-49 (Full-case overview); D17
FM-21	Requesting lab examinations	Uploads allowed for a group of procedures	File created with no valid owner in the data model	5	6	3	90	DEC-58 (disable uploads for procedure groups); D23, D57
FM-22	Frontend types	Types drift from backend	Compile passes, runtime 500s	6	5	3	90	DEC-80 (Mirror DB types); D50

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-23	Mock mode	Mock diverges from backend	Services pass in mock, fail in prod	6	4	3	72	DEC-87; D59 (mock API removed for final product)
FM-24	Login	Two login paths behave differently	Users may be able to access features through one path but not the other	5	4	3	60	— (No confirmed mitigating DEC; login path consistency requires verification against FR-33, FR-34 and FR-39)
FM-25	Procedure categories	Enum mismatch	Dropdown shows wrong categories	6	5	3	90	DEC-16 (fixed categories retained; enum drift remains a known integration risk); D30
FM-26	Procedure categories	Backend adds a procedure category that is not defined in the frontend	New category does not appear in the selection interface	6	5	4	120	DEC-16 (known limitation: frontend categories stay fixed, so new backend categories need a frontend update); D30
FM-27	Module switching	Form state from the previous module remains after switching modules	Case may be saved under the wrong module	6	4	4	96	— (Module-switch state handling requires verification); D07
FM-28	Popup	`window.confirm` used in new screen	Inconsistent UX, untestable	3	6	3	54	DEC-11, DEC-86 (custom confirmation components); D52
FM-29	Admin/tutor reuse	`AdminViewSwitch` breaks when tutor route changes	Admin view stale	5	4	4	80	DEC-12 (administrator reuse of tutor view); D53
FM-30	Student page	Module/group state, props, or imports missing	Page broken after model change	6	5	3	90	D27 (student-page module and group state restored; no separate DEC recorded)
FM-31	Student page	Member list shows wrong group	Confusing member display	6	4	3	72	D27 (current group members restored; no separate DEC recorded)
FM-32	Student page	Case dropdowns show IDs, not patient names	Hard to pick the right case	5	5	3	75	D27 (student-case endpoint and patient names restored; no separate DEC recorded)

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-33	Group creation	Undefined variables or missing validation	Group not saved, or saved without a case	6	5	3	90	D28 (group creation state and validation fixes; no matching DEC recorded)
FM-34	Project delivery	Remaining features not finished before deadline	Incomplete system, missing requirements	9	7	3	189	D29 (time and scope prioritisation; no separate delivery DEC recorded)
FM-35	PDF generation	Exported PDF contains data that differs from the current case state	Student receives an outdated or incorrect overview	5	5	3	75	DEC-63 (frontend PDF generation; data freshness still requires testing); D31
FM-36	PDF preview	Browser viewer exposes annotation tools	Read-only dossier compromised	4	4	3	48	DEC-64 (Read-only preview); D32
FM-37	Preview/PDF	Different layout operations	Content and pagination drift	5	5	3	75	DEC-65 (Shared layout); D33
FM-38	Export filter	Other groups' logs included	Information leak	8	4	3	96	DEC-66 (Filter selected group); D34
FM-39	Request/response layout	Entries shown separately	Reasoning hard to follow	4	5	3	60	DEC-67 (Combine block); D35
FM-40	PDF export	Content differs from preview	Reviewed content ≠ exported content	5	5	3	75	DEC-65 (shared preview/PDF layout); D33
FM-41	PDF export	Export breaks existing log view	Log view unusable during export work	5	5	3	75	DEC-63 (Isolate export); D31
FM-42	Response linking	Heuristic match attaches wrong response	Wrong decision shown	7	4	4	112	DEC-68 (Explicit request ID); D36
FM-43	Authorship	Names only, no account numbers	Ambiguous authorship	4	5	3	60	DEC-71 (Names + numbers); D37
FM-44	Request heading	Generic or numeric heading	User cannot see what was requested	4	6	3	72	DEC-69 (Procedure name); D38
FM-45	Motivation/response	One continuous block	Roles blur together	3	5	3	45	DEC-72 (Horizontal line); D39
FM-46	Dossier grouping	Flat list of notes and requests	Progression hard to scan	3	4	3	36	DEC-73 (Stadium grouping); D40
FM-47	Response linking	Response attached to wrong request (export)	Wrong decision shown	7	4	4	112	DEC-68 (explicit request identifier for linking); D36

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-48	Export styling	Export theme differs from study	Identity mismatch	4	4	3	48	DEC-70 (Reuse TG styling); D41
FM-49	Export layout	Group and patient info stacked vertically	Wasted vertical space	4	5	3	60	DEC-74 (Side by side); D42
FM-50	Terminology	“fase/fasen” used inconsistently	Confusing UI and export	5	4	3	60	DEC-75 (Stadium/stadia); D43
FM-51	Procedure catalogue	Catalogue fetched again in export	Duplicate network work	4	4	3	48	DEC-76 (Reuse catalogue); D44
FM-52	DICOM viewer	Viewer or DICOM file fails to load	Scan unreadable	5	4	3	60	DEC-88, DEC-89 (DICOM viewer and controls; representative files still need testing)
FM-53	Log rendering	`advance_procedure` not rendered	Missing entry in export	5	5	3	75	DEC-56 (Exhaustive switch); D47
FM-54	Export component input	Full case object passed	Duplicated data, risk of staleness	5	4	4	80	DEC-77 (Pass caseId only); D45
FM-55	Log tracking	Stage advancement or missed-deadline actions are not shown in the activity history	Students and tutors cannot reconstruct when or why progression occurred	6	5	3	90	DEC-57 (distinct progression and missed-deadline action types); D54
FM-56	PDF action handling	New action type not rendered	Silent missing content	5	4	2	40	DEC-78 (TS check); D47
FM-57	File uploads	Uploads allowed for a group of procedures	Orphaned file, no display	5	6	3	90	DEC-58 (Disable uploads); D57
FM-58	PDF preview	Preview fails without showing an error to the student	Student cannot inspect the dossier or understand why the preview is unavailable	5	4	3	60	DEC-64 (preview implementation; failure feedback still requires verification); D56
FM-59	Procedure categories	Teacher needs a procedure category that is not supported by the frontend	Category cannot be selected without updating the application	6	6	3	108	DEC-16 (Known limitation: fixed procedure categories); D30

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-60	List of procedure results	A student cannot find the result they are looking for	Student becomes frustrated	4	5	2	40	DEC-59 (stage and result-type list); D58
FM-61	Display procedure results	The results are not clearly viewable	Student misses important information	7	5	3	105	DEC-52, DEC-59 (readable result viewer and dialog); D16, D58
FM-62	Time constraints	Remaining work does not fit the deadline	Incomplete system, missing requirements	9	7	3	189	D29 (scope and delivery prioritisation; no separate delivery DEC recorded)
FM-63	Dynamic categories	Would require a full rewrite	High rework, deadline risk	8	7	3	168	DEC-16 (Defer to future); D30
FM-64	Tutor navigation	Tutor dashboard contains too much information on one page	Tutor loses overview and needs more effort to find the relevant group or case	5	6	3	90	DEC-02; D60 (Separate pages and click-through navigation)
FM-65	Procedure configuration	Teacher selects multiple procedures of the same type within one stage	Only the first configured procedure is returned to the student	7	5	4	140	DEC-23; D61 (prevent duplicate procedure types within a stage)
FM-66	Tutor request review	Tutor can accept a procedure request that belongs to the wrong stage or is not part of the case	Invalid requests can be approved and the student can receive results inconsistent with the case setup	7	5	4	140	DEC-84; D62 (Highlight the issue and block acceptance)
FM-67	Mixed laboratory requests	A request for blood, urine, or blood-gas tests contains both in-case and out-of-case procedures	A blanket accept/reject rule may incorrectly treat valid and invalid items alike; tutor intent is unclear	6	5	5	150	DEC-85; D63 (Highlight in-case and out-of-case items; keep acceptance available)

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-68	Final product delivery	Mock API setup is kept in the final product instead of being removed	The delivered system carries unused mock code and a second data path, which takes up space and makes the workflow harder to follow for future maintainers	4	4	3	48	DEC-87; D59
FM-70	Procedure request storage	Stored requests and uploaded files exceed the available storage capacity	Uploads or requests may fail, and system performance may deteriorate	9	2	5	90	— (Storage limits and monitoring require verification)
FM-71	Procedure request processing	Processing a request takes too long	Students experience delays and server resources remain occupied longer than necessary	6	5	4	120	— (Performance testing and optimization require verification)
FM-72	Request implementation	Request-handling logic becomes difficult to maintain	Changes take longer and introduce a greater risk of implementation errors	6	7	4	168	DEC-90 (request status stored in the procedure/log linking table)
FM-73	Request transaction	A request fails after only some database operations have completed	Request status, case progression, or activity records may become inconsistent	4	1	6	24	— (Transaction handling requires verification)
FM-74	Request review	An authorized user accepts or rejects a request incorrectly	The student receives an incorrect decision or feedback	3	6	4	72	DEC-84 (stage/case mismatch prevention only; other review errors require verification)
FM-75	Automated request handling	Automated review fails or returns an incorrect decision	Requests remain unanswered or receive incorrect decisions	9	3	6	162	D01 (automation setting at module level; correctness of automated decisions requires separate tests)

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-76	Database deployment	Incompatible database version	Data integrity issues or application crash	8	3	4	96	DEC-91 (PostgreSQL selected; version compatibility still requires testing)
FM-77	Server framework integration	Library conflicts or missing dependencies	Slow down development, deployment fails	7	4	3	84	DEC-92 (Jakarta EE selected; dependency management and testing remain necessary)
FM-79	Team knowledge gap	Slowdown in development due to unfamiliar technology	Deadline missed, features cut	8	4	3	96	DEC-92 (familiar Java framework selected)
FM-80	Document formatting	Poor formatting distracts from content	Unprofessional appearance, lower grade	6	3	2	36	DEC-93 (existing report template chosen)
FM-81	Template rendering	Layout breaks with complex tables	Tables overflow, content unreadable	5	4	3	60	DEC-93 (single-column template selected; wide-table layout still needs checking)
FM-82	LaTeX compilation	Missing packages or compile errors	Delay in submission	4	5	2	40	— (No separate DEC; compile the report regularly)
FM-83	Authentication and authorization	Protected routes fail to validate the session or enforce the required account role	Unauthorized users can access restricted information or perform administrative actions	9	3	4	108	DEC-20 (role distinctions); implementation control: RouteSupport.authenticate / RouteSupport.hasRole with HTTP 401/403 checks (FR-17, FR-18, FR-39, NFR-06)
FM-84	Database connection handling	Database connections remain open after a request completes or an exception occurs	Database resources become unavailable, causing slow responses or failed requests	6	4	3	72	— (Implementation control, not a DEC: RouteSupport.withDatabase and connection cleanup on success and failure)
FM-85	API error handling	Routes return inconsistent error formats or expose internal exception details	The frontend cannot handle failures consistently, or sensitive implementation details are disclosed	7	4	4	112	— (Implementation control, not a DEC: shared RouteSupport JSON error responses and generic client-facing database errors)

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-86	File retrieval	File access is not checked against the authenticated user and their assigned group or case	non users may obtain medical images or other files	9	3	4	108	DEC-97 (OPT-108); `FileRoute` checks authentication and authorization before serving files; FR-09, FR-10, FR-37, NFR-06
FM-87	File-system access	A stored or manipulated file path resolves outside the intended storage directory	Files outside the EPD storage directory may be read, overwritten, or disclosed	9	2	5	90	DEC-98 (OPT-110); file paths are normalized and restricted to the configured storage root
FM-88	File upload	Upload fails after the file has been partially written to disk	Incomplete or corrupted files remain on the server and storage becomes inconsistent	6	4	4	96	DEC-99 (OPT-112); incomplete uploads are removed when an operation fails
FM-89	File registration	File is successfully stored on disk but insertion of the corresponding `StoredFile` record fails	An orphaned file remains without a database reference and cannot be reached through the application	6	3	5	90	DEC-99 (OPT-112); stored files are removed when database registration fails
FM-90	File upload	A user uploads an excessively large file	Server storage or memory may be exhausted, causing failed requests or degraded availability	7	3	3	63	DEC-100 (OPT-114); maximum upload size is enforced by `FileRoute`
FM-91	DICOM processing	Separate `.dcm` files from different series have the same resolution and are grouped together during bulk upload	Slices from different series may be combined into an incorrect medical image	8	3	4	96	DEC-101 (OPT-116); bulk uploads currently group separate DICOM files by resolution, which may combine different series with matching resolutions

FM	Process step	Potential failure	Failure effect	S	O	D	RPN	Related DEC / design log / control
FM-92	Development identity provider	Authentication development depends on university services before access and configuration are available	Login functionality cannot be fully developed or tested, delaying implementation	7	5	3	105	DEC-102 (OPT-86).
FM-93	Migration to Microsoft authentication	The authentication integration is incompatible with Microsoft's configuration, identity claims, or session behaviour	Production login fails or users are linked to an incorrect application account	9	5	3	135	DEC-105 (OPT-119).

Some FMEA entries were removed because they were duplicates or described failure modes that were no longer relevant to the selected implementation. The original numbering was kept, resulting in gaps between identifiers.

E Design Decisions (DEC)

The DEC table records the chosen approach, relevant failure modes, requirements and sub-goals, and reasons for rejecting alternatives. Where an option was compared in the Options tables, its OPT code is included. Smaller implementation decisions are described without an OPT code rather than being assigned an unsupported score. The Related FM column points to failure scenarios that informed the decision; a decision may reduce only part of a failure mode or leave a residual risk. Where no related failure mode is documented, the entry is marked with a dash.

Table 31: Design decisions.

