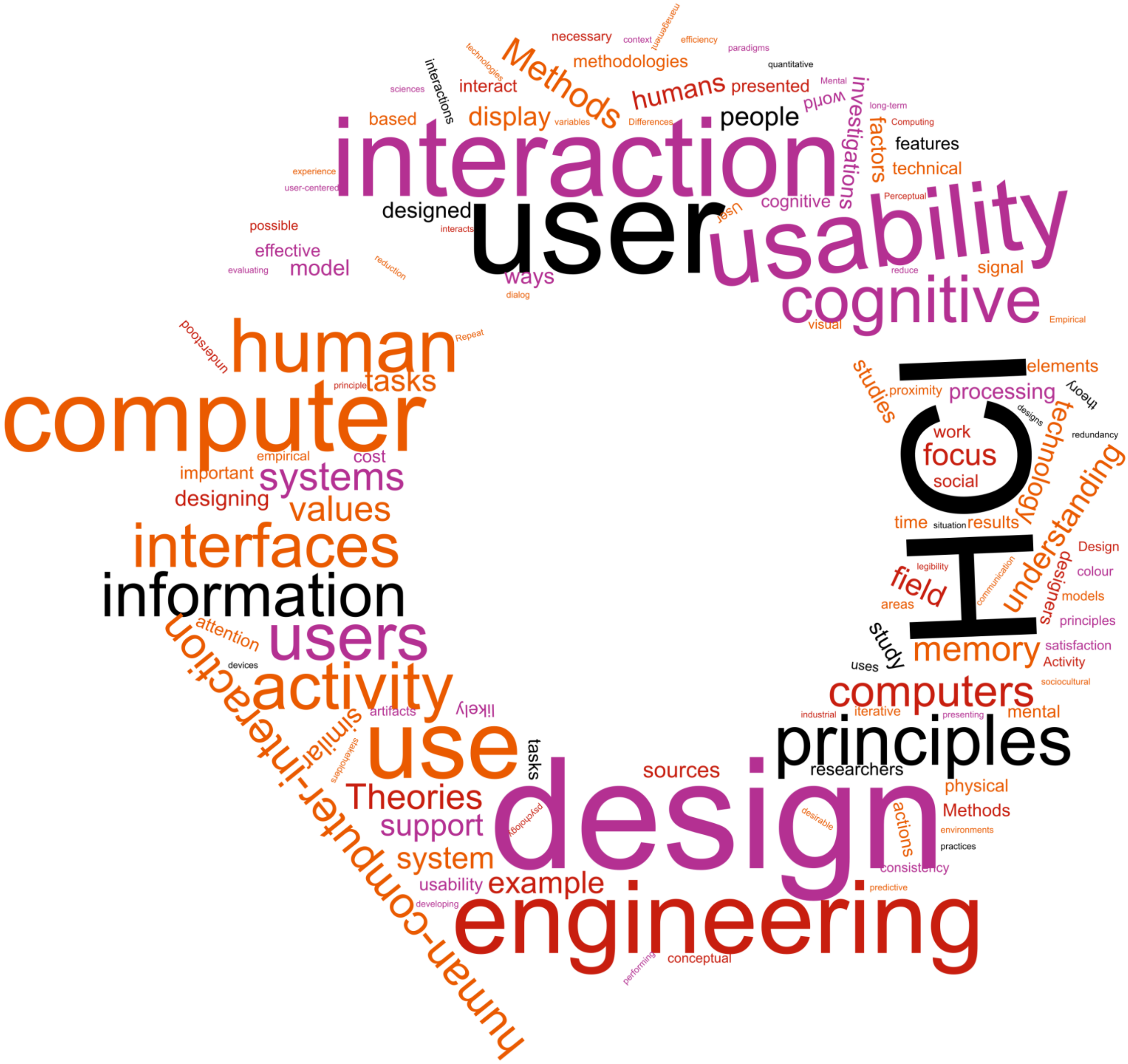
Mensch Computer Interaktion – Designspezifikation Meilenstein 2

Team

Lukas Gobelet	*11118011*
Joey Gobelet	*11118012*
Erik Gödel	*11117480*
Jonas Faßbender	*11117674*
Hamza Al-Dadda	*?*



Einleitung

Dieses Dokument gliedert sich nach den Meilensteinen des Praktikums. Die jeweiligen Kapitel der Meilensteine stellen den Fortschritt und die Entwicklung des Teams im Laufe des Semesters dar. Ab Meilenstein 2 werden die vorhandenen Artefakte auf Basis von Feedback und Evaluationen überarbeitet. Da es sich beim Rapid Prototyping um eine Darstellung des aktuellen Konzepts handelt ist es nicht notwendig den jeweiligen Prototyp (Meilenstein 1: Story Board, Meilenstein 2: Wireframes, Paperprototyp) zum folgenden Meilenstein zu überarbeiten. Die Überarbeitung dieser Artefakte wird in Form des nächsten Prototypens dargestellt. D.h. die Überarbeitung des Storyboards zeigt sich in den Wireframes und dem Paperprototypen. Die Überarbeitung der Wireframes und des Paperprototypen zeigt sich im funktionalen Prototyp.

Was in den jeweiligen Abschnitten zu erbringen ist wird immer über *[Platzhalter]* gekennzeichnet. Hierbei gibt es 3 Arten von Kennzeichnungen:

[(Überarbeitete/Finale) Artefakt/e]

Platzhalter für das jeweilige Artefakt welches es anzufertigen bzw. zu überarbeiten gilt z.B. Personas.

[Erläuterung]

Platzhalter für eine Erläuterung des jeweiligen Artefakts. Hierbei ist keine Erläuterung der Methode oder Technik gemeint sondern vielmehr eine Begründung von Entscheidungen. Warum wurden ausgerechnet diese Personas erstellt? Warum sind Elemente im Wireframe wie im Artefakt ersichtlich angeordnet?
Dieser Punkt ist essenziell für das Bewertungskriterium „Dokumentation“. Der wichtigste Anhaltspunkt ist hierbei das Feedback seitens der Testnutzer als auch der Praktikumsbetreuer. Sie sollten stets versuchen ihre Entscheidungen durch Nutzer oder Quellen zu stützen. (Bspw. Style Guides, MCI Vorlesung etc.) Entsprechend verwendete Quellen sind sowohl in der Erläuterung als auch im Quellenverzeichnis zu dokumentieren.

[Überarbeitung]

Ab dem 2. Meilenstein dokumentieren Sie Veränderungen an einzelnen Artefakten. Dies kann stichpunktartig erfolgen. Achten Sie jedoch darauf, dass sich diese Änderungen auch in der Erläuterung wiederfinden. Bspw.:

-zusätzliche Persona Max aufgenommen, wir hatten die Perspektive dieses Nutzers übersehen

-Repositionierung des Login, 2 Tester hatten Probleme diesen Button zu finden, sie hätten ihn an einer anderen Position erwartet

Im Laufe des Projekts mag ein Artefakt so solide sein, dass eine Überarbeitung nicht mehr notwendig ist. In diesem Fall kopieren Sie einfach Artefakt und Erläuterung aus dem vorherigen Abschnitt in diesen und notieren „Keine Überarbeitung erfolgt“..

Szenario

Nach Ihrem erfolgreichen Studium sind Sie Mitarbeiter/-in der Softwareschmiede "Best Practice GmbH". Ihr Unternehmen konzentriert sich auf maßgeschneiderte Softwarelösungen und die Digitalisierung von Arbeitsprozessen.

Das neueste Projekt, welches gleichzeitig Ihren Einstieg in den Geschäftsalltag darstellt, erhielt Ihr Unternehmen durch die "TH-Köln".

Am Campus Gummersbach ist es mit entsprechenden Berechtigungen möglich, Transponder zum Öffnen von Räumen auszuleihen. Der Prozess zum Ausleihen, dem Prüfen von Berechtigungen, als auch dem Verleihen von Berechtigungen erfolgt dabei rein auf ausgedruckten Listen.

Die Raumverantwortlichen übermitteln hierzu Berechtigungen an die Pforte. Hier führen die Mitarbeiter/-innen an der Pforte eine Liste mit allen Schlüsseln und den dazugehörigen Personen, die die Berechtigung zum Ausleihen eines Transponders besitzen. Ein Transponder kann dabei mehrere Räume öffnen. Möchte eine Person nun einen Transponder ausleihen, ist es durch die Mitarbeiter/-innen an der Pforte notwendig die Person und seine entsprechende Berechtigung in den entsprechenden Listen zu überprüfen. Besitzt diese Person die entsprechende Berechtigung trägt er sich mit Datum, Uhrzeit und Namen in eine Verleihliste ein.

Die aktuelle Handhabung ist sehr Pflege intensiv und widerspricht den Leitkriterien guter Usability: Effektivität, Effizienz und Zufriedenheit.

Ihre Aufgabe ist es den Transponderverleih der TH Köln am Campus Gummersbach zu optimieren. Hierzu entwickeln Sie im Laufe des Projekts ein Konzept welches sie prototypisch implementieren und testen.

Meilenstein 2

Nutzermodellierung

Stakeholder Analyse

[Überarbeitetes Artefakt/e]

The Stakeholder analysis is a description of all the types of people who are fundamental for the project, in our case the transponder-lending-system. One of each type were at the stakeholder interview, which was in the building complex where the system will be used. It was held before anything was planned, so that we know what the stakeholders expect from the new system.

Gatekeeper

He/she lends the transponders to the borrowers.

For this they need the authorization lists from the room staff (the person in charge of rooms), to know to whom they can lend transponders.

They are also holding a register about who borrows which transponder and the lending time.

He is interested in automating the authorization lists, because it takes way too long to look through and manage them per hand.

Persons in charge of rooms

They grant the borrowers the timely limited permissions for the transponders.

They must give these to the gatekeeper.

They are interested in automating the system of delivering the lists to the gatekeeper.

Borrowers

The borrower can be any person. He can borrow the transponder for which he has the authorization granted.

While he has the transponder he also has the alone responsibility for it and he must sign this with his name when borrowing a transponder at the gatekeeper.

To borrow a transponder, he must give his name and the transponder- or room number to the gatekeeper.

The borrower is just interested in a fast and smooth lending process without having to give much data.

Central office in cologne

The office in cologne is responsible for maintenance of the transponders.

To this counts the programming, repairing and fabrication of the transponders.

[Erläuterung]

The stakeholder analysis gave us an overview about the things that are important for the stakeholders and what they expect from the new system.

As you can see we have four stakeholders in the system, but there are three main stakeholders, person in charge of rooms, borrower and gatekeeper. The Central office in cologne is a passive stakeholder because it doesn't affect the system in a direct way and only for maintenance.

The most important thing to see is that the communication between Person in charge of rooms and the gatekeeper should be simplified, because the gatekeeper momentary has these big paper lists which he must manage and the person in charge of rooms must deliver these lists manually to the gatekeeper. A disposal of the paper lists would also accommodate the borrower because the lending process would be faster.

[Überarbeitung]

- added introduction
- added conclusion

User Profiles

[Überarbeitetes Artefakt/e]

The user profiles introduce the users of the transponder-lending-system.

They state which relationship the users have to the product, which problems the system tries to solve and what the motivation for the system for every type of user is.

Gatekeeper

Motivation:

The current solution for the lending of transponders is associated with a very high administration effort.

Huge list must be looked up and the synchronisation for new permissions works very badly.

So, the gatekeeper has a hard time managing all these lists and the process occupies accordingly a lot of time.

The gatekeepers are hoping for a new solution, that makes all this easier, so they can concentrate on more important aspects of their work and don't need to struggle anymore with this huge amount of paperwork.

Persons in charge of rooms

These are professors and staff of the TH Köln at the Campus Gummersbach, who oversee rooms and can give permissions for transponders that open their rooms to other people.

Characteristics:

They are very busy and don't have much time.

Motivation:

They want to give and remove permissions for transponders in an easy and quick way. Permissions should be removed automatically after certain amount of time (e.g. half a year).

Borrowers

There are three different types of borrowers: 1. students, 2. professors and scientific staff and 3. external borrowers.
All borrowers are responsible for any loss or break of the transponder. There are differences how these user types use the system and which role the lending system has for them. These differences are depicted below.

Type A students: These are the primary users of the system. They usually identify themselves when they want to borrow a transponder with their multicard. When they want to view their permissions for the borrowing of transponders they can log in to the system with their gmid-account (or campus-it account).

Type B professors and scientific staff: These are secondary users of the system. They use the same interface as the student, but they may have additional rights and options for using the system.

Type C external borrowers: These are all the borrowers that don't fit in the above categories. So that can be cleaning staff who need to access the rooms for cleaning or a craftsman who need to repair something. They don't use the interface for viewing their permissions for borrowing transponders. They either borrow transponders ad hoc, e.g. a craftsman who needs to repair something just need the transponder one time and then no more, or they have a long-term permission, e.g. cleaning staff needs to access rooms for the time they are doing this job.

System Administrator

The system administrator is an expert user and it's a user that wasn't there beforehand but is needed for the system to work correctly. He/she is responsible for the data maintenance like adding, changing and removing transponders and users. He/she has a deep understanding of the lending-system and is also responsible for any exceptions and questions people have. Although he/she is an expert, they wish for an interface that is practical and can get their job done quickly, because they also have lots of other tasks in their job.

