Sports & Culture

A new proof of concept for a brand new Sports & Culture app

Niek Aukes, Eli Saado, Laura Vuurens, Femke Weijsenfeld & Mette Weisfelt



University of Twente

UNIVERSITY OF TWENTE.

Sports & Culture

A new proof of concept for a brand new Sports & Culture app

Niek Aukes - s2720604
Eli Saado - s2790343
Laura Vuurens - s2827506
Femke Weijsenfeld - s2792699
Mette Weisfelt - s2586665

Supervisor: dr. ir. R. Klaassen
Client: drs. Gijs Verburg - Campus Facility Management
Faculty: Faculty of Electrical Engineering, Mathematics and Computer Science

Contents

1 Introduction **1**
- 1.1 Introduction . . . 1
- 1.2 Outline . . . 1
- 1.3 AI Statement . . . 1

2 Context Analysis **3**
- 2.1 Stakeholders . . . 3
  - 2.1.1 Students and employees . . . 3
  - 2.1.2 Student Union (SU) . . . 3
  - 2.1.3 Campus Facility Management (CFM) . . . 3
  - 2.1.4 Associations . . . 4
  - 2.1.5 External parties . . . 4

3 Planning and Approach **5**
- 3.1 Planning . . . 5
- 3.2 Description of project organisation . . . 6
  - 3.2.1 Procedures . . . 6
  - 3.2.2 Risk Analysis . . . 6
  - 3.2.3 Responsibilities . . . 7
- 3.3 Approach . . . 7
  - 3.3.1 Working in phases . . . 7
  - 3.3.2 Working together . . . 7
  - 3.3.3 Designing based on feedback . . . 7
  - 3.3.4 Communication with our client and supervisor . . . 8
- 3.4 Planned Deliverables . . . 8

4 Initial Interviews **9**
- 4.1 Results . . . 9
  - 4.1.1 Positive opinions about the Sports & Culture app currently in use . . . 9
  - 4.1.2 Issues with the Sports & Culture app currently in use . . . 9
  - 4.1.3 Ideas for a possible new Sports & Culture app . . . 10
- 4.2 Problems . . . 11
  - 4.2.1 Functionality . . . 11
  - 4.2.2 Design and User Experience . . . 12
  - 4.2.3 Safety and Privacy . . . 12
- 4.3 Our Solutions . . . 12
  - 4.3.1 Improving User Experience . . . 12
  - 4.3.2 Improving Ticketing system . . . 13
  - 4.3.3 Improving Reservation system . . . 13
  - 4.3.4 Improving visibility of Culture . . . 13
  - 4.3.5 Adding insights . . . 13

5 User Stories and Requirements **14**
- 5.1 User Stories . . . 14
- 5.2 Requirements . . . 15
  - 5.2.1 MoSCoW Functional Requirements . . . 16
  - 5.2.2 MoSCow Non-Funtional Requirements . . . 17

6 Initial Design **18**
- 6.1 Profiles & Navigation . . . 18
- 6.2 Home . . . 18

6.3 Reservations . . . 19
6.4 Scanning . . . 20
6.5 Profile . . . 21
6.6 Member management . . . 21
6.7 Statistics (CFM) . . . 22
6.8 Association Management (CFM) . . . 22
6.9 Technical System Design . . . 23
6.9.1 Database Design . . . 23
6.9.2 Authentication . . . 24
6.10 Feedback on our shown Figma design . . . 25
6.10.1 Positive feedback . . . 25
6.10.2 Possible improvements . . . 25

7 Prototype Development **26**
7.1 Front-end Design . . . 26
7.1.1 UTwente House Style . . . 27
7.1.2 Activities and My Activities . . . 27
7.1.3 Event and Facility detail pages . . . 28
7.1.4 Shop . . . 28
7.1.5 Ticketing System . . . 29
7.1.6 Profile Page . . . 29
7.1.7 Content accessible without logging in . . . 31
7.1.8 Creating Events . . . 31
7.1.9 Creating and Managing Associations . . . 32
7.1.10 Statistics . . . 32
7.1.11 Association Management . . . 32
7.2 Back-end Design & Modelling . . . 33
7.2.1 Activities . . . 33
7.2.2 Repetitions . . . 34
7.2.3 Reservations . . . 34
7.2.4 Tag System . . . 35
7.2.5 Automatic Hibernate Tables . . . 36
7.2.6 Refresh Token System . . . 36
7.3 Feedback Sessions . . . 36
7.3.1 Feedback from client . . . 36
7.3.2 Feedback from other users . . . 36
7.3.3 New requirements . . . 37

8 Testing and Evaluation **39**
8.1 Unit Testing . . . 39
8.2 User Testing . . . 40
8.2.1 Task-based Testing . . . 40
8.2.2 User Feedback . . . 41
8.2.3 Results . . . 41

9 Final Prototype **44**
9.1 Changes . . . 44
9.1.1 Sharing Links . . . 45
9.1.2 Create Association & Create Activity . . . 45
9.2 Requirements Overview . . . 46

10 Discussion **50**
10.1 Analysis of Findings . . . 50
10.2 Future Work . . . 50
10.2.1 SURFConext and Integration with University Systems . . . 50
10.2.2 NFC . . . 51
10.2.3 Shop . . . 51
10.2.4 Web Interface . . . 51

10.2.5 Extended Tag System 51
10.3 Reflection 51

11 Conclusion **53**

A Overview Interactions **54**

B Manuals **55**
B.1 Manual for students, employees and external users 55
B.1.1 Activities 55
B.1.2 Union Card / Campus Card 57
B.1.3 Associations 58
B.2 Manual for associations and external parties 59
B.2.1 Activities 59
B.2.2 Scanning 59
B.2.3 Member Management 59
B.3 Manual for Campus Facility Management and Student Union 61
B.3.1 Statistics 61
B.3.2 Activities 61
B.3.3 Association Management 61
B.3.4 Approvals 61

1 Introduction

1.1. Introduction

The University of Twente has around 60 sports and culture associations. These associations make use of the multitude of facilities that are offered on campus. In order to make use of these facilities and join associations, students and employees of the University of Twente are required to purchase a Union Card.

Currently, the university offers both a web application and an Android and iOS app for students to use. The system behind it is very similar for both the app and the web app. In theory, students are able to buy a Union Card, pay their association fees, and make reservations for the use of facilities. In practice, however, a lot of users experience problems using this app or web application.

Why does the Sports & Culture app have an average of 1.5 stars on the Google Play Store? Why do users experience inconvenience while using the current Sports & Culture app? How can we reduce this or prevent this entirely? In what ways can we improve the functionality of the app? In this design project, we will mainly focus on answering these questions. We will put our efforts into the design of a new mobile-based application.

1.2. Outline

We have structured our report and our project approach into four phases. In the first phase, phase 0, we talk to our client, define stakeholders, and identify the problem. We will also make a planning and decide on our approach. After phase 0 we start phase 1, in which we investigate the needs of our stakeholders, do interviews to create clear requirements, and create an initial design of the product in Figma. After the initial design is created, we will enter phase 2. In phase 2 we will start the actual development process and make system design choices. During this phase, we will routinely interview our stakeholders for feedback. By doing this, we can iterate over our application in multiple smaller cycles, and incorporate feedback faster. We will discuss our final prototype in phase 2 and talk about our approach to testing in phase 3. When phase 3 has ended, we will discuss our process and conclude. These phases are illustrated in Figure 1.1.

The manuals we have written for our stakeholders can be found in Appendix B.

Our source code can be found in our GitLab project: https://gitlab.utwente.nl/s2586665/design-project.

1.3. AI Statement

During the preparation of this work, we used Grammarly for spelling checking, GitHub Copilot for writing code, and ChatGPT for explanation of Spring Annotations and discussion of creative slogans. After using these tools and services, we thoroughly reviewed and edited the content as needed, taking full responsibility for the final outcome.

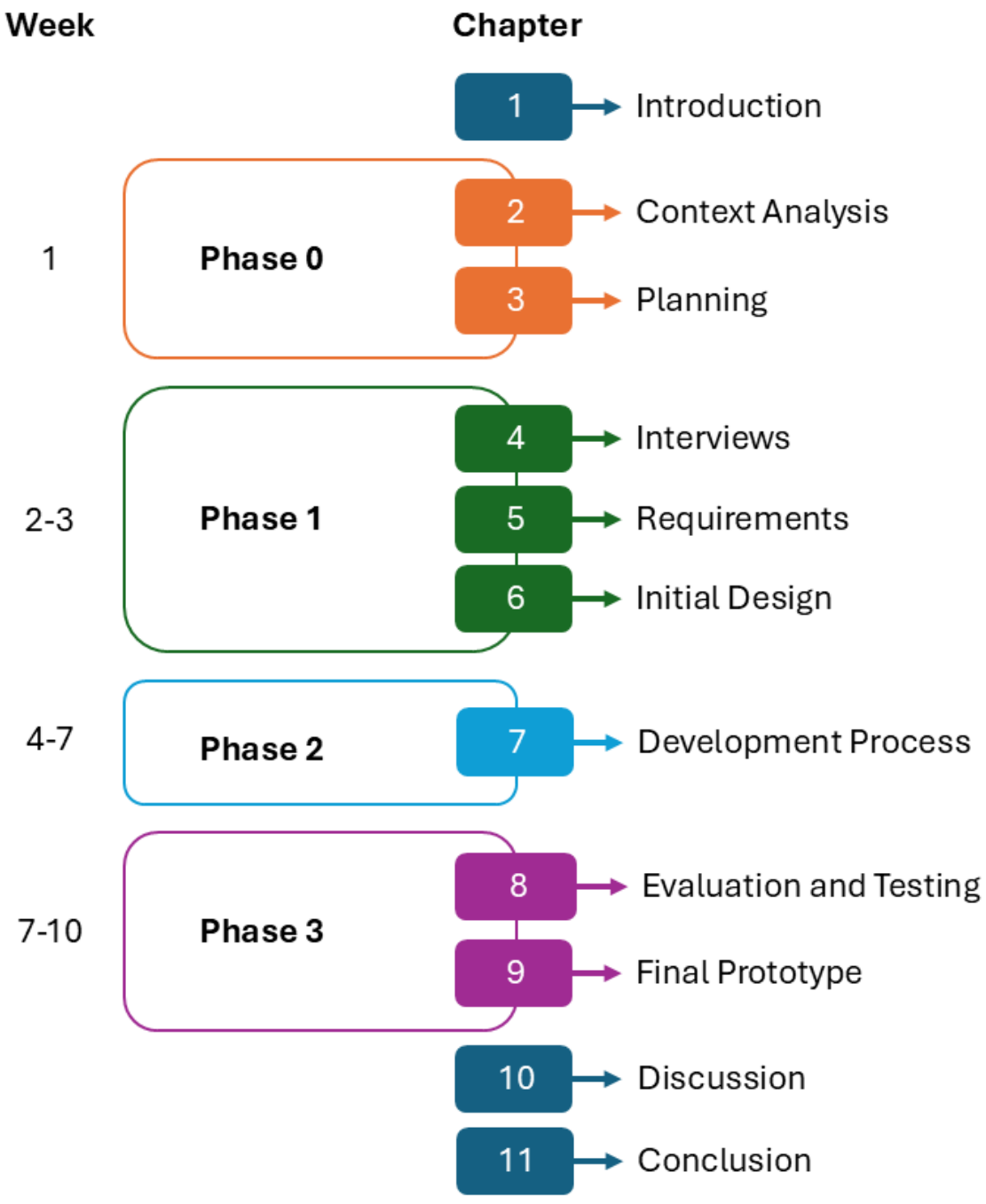


Figure 1.1: Our report structured in different phases per chapter

2

Context Analysis

In order to identify the problem we will start by describing our stakeholders. In order to define our stakeholders and identify current problems with the app, we have talked with both the Student Union and Campus Facility Management. In addition, we have had several interviews with different associations and held a survey among students of this university. A concise overview of interviews can be found in Chapter 4.

2.1. Stakeholders

After conducting the interviews, we have identified five major stakeholders. These are

1. Students and employees of the University of Twente
2. Associations of the university
3. The Student Union
4. External parties
5. Campus Facility Management

All these stakeholders have different expectations from our platform. In the sections below, we will go over these expectations.

2.1.1. Students and employees

Students who have bought a Union Card, or people and employees with a Campus Card, want to be able to join associations. They also want the ability to reserve a facility on campus such as the gym, and sign up for events. They want to be able to open gates of facilities using the app or their student or employee card. They also want to make use of the discounts provided by the Student Union from external parties.

2.1.2. Student Union (SU)

The Student Union sells Union Cards to students. Student Union wants to sell as many Union Cards as possible. With Union Cards, students can get access or discounts to certain locations or activities.

Additionally, the Student Union wants to know how many Union Cards are scanned and at what location in order to better assess the current contracts with external parties. They want to have insight into the use of this card.

2.1.3. Campus Facility Management (CFM)

Campus Facility Management is interested in stimulating students and employees to join sports and cultural activities that are offered on campus. They also manage the physical facilities at the University of Twente.

They want a single channel of communication regarding everything that can be done on campus.

They are our client and main contact person regarding this project.

2.1.4. Associations

Associations require their members to own a Union Card, as mandated by the Student Union. Associations are therefore interested in checking whether members own a Union Card and whether they have paid the association fee.

Some associations are interested in creating activities for their members and for people outside their association.

2.1.5. External parties

External parties are the owners or organisers of several locations or activities. They have a contract with Student Union in which they give a discount to Union Card holders. Therefore, external parties are interested in checking easily and quickly whether a student has a Union Card or not.

In addition, external parties are interested in knowing how many people with a Union Card enter the location, or other relevant statistics and insights.

3

Planning and Approach

Now that we have identified the problems and offered some solutions, we need a plan on how to approach this project. We will discuss the planning, our project organisation, and our procedures below.

3.1. Planning

We will start with the planning for our project. This project will be finished throughout one quartile, which consists of ten weeks. In the table below, we will go over each week and our goals for that week. In Figure 3.1 we display a more global view of the project and when we will work on which part.

Week	**General Planning**
Every	- Work on Design Report
1	- Create planning - Discuss the project
2	- Interviews with stakeholders - Figma Design - Write project proposal - Setup project
3	- Figma Design - Start development back-end + front-end
4	- Continue development - Focus on Home page, Ticketing, and Reservations - Continue app design
5	- Continue development - Focus on Home page, Ticketing, and Reservations - Continue app design
6	- Continue development - Finish Home page, Ticketing, and Reservations - Continue app design
7	- Continue Development - Finish Home page, Ticketing, and Reservations - Start with user testing
8	- Adding additional features to the app - Continue user testing
9	- Continue user testing
10	- Final presentation(s) - Finish Design Report

As discussed before, we will work in different phases in order to receive feedback and have smaller development cycles.

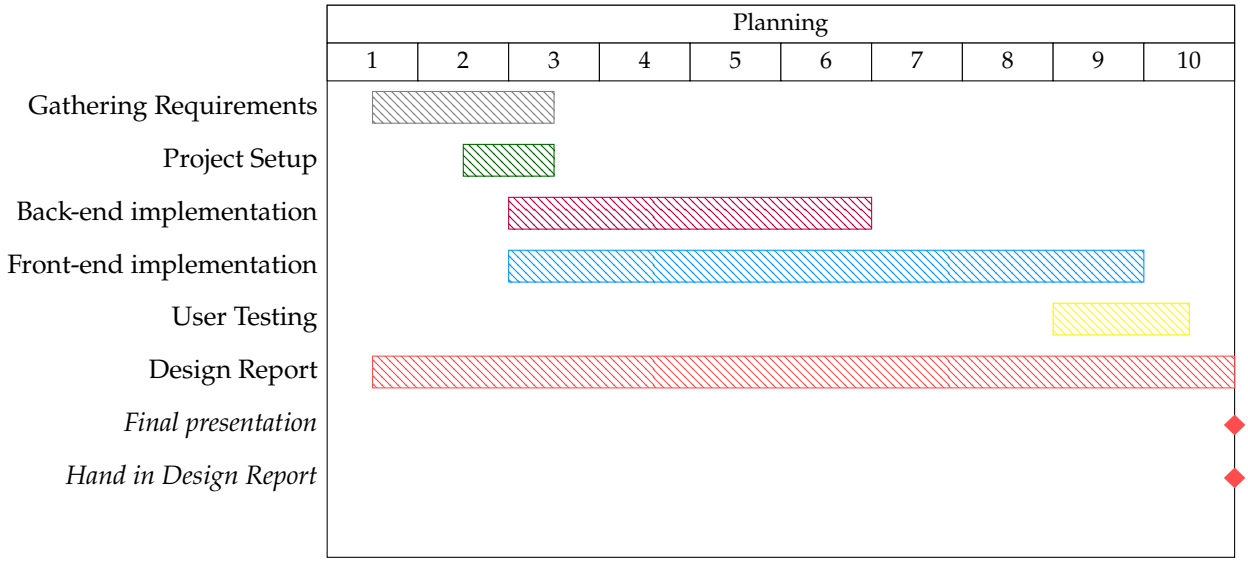


Figure 3.1: Planning of the project and deadlines

3.2. Description of project organisation

3.2.1. Procedures

Team Communication

For team communication, we use a group chat in which we discuss things when we do not meet in person. For tracking of non-development tasks we use Trello. We are all responsible for timely communication of personal issues.

Git

We have some rules regarding code pushing into GitLab. Before merging with main, each request has to be approved by at least one code reviewer, otherwise, the merge will be blocked by GitLab. Also, our created pipelines should pass first. No one can merge their own branch with main without approval. We use feature and bug branching as our main git strategy.

As for development tasks, we use the GitLab issue feature. All team members can create issues, apply tags and assign themselves and a reviewer. These issues should be kept small and well-defined in order to prevent issues with merging later. We think this way of assigning small tasks will make it very clear for everyone.

3.2.2. Risk Analysis

There are different parts of our project that may take longer than expected. That is why we planned a full week of extra time for development. One of the parts that may be risky is the SURFconext connection for logging in. SURFconext is the third party the University of Twente uses to authenticate with Microsoft. This working is partly dependent on another party (LISA) and no one in our group has made a log-in system with SURFconext yet.

In addition, using NFC implementation to read student cards may be risky, since none of us have implemented NFC before. In order to use NFC on an iOS app that can be downloaded from the Apple Store, we need to be an accredited company and apply for this permission to Apple. This is far out of the scope of this project, and we will therefore only implement this feature in a Development build for iOS.

At last, we need to define exactly what our ambition is. We have a very ambitious group and need to communicate clearly what the scope of our project will be. We want to avoid taking on too much and not be able to finish the project within this time period.

3.2.3. Responsibilities

Each person in the group has a main priority and responsibility. This person is not solely working on this part of the project but has the responsibility to make that part a success.

Femke Weijsenfeld: Front-end Development, Communication, Planning
Eli Saado: Front-end Development, Authentication, Hosting, CI/CD
Niek Aukes: Back-end Development, NFC Integration
Bjorn Vuurens: Front-end Development, UI/UX
Mette Weisfelt: Back-end Development, Security and Roles, Database Design

We are all responsible for the final design report and other reports regarding reflection.

3.3. Approach

3.3.1. Working in phases

As described in the introduction of this report and in figure 1.1, we will work through our design project in our self-defined phases.

3.3.2. Working together

We have decided on each of our team member's responsibilities in section 3.2.3. With this, we all have something specific to take care of, but that does not mean someone is working only on that part of the project. We think that having responsibilities will help us guide each other in the process.

