

Design Report
FMT: ETAPS Conference
Management System v1.1

Luna Peshkov, s2782200 Manya Narkar, s2297809
Harm Dreteler, s2337568 Egor Krasnoperov, s2384361
Stefan Simionescu, s2391821
Robert-Jan Nijhuis, s2621517
Supervised by: Dr. Marieke Huisman

19th April, 2024

Contents

1	Introduction	4
1.1	ETAPS and committees	4
1.2	Existing program and motivation for improvement	4
1.3	Existing functionalities and development approach	4
1.4	User interfaces	5
1.5	Enhancements and contributions	5
2	Responsibilities	7
2.1	The team	7
2.2	Roles	7
2.2.1	Overall developer	7
2.2.2	Senior developer	8
2.2.3	Project management	8
2.2.4	External communications	9
2.2.5	Cross platform engineer	9
2.2.6	Code reviewer	9
3	Process	10
3.1	Development Methodology	10
3.2	Client Meetings	10
3.3	Technologies Used	11
3.4	Planning	12
4	Requirement Specification	13
4.1	Stakeholders	13
4.2	Requirement prioritization	13
4.3	User Stories	15
5	Updates	17
6	Design	23
6.1	Admin Interface	23
6.1.1	Conference Creation	24
6.1.2	Selection Committee	24
6.1.3	Account Management	24
6.2	Non-Admin Interface	25
6.2.1	Sessions	25
6.2.2	Committee Selection	26
6.2.3	Overview	26

7 Risk Analysis	27
7.1 Risks within Team	27
7.2 Risks within Project	28
8 Testing	29
8.1 Manual testing	29
8.2 Unit testing	29
9 Conclusion	32
10 Future Work	33
11 APPENDIX	35
A	
GANTT Chart for Planning	35
B	
Sorting and Filtering	36
C	
Duplicate detection	37

1 Introduction

1.1 ETAPS and committees

The European joint conferences on Theory And Practice of Software (ETAPS) [1] is a non-profit association that addresses the dissemination of information relating to the field of Computer Science. It is a confederation of four main committees: ESOP, FASE, FoSSaCS and TACAS, which cover topics related to software systems, such as compiler advancements, formal approaches to software engineering and security, among others. ETAPS conducts a main conference program that takes place annually, during which researchers submit and present their research in various sessions, as well as get an opportunity to interact with other researchers.

1.2 Existing program and motivation for improvement

A crucial part of this conference program is the organization of it. It is important for the chairs of each committee to be able to schedule these sessions, and for the steering chair to oversee and finalize the same whilst ensuring there are no conflicts. To facilitate various management aspects of this conference program, a conference management system was developed. Our project builds upon an existing web application - originally developed by a previous group - to introduce significant enhancements and address critical areas of improvement. The existing conference support system has laid an excellent foundation for processing and planning various aspects of the digital conferences. However, through a thorough analysis, there are several identified areas of improvement. This design report provides a complete documentation of the enhancements and changes introduced into the ETAPS convention management software. It emphasizes the underlying motivations behind the changes, insights into methodologies employed in the implementation, and gives a clear picture of expected benefits and effect on the stakeholders.

1.3 Existing functionalities and development approach

The existing conference support system has made numerous activities possible for the user from 2 parts of the system: the conference scheduling system and the committee seat system. For the conference Scheduling system the already existing functionalities are: creating a schedule, adding a conference program, exporting program in YAML [2]; which can be used by the ETAPS website to display the conference programs. For the committee seat system the existing functionalities are : adding a new committee, accepting com-

mittee members, confirming participants. Moreover, the project group went through extensive consultations with the client and created user stories that acted as a blueprint for system enhancement. These user stories laid down the requirements and the expectations of stakeholders to guide the development process. Next, tasks were meticulously crafted and distributed among group members, every task being labeled with the categories of importance, with "must have," "could have," "should have," and "bugs". The most critically needed tasks were organized with a priority tag according to how much value they added to the functionality and usability of the system.

1.4 User interfaces

Furthermore, it is necessary to consider that the conference management system is used by two distinct user interfaces: the administrative side and the non-administrative side. Each side has a different purpose for it, focusing on the roles and needs of the users. The administrative side enables authorized personnel to perform overall management and coordination of the conference proceedings effectively and efficiently. Besides the non-admin functionalities, the admin side can export a program in YAML format, can manage user accounts, edit conference committees, nominate best papers. The other side, the non-administrative side, is crafted for providing participants and contributors with a simple and easy-to-use interface to take part in the conference schedule, see committees, as well as editing committees and its members, upload papers in conferences, adding presentations.

1.5 Enhancements and contributions

The wide range of enhancements that has been put in place covers functionalities and features, which are being optimized for general user experience and operational efficiency within conference management systems. This comprises functionalities like enhancing the user interface and adding advanced features such as CSV imports/exports and automation processes; hence, it becomes an apparatus set to minimize manual intervention, streamline workflow, and facilitate collaboration in a collaborative and effective environment. Our contributions to the ETAPS conference control system are featured in this report, and they include improving the entire system in a number of ways:

- **Package Updates:** Since the system was developed last year, there was quite a list of package updates to be made. This is mostly a maintenance task to prevent issues with running out of date packages.

- User Interface Improvements: Improving accessibility and aesthetics to make the interface more user-friendly in response to stakeholder complaints on usability.
- Feature Enhancement: Add features such as CSV import/export, first-class paper choice, and swapping of time tables to complement the current program.
- Automation: Automate a number of routine tasks that free up the user from routine work and help in streamlining the administrative tasks.
- Optimization: Seek out and remedy bugs and step up the performance for a more smooth operation of the program.

2 Responsibilities

2.1 The team

The process of building this system was driven by a dedicated team of students from the University of Twente. Together, we were able to shed light on differing perspectives and collaborate in order to successfully create and deliver a product aligning with our clients expectations and requirements. A notable mention to Stefan Simionescu who due to personal circumstance had to participate remotely.

The team is comprised of: Luna Peshkov, Manya Narkar, Harm Dreteler, Egor Krasnoperov, Stefan Simionescu and Robert-Jan Nijhuis.

2.2 Roles

Name	Roles
Luna Peshkov	Senior developer, Code reviewer
Stefan Simionescu	Overall developer
Manya Narkar	Overall developer, External communication
Harm Dreteler	
Egor Krasnoperov	Overall developer, Project management, Cross-platform engineer
Rober-Jan Nijhuis	Overall developer, Project management

Table 1: Roles and responsibilities of members

2.2.1 Overall developer

An **overall developer** oversees the entire development process and implements the tasks that they were assigned. A more detailed explanation of their responsibilities:

- **Task Creation:** At the initial stage, all overall developers take part in the stakeholder meetings to extract the user requirements. They should understand these requirements fully to be able to create relevant, brief tasks that will later be implemented.
- **Code Quality Assurance:** Ensures that the code quality has been maintained to an expected level throughout the implementation process. This includes code review and maintenance of "clean code." (reference to Clean Code uncle bob)
- **Optimization of Performance:** They manage the performance optimization of both frontend and backend by optimizing code to ensure faster response times, and making sure that the implemented code is effectively using resources.
- **Integration and Review:** The overall developer is responsible for reviewing the pull requests of other team members to ensure that the code on the main branch is up to standards and that no bugs are introduced by mistake.

2.2.2 Senior developer

A senior developer has the same responsibilities as an overall developer (2.2.1), but has more experience and therefore can take on more difficult features and implement features faster.