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-01	Menu's	—	Vertical expandable bar (OPT-03)	SG-01, FR-12; the expandable sidebar keeps the tutor navigation available without taking up the entire screen	Horizontal bars (OPT-01): Use more of the available workspace (SG-01, RISK-01). Vertical bars (OPT-02): Do not expand to additional navigation levels (SG-01, RISK-01). Circle-to-ring (OPT-04): An unfamiliar interaction can make navigation harder to discover (SG-01, RISK-01).
DEC-02	Tutor dashboard	FM-64	Separate pages with click-through navigation (OPT-06)	SG-02, FR-12, FR-13, FR-16; FM-64 showed that the split-screen dashboard placed too much information on one page, so tutors navigate into the relevant group/case when needed	Split screen (OPT-05): Retains information overload on one page, conflicting with FR-12 (FM-64, RISK-01). Pop-up (OPT-07): Adds transient interaction state and can obscure context, conflicting with FR-13 (RISK-01).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-03	Requests & notes	FM-08	Pop-up for student requests and notes (OPT-09)	SG-03, FR-03, FR-04, FR-05; students can enter requests and notes without leaving the current case view	Separate page (OPT-08): Requires leaving the case view and returning to continue work (RISK-01). Split screen (OPT-10): Reduces the space available for patient information and requests (RISK-01).
DEC-04	Navigation	—	Minimize nav	SG-01, FR-12; allow the navigation bar to collapse when more workspace is needed	Always expanded: Takes up unnecessary screen space, conflicting with SG-01 (RISK-01).
DEC-05	Sidebar	—	Fixed icon	SG-01, FR-12; a fixed sidebar control avoids moving the navigation target	Repositioning icon: A moving target can make the navigation harder to use consistently (SG-01, RISK-01).
DEC-06	Screen layout	FM-08	Separate info from actions	SG-12, FR-02, FR-04, FR-06	Mixed layout: Confusing UI, conflicting with FR-02 (RISK-01).
DEC-07	Patient vignette	FM-09	2-line vignette	SG-12, FR-02	No vignette: Misses context, conflicting with FR-02 (RISK-01).
DEC-08	Tab structure	FM-15	One tab	SG-12, FR-04, FR-06	Too many tabs: Overwhelming, conflicting with FR-06 (RISK-01).
DEC-09	Header	FM-09	Informative header	SG-12, FR-02; the header keeps patient and case context visible	Not informative: Missing context, conflicting with FR-02 (RISK-01).
DEC-10	Central screen	—	Scrollable	SG-12, FR-02, FR-06; scrolling keeps longer patient information and case content accessible without splitting it across additional screens	Non-scrollable layout: Content may be cut off or become difficult to access (FM-08, RISK-01).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-11	Popup UX	FM-28	Custom `PopupProvider` / `usePopup` (OPT-31)	SG-09, FR-27, NFR-11; reusable custom dialogs make confirmations consistent and avoid duplicating confirmation logic across screens	Native browser dialogs (OPT-82): Do not match the interface styling and allow less control over the confirmation flow (FM-28, RISK-01).
DEC-12	Admin/tutor reuse	FM-29	`AdminViewSwitch`	SG-02, FR-12, FR-13, NFR-11; reuse the tutor view instead of maintaining a separate administrator implementation	Duplicate tutor screen (OPT-33): Similar admin and tutor pages may drift apart, increasing maintenance work and inconsistent navigation (FM-29, NFR-11, RISK-10).
DEC-13	Patient selector	—	Centered	SG-12, FR-02	Misaligned: Poor layout, conflicting with FR-02 (RISK-01).
DEC-14	Requests flow	FM-17	Dedicated tutor review page within the tutor navigation (OPT-06)	SG-02, SG-03, FR-13, FR-14, W-06, W-08; a dedicated review page gives tutors room to examine the request and reasoning without obscuring the group context	Tutor dashboard pop-up (OPT-07): Adds an overlay to an already information-heavy review workflow (FM-17, RISK-01).
DEC-15	Grouping	—	Drag & drop split screen (OPT-11)	SG-04, FR-20, FR-22	Create forms (OPT-12): Time-consuming, conflicting with FR-20 (RISK-05). No D&D split (OPT-13): Slower, conflicting with SG-04 (RISK-01). Student-focussed (OPT-14): Very time-consuming, conflicting with FR-22 (RISK-01).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-16	Procedure categories	FM-25, FM-26, FM-59, FM-63	Fixed categories in frontend code	SG-06, SG-17, FR-04, FR-31, NFR-03; existing procedure categories were retained to avoid late integration work. Adding new categories still requires a frontend update, as documented as a limitation in NFR-03 (FM-26, FM-59)	Dynamic categories (OPT-25, OPT-34): Requires additional integration and testing work that could not be completed within the remaining time (FM-63, RISK-06, RISK-10, RISK-13).
DEC-18	Case–module link	FM-06	Multiple modules	SG-11, FR-19, FR-28	One module per case: Prevents the intended reuse of cases across modules (FR-28, SG-11, RISK-05).
DEC-19	Case reuse	—	Duplicate older cases	SG-11, FR-19, FR-28, W-01, NFR-03; copying an existing case lets staff reuse case content without changing source code	Manual rework: Inefficient for teachers, conflicting with W-01.
DEC-20	Permission model	FM-83	Teacher = highest	SG-07, SG-17, FR-18, FR-26, FR-39, NFR-06; different roles require different permissions, although the role model must still be enforced by backend authorization checks	Flat permission model: Does not distinguish the roles required by FR-39 and may permit unauthorized actions if not checked correctly (FM-83).
DEC-21	Case pages	—	Combine create + edit	SG-10, FR-19, NFR-03; a shared create/edit form lets staff configure ordinary case content without source-code changes	Two separate pages: Adds navigation between case creation and editing, making the workflow less direct (SG-10, RISK-01).
DEC-22	Cases per group	FM-06	Multiple	SG-11, FR-21	One case per group: Does not meet the requirement to assign multiple cases to a group (FR-21, RISK-05).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-23	Exam selection per case	FM-05, FM-65	Teacher selects examinations; duplicate procedure types blocked within a stage (OPT-59)	SG-06, SG-08, SG-10, FR-19, FR-24, FR-43, W-01, NFR-03; teachers configure examinations per case without changing source code. Duplicate procedure types are blocked within one stage because only the first duplicate would otherwise be returned to students; new procedure categories still require a frontend update	Fixed examination set: Prevents teachers from configuring case content (FM-05, W-01). Allow duplicate procedure types: Later procedures of the same type may not be returned to students (FM-65, FR-43, RISK-02).
DEC-24	Module management	—	Add button	SG-07, SG-17, FR-31, NFR-03; give authorized staff a visible way to create educational modules	No module creation action: Staff cannot initiate the module creation workflow from the interface (FR-31, RISK-01).
DEC-25	Module self-service	—	Teacher creates	SG-07, SG-17, FR-31, NFR-03; teachers can create modules without asking developers to change source code	Fixed module structure: Staff would have to request code changes to manage modules, conflicting with FR-31 and NFR-03.
DEC-26	CSV parsing	—	Raw CSV to backend	SG-04, FR-20; send the CSV content to the backend for consistent parsing and validation	Frontend parses CSV independently: Could introduce inconsistent group data handling between frontend and backend (RISK-05).
DEC-27	Group management	—	Remove on add	SG-04, FR-22, W-05; remove a student from the available list after assignment to make duplicate assignment less likely	Keep in list: Duplicate assignments, conflicting with W-05 (RISK-05).
DEC-28	Tutor management	—	Count memberships	SG-04, FR-22; membership counts help staff see how many groups are assigned to each tutor	No membership counter: Makes tutor workload less visible when assigning groups (RISK-05).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-29	Student list	—	Student numbers	SG-04, FR-20, FR-22, W-05	Names only: Cannot distinguish students, conflicting with FR-22 (RISK-01).
DEC-30	Drag & drop colors	—	Green highlight	SG-04, FR-20, FR-22; highlighting drop targets makes it clearer where students will be assigned	No highlight: Unclear drop targets, conflicting with FR-20 (RISK-01).
DEC-31	Group onboarding	—	Clearer UI	SG-04, FR-20, FR-22, W-05	Unclear UI: Slow onboarding, conflicting with FR-20 (RISK-05).
DEC-32	Create group button	—	Visible	SG-04, FR-20; keeping the group creation action visible makes it easier to start the workflow	Hidden button: Cannot create groups, conflicting with FR-20 (RISK-05).
DEC-33	Group assign lock	—	Lock page	SG-04, FR-22, NFR-06; lock the assignment interface when changes should no longer be made. This UI lock does not replace server-side authorization checks	No assignment lock: Users could unintentionally change group assignments after they have been finalized (RISK-05).
DEC-34	Submit confirmation	FM-28	Popup	SG-10, FR-19, W-01; a confirmation dialog reduces accidental submissions when saving case changes	No confirmation: Unintended changes may be submitted without an additional check (RISK-05).
DEC-35	Case start state	—	Stage 1 with no procedures selected	SG-10, FR-19, FR-23, W-01, NFR-03; new cases start in a defined state before teachers configure their content	Uninitialized case state: Could leave the case without a clear starting stage (RISK-05).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-36	Examination selection	—	Categorized selection menu (OPT-16)	SG-06, FR-04, FR-05; grouping procedures by main category and subcategory makes the correct test easier to find than a single flat list	Flat dropdown (OPT-15): Makes it harder to find the right examination (FR-04, RISK-01, RISK-02). Checkboxes (OPT-17): Large lists can obscure related examination categories (RISK-01).
DEC-37	Results view	FM-11	Organize examination results using tabs and buttons (not separately scored)	SG-13, FR-06, W-07, W-11; separating results makes individual examinations easier to find and read	One large text block: Makes results difficult to scan and can hide important information (FM-11, RISK-09).
DEC-38	Results interaction	FM-12	Clickable	SG-13, FR-06, W-11; results should be clearly selectable instead of appearing as static labels	Static text: Students may not realize that individual examination results can be opened (FM-12, W-11).
DEC-39	Phase locking	FM-03	Locked by deadline/admin	SG-08, FR-23, FR-25	Open progression: Students skip steps, conflicting with FR-23 (RISK-02).
DEC-40	Stage dates	FM-04	Optional + soft end-date	SG-08, FR-23, NFR-03; staff can configure optional stage dates without changing source code	Hard cutoff: A group may be blocked even when the teacher intends to allow progression (FM-04, RISK-02).
DEC-41	GP referral	FM-14	Base element	SG-12, FR-02, W-12	Not shown: Missing context, conflicting with FR-02 (RISK-01).
DEC-42	Request list	FM-01	Virtually infinite	SG-06, FR-04, FR-05; students may reason about and request examinations even when no prepared result exists. Tutor validation remains governed by DEC-84 and DEC-85	Only allow prepared examinations: Prevents students from considering tests beyond the prepared result set (RISK-02).
DEC-43	Phase management	FM-03	Fase 0 → Fase 1	SG-08, FR-23, W-02	No transition: Cannot start case, conflicting with FR-23 (RISK-02).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-44	Notification link	FM-17	Links to review page	SG-02, SG-03, FR-13, W-08; link notifications directly to the relevant tutor review page	Hard to find: Missed requests, conflicting with W-08 (RISK-01).
DEC-45	Exam request label	FM-10	“aanvullend onderzoek”	SG-06, SG-09, FR-04, FR-46, W-13; use the agreed Dutch term for supplementary examinations	Keep “Onderzoek aanvragen: The client preferred the term “aanvullend onderzoek for this workflow (FR-46, W-13).
DEC-46	Supplementary exams	FM-13	Decoupled	SG-12, FR-02, FR-04, FR-06; keep supplementary examinations separate from the patient information block	Coupled: Confusing structure, conflicting with FR-06 (RISK-01).
DEC-47	Logging interface	—	Log panel on the right	SG-14, FR-08, FR-16, W-09; the panel makes recorded activity visible while users work with the case	No visible log panel: Users would need to navigate elsewhere to review recorded actions (RISK-04).
DEC-48	Deadline visibility	FM-18, FM-19	Prominent + flag	SG-14, FR-16, FR-23, W-04	Small text: Teacher misses it, conflicting with W-04 (RISK-03).
DEC-49	Teacher overview	FM-20	Full case incl. hidden	SG-15, FR-41, W-03; teachers can inspect the full case, including information not yet released to students	Separate work file: Cannot assess, conflicting with W-03 (RISK-03).
DEC-50	Work file	FM-20	Integrated	SG-15, FR-41, W-03; keep the supporting teacher work information in the case rather than a separate file	Manual file: Out of context, conflicting with W-03 (RISK-03).
DEC-51	Info release status	FM-20	Show later/never	SG-15, FR-41, W-03; distinguish information released to students from information held for later or not released	Not shown: Wrong assumptions, conflicting with W-03 (RISK-03).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-52	Result viewer	FM-61	Elaborate result viewer	SG-13, FR-06, FR-07, W-11; give students sufficient space to inspect results and medical images	Basic viewer: Results and images may be difficult to inspect, affecting FR-07 (FM-61, RISK-09).
DEC-53	Exam request discoverability	FM-10	More visible	SG-03, SG-06, FR-04; make the examination request entry point easier for students to find	Poor placement: Students may overlook the entry point for requesting an examination (FM-10, RISK-01).
DEC-54	Imaging support	FM-07	CT + thorax photos	SG-13, FR-07; support CT and thorax images in the case results	No imaging support: Students cannot inspect the medical images required for diagnostic cases (FR-07, RISK-09).
DEC-55	Blood work selection	FM-02	Selection list	SG-06, FR-04, FR-42, W-10; a selection list lets students choose the individual blood values required for a laboratory request	Free text: Makes it harder to select the intended individual blood values consistently (FR-42, W-10, RISK-02).
DEC-56	Log action handling	FM-53	advance_procedure	SG-14, FR-08, FR-16, W-09; render the progression action as a distinct activity-history entry	Single `note` action: Progression events are not represented as a distinct type in the log (FM-53, RISK-04).
DEC-57	Log tracking	FM-55	Support `advance_procedure` and `deadline_missed` (OPT-65)	SG-14, FR-08, FR-16, W-04, W-09; the frontend can distinguish progression and missed-deadline events from ordinary notes when displaying the activity history	Represent all events as notes (OPT-54): Makes progression events harder to identify and handle consistently (FM-55, RISK-04).
DEC-58	File upload restriction	FM-21, FM-57	Disabled for labs	SG-06, SG-16, FR-04, FR-24, FR-38; group-level laboratory uploads were disabled because files need an unambiguous procedure owner	Allow uploads for entire procedure groups (OPT-35): The resulting file may not have a clear procedure owner, risking invalid or orphaned data (FM-21, FM-57, FR-38, RISK-05).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-59	Displaying results	FM-61	Dialog list (OPT-57)	SG-13, FR-06, FR-07, W-07, W-11	Display all results at once (OPT-71): Makes long result lists and images difficult to inspect (FM-61, RISK-09). Accordion rows (OPT-72): Several open results can clutter the page (RISK-09). Separate result pages (OPT-73): Adds navigation between related results (RISK-01).
DEC-60	Test data	—	Verify 5–6 results per case	SG-13, FR-06, FR-15; representative case data should cover several examination results so the display can be checked with realistic content	Use an untested assumption about result volume: Layout problems may remain unnoticed until actual cases contain more results (RISK-09).
DEC-61	Overview progress	FM-46	Acceptance timeline (OPT-18)	SG-05, FR-08, FR-16, FR-29, FR-30; chosen because it provides a clear overview of progress without relying on a complex tree structure that did not visibly improve the exported document	Progress stage tree (OPT-19): Did not clearly improve the exported overview and needed more implementation time (RISK-10). Point/grade score (OPT-20): Introduces grading rather than describing the group's progress (SG-05). Copy of raw log (OPT-21): Makes the progression harder to scan and interpret (FM-46, RISK-04).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-62	Overview format	—	PDF (OPT-22)	SG-05, FR-29, FR-30, W-14; PDF gives the overview a consistent layout while reducing casual editing. It is not tamper-proof (RISK-14)	DOCX (OPT-23): Can be edited directly in common word processors and may display differently between applications (RISK-07, RISK-14). TXT (OPT-24): Can be edited directly and does not preserve the images and formatting needed for the dossier (RISK-07, RISK-14).
DEC-63	EPD PDF generation	FM-35, FM-41	Generate PDF in frontend (OPT-36)	SG-05, FR-29, FR-30; uses the case data already available in the frontend and avoids implementing a separate PDF generation endpoint	Backend PDF endpoint (OPT-37, OPT-46): Requires additional implementation and integration work within the remaining project time (RISK-10, RISK-12, RISK-13).
DEC-64	PDF preview	FM-36, FM-58	Read-only HTML/SVG preview (OPT-38)	SG-05, FR-29, FR-30; allows students to inspect the dossier without exposing editing controls in the preview interface	Browser PDF viewer (OPT-39, OPT-47): Built-in annotation tools provide less control over the read-only preview (RISK-14).
DEC-65	Preview/PDF consistency	FM-37, FM-40	Same layout (OPT-40)	SG-05, FR-29, FR-30	Build separately (OPT-41): Content drifts, conflicting with FR-30 (RISK-07).
DEC-66	Export scope	FM-38	Selected group only (OPT-42)	SG-05, FR-09, FR-10, FR-29, NFR-06, NFR-10; limit the export to the selected group to avoid including other groups' information. Server-side authorization remains necessary	All groups' logs (OPT-43): May expose information belonging to another group (FR-09, FR-10, RISK-08).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-67	Request/ response	FM-39	One block (OPT-44)	SG-05, FR-08, FR-13, FR-30	Separate request and response entries (OPT-45): The clinical motivation and tutor decision become harder to follow as one exchange (FM-39, RISK-04).
DEC-68	Response linking	FM-42, FM-47	Explicit ID (OPT-48)	SG-05, FR-13, FR-30	Heuristic match (OPT-49): Wrong attachment, conflicting with FR-13 (RISK-04).
DEC-69	Request headings	FM-44	(type) name + decision (OPT-50)	SG-05, FR-04, FR-30	Generic request heading (OPT-51): Readers cannot easily identify the examination requested (FM-44, RISK-04).
DEC-70	Export styling	FM-48	Reuse TG styling and Univers fonts (OPT-52)	SG-05, FR-29, W-14, NFR-11; reuse the Technical Medicine styling rather than maintaining a separate export theme	Separate export theme (OPT-53): Requires additional styling work and creates another set of styles to maintain (NFR-11, RISK-07, RISK-10).
DEC-71	Authorship	FM-43	Names, numbers, time	SG-05, FR-08, FR-30, W-09	Names only: Multiple users may have similar names, making action attribution ambiguous (FR-08, W-09, RISK-04).
DEC-72	Motivation/ response separator	FM-45	Line	SG-05, FR-30; visually separate the student motivation from the tutor response in the dossier	No clear separator: Student motivation and tutor response can appear as one continuous text block (FM-45, FR-30, RISK-04).
DEC-73	Dossier grouping	FM-46	By stadium	SG-05, FR-08, FR-30, W-14	Flat list: Progression hard to scan, conflicting with FR-30 (RISK-04).
DEC-74	Export layout	FM-49	Side by side	SG-05, FR-29, W-14; placing group and patient details next to each other uses the available page space more efficiently	Vertical stacking: Uses more page space and makes the dossier less compact (FM-49, W-14).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-75	Stage terminology	FM-50	stadium/stadia	SG-09, FR-46, W-13; use the Dutch terminology agreed with the client	fase/fasen: Inconsistent, conflicting with W-13.
DEC-76	Procedure catalogue	FM-51	Reuse dashboard	SG-05, FR-29, FR-30, NFR-11; reuse existing procedure information rather than maintaining a separate catalogue-fetching path for export	Fetch catalogue again: Repeats work and introduces another dependency during PDF generation (NFR-11, RISK-07, RISK-10).
DEC-77	Export input	FM-54	caseId only	SG-05, SG-16, FR-29, FR-30, NFR-06, NFR-11; pass a case identifier and retrieve data from one source to reduce duplicated, stale input. Server-side access to the requested case must still be verified	Pass full case object: Duplicates already available data and can make the export depend on stale inputs (FM-54, RISK-07).
DEC-78	PDF action handling	FM-56	Switch + TS	SG-05, FR-29, FR-30, NFR-11; an explicit action-type switch makes missing action types easier to identify and future log actions easier to add	Chain of conditions: Newly introduced action types are easier to miss, causing missing content in the exported dossier (FM-56, FR-30, RISK-07).
DEC-79	Frontend framework	—	React (OPT-58)	SG-18, SG-01, NFR-11, FR-12; React was chosen for component reuse and state management in the frontend	Vanilla JavaScript (OPT-26): Would require building and maintaining more component and state-management code (NFR-11, RISK-10). WordPress (OPT-27): Does not fit the custom EPD workflows and selected deployment setup (RISK-06, RISK-12).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-80	Frontend types	FM-22	Use shared frontend types aligned with the backend data model (not separately scored)	SG-16, SG-18, NFR-11; aligning shared types with the backend reduces inconsistent assumptions across frontend services	Separate DTOs for every screen (OPT-29): Duplicated type definitions may diverge from the API contract (FM-22, RISK-06).
DEC-81	Frontend service layer	FM-22	Split domain	SG-16, SG-18, NFR-11; domain-specific service modules keep frontend API operations easier to maintain	Combined service functions: Makes API changes harder to isolate and maintain (FM-22, NFR-11, RISK-06).
DEC-82	Mock data organisation	FM-23	Split routes	SG-18, NFR-11; during development, separating mock routes helped keep test data organized. The mock setup was later removed from the final product (DEC-87)	Monolithic mock data file: Makes development-only mock routes harder to maintain and test (FM-23, NFR-11, RISK-06).
DEC-84	Tutor request validation	FM-66	Highlight mismatch and disable Accept (OPT-60)	SG-06, SG-08, FR-14, FR-44; requests outside the active stage or case are highlighted and cannot be approved. Mixed laboratory requests follow the exception in DEC-85.	Allow acceptance despite the mismatch (OPT-61): Invalid requests could be approved, conflicting with FR-44 (FM-66, RISK-02).
DEC-85	Mixed laboratory requests	FM-67	Keep Accept available and highlight each in-case or out-of-case item (OPT-63)	SG-06, FR-14, FR-45; mixed blood, urine and blood-gas requests remain reviewable, with each item marked as inside or outside the case. The final policy on whether a mixed request should be accepted is not yet agreed	Block the entire mixed request (OPT-62): Could prevent valid items from being accepted before the teaching policy is confirmed (FM-67, RISK-02).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-86	Delete control	FM-28	Custom `RemoveButton` with confirmation popup (OPT-32)	SG-04, SG-19, FR-20, FR-27; the reusable delete control asks for confirmation before removing group or case information.	Native `confirm()` (OPT-64): Offers less control over consistent deletion confirmations (FM-28, RISK-05).
DEC-87	Mock setup in final product	FM-68	Rip out the mock setup and use the real backend for the final product (OPT-67)	SG-18, NFR-11; the mock setup was useful for development but would introduce an unused alternative data path and extra maintenance in the delivered application (FM-68, RISK-06)	Keep the mock API in the final product (OPT-66): the unused mock code and second data path take up space and make the workflow harder to follow for future maintainers, which works against the handover goal (RISK-06). A secondary reason is that mock responses can conceal backend integration defects (FM-23). Removing the mock too early would have blocked frontend work before the backend was ready, so the removal was timed at final integration, not at the start (D59).
DEC-88	DICOM rendering	FM-52	Cornerstone3D (OPT-74)	Supports browser-based DICOM viewing and image manipulation without a custom renderer (FR-07, NFR-11).	Full viewer platform (OPT-75): Unnecessary integration complexity (RISK-09). Custom implementation (OPT-76): Higher development and maintenance effort (FM-52, NFR-11, RISK-09).
DEC-89	DICOM viewer controls	FM-52	Mouse gestures with on-demand help (OPT-77)	Mouse gestures were selected to keep the image area free of unnecessary controls. A help button explains the available interactions when needed (FR-07).	Persistent controls (OPT-78): Take up space that could otherwise be used to display the scan (RISK-09). Gestures without help (OPT-79): Users may not know which controls are available for inspecting the image (RISK-09).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-90	Request handling	FM-72	Store request status in the linking table (OPT-81)	SG-16, FR-04, FR-13, FR-14, NFR-11; storing request status in the existing link between procedures and logs keeps the status associated with the request without introducing another table	Attach status to log (OPT-80): The current status would need to be found and filtered from log entries, complicating request handling (FM-72). Separate request object (OPT-83): Adds an extra object and lookup path without a clear benefit for the required workflow (FM-72). Attach status to group (OPT-84): Cannot represent multiple independent pending requests for the same group (FR-13, FR-14).
DEC-91	Database selection	FM-76	PostgreSQL (OPT-89)	SG-16, SG-18, NFR-09, NFR-11; PostgreSQL supports the relational case, group, request and review model and was familiar to the team. Backup and recovery still require separate testing	MySQL (OPT-90): Also suitable, but PostgreSQL was preferred because of team familiarity. SQLite (OPT-91): Simpler deployment, but less suitable for the intended multi-user server setup. Microsoft SQL Server (OPT-92) and Oracle (OPT-93): Would introduce additional licensing or deployment considerations without a clear benefit for this project. MongoDB (OPT-94) and CouchDB (OPT-95): Document-oriented models were not preferred for the relational structure of cases, groups, requests, and reviews.