In addition, stand-ups are very common in the industry. That is why we will update each other every day on what each team member is doing in the form of a stand-up. In this stand-up, we tell the others what we have been working on yesterday and what we are going to work on that day. By doing this, we know very well what other people are working on and avoid doing the same work twice. In order to do this in the most effective way, we will work offline as much as possible. We will book project rooms at the University of Twente in order to be together. This will make our communication as effective as possible.

3.3.3. Designing based on feedback

We have decided on quite a unique way of completing our project, namely by designing based on feedback. Since our app has a very specific group of people that might use it in the end, we have the ability to get very detailed feedback regarding our design and functionality. Therefore, we started by asking for approval from the Ethics Committee for interviews and surveys regarding our app.

The idea to design based on feedback is well nested within our phases approach. In every phase, we will use interviews to guide us in the right direction. In phase 0, we have used our meetings with Campus Facility Management, the Student Union and four associations, and our survey to create a list of problems with the Sports & Culture app that is currently in use by students and associations. This list can be found in section 4.2.

We will create an initial design prototype in Figma during phase 1. During this design process, we will have more interviews in order to gather feedback from possible end users. In the second phase, we will have started development and will therefore be able to do some user testing to check for intuitive UI.

In the last phase, we will use do more user testing and find the last problems in our app. This will help us get a broader view of future work necessary for our app, and promote discussion of what could have been done differently.

We think this iterative process will significantly improve our design process and thus our final design product. This process is visualised in figure 3.2.

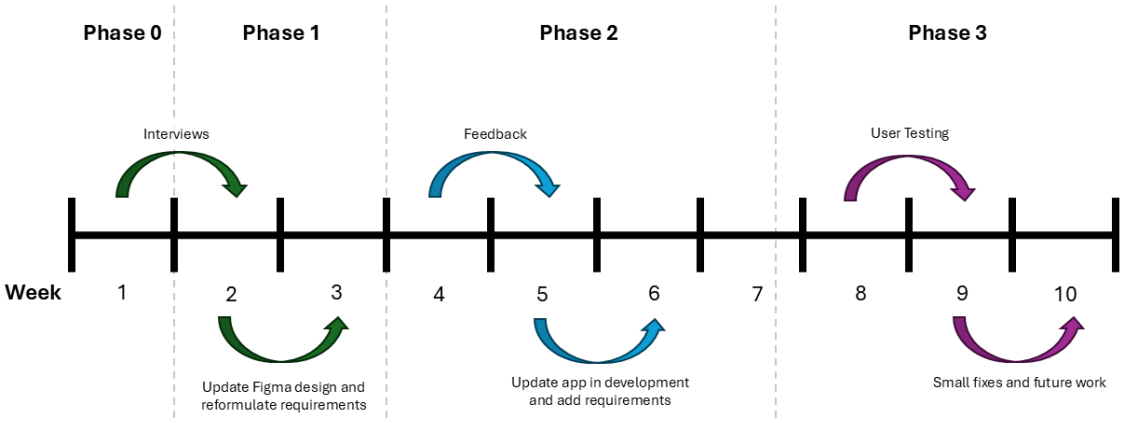


Figure 3.2: Our way of designing our product based on feedback from end-users

3.3.4. Communication with our client and supervisor

We think communication is very important. Therefore, we will prepare the meetings we have with our client and supervisor by creating an agenda for each meeting. This will make sure that we use the time we have during the meeting as effective as possible, by staying on topic.

We will meet our supervisor every week in order to get feedback on our process and report. We will make sure our client is up to date with the current developments, in order to keep our client satisfied. We will regularly plan meetings with our client to do demonstrations so that we get feedback from our client in time to incorporate it into the app.

3.4. Planned Deliverables

We aim to deliver a working proof of concept with a mobile app as the interface. We will build the app in such a way that it would be easily extended to a web interface as well. We, however, consider this to be out of the scope of this project due to the available time. We will host the platform ourselves during the quartile to make it possible to demonstrate our product. Along with this working product, we will deliver a working manual on how the app can be used and what features it offers. We will also produce sufficient documentation so future developers can work on this product.

4

Initial Interviews

Now that we have defined our approach, it is time to enter phase 1 and do semi-structured interviews. The choice of semi-structured interviews lies in the fact that we can then ask extra questions to make the feeling or opinion of the interviewee more clear.

We have interviewed five different people who either have a board or management role at an association or external party. We will divide the interview into two sections. The first section is to further create and specify requirements. We have asked them what they like about the Sports & Culture application that they currently use, what issues they encounter, and what they would like to see if a new Sports & Culture app would be developed.

In the second section of the interview, we also asked the interviewees for feedback about our Figma design, which we created partly in phase 0 and have updated after each interview to get feedback on the latest changes. This way, we try to solve problems along the way in order to get feedback on what we still do not know. The results of these interviews are presented below. They are grouped into different topics.

4.1. Results

4.1.1. Positive opinions about the Sports & Culture app currently in use

In order to start with gathering ideas, we started with discovering what people think is positive about the current application that they use. This way, we can find out what we definitely need to implement.

At first, we found out that the interviewees really liked the idea of different sports combined in one app. This already mentions one problem that our client noticed: when we interview sports associations, they only talk about the idea that it is nice to combine different sports into one app. Culture is left out in the app to such an extent that people do not even consider the possibility of adding culture. In addition, the logo of the app that is currently in use says 'Sports Centre University of Twente', which does not promote culture.

We also noticed that the interviewees really liked overviews to check whether something went right. For example, some said they like the clear overview of all subscriptions they currently own or have had. Others pointed out that it is helpful that they can see whether members still need to pay a fee or not.

4.1.2. Issues with the Sports & Culture app currently in use

Even though the existence of the Sports & Culture app currently in use is really nice and beneficial, most interviewees tell us about the problems they face with the current system. In order to structure their complaints, we will discuss them below in different subsections.

Activities

The first thing we hear our interviewees complain about is that the Schedule page does not show any activities unless you apply some filters. This does not activate the user to explore new activities. Also,

some of our interviewees think it is annoying since it costs more button presses to get somewhere.

Scanning QR-code
Another thing some of our interviewees have an issue with is that scanning QR codes takes up a long time. Our interviewees from external parties think that scanning QR codes is the main purpose of the app since they only use that. The reason why it takes up a long time according to our visitors is because our interviewee's visitors often do not know that they need an app to show their Union Card. "Sometimes people show their student card when we ask them to show their Union Card. We then need to ask them to download the app, log in with two-factor authentication and then scan the QR code." Even though this sounds doable with a reasonable amount of time, our interviewees explain that this can take up a lot of time, especially because some people get errors while logging in. "If they set their standard browser to Chrome, it works again."

Associations and member management
Associations pointed out some issues regarding member management with the current application in use. One thing is that members can show up twice in the member database, which makes it more work for the secretary to manage.

Also, more specific problems were mentioned: when students press the 'join association button' multiple times, the association gets multiple requests. When for example an association accepts one, and then e.g. rejects the other three, then the student requesting gets three rejection messages, which is unwanted because the student was accepted.

In addition, even though our interviewees think that it is very nice to see which members have paid already or not, some interviewees point out more information could be provided. For example, if a student did either not pay a Union Card or membership fee, the secretary cannot see which one, and should then ask the student itself in order to help that student out. The same holds for the student side. The students cannot see the relation they have with an association. For example, they do not know whether their request to become a member of an association is still pending or whether they are already a member of an association.

At last, someone pointed out they have received fines from the Student Union for students not owning a Union Card while being a member. "Apparently, it is not clear for the Student Union that some of our members do not need a Union Card." It would be nice if this issue could be resolved.

Other
Some of our interviewees wanted to point out some issues that we can not fit in the categories above. One of them is that one interviewee thinks that the confirmation email that you get when you buy something looks like a phishing mail. "There is all kinds of information in it that I do not understand."

Secondly, some interviewees say that they encounter the issue that they need to log in every time they open the app, which they think is very annoying.

In addition, some people noted that culture was not represented well in the app, even though it is called 'Sports & Culture'.

At last, some interviewees point out that it is very hard to pay the association fee. It is not clear whether it is a subscription or course (which are the only two categories). In addition, it is only possible to find the association fee via a direct link.

4.1.3. Ideas for a possible new Sports & Culture app
Like described earlier, we have also asked our interviewees what ideas they have for a possible new Sport & Culture app if one is implemented newly. We will describe their ideas grouped together in the topics below.

Activities
Most associations we spoke pointed out that the kick-in period related to intro trainings may be very confusing for new students. A lot of associations host intro trainings and courses, which every student can join to checkout the association and maybe become member later. But now, sign-ups are organised

by associations individually, resulting in Google forms, Microsoft forms or a forms on the association's website to sign-up. Associations point out that it would be nice to centralise these organised events. "Signing up for intro trainings on the Sports & Culture app sounds natural to me. In addition, new students can then also see other activities that the student life here at the University of Twente has to offer." One association points out that they manually add students to an Excel sheet who signed up through email, which costs unnecessary manual labour. All in all, the possibility to create events would be wanted by associations.

One interviewee points out that the idea also has a disadvantage: it would be weird if associations can just create events that everyone can see on the app. "Maybe it needs a sort of approval system?" This is a very good point, which we will take to our client when we meetup with our client again in the next phase.

Scanning QR-code
Some interviewees from external parties think it would be very nice if the current QR-code system would just work with a student card. We proposed the idea of reading the NFC chip from the student card, and then show whether the student owns a Union Card or not. The interviewees think that would be a very great idea.

Associations and member management
A lot of small ideas were presented regarding associations and member management. For example, an interviewee proposed an option to show all possible associations, to show to new students what is possible. Also, it would be nice according to this interviewee if these associations have a logo, since some associations have a similar name here at the University of Twente, which results in new students sometimes joining the wrong association.

Some other interviewees think that it would be nice to show in the My Profile screen what associations they are in, and what role they have. "It would just be very nice to check whether everything is alright."

Statistics
Since our client asked for statistics, we thought we can ask whether associations and external parties would like to see statistics too. All replied positive, so we asked them what statistics they would see as useful.

Our interviewees from associations would like to know how many members they have per module, the number of members in the previous years, and how many members paid for the Union Card and/or association fee. In addition, if they have the ability to create events, some interviewees would like to see how many people signed up for their events in the past week or month. "It would also be nice if we can see how many students are active in the association: e.g. how many different students signed up for our events in the last month."

Our interviewees connected to external parties have slightly different statistics that they would like to see. They are mostly interested in how many people entered their location in the past month, and how many of those people are unique to see the part of visitors that is recurring often. "I would like a graph to see that, but it should also show when people visit our location. This way we can be smarter in when we should be open and when it is possible to close."

4.2. Problems

In this section, we will discuss the problems we have found with the current state of affairs. We have found these through our own experiences as users, through a survey sent to students, through four interviews with several (board) members of sports and culture associations, and through regular meetings with our client (CFM). More specific information on these interviews can be found in Chapter 4. We will note between brackets where we heard about the problem or limitation. We will separate these problems in three categories. One for functionality, one for design and user experience, and lastly one for safety and privacy.

4.2.1. Functionality

1. Some students have trouble logging in to the app. *[survey & own experience]*

2. Associations can do member administration, but this does not fulfil several needs. Because of this, associations are forced to use multiple systems in order to do their administration correctly, which in turn causes a lot of duplicate and outdated data. *[interviews]*
3. A student or employee can accidentally request to be a member of an association multiple times, which causes them to receive multiple rejection messages even when they are accepted by the association. *[interviews]*
4. It is impossible to buy a Union Card as a new member since you have to log in to buy one. However, you can only log in if you (used to) own a Union Card. *[interview & own experience]*
5. Users cannot see what activities they can do with a Union Card, without first buying one. *[survey, client & own experience]*
6. The Campus Facility Management, the Student Union, associations, and external parties cannot see any information about how many students enter their location physically with the Union Card as an entrance ticket. *[client]*
7. The Campus Facility Management is not satisfied with the current possibilities to see insights into the amount of Union Card scans and other statistics that would be helpful and does not have control over the data gathered. *[client]*

4.2.2. Design and User Experience

8. It takes a long time for students and employees to show with the app that they own a Union Card, so that they can enter a facility or get a discount. *[interviews & own experience]*
9. The culture part of Sports & Culture is not represented well in the current app. *[interviews & client]*
10. One needs to be logged in, in order to see what activities are available. This decreased the visibility of events on campus. Because of this, students often do not see what they 'miss' and are therefore less likely to buy a Union Card. *[client]*
11. It is hard to see what activities take place around both campus and other external locations when logged in, therefore users may not be stimulated to try out new activities. *[client & own experience]*

4.2.3. Safety and Privacy

12. Data leaks have been recorded, and therefore the privacy of users have been compromised. *[survey, client]*
13. An admin of an association, such as the secretary, can delete themselves, and therefore an association can become admin-less. *[interviews]*
14. Union Card ownership can be faked. *[own experience]*
15. People have been able to get admin rights of other associations, which is a security vulnerability. *[survey & own experience, client]*

4.3. Our Solutions

During the development of the app, we will have to prioritize certain problems. A list of the problems we will focus on, with their respective solutions, can be found below.

In consultation with our stakeholders and after analysing our intial interviews, we have decided to prioritize the problems that follow the topics like improving the user experience, the ticketing system and the reservation system.

4.3.1. Improving User Experience

With the current Sports & Culture app, app users of both Android and iOS can only log in when Chrome is the default browser. Other browsers, like Firefox, lead to unexpected errors. We would like to improve the login functionality by making sure it is possible to log in using any preferred browser.

When users finally log in, they are directed to an empty event and activity table. To see activities and events, a filter must be applied first. This is unwelcoming and decreases the visibility of events and activities that one can join. Since the Campus Facility Management of the University of Twente would

like to stimulate campus activities, we would like to improve the landing page with a new design showing activities on the home page, even for people who are not logged in.

4.3.2. Improving Ticketing system

Currently, users have to download the Sports & Culture app to scan a QR code in order to prove they own a Union Card. If they do own a Union Card, their screen will turn green to signal that they indeed possess a Union Card. This system has a few drawbacks. Firstly, this mechanic forces students to download the app and log in. This is a time-consuming process that annoys many users who do not even want to download the app in the first place, as found in the survey. The technical issues surrounding logging in exacerbate this problem.

Secondly, by making a user scan the QR code themselves, the system is open to malicious users who are able to build an application that mimics scanning a code and then showing a green screen even if they do not own a Union Card. This harms the trust that external clients and organisations have in the system, and furthermore decreases the 'added value' of owning and paying for a Union Card.

To address these issues, we propose a system that does not require students to install the app. Union Card and reservation checks can be done by associations and external parties by scanning a student card with NFC technology or a QR code with the camera. By offering both options, we make sure that downloading the app is not a requirement for making use of Union Card extras. If the student forgot their student card, we want to offer the possibility that the student can still prove ownership of the Union Card by downloading the new Sports & Culture app. This app is connected to their university account and can therefore function as the Union Card itself by generating a QR code of their personal Union Card. They can then show this code to associations and external parties so that they in turn can scan with their app interface. This will make sure students can enter the activity, even though they forgot their student card.

4.3.3. Improving Reservation system

Currently, the Sports & Culture app does not show any information if you are not logged in. To log into the app, one needs to have already bought a Union Card. This makes it impossible for users without a Union Card to see any events. To increase visibility of all offered events, we do not require users to log in by default. If one wants to sign up for an event or do something that requires a Union Card, they do have to log in.

4.3.4. Improving visibility of Culture

Furthermore, even though the app is called Sports & Culture, the app currently does not list cultural activities by default, which decreases the discoverability. We want to list these activities in the same views as upcoming sports activities.

4.3.5. Adding insights

Insights are really wanted by organisations such as the Campus Facility Management, Student Union, associations and external parties. Right now, these parties do not have any insights into the data generated by the current app. An example of a use-case for this feature is the swimming pool. It is required by law to keep track of how many people are entering, in order to keep the right chemical concentration. Right now, this is done via a separate system. We would like to create a view for insights into this data.

5

User Stories and Requirements

Now that we have a plan, we will create user stories. By doing this, we are defining the workflows of our users, which in turn helps us keep an overview of all necessary functionalities of our app. After the user stories, we will create specific functional and non-functional requirements.

5.1. User Stories

The following user stories are created with the help of meetings with our client and interviews with associations, sorted by the categories: card checking, activities, statistics and management.

Student or employee at the University of Twente

Card checking

- As a student or employee, I would like to prove ownership of my Union or Campus Card, so that I can get access to the location or activity, or that I can get a discount for the location or activity.

Activities

- As a student or employee, I would like to make a reservation (e.g. for a tennis court, or a spot in the fitness hall).
- As a student or employee, I would like to see all activities on campus.
- As a student or employee, I would like to see all activities that I can join with my Union or Campus Card

Student Union (SU) or Campus Facility Management (CFM)

Statistics

- As the SU or CFM, I would like to see statistics showing where Union or Campus Cards are scanned.

Management

- As the SU or CFM, I would like to be able to add an association to the system.
- As the SU or CFM, I would like to be able to add an external party or location, so that they can check for a Union or Campus Card.
- As the SU or CFM, I would like to be able to delete an external party or location, so that events limited to a period of time are possible.

Association board

Activities

- As an association board, I would like to create events.
- As an association board, I would like to be able to check whether a certain visitor has really singed up for an event that I created.

Statistics

- As an association board, I would like to see the number of members over time.

Management

- As an association board, I would like to see who are my members.
- As an association board, I would like to see who of my members owns a Union or Campus Card.
- As an association board, I would like to add and delete members of my association.
- As an association board, I would like to assign tags/roles to people in my association.
- As an association board, I would like to see which membership requirements my association members fulfil.

External party

Card checking

- As an external party, I would like to be able to check whether a visitor owns a Union or Campus Card, so that I fulfill my part of the contract with Student Union or Campus Facility Management.

Activities

- As an external party, I would like to create spots that Union or Campus Card holders can reserve, so that I can share parts of the facility that I can offer.
- As an external party, I would like to be able to check whether a certain visitor has really made a reservation.
- As an external party, I would like to create events.
- As an external party, I would like to be able to check whether a certain visitor has really singed up for an event that I created.

Statistics

- As an external party, I would like to see how many people visit my facility.

5.2. Requirements

Based on these user stories we have made a list of requirements. Our client has agreed to this list of requirements. We have ordered them according to the Moscow method (Must, Should, Could & Won't). The must-haves are non-negotiable needs for the app that we will build. Without these, we won't have a working product that can satisfy our client. The should-haves are not essential for the product, but would add substantial value and functionality. The could-have requirements are not necessary to the core function of the product as they have a much smaller impact on the outcome of the product compared to the should-haves. We will end with the won't-have requirements, which are the requirements that are unfortunately not within the current scope of the project.

Before we list our project's requirements, we have some terminology that needs an explanation on what it means in our scenario.

Term	Definitions
Activity	An activity is a collective name containing both events and facilities.
Course	Some organisations offer courses like learning how to play guitar or improving your tennis skills. Courses usually have a certain amount of lessons included and are finished after that. Users can buy these temporary courses for the time they take.
Event	An event is something a user can sign up for (e.g. intro training, music show or dancing lessons).
Sign-up	Users can sign up for an event to participate
Facility	A facility is a location that users can reserve (e.g. a tennis court, a spot in the fitness hall, or a gaming computer).
Reservation	Users can make reservations for facilities to reserve a spot at a specific time
Subscription	Some organisations offer subscriptions like group lessons for a year, or fitness hall access for half a year. Users can then buy subscriptions like these.
Ticket	Whenever a user signs-up for an event or reserves an activity, the user gets a ticket integrated within its personal QR code to be able to prove its sign-up or reservation.
QR code	Every user has one personal QR code linked to its account. This QR code is a substitute for people without a student card, or students who forgot their student card. Associations and external parties will be able to scan this QR code, to check for a Union Card or a ticket for a certain activity.