2.2.3 Project management

Out of the 6 members, we have assigned 2 of them for **project management**, which entails planning the meetings and supervising team progress. Their responsibilities are:

- **Planning and Scheduling:** The project managers are in charge of making a coherent plan in order to ensure the delivery of the assignments and the availability of members in the arranged meetings.
- **Scrum Masters:** These teammates are responsible for conducting and guiding the stand up meetings. They help monitor the teams goals and guide teammates into working effectively in the case of obstacles.

2.2.4 External communications

Our team has appointed one member for external communications. They were responsible for ensuring clear communication and addressing concerns with our clients through the project. Their responsibilities include:

- **Point of Contact:** They are the primary point of contact for clients, professors and supervisors and reach out to them on behalf of the group for communication.
- **Distributing Information:** They are responsible for receiving and distributing information received from clients to the group as and where required.

2.2.5 Cross platform engineer

Since we had team-members working on both Linux and Windows, a member using both operating systems was assigned to address any inconsistencies or issues.

2.2.6 Code reviewer

A code reviewer reviews all the pull requests on GitHub and gives helpful feedback for code authors to improve their code. They also approve and merge the pull requests once they are in a satisfactory state.

3 Process

This section will detail the process of development throughout the project. This section will further include: communication methods, software development method, and tools employed to make the above feasible.

3.1 Development Methodology

For this project, the team decided to deploy an Agile methodology [3] of development. Since the goal of the project was to enhance an existing system, discovering varying existing bugs and evolving requirements was but natural. Hence, this method of development was chosen due to the dynamic nature of the project requirements. The approach revolves around incremental and iterative steps leading to the completion of projects. These steps are organised into development cycles known as sprints. Stand-up meetings to share progress - known as scrum meetings - were also employed. During the first two weeks of development, each sprint was 7 days long, and each scrum meeting took place daily. Eventually, as the system was set up successfully and each team member was more familiarised with their task(s), the scrum meetings took place twice a week - on Mondays and Thursdays. Following this, the team would proceed to work together in order to resolve any potential obstacles that anyone encountered. A popular framework used to implement Agile software development is the Kanban methodology. This has more to do with the project management aspect of the development and requires communication and transparency of work. We used Trello to recreate a Kanban template for the planning and organization of our tasks.

3.2 Client Meetings

Involving our clients in this development process and incorporating their feedback to improve the system is an important feature of agile development. For the project, the team maintained close collaboration with the stakeholders - meeting once almost every week: on Fridays at 13.00. This ensured our alignment with their expectations, and their awareness of our progress. At the near end of every sprint, a meeting with the clients was held where we not only revised and reconfirmed requirements but also showcased updates we had. The client further had the opportunity to accept or warrant changes from certain features based on their requirements.

3.3 Technologies Used

- MongoDB [4]: This NoSQL document-based database: MongoDB was used for the project. We decided to continue with this due to the positive experience of the previous team with the technology.
- React [5]: For our frontend development, we utilised the JavaScript library: React, to make the web interfaces. The component-based architecture allowed us to reuse various components, promoting code reusability.
- FastAPI [6]: FastAPI is an open-source Python framework for web development and the technology we used for our backend. The API endpoints, routes and models, helped facilitate communication between the frontend and backend. It also provided support for integrating Create, Read, Update and Delete (CRUD) operations to database entries.
- Docker [7]: This is an open-source platform that aids the creation, deployment and management of virtualized application containers. We opted to use Docker to enable a consistent setup of the development environment across every team members' system. This ensured every team member had the necessary configurations and dependencies to reproducibly run the system.
- GitHub [8]: Version control is crucial for facilitating coordination, sharing and collaboration among a software development team. One popular DevOps tool that further aids in source code management is Git. Although, we had originally intended on using GitLab - we switched to GitHub to ensure repository access for our clients. Team members worked on their own branches for varying tasks, and then created pull requests before merging their branch with main. This allowed for a peer review of said branch before the merge.
- Trello [9]: To ensure proper task management, our team used the Trello collaboration tool. Organised into varying boards, this tool allows for separating, explaining and assigning different tasks in a group setting. The requirements for the project were divided based on the MoSCoW model, and the trello lists were created based on the same - where every list represented a priority level in the model, and entries on the board represented project requirements. We further assigned tasks to members and had additional lists to track what tasks were under 'Doing' and 'Done'.

- Discord [10]: Effective communication within the team was crucial to success. We used Discord chats and voice calls as our primary communication platform. This served as a centralized platform for updates, questions, discussions and fostering a pleasant team atmosphere. All stand-ups were conducted via discord, as well as the sessions on Mondays and Thursdays when we were required to work together. In addition to online communication, the team also met physically once every week: on Tuesdays. These meetings allowed for efficient brain-storming and collaborative sessions. We also used discord events as a reminder of upcoming client meetings and such.

3.4 Planning

The duration of this project had been set to 9 weeks. After the initial system set-up and requirement elicitation, sessions were divided into sprints - each of 7 days - to work on implementing certain requirements. Due to the iterative nature of this development methodology, each sprint involved reviewing/listing (new) requirements, developing and then testing them. Once every week, we had a meeting with the client to gain feedback on the process. Alongside this, our schedule also accounted for documents required for the module, such as the ethics report, and other deadlines. A visual depiction of our planning can be found in the form of a GANTT chart in the appendix of the report.

The time had been split up into 5 Sprints. It also consists of time taken for complementary assignments such as the Ethics Report, Project Proposal, Test plan, and feedback for the same. Additionally, weekly once it can be seen that a meeting with our clients has been scheduled as well.

4 Requirement Specification

4.1 Stakeholders

- ETAPS system admin
- Committee chair

Both of these stakeholders correlate with roles that are given to people using the system, each being granted a different level of access. The ETAPS system admin has extra permissions and all the permissions of the committee chair. As such, they also inherit all the user stories of the committee chair.

4.2 Requirement prioritization

The delivery of a successful project relies on the ability to encapsulate the clients requirements well. Requirement elicitation is the first step in this. It includes systematically acquiring, analysing and understanding the requirements in order to produce a system that aligns with these needs. Once our requirements had been elicited, we deemed it important to prioritise them to ensure optimal project management and resource allocation. For this, we used the MoSCoW method [11]. This divides the requirements in four distinct categories: Must have, Should have, Could have and Won't have. The requirements in Must have are expected to be finished during this project and the main focus, and the lower categories only become prioritized when Must requirements are sure to be completed. On the other hand, items in the Won't have category are not expected to be reached within this project. We further confirmed the prioritisation of these requirements with our clients to ensure the development foundations were closely aligned with their expectations.

Must have:

- Allow archiving of earlier years and their versions.
- Warning on duplicates such as: Allow speakers submissions to not have duplicate papers.
- Allow participants to see every (unpublished) schedule
- Allow participants to leave conferences without affecting the rest of the system adversely.
- Filtering of talks, where it is possible to filter on multiple attributes and extended latin characters.

- Delete floating papers after session deletion.
- Allow sessions that are scheduled in parallel to give a warning when an author is at the same time slot more than once.

Should have:

- Allowance for track chair and program committee to edit schedules.
- Post updates to schedule on the designated page/in the designated section.
- Make warnings such as those for ‘duplicates’ sufficiently descriptive, informing users about what they are, which committee they’re in, etc.
- Double name warning does not currently work entirely.
- Minimized hardcoded features.
- Improved user interface
- Allow committees to see the schedules of other committees without having to go back to the previous page.
- Allow nomination of best paper.
- Allow adding sessions to a conference.