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-92	Server framework	FM-77, FM-79	Jakarta EE (OPT-96)	SG-18, NFR-11; Jakarta EE was selected because the team had relevant Java experience and could develop the required API within the project schedule	Spring Boot (OPT-97): A viable Java alternative, but would require additional framework-specific setup and learning. Flask (OPT-98) and Django (OPT-99): Would introduce a Python-based stack that the team was less familiar with. Laravel (OPT-100): Would require adopting a PHP framework rather than using the selected Java stack. .NET (OPT-101): Would introduce a C# development environment that the team was less familiar with (FM-79).
DEC-93	Document formatting	FM-80, FM-81	Springer LNCS (OPT-103)	The LNCS template was selected because its single-column layout accommodates the report's tables and figures, and the team was already familiar with it	No template (OPT-102): Would require creating and maintaining the document formatting ourselves (FM-80). NJD-v5 (OPT-104) and BSCP (OPT-105): Their two-column layouts make the report's wide tables more difficult to fit and read (FM-81).
DEC-94	Request Automation	–	Module toggle	FR-25, W-15; We attached the automation toggle to the module class, since the module object is already available when handling the request and because teachers may want all cases in a module to be automated together	Attaching the toggle to the patient case was rejected since the client wanted to toggle all cases in a module at once together instead of having to automate them separately (W-15)

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-95	Stage advancement	–	Automatic advancement based on correct procedures, deadlines and start dates	This allows for the groups to progress through stages without always needing approval, while still giving the teacher control on how quick groups advance to the next stage	Always asking for approval: Would slow down progress and annoy the teacher too much. Automatic advancement based on only correct procedures: Allows students to complete a case way before they are supposed too and gives no control to teaher.
DEC-96	Group Management	–	CSV-based group import (OPT-106)	CSV import was selected because it allows teachers to populate multiple groups at once, reducing repetitive work when configuring student groups.	Manual assignment (OPT-107): Requires teachers to assign students individually, increasing the number of interactions needed to configure groups (RISK-01). Repeated manual assignments also increase the opportunity for incorrect or missing group memberships (RISK-05).
DEC-97	File access	FM-86	Dedicated `FileRoute` with authentication and authorization (OPT-108)	A dedicated file route was selected to centralize access checks and ensure that users can only retrieve or modify files they are authorized to access.	Direct static file serving (OPT-109): Does not provide the required per-user and per-group authorization checks, potentially exposing medical images to unauthorized users (FM-86, RISK-08).
DEC-98	File-system access	FM-87	Restrict file paths to a configured storage root (OPT-110)	Resolving and normalizing file paths against a configured storage root was selected to prevent file operations from accessing locations outside the intended storage directory.	Unrestricted file paths (OPT-111): Manipulated paths could allow access to files outside the intended storage directory, leading to unauthorized file disclosure or modification (FM-87, RISK-08).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-99	File consistency	FM-88, FM-89	Remove incomplete uploads and unregistered files (OPT-112)	Failed uploads and database registration operations trigger cleanup to prevent incomplete or unregistered files from remaining on the server.	Retain files after failed operations (OPT-113): Partial uploads could leave corrupted files on disk (FM-88), while failed database registration could leave files without a corresponding database record (FM-89, RISK-15).
DEC-100	Upload limits	FM-90	Enforce a maximum upload size through `FileRoute` (OPT-114)	An upload size limit was selected to prevent individual uploads from consuming excessive server resources and affecting the availability of the application.	No application-level upload limit (OPT-115): Excessively large files could exhaust available memory or storage and cause requests to fail (FM-90, RISK-15).
DEC-101	DICOM processing	FM-91	Group by `SeriesInstanceUID` and validate compatibility (OPT-116)	Grouping by `SeriesInstanceUID` was selected to distinguish between DICOM series when uploading separate files. The current bulk upload implementation groups files by resolution, which may combine different series if their resolutions match (FM-91).	Combine slices without series validation (OPT-117): Files from different or incompatible series could be combined into an unreadable or medically incorrect image (FM-91, RISK-09).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-102	Authentication	FM-83, FM-92	Keycloak as the development OIDC provider (OPT-86)	SG-18, NFR-02, NFR-11; Keycloak allows the complete OIDC login flow to be developed and tested with controlled accounts before university access and configuration are available (FM-92). It avoids implementing a separate primary authentication system. Microsoft remains the intended production provider under FR-01 and FR-33; backend authorization checks remain necessary regardless of the provider (FM-83, NFR-06).	Custom login system (OPT-85): Adds authentication implementation, maintenance, and security testing work (FM-83, FM-92, RISK-13). Direct UT Microsoft authentication during development (OPT-118): Makes development and testing dependent on university access and app registration, which may delay implementation (FM-92, RISK-12). This alternative is rejected for development, not for production.
DEC-103	Account hierarchy	—	Subclass hierarchy (OPT-87)	A shared base account class allows common account information and functionality to be reused while keeping role-specific properties in separate subclasses.	Individual account classes (OPT-88): Would duplicate common account fields and functionality, making changes harder to maintain consistently (NFR-11, RISK-10).
DEC-104	Role changes	FM-83	Apply role changes using Save (OPT-68)	Role changes remain pending until the administrator selects Save and confirms the changes. This allows multiple account roles to be reviewed before they are applied.	Immediate role changes (OPT-69): Changes would take effect before the administrator could review them, increasing the chance of unintended permission changes (FM-83, NFR-06). Separate edit dialogs (OPT-70): Would require additional steps when changing several accounts, making the process less convenient (RISK-01).

DEC	Area	Related FM	Chosen Option	Decision rationale / evidence	Rejected alternatives / reason + evidence codes
DEC-105	Microsoft authentication migration	FM-93	Adapt the existing configurable OIDC integration for direct UT Microsoft login (OPT-119)	SG-18, FR-01, FR-33, NFR-02, NFR-07; the selected handover approach reuses the existing OIDC flow and application authorization instead of replacing the authentication integration. UT's app registration and environment settings must be supplied, and token exchange, trusted identity mapping, and logout must be adapted where necessary. End-to-end tests for students, tutors, and administrators are required before production use. Compatibility with UT's Microsoft tenant has not yet been validated.	Separate Microsoft-specific login implementation (OPT-120): Requires replacing and retesting the existing integration, increasing handover work and regression risk (FM-93, RISK-10). Keycloak as a production identity broker (OPT-121): Can support university login, but introduces another identity service to configure, secure, back up, and maintain (FM-93, NFR-01, NFR-11, RISK-12).

Some decisions do not reference specific options or risks, as they were made based on the team's technical expertise and experience rather than a formal comparison of alternatives. Some entries were also removed because they duplicated existing decisions, resulting in gaps in the numbering.

F Option Scoring Matrix

The Option Scoring Matrix compares alternatives using the metrics in Table 11. Each total is the weighted sum of the nine metric scores. A higher score does not automatically determine the selected option: requirements, risks, technical constraints, and previous decisions also affect the choice. Selected options are highlighted in green. The weights for SM-01 through SM-09 are 5, 5, 4, 3, 1, 5, 3, 4, 3, respectively.

Table 32: Scoring matrix for all options.

OPT	SM-01	SM-02	SM-03	SM-04	SM-05	SM-06	SM-07	SM-08	SM-09	Total
OPT-01	4	4	3	3	4	5	2	4	4	124
OPT-02	4	4	3	3	4	5	2	4	4	124
OPT-03	5	4	4	3	4	5	4	4	4	139
OPT-04	2	3	2	2	4	5	3	3	3	98
OPT-05	4	4	4	3	4	5	3	4	4	131
OPT-06	4	4	5	3	4	5	3	4	4	135
OPT-07	4	4	3	3	4	5	3	4	4	127
OPT-08	4	4	3	3	4	5	3	4	4	127
OPT-09	4	4	4	3	4	5	3	4	4	131
OPT-10	3	4	3	3	4	5	3	4	4	122
OPT-11	4	4	4	3	4	5	4	4	4	134
OPT-12	3	4	3	3	4	5	4	4	4	125
OPT-13	4	4	3	3	4	5	3	4	4	127
OPT-14	3	4	2	3	4	5	3	4	4	118
OPT-15	4	4	3	3	4	5	3	4	4	127
OPT-16	4	4	4	3	4	5	4	4	4	134
OPT-17	4	4	3	3	4	5	2	4	4	124
OPT-18	4	4	4	3	4	5	4	4	4	134
OPT-19	4	4	4	3	4	5	4	4	4	134
OPT-20	2	3	3	3	4	5	3	4	4	112
OPT-21	4	4	2	3	4	5	3	4	4	123
OPT-22	4	4	4	3	4	5	4	4	4	134
OPT-23	4	4	3	3	4	5	3	4	4	127
OPT-24	4	4	2	3	4	5	2	4	4	120
OPT-25	2	4	4	3	4	5	4	4	4	124
OPT-26	2	3	3	3	4	5	3	3	2	102
OPT-27	2	3	2	3	4	5	2	3	1	92
OPT-28	3	4	3	3	4	5	3	3	4	118
OPT-29	3	3	3	3	4	5	2	3	3	107
OPT-30	4	3	3	3	4	5	3	3	3	115
OPT-31	4	4	4	4	4	5	4	4	5	140
OPT-32	4	4	4	4	4	5	4	4	5	140
OPT-33	3	4	3	3	4	5	3	3	3	115
OPT-34	3	4	3	3	4	5	3	3	3	115
OPT-35	3	3	3	3	4	5	3	3	4	113
OPT-36	4	4	4	3	4	5	4	4	4	134
OPT-37	3	4	4	3	4	5	4	4	4	129
OPT-38	4	4	4	4	4	5	4	4	4	137
OPT-39	5	4	3	3	4	5	4	4	4	135
OPT-40	4	4	4	3	4	5	4	4	4	134
OPT-41	3	4	3	3	4	5	3	4	4	122
OPT-42	4	4	4	3	4	5	4	4	4	134
OPT-43	2	4	2	3	4	5	4	4	4	116
OPT-44	4	4	4	3	4	5	4	4	4	134
OPT-45	4	4	3	3	4	5	4	4	4	130
OPT-46	3	4	4	3	4	5	4	4	4	129
OPT-47	5	4	3	3	4	5	4	4	4	135
OPT-48	4	4	4	3	4	5	4	4	4	134

OPT	SM-01	SM-02	SM-03	SM-04	SM-05	SM-06	SM-07	SM-08	SM-09	Total
OPT-49	3	4	2	3	4	5	3	3	4	114
OPT-50	4	4	4	3	4	5	4	4	4	134
OPT-51	4	4	2	3	4	5	4	4	4	126
OPT-52	4	4	4	3	4	5	4	4	5	137
OPT-53	2	3	3	3	4	5	3	4	2	106
OPT-54	4	4	2	3	4	5	3	4	4	123
OPT-55	4	4	2	3	4	5	3	5	4	127
OPT-56	2	4	4	3	4	5	5	3	3	120
OPT-57	4	4	5	3	4	5	4	4	4	138
OPT-58	4	4	4	4	4	5	4	4	5	140
OPT-59	4	4	5	4	4	5	4	4	5	144
OPT-60	4	4	5	4	4	5	4	4	4	141
OPT-61	3	4	2	3	4	5	3	4	4	118
OPT-62	3	4	3	3	4	5	3	4	4	122
OPT-63	4	4	4	4	4	5	4	4	4	137
OPT-64	2	3	2	3	4	5	2	4	2	99
OPT-65	5	4	3	3	4	5	4	4	4	135
OPT-66	4	2	2	3	3	5	2	3	3	102
OPT-67	4	4	3	3	3	5	4	4	4	129
OPT-68	5	4	4	4	5	5	4	4	4	143
OPT-69	4	4	3	4	5	4	3	4	3	123
OPT-70	3	4	3	4	4	3	2	4	4	112
OPT-71	4	4	3	3	2	5	3	4	3	122
OPT-72	4	4	4	3	2	5	4	4	4	132
OPT-73	3	4	3	3	2	4	3	4	3	112
OPT-74	4	4	5	3	2	4	4	4	4	131
OPT-75	2	3	5	4	2	2	5	4	3	109
OPT-76	2	3	4	3	3	2	3	2	3	89
OPT-77	5	4	4	2	5	5	4	4	4	137
OPT-78	3	3	5	4	4	3	4	5	4	125
OPT-79	5	4	2	1	5	5	4	3	3	119
OPT-80	2	2	3	3	3	2	3	2	3	80
OPT-81	4	4	4	3	4	4	4	4	4	129
OPT-82	5	3	3	3	4	5	3	3	2	117
OPT-83	2	2	3	3	2	2	3	3	2	80
OPT-84	3	3	3	3	3	4	2	3	3	101
OPT-85	3	2	3	3	3	2	3	3	3	89
OPT-86	4	4	4	4	3	4	4	4	4	131
OPT-87	4	4	4	3	3	4	4	4	4	128
OPT-88	3	2	3	3	3	2	2	3	2	83
OPT-89	5	5	4	5	5	4	5	5	5	156
OPT-90	3	4	4	4	5	4	4	5	4	132
OPT-91	3	4	3	4	5	2	4	4	3	111
OPT-92	3	4	4	5	1	5	4	4	4	132
OPT-93	3	4	4	5	1	5	4	4	4	132
OPT-94	4	3	3	2	5	3	3	3	3	103
OPT-95	4	3	3	3	5	3	3	3	3	106
OPT-96	5	4	5	5	5	4	5	5	5	155
OPT-97	4	4	4	5	5	4	4	5	4	140
OPT-98	2	4	4	4	5	4	4	4	4	123
OPT-99	2	4	4	4	5	4	4	4	4	123
OPT-100	2	3	3	1	5	3	3	3	3	90
OPT-101	2	4	4	4	5	4	3	3	4	116
OPT-106	4	4	5	3	4	4	5	4	5	139
OPT-107	5	4	2	4	3	5	2	5	3	128
OPT-108	4	5	4	4	4	4	5	5	5	147
OPT-109	5	2	2	3	4	5	2	2	2	101
OPT-110	5	5	4	4	4	5	5	5	5	157
OPT-111	5	2	2	3	4	5	2	2	2	101
OPT-112	4	5	4	4	4	4	5	5	5	147
OPT-113	5	2	2	3	3	5	2	2	2	100

OPT	SM-01	SM-02	SM-03	SM-04	SM-05	SM-06	SM-07	SM-08	SM-09	Total
OPT-114	5	5	4	4	5	5	5	5	5	158
OPT-115	5	2	2	3	2	5	2	2	2	99
OPT-116	3	4	4	4	4	3	5	4	5	128
OPT-117	5	2	2	3	4	5	2	2	2	101

Some options were removed because they were no longer relevant or were covered by other entries. The original numbering was retained, resulting in gaps in the option IDs.

G Design Log

Table 33: Chronological design log of the EPD project.

ID / Date(s)	Design record
D01 07–08 Sep 2026	**Issue / requirement:** Level at which procedure-request automation is configured. **Alternatives considered:** Configure automation separately for each request, or configure it at a broader case/module level. **Criteria / risks:** Administrative effort; consistency of behaviour; ability of teaching staff to control the educational workflow. **Decision / status: Selected.** Automation is configured at module level, with the administrator choosing the setting while the module/case is configured. **Evidence / evolution:** Automation granularity was an explicit open question on 07 September. The requirements meeting on 08 September selected module-scale configuration controlled by the administrator.
D02 07–08 Sep 2026	**Issue / requirement:** Whether tutors must explain an approval or rejection of a diagnostic request. **Alternatives considered:** No explanation field; mandatory explanation; optional explanation. **Criteria / risks:** Traceability and communication versus unnecessary mandatory input for routine decisions. **Decision / status: Selected.** The system permits an explanation for an approval or rejection, but the explanation is optional. **Evidence / evolution:** The question was raised on 07 September and resolved in the 08 September requirements meeting.
D03 07–08 Sep 2026	**Issue / requirement:** Structure of the exported overview of a group's work. **Alternatives considered:** A fixed report format; a case-dependent format; a mandatory or optional conclusion; ordering entries by category or by actual occurrence. **Criteria / risks:** Ability to reconstruct the student's clinical reasoning; differences between cases; configurability for teaching staff. **Decision / status: Selected.** Actions are shown in their actual chronological order. Report structure may vary by case, and teaching staff can determine whether a conclusion is required. **Evidence / evolution:** Report contents and ownership were unresolved on 07 September. The 08 September requirements meeting specified chronological presentation and case-dependent configuration.