5.2.1. MoSCoW Functional Requirements

Must-have requirements

- A student or employee must have the ability to make a reservation for a facility.
- A student or employee must have the ability to sign up for an event.
- A student or employee must have the ability to see all possible activities.
- Campus Facility Management must have the ability to create activities.
- A student or employee must be able to log in using a student/employee account.

Should-have requirements

- A student or employee should have the ability to see all events for the upcoming two weeks on the home page, in order to be able to discover new events.
- A student or employee should have the ability to show a QR code that can be scanned by an association, external party or Student Union to prove their ownership of a Union Card or a reservation.
- An association, external party or the Student Union should have the ability to scan a QR code to check whether a student or employee owns a Union Card, has a reservation, or has signed up for an event.
- Campus Facility Management should have the ability to see statistics about the usage of all Union Cards.
- Campus Facility Management should be able to see statistics about the amount of sign-ups for activities.
- Campus Facility Management should have the ability to modify existing activities.
- Campus Facility Management or Student Union could have the ability to cancel any activity.
- Campus Facility Management should be able to add an association to the app.
- Campus Facility Management should be able to add an external party to the app.
- A student or Employee should be able to see all events and facilities without logging in.

Could-have requirements

- The QR code could have an expiry date and time.
- Associations and external parties could have the ability to create activities.
- Associations and external parties could have the ability to cancel their activity.
- Associations or external parties could have the ability to see statistics about their association or organisation.
- Campus Facility Management could have the ability to revoke an external party's ability to create events.
- Campus Facility Management could have the ability to add subscriptions/courses.
- Students could have the ability to prove ownership of a Union Card using their student card with NFC (instead of the QR code).
- Associations or external parties could be able to add an account to their association or organisation so that these members can scan QR codes too.

Won't-have requirements

- The database won't have a connection with the current S&C database for true insights.
- Students and employees won't be able to scan QR codes themselves to prove ownership of the Union Card.
- Students and Employees won't be able to purchase courses, pay association fees and buy subscriptions through the app.

5.2.2. MoSCow Non-Funtional Requirements

Must-have requirements

- Students and employees should be able to sign-up to an event within 5 clicks.
- Students and employees should be able to reserve a facility within 5 clicks.
- The app must have a modular architecture that allows for easy maintenance and updates.
- The app must be compatible with iOS 16, 17 & 18 and Android 13.0 and 14.0.
- Every API endpoint must only be accessible by users who are authorized to do so.
- The app must be deployable using a CI/CD pipeline.

Should-have requirements

- Students and employees should be able to show their student card or the QR-code on the app when someone asks them to show their Union Card.
- The app design should follow the house style guidelines of the University of Twente.
- The server should have a latency of less than 500ms for any endpoint.
- The app should be able to handle at least 1,000 concurrent users without any performance degradation.

Could-have requirements

- The app could comply with WCAG 2.1 Level A accessibility guidelines.

Won't-have requirements

- The app won't have to comply with WCAG 2.1 Level AAA accessibility guidelines.

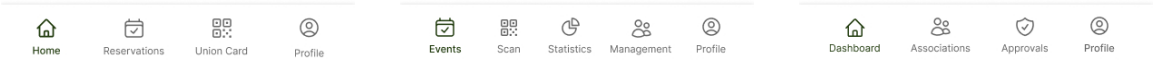
6 Initial Design

After our interviews and gathering all requirements, we started with our first design of the system. In order to show a UI design for the front-end, and to be able to discuss this with our stakeholders, we developed our first prototype in Figma, see also figure 3.2. In this chapter, we will discuss certain aspects of our initial design and its functionalities. Afterwards, we will also discuss our supporting back-end design and go into detail about our database design.

6.1. Profiles & Navigation

The first problem we faced was that our stakeholders all have different needs from our platform. The association boards need to have control over members and activities, while students need the ability to buy union cards and register for activities. Campus Facility Management and the Student Union need neither, but would like to have insights regarding these activities and the possibility to approve these. During our interview process, we received feedback that the Sports and Culture app currently in use is too 'cluttered' with functions that most users do not not need.

In order to create a design that is less cluttered, we have decided on a specific view per 'role'. Authorized users are able to switch their role in the profile page. When they switch roles, the buttons in the bottom navigation bar update and reveal different functionalities. In the Figures 6.1a, 6.1b and 6.1c, it is shown which buttons are displayed for each role.



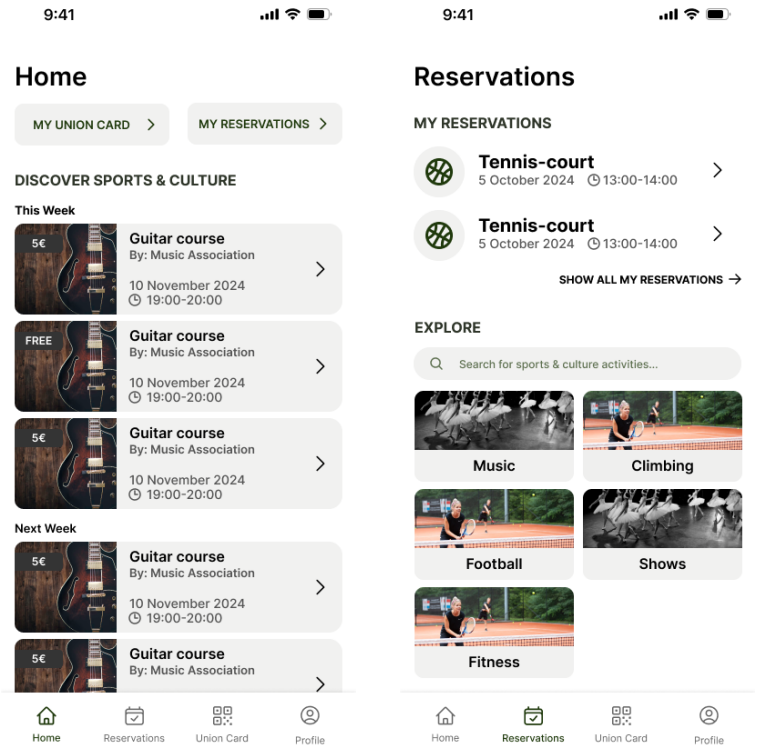
(a) Student / Employee Bottom Bar **(b)** Association / External Parties Bottom Bar **(c)** Management (CFM / SU) Bottom Bar

Figure 6.1: The bottom navigation bar for our different profiles.

We decided on this design because we thought it would be the most intuitive, making it feel like you have three applications in one, depending on your role, while not causing additional clutter for people not possessing a certain role (e.g. a normal student will never see all the management buttons).

6.2. Home

When our app opens, it starts on the home page. Our home page design can be seen in Figure 6.2a. This is a page for students and employees to view everything relevant to them. It features two buttons at the top, 'My Union Card', which routes to a page showing a QR code of your union card, and 'My Reservations' which will show the user a list of their reservations. We believe these are two often-used functions of the app, so we put them at the top of the default page. The rest of the page is dedicated to different recommendations. We designed it like this because CFM indicated that they want activities to be more discoverable. They want people to know the value of having a Union Card instead of it being a necessity to join an association.



(a) Home page **(b)** Reservations page

Figure 6.2: The Home and Reservations page in our initial design

6.3. Reservations

Another important feature for our stakeholders is the reservation system. The Reservations page should fulfil three main objectives:

1. Show the user their existing reservations
2. Help the user discover and make new reservations for activities
3. Allow the user to search all possible activities in the system

To achieve these objectives, we have split the page into two sections, as shown in Figure 6.2b. The first section shows two upcoming reservations, to give users a quick option to find the information they need. We have limited this to a maximum of two reservations, to make sure the page does not become too cluttered. If people do have more than two upcoming reservations, they can choose to click the 'show all my reservations' button. The second section provides a category-based browser for searching available activities.

Category Cards

To help users find new activities of interest, the available activities are organized into category cards. This design was inspired by platforms like Spotify, where users browse through music categories. The idea behind this structure is that users are generally only drawn by a few activity types. By grouping activities into relevant categories, the system encourages exploration while keeping the options manageable.

For example, a user with a specific interest in sports might be more inclined to browse categories like Cue Sports or Fitness. While a user who enjoys cultural activities would browse categories like Ballet or Shows.

Example: Tom owns a union card and wants to explore campus activities. On the reservations page, he notices the category 'Cue Sports.' Since he enjoys playing pool with friends, Tom clicks the card and discovers he can book a pool table directly through the app.

Searching Activities
While category browsing is helpful for casual discovery, some users might have a specific activity in mind. For this, we have a search bar that allows users to search for activities by name or keyword. This makes the reservation process faster and more efficient for users who already know what they're looking for.

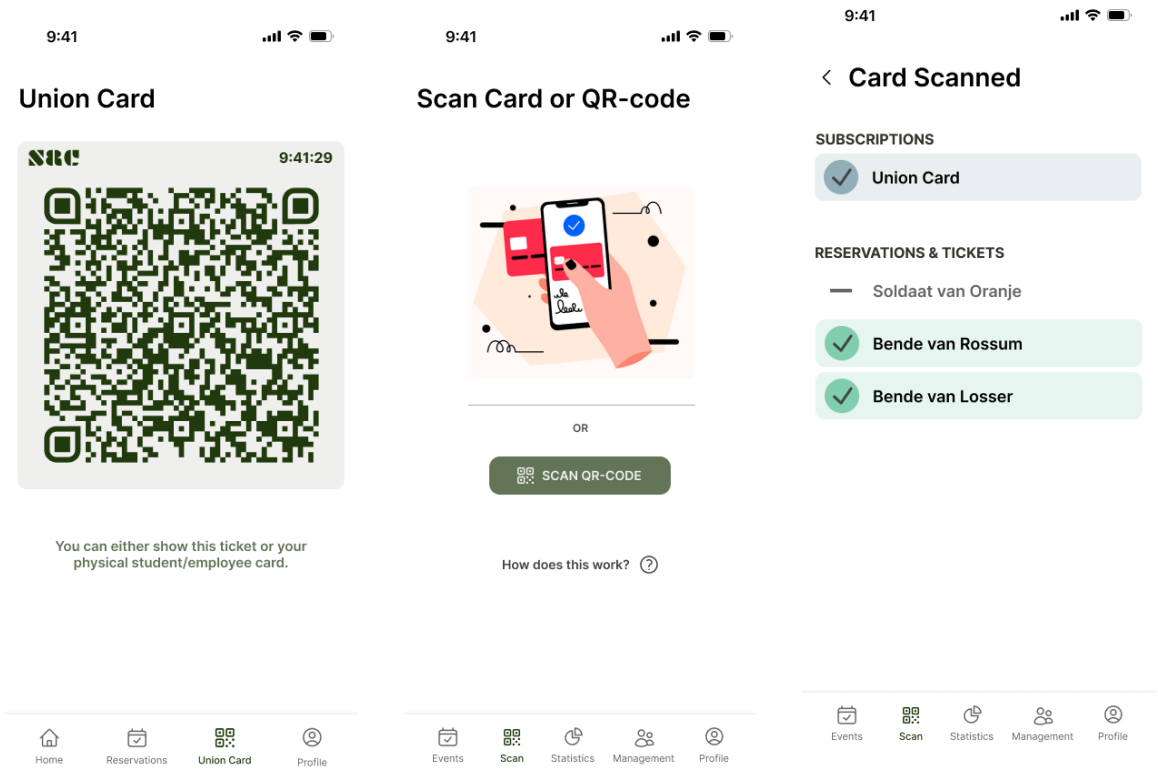
Example: Alex is a frequent gym-goer. When he needs to reserve a gym spot, he simply types 'gym' into the search bar. The system shows the relevant activity, allowing him to book a spot easily.

6.4. Scanning

As described in Chapter 4.1.2, the university's current app has shortcomings regarding checking membership and reservation status. The main issue with the current system is that scanning is done by the student, which is then shown to a ticket inspector. Not only is this system easy to fake and manipulate, but it is also unintuitive and complex for students. To resolve these issues we reverse the system by providing a QR-code to the student and the option to scan and check to the inspector.

On the app, a student is provided with a QR-code ticket which is connected to their account (Figure 6.3a). This QR-code can be scanned by an inspector on the scan page in the association view (Figure 6.3b). Once someone has scanned either the QR-code ticket or a student/employee University of Twente card, the app will show if the person has a Union Card subscription and if they reserved a sport for an event or facility.

We want it to be as easy as possible to quickly see if a person has reserved or signed up for an event or facility, therefore we only show the activities that the scanning organisation organises. In this way, you do not see any information you do not need. However, we do show events the person did not sign up for. With this design, the resulting page only changes in colours between multiple scanned students. This allows the inspector to quickly see at a glance, without having to read, if a person is allowed in or not Figure 6.3c).



(a) Union Card page with QR-code for students

(b) Scan page for associations

(c) The result of scanning a student card or QR-code with our app

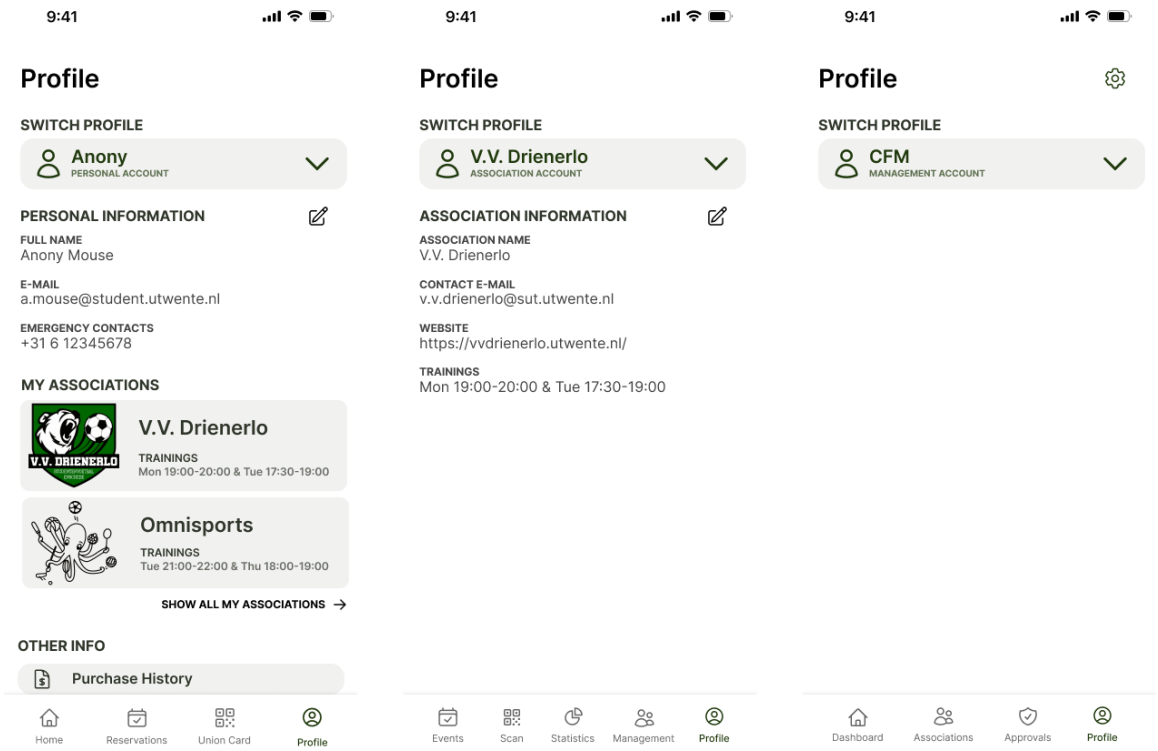
Figure 6.3: Showing and scanning a QR-code or student ID

Additionally, we extend this system with the ability to scan physical student IDs as well. Most modern mobile phones have NFC scanners built into them, and the university already has the necessary data to identify which student ID belongs to who. This feature provides 2 additional benefits to the system. Firstly, students do not even need to install the app to show they have a union card, bringing your student ID is good enough for this purpose. Secondly, it simplifies and streamlines the scanning process even further, as now it is as simple as holding the student ID near your phone, seeing the result, and scanning the next card.

6.5. Profile

The profile page, as show in Figure 6.4a, is the page where the user can see and edit their personal information. This includes their first and last name, their email, their emergency contact, and the associations they are a member of.

The page also displays their role, which includes student or employee for normal users, and possible board functions for students who are a member of an association board. It could also include a management account for CFM and SU users. These roles are displayed as a drop-down, allowing users to select which role they want to use in the app at this time. We have chosen to put the role-selector here, as roles are tightly coupled to your profile so this is the most logical place. We chose a drop-down because it integrated well into the design, and the little arrow, that only displays if you have multiple roles, invites users to click on it to find out what it does. Switching between a profile will result in switching to that view, if you switch to an association account you will see a page similar to the page displayed in Figure6.4b and if you switch to a CFM profile a page similar to Figure 6.4c.



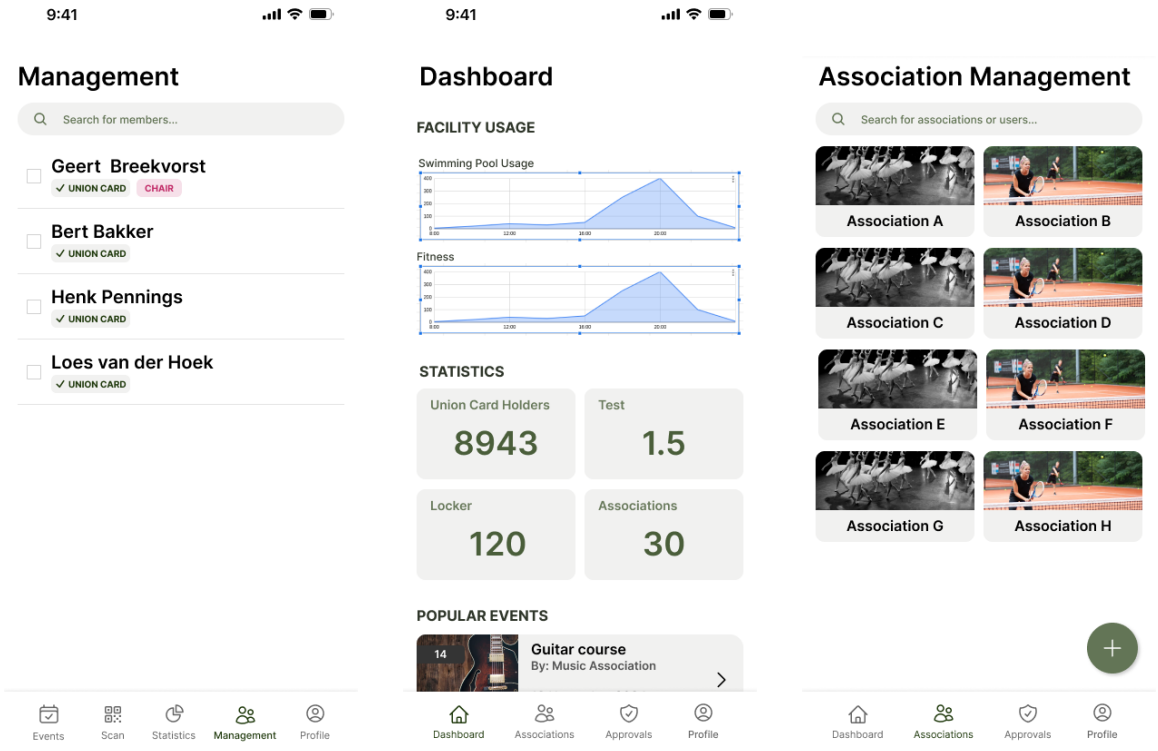
(a) Student / Employee **(b)** Association / External Parties **(c)** Management (CFM / SU)

Figure 6.4: The profile pages for the different profiles within the app.