Could have:

- An automated deployment pipeline to make further changes easier
- Update schedule automatically once changes in the schedule are made.
- Coloring sessions and adding color to the database and to export files.
- Frontend improvements
- Security improvements
- Collect front end improvements
- Dark mode
- Updating packages
- Allow paper export via Excel

- Make papers accessible to users through means other than just through the database, eg., list all the papers somewhere.
- The application should detect and handle database failure gracefully.

Won't have:

- Unnecessary complexity
- Cloud database only

4.3 User Stories

As a team, we decided to put users at the center of the development process, hence adopting a user-centric approach to the development process. To implement this, we further adopted user stories so that the requirements could be encapsulated from the perspective of an end-user. This helped further our understanding of the system and requirements. The user stories identified are as follows.

As an ETAPS system admin, I want to:

- Archive earlier years and versions of the conference.
- The database to be as dynamic as possible and have a minimum of hardcoded assertions.
- Have an automated deployment pipeline to make further changes easier.
- Have the schedule update automatically once changes are made.
- Have coloured sessions, and save these colours in the database and export files.
- Have the security of the program be improved
- Have an application that detects and handles database failure gracefully.

As a committee chair, I want to:

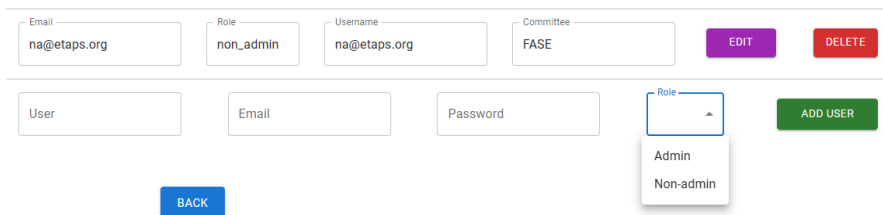
- Nominate a paper for best paper.
- Add a session to a conference.

- Receive a warning if a paper is a duplicate of another paper in the system
- Automatically delete the paper that is tied to a session that I delete.
- Edit the conference schedule.
- Post updates to schedule on the designated page/in the designated section.
- Receive informative warnings about duplication issues, informing me about what they are, which committee they're in, etc.
- Receive a warning if a name I am trying to add is already in the system.
- Have convenient access to schedules of other committees, without having to look through previous pages.
- See the selections of other committees.
- Import and export the program with csv or yaml.
- Receive a warning before I permanently delete something.
- See every schedule, published and unpublished
- Leave the conference without disrupting the system
- Search and filter based on multiple attributes and extended latin characters
- See a warning if a speaker is scheduled for more than once for a single time slot.
- Have an intuitive and functional user interface.
- Switch to dark mode.
- Have access to all the papers without going through the database.
- See what committee something belongs to in the general overview

5 Updates

Version 1.1 of the ETAPS Conference Management System represents a major step in responsiveness to client feedback and improving the overall user experience of the system. As a team, we were able to complete all the requirements from the "Must Have" and "Should Have" sections of requirements from the MoSCoW model in section 4.2. This section will highlight some key updates of this version such as additional features and fixed bugs.

1. Archived Years: Originally, immediately upon logging in, the admin and non-admin users would be required to manually select the current - or any other year of their choice, for the details such as schedules and committees corresponding to those years to be loaded. In the updated version, the page loads on the current year by default. The choice to select other years is still retained.
2. Account Management: An admin user can now add a committee when creating a non-admin users account. This is the committee said non-admin user belongs to and can only be made via an admin account. This page is now also made accessible from the navigation bar.



The screenshot shows a web form for adding a committee. At the top, there are four input fields: 'Email' (na@etaps.org), 'Role' (non_admin), 'Username' (na@etaps.org), and 'Committee' (FASE). To the right of these fields are two buttons: 'EDIT' (purple) and 'DELETE' (red). Below these fields are three more input fields: 'User', 'Email', and 'Password'. To the right of these is a dropdown menu for 'Role' with a blue border, showing 'Admin' and 'Non-admin' options. To the right of the dropdown is a green 'ADD USER' button. At the bottom left, there is a blue 'BACK' button.

Figure 1: . *Account Management Committee Adding.*

3. Session Titles and Descriptions: The session titles and tutorial descriptions are now visible on the main schedule page - enabling an admin user to understand the topic of the sessions and tutorials without having to click on them for details.

16:30					
16:45					
17:00	Conference TACAS - 3 slots	Miscellaneous session SVCComp	Miscellaneous session Rust	Tutorial Tamar Sharon	EDIT
17:15					DELETE
17:30					
17:45					
18:00					
18:15	Testing	Plenary session			EDIT
18:30					DELETE
18:45					
19:00					

Figure 2: . Descriptions and titles of varying sessions visible.

4. Metadata: This iteration of the system has accounted for more metadata capabilities. For example, the admin is now able to add a chair, title and the room it will be conducted in when scheduling a session. This change is also reflected in the YAML file upon generation. A committee chair is also able add a chair and a room for a presentation in their own sessions.

Adding an activity

Start time Duration (in minutes)

Type of activity

Title

Chair
 [ADD ACTIVITY IN PARRALELL](#)

Description

Room

Figure 3: . Admin can edit new session details.

5. Sorting: Lists on different pages can now be sorted based on different

attributes such as: start time, end time, date and the paper titles, speakers, committees, and topics in alphabetical order.

Conference	Date (dd-mm-yyyy)	Topic	Start...	End Time	Title of Presentation	Speakers
FASE	08/04/2024	Runtime Approach...	10:30	11:00	Foundations for Query-based Runtime Monitoring of Temp...	Lucas Sakizloglou, Holger Giese, and Leen Lambers
TACAS	08/04/2024	SAT and SMT	10:30	10:45	TaSSAT: Transferring and Sharing in SAT	Md Solimul Chowdhury, Cayden Codel and Marijn Heule
FASE	09/04/2024	System Comprehe...	10:30	11:00	Fuzzy quantitative attack tree analysis	Thi Kim Nhung Dang, Milan Lopushaa-Zwakenberg, and Ma...
ESOP	11/04/2024	Untitled session	10:30	11:00	Verified Inlining and Specialisation for PureCake	Hrutvik Kanabar, Kacper Korban and Magnus O. Myreen.
FOSSACS	10/04/2024	Automata and synt...	10:30	11:00	Determinization of Integral Discounted-Sum Automata is ...	Shaull Almagor and Neta Dafni
TACAS	08/04/2024	Untitled session	10:30	11:00	Fully Generalized Reactivity(1) Synthesis	Rüdiger Ehlers and Ayrat Khalimov
TACAS	09/04/2024	Program Analysis ...	10:30	11:00	SootUp: A Redesign of the Soot Static Analysis Framework	Kadiray Karakaya, Stefan Schott, Jonas Klauke, Eric Bodde...
TACAS	10/04/2024	Software Verificati...	10:30	11:00	Accelerated Bounded Model Checking Using Interpolation ...	Mayank Solanki, Prantik Chatterjee, Akash Lal and Subhaji...
TACAS	11/04/2024	Games	10:30	11:00	Auction-Based Scheduling	Guy Avni, Kaushik Mallik and Suman Sadhukhan.
ESOP	10/04/2024	Domain-Specific L...	10:30	11:00	Circuit Width Estimation via Effect Typing and Linear Depe...	Andrea Colledan and Ugo Dal Lago.