ID / Date(s)	Design record
D04 07–08 Sep 2026	**Issue / requirement:** Retention and deletion of student data. **Alternatives considered:** Keep student data indefinitely, or permit deletion after a defined retention period. **Criteria / risks:** Long-term storage; administrative control; avoiding indefinite retention of old student records. **Decision / status: Selected requirement.** Student data may be deleted by academic year after two years, with deletion reserved for a super-administrator. **Evidence / evolution:** The deletion period was an open question on 07 September and was specified in the 08 September requirements meeting.
D05 07–08 Sep 2026 02 Oct 2026 06 Oct 2026	**Issue / requirement:** Case phases, progression, and deadline handling. **Alternatives considered:** No phases; manually controlled phases; completion-based unlocking; deadline-based progression; or a combination of completion and deadline rules. **Criteria / risks:** Educational sequencing; preventing premature access to information; ensuring students do not become permanently blocked; handling missed deadlines consistently. **Decision / status: Revised target design.** A case can contain multiple phases with phase-specific examinations and information. A later phase is unlocked when the required work in the current phase is completed. Each phase can also have a deadline; if the deadline is missed, the group can be advanced automatically so the educational process continues. **Evidence / evolution:** On 08 September, stages and required tests were discussed but progression was still unresolved. Prototype feedback on 02 October added optional dates, phase locking and soft deadlines. Teacher feedback on 06 October made the model concrete by requiring both completion-based unlocking and automatic progression after a missed deadline.
D06 07–08 Sep 2026 06 Oct 2026	**Issue / requirement:** Which diagnostic examinations students may request. **Alternatives considered:** Show only a fixed case-specific list; let teachers select available examinations/categories; or allow students to request items beyond the prepared case data. **Criteria / risks:** Educational realism; teacher control over prepared case content; support for learning from inappropriate requests; avoiding an interface that teaches students the answer set by construction. **Decision / status: Revised design.** Decision / status: Revised design. Teachers configure the examinations... For the student's selection interface, a categorized selection menu was chosen (main category followed by subcategory) over a flat list or autocomplete, as it provides a clear structure for a large number of procedures. **Evidence / evolution:** The 07 September meeting raised whether all tests should be visible. On 08 September, case-specific teacher selection was preferred as a nice-to-have. On 06 October, teachers explicitly requested both teacher-configured examinations and an effectively open-ended request list.

ID / Date(s)	Design record
D07 07–08 Sep 2026 02 Oct 2026	**Issue / requirement:** Case creation and maintenance workflow. **Alternatives considered:** Continue the previous workflow in which a teacher supplies a Word document for an administrator to import; allow teaching staff to create cases directly in the EPD; keep separate create/edit screens or combine them into a case-management view. **Criteria / risks:** Administrative workload; maintainability; ownership of case content; dependence on technical staff; usability for repeated editing. **Decision / status: Selected direction.** Teaching staff should be able to create and configure cases directly in the system. The 02 October feedback further proposed combining creation and editing into a clearer case overview. **Evidence / evolution:** Direct case creation by the teacher was selected during the 08 September requirements meeting. Prototype feedback on 02 October identified the create/edit workflow as an area for further refinement.
D08 07–08 Sep 2026	**Issue / requirement:** Representation of individual and collaborative student work. **Alternatives considered:** Model individual students separately from groups, or use the same group abstraction for all case assignments, including one-person groups. **Criteria / risks:** Simplicity of authorization and assignment logic; consistency between individual and collaborative cases. **Decision / status: Selected.** All case work is represented through groups; individual work is represented by a group containing one student. **Evidence / evolution:** The need for separate individual support was raised on 07 September and resolved on 08 September by using the group model for both situations.
D09 08 Sep 2026 02 Oct 2026	**Issue / requirement:** Number of patient cases that can be associated with a group. **Alternatives considered:** Restrict every group to one case, or allow a group to work on multiple cases. **Criteria / risks:** Data-model complexity; reuse of groups; flexibility across modules and educational formats; avoiding assumptions that would obstruct future extension. **Decision / status: Revised target design.** The initial one-case-per-group requirement was changed to support multiple cases per group. **Evidence / evolution:** The 08 September requirements meeting explicitly stated one case per group. During the 02 October prototype evaluation, the requirement was revised to allow multiple cases per group.

ID / Date(s)	Design record
D10 08 Sep 2026 02 Oct 2026	**Issue / requirement:** Overlap between tutor, teacher, and administrative permissions. **Alternatives considered:** Keep the roles strictly separate, or use a hierarchy in which higher teaching roles can also perform tutor functions. **Criteria / risks:** Least privilege; avoiding duplicate workflows; practical need for teaching staff to review requests as well as manage modules and cases. **Decision / status: Selected direction.** Higher teaching roles have cumulative capabilities: teachers and administrators can access tutor functionality in addition to their management functions. **Evidence / evolution:** The 08 September meeting stated that administrators can act as tutors. The 02 October feedback clarified that teachers are the highest permission level, can approve or reject requests, and administrative users need access to tutor pages.
D11 07–08 Sep 2026 02 Oct 2026 06 Oct 2026	**Issue / requirement:** Relationship between cases, modules, academic years, and student enrolment. **Alternatives considered:** Recreate cases each year; copy complete modules; keep reusable cases separate from modules; or link cases to a module-year while allowing reuse and students to participate in more than one module. **Criteria / risks:** Reuse of teaching material; maintenance effort; correct association with the academic year; support for students enrolled in multiple modules; risk of staff selecting the wrong cohort. **Decision / status: Selected direction.** Cases are explicitly linked to a module-year using an academic-year label such as 2026/2027; earlier cases can be duplicated for reuse; the model must not assume that a student belongs to only one module. **Evidence / evolution:** Reuse and module/case relationships were raised as open questions on 07 September. On 02 October, feedback requested module-year linkage, academic-year labels, and duplication of older cases. On 06 October, teacher feedback additionally required cases to be linked to modules while allowing a student to participate in multiple modules.

ID / Date(s)	Design record
D12 02 Oct 2026 06 Oct 2026	**Issue / requirement:** Discoverability and organisation of supplementary examination functionality. **Alternatives considered:** Keep the existing navigation; add more separate tabs; link notifications directly to request/review pages; or create one clear supplementary-examination area separated from patient information. **Criteria / risks:** Discoverability; screen complexity; number of interactions; risk of users overlooking pending requests or confusing patient information with actions. **Decision / status: Revised design.** Supplementary examinations are separated from the patient-information block and presented as one clear area rather than many separate tabs. The label “onderzoek aanvragen” is changed to “aanvullend onderzoek”. Relevant notifications should lead users toward the corresponding request/review workflow. **Evidence / evolution:** On 02 October, both the student request function and tutor review page were difficult for users to find. Teacher feedback on 06 October specified the clearer separation and renamed the supplementary-examination area.
D13 02 Oct 2026	**Issue / requirement:** Group-assignment interaction. **Alternatives considered:** Retain the existing drag-and-drop behaviour unchanged, or add stronger interaction cues and assignment-state indicators. **Criteria / risks:** Discoverability; prevention of accidental assignments; visibility of student identity; clarity of the current assignment state. **Decision / status: Revision requested.** Keep the drag-and-drop approach but improve it with valid-drop highlighting, a hand cursor, student numbers, clearer selected states, and visible indicators such as check marks. A lock for the assignment page was also proposed. **Evidence / evolution:** Multiple observations during the 02 October demonstration showed that adding students to groups and interpreting assignment state were not immediately clear.
D14 02 Oct 2026	**Issue / requirement:** Protection against consequential human input errors. **Alternatives considered:** Apply important changes immediately, require confirmation before selected actions, or make operations reversible where practical. **Criteria / risks:** Consequence of accidental actions; interruption caused by excessive confirmations; traceability and recoverability. **Decision / status: Proposed control.** Important submissions should use a confirmation step where the cost of an accidental action justifies it, and the system design should document how significant user errors can be prevented or recovered from. **Evidence / evolution:** The 02 October feedback explicitly proposed an “Is all data correct?” confirmation for important submissions and requested that human-error handling be addressed in the report.

ID / Date(s)	Design record
D15 06 Oct 2026	**Issue / requirement:** Separation of patient context from actions on the student screen. **Alternatives considered:** Keep patient information and actions together, or retain patient context as a compact persistent vignette while moving actions into separate controls. **Criteria / risks:** Continuous visibility of clinical context; separation between information and actions; efficient use of screen space; avoidance of visual clutter. **Decision / status: Revision requested.** Patient information is shown on the left as a compact vignette, while supplementary examinations and EPD export are separate actions. The huisartsverwijzing remains a base element of the case view. **Evidence / evolution:** Teacher feedback on 06 October identified the existing left-side design as too tightly coupling patient information with actions and made this restructuring a top priority.
D16 06 Oct 2026	**Issue / requirement:** Presentation of examination results. **Alternatives considered:** Display results as one long text block, create a separate page/tab for every result, or use a compact selector with individually clickable results. **Criteria / risks:** Readability; number of results per case; navigation overhead; ability to compare or move between results quickly. **Decision / status: Revision requested.** Results are individually selectable instead of being combined into one large block. A compact row of tabs or buttons can be used for result selection, while supplementary examinations remain grouped in a single main area. **Evidence / evolution:** Teacher feedback on 06 October estimated approximately 5–6 results in a typical case and up to 7–8 in larger cases, making a compact result selector practical.

ID / Date(s)	Design record
D17 06 Oct 2026	**Issue / requirement:** Teacher visibility of case progress, hidden information, and missed deadlines. **Alternatives considered:** Show staff only the information currently released to students and retain supporting information in separate files, or provide one complete teacher view containing the full case, activity history, unreleased information, and deadline status. **Criteria / risks:** Oversight; traceability; timely intervention; risk of missed deadlines going unnoticed; duplication and inconsistency caused by information stored outside the EPD. **Decision / status: Revision requested.** Teachers receive a full-case overview, including information that is not yet or never released to students. The activity log records requests, approvals, and the responsible user. Missed deadlines are made visually prominent with a flag/alert, and information from the separate student-assistant work file should be integrated into the EPD. **Evidence / evolution:** These items were identified as top priorities during the 06 October teacher feedback session.
D18 06 Oct 2026	**Issue / requirement:** Granularity of imaging and laboratory requests. **Alternatives considered:** Use only broad examination categories, or expose clinically meaningful examination types and selectable laboratory values. **Criteria / risks:** Educational realism; clarity of the student's requested investigation; compatibility with available result data; interface complexity. **Decision / status: Requirement added.** Imaging should support CT scans and thorax images, and blood-work requests should let the student select the specific value to be tested. **Evidence / evolution:** Both changes were requested directly in the 06 October teacher feedback session.
D19 06 Oct 2026	**Issue / requirement:** Consistency between a request, its approval, and the eventual result. **Alternatives considered:** Treat approval and result availability as one implicit state, or model and display them as distinct steps in the workflow. **Criteria / risks:** Correctness; traceability; avoiding the impression that an approved examination already has a result; reliable frontend/backend synchronization. **Decision / status: Clarification and test action.** The interface and backend should distinguish clearly between requested, approved/rejected, and result-released states. The separate issue in which requested examinations sometimes failed to appear must be investigated through testing rather than being treated as an intended design behaviour. **Evidence / evolution:** On 06 October, teachers identified both an inconsistency where approved examinations appeared without results and a server issue where requested examinations sometimes did not appear.

ID / Date(s)	Design record
[D20–D22 *gap*]	**[D20–D22 were planned but never written. They are kept as a numbering gap so that references from the Decisions and Risks tables still point to the intended positions.]**
D23 16 Oct 2026	**Issue / requirement:** Whether file uploads are appropriate for examination categories that are groups of procedures rather than single procedures. **Alternatives considered:** Allow uploads for all categories; attach one file to every procedure in the group; or disable uploads for categories that are groups of procedures and require a specific procedure or value instead. **Criteria / risks:** Correctness of the data model; prevention of files with no valid owner; consistency between what a category is and what can own an attachment. **Decision / status: Selected.** Uploads are disabled for Bloedonderzoek, Urineonderzoek and Bloedgasanalyse, and any previously selected files are cleared when switching to one of these categories. **Evidence / evolution:** These categories are technically groups of procedures. A file can only be attached to a concrete procedure, not to a group, so allowing an upload would create a file with no valid owner in the data model.
D27 19 Oct 2026	**Issue / requirement:** After the data-model changes, the student page had lost module and group state, sidebar properties and imports, and the member list and patient details no longer matched the current group and case. **Alternatives considered:** Patch the page incrementally, or restore the missing state and properties and load patient details through the student-case endpoint. **Criteria / risks:** Correct student context; visible patient names in case dropdowns; consistency with the new data model; risk of stale or missing data. **Decision / status: Resolved.** Module and group state, sidebar properties and imports were restored. The member list now uses the current group's members, and patient details are loaded through the student-case endpoint so that case dropdowns show patient names. **Evidence / evolution:** The page was broken by the model change. Restoring state and loading patient details through the correct endpoint brings the page back in line with the new data model.

ID / Date(s)	Design record
D28 20 Oct 2026	**Issue / requirement:** Creating a new group failed on undefined variables in the save logic, groups were not built from their draft member and case assignments, and a group with no case could be saved without feedback. **Alternatives considered:** Fix only the undefined variables; also build groups from their drafts; also add validation feedback for missing cases. **Criteria / risks:** Group-creation correctness; consistency with the draft member and case assignments; preventing invalid groups from being saved; usability of the validation message. **Decision / status: Resolved.** Undefined variables in the new-group save logic were fixed, groups are built from their draft member and case assignments, and a visible validation message is shown when a new group has no case. **Evidence / evolution:** The flow was failing silently and accepting groups without a case. The fixes make the flow correct and give the user clear feedback.
D29 20 Oct 2026	**Issue / requirement:** The remaining work may not fit the project deadline, leaving some requirements unfinished. **Alternatives considered:** Attempt to finish everything; cut nice-to-have features; or reduce scope to the must-have requirements and freeze features two weeks before the deadline. **Criteria / risks:** Delivering a working system on time; maintaining quality on the must-have requirements; avoiding last-minute rework. **Decision / status: Selected control.** A weekly scope review is held, must-have and nice-to-have requirements are triaged explicitly, and features are frozen two weeks before the deadline. **Evidence / evolution:** With the number of open items at the time of this decision, finishing everything was not realistic. Prioritising the must-have requirements protects the core product and the deadline.

ID / Date(s)	Design record
D30 20 Oct 2026	**Issue / requirement:** Examination categories are defined statically in code. A more flexible design would render categories and subcategories dynamically from the server, letting users add new ones at runtime. **Alternatives considered:** Render categories and subcategories from server data; keep the current static categories for this release; or load some categories dynamically and keep the rest static. **Criteria / risks:** Flexibility for teaching staff; effort to rewrite the current code; database rework; risk to the deadline; stability of already-working features. **Decision / status: Not implemented.** Categories and subcategories remain hardcoded in the frontend for the duration of this project. Dynamically rendering them from the server is not implemented because the amount of frontend and backend code that would need to be rewritten is too large for the time remaining in the module. **Evidence / evolution:** The project is too far along to rewrite every place that uses categories and to rework the backend enum. That rewrite would put already-working features at risk and is not worth it within the remaining time. Dynamic loading is recorded as a rejected option and as a project risk.
D31 20 Oct 2026	**Issue / requirement:** The EPD export needed to produce a PDF, but the backend did not provide a dedicated PDF-export endpoint. **Alternatives considered:** Add a backend PDF-export endpoint, or generate the PDF in the frontend from the existing case logs. **Criteria / risks:** Reuse of existing endpoints; implementation effort; consistency of the exported document with the on-screen data. **Decision / status: Selected.** The EPD PDF is generated in the frontend using the case logs. **Evidence / evolution:** Reusing the existing logs endpoint avoids adding a separate backend implementation and keeps the export consistent with what the frontend already displays.
D32 20 Oct 2026	**Issue / requirement:** The browser's built-in PDF viewer exposes annotation tools, which are not appropriate for a dossier that is meant to be read-only. **Alternatives considered:** Use the browser's PDF viewer, or build a read-only HTML/SVG preview. **Criteria / risks:** Read-only behaviour of the dossier; retention of printing and downloading; consistency with the visual style of the application. **Decision / status: Selected.** A read-only HTML/SVG preview is used instead of the browser's PDF viewer. **Evidence / evolution:** The custom preview removes the annotation controls while still allowing the document to be printed and downloaded.

ID / Date(s)	Design record
D33 20 Oct 2026	**Issue / requirement:** The preview and the downloaded PDF could differ in content, styling or pagination if they were built separately. **Alternatives considered:** Build the preview and the PDF separately, or generate both from the same layout operations. **Criteria / risks:** Consistency between preview and export; maintainability of the layout code; risk of the exported document not matching what the user reviewed. **Decision / status: Selected.** Preview and PDF are generated from the same layout operations. **Evidence / evolution:** Sharing the layout operations keeps content, styling and pagination consistent between what the user sees and what is exported.
D34 20 Oct 2026	**Issue / requirement:** The dossier must contain only the selected group's logs. Including other groups' notes and requests would leak information. **Alternatives considered:** Include all logs the case logs endpoint returns, or filter to the selected group. **Criteria / risks:** Correctness of the export; confidentiality between groups; consistency with how the rest of the frontend scopes data. **Decision / status: Selected.** Only logs belonging to the selected group are included in the export. **Evidence / evolution:** Filtering prevents other groups' notes and requests from appearing in the exported dossier.
D35 20 Oct 2026	**Issue / requirement:** Requests and their tutor responses were displayed as separate log entries, which made the reasoning hard to follow. **Alternatives considered:** Keep requests and responses as separate entries, or combine each request and its tutor responses into one block. **Criteria / risks:** Readability of the reasoning; length of combined blocks; consistency with the rest of the dossier layout. **Decision / status: Selected.** Each request and its tutor responses are combined into one block. **Evidence / evolution:** Keeping the student's motivation and the tutor's decision together makes the reasoning easier to follow.
D36 21 Oct 2026	**Issue / requirement:** Responses must be attached to the correct request; a wrong match would display a tutor's decision against the wrong motivation. **Alternatives considered:** Match responses to requests heuristically, or match by explicit request ID with an unambiguous fallback. **Criteria / risks:** Correctness of the pairing; risk of silently displaying an incorrect decision; transparency when a match cannot be made. **Decision / status: Selected.** Responses are linked using an explicit request ID. A fallback is used only when the match is unambiguous, and ambiguous matches produce an error rather than silently showing incorrect information. **Evidence / evolution:** Using an explicit ID avoids attaching a tutor's response to the wrong request, and surfacing ambiguous cases prevents incorrect information from being hidden.