6.6. Member management

On the association side, member management is important for our stakeholders. Currently, a lot of associations use many different systems to manage their members. In order to make it possible for the association to configure board/chair members themselves we created a tag system. The tags are

displayed below the name of the user as can be seen in Figure 6.5a. Tags can be added to users to make them a member of the association or to give them certain permissions in the association like scanning union cards or creating events. Additionally, associations can check if members have bought their Union Card subscription.



(a) Management page for associations **(b)** CFM view for statistics **(c)** CFM view for association management

Figure 6.5: The Association Management view, and 2 CFM views

6.7. Statistics (CFM)

During one of the meetings we held with our stakeholders, they informed us that having statistics (about, for example, the usage of the outdoor pool) would be quite helpful to provide better service. We have made a dashboard with some statistics for the Campus Facility Management employees and Student Union, see Figure 6.5b. Relevant statistics were not expressed, so our initial design features two graphs, 4 statistics cards and a list of the most popular events. The graphs show the amount of reservations for the pool and fitness room of the previous day. The so-called statics cards show certain things like the amount of current Union Card holders and the amount of associations. The popular event list shows, like the name implies, a list of the most popular events the past month. We choose these statistics as they seemed to most relevant to us, but we are aware of the fact that we do not exactly know what kind of statistics are relevant to the SU or CFM. The statistics cards are designed to give a quick insight into the big important numbers, with a concise title and the number shown prominently below the title, allowing users to see these numbers at a glance.

6.8. Association Management (CFM)

In addition to this dashboard we also design a page to show an overview and manage all associations. In Figure 6.5a you can see this page. Since the university has a significant amount of association we opted for a list where we display 2 items per row, and a search-bar to make CFM quickly find associations.

6.9. Technical System Design

The app will be supported by our web API, using the Spring Boot framework and a Hibernate PostgreSQL database. For our front-end, we chose to use the React Native framework, due to its support for both iOS and Android. This framework theoretically allows us to convert the app to a website as well, but we decided that was out of the scope of the current project. In the following subsections, we will discuss our database design and authentication structure.

6.9.1. Database Design

Several entities in our system need to be modelled in our database for the proper functioning of the app. An overview of these entities and their relations to each other can be found in Figure 6.6.

Association & Tag

One of our core entities is the Association, representing both associations within the university and the external parties associated with the university. An Association has many members (Users) with Tags controlling the permissions of its members within the organisation. An Association can also organise Activities like intro trainings or open events.

User & Role

The User represents any person who logged in to our application and tracks some basic information about them, such as their university email address, name, emergency number, and student ID number. A User can sign up for activities organised by associations or reserve a spot at a facility. Roles are used to specify the general permission level on the back-end. For example, anyone without an association role or similar will automatically be `forbidden` to access association access points.

Activity & Reservation

Activities represent the events and facilities organised by associations and external parties. An activity always occurs in a selected timeframe and may have a price, or be restricted to union card holders. Organisers of the Activity can restrict the number of participants that can sign up. For reservable Activities, like tennis courts or the gym, possible Reservations are made when the activity is created, which can in turn be reserved by users in the app.

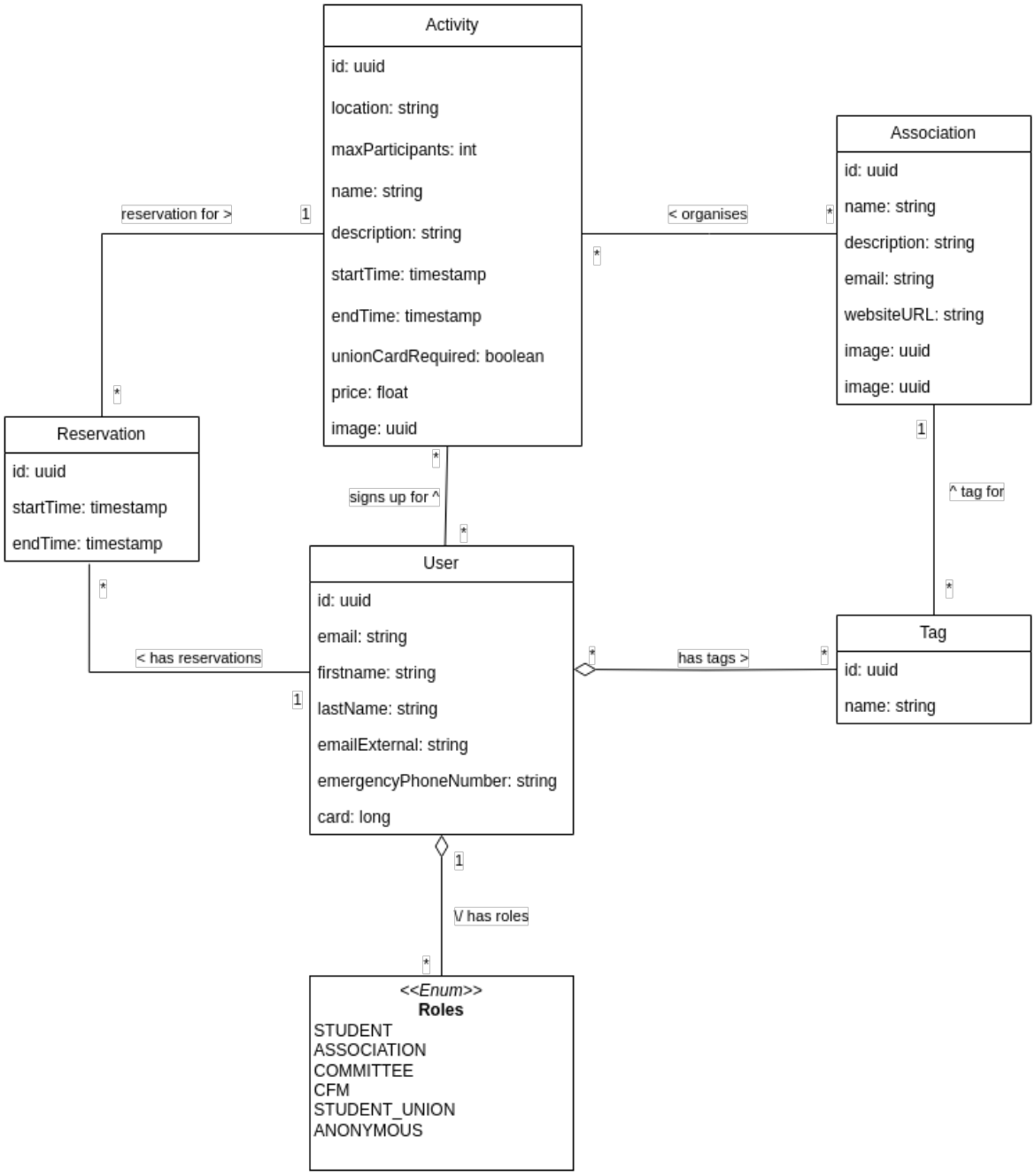


Figure 6.6: Initial model of our system

6.9.2. Authentication

We have decided to only implement authentication through Microsoft, as this is the sole authentication method used by the University. External users are out of scope for this project.

Following Figure 6.7, when a user wants to sign in, they are first directed to Microsoft's services. Once they have successfully authenticated themselves to Microsoft, our app receives an 'identity token' from Microsoft which contains the information of the user. This information includes their first name, last name, and email. This token is cryptographically signed by Microsoft, and is sent to our back-end for verification.

The back-end then verifies if the identity token is signed by Microsoft. This process ensures that tokens

not signed by Microsoft cannot be crafted by a bad actor. Then, the back-end does one of two things: it either creates a user if it has never seen them, or it loads their information based on their University of Twente email address. Finally, the back-end sends its own two tokens back to the front-end, we do this because the Microsoft are intended only for a single use. We have implemented our own refresh token system that allows users to be logged in for longer periods of time.

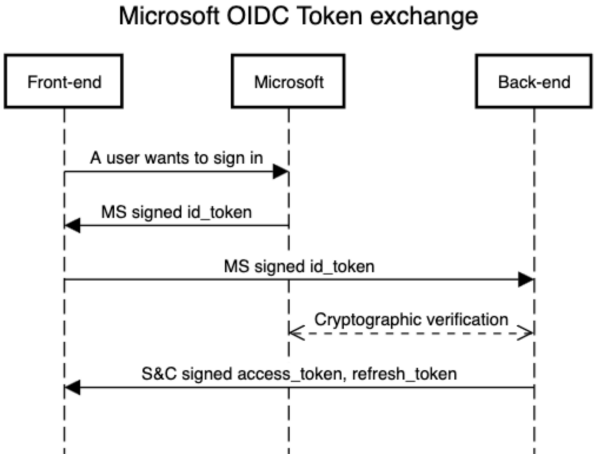


Figure 6.7: Authentication Flow with Microsoft

6.10. Feedback on our shown Figma design

In order to check whether people like this new design, we have asked six people to give feedback on our Figma design. We have gathered their feedback and structured them in positive feedback, and possible improvements.

6.10.1. Positive feedback

At first, the people we asked liked that they did not need to log in, or apply filters to see activities that are coming up. The buttons 'My Reservations' and 'My Union Card' on top of the home page were also appreciated.

In addition, people were very positive about the idea that association rights are linked to a student account, rather than having a completely separate account for each association. Most people liked the idea of everything being in one place: reservations, intro trainings and culture.

Finally, the integration of an NFC scanner into our product was well received by people we have interviewed who are related to an external party or association. They are very involved in the scanning of Union Cards.

6.10.2. Possible improvements

Some improvements were suggested as well. For example, some people noticed that the shop was missing. "How should I make use of Union Card facilities if I can not even buy one?". This is a valid point, however, the shop was out of the scope of our project due to several reasons discussed earlier. We decided to implement a non-functional shop following this suggestion.

Also, some people gave the idea to see how busy it will be at an event or facility. This way, they can try to find a spot when it is still relatively unoccupied. We decided to implement this feature into our app.

At last, one of the first people we interviewed noticed that it may be hard to switch back to a student view in our initial design, because the 'Profile' tab got a different logo and was called 'Organisation'. We thought that was a fair point and immediately implemented that in the initial design before the other user feedback sessions. No other person has then noticed that problem any more.

7

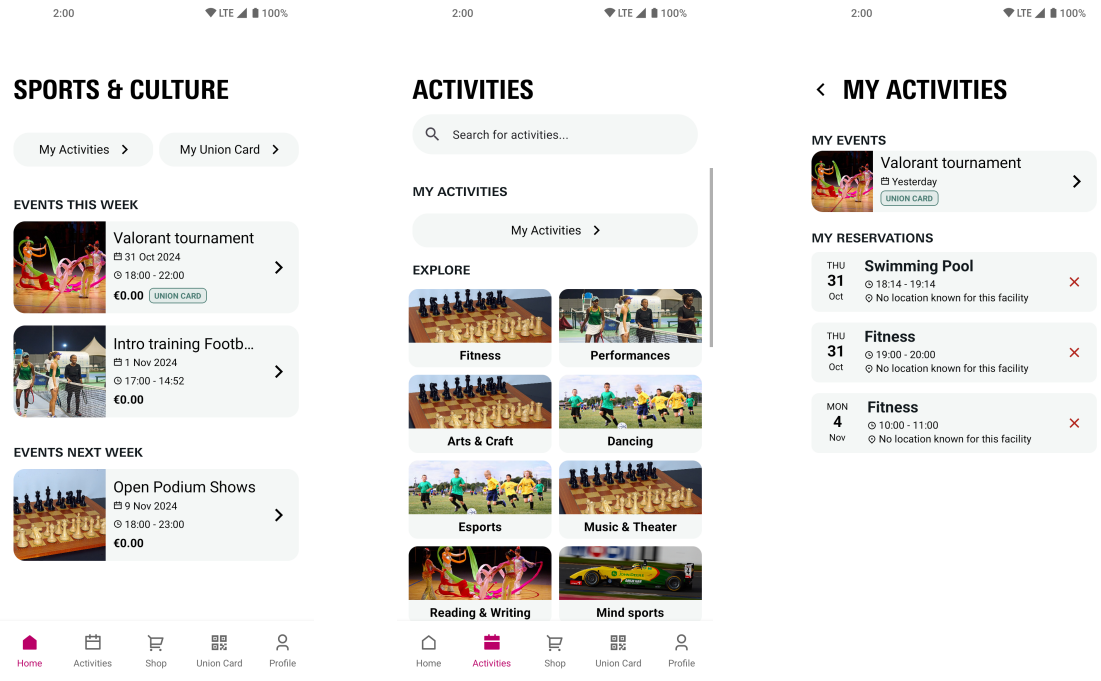
Prototype Development

After our initial design, we will start with the actual development of our system and with that, we enter a new phase in the development of this system. In Phase 2, we have built a working prototype based on the requirements and designs gathered and created in Phase 0 and 1. We have tested the prototype with our stakeholders multiple times during this development phase and integrated the feedback they have given.

In this chapter, we will discuss the changes we made compared to the initial Figma design and our reasoning behind them.

7.1. Front-end Design

Firstly, we will start with the front-end design choices. We will go through all the different functionalities and pages of the app one by one.



(a) Home page for students

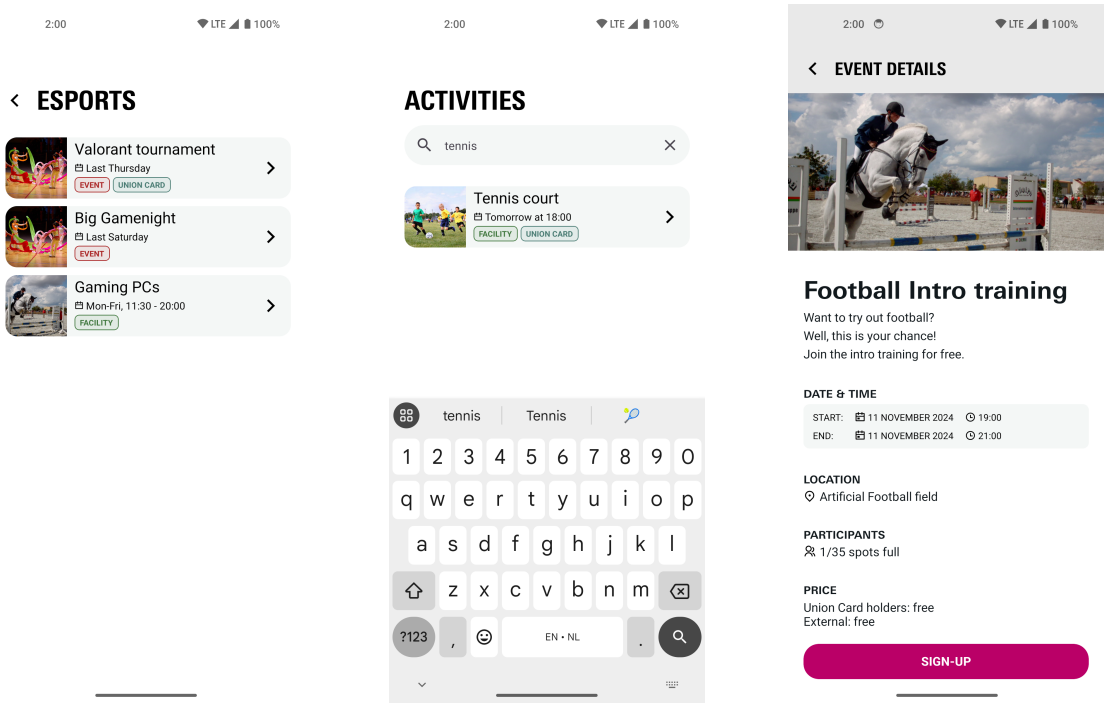
(b) Activities page for students

(c) My Activities page for students

Figure 7.1

7.1.1. UTwente House Style

The first change to the front-end design that we will discuss is that the UI now has a new font and uses a different accent colour. When gathering feedback on our initial front-end design, Campus Facility Management suggested that they would like to see a more UTwente-like style within the app, that is, the style that the University of Twente uses themselves. In our Figma design, as shown in Chapter 6, you can see a green accent colour in our design. To make it look more like a University of Twente app, we have opted for a pink colour. This colour is also used in the 'MYUT' app. An important part of the UTwente design language is the font, which we have copied from the UTwente style guide as well. This new design can be seen in all the screenshots in this chapter, a good example of this is Figure 7.1a, where we updated the home page incorporating the new design. We also adjusted the secondary colours in the app to fit better with the new primary accent colour.



(a) Activities category page

(b) Searching on the activities page

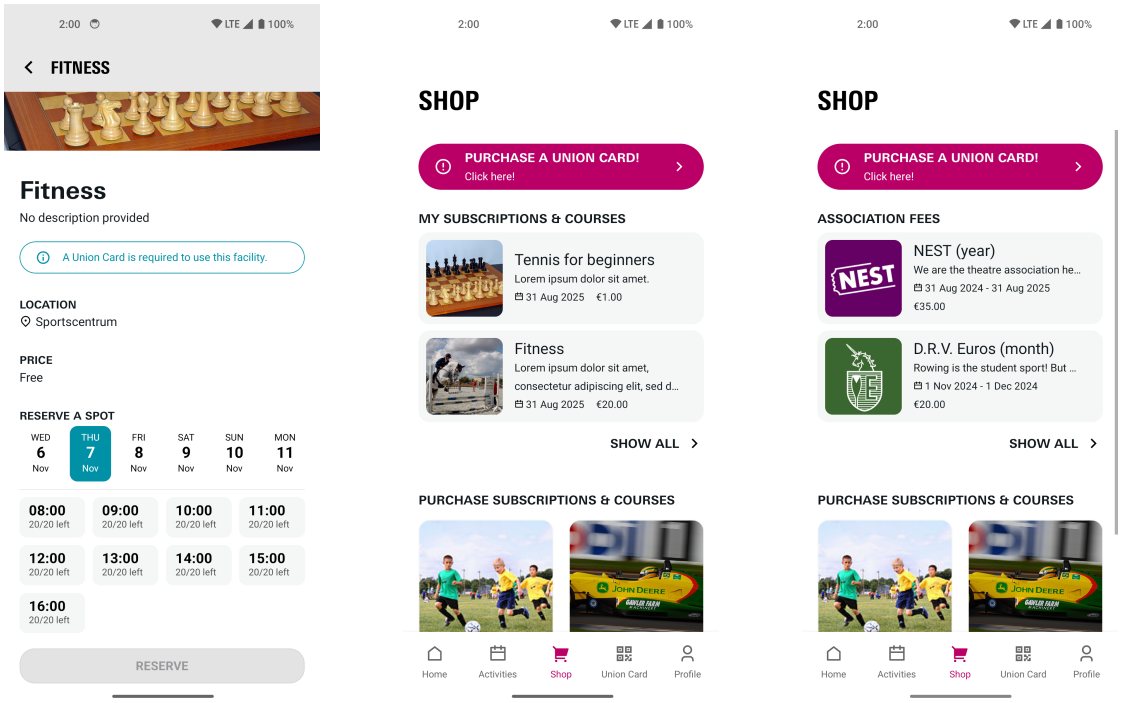
(c) Event details page

Figure 7.2

7.1.2. Activities and My Activities

During the feedback sessions, people informed us that the terms reservation and event were confusing since we did not always use them consistently. To combat this, we opted to use the term 'Activities' to represent events and reservable facilities such as tennis courts. We will discuss the back-end changes related to this in Section 7.2.1. In Figure 7.1b, you can see the new tabs we created for the activities page. On this page, we also removed the My Reservations section and replaced it with a single button called 'My Activities', which will redirect you to the page in Figure 7.1c. We have done this to reduce the amount of clutter on the page, a small version of the reservation would either not show a lot of info or would take up too much space, which makes you scroll to see the categories. If you click one of these categories, you will open a screen as displayed in Figure 7.2a, here you can scroll through all activities of that category.