Figure 4: . *Sorting on start date.*

Conference	Date (dd-mm-yyyy)	Topic	Start Time	End Time	Title of Presentation ↑	Speakers
TACAS	08/04/2024	Proof Checking	15:00	15:30	A State-of-the-Art Karp-Miller Algorithm Certified in Coq	Thibault Hilaire, David Ilicinkas and Jérôme Leroux.
TACAS	10/04/2024	Software Verificati...	12:00	12:30	A Comprehensive Specification and Verification of the L4 ...	Leping Zhang, Yongwang Zhao and Jianxin Li
ESOP	11/04/2024	Verification	14:00	14:30	A Denotational Approach to Release-Acquire Concurrency	Yotam Dvir, Ohad Kammar and Ori Lahav.
ESOP	10/04/2024	Bidirectional typin...	14:00	14:30	A formal treatment of bidirectional typing	Liang-Ting Chen and Hsiang-Shang Ko.
TACAS	08/04/2024	Proof Checking	15:30	16:00	A Logical Treatment of Finite Automata	Nishant Rodrigues, Mircea Sebe, Xiaohong Chen and Grigo...
ESOP	11/04/2024	Abstract Interpreta...	16:30	17:00	A Modular Soundness Theory for the Blackboard Analysis ...	Sven Keidel, Dominik Helm, Tobias Roth and Mira Mezini.
TACAS	10/04/2024	Simulations	17:30	18:00	A Parallel and Distributed Quantum SAT Solver Based on E...	Shang-Wei Lin, Tzu-Fan Wang, Yean-Ru Chen, Zhe Hou, Da...
TACAS	10/04/2024	Software Verificati...	10:30	11:00	Accelerated Bounded Model Checking Using Interpolation ...	Mayank Solanki, Prantik Chatterjee, Akash Lal and Subhaji...
TACAS	10/04/2024	Probabilistic Verifi...	14:00	14:30	Accurately Computing Expected Visiting Times and Statio...	Hannah Mertens, Joost-Pieter Katoen, Tim Quatmann and ...
TACAS	08/04/2024	Logic and Decidab...	16:30	17:00	Answering Temporal Conjunctive Queries over Description...	Lukas Westhofen, Christian Neurohr, Daniel Neider and Je...

Figure 5: . *Sorting on presentation title.*

- Committee modification: The existing implementation had some problems with state. There was a case where on data modification the client's state was changed, even if the backend rejected the changes due to a malformed request or invalid data entered. Making use of JavaScript Promises this was solved, as the client's state now only updates if the underlying database transaction succeeds.

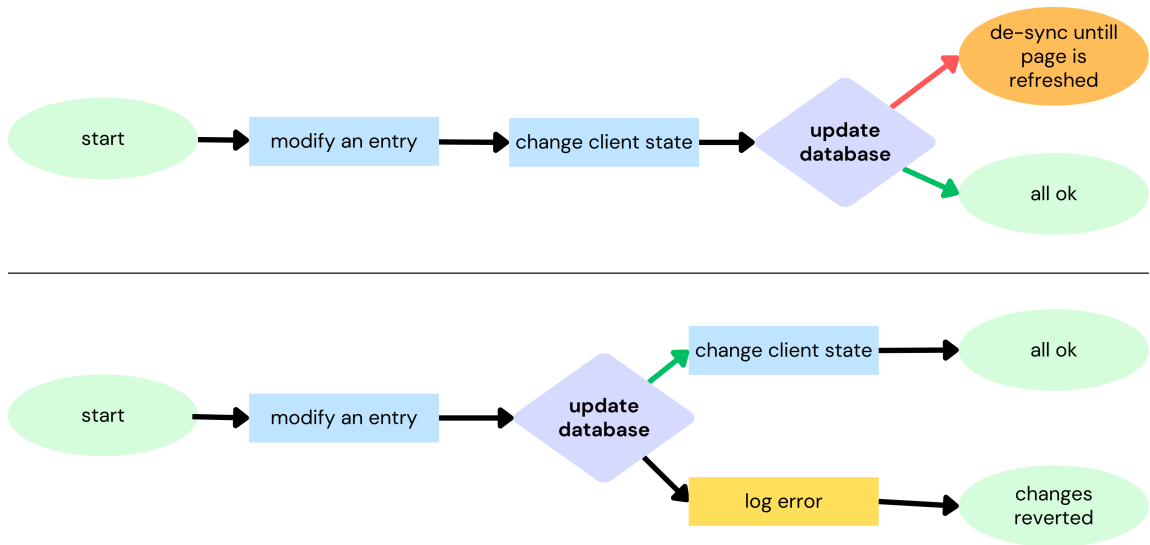


Figure 6: *top: existing mutation flowchart, bottom: updated mutation flowchart*

7. Duplicate Detection: The duplicate detection system has been enhanced to provide more detailed and comprehensive warnings. The current warnings provide the similarity in percentage, the field(s) in which this similarity was observed, and the name and email of the associated individuals.

Status	First Name	Last Name	Email	Country	Affiliation
PENDING	Aleksy	Schubert	alx@mimuw.edu.pl	Poland	University of Warsaw
PENDING	Alexandra	Silva	alexandra.silva@c...	USA	Cornell University
PENDING	Alwen	Tiu	alwen.tiu@anu.ed...	Australia	Australian National University
PENDING	Abrus	Kaposi	akaposi@inf.elte.hu	Hungary	Eötvös Loránd University
PENDING	Barbara	König	barbara.koenig@u...	Germany	niversity of Duisburg-Essen
PENDING	Bartek	Klin	bartek.klin@cs.ox...	UK	University of Oxford
PENDING	James	Worrell	jbw@cs.ox.ac.uk	UK	University of Oxford

Figure 7: . *Updated duplicate detection.*

8. Import and Export: another endpoint was added for the export, to simplify file generation. Importing was implemented through the existing "committee modification" endpoint. Addition functionality for exporting in CSV and YAML format has been added, as well as CSV imports.
9. Automatic import of database: For this version of the system, Docker was employed and configured such that the data in the database is imported automatically when the container is started. This ensures automation and consistency throughout the systems.
10. Moving Sessions: Instead of deleting an activity and creating it again on a different day, admin users can now 'move' a session to the same time but on a different day - as the availability permits. This retains all metadata about the activity, simply changing the date - allowing an admin to save time.

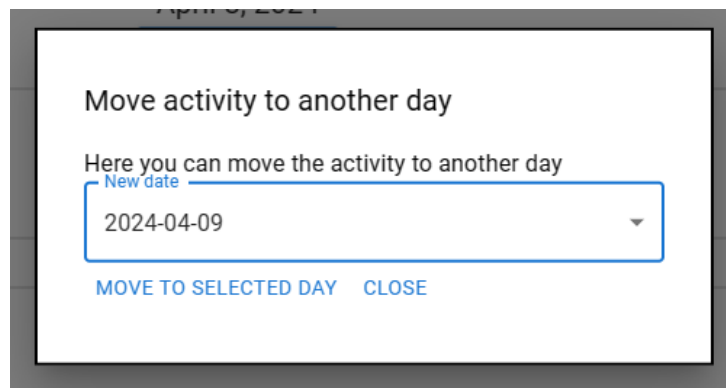


Figure 8: . *Activities can now be moved.*

11. Committee Chairs Permissions: Role-based access control mechanisms have been extended to areas such as schedule editing. Committee chairs can now view and edit schedules and committees of their own, but only view the schedules and committee members of other committees. They are restricted to read-only privileges concerning the schedules of committees they are not a chair of.
12. UI improvements: Features to enhance the interface were also added. For example, the description and titles of activities is now visible on the main schedule page from the admin side, the system no longer glitches during scrolling, among others.
13. UX improvements: Features to smoothen user experience have been added. For example, admin receives a confirmatory notification before

completely deleting data for an entire year, also as previously mentioned, 'Accounts' is now accessible from the navigation bar through an admin account instead of solely through the homepage.