ID / Date(s)	Design record
D37 21 Oct 2026	**Issue / requirement:** The dossier needed to make authorship and the timing of requests and decisions clear. **Alternatives considered:** Show only names, or show names together with account numbers, dates and times. **Criteria / risks:** Clarity of authorship; ability to distinguish students and tutors with similar names; traceability of when actions happened. **Decision / status: Selected.** Names and account numbers for students and tutors are shown, together with dates and times. **Evidence / evolution:** Showing account numbers as well as names makes authorship and the timing of requests and decisions clear in the exported document.
D38 21 Oct 2026	**Issue / requirement:** Request headings used the generic label "Onderzoeksaanvraag" or numeric procedure references, which made it hard to see what was actually requested. **Alternatives considered:** Keep the generic heading; use a numeric reference; or use the (type) procedure name followed by the decision. **Criteria / risks:** Immediate readability of the request; consistency with the procedure catalogue; risk of the reader having to look up a numeric reference elsewhere. **Decision / status: Selected.** Headings show the (type) procedure name followed by the decision. **Evidence / evolution:** Identifying the actual investigation in the heading avoids the generic or numeric label that made requests hard to read.
D39 21 Oct 2026	**Issue / requirement:** The student's motivation and the tutor's response appeared as one continuous block, which made their different roles hard to see. **Alternatives considered:** Add separate request and response headings, or separate them with a horizontal line. **Criteria / risks:** Clarity of who wrote what; length of the block; avoidance of unnecessary extra headings. **Decision / status: Selected.** The student's motivation and the tutor's response are separated by a horizontal line. **Evidence / evolution:** A horizontal separator distinguishes the two contributions without adding extra headings.

ID / Date(s)	Design record
D40 21 Oct 2026	**Issue / requirement:** Notes and requests were not grouped by stage, which made it hard to scan the dossier's progression. **Alternatives considered:** Show notes and requests as a flat list, or group them by stadium in softly shaded, outlined containers. **Criteria / risks:** Readability of the progression; visual noise; distinguishability between notes, approvals and rejections. **Decision / status: Selected.** Notes and requests are grouped by stadium in softly shaded, outlined containers. Individual blocks have rounded outlines, with subtle colours distinguishing notes, approvals and rejections. **Evidence / evolution:** Grouping by stadium makes the dossier's progression easier to scan and the subtle colours distinguish the three kinds of block at a glance.
D41 22 Oct 2026	**Issue / requirement:** The exported dossier needed to match the visual identity of the study's materials. **Alternatives considered:** Build a new export-specific theme, or reuse the existing application's TG styling and Univers fonts. **Criteria / risks:** Visual consistency with the study; implementation effort; availability of the supplied theme. **Decision / status: Selected.** The existing application's TG styling and Univers fonts are used. The supplied Office-theme link required authentication, so the existing application theme was used with the client's approval. **Evidence / evolution:** Reusing the existing theme aligns the export with the study's visual identity without introducing a separate theme. The client approved using the existing theme after the supplied link could not be accessed.
D42 22 Oct 2026	**Issue / requirement:** Group and patient information were displayed one above the other, which used vertical space inefficiently. **Alternatives considered:** Keep the vertical layout, or place group and patient information side by side under a prominent EPD header. **Criteria / risks:** Efficient use of space; clarity of the document hierarchy; readability on print. **Decision / status: Selected.** Group and patient information are placed side by side under a prominent EPD header. **Evidence / evolution:** Placing them side by side uses space efficiently and gives the document a clearer hierarchy.

ID / Date(s)	Design record
D43 22 Oct 2026	**Issue / requirement:** The term “fase/fasen” was used in the interface, but the client wanted “stadium/stadia” used consistently. **Alternatives considered:** Keep “fase/fasen”, or switch to “stadium/stadia” throughout the frontend. **Criteria / risks:** Terminology consistency across the interface and the export; consistency with the study’s materials. **Decision / status: Selected.** “Stadium/stadia” is used throughout the frontend, including the exported document. **Evidence / evolution:** Using one term throughout keeps the interface and the export consistent with the study’s terminology.
D44 22 Oct 2026	**Issue / requirement:** Resolving procedure names and categories for the export required procedure information, which the student dashboard had already loaded. **Alternatives considered:** Fetch the procedure catalogue again in the export component, or reuse the catalogue already loaded by the dashboard. **Criteria / risks:** Consistency of the resolved names; number of network requests; risk of the reused catalogue going stale. **Decision / status: Selected.** The export reuses the procedure catalogue already loaded by the student dashboard. **Evidence / evolution:** Reusing the catalogue avoids fetching the same information again just to resolve names and categories.
D45 23 Oct 2026	**Issue / requirement:** The export component required a separate case object alongside the studentCase, which duplicated data and risked going stale. **Alternatives considered:** Pass a full case object to the export component, or pass only the case ID. **Criteria / risks:** Duplication of data; risk of stale values; ability to perform stale-case checks before rendering. **Decision / status: Selected.** Only the case ID is passed to the export component. **Evidence / evolution:** The export needs only the selected case’s ID alongside the detailed studentCase. Keeping the ID separate still allows loading and stale-case checks.

ID / Date(s)	Design record
D46 23 Oct 2026	**Issue / requirement:** The frontend needed to be ready for the planned `advance_procedure` action, which is not yet implemented in the backend. **Alternatives considered:** Wait until the backend is implemented, or prepare frontend support now. **Criteria / risks:** Frontend readiness; dead code if the backend is never implemented; consistency with how actions are rendered in the dossier. **Decision / status: Selected.** Frontend support for `advance_procedure` is prepared, without implementing it in the backend. Its stage identifies the stadium being completed, so it appears after that stadium and outside its container. **Evidence / evolution:** Preparing frontend support makes the frontend ready for the planned action while leaving backend implementation to its owner.
D47 23 Oct 2026	**Issue / requirement:** PDF action handling needed a structure that would make new action types easy to add. **Alternatives considered:** Handle actions with a chain of conditions, or use a switch with an exhaustive TypeScript check. **Criteria / risks:** Extensibility of action handling; risk of silently missing a new action type; ease of reading the rendering logic. **Decision / status: Selected.** A switch is used for PDF action handling, with an exhaustive TypeScript check that flags new action types that are not yet handled. **Evidence / evolution:** A switch with an exhaustive check gives each action an explicit rendering path and makes additional types easier to add without silently missing cases.
D48 16 Oct 2026	**Issue / requirement:** Choice of frontend framework and development workflow. **Alternatives considered:** React with Node development environment; vanilla JavaScript; WordPress. **Criteria / risks:** Team familiarity; component model; separation between frontend and backend; maintainability. **Decision / status: Selected.** React with a separate Node development environment. **Evidence / evolution:** Nijs had React experience and set up the initial project structure. Vanilla JS was rejected because the project needs a component model. WordPress was rejected as inappropriate for a CS design project.

ID / Date(s)	Design record
D49 16 Oct 2026	**Issue / requirement:** How the frontend is served alongside the backend. **Alternatives considered:** Single servlet page; React served at `/TG-EPD/` with API at `/TG-EPD/api/...`. **Criteria / risks:** Clean separation between UI and API; ability to run frontend separately during development; deployability as a single WAR. **Decision / status: Selected.** React served at `/TG-EPD/`, API at `/TG-EPD/api/...`. Frontend can also be run separately with `npm start`. **Evidence / evolution:** Nijs integrated the frontend into the WAR via Maven, while retaining a separate Node development environment.
D50 16 Oct 2026	**Issue / requirement:** How to organise frontend types to match the database. **Alternatives considered:** Ad-hoc per-screen DTOs; types mirroring the database with composite types named by consumer. **Criteria / risks:** Type safety; consistency between frontend and backend; readability of the type definitions. **Decision / status: Selected.** Types in `types/` mirror the database tables. Composite types are named by the user who sees them: `AdminCase`, `AdminStage`, `AdminGroup`, `StudentCase`, `ReviewerCase`. **Evidence / evolution:** Jasper rebuilt the frontend types to align with the database diagram after the schema changes. Nijs aligned his frontend branch to the same types.
D51 16 Oct 2026	**Issue / requirement:** How the frontend switches between mock data and the real backend. **Alternatives considered:** Multiple scattered flags; a single `USE_BACKEND` flag in `config.ts`. **Criteria / risks:** Simplicity; ability to demo without a working backend; testability. **Decision / status: Selected.** A single `USE_BACKEND` flag in `config.ts` controls whether the frontend uses mock data or the real API. **Evidence / evolution:** Jasper introduced the flag so the system could be demoed and developed without a fully working backend.
D52 16 Oct 2026	**Issue / requirement:** How users confirm or dismiss important actions in the frontend. **Alternatives considered:** `window.confirm` / `window.alert`; a single `PopupProvider` / `usePopup` component. **Criteria / risks:** Consistent UX; testability; avoiding browser-native dialogs that break the visual style. **Decision / status: Selected.** A single `PopupProvider` / `usePopup` is used for all confirmations and error cards. **Evidence / evolution:** Jasper added the popup handler to replace scattered `window.confirm` calls. The custom popups were chosen over native browser dialogs because they prevent accidental confirmations, look better, and align with the TG visual style. **Evidence / evolution:** Jasper added the popup handler to replace scattered `window.confirm` calls.

ID / Date(s)	Design record
D53 16 Oct 2026	**Issue / requirement:** How admins access the tutor environment. **Alternatives considered:** Duplicate the tutor screen inside the admin screen; expose the existing tutor environment through an `AdminViewSwitch`. **Criteria / risks:** Code reuse; consistency of behaviour between admin and tutor; risk of duplication drift. **Decision / status: Selected.** Admin opens the existing tutor environment through `AdminViewSwitch`. The route guard has an `allowAdmin` option. **Evidence / evolution:** A duplicate tutor screen was built and then removed. Reusing the existing environment ensures consistency.
D54 22 Oct 2026	**Issue / requirement:** How the frontend should render the newly added `advance_procedure` and `deadline_missed` log actions. **Alternatives considered:** Leave the new actions unrendered; render `advance_procedure` after the completed stadium and outside its container; render both normally; render `deadline_missed` in red. **Criteria / risks:** Readability of the export PDF; differentiation between natural progression and forced progression; teacher's ability to spot missed deadlines at a glance. **Decision / status: Selected.** Frontend support for `advance_procedure` is prepared. Its stage identifies the stadium being completed, so it appears after that stadium and outside its container. Missed deadlines are highlighted in red. **Evidence / evolution:** Jasper and Brian agreed on the log action scheme. Jasper pushed the frontend support while the backend action was still being finalized.
D55 23 Oct 2026	**Issue / requirement:** Where to generate the EPD PDF export. **Alternatives considered:** Backend PDF endpoint; frontend generation from the existing case logs endpoint. **Criteria / risks:** Reuse of existing endpoints; implementation effort; consistency with on-screen data. **Decision / status: Selected.** The PDF is generated in the frontend from the case logs. **Evidence / evolution:** Jasper asked three days in a row whether the backend PDF was being implemented. No answer came, so he removed the backend endpoint and moved PDF generation to the frontend.
D56 23 Oct 2026	**Issue / requirement:** How the EPD PDF is previewed before download. **Alternatives considered:** Use the browser's built-in PDF viewer; build a read-only HTML/SVG preview. **Criteria / risks:** Read-only behaviour; retention of printing and downloading; consistency with the visual style. **Decision / status: Selected.** A read-only HTML/SVG preview is used instead of the browser's PDF viewer. **Evidence / evolution:** The custom preview removes the annotation controls while still allowing printing and downloading.

ID / Date(s)	Design record
D57 23 Oct 2026	**Issue / requirement:** Whether files may be attached to a procedure category in the frontend. **Alternatives considered:** Allow uploads everywhere; disable uploads for Bloedonderzoek, Urineonderzoek and Bloedgasanalyse and clear selected files when switching to them. **Criteria / risks:** Correctness of the data model; prevention of files with no valid owner; consistency with what a category represents. **Decision / status: Selected.** Uploads are disabled for Bloedonderzoek, Urineonderzoek and Bloedgasanalyse; any previously selected files are cleared when switching to one of these categories. **Evidence / evolution:** These categories are groups of procedures, not single procedures. A file can only be attached to a concrete procedure, not to a group.
D58 21 Oct 2026	**Issue / requirement:** How to display results from multiple requested procedures to students, which can include images. **Alternatives considered:** Full page per result; Inline list; List with expandable rows (accordion); Separate drawer. **Criteria / risks:** Easy to find specific result; Clearly readable results. **Decision / status: Selected.** Display a list of procedure names that are clickable, clicking brings up a dialog with results. **Evidence / evolution:** The textual list takes up minimal space and is easily readable and searchable. The dialog popup provides a large window in which to display images, for better detail.
D59 28 Oct 2026	**Issue / requirement:** The prototype still contained a mock API setup, which was useful during early development but was no longer appropriate for the final product. **Alternatives considered:** Keep the mock API in the final product; retain a runtime switch between mock and real data; remove the mock API setup and use the real backend for the final product. **Criteria / risks:** Avoiding differences between mock and production behaviour; ensuring integration tests exercise the real API; reducing the chance that the final build accidentally uses mock data. **Decision / status: Selected.** The mock API setup was removed in preparation for the final product. The final product uses the real backend rather than relying on mock responses. **Evidence / evolution:** The mock setup supported development before backend integration. As the project moved toward the final product, the team removed it to reduce the divergence risk described by FM-23 and RISK-06; see DEC-87.

ID / Date(s)	Design record
D60 28 Oct 2026	**Issue / requirement:** The tutor dashboard placed too much information on one page, making it difficult to locate the relevant group or case. **Alternatives considered:** Keep a split-screen dashboard; use a pop-up; separate the tutor workflow into pages and navigate to the selected group/case. **Criteria / risks:** Information overload; ability to retain context; speed and clarity of tutor navigation. **Decision / status: Selected.** The split-screen option was not chosen. Separate pages with click-through navigation to the relevant group or case were chosen (OPT-06, DEC-02). **Evidence / evolution:** The observed information overload was recorded as FM-64 and informs RISK-01.
D61 28 Oct 2026	**Issue / requirement:** Teachers could configure multiple procedures of the same type within one stage even when no distinct subcategories separated them. **Alternatives considered:** Allow repeated procedure types; prevent selecting a procedure type again within the same stage. **Criteria / risks:** Correct delivery of configured procedures to students; avoiding hidden or missing procedures; retaining teacher control over the stage's procedure selection. **Decision / status: Selected.** The interface prevents teachers from selecting another procedure of the same type within the same stage (OPT-59, DEC-23). **Evidence / evolution:** Without this restriction, the student received only the first procedure of a repeated type. This failure mode is recorded as FM-65 and linked to RISK-02.
D62 28 Oct 2026	**Issue / requirement:** A tutor could accept a student's procedure request even when the procedure belonged to the wrong stage or was not part of the case. **Alternatives considered:** Allow acceptance and rely on the tutor to notice the mismatch; highlight the mismatch and disable Accept. **Criteria / risks:** Preventing invalid approvals; making the reason for a blocked decision visible; maintaining the relationship between requests, stages, and case configuration. **Decision / status: Selected.** The interface highlights the problem with the request and disables the Accept button when a request is invalid for the stage or case. Mixed blood, urine, and blood-gas requests use the separate exception in DEC-85 (OPT-60, DEC-84). **Evidence / evolution:** This addresses FM-66 and is linked to RISK-02.

ID / Date(s)	Design record
D63 28 Oct 2026	**Issue / requirement:** Blood, urine, and blood-gas requests can contain multiple procedures, some included in the case and others outside it; the team had not established whether a tutor should accept such a mixed request. **Alternatives considered:** Block the whole request if any procedure is outside the case; leave acceptance available and highlight which procedures are in-case and which are out-of-case. **Criteria / risks:** Avoiding accidental approval of an out-of-case item while not automatically blocking valid in-case items; making the ambiguity visible to the tutor without imposing an unagreed policy. **Decision / status: Selected pending policy clarification.** The Accept button remains available, while each procedure is highlighted as in-case or out-of-case (OPT-63, DEC-85). This is a deliberate unresolved policy decision, not a claim that every mixed request should always be accepted. **Evidence / evolution:** This behaviour records the current compromise for FM-67 and RISK-02. A future teaching-policy decision may change whether mixed requests can be accepted as a whole or require item-level decisions.
D64 29 Oct 2026	**Issue / requirement:** The progress stage tree (OPT-19) was initially considered alongside the acceptance timeline (OPT-18) to give reviewers a quick visual overview of progress. **Alternatives considered:** Use only the acceptance timeline (OPT-18); use only the progress stage tree (OPT-19); combine both (OPT-18 + OPT-19). **Criteria / risks:** Clarity and visual appeal of the exported PDF; available development time; risk of the export becoming cluttered or unfinished (RISK-10). **Decision / status: Revised.** OPT-19 (progress stage tree) was not chosen. The acceptance timeline (OPT-18) remains the chosen solution for the overview progress. **Evidence / evolution:** During final review, the stage tree did not provide a clear enough benefit to the look of the exported document. Because time was limited, the team decided to focus on polishing the acceptance timeline (OPT-18) rather than implementing and validating a second, complex visual component.

EPD

Studenten handleiding

Inhoudsopgave

1 Inloggen 2

2 Navigatie en Patiëntinformatie 2

3 Notities Toevoegen 3

4 Onderzoeken Aanvragen 4

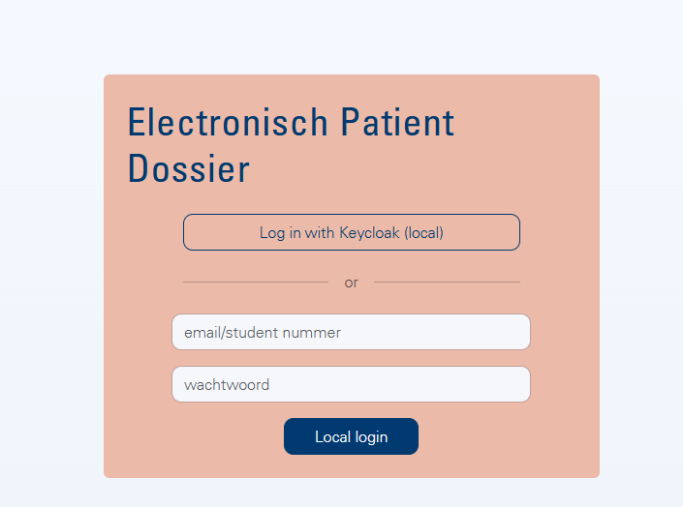
5 Testresultaten Bekijken 5

6 EPD Exporteren 5

1 Inloggen

Wanneer je naar het Elektronisch Patiëntendossier (EPD) gaat, zie je het inlogscherm.