[Erläuterung]

The user profiles should have given us an overview over the users and a short introduction to each user and their characteristics (when important) and the motives.

As you can see, we did not mention the Central office in cologne from the *Steakholder Analysis* as a user. This is because the central office is no user and is therefore an anti-user. Our system only covers the lending process and the managing of permissions for transponders and not the maintenance and production of the transponders.

We also added an additional user that didn't appeared in the *Steakholder Analysis*, the system administrator. We need the administrator to keep the system intact. The system administrator doesn't use the product like all the users. The system doesn't simplify the work or is practical for the administrator. The administrator is rather part of the system itself. The administrator task is it to for example to add transponders to the system or to let a person in charge of rooms to be responsible for more rooms and therefore more transponders.
To accomplish these tasks, the administrator could tinker around with the systems databases itself because he/she is an expert. But that would be very inconvenient for the admin. Also, the system would be unstable and there is the chance to break something in the system. Therefore it would be helpful if the system would take the needs and goals of the system administrator into account and could provide interfaces (and logic) that helps the administrator. But these things would come with an extra cost for the design and development, so this is also a question of the budget.

[Überarbeitung]

- added introduction

- divided borrowers in different types
- added system administrator
- added conclusion

Personas

[Überarbeitetes Artefakt/e]

“A persona, […] in user-centred design and marketing is a fictional character created to represent a user type…” (Wikipedia: Persona [user experience]).

Personas are depicted as individuals, but they serve as archetypes and therefore represent a class or a type of users with special needs and interaction with the product (c.f. Alan Cooper, Robert Reimann & David Cronin 2010, S. 105).

The personas are going to help us to understand the users that are using the system and to create the system for the user’s needs and goals. To understand the user is critical for the design choices we will make in the design process later.

Here are our personas:

- Andreas Fischer (gatekeeper)
- Prof Dr. Sieglebud (person in charge of rooms)
- Sandra Meyer (master-student)

Andreas Fischer (gatekeeper)

Andreas Fischer works as a gatekeeper at the TH Köln in Gummersbach. In general, he really likes his job. He has a lot to do with people and there is always something going on. But he sometimes struggles with the paperwork he needs to do during his work day. Especially the lending of transponders causes him headaches sometimes. The permission lists need frequent editorial work, and this is very time consuming and error-prone. Also, when there is somebody who wants to borrow a transponder, which happens quite often a day, he need to search in these huge lists and this is causing him stress, because people can be quite impatient sometimes. He works at the faculty for computer and engineering science and in general the building is technically very well equipped, so he wonders why there is no technical system that helps him with his problem. Luckily, he heard about a student project, that tries to come up with a design-solution for his problem. He hopes that the project will move in the right direction and that there will be something soon, that supports him at work, so he can concentrate on more important aspects of his work.

Andreas wants:

- to look up permissions for transponders and the availability of transponders uncomplicated and quickly
- he doesn’t want to manage the permission lists by hand

- he wants a system that is less error-prone and let him quickly lend transponders to people

Prof. Dr. Siegelbud

Prof. Dr. Siegelbud is a professor for communication and network technology at the TH Köln in Gummersbach. This semester he supervises the master-project of some students. The students need access to the laboratories, so he needs to give them permissions to borrow the corresponding transponders that can open the rooms. So, he needs to go down to the gatekeeper, where he needs to authorize and give the gatekeeper the list of students, that need access to the laboratories. He is very busy with other things, so this is quite inconvenient for him. His to-do-list has already stacked up higher than he would normally allow it. It would be much easier if he just could give the permissions straight from his desk.

Prof. Dr. Siegelbud wants:

- to give permissions for transponders (rooms) to people right at his desk (or anywhere)
- to look up the permissions he gave (also at his desk or anywhere)
- to remove permissions in the same easy manner

Sandra Meyer (master-student)

Sandra Meyer is a master-student and currently in her third semester. For her master-project about IT-Security she needs to access the laboratories several times a week. She has spoken with her supervisor Prof. Dr. Siegelbud, who already gave her access. Every time she goes in the laboratories she goes to the gatekeeper in the lobby. There she waits for the gatekeeper to check her permissions and give her the transponder, that let her open the room. Quite often the transponder was already lent, which is impractical for her because the laboratories are in the opposite direction of the lobby, so it always takes her a while to get her there. The borrowing of transponders also takes a while. She feels sorry for the gatekeeper, because he needs to work through some confusing lists, but she also wishes sometimes, that the process would be faster. It's not much time the process takes, but it is stacking up and she really could use this time for her studies.

Sandra wants:

- that the process of borrowing a transponder is quicker
- that she can look beforehand whether a transponder is lent or not

[Erläuterung]

Although there are more than 3 user types, we decided to create only 3 personas, on which we will focus, and which are the most important user types for the system.

The persona, that represents the gatekeeper is Andreas Fischer.

The persons in charge of rooms are represented by the professor Dr. Siegelbud.

And the borrowers are represented by Susi Meyer the master-student.

For the borrowers we chose to only represent the most important of the three borrower-types (students, professors and scientific staff, external, c.f. User Profiles – Borrowers), because we identified the students as the primary users of the system and professors and scientific staff as secondary users, because they use the interface, that we will design, just like the students, but they have further needs, so there are some more requirements.
The external borrowers would belong to *Served Personas*, which are personas-types that don't use the system directly but are affected by the system (c.f. Alan Cooper, Robert Reimann & David Cronin 2010, S. 105), because they don't interact with the system directly unlike the other borrowers, who have an interface for seeing their permissions, and because they are only affected by it during the lending process.
Therefore, we created only one persona 'Susi Meyer' because as a *Primary Persona*, it's the user type we should concentrate on the most.

We also excluded the administrator for the personas, because we decided that we won't focus on him by now, due to time constraints. We know that the administrator should not be neglected, because he/she plays a crucial role for the maintenance of the system and will probably work the most with the system, but as an expert the system administrator will be able to do his/her work without well-designed interfaces specially designed for him. A well-designed interface for the administrator would be very nice, but the admin is not that high in our priorities, to justify the costs. But if enough budget is left, we will definitely create an interface for the admin.

We tried to tie the stories of the personas a bit together, so they are all part of a big picture. So, the professor Siegelbud is the supervisor for Sabine Meyer's master project, and he needs to grant her access to the laboratories. Sabine Meyer show empathy with the gatekeeper because of the complex non-automated system the gatekeeper must deal with, but also shows her dissatisfaction with the current solution that takes up a lot of her valuable time.

[Überarbeitung]

- added introduction
- added explanation

User Stories

[Überarbeitetes Artefakt/e]

A user story is a description of one or more features of a software system, written in the perspective of the user. They are part of the definition of requirements.

They help us to know, what the system should be capable of. they are an excellent method to define our requirements in our user-centred design process, because they also state why a specific function is important for users and what the goal is the user wants to achieve.

Here we created user stories for our three general user types (borrowers, gatekeeper and persons in charge of rooms). The personas we created have special needs and tasks they need to do during their day. Now here is described which needs the system must satisfy and what the users should be able to do to achieve their goals.

Borrowers

1. As a borrower, I can borrow a transponder at the gatekeeper's office without much effort, so I can open rooms with it. I need to have access to the rooms to access tools and equipment I need to carry out my responsibilities (projects, assignments, preparation of lectures, …) and to have an environment in which I can work and do my job.

2. As a borrower, I can look beforehand whether a transponder is already lent or not, so I save time and I only borrow a transponder if it is available.

3. As a borrower, I can view a list of all permission for borrowing transponders and the rooms I can open, so that I have an overview which transponder I can borrow which helps me to plan my actions.

4. As a borrower, I can view details for each permission, that is granted to me, with information like which rooms I can open with the transponder, when the permission was granted, who granted the permission, when it will expire, so that I can use this information to plan, e.g. I know that the permission will expire in two weeks and I can renew the permission.

5. As a borrower I can quickly check whether I have a permission for a specific transponder / room or not, so that I can ask for a permission if I haven't one already.

6. As a borrower I can ask for a permission for a transponder, so that I have the chance to borrow transponders.

7. As a borrower I can ask for a permission for a single room and then get the permission for all the transponders, that can open the room, so that I have access to all possible transponders that can open the room, so that I just can take another transponder that is available when the other is not currently available and I don't need to ask for every single transponder for that room manually.

8. As I borrower I get a message when a request for a permission was accepted or denied, I got a new permission, or a permission was removed, so that I am informed, and I always know what changed, so that I can adjust to the situation.

9. As a borrower I can see my open requests for permissions with the time I requested them, so that I know which permissions for room / transponders I requested and I can complain to the persons in charge of the rooms, when it takes too long.

Gatekeeper

1. As a gatekeeper, I can check the availability of a specific transponders, so that I can give the person, who wants to borrow the transponder, information, so they know whether it is already lent or not.

2. As a gatekeeper, I can check whether a person has a permission to borrow a certain transponder or not, in an easy and uncomplicated way, so that the process is quick, the person doesn't need to wait long and can get the transponder and I can go on with my work.

3. As a gatekeeper, I can lend transponders to people without much administrative effort, so I don't have to manage huge lists and the persons who want the transponders are happy.

4. As a gatekeeper, I can view a list of all transponders that are currently lent with additional information like the borrower's name and the time when the borrower needs to give the transponder back, so that I have an overview about the lent transponders and can contact the authority when a transponder is missing for too long or is not brought back.

Persons in charge of rooms

1. As a person in charge of rooms, I can give/remove permissions to other people for borrowing transponders to open the rooms I oversee, in an easy and quick way, so I don't have to go anywhere to do this and persons I trust, can open the rooms, I am responsible for.

2. As a person in charge of a rooms, I can also give/remove permissions to a group of people that share common traits like students studying *this* in *that* semester or students attending a special lecture or course, so that I don't have to grant permissions for every single person of a group which would be annoying and time-consuming.

3. As a person in charge of rooms, I can view a list of persons to whom I granted a permission, so that I have an overview which persons can access the rooms, I am responsible for.

4. As a person in charge of rooms, I can view details for each permission I gave, like which rooms can be opened by the transponder, when a permission was granted or when it will expire, so that I can use this information to plan and to reason my decisions.

5. As a person in charge of rooms, I can quickly find a permission I gave, so that I can quickly access it and can then edit or remove it (or do other stuff with it).

6. As a person in charge of rooms, I can view open permission-requests which I can then either accept or deny or ignore in which case it is denied automatically after a specific amount of time, so that I can react on permission-requests which is one of my responsibilities as a person in charge of a rooms.

7. As a person in charge of rooms, I can view the rooms and the corresponding transponders I have the responsibility for, so I know what I am responsible for.

8. As a person in charge of rooms, I can give the transponders, which I manage, custom names, so that I can identify transponders with their names and not the transponder number, which is hard to memorize. In this way I can find single transponders more quickly.

[Erläuterung]

Here we can see which role every type of user has and what functionality they need in the system, to achieve their goals.

For the lenders, we didn't differentiate between the different types of lenders (see User Profiles: Borrower), because the different user types have very similar goals. The main focus is on the primary users, the students. Then we extended the user stories for the secondary users (for borrowing transponders), the scientific staff and professors. The difference between the students and the scientific staff and professors is that the second group have also the need for borrowing special transponders, that can open multiple rooms. (e.g. a transponder that can access all the rooms of the moxd-laboratory). The students do not really have this use case, because they only want to access only one (or more) specific rooms and not like the scientific staff, who need to have access to a room complex to prepare stuff there, like installing new software on the computers.

The gatekeepers just want a system, that makes the lending of transponders easier. So they need a system, that checks the permission and the availability of transponders, so that they do not have to check these things manually by looking through huge lists. Furthermore, they want to have an overview over the currently lent transponders, so that they can inform their authority if something is not right, e.g. a transponder is missing for too long.

The persons in charge of room need to manage the permissions for room / transponders for which they have the responsibility for. So they can view, add, remove and edit the permissions. They can also accept and refuse permission requests and name specific transponders.

The users stories gave us a great insight into which problems the system we design needs to solve and which functionalities it needs to have to accomplish the user goals.

What the user stories don't show is the context and in which relation the user stories stand to each other. But this problem is solved by the next step the *Hierarchical Task Analysis*, that will show of which smaller tasks a feature is composed.

[Überarbeitung]

- added user stories for borrowers 3-9 (they were necessary for the user goals)
- added user stories for gatekeepers 3-4 (they were necessary for the user goals)

- added user stories for persons in charge of rooms 2-8 (they were necessary for user goals)
- added introduction
- added conclusion

Top Level User Tasks

[Überarbeitetes Artefakt/e]

The Top Level User Tasks should give us a comprehensive insight into the main tasks our users will try to fulfil when using our application. Each Top Level User Task represents the root of a task tree in the Hierarchical Task Analysis. All tasks combined should make every User Story possible to accomplish.

0. borrow transponder
1. give/remove permission
2. verify borrower's permission
3. check for available transponder
4. check permission for room/transponder
5. ask for permission

[Erläuterung]

This list is a very raw representation of our users' tasks, but gives us a great starting point for our Hierarchical Task Analysis.