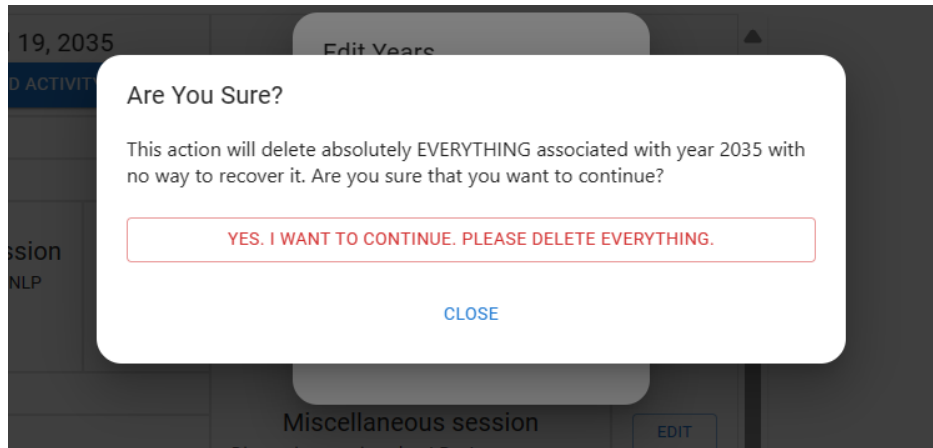


Figure 9: . Confirmation message before deleting year.

6 Design

Making system design decisions was a bit different than working from scratch. Often, there was the choice of re-building a given feature from scratch, or reuse and modify the existing codebase. For example, the "Import from CSV" feature 5.8 was implemented purely in the front-end by calling an existing end-point; "Export to YAML" on the other hand was implemented by adding a new end-point to the backend. Both features were built within the existing project structure.

6.1 Admin Interface

Upon logging in as a superuser or admin user, the main page is displayed, where 3 buttons redirect to different pages: "Conference Creation," "Account management" and "Selection Committee". The navigation bar is present after logging in and is persistent across all pages to make navigation more efficient between the three options. It also contains the option to select or edit the year, an option that only the admin has.

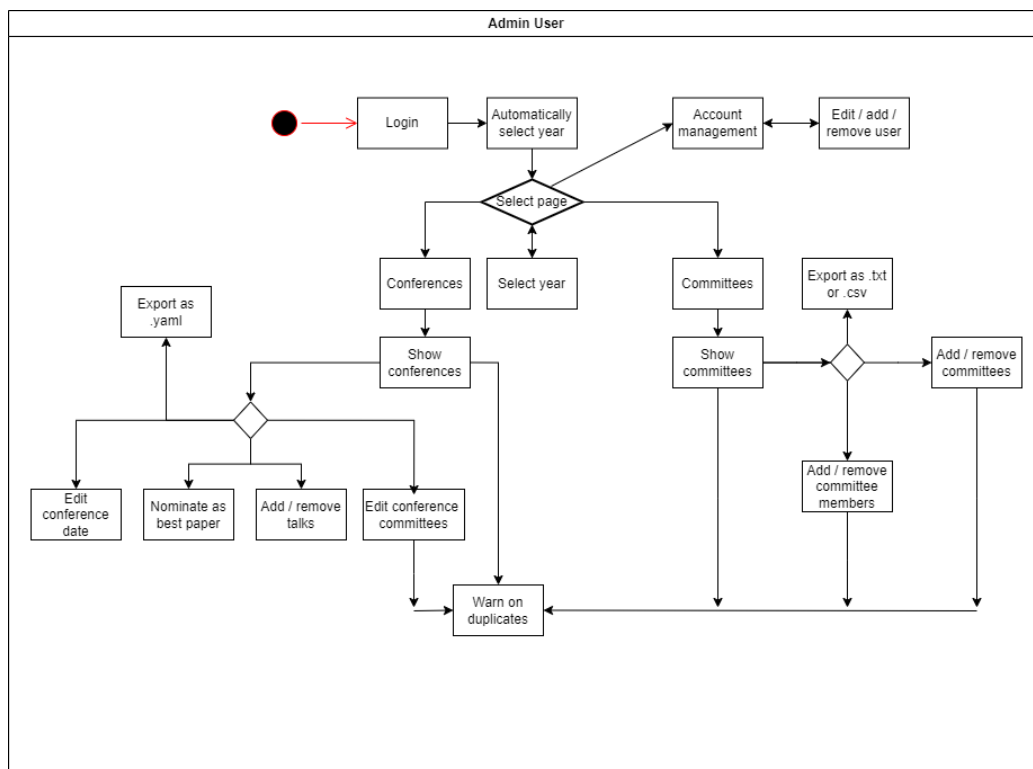


Figure 10: . Walkthrough of admin interface.

6.1.1 Conference Creation

When pressing on “Conference Creation”, the admin is redirected to the scheduler, where they can create a multi-conference schedule. In the scheduler interface, the admin is shown an existing schedule, if one exists for the selected year, or the option to create a new schedule. When creating a new schedule, the admin is required to enter the start date and end date and to add the participating committees. When a schedule already exists, the admin can edit dates and committees as well as edit individual activities or delete them. When editing an activity, the admin can change the speaker, modify the start date and duration, and also add an activity in parallel. There is also the possibility to add an activity with a specified speaker as well as the option to export the entire schedule in YAML format, which is accepted by the ETAPS website. The new functionality that was added handles swapping sessions. An admin can now press the button “move” located on an activity and swap places with another activity.

6.1.2 Selection Committee

If the admin chooses “Selection Committee”, they are redirected to the Committees page. There they are shown a table with all members of a committee for the specified year in the navigation bar. An admin can commute between different committees and edit/add members. The admin is also provided with the possibility to export the table, delete it, or see a general overview of the members of the selected committee. On the overview page, the admin can check the improved warnings tab by pressing “warnings”. In the “warnings” tab, possible errors can be high similarity in first name, last name, email, and a person being in more than one committee. The admin can filter from the warning to see more details about that member in the table. A warning can be deleted, and if done so by mistake, it can be refreshed from the warnings tab.

6.1.3 Account Management

Other functionalities of the committee’s page are described in the “Non-admin” subsection. When selecting “Account Management”, the admin is shown the list of all users and has the ability to edit, delete, or add users. When creating a user, the admin can choose its role. Therefore, from this page, non-admin and admin users can be created.

6.2 Non-Admin Interface

Upon logging in as a committee chair, or a non-admin user, one is greeted with the main page, from where the user can further fork into three different pages: sessions, committee selection, and overview.