- **Keycloak Login:** Klik op de bovenste knop om veilig in te loggen met je Universiteit Twente-account (Canvas).
- **Local Login:** Voor testdoeleinden kun je de onderste velden gebruiken om in te loggen met een lokaal e-mailadres/studentnummer en wachtwoord.

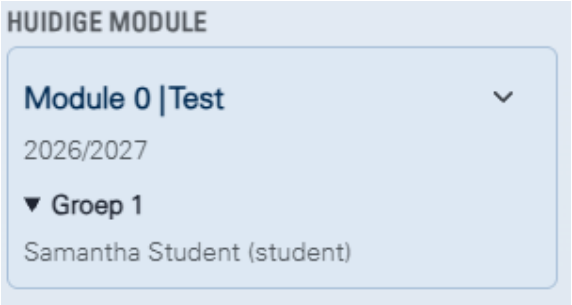


2 Navigatie en Patiëntinformatie

Zodra je bent ingelogd, kom je in de werkruimte van jouw groep. De linkerbalk is je belangrijkste navigatiemiddel en bestaat uit twee kaarten:

De Module-kaart

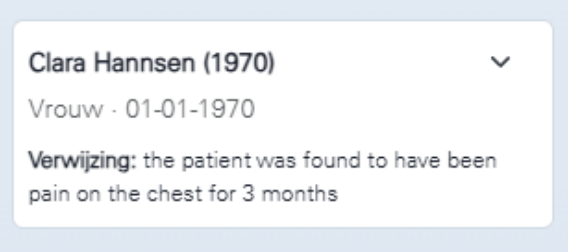
Bovenaan de linkerbalk zie je de **Module-kaart**. Hier zie je in welke module je zit en het bijbehorende academische jaar. Daaronder zie je bij welke groep je hoort en wie de leden van jouw groep zijn. Heb je toegang tot meerdere modules? Dan kun je via het dropdown-pijltje in deze kaart eenvoudig wisselen van module.



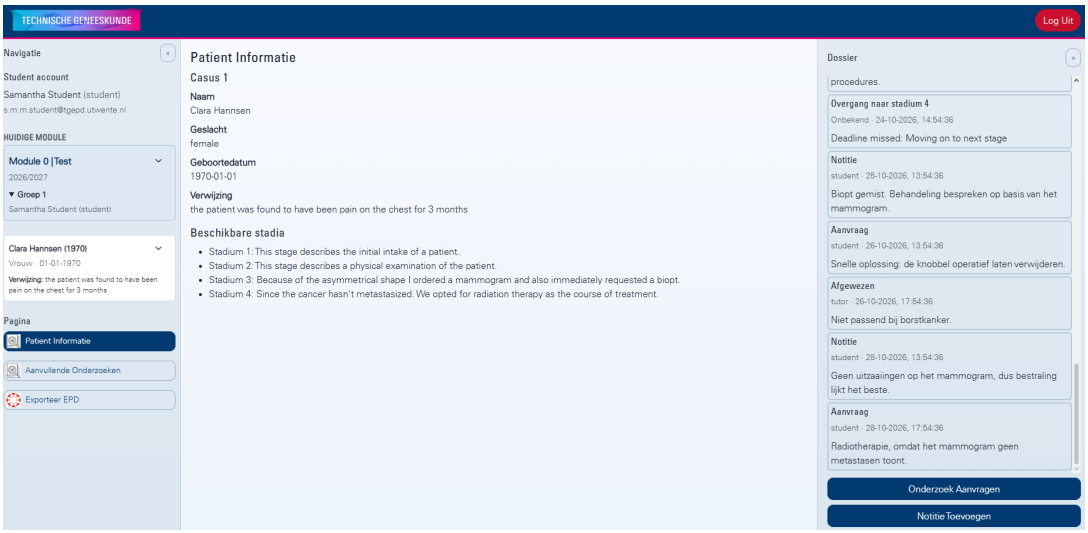
De Patiënt-kaart

Direct onder de Module-kaart vind je de **Patiënt-kaart**. Deze kaart toont de basisgegevens van de huidige patiënt (naam, geslacht, geboortedatum en de verwijzing). Als jullie groep in deze

module met meerdere casussen werkt, kun je via het dropdown-pijltje in deze kaart wisselen tussen de verschillende patiënten.



Onderaan de linkerbalk vind je de pagina's: **Patient Informatie**, **Aanvullende Onderzoeken** en **Exporteer EPD**. Klik op **Patient Informatie** om het volledige dossier van de patiënt te openen. Hier zie je de basisgegevens en de **Beschikbare stadia**.



Het Dossier (Rechterpaneel)

Aan de rechterkant van het scherm zie je het lopende **Dossier**. Dit is een chronologisch overzicht van alle acties die jij (en je groepsgenoten) hebben ondernomen, zoals notities, aangevraagde onderzoeken en de beoordeling daarvan door de tutor.

Let op

Automatische doorgang:
Soms zie je bovenaan het dossier een melding zoals *"Deadline missed: Moving on to next stage"*. Dit betekent dat de einddatum van het huidige stadium verstreken is en het systeem je automatisch naar het volgende stadium heeft doorgestuurd.

3 Notities Toevoegen

Tijdens je onderzoek wil je observaties of tussenconclusies vastleggen in het dossier.

1. Klik op de knop **Notitie Toevoegen** onderaan het dossier.
2. Er opent een pop-upvenster.

3. Typ je notitie in het tekstvak (bijv. je differentiaaldiagnose of observaties).
4. Klik op **Opslaan** om de notitie aan het dossier toe te voegen. Je kunt dit ook annuleren.



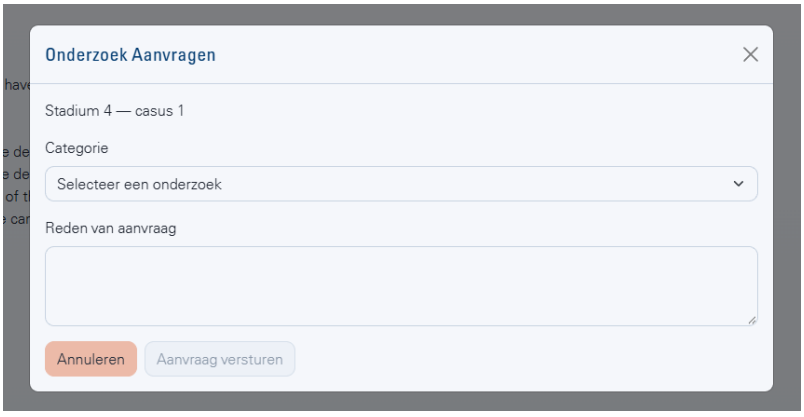
4 Onderzoeken Aanvragen

Als je een specifiek onderzoek wilt uitvoeren (zoals een bloedtest of een scan), moet je dit formeel aanvragen. Dit kan alleen als je in het juiste stadium zit.

1. Klik onderaan het dossier op de knop **Onderzoek Aanvragen**.
2. Selecteer in de pop-up de **Categorie** van het onderzoek dat je wilt aanvragen.
3. Geef een duidelijke **Reden van aanvraag**. Leg uit waarom je dit onderzoek nodig acht op basis van de klachten van de patiënt.
4. Klik op **Aanvraag versturen**.

Extra informatie

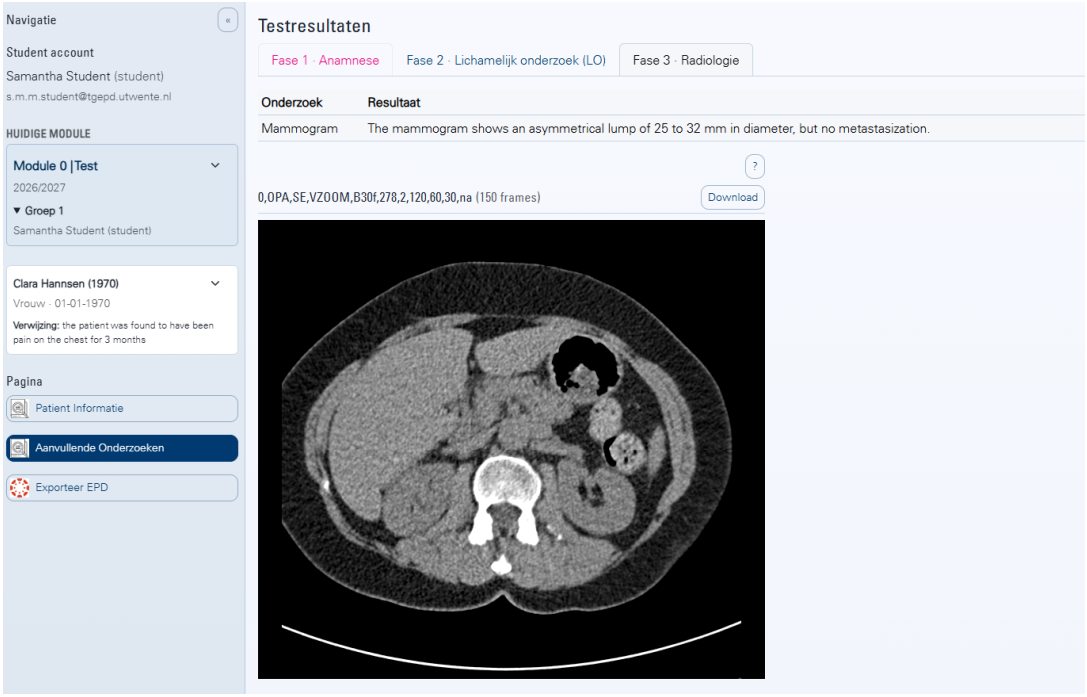
Goedkeuring door de Tutor
Je aanvraag wordt naar je tutor gestuurd. De tutor kan de aanvraag goedkeuren of afwijzen. Als een aanvraag wordt afgewezen, staat de reden van afwijzing in je dossier. Je zult dan een nieuwe, betere aanvraag moeten indienen.



5 Testresultaten Bekijken

Zodra je tutor een onderzoek heeft goedgekeurd, kun je de resultaten inzien.

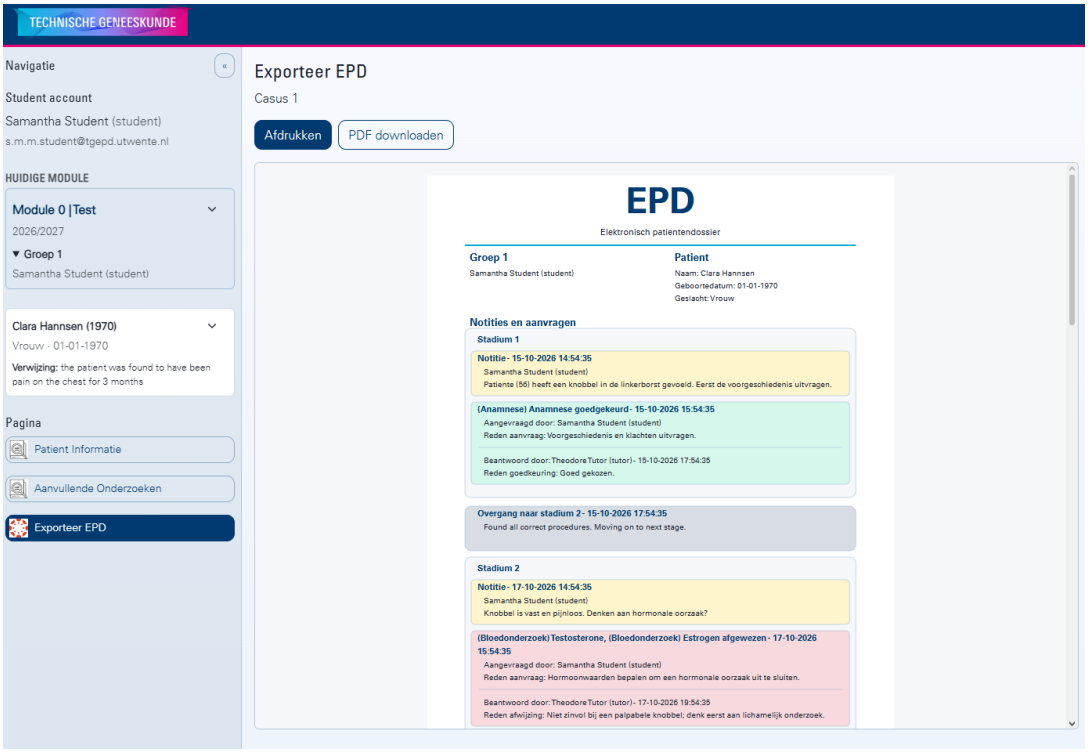
1. Klik op **Aanvullende Onderzoeken** in de linkerbalk.
2. Bovenaan zie je de stadia (bijv. Fase 1 - Anamnese, Fase 3 - Radiologie). Klik op een fase om de resultaten van dat stadium te bekijken.
3. Je ziet een tabel met het uitgevoerde **Onderzoek** en de **Resultaten**.
4. Bij beeldvormend onderzoek (zoals een mammogram of CT-scan) kun je de afbeeldingen bekijken en indien nodig de ruwe data (DICOM) downloaden via de knop **Download**.



6 EPD Exporteren

Aan het einde van de casus, of wanneer je een overzicht wilt hebben van al je bevindingen, kun je het volledige dossier exporteren.

1. Klik op **Exporteer EPD** in de linkerbalk.
2. Je krijgt een voorbeeldweergave van het gehele dossier, inclusief alle notities, aanvragen en goedkeuringen per stadium.
3. Klik op **Afdrukken** om het document direct te printen.
4. Klik op **PDF downloaden** om een digitaal bestand op te slaan voor je eigen administratie of om in te leveren bij je docent.


TECHNISCHE GENEESKUNDE
Navigatie
Student account
Samantha Student (student)
s.m.m.student@tgepd.utwente.nl
HUIDIGE MODULE
Module 0 | Test
2026/2027
▾ Groep 1
Samantha Student (student)
Clara Hannsen (1970)
Vrouw · 01-01-1970
Verwijzing: the patient was found to have been pain on the chest for 3 months
Pagina
Patient Informatie
Aanvullende Onderzoeken
Exporteer EPD
Exporteer EPD
Casus 1
Afdrukken
PDF downloaden
EPD
Elektronisch patientendossier
Groep 1
Samantha Student (student)
Patient
Naam: Clara Hannsen
Geboortedatum: 01-01-1970
Geslacht: Vrouw
Notities en aanvragen
Stadium 1
Notitie - 15-10-2026 14:54:35
Samantha Student (student)
Patiente (56) heeft een knobbel in de linkerborst gevoeld. Eerst de voorgeschiedenis uitvragen.
(Anamnese) Anamnese goedgekeurd - 15-10-2026 15:54:35
Aangevraagd door: Samantha Student (student)
Reden aanvraag: Voorgeschiedenis en klachten uitvragen.
Beantwoord door: TheodoreTutor (tutor) - 15-10-2026 17:54:35
Reden goedkeuring: Goed gekozen.
Overgang naar stadium 2 - 15-10-2026 17:54:35
Found all correct procedures. Moving on to next stage.
Stadium 2
Notitie - 17-10-2026 14:54:35
Samantha Student (student)
Knobbel is vast en pijnloos. Denken aan hormonale oorzaak?
(Bloedonderzoek) Testosterone, (Bloedonderzoek) Estrogen afgewezen - 17-10-2026 15:54:35
Aangevraagd door: Samantha Student (student)
Reden aanvraag: Hormoonwaarden bepalen om een hormonale oorzaak uit te sluiten.
Beantwoord door: TheodoreTutor (tutor) - 17-10-2026 19:54:35
Reden afwijzing: Niet zinvol bij een palpabele knobbel; denk eerst aan lichamelijk onderzoek.

EPD

Tutor handleiding

Inhoudsopgave

1 Inloggen 2

2 Het Beginscherm 2

3 Groep Overzicht 3

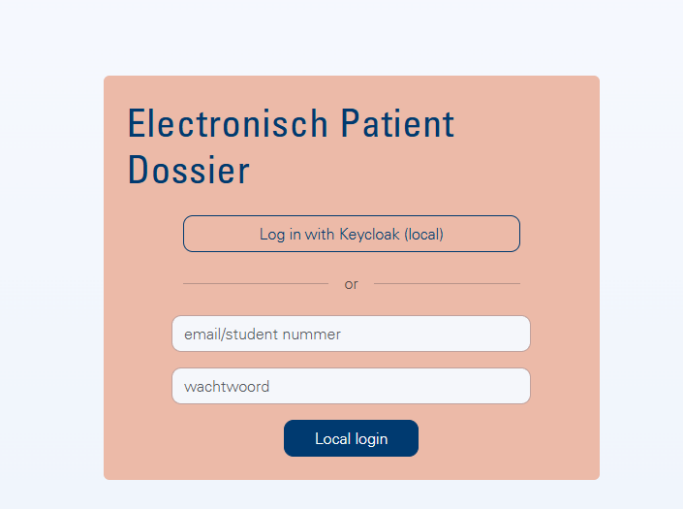
4 Onderzoeksaanvragen Beoordelen 4

5 De Case Pagina 6

1 Inloggen

Wanneer je naar het Elektronisch Patiëntendossier (EPD) gaat, zie je het inlogscherm.

- **Keycloak Login:** Klik op de bovenste knop om veilig in te loggen met je Universiteit Twente-account (Canvas).
- **Local Login:** Voor testdoeleinden kun je de onderste velden gebruiken om in te loggen met een lokaal e-mailadres/studentnummer en wachtwoord.