[Überarbeitung]

- added description
- added conclusion
- added tasks 3 – 5

Hierarchical Task Analysis

[Überarbeitetes Artefakt/e]

Hierarchical Task Analysis

Description

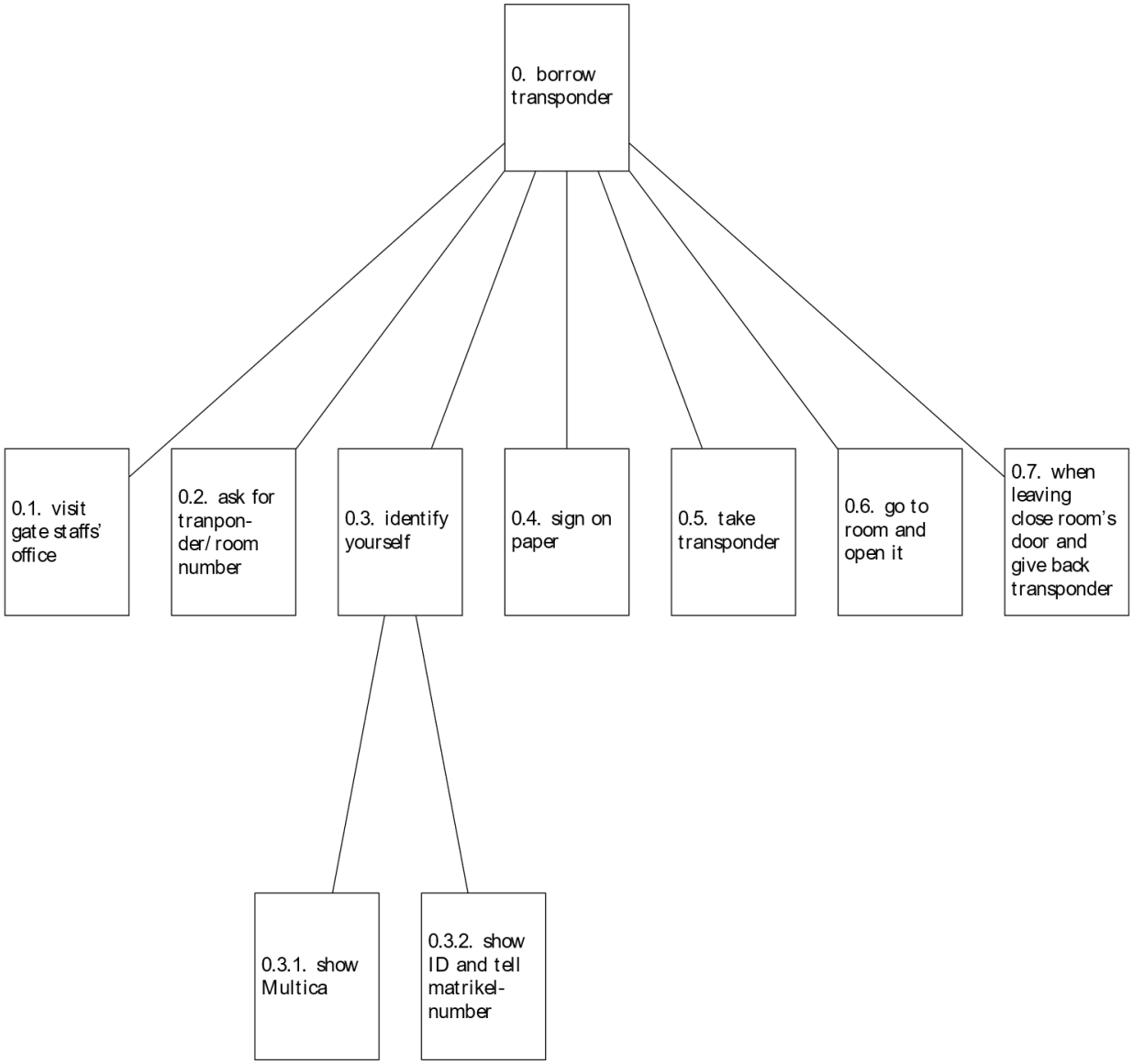
Our Top Level User Tasks decomposed in smaller, hierarchical tasks/subtasks and actions a user currently performs/will perform using the current system based on paper sheets/our application.

We made graphs for both, the current state (application based on paper sheets) and our target state (automated application) of how tasks are done.

We found that using the Hierarchical Task Analysis outside of it's supposed use case (only modelling the current state of how tasks are done) was very insightful for us, because it allowed us to specify plans, which helps us to model state (e.g. transponder not available, borrower has no permission), conditions (e.g. only if borrower has permission he can borrow a transponder, else the gatekeeper resents the request) and descisions (e.g. borrower decides to use his/her Multica as identification), which should make it easier to implement the different tasks/subtasks in a safe and useful manner later in the development proccess.

0. Borrow transponder

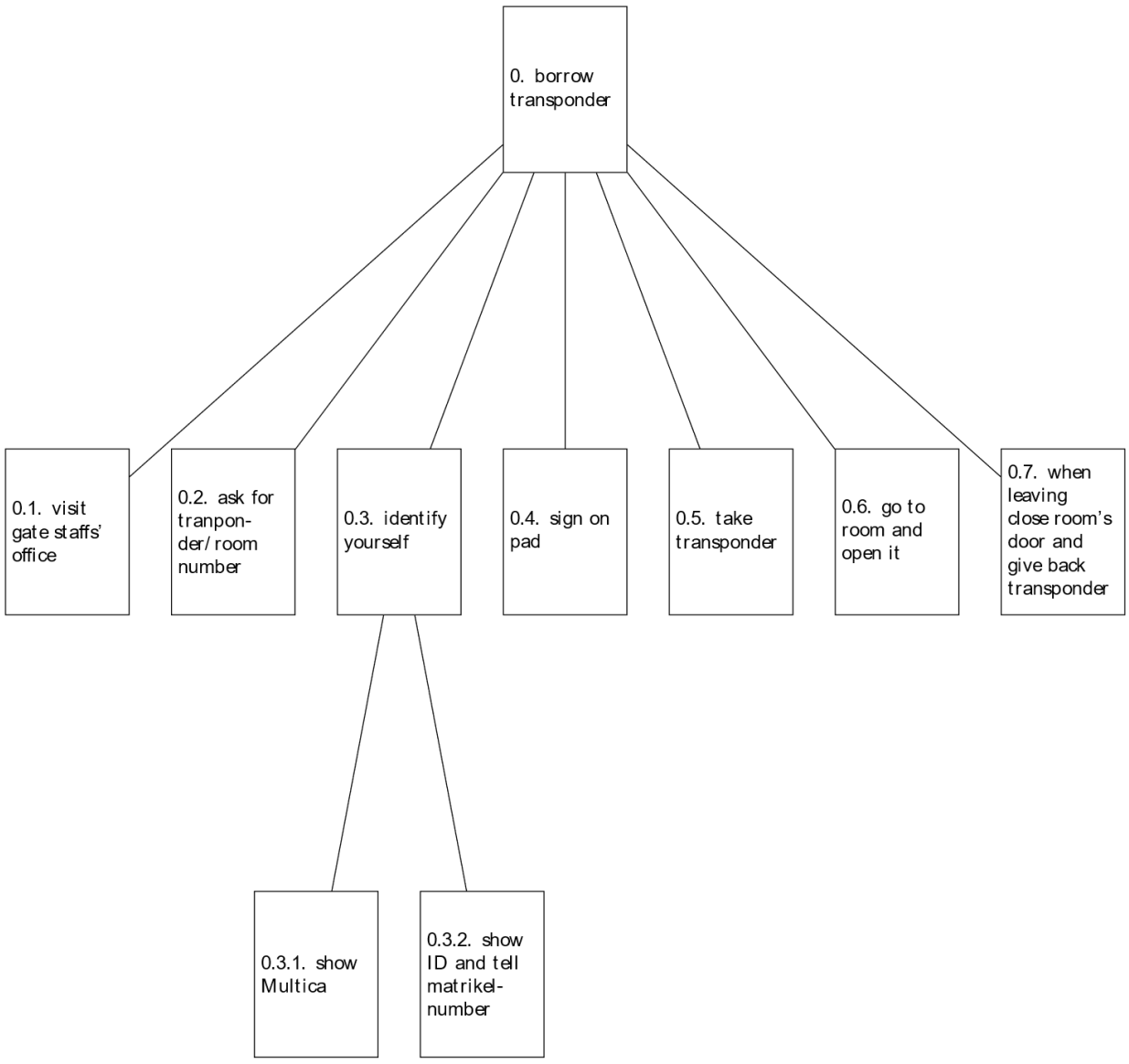
Current state



Plan 0.3.1 current: do 0.1-0.2. If you choose to identify with your Multica do 0.3.1 and continue with 0.4.

Plan 0.3.2 current: do 0.1-0.2. If you choose to identify with your ID do 0.3.2 and continue with 0.4.

Target state



Plan 0.3.1 target: do 0.1-0.2. If you choose to identify with your Multica do 0.3.1 and continue with 0.4.

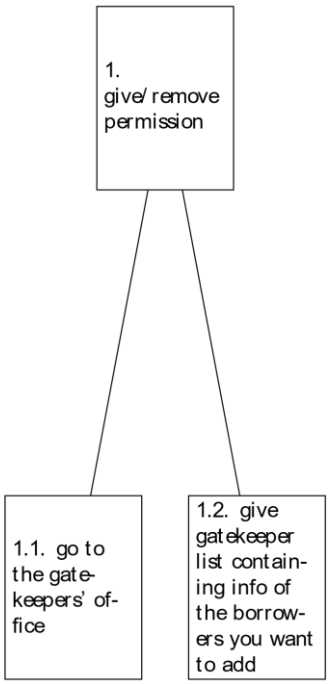
Plan 0.3.2 target: do 0.1-0.2. If you choose to identify with your ID do 0.3.2 and continue with 0.4.

Description

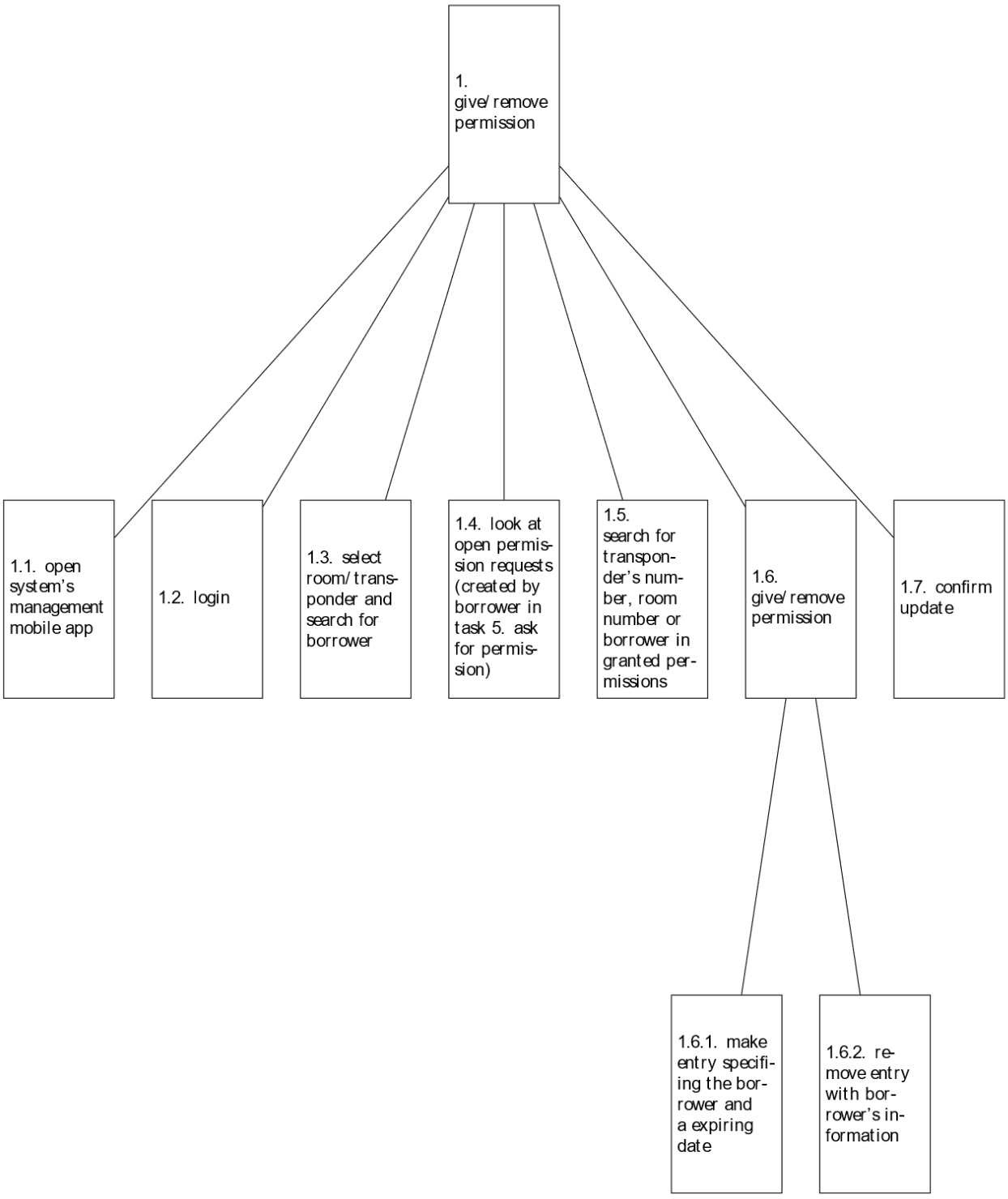
The main task a borrower does. He/she wants to use this system to gain access to a room he/she has permission to use.