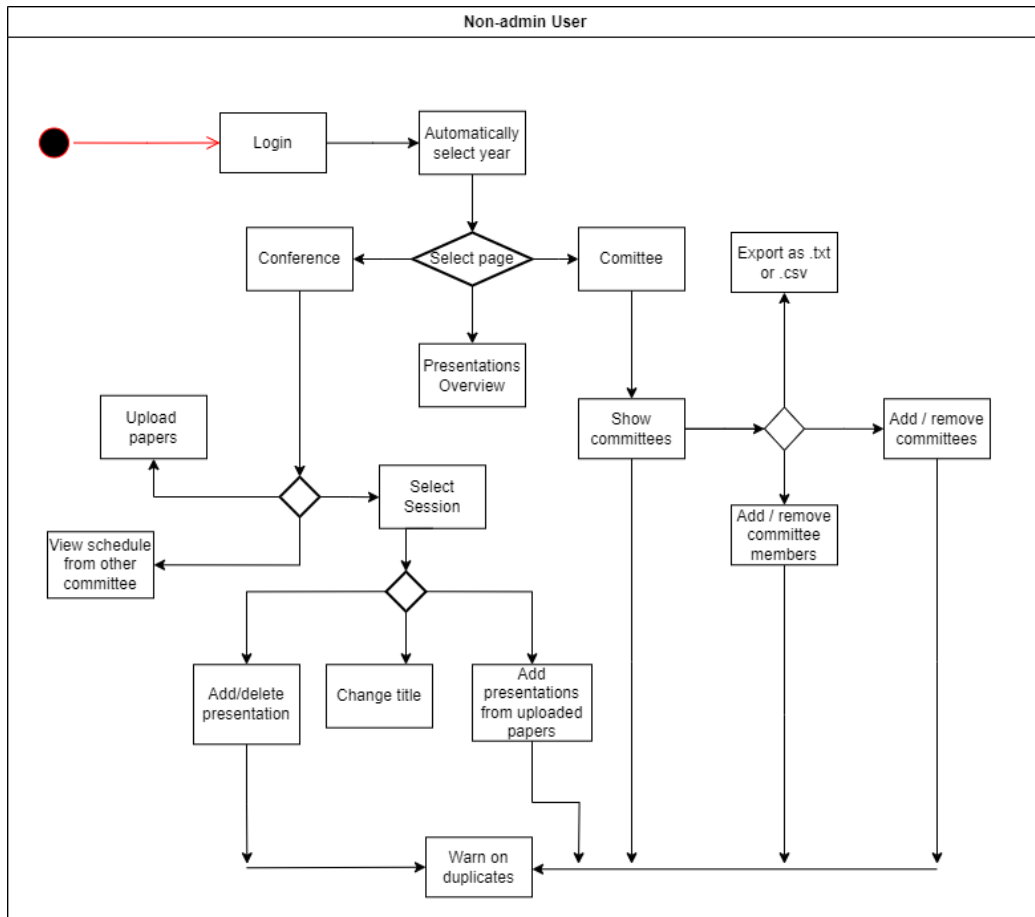


Figure 11: . Walkthrough of non-admin interface.

6.2.1 Sessions

The sessions page opens to the committee of the committee chair and the current year by default. This page displays all the sessions scheduled within said committee. The committee chair can also upload papers on this page. Additionally, sessions for differing years can be accessed here as well. The user can also navigate to view the schedules of other committees - however, this is with read-only access. On the page of their own committee, each session displays the date it is conducted on, the title, the time slot of the overall

session, and the number of presentations in each session, also known as 'slots'.

Upon clicking on a session, the user is taken to a page displaying each presentation in said session. Metadata such as the chair of the session and room it will be conducted in can be edited on this page as well. The page also provides information such as the time slot, duration, title and authors of each presentation. It is also important to realise each presentation is associated with a paper. The user can further add a presentation manually, or add one corresponding to a paper that already exists in the system from 'Add presentations from uploaded papers'. The committee chair can choose to delete a paper, and now also select multiple papers as Best Papers.

6.2.2 Committee Selection

This page is concerned with managing a committee. Similar to the previous page, this page displays the committee members of the committee the user belongs to, for the current year by default. Attributes such as the first and last names, email, country and university each member is affiliated with are displayed on the screen. An entry also has a URL that directs a user to the personal page of the person corresponding to said entry. A committee chair can further add a new member to their committee, and view members of other committees. The permissions for this page are similar to that of the previous one: user has read and write access for their own committee, and read-only access for pages concerning other committees - which is unlike the admin permissions for the same page. A user can further filter entries based on columns and values as well. This table can now be exported in CSV and YAML, and imported successfully from CSV as well.

6.2.3 Overview

This page successfully displays an overview of all presentations belonging to their respective conferences and committees.

7 Risk Analysis

This section consists of the identification, prioritization and solution of various risks during the process. These risks can be divided into two distinct categories: team and code. The former deals with issues that could be experienced within human side of the project. This includes teamwork and client communication. The latter addresses issues that may be encountered within the system and directly affect the final product and therefore the stakeholders. Each issue is assigned a priority level of high, medium or low based on how much damage this issue could cause and how fast we have to deal with it.

7.1 Risks within Team

- Issue: Member absence and/or inconsistency
Priority: High
Solution: Identify reason for absence and/or inconsistency, provide further support or redistribute tasks as per necessary.
- Issue: Lack of communication
Priority: High
Solution: Establish communication channels that must be checked regularly for updates. Lack of communication or miscommunication must be addressed immediately.
- Issue: Conflict among team members
Priority: High
Solution: Be respectful to your teammates throughout the project. In the case of conflict, identify the reason for conflict, communicate honestly and work on conflict resolution.
- Issue: Unclear responsibilities
Priority: Medium
Solution: This can either lead to confusion or double work. Clearly define roles, and verify in case of any confusion what a members' responsibilities are.
- Issue: Technical difficulties
Priority: Low
Solution: Systems should be up and running on everyone's devices, and on git. This is so that damage will be minimum in the case that a member experiences technical difficulties.

7.2 Risks within Project

- Issue: Stakeholder Misalignment
Priority: High
Solution: Communicate with clients as much as possible (end of every sprint) in order to ensure the features developed align with their requirements and/or expectations.
- Issue: Quality Assurance
Priority: High
Solution: It is crucial that the implemented features and or updates are functional, therefore regular testing must be conducted to ensure the same.
- Issue: Database Failure
Priority: High
Solution: Web-app should be able to detect and gracefully handle database failures to minimise disruption or data loss. Regular back-ups of the database should also be taken.
- Issue: Privacy and security concerns
Priority: Medium
Solution: Hash sensitive data, do not collect data that is not necessary, update outdated packages, provide RBAC for specific features.
- Issue: Maintainability
Priority: Medium
Solution: Use popular open-source libraries, without paid variants.

8 Testing

After discussion with our client, it was agreed upon that testing was a relatively low priority task. The reasoning behind this is that the our product is a management support system that will only be used by a select few people who are directly involved in organising the ETAPS conference. Because of this, the occasional edge case will not be as severe of an issue as it would be in a product which is used by a wider audience.

However, it is impossible to guarantee that the end product is functional without any testing. Therefore we used two different testing approaches.

8.1 Manual testing

Manual testing involves executing test cases manually to verify the behaviour of the system. We considered this important since the system was to be developed for two main stakeholders, one of which was one of our clients. This meant we were able to get feedback directly from one of the stakeholders of the system. Our client was able to give us feedback still after this, proving places of improvement. Additionally, we also got team members that worked on different parts of the code to test other parts and assess whether parts of the system were functional and intuitive.

8.2 Unit testing

The ability to create, edit and view data together, concerns the primary functionality of the system. Hence, in order to ensure the proper working of these data operations, we decided to employ unit testing for each functionality. We decided to test the functionality of the GET, POST, PATCH and DELETE methods for the routers concerning committees, conferences, exports, programs, sessions, users and years. The framework used to conduct these tests is PyTest and the methods below have also considered edge cases in order to ensure robustness.