2 Het Beginscherm

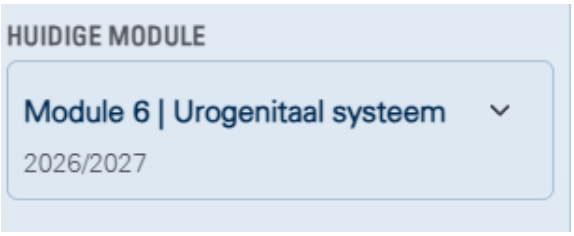
Zodra je bent ingelogd, kom je op je beginscherm. De linkerbalk is je belangrijkste navigatiemiddel, terwijl het rechterpaneel het dossier van de geselecteerde groep toont.



De Module-kaart

Bovenaan de linkerbalk zie je de **Module-kaart**. Hier zie je in welke module je actief bent (bijv. Module 6 | Urogenitaal systeem) en het bijbehorende academische jaar. Daaronder zie je een

dropdown-menu met de groepen waaraan je bent toegewezen. Via dit menu kun je eenvoudig wisselen tussen de verschillende groepen die je begeleidt.



Het Dossier

Aan de rechterkant van het scherm zie je het lopende **Dossier** van de geselecteerde groep. Dit is een chronologisch overzicht van alle acties die de studenten hebben ondernomen. Hier komen alle notities, aangevraagde onderzoeken en jouw beoordelingen samen.

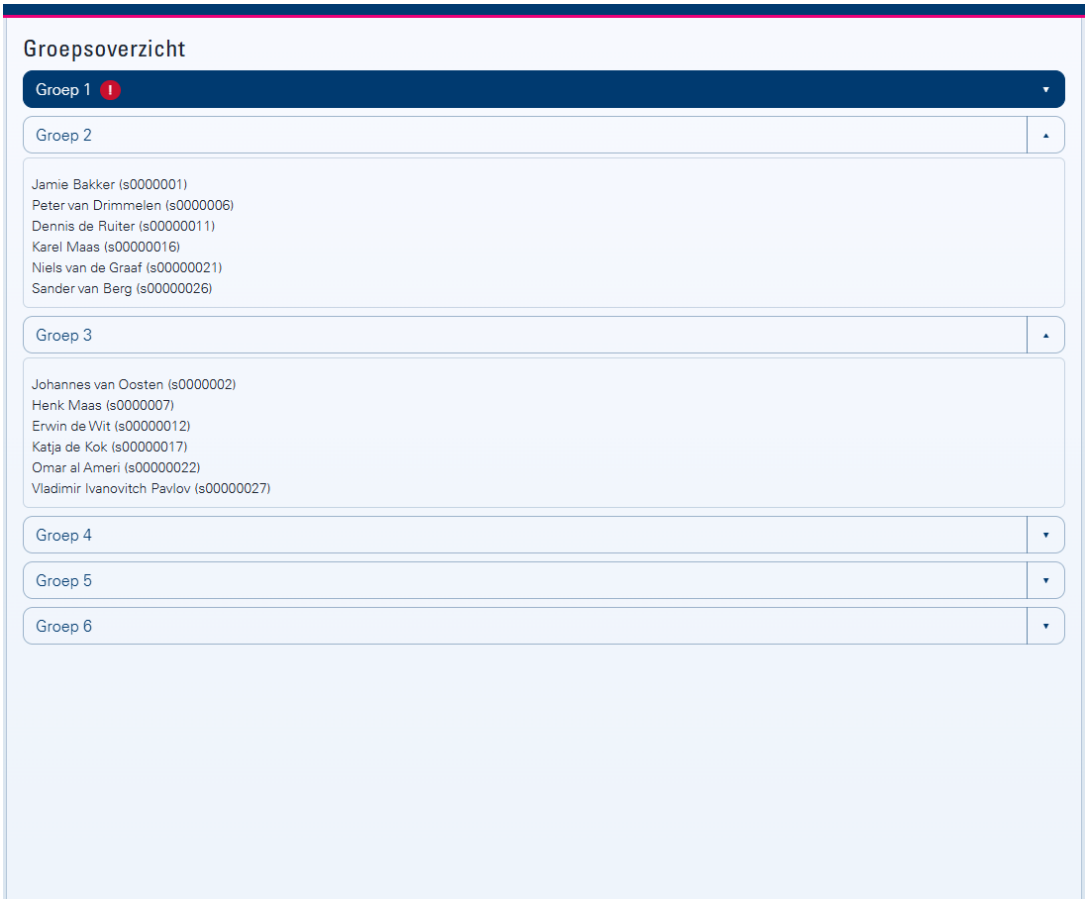
Let op

Automatische doorgang:
Soms zie je bovenaan het dossier een melding zoals *"Deadline missed: Moving on to next stage"*. Dit betekent dat de einddatum van het huidige stadium verstreken is en het systeem de studenten automatisch naar het volgende stadium heeft doorgestuurd.

3 Groep Overzicht

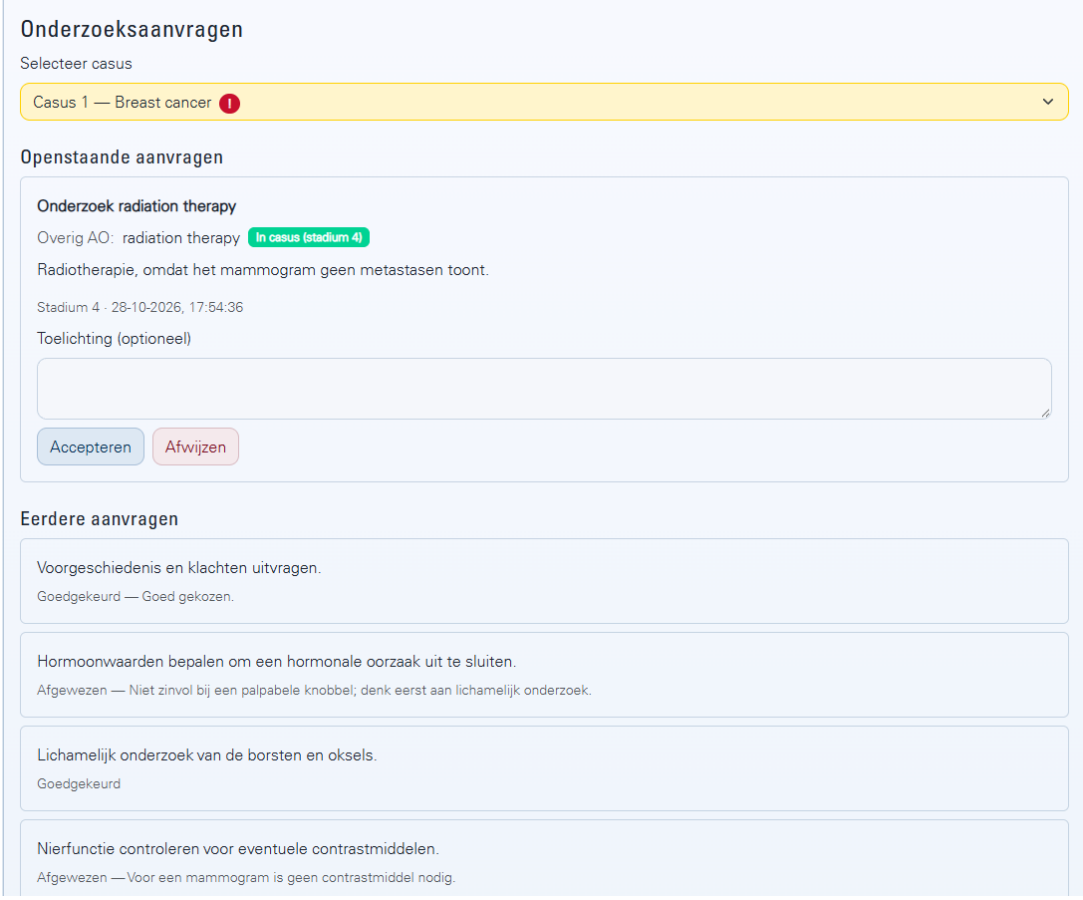
Om een overzicht te krijgen van alle groepen en hun voortgang, kun je gebruikmaken van het Groep Overzicht. Dit scherm bestaat uit een lijst met alle groepen binnen de module.

- **Inhoud van de groep:** Door het dropdown-pijltje naast de groepsnaam te gebruiken, zie je direct welke studenten er in die specifieke groep zitten.
- **Openstaande aanvragen:** Als een groep een **openstaande onderzoeksaanvraag** heeft, wordt deze groep in het overzicht **gemarkeerd met een rode '!'**.
- **Naar de aanvraag:** Klik op de gemarkeerde groep. Je wordt dan direct doorgestuurd naar de specifieke onderzoeksaanvraag in het dossier van die groep.



4 Onderzoeksaanvragen Beoordelen

Zodra je via het Groep Overzicht op een gemarkeerde groep hebt geklikt, kom je op de pagina **Onderzoeksaanvragen**. Op deze pagina beoordeel je de verzoeken van de studenten.



De Case Selecteren

Bovenaan de pagina zie je een dropdown-menu: **Selecteer casus**.

- De casus waarvoor een aanvraag openstaat, is **geel gemarkeerd** en aangegeven met een rood uitroepteken in dit dropdown-menu.

Openstaande Aanvragen

Onder het kopje **Openstaande aanvragen** zie je het verzoek van de student.

- Je ziet welk onderzoek er wordt aangevraagd (bijv. *Onderzoek radiation therapy*) en de bijbehorende **Reden van aanvraag** van de student.
- Er staat een groen label bij, bijvoorbeeld **"In Casus (stadium 4)"**. Dit betekent dat het onderzoek **correct is aangevraagd** in het stadium waar de studenten zich nu bevinden.

> **Let op**
>
> **Verkeerd aangevraagd?**
> Als het onderzoek niet in het juiste stadium is aangevraagd, of als er iets anders mee mis is, zal het groene label er anders uitzien of ontbreken. Je kunt de aanvraag dan direct afwijzen (accepteren is niet mogelijk).

- Je kunt optioneel een **Toelichting** typen in het tekstvak.
- Klik op **Accepteren** (om het onderzoek goed te keuren en de resultaten vrij te geven) of **Afwijzen**.

Eerdere Aanvragen

Onderaan de pagina, onder het kopje **Eerdere aanvragen**, zie je een chronologisch overzicht van alle onderzoeken die deze groep al heeft aangevraagd.

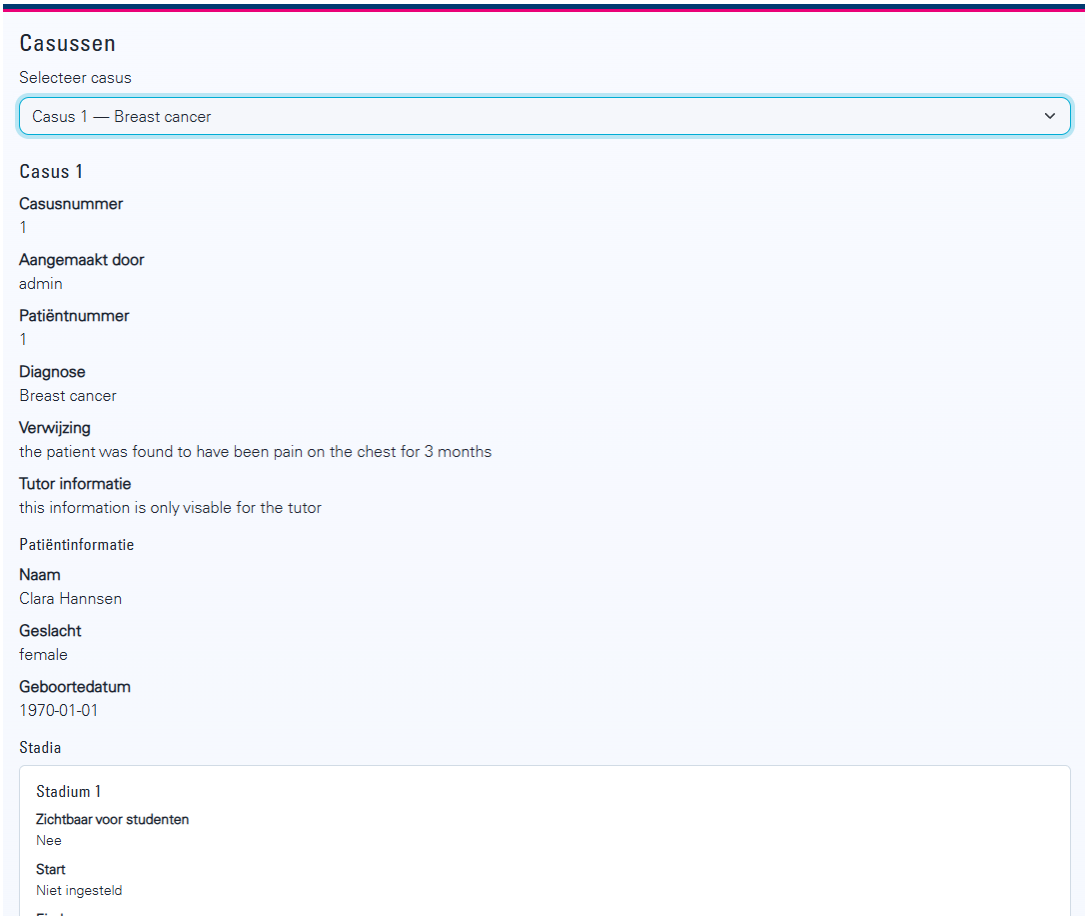
- Je ziet direct de status: **Goedgekeurd** of **Afgewezen**.
- Bij afgewezen aanvragen staat de reden vermeld (bijv. *"Niet zinvol bij een palpabele knobbel; denk eerst aan lichamelijk onderzoek"*). Dit geeft je direct context over de voortgang en het denkniveau van de studenten.

Extra informatie

Reden van afwijzing
Wijs je een aanvraag af? Dan is het verplicht (en zeer aan te raden) om een duidelijke reden van afwijzing op te geven. De student ziet deze reden direct in het dossier en kan vervolgens een nieuwe, betere aanvraag indienen.

5 De Case Pagina

Wanneer je een specifieke groep en casus selecteert, krijg je toegang tot de Case Pagina. Hier kun je de details van de casus bekijken.



Patient Informatie en Dossier

Op de pagina **Patient Informatie** zie je de basisgegevens van de patiënt, de beschikbare stadia en de aanvullende onderzoeken die de studenten moeten gaan aanvragen.

Let op

Tutorinformatie
Let goed op: op deze pagina staat een sectie **Tutorinformatie**. Deze informatie is expliciet bedoeld voor jou als tutor (en dus niet zichtbaar voor studenten) om je te helpen bij de begeleiding.

Aanvullende Onderzoeken

Je alle huidige en toekomstige onderzoeken bekijken die van toepassing zijn van deze casus. bij elke procedure zou je ook de relevant pictures of dicom viewer zien. Dit kan mogelijk behulpzaam zijn om een student beter te begeleiden met waar ze naartoe moeten werken.



EPD

Leraren handleiding

Inhoudsopgave

1 Inloggen 2

2 Modules Beheren 2

3 Casussen Aanmaken en Beheren 5

4 Groep Overzicht en Toewijzingen 8

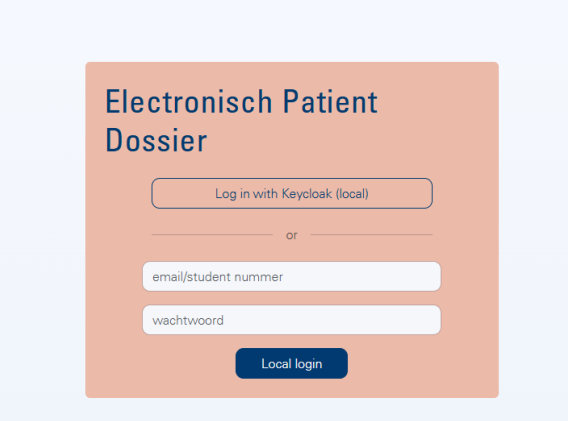
5 Gebruikersbeheer (Rollen) 8

6 De Tutoromgeving Bekijken 9

1 Inloggen

Wanneer je naar het Elektronisch Patiëntendossier (EPD) gaat, zie je het inlogscherm.

- **Keycloak Login:** Klik op de bovenste knop om veilig in te loggen met je Universiteit Twente-account.
- **Local Login:** Voor testdoeleinden of lokale omgevingen kun je de onderste velden gebruiken om in te loggen met een lokaal e-mailadres/studentnummer en wachtwoord.



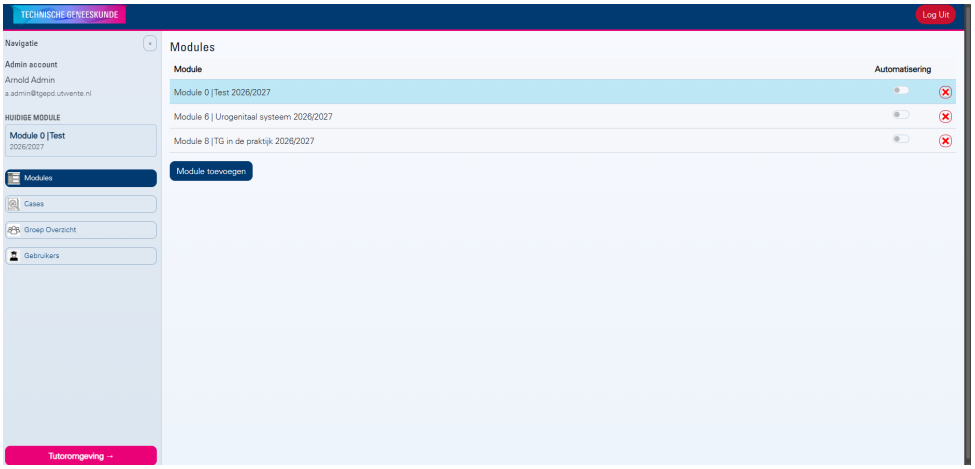
Zodra je bent ingelogd, kom je op het Admin-dashboard. De linkerbalk is je belangrijkste navigatiemiddel en bevat de opties **Modules**, **Cases**, **Groep Overzicht** en **Gebruikers**.

2 Modules Beheren

Modules vertegenwoordigen de overkoepelende vakken of blokken (bijv. *Module 6 | Urogenitaal systeem*).

Modules bekijken of aanpassen

1. Klik op **Modules** in de linkerbalk.
2. Je ziet een lijst van alle bestaande modules, inclusief het academisch jaar (bijv. 2026/2027).
3. Je kunt de schakelaar **Automatisering** aan of uit zetten voor een specifieke module. Staat dit aan? Dan worden de verzoeken van studenten voor extra onderzoeken automatisch verwerkt.
4. Klik op de **rode 'X'** om een module te verwijderen.