1. Give/remove permission

Current state



Target state



Plan 1.3/4 target: do 1.1-1.2. If you wnat to give permssion do either 1.3 or 1.4. Continue with 1.6.1 and after that 1.7.

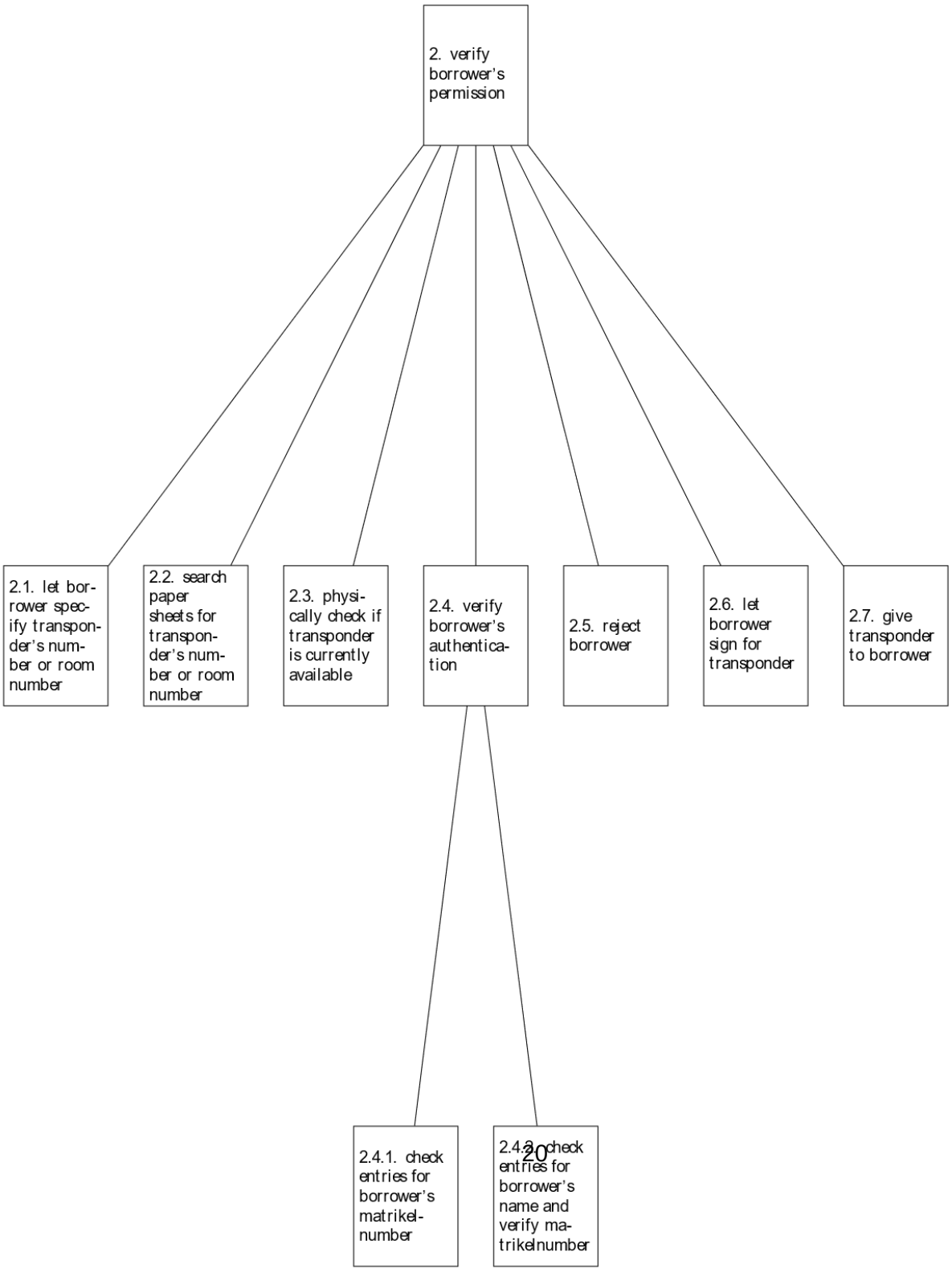
Plan 1.5 target: do 1.1-1.2. If you want to remove permission do 1.5 and continue with 1.6.2.

Description

If a person in charge of a room or transponder wants to give/take permission to/from a borrower he/she has to complete this task.

2. Verify borrower's permission

Current state



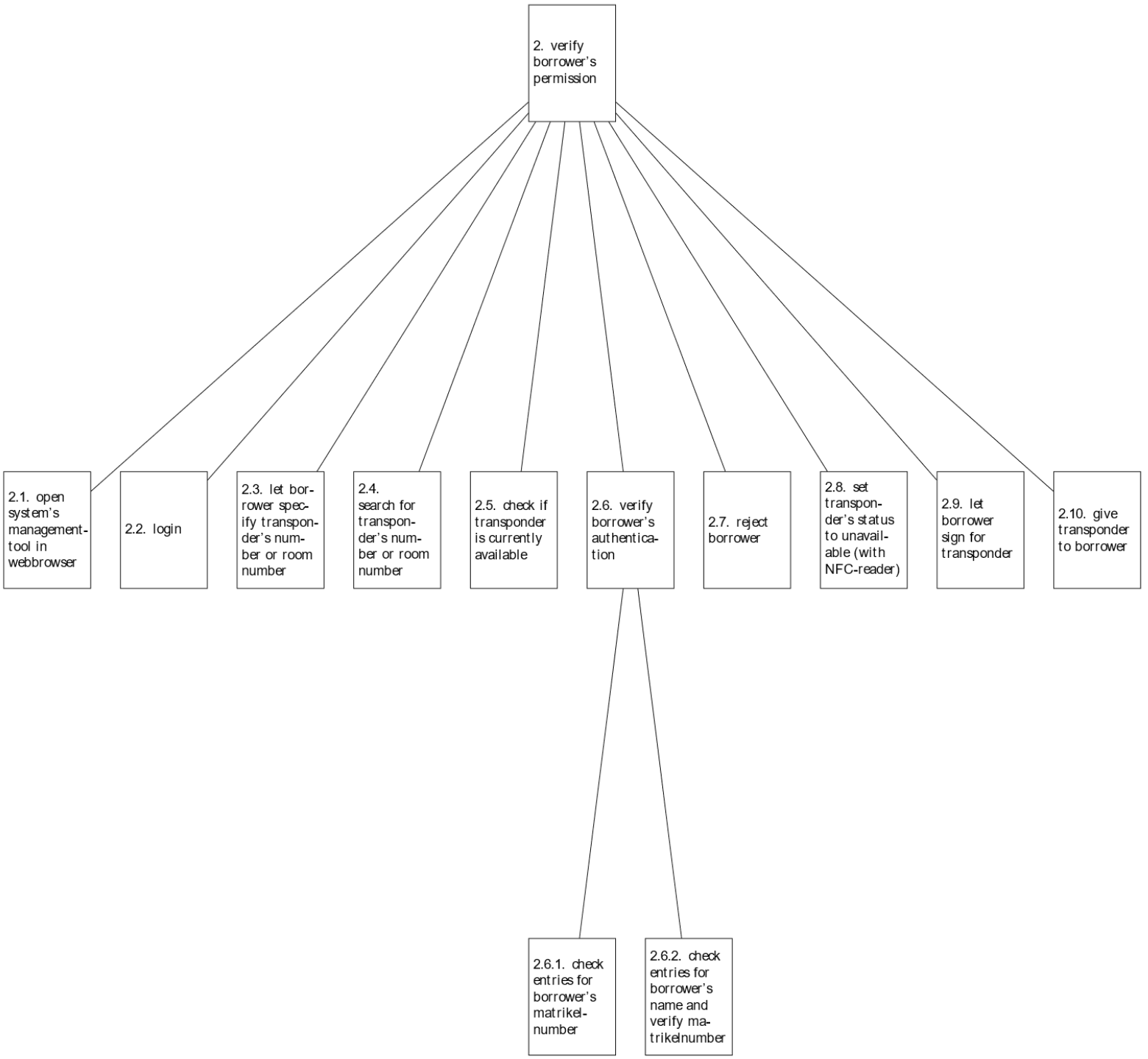
Plan 2.3.1 current: do 2.1-2.3. If transponder is available continue with 2.4.

Plan 2.3.2 current: do 2.1-2.3. If transponder is not available do 2.5.

Plan 2.4.1 current: do 2.1-2.3. If the borrower uses his Multica as identification do 2.4.1 and continue with 2.6.

Plan 2.4.2 current: do 2.1-2.3. If the borrower uses his ID as identification do 2.4.2 and continue with 2.6.

Target state



Plan 2.5.1 target: do 2.1-2.5. If transponder is available continue with 2.6.

Plan 2.5.2 target: do 2.1-2.5. If transponder is not available do 2.7.

Plan 2.6.1 target: do 2.1-2.5. If the borrower uses his Multica as identification do 2.6.1 and continue with 2.8.

Plan 2.6.2 target: do 2.1-2.5. If the borrower uses his ID as identification do 2.6.2 and continue with 2.8.

Description

This task is done by a member of the gatekeeping staff. It basically runs parallel to the borrower's task "0. borrow transponder".

Only for target state:

Since one of our main goals is to keep the university's property safe, this task of verification is utterly important. We can not allow that somebody permissionless, unverified or without the system's/the person in charge of the room's knowledge gains access to a room.

This is the reason why every borrowing process gets protocolled, so we can always keep track of who has and has had a transponder. Now, if some damage gets reported, we always know who had access during the aproximate time period the damage was done.

We want to automate every task as much as possible and make the task for every user as seamless as possible, too, so we thought it would be easiest if we would introduce NFC (Near-Field Communication) to this particular task.

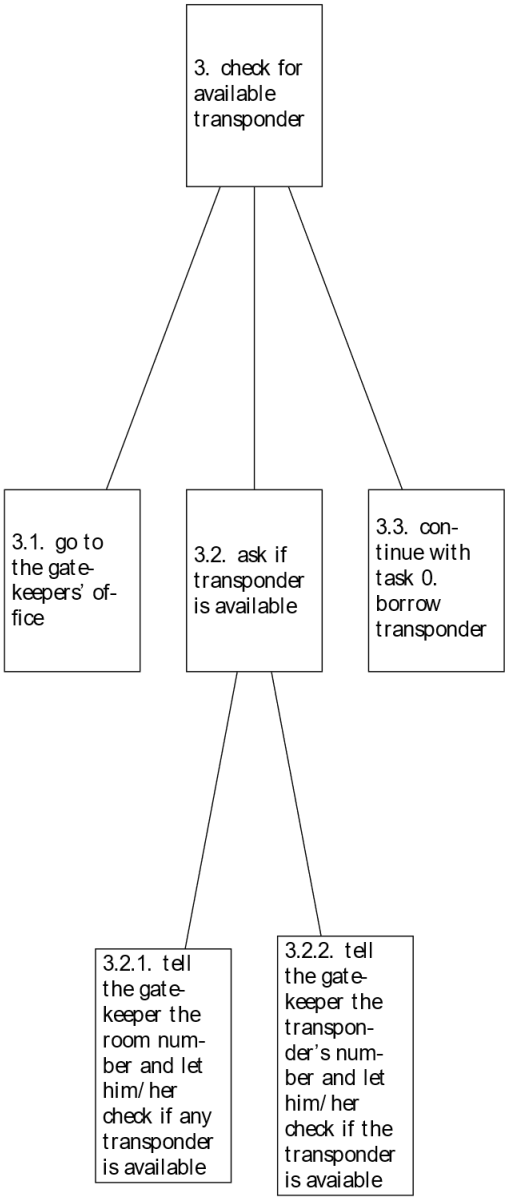
We want a NFC-reading device connected to the workstation the gatekeeper uses, which we can access with our application running on this workstation. Now, if the gatekeeper hands out the transponder, he/she just has to hold it against the NFC-reading device and our app knows which transponder was just lent. If the borrower used his Multica to identify himself he used the same NFC-reading device, so our app can combine these two information and we know which user borrowed the transponder. The gatekeeper doesn't has to make interactions with our application interface that could lead to errors, for example giving out a different transponder than the one selected in the application.

If the borrower used his ID for identification, the gatekeeper presses a button on his interface which tells the app who is the borrower. After that he holds the transponder against the NFC-reading device and the the app knows which transponder was lent.

We think this is easier, faster and safer than the gatekeeper having to always manually specify the transponder in his application interface opened in his browser.