In addition, we also changed the location of the search bar, pictured in Figure 7.2b. When someone starts searching, all the previous content of the page is cleared, and only events and facilities related to the search query are shown. To make this animation go smoothly, it was better to have the search bar at the top of the screen instead of on the page itself, so it stays in a fixed position while searching.



(a) Facility details page

(b) Old shop on the student side

(c) New shop on the student side

Figure 7.3

7.1.3. Event and Facility detail pages

When clicking on an event or facility, it is logical to show more information about the event or facility. On the event details page, as in Figure 7.2c, we display the start and end date and time, who organised it, the number of people participating and the price. Students are then able to enrol or un-enroll for this activity.

When trying to reserve a facility, we will show the page in Figure 7.3a, this page is similar to the activity details page, except for the calendar system in which you can select any day in the upcoming two weeks to reserve a timeslot and then reserve it.

7.1.4. Shop

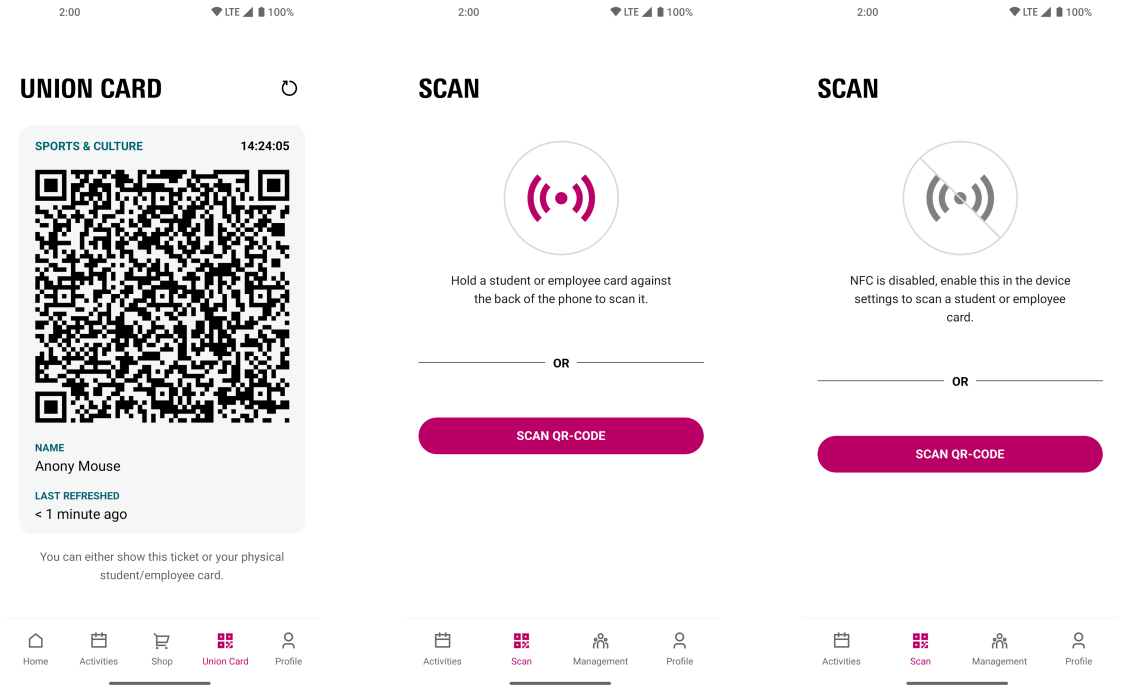
Since our client does not offer integrations with the current platform and we are developing a proof-of-concept. Integration of payment solutions and therefore a working shop is out of the scope of this project. Hence, during the initial design, we decided to not build a shop page. However, when we were showing people our design during later interviews, multiple interviewees reported missing the shop page. We then decided to only design the shop page, but not implement any features that would make it possible to make transactions.

During the design of the shop page, we have several priorities. We want it to be simple to find a way to purchase the Union Card since this was one of the main reasons people used the shop. We also want people to be able to check their currently owned subscriptions and courses.

To improve the flow to buy a Union Card we show a banner at the top to guide users to buy a Union Card, when they do not currently own one. In Figure 7.3b you can see our design for the shop with this banner, below the banner the shop shows your currently paid subscriptions and courses. The space below is reserved for discovery and exploration, showcasing two buttons users can click to explore either subscriptions or courses.

The feedback sessions gave feedback that users do not find it logical to have a section dedicated to subscriptions and courses they already own on the shop page. They navigate to this page in order to buy or explore new items, not to view items they already own. Another point of feedback was that they

were not able to find a way to pay the association fees. Hence we updated the design with these two points in mind, our new design, displayed in Figure 7.3c, shows these changes.



(a) Union Card page for students

(b) Scanning page for associations

(c) Scanning page when NFC is disabled

Figure 7.4

7.1.5. Ticketing System

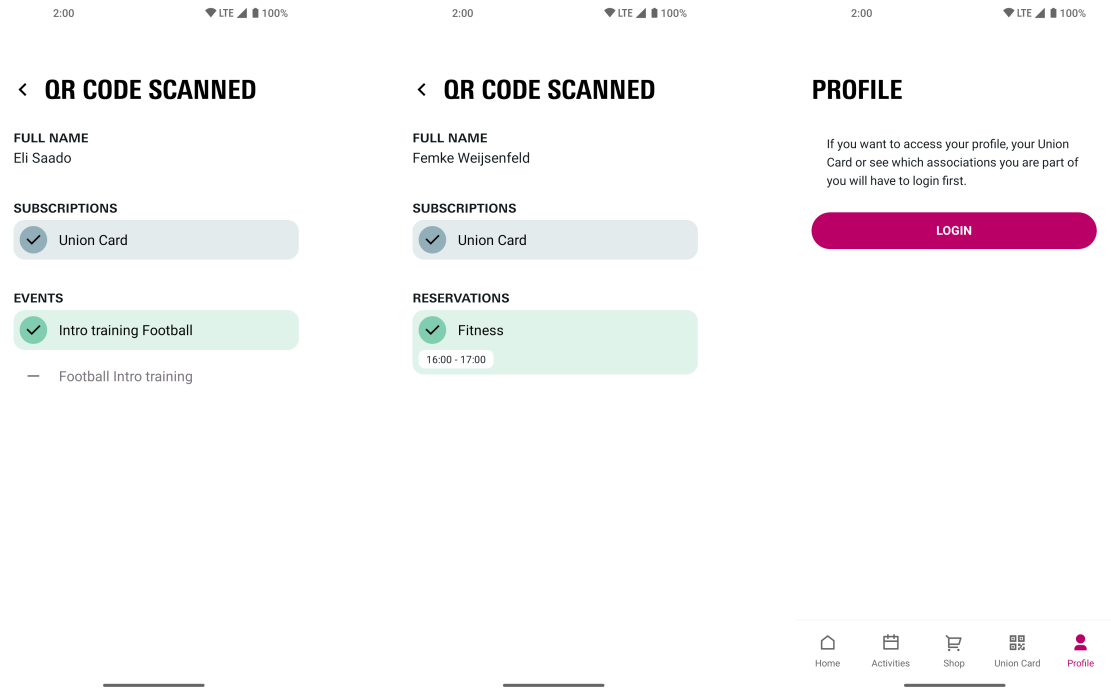
As discussed in Chapter 6, we use a combination of scanning physical student IDs with a QR code as a fallback. On the student side, the app will request a JSON Web Token (JWT) from the back end. This JWT contains the UUID of the user, which is a uniquely generated code for each user, until when the token is valid and the option to verify the authenticity of the token. This token will then be shown as a QR token on the phone of the user, as can be seen in Figure 7.4a. Someone else can then verify this token from the association side of the app to check if the user is allowed in this event by scanning this QR code.

To prevent people from sharing these QR codes, we have taken two security measures. Firstly, we limit the validity of the QR code to one hour. When you open the page in the app, the QR code will automatically refresh, so this should not be an issue for the regular user. The user can also always press the refresh button in the top right corner to immediately refresh the QR code if they experience any problems. Secondly, we also block users from taking screenshots when they are on the Union Card page. We do allow screenshots on the other pages to make sure users can share content from the app.

On the association side of the app, you will see a scanning screen, see Figure 7.4b. In the case that you have NFC disabled you will get an info screen (see Figure 7.4c) informing you to turn on NFC, if you want to scan cards. Once on this page with NFC enabled, you can choose to either scan student or employee cards or scan a QR code. When you have successfully scanned either of those, a screen similar to the screens in Figure 7.5a and 7.5b will pop up. This screen will show if the person has a Union Card subscription, all the events they signed up for, and the time slots they reserved at a facility.

7.1.6. Profile Page

We made our Profile Page almost exactly as discussed in the initial design. We kept the way of switching profiles, the profile information, and the overview of associations the same. You can see this design in Figure 7.6a. When you want to switch to another profile you can click on the drop-down, as shown in Figure 7.6b. We added the ability to ask to join an association on the profile page, under the complete overview of associations shown when clicking on the plus icon. We removed the other info section, as

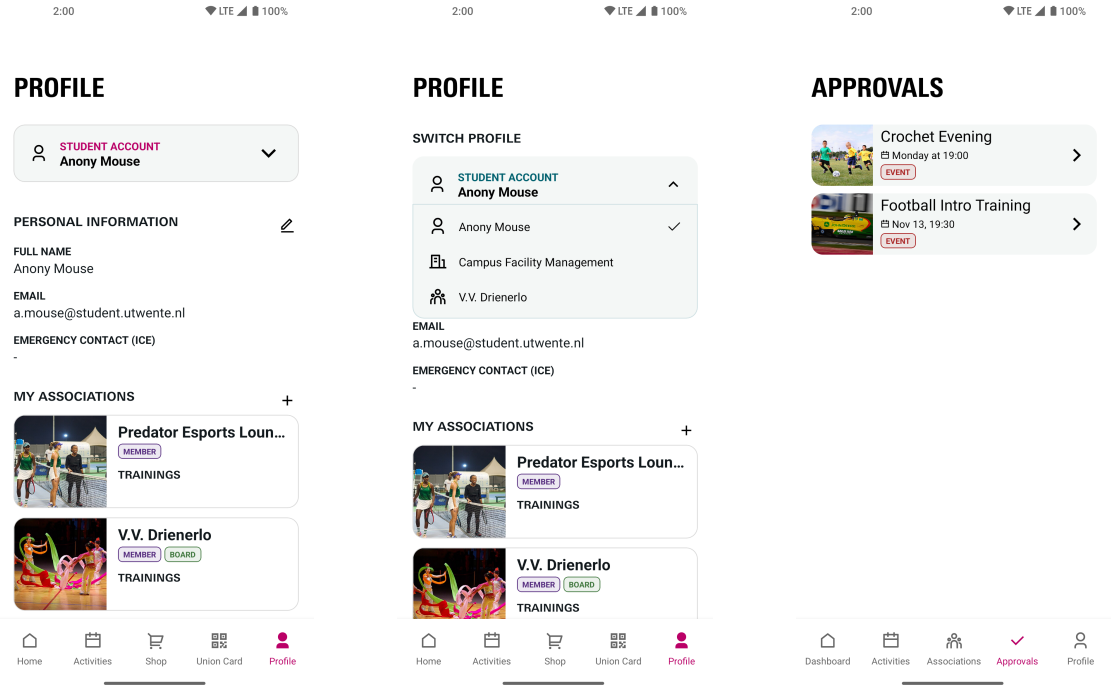


(a) Scanned page with 2 events

(b) Scanned page with a facility

(c) Profile page when you are not logged in.

Figure 7.5



(a) Profile page for students

(b) Profile page with menu expanded

(c) CFM's approvals page

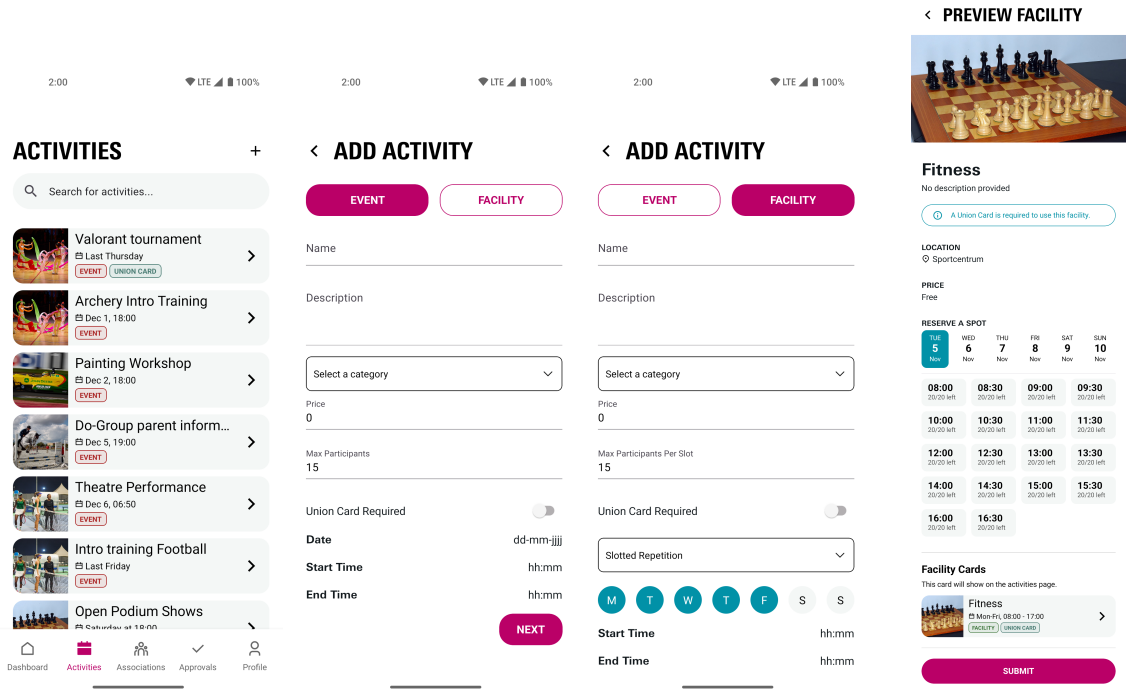
Figure 7.6

the only thing shown previously was the purchase history. Since we did not add any features regarding making purchases, we decided to remove that section entirely. We also made some purely visual changes to the drop-down, such as a border and turning the name pink instead of the default dark grey colour.

7.1.7. Content accessible without logging in

One important feature for our client, Campus Facility Management, is that the app should be accessible without logging in. This allows users without an account or without a Union Card to see what is possible.

We made the application as accessible as possible without an account, allowing users to see and explore activities. When they land on a page where authentication is needed, like their profile page or their Union Card page, a message is displayed saying the user needs to log in to view this page. Figure 7.5c shows how the profile page looks when you are not logged in. We try to limit the amount of pages where authentication is needed, to make the app as accessible as possible for unauthenticated users.



(a) Activities tab for associations **(b)** Create activity page (Event) **(c)** Create activity page (Facility) **(d)** Preview facility page

Figure 7.7

7.1.8. Creating Events

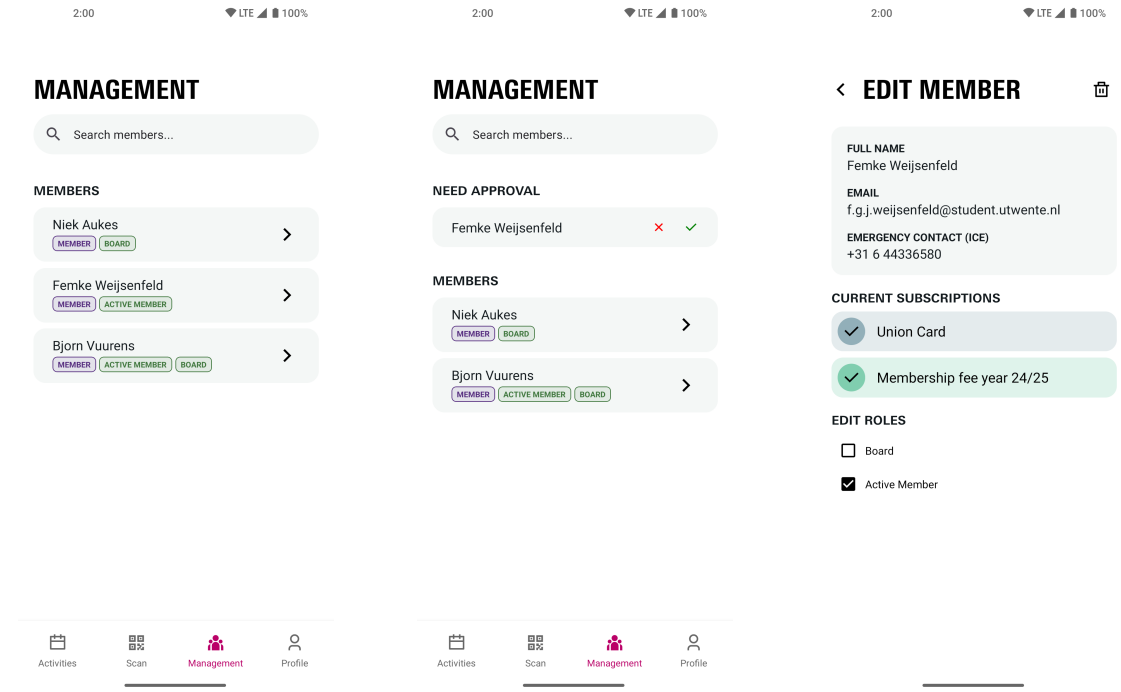
Associations and Campus Facility Management need to be able to see and create events and facilities in our app. For this, we have a set of pages users need to navigate through to create an event or facility. The association and management view both have an activities tab, pictured in Figure 7.7a. On this page, associations and management can see the activities they created and add new activities via the add button on the top-right of the page.

Clicking this button takes the user to the Create Activity page. The user can create either an Event (Figure 7.7b) or a Facility (Figure 7.7c). On these pages, details about the new activity need to be filled in that are necessary for activities. Once these details are filled, the user can click the 'NEXT' button to display a preview of what the event or facility will look like on the home, detail, and search pages. Figure 7.7d shows how this preview looks for a facility example.

Approvals

Associations can create their own events, but since all upcoming activities are prominently displayed on the homepage of the app, there needs to be some kind of final check to make sure all featured activities are up to standard. This is achieved by letting Campus Facility Management (CFM) approve every facility or event created by an association. They do this on the 'Approvals' page, as shown in 7.6c. On this page, CFM can see activities submitted by associations, and visit the preview page to make a judgement about the activity. An activity can be approved if it is deemed to be suitable, making it show

up on the home page among other pages and allowing students to enrol or reserve for it.