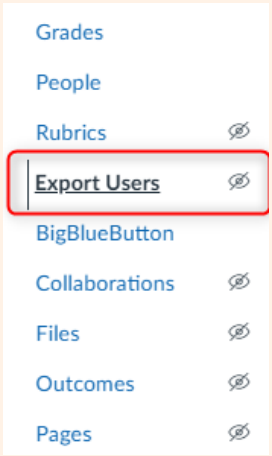


Een nieuwe module aanmaken

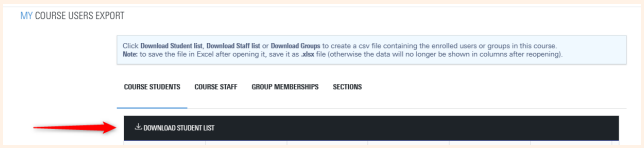
Let op

Het Canvas CSV-bestand exporteren
Om een nieuwe module in het EPD aan te maken, heb je een CSV-bestand met de studentenlijst nodig. Dit bestand exporteer je uit Canvas:

1. Ga in Canvas naar je cursus en klik in het menu op **Export Users**.



2. Kies vervolgens voor de optie **Group Memberships** en klik op **Download Student List**.



3. Upload het gedownloade CSV-bestand vervolgens in de EPD-pop-up wanneer je een nieuwe module aanmaakt.

1. Klik op de blauwe knop **Module toevoegen**.

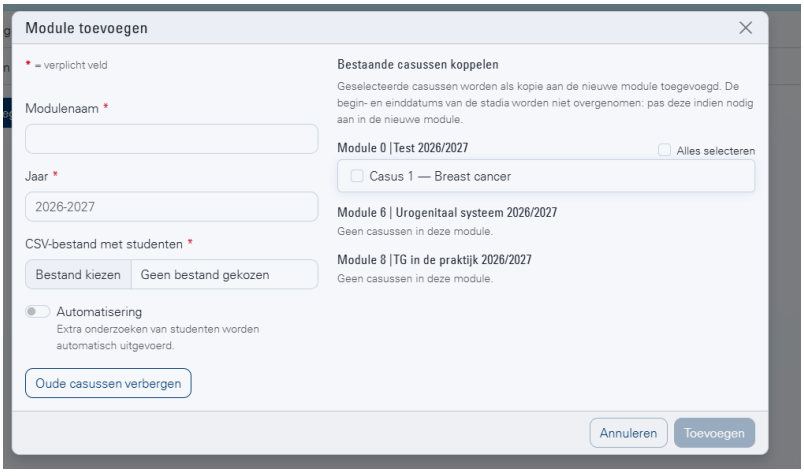
2. Er verschijnt een pop-up. Vul de volgende gegevens in:

   - **Modulenaam:** De naam van de module.
   - **Jaar:** Het academische cohort (bijv. 2026-2027).
   - **CSV-bestand met studenten:** Upload hier de studentengroeplijst die je uit Canvas hebt geëxporteerd.
   - **Automatisering:** Zet deze aan als je wilt dat onderzoeken automatisch worden afgehandeld.



3. **Oude casussen importeren:** Wil je casussen uit een vorige module hergebruiken? Klik dan op **Oude casussen importeren**. Er opent een zijpaneel. Vink de vakjes aan naast de casussen die je wilt kopiëren naar je nieuwe module.

4. Klik op **Toevoegen** om de module op te slaan.



Let op

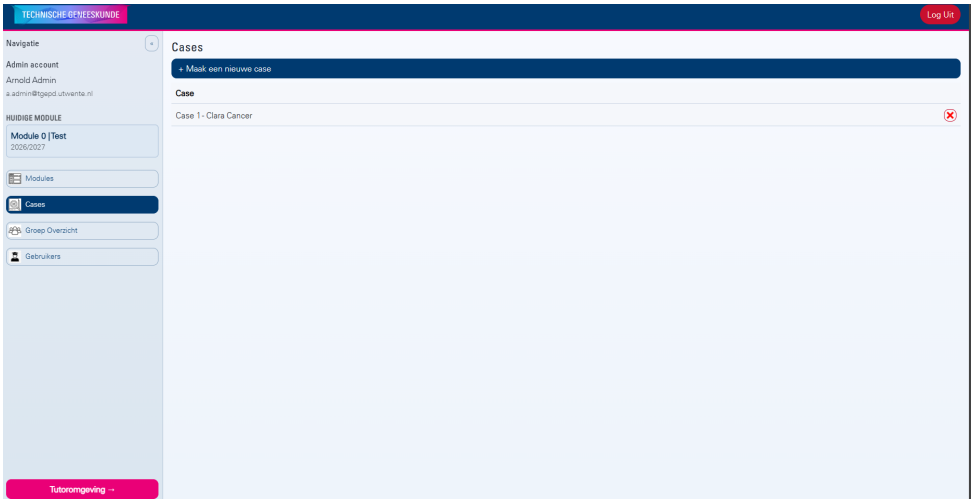
Let op: de begin- en einddatums van de oude stadia worden niet overgenomen wanneer je oude casussen importeert. Deze moet je opnieuw instellen.

3 Casussen Aanmaken en Beheren

Casussen zijn de klinische scenario's (bijv. *Clara Hanssen*) waarmee studenten aan de slag gaan.

Casussen bekijken

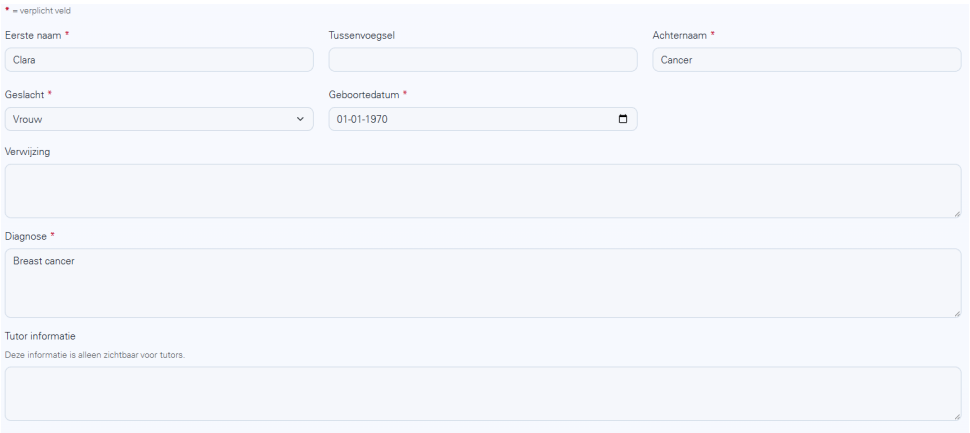
1. Klik op **Cases** in de linkerbalk.
2. Je ziet een lijst met casussen die bij de geselecteerde module horen.
3. Je kan een casus aanpassen door op de casus zelf te klikken.
4. Met de rode X kun je een casus verwijderen.
5. Klik op + **Maak een nieuwe case** om een nieuw scenario te starten.



Het Patiëntprofiel opbouwen

Bij het aanmaken van een nieuwe case vul je eerst de basisgegevens van de patiënt in:

- **Naam**, **Tussenvoegsel** en **Achternaam**.
- **Geslacht** en **Geboortedatum**.
- **Verwijzing:** De verwijzingsinformatie (bijv. vanuit de huisarts).
- **Diagnose:** De onderliggende aandoening (bijv. Borst Kanker).
- **Tutor informatie:** *Dit veld is alleen zichtbaar voor tutoren/docenten, niet voor studenten.* Gebruik dit voor aantekeningen, leerpunten of verborgen context over de casus.



Stadia en Onderzoeken toevoegen

Extra informatie

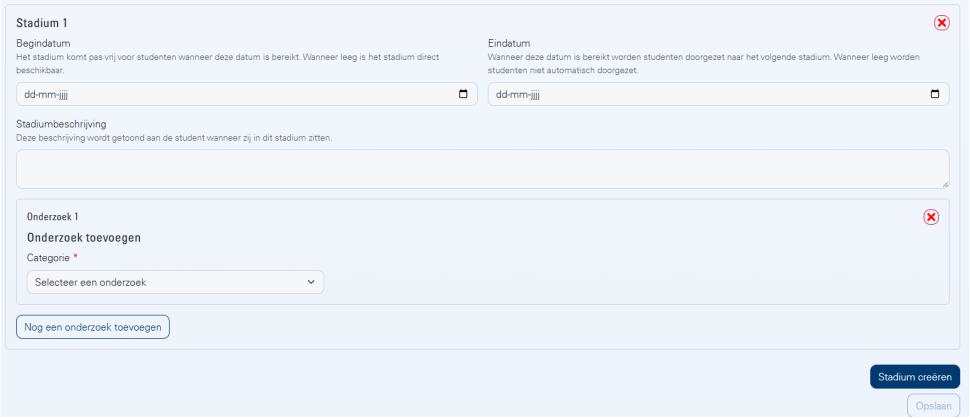
Wat is een stadium?
Een patiëntcasus in het EPD is opgebouwd uit meerdere **stadia** (fases). Elk stadium representeert een stap in het klinisch redeneren en het diagnostisch proces van de student. Je kunt een stadium zien als een "hoofdstukïn het patiëntdossier. Studenten beginnen altijd in Stadium 1 en kunnen pas door naar het volgende stadium wanneer ze de vereiste stappen hebben doorlopen (of wanneer de deadline verstreken is).
Binnen een stadium bepaal je als docent:

- **Wat de student te zien krijgt:** Een korte inleidende tekst die extra informatie kan geven aan de student.
- **Welke onderzoeken gedaan mogen worden:** Je koppelt specifieke onderzoeken (zoals een CT-scan of een bloedwaarde) aan dat specifieke stadium.
- **Wanneer het stadium afgerond is:** Door middel van datums (start- en einddatum) of doordat de student de juiste onderzoeken heeft aangevraagd.

Door casussen op te delen in stadia, dwing je een gestructureerde klinische workflow af. Een student kan bijvoorbeeld pas een behandeling voorstellen in Stadium 3, nadat de uitslagen van de diagnostische onderzoeken uit Stadium 1 en 2 bekend zijn.

Zodra het patiëntprofiel staat, kun je de fases van de casus opbouwen:

1. **Stadium 1** wordt standaard aangemaakt.
2. **Begindatum / Einddatum:** Stel in wanneer het stadium opengaat voor studenten en wanneer ze automatisch naar het volgende stadium worden doorgestuurd. Laat je dit leeg? Dan worden studenten automatisch doorgestuurd wanneer ze alle onderzoeken van dit stadium gecompleteerd hebben.
3. **Stadiumbeschrijving:** Deze informatie wordt aan de student getoond zodra ze dit stadium betreden.





4. **Onderzoek toevoegen:**
   - Selecteer de **Categorie** (bijv. Radiologie, Bloedonderzoek).
   - Selecteer het specifieke **Onderzoek**.
   - Voeg een **Omschrijving/Resultaten** toe, zodat de student weet wat de uitkomst is.
   - Upload eventuele relevante **Afbeeldingen of DICOM-bestanden** direct hier.

Let op

Categorieën en subcategorieën: Sommige categorieën (zoals Anamnese) hebben **geen subcategorieën**. Bij deze categorieën verschijnt er een extra tekstveld. Dit stelt jou als docent in staat om handmatig de exacte naam of details van het onderzoek te typen. Dit maakt het overzichtelijker en flexibeler tijdens het bewerken van de casus.
Let op voor labwerk (Bloedonderzoek, Urineonderzoek en Bloedgasanalyse): Selecteer je een van deze drie hoofdcategorieën? Dan zie je een uitklapbare lijst met subcategorieën (zoals Hematologie of Ontsteking). Gebruik de zoekvelden om de specifieke waarden te selecteren die je aan de student wilt tonen. Bestanduploads zijn uitgeschakeld voor deze gegroepeerde categorieën; je uploadt bestanden alleen bij specifieke onderzoeken zoals Radiologie of bij overige losse onderzoeken.

5. Klik op **Nog een onderzoek toevoegen** om meerdere onderzoeken aan één stadium toe te voegen.
6. Klik op **Stadium creëren** om het stadium op te slaan en een nieuw stadium toe te voegen.
7. Je kan stadia en onderzoeken weer verwijderen met de rode X in het formulier.
8. Wanneer je klaar bent met het creëren van de casus, kun je deze opslaan met de blauwe opslaan knop (items met een * zijn verplicht om in te vullen).

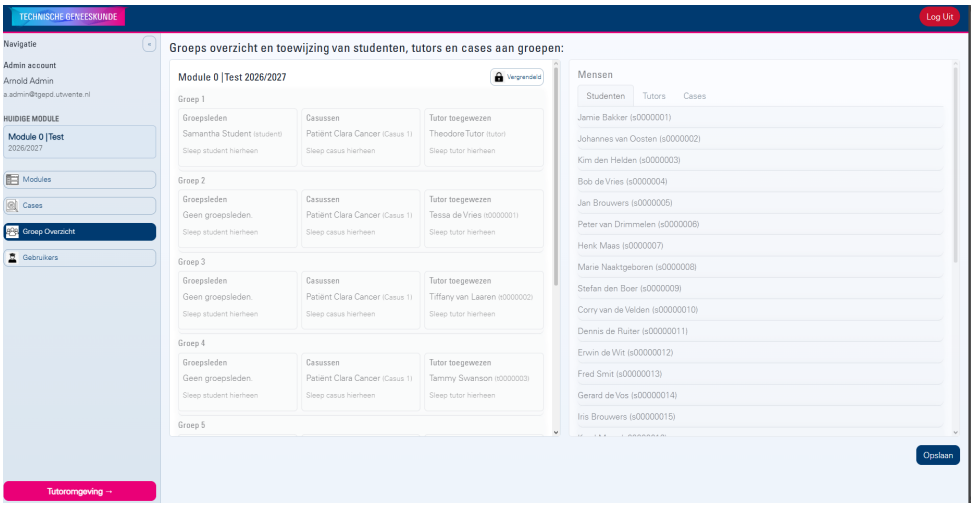
Extra informatie

Case opslaan als kopie
Wanneer je een bestaande case aan het aanpassen bent, heb je de optie om deze op te slaan als een **kopie**. Dit is enorm handig als je een bestaande casus wilt hergebruiken voor een nieuw cohort of een variant wilt maken, zonder dat je alle patiëntgegevens en stadia helemaal opnieuw hoeft in te voeren. Het systeem maakt dan een volledige duplicatie van de case aan, die je vervolgens naar wens kunt aanpassen.

4 Groep Overzicht en Toewijzingen

Op deze pagina koppel je studenten, tutoren en casussen aan elkaar.

1. Klik op **Groep Overzicht** in de linkerbalk.
2. Het scherm is verdeeld in twee panelen: **Groepen** aan de linkerkant en **Mensen** aan de rechterkant.
3. **Groepen maken:** Binnen elk moduleblok kun je mensen van het rechterpaneel naar de groepen aan de linkerkant slepen (drag & drop).
   - Sleep studenten naar het vak **Groepsleden**.
   - Sleep casussen naar het vak **Casussen**.
   - Sleep tutoren naar het vak **Tutor toegewezen**.
4. Maak je een fout? Sleep ze terug naar het rechterpaneel.
5. Klik op **Opslaan** (rechtsonder in het scherm) als je klaar bent.



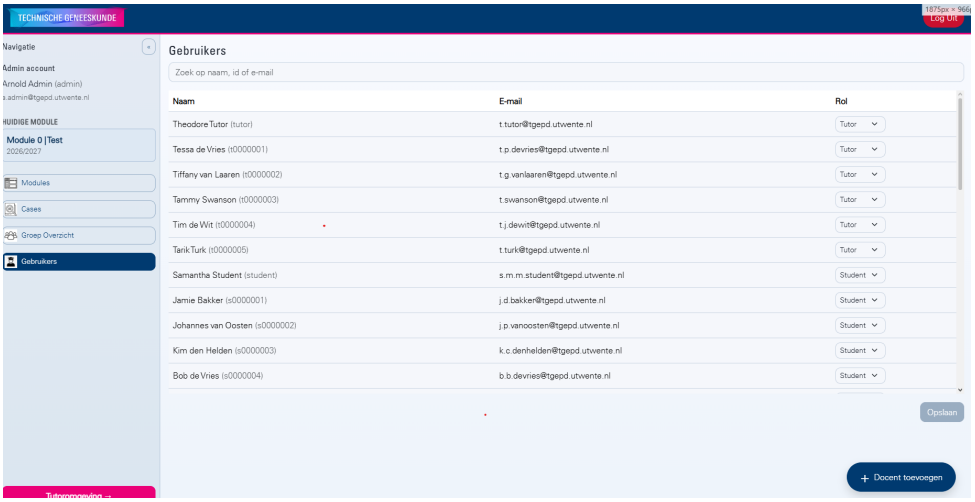
5 Gebruikersbeheer (Rollen)

Op de Gebruikers-pagina kun je alle gebruikers inzien, zoeken en de rollen aanpassen van de mensen die via Canvas zijn geïmporteerd.

1. Klik op **Gebruikers** in de linkerbalk.
2. Gebruik de zoekbalk bovenaan om te zoeken op naam, ID of e-mailadres.

3. Kijk naar de kolom **Rol**. Geïmporteerde gebruikers staan standaard op “Student”.
4. Moet iemand tutor of admin zijn? Klik dan op het dropdown-menu in de kolom **Rol** en verander de rol naar **Tutor** (of **Admin**).
5. Klik rechtsonder op **Opslaan** om de wijzigingen door te voeren.
6. *Let op:* Verschijnt er een rood stipje naast een naam? Dan zijn er nog niet-opgeslagen wijzigingen voor die gebruiker.

Om handmatig een nieuwe tutor aan het systeem toe te voegen (die niet via Canvas is geïmporteerd), klik je op de donkerblauwe knop + **Docent toevoegen** rechtsonder in het scherm.



6 De Tutoromgeving Bekijken

Wil je zelf als tutor te werk gaan, of zien wat tutoren zien (bijv. om verzoeken van studenten goed of af te keuren, of om de verborgen tutorinformatie te lezen)? Je hebt hiervoor geen apart account nodig.

- Klik op de felroze knop **Tutoromgeving** → helemaal onderaan de linkerbalk. Je schakelt hiermee je eigen view om naar het perspectief van de tutor.