- **GET Methods:** These methods are responsible for retrieving data from the database.
Test: For each of these tests, data is retrieved from the database by using a unique identifier. An identifier that corresponds to no entry in the database is also used, to test an edge case.
Verification: The system verifies that the response has a status code of 200 (OK) and is not null. The ID that does not correspond to an entry in the database returns a 404 (Not Found) status.

- **POST Methods:** These methods are responsible for sending data to the database.
Test: For each of these tests, dummy data is sent to the database using the API endpoint corresponding to a POST request for that type of data.
Verification: The system again verifies that the response has a status code of 200 (OK) indicating that the data was posted successfully.
- **PATCH Methods:** These methods are responsible for modifying or updating entries in the database.
Test: For each of these tests, data is sent to the database using the corresponding POST request endpoint. Then, ID of the data is extracted from the response. Finally, a PATCH request is made updating certain attributes of the data corresponding to the ID.
Verification: As with the rest, the system verifies that the response has a status code of 200 (OK) if the entry exists and is updated successfully, and 404 (Not Found) if the id does not exist.
- **DELETE Methods:** These methods are responsible for deleting entries in the database.
Test: For each of these tests, dummy data is sent to the database via a POST request, and then deleted using a DELETE request.
Verification: Successful deletion of an entry results in a 200 (OK) response code, while attempts at deleting an entry that does not exist, results in a 404 (Not Found) response code.

```
===== test session starts =====
platform linux -- Python 3.11.8, pytest-7.3.0, pluggy-1.4.0
rootdir: /home/luna/git/etaps-management-system/backend/app
plugins: anyio-4.3.0, cov-4.0.0, asyncio-0.21.0
asyncio: mode=Mode.STRICT
collected 38 items

tests/test_routers/test_committees.py ..... [ 18%]
tests/test_routers/test_conferences.py ..... [ 44%]
tests/test_routers/test_exports.py .. [ 50%]
tests/test_routers/test_programs.py ..... [ 63%]
tests/test_routers/test_sessions.py ..... [ 76%]
tests/test_routers/test_users.py ..... [ 94%]
tests/test_routers/test_years.py .. [100%]

===== warnings summary =====
venv/lib/python3.11/site-packages/passlib/utils/__init__.py:854
  /home/luna/git/etaps-management-system/backend/app/venv/lib/python3.11/site-packages/passlib/utils/_
__init__.py:854: DeprecationWarning: 'crypt' is deprecated and slated for removal in Python 3.13
    from crypt import crypt as _crypt

tests/test_routers/test_committees.py: 3 warnings
tests/test_routers/test_programs.py: 9 warnings
  /home/luna/git/etaps-management-system/backend/app/venv/lib/python3.11/site-packages/pandas/core/dt
ypes/cast.py:1429: DeprecationWarning: np.find_common_type is deprecated. Please use `np.result_type`
or `np.promote_types`.
    See https://numpy.org/devdocs/release/1.25.0-notes.html and the docs for more information. (Deprec
ated NumPy 1.25)
    return np.find_common_type(types, [])

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 38 passed, 13 warnings in 37.23s =====
```

Figure 12: . Passed unit tests using PyTest

9 Conclusion

Our final product, has successfully incorporated all the requirements from the "Must Have" and "Should Have" sections of the requirements. Throughout the project, we were able to utilise the strengths of each of our teammates in varying departments to ensure success.

Our journey throughout the lifespan of this project has come with its own set of challenges. From struggling a bit to set the system up on everyone's devices to bugs that were difficult to fix, as a team, we were able to learn a lot. This includes learning how to work with frameworks that were new to some of us, discovering solutions to different problems, gaining insight into ETAPS and its conference management, as well as effectively collaborating within a team.

This project has been an insightful experience for members of our team, allowing us to apply the skills honed during the years of our study into a final project. We believe that our project has indeed satisfied many of the requirements of our clients and enhanced the overall user experience, and structure of the system. We would also like to extend our gratitude to Marieke Huisman and Jan Kofron for their guidance and insights during the developmental process.

10 Future Work

There is much scope to improve this product even further in a next iteration. Individuals working on future enhancements may begin with the "Could Have" section of the requirements, as these were not implemented due to time constraints and other problems. Additionally, improvements our clients had not mentioned as 'requirements' that we noticed during the lifespan of this project include:

- Changing the Scheduling table: Right now it is a table, which makes it difficult to arrange multiple sessions in the same block when they have different lengths and start times. The reason for this is that each 15 minute interval is actually a row in the table and a session is added to the row of its start time and extended as row duration to its end time. In the future, the best solution would be to create a calendar like schedule instead of a table, since with a table you will always have to work with indices, duration and so on which are likely to introduce bugs and create code that is hard to read. Another solution would be to keep on working with a table, but then all edge cases need to be considered before writing the functionality.
- Runtime failure handling: It was pretty clear that this program wasn't built to be mission critical, prioritizing functionality over robustness, fail-safes, and error logging. However, if designed well, failures can be minimized preventing frustration, *especially* when modifying data in the database. Implementing chained Promises¹ would make the system way more robust against unexpected behaviour, since "things only happen" if everything goes well. Additionally, the Promise design pattern encourages developers to consider failure at every step.

¹https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise#chained_promises

References

- [1] *About ETAPS*. <https://etaps.org/about/etaps-conferences/>.
- [2] *What is YAML*. <https://www.redhat.com/en/topics/automation/what-is-yaml>.
- [3] Inc. Project Management Institute. *Agile Practice Guide*. Project Management Institute, Inc. (PMI), 2017. ISBN: 978-1-62825-199-9. URL: <https://app.knovel.com/hotlink/toc/id:kpAPG00001/agile-practice-guide/agile-practice-guide>.
- [4] *What is MongoDB*. <https://www.mongodb.com/company/what-is-mongodb>.
- [5] *React*. <https://react.dev/>.
- [6] *FastAPI features*. <https://fastapi.tiangolo.com/>.
- [7] *Docker platform overview*. <https://docs.docker.com/get-started/overview/>.
- [8] *Github*. <https://github.com/>.
- [9] *What is Trello*. <https://trello.com/tour>.
- [10] *Discord*. <https://discord.com/>.
- [11] *What is MoSCoW and how does it work*. <https://www.productplan.com/glossary/moscow-prioritization/>.