3. Check for available transponder

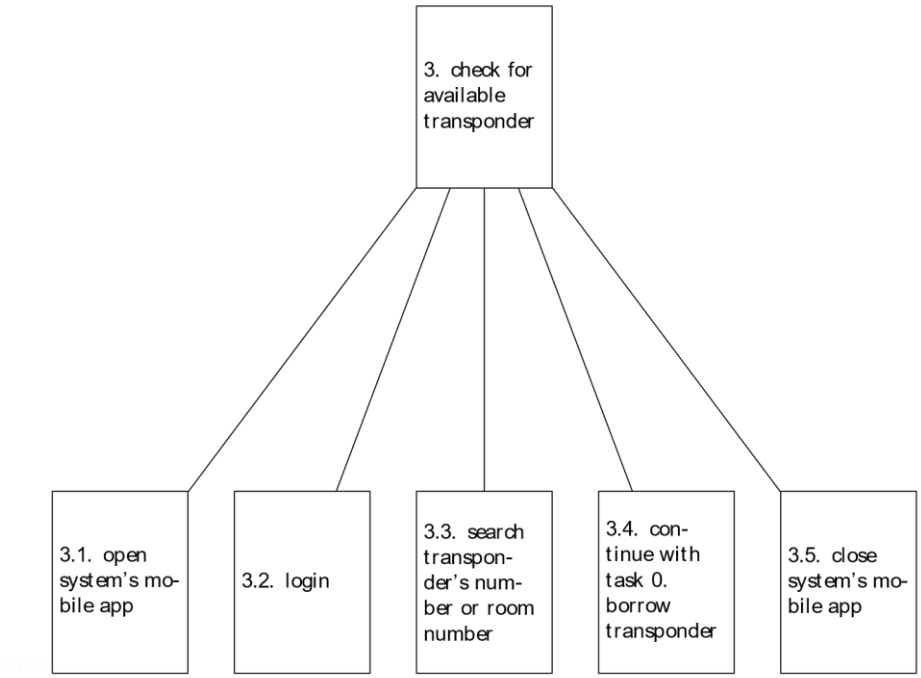
Current state



Plan 3.2 current: do 3.1. Continue with asking for the room number (3.2.1) or the transponder's number (3.2.1).

Plan 3.3 current: do 3.1-3.2. If transponder is available continue with 3.3, else terminate task.

Target state



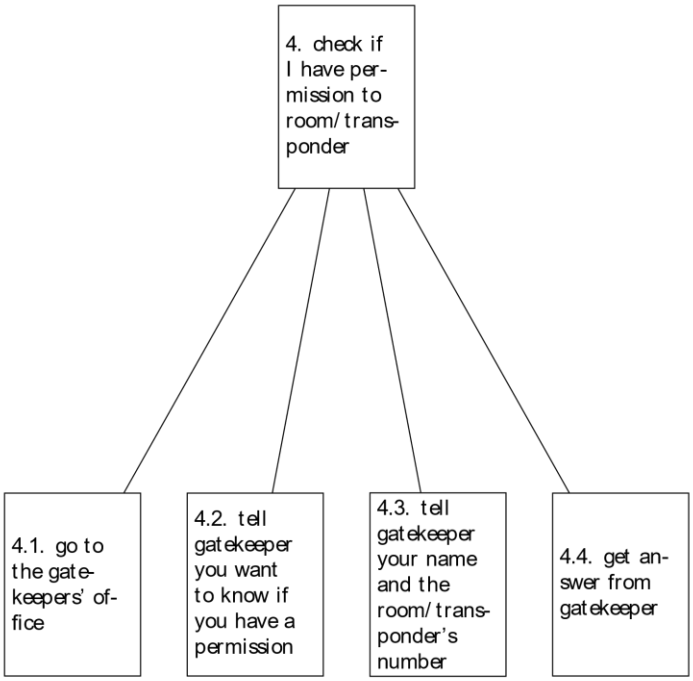
Plan 3.4 target: do 3.1-3.3. If transponder is available continue with 3.4, else continue with 3.5.

Description

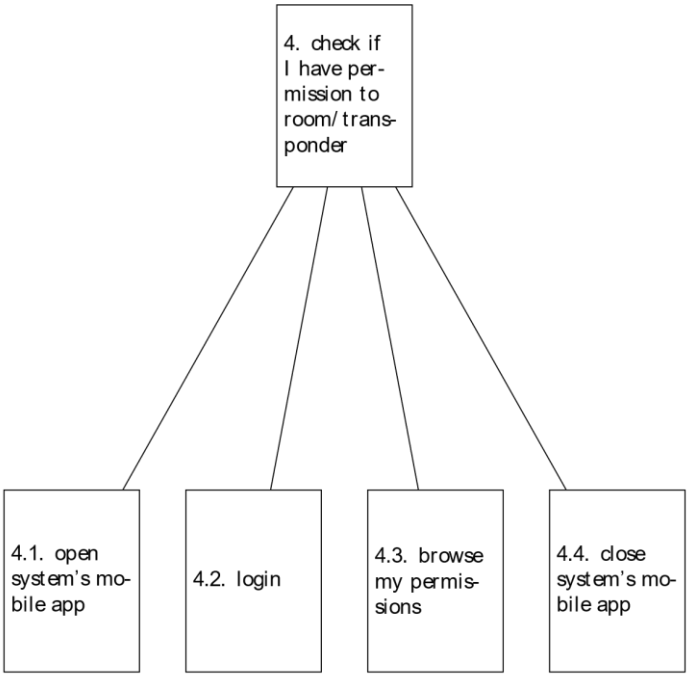
A borrower can check if his desired transponder or a transponder to his desired room is available. While this task is very crucial for our target state application, in our current state it only resembles a subset of task 0. borrow transponder.

4. Check permission for room/transponder

Current state



Target state



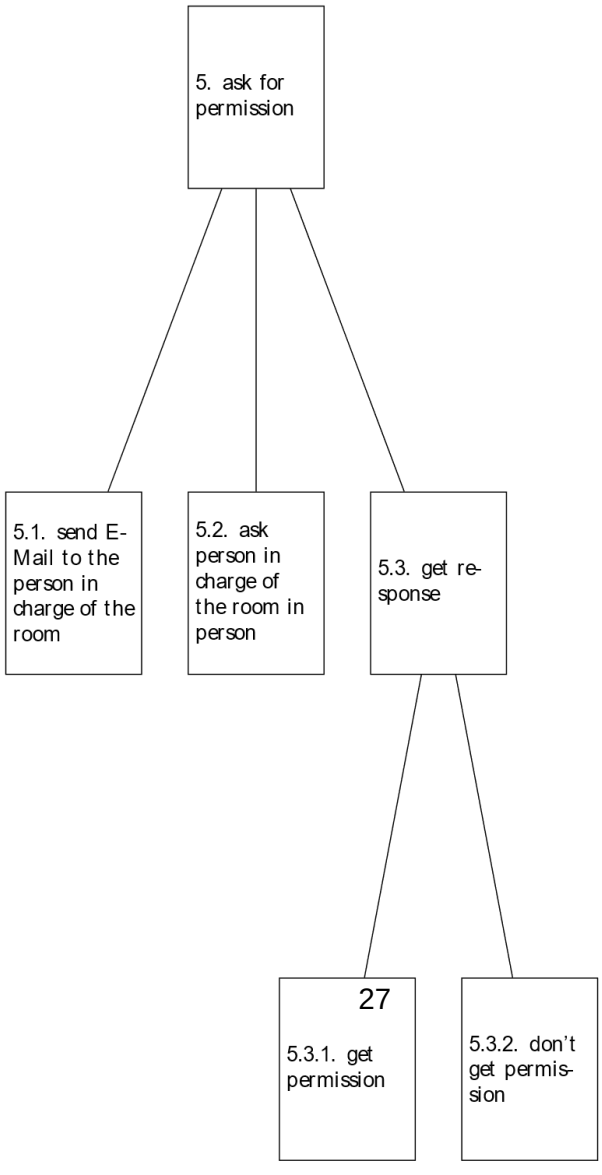
Description

While, again (like task 3. check for available transponder), this task is not very reasonable in our current state application, it will simplify the users' interaction in our target state application hugely. In our current state, this would again be a subset of task 0. borrow transponder and would require two parties (the gatekeeper and the borrower) to interact with each other and the system.

In our target state application the borrower only has to use his smartphone and can check, before having to go to the gatekeepers' office, if he has permission or not (e.g. if he asked for permission (task 5) beforhand and now wants to know if the person in charge of the room has granted it).

5. Ask for permission

Current state

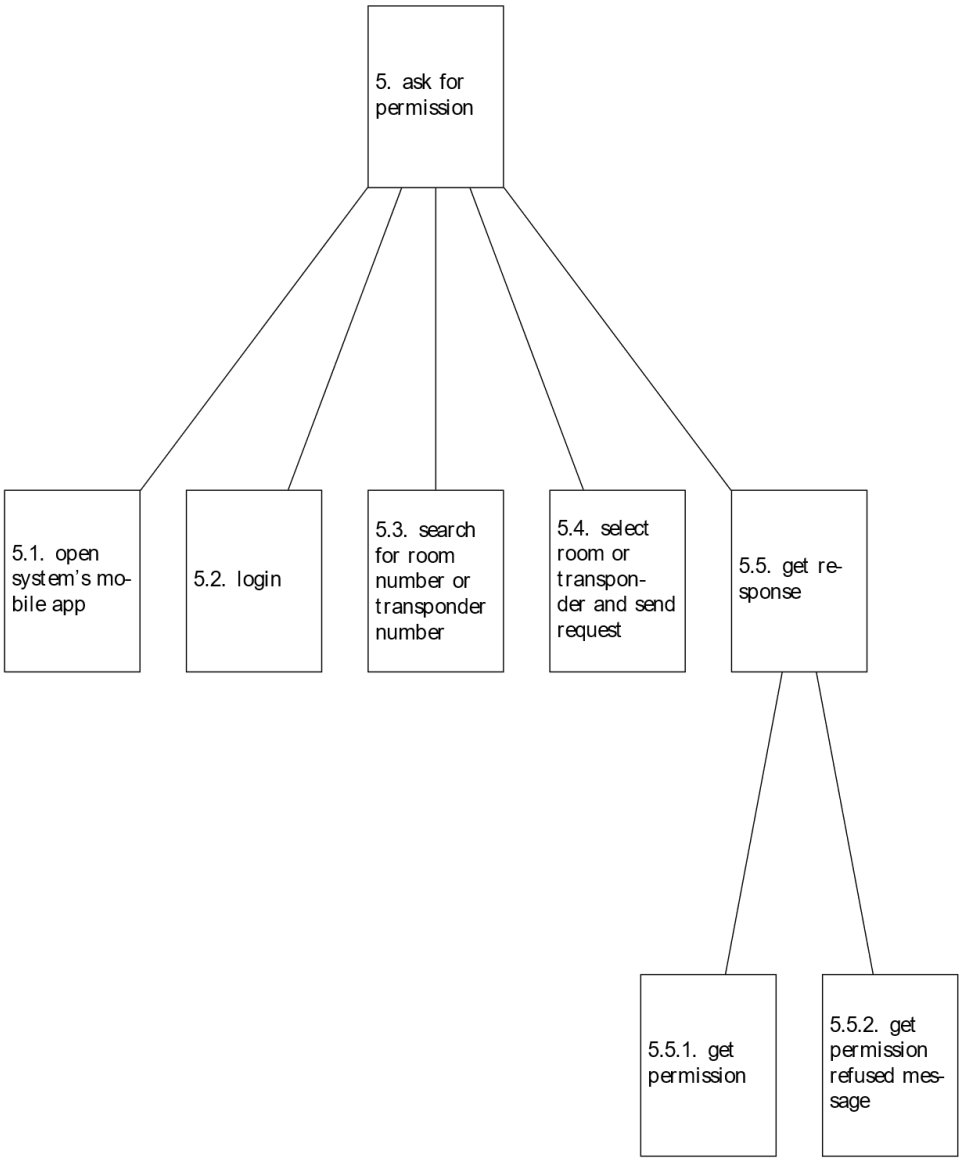


Plan 5.1/2 current: do 5.1 or 5.2. Continue with 5.3.

Plan 5.3.1 current: do 5.1/5.2. If permission is granted continue with 5.3.1.

Plan 5.3.2 current: do 5.1/5.2. If permission is refused continue with 5.3.2.

Target state



Plan 5.5.1 target: do 5.1-5.4. If permission is granted continue with 5.5.1.

Plan 5.5.2 target: do 5.1-5.4. If permission is refused continue with 5.5.2.

Description

A borrower can ask for permission. The person in charge of this room or transponder can either grant (would continue with task 1. give/remove permission) or refuse.

[Erläuterung]

The Hierarchical Task Analysis gave us a great entry point and some insights to how our application could be structured (it's components, interfaces, hardware, technologies, etc.).