(a) The management page for associations

(b) Requesting to join an association

(c) Edit member page

Figure 7.8

7.1.9. Creating and Managing Associations

Associations are created and managed by CFM, this is done on the Associations page. On this page, shown in 7.9a, a list of associations is shown, with the ability to search through, delete and add associations. When you want to create a new association you can click the plus icon and a new page will show up, requiring you to enter an association name and the email of the board member, this will be the person who will be the first board member of this association. This design can be seen in Figure 7.9b.

7.1.10. Statistics

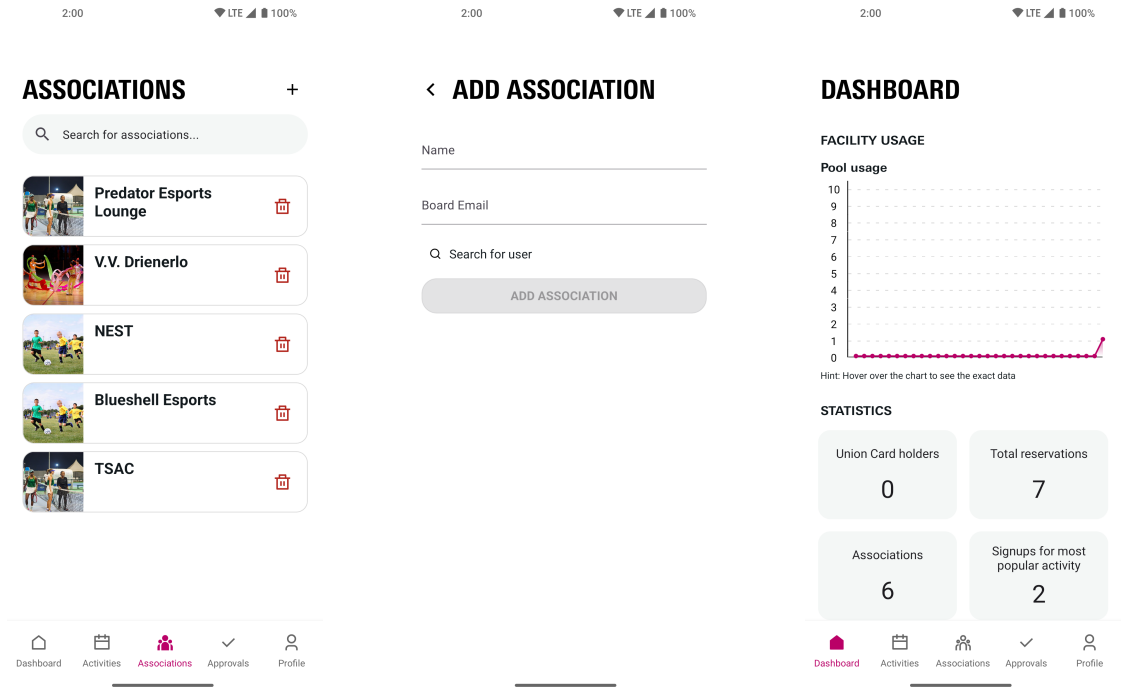
Our stakeholders informed us that having statistics about, for example, the usage of the outdoor pool would be quite helpful in providing better service. We have made a dashboard with some statistics for the Campus Facility Management employees and people from the Student Union responsible for managing Sports & Culture.

In this stage, this dashboard, as seen in 7.9c, functions as a proof of concept to show what is possible. It features a graph showing the usage of the outdoor pool last month, 4 tiles showing various statistics, and below that the most popular activity is shown. The 4 tiles show the total number of associations, the number of union card holders, the number of reservations this month, and the number of sign-ups for the most popular activity. The last statistic can be used to understand how popular the most popular activity was.

As mentioned previously this dashboard is merely a way to show that statistics are being collected, more statistics are possible since the data is already there.

7.1.11. Association Management

We have implemented the design as discussed in Chapter 6, this is shown in Figure 7.8a and 7.8b. In the first figure, you can see all the members of the association including their tags, the second image shows how the view will look when you want to accept new members. In this case, the people on the board of the association can decide if 'Femke' is allowed to join the association or not. The cards with the names and tags of the members are clickable and will bring you to the page shown in Figure 7.8c where you



(a) CFM page for managing associations

(b) CFM association creation page

(c) CFM statistics page

Figure 7.9

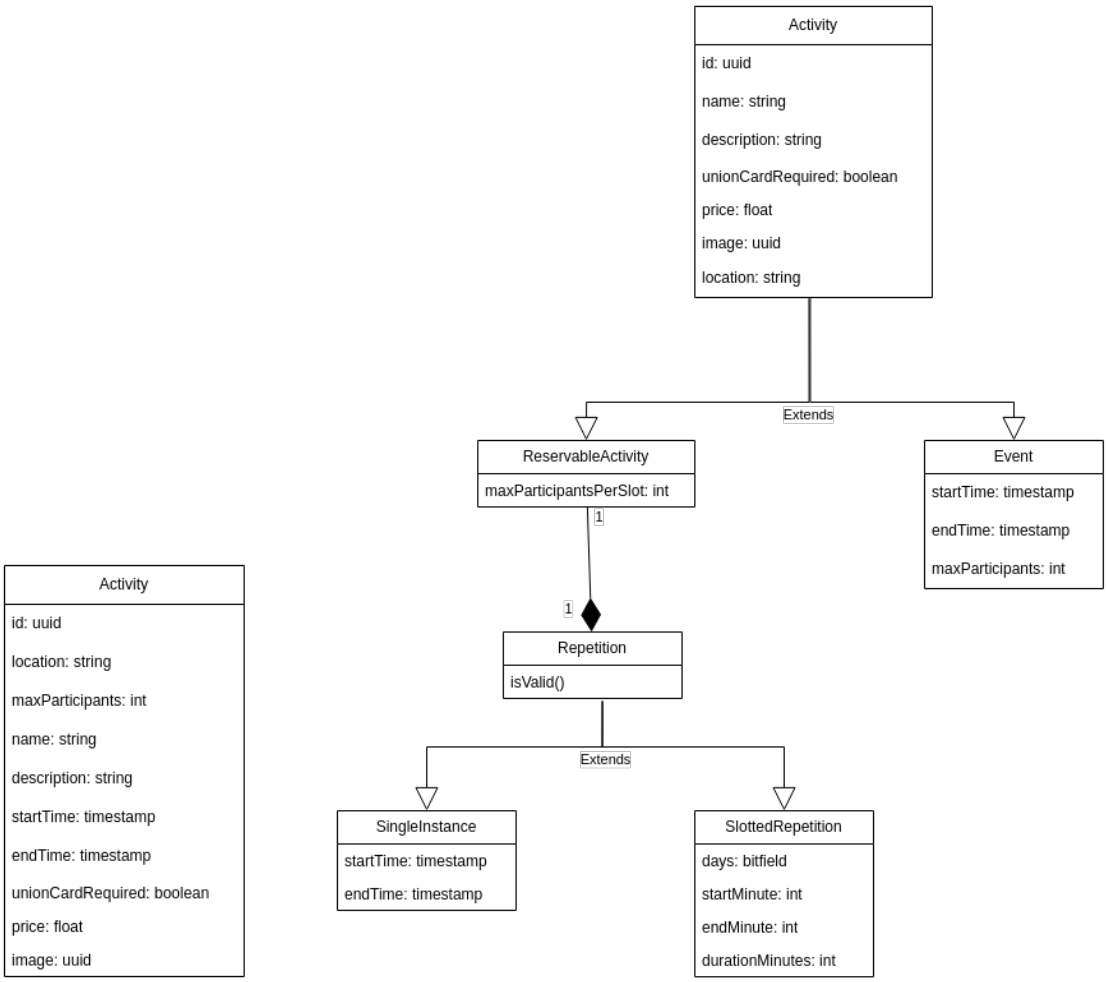
can view more info about the user, remove the user from the association or edit their tags.

7.2. Back-end Design & Modelling

In this section, we will discuss the changes we made to the back-end during our development process. We will first explain what we did before and what caused us to change our approach. A complete overview of our back end in the form of a diagram can be found in the appendix.

7.2.1. Activities

At first, we modelled every single event and activity in a single class, as can be seen in Figure 7.10a. During the actual development of the project and as found in interviews, this model got extended a lot since new features were asked for and thus implemented. At some point, this class became so large and cluttered with fields that most instances did not need it, so we decided to split everything regarding events up in different models. We decided to have a parent Activity class, and two children called Event and ReservableActivity that inherit from this class, see Figure 7.10b.



(a) Initial Model **(b)** Final Model

Figure 7.10: The changes in the modelling of Activities in our database

7.2.2. Repetitions

After interviews with Campus Facility Management and the culture coordinator of the University of Twente, we came to the conclusion that in order to support the culture side of the Sports & Culture app, we needed some more implementation details for a Repetition. We decided to support multiple types of repetitions to accommodate the different facilities available on campus. Facilities in our new database model can choose between a single instance repetition, with a single date and time, or a slotted repetition, with timeslots on a few selected weekdays. The system is thus designed to easily add new repetition types if that need arises. Figure 7.10b shows these changes to our model.

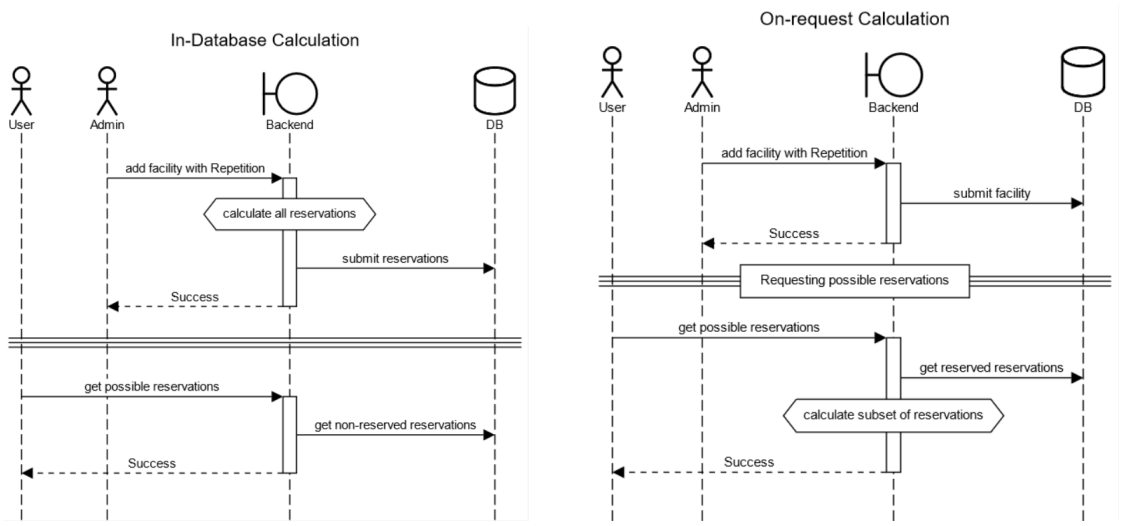
7.2.3. Reservations

Another critical design choice was when to calculate and store which reservation slots are still available to reserve. For a reservable facility, we can calculate the possible timeslots with its repetition details. This, however, is an expensive operation that could slow down our application significantly. Additionally, reserved timeslots should be shown as taken, adding even more complexity to the calculation. We have thought of a few possible implementations of this algorithm with different advantages and disadvantages.

Database cache calculation

The first implementation approach is conceptually very simple. When a new reservable facility is made, all possible timeslots are calculated at once and inserted into the reservations table as an unreserved

timeslot. This would effectively make the generation of timeslots a one-time operation, Figure 7.11a describes how this takes place.



(a) in-Database performs all reservation generation when a facility is created.

(b) On-request calculation performs the selected reservation generation on each request.

Figure 7.11

However, numerous issues arise within the details of this algorithm. Firstly, it completely disregards the perpetuality of repetitions. It is simply impossible to generate all possible timeslots because there are infinitely many of them. This means that either all repetitions should be finite, or that new timeslots be occasionally generated.

Another limitation of this method is the high storage requirements, a single activity could have as many as 3000 reservation slots per month, which would have serious effects on query speeds as our application scales up.

Due to these issues, we decided that this would not be the right approach for this algorithm.

On-request calculation

On the other end of the spectrum, we generate the available timeslots on a per-request basis. All possible timeslots are generated for the time range specified by the request, after which reserved timeslots are collected from the database and removed from the possible timeslots. This process is pictured in Figure 7.11b.

This method solves many issues mentioned in the previous approach. It does not involve adding table entries for unreserved slots and is locally bounded by request parameters. The disadvantage of this method is the performance hit of the timeslot generation.

A possible remedy is to memoize the timeslot generation and restrict it to only a single day. This could greatly improve the performance if it becomes an issue in our application.

7.2.4. Tag System

In order to support the ability to have multiple profiles as a person, we have implemented what we call our 'tag system'. In this system, a person with a board function can assign a tag to one of their association members. If one possesses such a tag, an extra profile for that association is added to their drop-down on the profile page, and they get the ability to scan tickets. Before our change, every board member could add a tag to their association, give it any type, and assign members to this tag. This type would, for example, be the committee name. In order to reduce complexity, we have decided against having multiple types of tags in the back end. Instead, a tag now has a name and is either a board tag, which also allows you access to event creation, statistics, and member management, or a committee tag, which allows you only access to scanning QR codes and student cards. This greatly reduces complexity in our back end, as we do not have to check for custom typing when trying to access a feature.

7.2.5. Automatic Hibernate Tables

In our project, we make use of Hibernate. Hibernate is an Object Relation Mapping framework for Java. In addition, Hibernate is also built upon the Java Persistence API specification. This means that it can automatically create a database for you based on the models you defined. It also makes custom SQL queries for you based on the naming of these queries. For example, findAllAssociationsByNameAndOrderByAscending would fetch all associations from the database and order them alphabetically based on the name of the association.

Because we have changed our models quite a lot during our development process, we have decided to quit making use of the automatic database creation feature. Instead, we have designed our own database. We have done this because Hibernate does not support migrations. This means that Hibernate is not able to change the tables of the database after it has decided what they 'should' be on initial creation and after adding data, even if your data structure changes. At some point, our database no longer reflected an efficient storage structure for our data and we therefore decided to design our own.

7.2.6. Refresh Token System

We have implemented our own JSON Web Token (JWT) authentication system, as discussed above in Chapter 6. These 'access' tokens expire after an hour. In order to support the ability to be logged in for more than an hour, we have implemented a refresh token system. This meant changing what we return in our authentication controller to the front end. Now, instead of returning only an access token, we also return a refresh token. If your access token expires, our back-end will return an error that the access token is expired. The front end will then catch this error and send a new request to the back end containing this refresh token. The back end then receives this refresh token and checks its validity. If this refresh token is valid, the back end will send a new access token to the front end to use in further requests. It will also generate a new refresh token to use when the access token expires again. Using this new system, users can be logged in for more than an hour without having to authenticate again.

7.3. Feedback Sessions

In order to get feedback we asked several people to give feedback on the app while developing. All this feedback was gathered in the middle of phase 2, so the app was not fully finished when we did the following feedback sessions. The screenshots in the previous section may already have shown a final version, which was not available at this time of writing.

7.3.1. Feedback from client

Firstly, our client, Campus Facility Management (CFM), provided feedback during the development process over different weeks, when we planned feedback sessions with them. At each meeting, we showed them the current state of the app we developed at that time. For example, we discussed with CFM the ideas regarding event creation by associations. They really liked that idea, because it stimulates student activism. As expected, they want to approve events created by associations first, so this became a new requirement.

In addition, our client proposed some other new ideas. For example, they need to know how many people are at the pool due to safety regulations. They would also like to see some kind of method to prevent people from making screenshots of their QR code, and they would like a share button to share events. These are all different ideas that we would like to implement during our development process.

Finally, we were given the feedback that culture shows or events, often organized by external parties, would like to see the option to support different prices for Union Card holders and non-Union Card holders.

7.3.2. Feedback from other users

Next to talking with our client CFM, we have also asked five people from different sports and culture associations for feedback. The moment of asking this feedback was halfway through phase 2. We let people walk through the app first, without any further information. This way we could find out their thoughts without an introduction. Then we asked them for their opinion.

Four out of the five people we asked for feedback thought that 'My subscriptions' in the shop was weird.

"Why is this in the shop if I have already paid for it?" They all expected it to be under the Profile Page. This let us ask the question: "Where would you like to go to My Activities then?" All five people we asked for feedback expected the My activities page to be under the Activities Page.

More feedback is given about the shop; people would like to have a search bar, to find what they need easily. In addition, the division between 'subscriptions' and 'courses' is unclear. Our five people were not able to tell the difference between them. Also, the people we asked for feedback think that it is unclear where to buy the association fee, which they consider important. However, also one positive feedback point was noted: most people really liked the pink 'Buy Union Card Here' button at the top of the shop. "It is the most important item in the shop in my opinion."

Also, feedback is given about our Profile Page at that stage of development in the middle of phase 2. Someone thought it was weird that you could edit your own name in the 'Edit Profile' part. "Just use the name you get from the university login, you should not be able to alter that." In addition, a suggestion was given that you could pay for the association fee when you tab on your association in the My associations" section of the profile page. "If that is not possible, I will be searching in the shop." At last, someone pointed out that he or she would like to be able to join an association in the app. We can do that easily by adding a '+' near 'My Associations'. The person who gave this feedback thinks that is a good idea.

At last, one person we asked for feedback was missing the calendar view in order to reserve a facility. That person thinks the current design is a bit confusing, and it does not show how the week looks like. We will test whether our design is intuitive later with user testing. An idea from our side to fix this would be to add the ability to synchronize the events people have enrolled for or the facilities people have reserved with a personal calendar, or just change to a calendar view if it is preferred by a big group.

Overall, people were very positive. Some associations pointed out that the addition of the ability to let associations scan for Union Card is very practical, especially for cultural associations who organise performances. Also, the idea that external parties could make use of the same functionality as associations was also well received.

All the feedback described above has been taken into consideration during the development process described in chapter 7.1.

7.3.3. New requirements

The feedback sessions from above resulted in new functional requirements for our app. We have ordered these functional requirements according to the MoSCoW method. These new requirements are listed below. Between brackets, we show who came up with the new requirement. All these requirements have been created and taken into account in phase 2.

Should have requirements

- An association, external party or Student Union should have the ability to see how many people signed up for an activity they created. *[feedback session]*
- Campus Facility Management should have the ability to see how many people are at a facility. *[client]*

Could have requirements

- Students and Employees could have a block to prevent them from being able to take a screenshot of the QR code. *[client]*
- Campus Facility Management could have the ability to approve activities created by associations and external parties. *[client]*
- Campus Facility Management, Student Union, an association or an external party could have the ability to specify if the activity is an event or facility. *[own needs]*
- Students and employees could have the ability to see how many people signed up for an activity. *[feedback session]*
- Students and employees could have the ability to buy Union Card and/or subscriptions/courses. *[feedback session]*

- Students and employees could have the ability to add contact information In Case of Emergency (ICE). *[feedback session]*
- Events and facilities could have a share button to share the activity. *[client]*
- Associations, external parties and Campus Facility Management could have the ability to enter different prices for activities for Union Card holders and non-Union Card holders. *[client]*

8

Testing and Evaluation

In the last phase of our project, phase 3, we will mostly focus on testing and evaluating the design we built during the previous phases. In this chapter, we will discuss all testing and evaluation that took place during Phase 3. We will start with our Unit Testing approach, and afterwards go into detail about the user testing.

A full overview of all interactions where we have gotten feedback can be found in Appendix A. Keep in mind that associations have also tested the student views, as they also make use of these views when personally dealing with the application.