11 APPENDIX

A

GANTT Chart for Planning

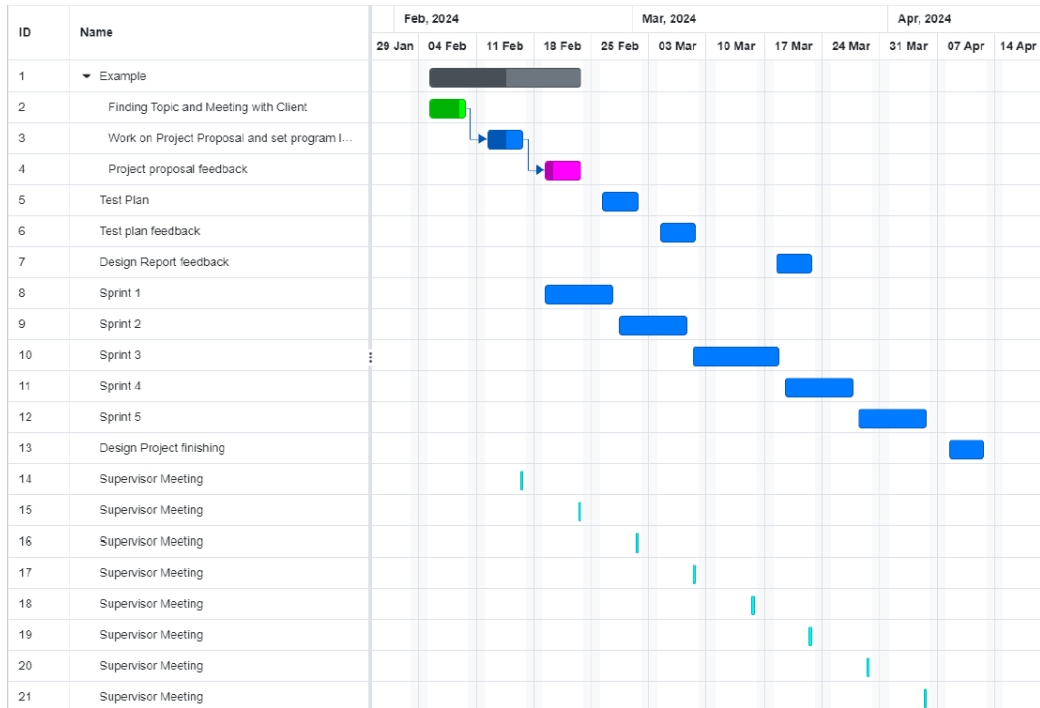


Figure 13: . GANTT Chart displaying the planning as scheduled.

B

Sorting and Filtering

Conference	Date (dd-mm-yyyy)	Topic	Start	End Time	Title of Presentation	Speakers
FASE	08/04/2024	Runtime Approach...	10:30	11:00	Foundations for Query-based Runtime Monitoring of Temp...	Lucas Sakizloglou, Holger Giese, and Leen Lambers
TACAS	08/04/2024	SAT and SMT	10:30	10:45	TaSSAT: Transferring and Sharing in SAT	Md Solimul Chowdhury, Cayden Codel and Marijn Heule
FASE	09/04/2024	System Comprehe...	10:30	11:00	Fuzzy quantitative attack tree analysis	Thi Kim Nhung Dang, Milan Lopuhaä-Zwakenberg, and Ma...
ESOP	11/04/2024	Untitled session	10:30	11:00	Verified Inlining and Specialisation for PureCake	Hrutvik Kanabar, Kacper Korban and Magnus O. Myreen.
FOSSACS	10/04/2024	Automata and synt...	10:30	11:00	Determinization of Integral Discounted-Sum Automata is ...	Shaull Almagor and Neta Dafni
TACAS	08/04/2024	Untitled session	10:30	11:00	Fully Generalized Reactivity(1) Synthesis	Rüdiger Ehlers and Ayrat Khalimov
TACAS	09/04/2024	Program Analysis ...	10:30	11:00	SootUp: A Redesign of the Soot Static Analysis Framework	Kadiray Karakaya, Stefan Schott, Jonas Klauke, Eric Bodde...
TACAS	10/04/2024	Software Verificati...	10:30	11:00	Accelerated Bounded Model Checking Using Interpolation ...	Mayank Solanki, Prantik Chatterjee, Akash Lal and Subhaji...
TACAS	11/04/2024	Games	10:30	11:00	Auction-Based Scheduling	Guy Avni, Kaushik Mallik and Suman Sadhukhan.
ESOP	10/04/2024	Domain-Specific L...	10:30	11:00	Circuit Width Estimation via Effect Typing and Linear Depe...	Andrea Colledan and Ugo Dal Lago.

Figure 14: . *Sorting on Start-time.*

Conference	Date (dd-mm-yyyy)	Topic	Start Time	End Time	Title of Presentation ↑	Speakers
TACAS	08/04/2024	Proof Checking	15:00	15:30	A State-of-the-Art Karp-Miller Algorithm Certified in Coq	Thibault Hilaire, David Ilicinkas and Jérôme Leroux.
TACAS	10/04/2024	Software Verificati...	12:00	12:30	A Comprehensive Specification and Verification of the L4 ...	Leping Zhang, Yongwang Zhao and Jianxin Li
ESOP	11/04/2024	Verification	14:00	14:30	A Denotational Approach to Release-Acquire Concurrency	Yotam Dvir, Ohad Kammar and Ori Lahav.
ESOP	10/04/2024	Bidirectional typin...	14:00	14:30	A formal treatment of bidirectional typing	Liang-Ting Chen and Hsiang-Shang Ko.
TACAS	08/04/2024	Proof Checking	15:30	16:00	A Logical Treatment of Finite Automata	Nishant Rodrigues, Mircea Sebe, Xiaohong Chen and Grigo...
ESOP	11/04/2024	Abstract Interpreta...	16:30	17:00	A Modular Soundness Theory for the Blackboard Analysis ...	Sven Keidel, Dominik Helm, Tobias Roth and Mira Mezini.
TACAS	10/04/2024	Simulations	17:30	18:00	A Parallel and Distributed Quantum SAT Solver Based on E...	Shang-Wei Lin, Tzu-Fan Wang, Yean-Ru Chen, Zhe Hou, Da...
TACAS	10/04/2024	Software Verificati...	10:30	11:00	Accelerated Bounded Model Checking Using Interpolation ...	Mayank Solanki, Prantik Chatterjee, Akash Lal and Subhaji...
TACAS	10/04/2024	Probabilistic Verifi...	14:00	14:30	Accurately Computing Expected Visiting Times and Statio...	Hannah Mertens, Joost-Pieter Katoen, Tim Quatmann and ...
TACAS	08/04/2024	Logic and Decidab...	16:30	17:00	Answering Temporal Conjunctive Queries over Description...	Lukas Westhofen, Christian Neurohr, Daniel Neider and Je...

Figure 15: . *Sorting on Title.*

C

Duplicate detection

The screenshot shows a web application interface with a blue header containing navigation links: 'E T A P S', 'Conferences', 'Accounts', 'Commitees', and a year selector '2025'. The main content area is titled 'General Overview' and contains a table with the following data:

Status	First Name	Last Name	Email	Country	Affiliation
PENDING	Aleksy	Schubert	alx@mimuw.edu.pl	Poland	University of Warsaw
PENDING	Alexandra	Silva	alexandra.silva@c...	USA	Cornell University
PENDING	Alwen	Tiu	alwen.tiu@anu.ed...	Australia	Australian National University
PENDING	Abrus	Kaposi	akaposi@inf.elte.hu	Hungary	Eötvös Loránd University
PENDING	Barbara	König	barbara_koenig@u...	Germany	niversity of Duisburg-Essen
PENDING	Bartek	Klin	bartek.klin@cs.ox...	UK	University of Oxford
PENDING	James	Worrell	jbw@cs.ox.ac.uk	UK	University of Oxford

Below the table is a 'WARNINGS' button. On the right, a 'Warnings' sidebar is open, showing a list of warnings:

- 1. Possible 95% similarity in first name ,100% and 90% similarity in last name and 100% similarity email, please check Sanjit A. Seshia with sseshia@eecs.berkeley.edu
- 2. Possible 100% similarity in first name ,100% and 90% similarity in last name and 89% similarity email, please check Prakash Saivasan with prakash@imsc.res.in
- 3. Possible 100% and 100% similarity in first name

Figure 16: . Detailed warnings for duplicates.