[Überarbeitung]

- Added descriptions and a conclusion
- Added tasks 3 - 5
- Added current state

Rapid Prototyping

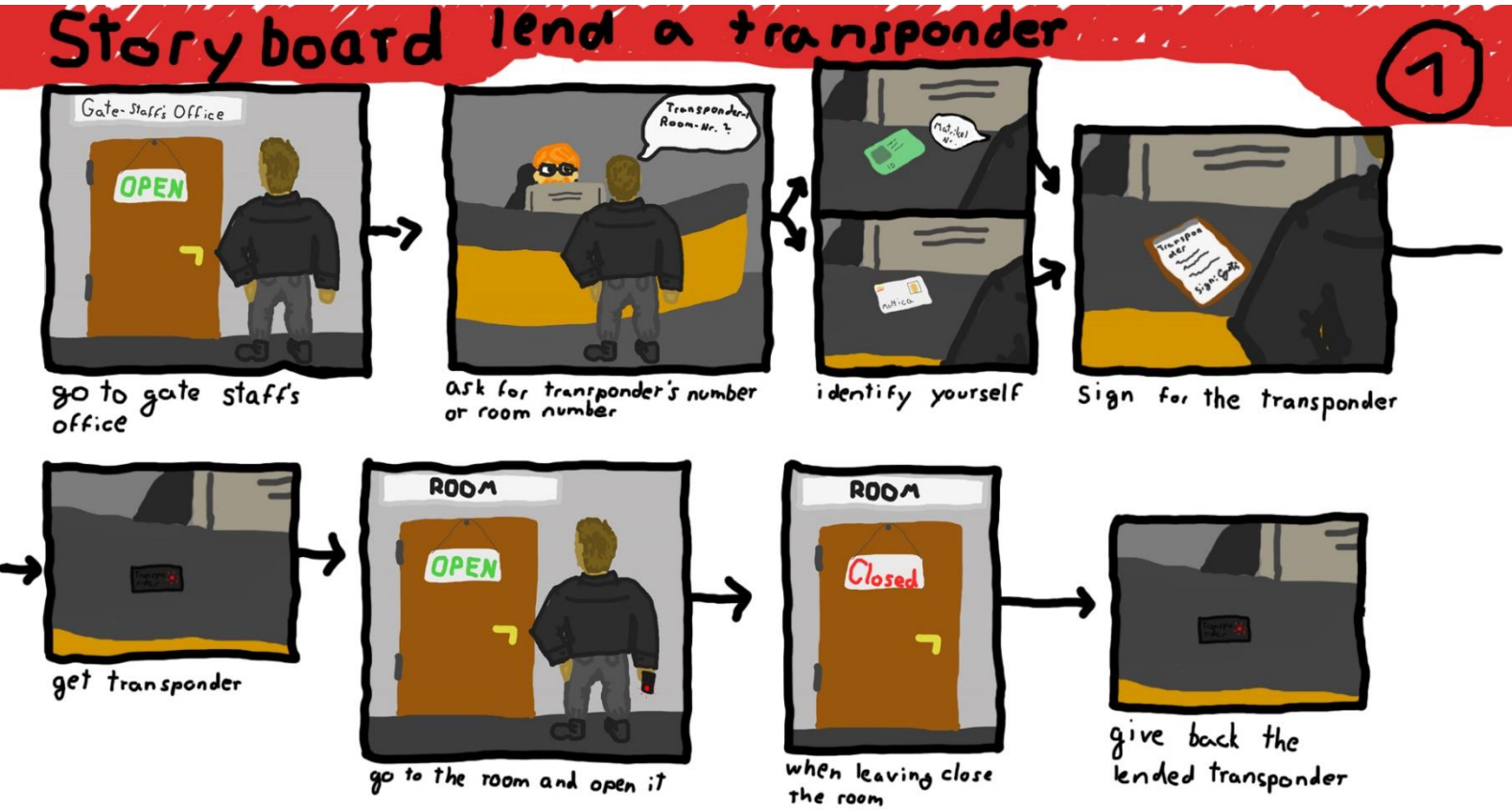
Story Board

[Artefakt/e]

They story board is first sketch about how we imagine the user interfaces of our system, so that the users can accomplish their goals in the easiest and most convenient way.

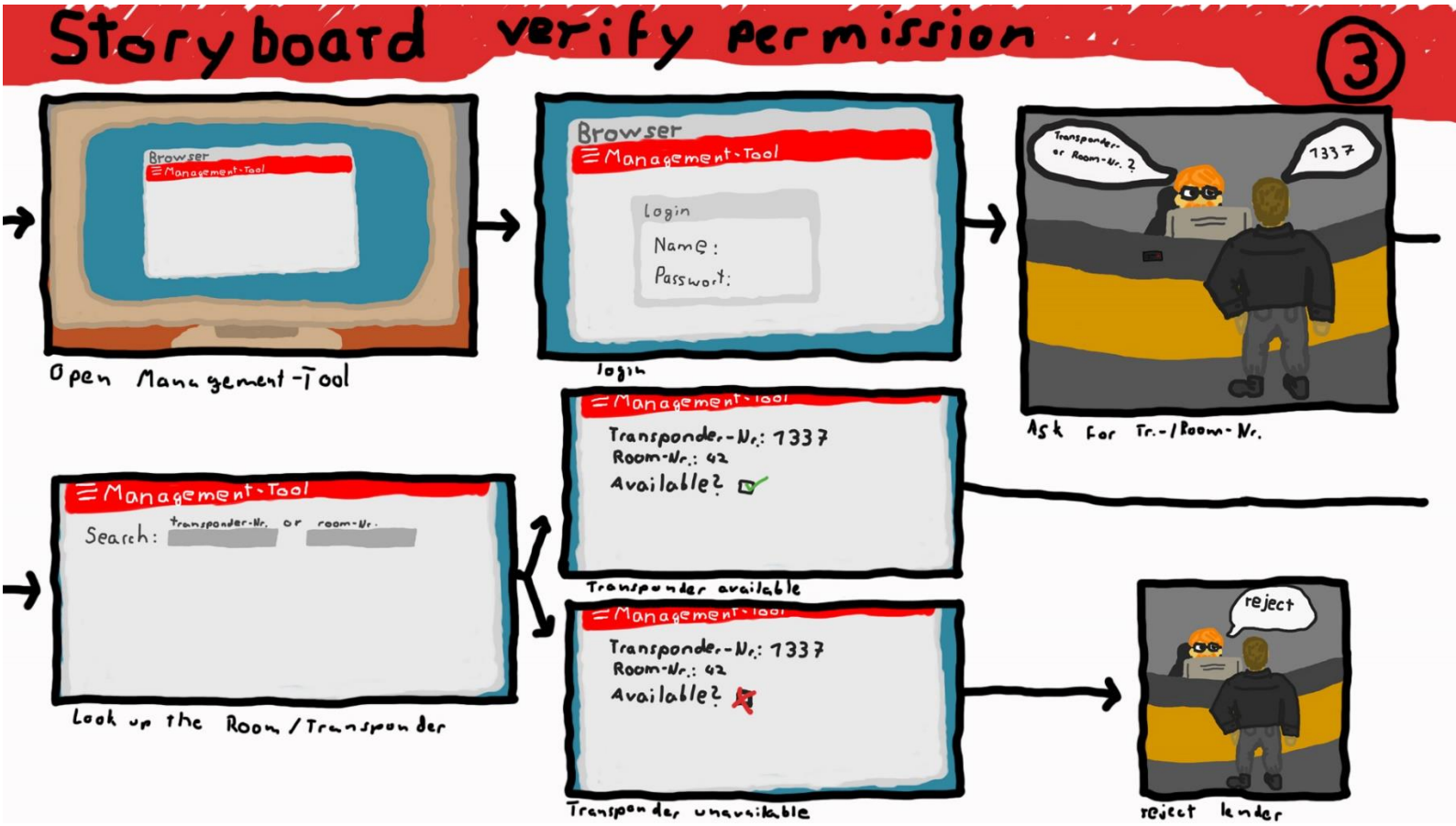
For each top level task, we created a story board, except for the last 3 top level tasks (check for available transponder, check permission for room/transponder, ask for permission), which we managed to fit in only one story board.

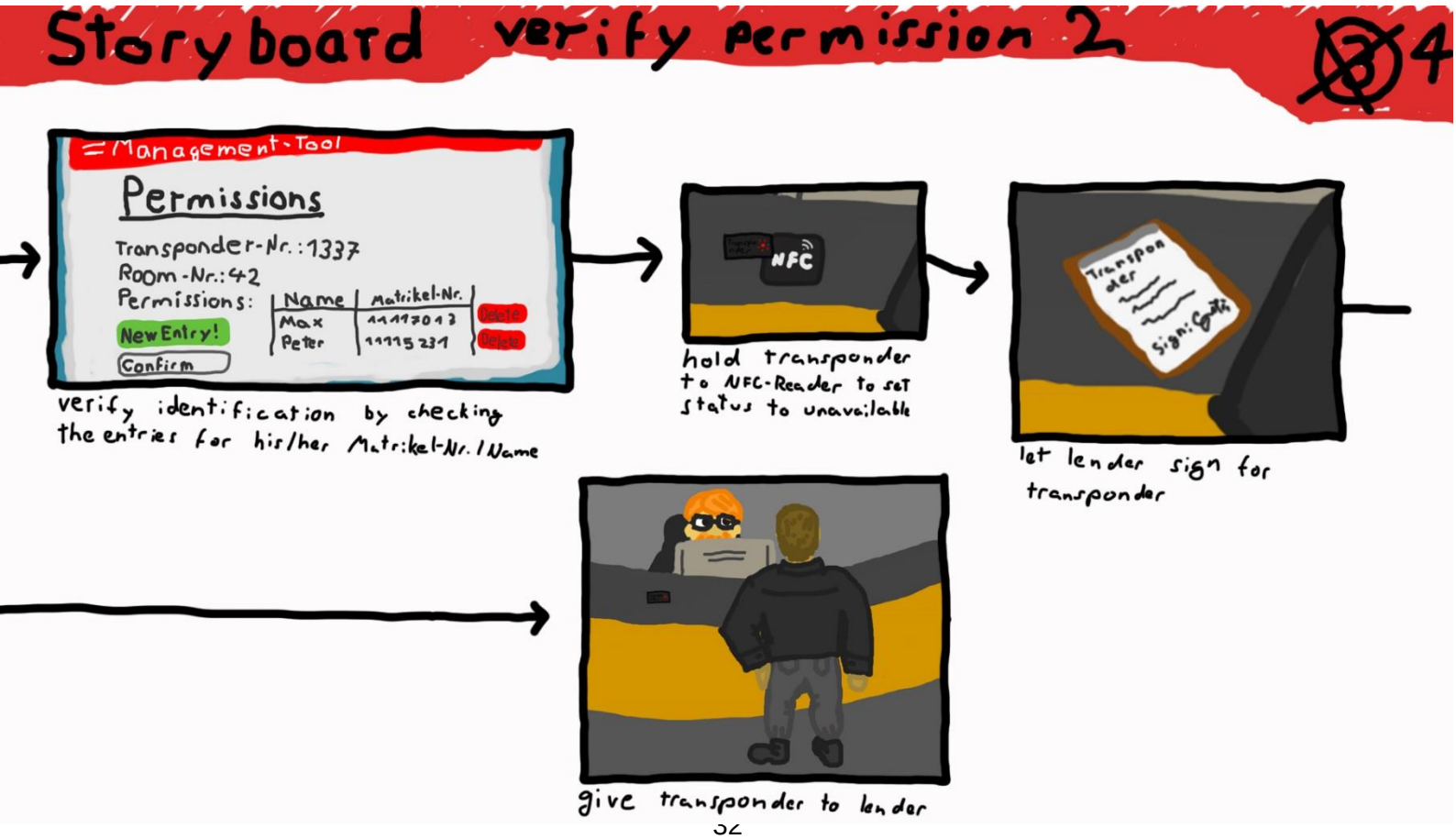
0. borrow a tranponder



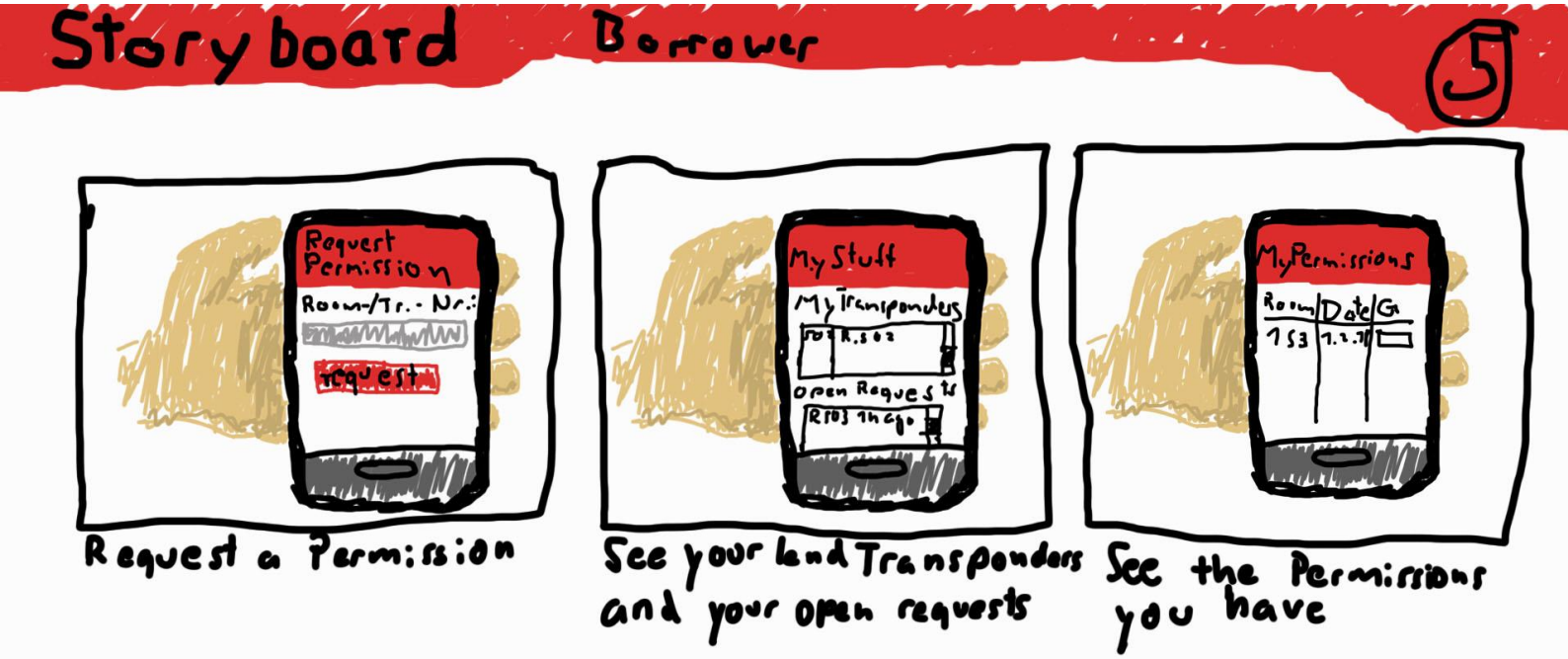
1. give / remove permission

2. verify borrower's permission





3. check for available transponder & 4. check permission for room / transponder & 5. ask for permission



[Erläuterung]

Here you can see our first draft of the interfaces of the transponder-lending-system.