8.1. Unit Testing

First, we will discuss our test plan. In this plan we will outline how we will approach testing and why it makes our application safer. We will also go over what we will not test. Afterwards, we will discuss our user testing strategy and the results thereof.

We will write tests for each service that is used in the back-end. These tests will check for common errors (null-safety) and what would happen if someone tried to abuse the system. During the development, we aimed to return clear messages in our errors so that a user of our system would also know what happened and why it did not work as supposed to. We also aim to test controllers, as this is what interacts with users when they send requests to our server.

See below for our test coverage.

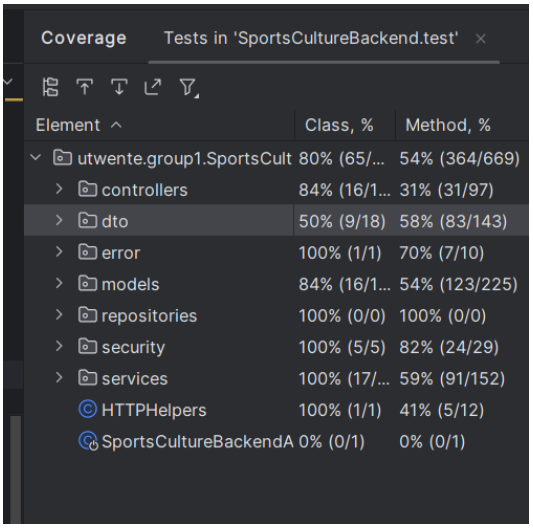


Figure 8.1: Testing Coverage of our back-end

We also will not automatically test the front end. We have decided against this because of time constraints. We of course did test the front end during development manually.

8.2. User Testing

To test our application, we undertook task-oriented testing with potential actual end-users of the app. We asked several students and employees to test the student and employee view, and a few associations and external parties to test the association view. We created these tasks by looking at our requirements, for each requirement we defined one or more tasks for the user to perform, and test whether and how quickly the user is able to finish the task without intervention.

When conducting user testing, caution is required regarding the researcher's actions. If the researcher is too involved in the process, they could introduce a bias that may skew results too favourably. The same holds for how the tasks themselves are articulated, which should be as 'neutral' and broad as possible, while still making it clear what the objective of the task is. To mitigate these issues, we worked with 4 main principles when performing our user tests:

1. **Neutrality**: Tasks should be conveyed as neutral as possible. This means we should only describe the result as a general objective of the stakeholder. Articulation hinting towards navigation features, such as the idea of profiles or views, should not be permitted.
2. **Text Communication**: Tasks must never be communicated with speech. If the researcher does talk with the tester, such as when the tester asks for clarification, it should be documented as an intervention.
3. **Observation**: When a tester is performing a task, the researcher may only observe what the tester is doing. Any other action, like taking notes or giving hints, is not permitted.
4. **Intervention**: The researcher may intervene if the user is struggling with completing a task. This, however, should be well documented if this happens.

8.2.1. Task-based Testing

We have separated the tasks for different stakeholders. First, we will present the tasks we have done with different students and employees. Then the tasks we have asked association boards and external parties to perform. At last, we will present the results of these tests. The tasks have been created in such a way so that it contains the main functionality of our application.

Tasks for Students and Employees

In this section, we will provide the tasks which we have asked students and employees to perform. Fourteen different people have performed the tasks for students and employees.

Task 1: See all possible activities
The task given to the student or employee: "In about two months there will be a painting workshop which you are thinking on joining. Could you please find information regarding this workshop? How many students already signed up for this painting workshop? Sign-up for this event."

Task 2: Reserve a facility
Task given to the student or employee: "Please reserve a spot in the fitness hall. Will it be busy on the day you reserved?"

Task 3: Show Union Card
Task given to the student or employee: "You have installed this app and let's say you own a Union Card. Then, some party asks you to show your Union Card. Please use this app to show your Union Card to me."

Tasks for Associations and External Parties

In this section we will provide the tasks which we will ask associations and external parties to perform. Six different people have performed the tasks for associations and external parties.

Task 1: Create an event
Task given to board member: "You are a board member of NEST. Your cultural association is hosting a theatre performance in Vrijhof. It is free for people with a Union Card and €3.50 for people without a Union Card. Create this performance in the app, so that people can sign up for it."

Task 2: Check tickets
Task given to board member: "You are checking tickets for the theatre performance. Check whether people have got a ticket for your performance. How can you do that in the app? What are the possibilities to check people's tickets?"

Task 3: Manage members
Task given to board member: "Go to the management page. Describe what the 'Needs Approval' section means."

Task 4: New board member
Task given to board member: "Your association NEST has a new board member: Bjorn Vuurens. Add him to the board, so that he can access these management pages too."

8.2.2. User Feedback
In order to test and find problems with users, we asked some people to test our application and we asked for their opinions. We will present the result here.

8.2.3. Results
From these user tests, we were able to extract valuable insights into the functionality, usability, and design of our app. In total we did

Tasks for Students and Employees

Task 1: See all possible activities
On average, 6.29 clicks were needed to perform this task. Almost all participants first searched on the home page, and could not find the painting workshop since it is not happening somewhere in the upcoming two weeks. Then people went to the activities page. 36% of the people choose to use the search bar, and the other 4 pressed the category to find the event that they are looking for. When people logged in, they were redirected back to the activities page, which costed 2 more clicks. The minimum amount of clicks for this action was 6. Because of this, some people suggested that you get routed back to the page where you left of before you log in. This way you do not need to do duplicate actions.

Some participants notice that searching the categories would go faster if the pictures are representing the category. "I did not expect a chess picture for the Arts & Craft category." In addition, some people said they like the idea of categories and that you can see how many people already signed up.

Task 2: Reserve a facility
On average, people needed 6 clicks to reserve for a facility, while 5 is the minimum amount (since the users were already logged in this time). Some participants went back to the home page to find the fitness hall there. When they could not find it, they went to the fitness category and reserved easily.

The people who tested our application think it was easy to reserve. However, someone notices that (s)he does not know how to cancel the reservation.

Task 3: Show Union Card
On the written task description, everyone was able to show the QR code in the app with 1 click, which is also the minimum. No problem was reported here.

Tasks for Associations and External Parties

Task 1: Create an event
Since we started in a position of the student view, people first needed to switch to the association view

to be able to create an event. On average 25+ clicks were needed to perform this task. People did not know they had to switch profile in order to create an activity. Some people clicked on the association card under My Associations but nothing happens there. Some participants propose an option that you can click on the association too to switch view. However, it may be unclear then how to switch back to the student/employee view then.

When the testees find out they had to switch to an association profile, an event was created with the minimum amount of clicks. From this we learned that the switching between profiles is not intuitive at all for the first time. Also, people did not read the text 'Switch profile' at all, and only noticed its existence after they have switched profiles. When people found out about the way switching to the association view goes, they had no problems with it later.

One participant proposed a new idea: make the switching between profiles more visual. This makes sure that people can see the switching between profiles option immediately when they visit the profile page. We think that is a very good idea. The user would see a person if the user is on its own personal profile, and sees logos of different associations in the back to switch to them. An illustration of this idea can be found in Figure 8.2.

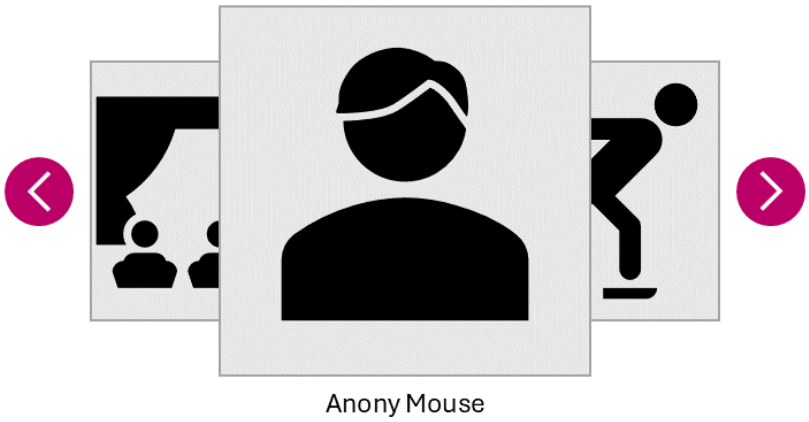


Figure 8.2: A visual alternative for our drop down menu to switch between profiles

Task 2: Check tickets
On average people needed 1.25 clicks to perform this action. They needed to go to the scan page and then tell the researcher what they had to do. All of them could figure out what to. Some read the text on the page and then understood what to do, while others already read 'NFC' and performed the action faster for the first time.

One participant however misread the page showing up when a student card was scanned. The researcher showing the student card was signed up for an event, but not the event that the participant was scanning for. That means that the participant had to read whether the researcher signed up for the right event. "I saw something green, so I thought it was alright." We think it is a rare situation that associations host two events at the same day, but we learned that it may be confusing to show multiple events at the same time. Another participant thinks it is weird that you can see events for which people did not sign-up. The participant proposes some button that you can click on the event in a way like: "Right now, I will be scanning for this event and nothing else." Then you will get either a green screen, or a red screen whether it was good or not.

Task 3: Manage members
Somewhat more than half of the people could describe perfectly what the 'Needs approval' section should represent: namely the option to accept or deny possible new members. This is lower than we expected. Some people propose to call this part 'Approve new members' instead, so that it clear that it is about approving new members, not about approving board requests or something else.

Another suggestion was made: add tags to the person asking to become member that show whether a

user already has a Union Card and/or paid the association fee. Or be able to click on that person, and see the same information as you can see with members. "At least some contact information would be necessary."

A participant also suggested to have a 'member' and a 'board' subheading on the member management page. "This way, the board and member tag are not necessary."

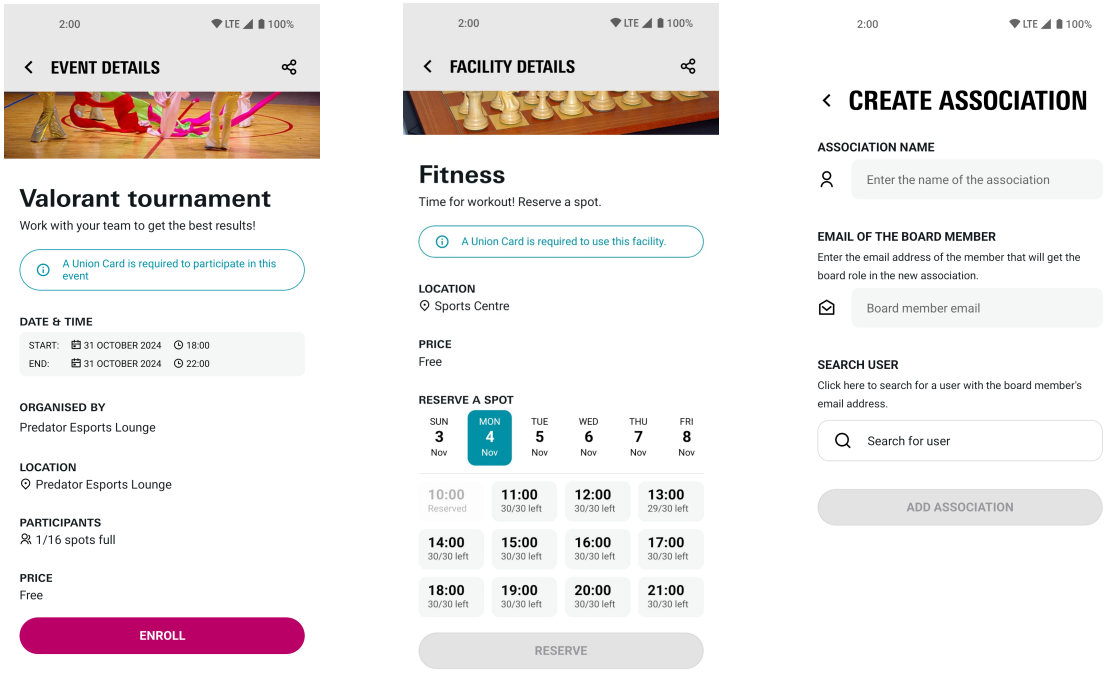
Task 4: New board member
No issues were found during this task. All our participants performed this task with the minimum amount of clicks and without any question.

9 Final Prototype

Building upon the tests and evaluations from 8, this chapter describes the final changes that resulted in the end product, and the requirements of the product itself. Most of the design is similar to the design shown in Chapter 7, however, some small changes have been made. We will walk through all the changes, highlighting improvements we made to the front- and back-end of the product. We will also present a summary of all the requirements we gathered throughout the project and whether we completed these requirements.

9.1. Changes

Based on the feedback and user testing during this last phase, we decided to make some final changes to improve our prototype. We will discuss these changes one by one.



(a) Event details page with share button

(b) Facility details page with share button

(c) Create association page

Figure 9.1

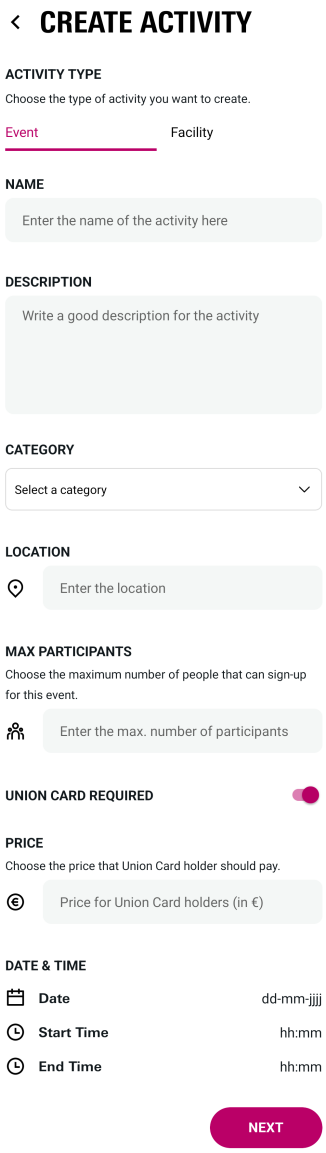
9.1.1. Sharing Links

To make the promotion of events within the app easier, we have decided to implement link-sharing of events. This was initially suggested by our client during one of our feedback rounds. In the top right corner of the page there is a new share button, as show in Figure 9.1a and 9.1b. Users can click this button of an event, which will generate a unique link for that event. If other users outside of the app click on this link, they will be redirected to our app and the specific event will immediately be opened for them to see. This way, users can easily share events they find interesting with their friends, improving the discoverability of events on our platform even more.

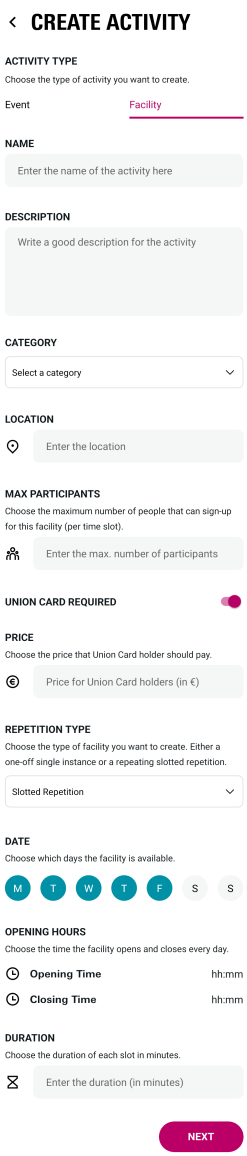
9.1.2. Create Association & Create Activity

Discoverability is important but there must be content to discover. The creation of events and associations could have been an issue, as during user tests, users were often confused about the layout of the 'add activity' and 'add association' pages, due to lack of information on the input fields. We changed the layout of these pages to make them clearer. On both pages, we added text above each input, and placeholders inside the input, making it more clear what the user needs to do and what fields are required.

On the Create Activity page specifically, we switched the order of the inputs to make the process more clear, and set the Union Card Required switch to be on by default, as most activities usually require a Union Card. The final Create Activity page can be seen in Figure 9.2, the improved Association page can be seen in Figure 9.1c



(a) Create activity page (Event)



(b) Create activity page (Facility)

Figure 9.2

9.2. Requirements Overview

After making these final changes, our third development phase is over. During the project we gathered many requirements and have implemented most, although not all, of them. Below is an overview of all requirements, with their status, and notes where extra clarification is needed.

MoSCoW Functional Requirements

Done	Type	Requirement	Notes
Yes	Must	A student or employee must have the ability to make a reservation for a facility.	
Yes	Must	A student or employee must have the ability to sign up for an event.	
Yes	Must	A student or employee must have the ability to see all possible activities.	

Yes	Must	Campus Facility Management must have the ability to create activities.	
Yes	Must	A student or employee must be able to log in using a student/employee account.	
Yes	Should	A student or employee should have the ability to see all events for the upcoming two weeks on the home page, in order to be able to discover new events.	
Yes	Should	A student or employee should have the ability to show a QR code that can be scanned by an association, external party or Student Union to prove their ownership of a Union Card or a reservation.	
Yes	Should	An association, external party or the Student Union should have the ability to scan a QR code to check whether a student or employee owns a Union Card, has a reservation, or has signed up for an event.	
Yes	Should	Campus Facility Management should have the ability to see statistics about the usage of all Union Cards.	
Yes	Should	Campus Facility Management should be able to see statistics about the amount of sign-ups for activities.	
No	Should	Campus Facility Management should have the ability to modify existing activities.	They only have the ability to remove and create a new one, which can be used for 'modifying' the users view
Yes	Should	Campus Facility Management or Student Union could have the ability to cancel any activity.	
Yes	Should	Campus Facility Management should be able to add an association to the app.	
Yes	Should	Campus Facility Management should be able to add an external party to the app.	
Yes	Should	A student or Employee should be able to see all events and facilities without logging in.	
Yes	Should	An association, external party or Student Union should have the ability to see how many people signed up for an activity they created.	
No	Should	Campus Facility Management should have the ability to see how many people are at a facility.	We did implement the gathering of this data, but did not implement specific statistic designs to show this number
Yes	Could	The QR code could have an expiry date and time.	
Yes	Could	Associations and external parties could have the ability to create activities.	
Yes	Could	Associations and external parties could have the ability to cancel their activity.	
No	Could	Associations or external parties could have the ability to see statistics about their association or organisation.	Did not implement this due to scope limitations

No	Could	Campus Facility Management could have the ability to revoke an external party's ability to create events.	They can only revoke this ability by deleting the external party or association
No	Could	Campus Facility Management could have the ability to add subscriptions/courses.	We did not implement the shop, therefore we also did not implement options to purchase
Yes	Could	Students could have the ability to prove ownership of a Union Card using their student card with NFC (instead of the QR code).	
Yes	Could	Associations or external parties could be able to add an account to their association or organisation so that these members can scan QR codes too.	
Yes	Could	Students and Employees could have a block to prevent them from being able to take a screenshot of the QR code.	
Yes	Could	Campus Facility Management could have the ability to approve activities created by associations and external parties.	
Yes	Could	Campus Facility Management, Student Union, an association or an external party could have the ability to specify if the activity is an event or facility.	
Yes	Could	Students and employees could have the ability to see how many people signed up for an activity.	
No	Could	Students and employees could have the ability to buy Union Card and/or subscriptions/-courses.	We did not implement the shop
Yes	Could	Students and employees could have the ability to add contact information In Case of Emergency (ICE).	
Yes	Could	Events and facilities could have a share button to share the activity.	
Yes	Could	CFM could have the ability to enter different prices for Union Card holders and non-Union Card holders.	
No	Won't	The database won't have a connection with the current S&C database for true insights.	Won't have requirement
No	Won't	Students and employees won't be able to scan QR codes themselves to prove ownership of the Union Card.	Won't have requirement
No	Won't	Students and Employees won't be able to purchase courses, pay association fees and buy subscriptions through the app.	Won't have requirement

MoSCow Non-Functional Requirements

Done	Type	Requirement	Notes
Yes	Must	Students and employees should be able to sign-up to an event within 5 clicks.	
Yes	Must	Students and employees should be able to reserve a facility within 5 clicks.	