The gatekeeper has a web-interface where he can see, if the transponders is available and if the borrower has the required permission for the transponder. The system also works with a NFC-device which automatically sets the status of the transponder when borrowing it.

For the persons in charge of rooms we decided, that an app would work best them, because they are busy and they don't have much time (c.f. Personas), so that they can give / remove permissions everywhere. In the app they can see their given permission and accept and decline requests for permissions they got.

For the borrowers we also decided for an app, because they are mostly students, who are in general experienced smartphone users, and we wanted the students to have also the benefit

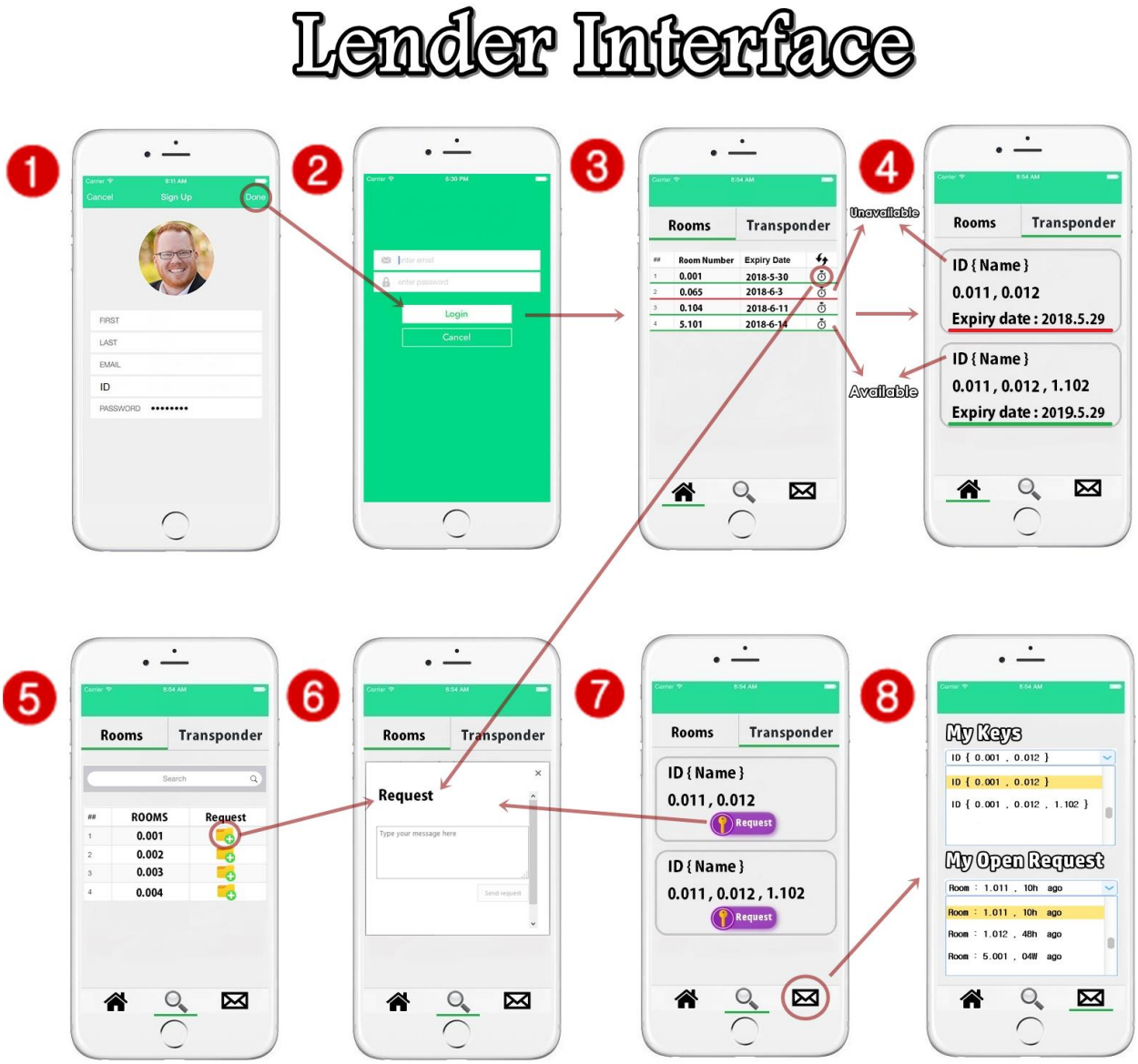
to use the system everywhere to every time. In the app the borrowers can see their permissions (Task 4) and in the same view they can also check if the transponder/s belonging to the permission is currently available or not (Task 3). They can also ask for new permissions by sending request to the persons in charge of room per the app.

Wireframes

Here we show the wireframes and explain the screens and the design decisions we made for each relevant screen.

[Artefakt/e]

1. borrower interface



1. the registration screen:

The registration screen is optional, because the users of the system don't need to register, because they use their university account.

2. login screen

3. permission for rooms

This is the first view, the user sees after the login. Here the users can see all their permissions for borrowing transponders for rooms. They can see when a permission will expire, and they have also the option to renew the permission with the clock symbol. When they press the clock symbol, the next view is the view where they can make a request.

4. permissions for transponders

There are special transponders, that can open multiple rooms, so a view where all permissions for single transponders can be viewed was necessary. The view is like the view for the rooms, just like for the transponders and there is also the information which rooms the transponder can open. We made this distinction also in view 5 and 7, where the users can search for rooms (5) and transponders (7) and make a request for permissions. We used the same top menu there for consistency, so that the users don't lose the orientation.

5. search for rooms

Here the user can search for rooms and request a permission for borrowing a transponder, when clicking the add symbol. The search bar is a smart search bar and the user can search by different search expressions like the room number or the name of the person in charge of room.

6. make a request

Here the user can make a request. They can add their custom message to the request, that will be shown to the person in charge of room. The thought was, that the message can also be used to add a password, on which the professor and student agreed in the lecture, so that the professor (the person in charge of rooms) can directly see the context of the request in the message which helps him to make the decision of accepting or refusing.

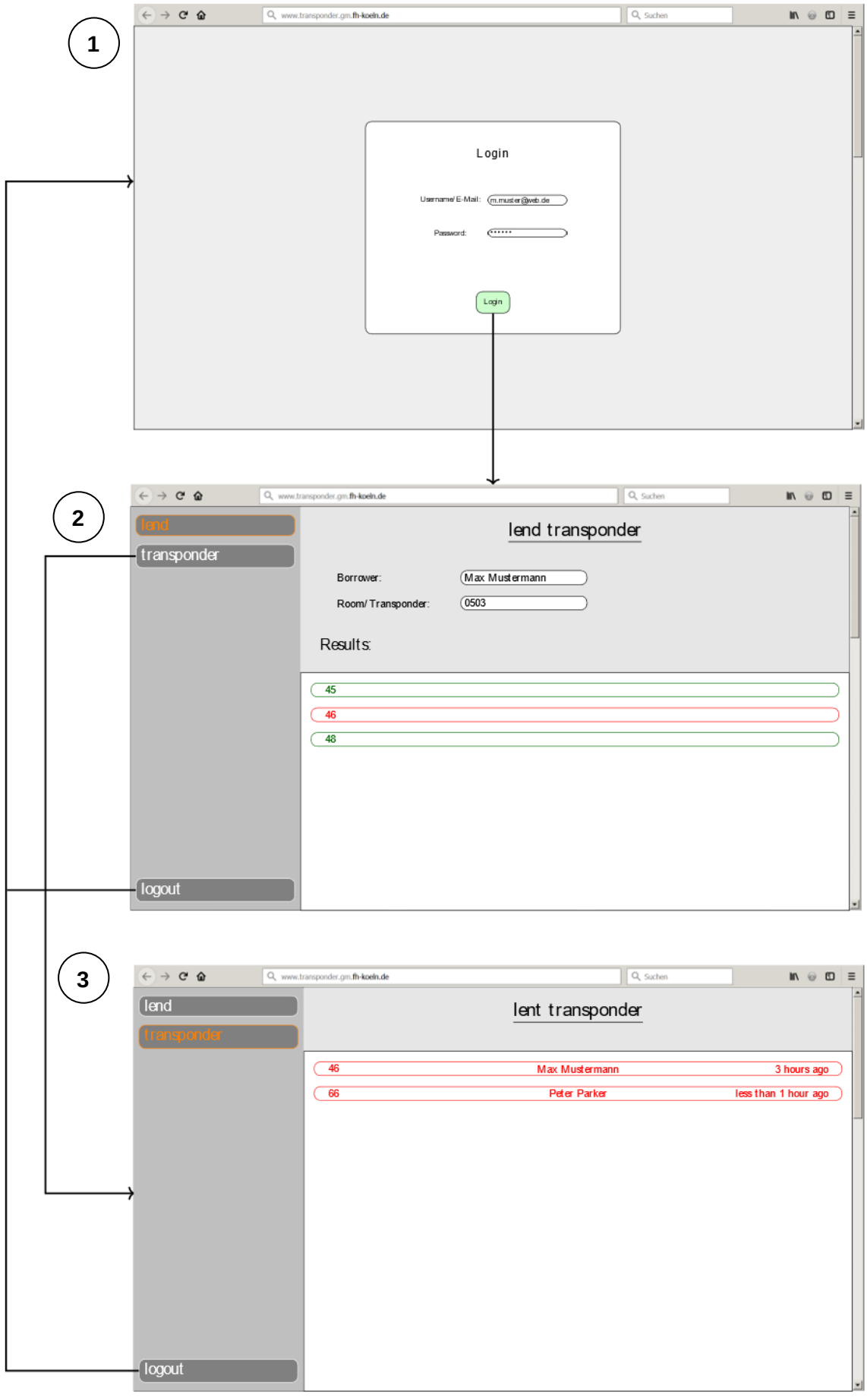
7. search for transponder-(numbers)

This is just like the view for searching rooms (5), just for single transponders, where the users can also search for special transponders, that can open multiple rooms. The search bar for searching is not visible on the picture, but it should be one just like in the view 5. The search bar is intelligent and can search transponders based on multiple different search expressions like transponder-, room numbers and the name of a transponder. So the design is very minimalistic and smart, which fits perfect to our persona (the student Susi Meyer), who has not much time and want to get quickly things done. When the user has found the desired transponders, he/she then can make a request by clicking on the request button.

8. borrow transponders and open requests

This view shows to different things. Firstly (below the header *My Keys*), the currently borrowed transponders, so that the users always know, which transponders they have borrowed at the moment. Secondly (below the header *My Open Requests*), the open permission requests, where the users can also see when the permission request was made, so that they have an overview over their made requests.

2. gatekeeper interface



This interface is for lending transponders. After the login screen the gatekeeper can see the page for lending a transponder. If a borrower comes and wants to borrow a transponder, the gatekeeper uses the searchbar "Borrower" to search for the borrower's name, username or matrikelnumber. The gatekeeper will see a dropdown view under the searchbar containing borrowers that match the current searchbar content (e.g. Google search).

The gatekeeper can click on one entry and will see every permission the borrower has under "Results".