Yes	Must	The app must have a modular architecture that allows for easy maintenance and updates.	
Yes	Must	The app must be compatible with iOS 16, 17 & 18 and Android 13.0 and 14.0.	
Yes	Must	Every API endpoint must only be accessible by users who are authorized to do so.	
Yes	Must	The app must be deployable using a CI/CD pipeline.	
Yes	Should	Students and employees should be able to show their student card or the QR-code on the app when someone asks them to show their Union Card.	
Yes	Should	The app design should follow the house style guidelines of the University of Twente.	
Yes	Should	The server should have a latency of less than 500ms for any endpoint.	
Yes	Should	The app should be able to handle at least 1,000 concurrent users without any performance degradation.	
No	Could	The app could comply with WCAG 2.1 Level A accessibility guidelines.	
No	Won't	The app won't have to comply with WCAG 2.1 Level AAA accessibility guidelines.	Won't have requirement

10 Discussion

In this chapter, we will discuss the results of the evaluation of our app, the future work for this project, and what it would need to actually be used by this university for sports and culture. Additionally, we also reflect on how we worked as a group and what progress has been made on a personal level.

10.1. Analysis of Findings

During our testing phase, we found some interesting results about the usability and user experience of our app. One major finding was that users found it difficult to switch to association profiles when starting on their student profile. However, we also observed that users had no trouble using this in future tasks. This raises the question if we should optimize our design more for first-time use or for all times after the first time. One idea that was floated in our feedback sessions was changing the drop-down button into some form of gallery where the available options are always visible to the user. A very different solution would be to offer a tour of our app to inform the user of the available options.

During our faculty presentation, it was made clear that there is another potential group of users of our app, namely visitors to our university for, for example, guest lecturers or conferences. Our client did not list this group as a group we should focus on. In order to accommodate this group of users, we would also need to accept non-University of Twente emails as valid login. Then Campus Facility Management could assign these users a Union Card.

With user testing, we found that the current system we use for approving members into an association was not entirely clear without an explanation of how it worked. Our client expressed an alternative to this system by tying membership to association fees, which need to be bought from the shop. Although this idea would work very well in theory, we identified some practical issues with this system. Associations usually have a member cap due to limited capacity at trainings, which means that entry to an association must be regulated. Another important requirement for associations for such a system is that members should have a higher priority for extending their membership over new members. These other requirements may make such a new system just as complicated as the existing system. However, it might still be useful to explore the different possibilities and alternatives to this system.

10.2. Future Work

Our project is a proof of concept. This means that, in its current state of affairs, it is not ready to be published and actually used. In this section, we will cover what would need to be done in order to make it usable.

10.2.1. SURFConext and Integration with University Systems

Right now, you can log in using Microsoft. Since the University of Twente has Microsoft accounts, this poses no immediate problem. We manually check in our back-end if the email the user is trying to log

in with is a valid University of Twente employee or student mail. Many systems of the University of Twente, however, make use of SURFConext. This is a third-party authentication flow for educational purposes. To offer a seamless experience with other systems at this university, our app would also need to make use of SURFConext. This was, however, out of the scope of our project since applying for a license for this product can take up to several months. Additional benefits of using SURFConext would be the possibility to connect this with the other data the University of Twente gathers, and the possibility to expand this application to more universities that do not make use of Microsoft, but instead use Google or any other provider.

10.2.2. NFC

Last year, all student cards have been replaced with a cryptographic NFC chip. As of writing this report, not all employee cards have been replaced. That means that some cards still only send a four or seven-byte code. This is only a maximum of 56 bits of entropy, which equals a seven character long password. This is easily fake-able by a random password generator that keeps trying until it finds the correct sequence of characters. The new cards have the ability to do asymmetric encryption, making use of a private and a public key. To make use of this system, our app would need to send a signal to the student card, instead of only receiving this code. These newest student cards would then 'sign' their response with a private key. Our app would then receive this signed response, and check the validity of the student card by using the public key. This would make the use of the student cards very secure. Right now the only way a person could check if the student card really belongs to the person tapping it on the back of the phone would be to check the picture on the card. Therefore, once all cards have this new cryptographic NFC chip, our app should support this.

10.2.3. Shop

As discussed previously, our new Sports&Culture app does not have shop functionalities. At the start of our project, we decided these shop functionalities requirement would fall in the won't have category, since we did not have access to the systems of the University of Twente as this would just be a proof of concept. We therefore did not design this page and implement these functionalities. After interviewees remarked that they missed this page, we decided on only designing this page without implementing any of the functionalities on the back-end. In order to make the app usable, the shop should be implemented in full. This would then also require integration with payment platforms and the systems of the University of Twente.

10.2.4. Web Interface

Currently, our app only supports native android and iOS devices. We chose our tech stack in such a way that our project can be expanded to also support a web interface, by using React Native. In order to offer our functionalities on a laptop or desktop as well, future work could be to expand our front-end to support this web interface.

10.2.5. Extended Tag System

Currently, users with a Board role can assign members of their association two different tags, either active member or committee. In addition to these two tags, it would be practical for a board member to have the ability to create more tags with different naming, and possibly even different permissions. That way, users can have different tags representing their respective committees, or access to for example member management or activity creation instead of only the ability to scan cards.

10.3. Reflection

In this section, we will discuss what went well in our group work, what we have improved upon, and what we still could improve on next time.

At the beginning of the project, we had issues in communicating when we were absent during physical meetings. This caused some frustration within the group during the first few weeks. In order to fix this issue, we decided on a system that would register when we were absent. We used the WhatsApp group description for this. We would each register when we would be absent from a meeting. In doing this, we were all informed sufficiently and could manage expectations of each other.

During the development process, we had the strategy of creating 'issues' on GitLab for each feature, as discussed in Chapter 3. Over the course of the project, these issues became too large and not so well-defined any more. This caused the problem that branches were very far behind on the main branch, which in turn caused very large merge conflicts that took a lot of time to solve. In addition to this, other people were sometimes 'waiting' on a feature that was already created, but which branch was not yet merged. This taught us about the importance of dividing a task into smaller stand-alone subtasks. In a future project, we would give more attention to this problem in advance, trying to prevent this from happening.

A part of our group also worked without a code formatter in the first few weeks, while the other part did. This caused large merge conflicts between code that only had to do with formatting. We fixed this by creating a configuration file that held the standard for our project. Everyone used this file and when saving code this formatting would be applied automatically.

11

Conclusion

In this report, we have delved into various aspects of our app. We have discussed our approach, our different development cycles, how we incorporated feedback, and our testing strategy. We have analysed the current state of affairs for the Sports & Culture app, and designed and implemented a prototype with the help of interviews and user testing. We also explored potential improvements, such as integrating SURFConext for authentication, using the new NFC security features on student cards, and expanding functionalities like the shop and web interface.

Our newly designed app provides a more stable platform for sports and culture at the University of Twente. Utilising Spring Boot, React Native, and Hibernate, we were able to create a modern mobile application that can also be extended to a web interface. In this new app, we have tried to address and improve upon the limitations of the current app. By integrating extensive user testing and multiple feedback rounds, we were able to quickly process the feedback we received. We believe our prototype improves on multiple points, such as security, user-friendliness and in general offering a better experience for those using the facilities of the University of Twente. While our app remains a proof of concept, we believe it has laid a strong foundation for further development.

A

Overview Interactions

Phase	Type	Role	Amount
0	Survey	Students (& Associations)	25
	Meeting	SU & CFM	1
		CFM	1
1	Semi-structured Interview	Associations	5
	Meeting	CFM	1
2	Feedback	CFM	4
		Students	5
3	User testing & Feedback	CFM	4
		Students	14
		Associations	6

Table A.1: Overview of the amount of feedback sessions, interviews and user-testing per phase

B

Manuals

B.1. Manual for students, employees and external users

As a student, employee or external user, you can perform several tasks. These tasks are the following, grouped into different categories:

B.1.1 Activities

1. Enroll for an event
2. Unenroll for an event
3. Reserve a facility spot
4. Cancel a facility spot reservation
5. Share an event or a facility

B.1.2 Union Card / Campus Card

6. Enter a facility or location in which a ticket or Union Card is required.

B.1.3 Associations

7. Join an Association
8. Check membership status of one of my associations
9. Change emergency contact information

How to perform each of these tasks will be described below.

B.1.1. Activities

Task 1: Enroll for an event

Enrolling for an event can be done in three ways:

- Option 1: Enroll for an event on the homepage
- Option 2: Enroll for an event via categories
- Option 3: Enroll for an event by searching

If the event is this week or next week, choose option 1. Otherwise choose option 2 or 3, depending on your preference.

Option 1: Using the Home Page

1. Go to the Home Page. The events for the upcoming two weeks are shown.
2. Press on an event of your choice.

3. You can now see the details of the event. Scroll down and press the login button if not yet logged in. If you have already logged in, the button will change to the 'Enroll' button. Press this button to enroll for the event.

Option 2: Using categories

1. Go to the Activities Page.
2. Under the subheading 'EXPLORE' click on the category you think you will find the event you are interested in.
3. Click on the event you are interested in.
4. You can now see the details of the event. Scroll down and press the login button if not yet logged in. If you have already logged in, the button will change to the 'Enroll' button. Press this button to enroll for the event.

Option 3: Using the search bar

1. Go to the Activities Page.
2. Press on the search bar on top of the page showing "Search for activities", and search for the event you are looking for. Tip: If you write 'Event' all events will show up.
3. Click on the event you would like to enroll for.
4. You can now see the details of the event. Scroll down and press the login button if not yet logged in. If you have already logged in, the button will change to the 'Enroll' button. Press this button to enroll for the event.

You have now successfully enrolled for the event.

Task 2: Unenroll for an event

1. Go to the Home Page.
2. Press on the "My Activities" button on the top-left of the page. (The "My Activities" button on the Activities Page will do the same.)
3. If you have not yet logged in, log in using the login button on that page.
4. Search in the subheading "My Events" for the event you would like to unenroll for. Click on that event.
5. Scroll down on the event detail page, and press the "Unenroll" button.
6. You are asked whether you are sure you want to unenroll. Press 'unenroll'.

You have now successfully unenrolled for an event.

Task 3: Reserve a facility spot

Reserving for a facility can be done in three ways:

- Option 1: Reserve a facility via categories
- Option 2: Reserve a facility by searching

Option 1: Using categories

1. Go to the Activities Page.
2. Under the subheading 'EXPLORE' click on the category you think you will find the facility you are interested in.
3. Click on the facility you are interested in.
4. You can now see the details of the facility. Press the date you want to visit the facility. Only if multiple timeslots are available, you should then also press the time slot you would like to visit.
5. Press the "Reserve" button on the bottom of the page to reserve the selected time slot.

Option 2: Using the search bar

1. Go to the Activities Page.
2. Press on the search bar on top of the page showing "Search for activities", and search for the facility you are looking for. Tip: If you write 'Facility' all facilities will show up.
3. Click on the facility you would like to reserve.
4. You can now see the details of the facility. Press the date you want to visit the facility. Only if multiple timeslots are available, you should then also press the time slot you would like to visit.
5. Press the "Reserve" button on the bottom of the page to reserve the selected time slot.

You have now successfully reserved the facility.

Task 4: Cancel a facility spot reservation

1. Go to the Home Page.
2. Press on the "My Activities" button on the top-left of the page. (The "My Activities" button on the Activities Page will do the same.)
3. If you have not yet logged in, log in using the login button on that page.
4. Search in the subheading "My Reservations" for the facility slot you would like to cancel the reservation for. Press on the red cross on the right of that spot.
5. You are asked whether you are sure you want to cancel this reservation. Press 'delete'.

You have now successfully cancelled the reservation.

Task 5: Share an event or facility

1. Click on the event or facility you would like to share. The page including the details of the event or facility should show up.
2. Press the share icon on the top right of that page.
3. Share the event or facility through your favourite medium.

You have now successfully shared the event or facility.

B.1.2. Union Card / Campus Card

Task 6: Show ticket or Union Card/Campus Card

Someone can ask you to show your ticket for an event you have enrolled for, to show your reservation proof for a facility you have reserved, or to show your Union Card or Campus Card. All this information is saved in your account, so per situation, you need to perform the exact same action. This way you can always perform the same action when someone asks you to show something!

There are two ways of showing your ticket or Union Card/Campus Card:

- Option 1: Show your Student Card or Campus Card
- Option 2: Use our app

Option 1: Show you Student Card or Campus Card

1. If you own a Student Card or Campus Card, simply show this card and let it be scanned by the person who checks for your ticket or Union Card/Campus Card. If you forgot this card, or do not own one, try option 2.

Option 2: Use our app

1. Go to the "Union Card" page.
2. If you have not yet logged in, log in using the login button on that page.
3. Show the QR code that appears on the screen.

You have now successfully showed your ticket, reservation proof or Union Card/Campus Card ownership.

B.1.3. Associations

Task 7: Join an association

1. Go to the Profile Page.
2. Press on the '+' icon right of the "My Associations" subheading.
3. Choose the association you would like to join and press on the icon on the right of that association

You have now successfully send a request to the association to become member. You need to wait for the association to accept your request.

Task 8: Check membership status

1. Go to the Profile Page.
2. Find your association under the "My Associations" subheading.
3. The coloured tags show your roles within the Association. They have the following meaning:
   A) Pending: you have send a request to become member, but this has not been accepted (yet).
   B) Member: you are a member of the association
   C) Active member: you are member of a committee that has rights to scan other people to check whether they have a ticket or own a Union Card.
   B) Board: you are a board member of the association, and thus owns the rights to create activities and do member management.

Task 9: Change emergency contact information

Some associations require your emergency contact (ICE) phone number. You have the capability of putting this ICE phone number in the app by following these steps:

1. Go to the Profile Page.
2. Press on the writing icon on the right of the "Personal Information" subheading.
3. Click on the Emergency Contact Details field and fill in your ICE phone number.
4. Press on the "Save changes" button to save your ICE phone number.

B.2. Manual for associations and external parties

As an association or external party, you can perform several tasks. In order to perform these tasks, you need to log in, and switch to your association profile. You can only do that if you are board or an active member of that association. You can switch profile by going to the Profile page, then pressing on the drop down menu on the top, and then choose the association you would like to switch to. You can now see a different navigation bar on the bottom, specific for your association. The tasks you can do are the following, grouped into different categories:

B.2.1. Activities

You need to be a board member in order to perform actions in this category.

Task 1: Create an activity

1. Go to the Activities page.
2. Press on the add icon in the upper right corner.
3. You can now add the details of the event.
4. Press 'NEXT' to preview your event. Here you can also see what others will see when looking at your event.
5. Press 'SUBMIT' to submit your event. The Student Union and Campus Facility Management have to approve your event before it will be shown on the home page.

You have now successfully created an activity.

B.2.2. Scanning

You need to be a board member, or an active member in order to perform actions in this category.

Task 2: Scan access code or Union Card

Scanning an access code or Union Card can be done in two ways:

- Option 1: Scan via QR code.
- Option 2: Use NFC to scan Student Card.

Option 1: Scanning QR Code

1. Go to the Scan page.
2. Press on the 'SCAN QR-CODE' button.
3. Aim your camera at the QR code presented by the student or employee.
4. A page will pop up that shows if the person has a Union Card and for what events or timeslots the person registered from your association.

You have now successfully scanned a QR code.

Option 2: Use NFC to scan Student Card

1. Enable NFC on your phone.
2. Go to the Scan page.
3. Press the student or employee card of your guest to the back of your phone.
4. A page will pop up that shows if the person has a Union Card and for what events the person registered.

You have now successfully used NFC to scan a student card.

B.2.3. Member Management

You need to be a board member in order to perform actions in this category.

Task 3: Approve User that wants to become Member
Users can send a request to become member. In order to actually become member, you need to approve their request first.

1. Go to the Management page.
2. If your association or external party has members that need approval, they will show up in a list called 'NEED APPROVAL'.
3. Click on the green check mark to approve this member.

You have now successfully added a user as a member for your association.

Task 4: Deny User that wants to become Member
1. Go to the Management page.
2. If your association or external party has members that need approval or have to be denied, they will show up in a list called 'NEED APPROVAL'.
3. Click on the red cross to deny this member.

You have now successfully denied a member from your association. This user will not be member of your association.

Task 5: Remove Member
1. Go to the Management page.
2. If your association or external party has members that need to be removed, they will show up in a list called 'Members'.
3. Click on the member you want to remove.
4. The information page of this member will pop up. Click on the trash icon in the upper right corner to remove this member from your association or external party.

You have now successfully removed a member from your association.

Task 6: Add Role to Member
To make sure other members can access these association pages too you can give them roles. The Board role will give access to all the association pages. The Active Member role will give access to the scan page. This way other members can scan cards too.

1. Go to the Management page.
2. If your association or external party has members, they will show up in a list called 'Members'.
3. Click on the member you want to assign a role.
4. The information page of this member will pop up. You will see a list of 'EDIT ROLES'. To assign a role to this member, please check the box in front of the role you want to assign.

You have now successfully added a role to a member.

Task 7: Delete Role from Member
1. Go to the Management page.
2. If your association or external party has members, they will show up in a list called 'Members'.
3. Click on the member you want to delete a role.
4. The information page of this member will pop up. You will see a list of 'EDIT ROLES'. To remove a role to this member, please unmark the box in front of the role you want to deassign.

You have now successfully added a role to a member.

B.3. Manual for Campus Facility Management and Student Union

As Campus Facility Management or Student Union, you can perform several tasks. In order to perform these tasked, you need to log in, and switch to the Campus Facility Management profile. You can only do that if you have the administrator permissions. You can switch profile by going to the Profile page, then pressing on the drop down menu on the top, and then choose Campus Facility Management. You can now see a different navigation bar on the bottom, specific for your management tasks. The tasks you can do are the following, grouped into different categories:

B.3.1. Statistics

Task 1: See Statistics

1. Go to the Dashboard page.

You can now see the statistics.

B.3.2. Activities

Task 2: Create an activity

1. Go to the Activities page.
2. Press on the add icon in the upper right corner.
3. You can now add the details of the event or facility.
4. Press 'NEXT' to preview your event or facility. Here you can also see what others will see when looking at your event or facility.
5. Press 'SUBMIT' to submit your event or facility.

You have now successfully created an activity.

B.3.3. Association Management

Task 3: Add Association

1. Go to the Associations page.
2. Press on the add icon in the upper right corner.
3. You can now add the name of the association.
4. Enter the email of a person that should be the first board of this association.
5. Press 'Search for user' to find the account of this board email, to add them to this association.
6. Press 'ADD ASSOCIATION' to add the association.

You have now successfully added an association. The first board member can now add other board members.

Task 4: Remove Association

1. Go to the Associations page.
2. Press on red trash icon next to the association you want to delete.

You have now successfully removed an association.

B.3.4. Approvals

Task 5: Approve Event or Facility

When associations create events or facilities, you need to approve the event or facility first before the public can see the event or facility.

1. Go to the Approvals page.
2. Click on the event or facility you want to approve.
3. Read through the event or facility and click on the 'APPROVE' button.

You have now approved an event or facility.