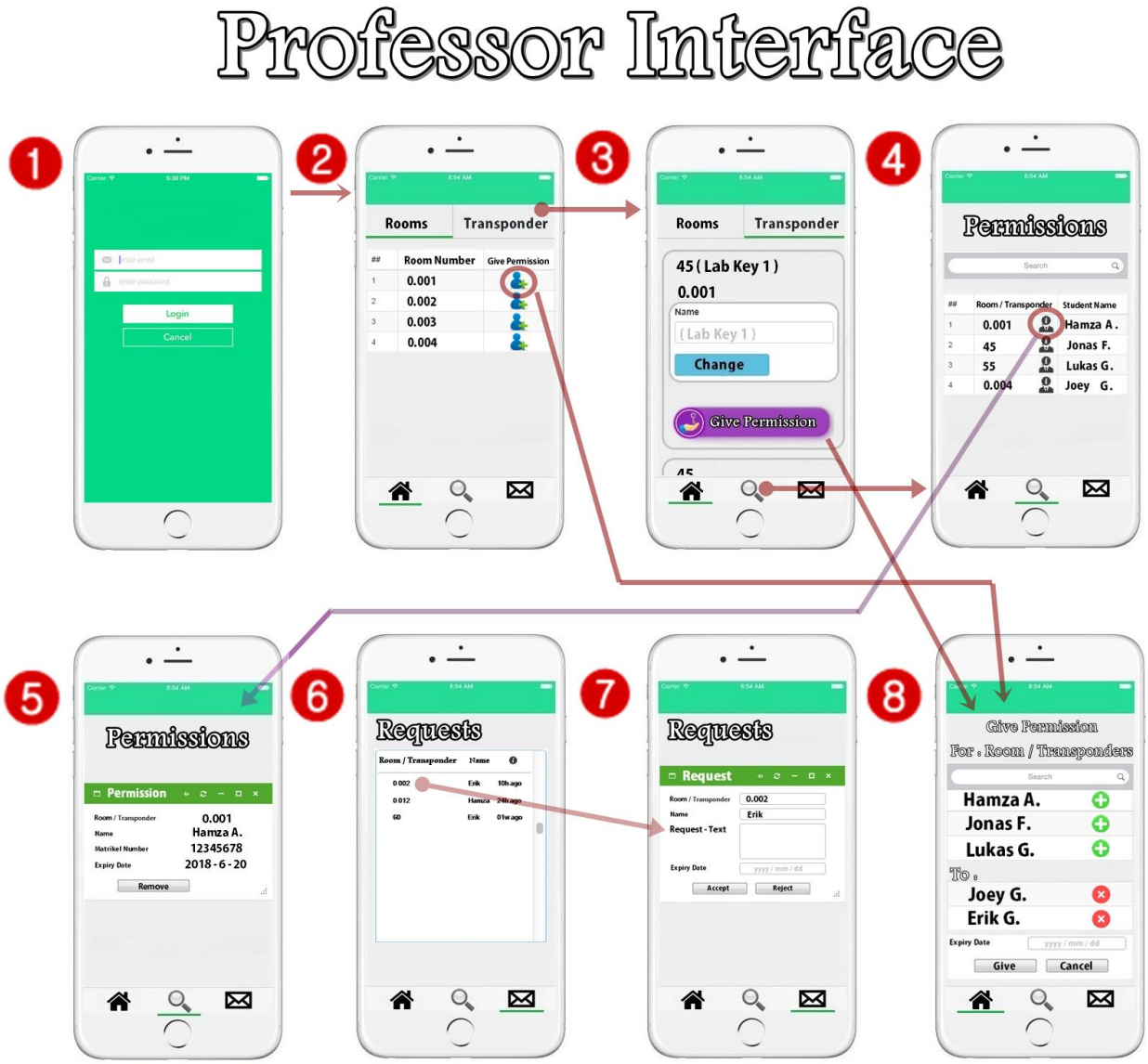
Alternatively the borrower uses his Multica for authorization he can press it to the NFC-Reader (cmp. Hierarchical Task Analysis) and the "Borrower" searchbar automatically fills with the right name.

Then he/she uses the "Room/Transponder" searchbar providing the system with the information which room/transponder the borrower wants to access, which further reduces the entries in "Results". Then the gatekeeper can choose one transponder from this "Results"-list (provided one transponder is available).

After the gatekeeper has verified the borrower the borrower signs digitally (on a tablet device, etc.) and confirms, which will trigger a popup on the gatekeepers interface telling the gatekeeper to hold the chosen transponder to the NFC-Reader. Done that, another popup opens with either a confirmation or an error (the gatekeeper has a wrong transponder).

Furthermore the gatekeeper can see all lent transponders (3rd frame from the top), so he knows, before closing the campus, if anyone forgot to return a transponder.

3. person in charge of room interface



1. The Login screen:

We need a Login screen, because the users how give permeation on the system need to be verified from the university as a Doctor or professor.

2. Give permission for rooms

This is the first view, the professors sees after the login. Here the professors can see all their rooms that they can give permeation for their students . They can add a permission when they when they press the “ symbol ”, the next view is the view where they can give a permission .

3. Permissions for transponders

There are special transponders, that can open multiple rooms, so a view where all permissions for single transponders can be viewed was necessary. The view is like the view for the rooms, just like for the transponders and there is also the information which rooms the transponder can open. where the professor can scroll down to see all transponders for rooms and they can change the name of the transponders and give a permission by pressing on the button give permission, the next view is the view where they can give a permission.

4. Search for permissions

Here the professor can search for permissions that he already gives it for students to borrow a transponder and by pressing on the "  symbol ", the next view is the view where he can remove the permission.

5. Remove a permission

Here the professor can remove the permission by pressing on remove button and he can also see the details for the permission before removing it.

6. List of the request

Here the professors can see all the request by pressing on the Message button. They can press on the request to move to the next view, that will be show the request details. the professor (the person in charge of rooms) can directly see the context of the request in the next view which helps him to make the decision of accepting or refusing the request and he can also set the expire date as will.

7. Request Decision

Making the decision of accepting or refusing the request and he can also set the expire date.

8. List of permission

The professor can search by student name and by pressing the on the "  symbol ", he will give the student a permission. On the down of the screen he has the list of the students that they have permission already and by pressing the on the "  symbol ", he will remove the student a permission.

Paperprototyping

[Artefakt/e]

For the paper prototype, we started out with the screens in the wireframe. We printed the screens multiple times for the transitions between the interactions. Then we defined user tasks, that our testers should try to accomplish. Next, we used pencils, scissor and glue to edit the paper screens to adapt them to the user tasks.

We had three test users that tested our paper prototype. All the testers tested all 3 interfaces on paper and we recorded the test for every tester and paper prototype.

The videos are available here:
https://www.youtube.com/playlist?list=PLDbCFFu3OrP8x_Un_RznTqVYnnEM2NqNq

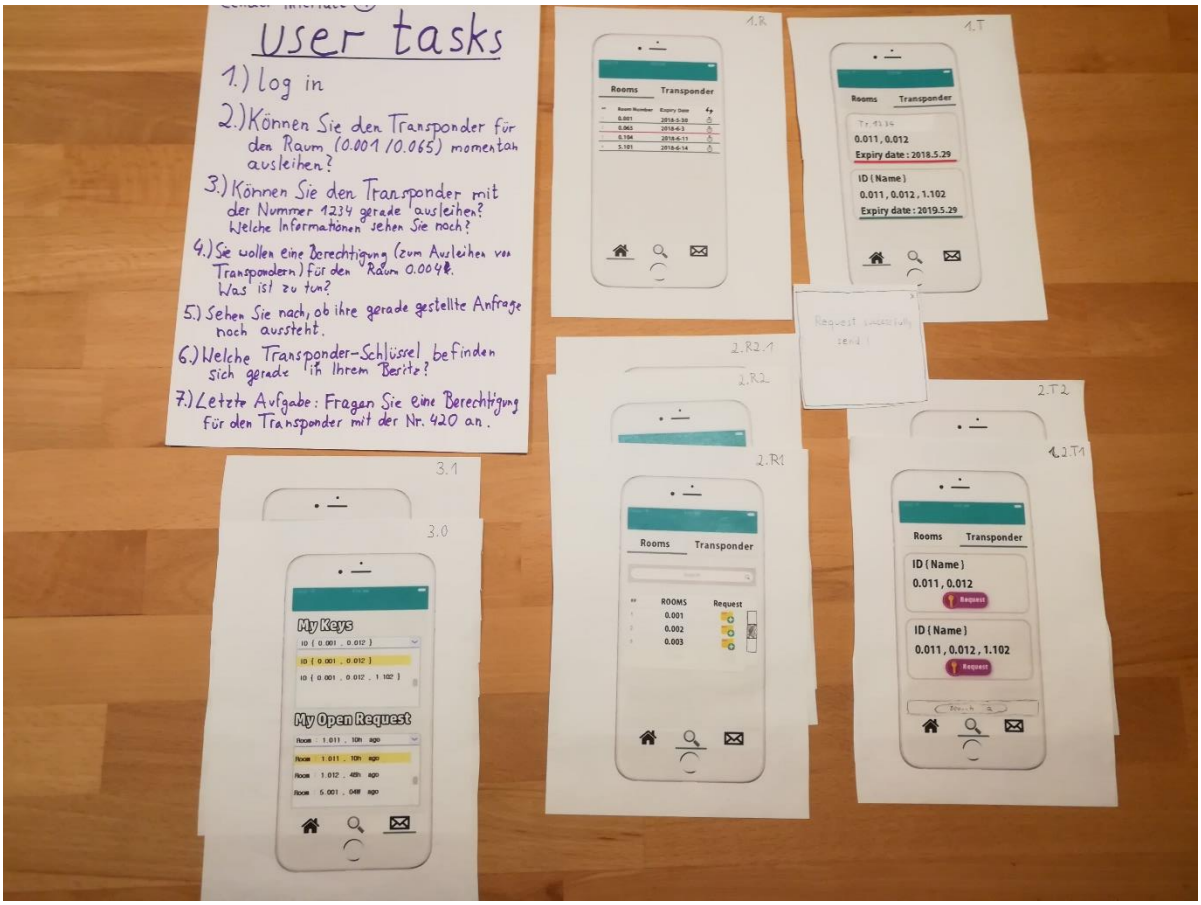
The test users

1. The test user is a 20 years old student and an experienced smartphone user.

2. The test user is a family father in his fifties. This was a special test, because the test user was not physically there and we carried out the test over facetime (video streaming). In this regard the test was carried out under harsh conditions and we need to take this into account.

3. The test user is a man at the age of 29. The user is no smartphone user and we need to take this into consideration.

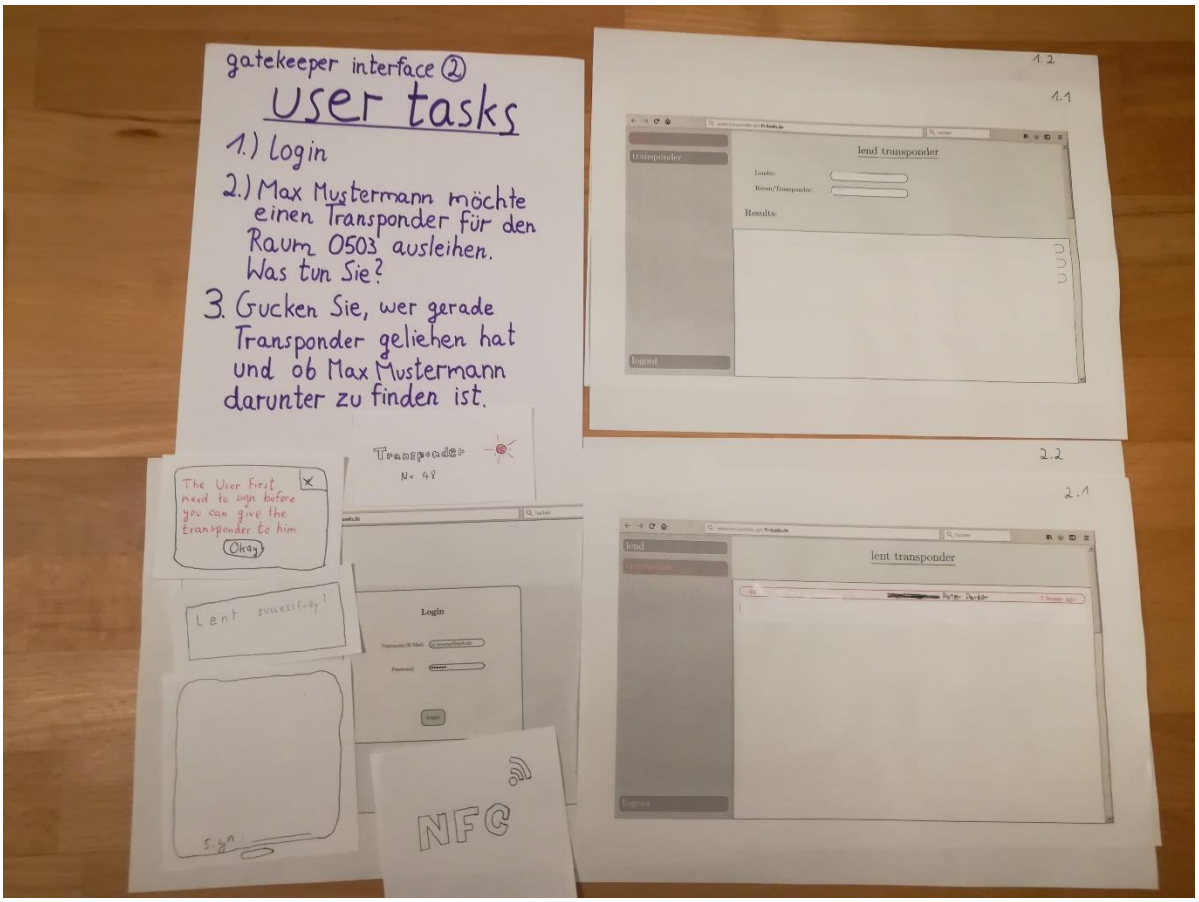
1. borrower interface



User tasks:

1. Log in.
2. Can you borrow the transponder for the room 0.001 (yes) and for the room 0.065 (no) now?
3. Can you borrow the transponder with the number 1234 now? Which information can you see?
4. You want to request a permission (for borrowing transponders) for the room 0.004? How can you do this?
5. Look if your request you sent is still pending.
6. Which transponder do you own right now?
7. Ask for a permission for the transponder with the number 420.

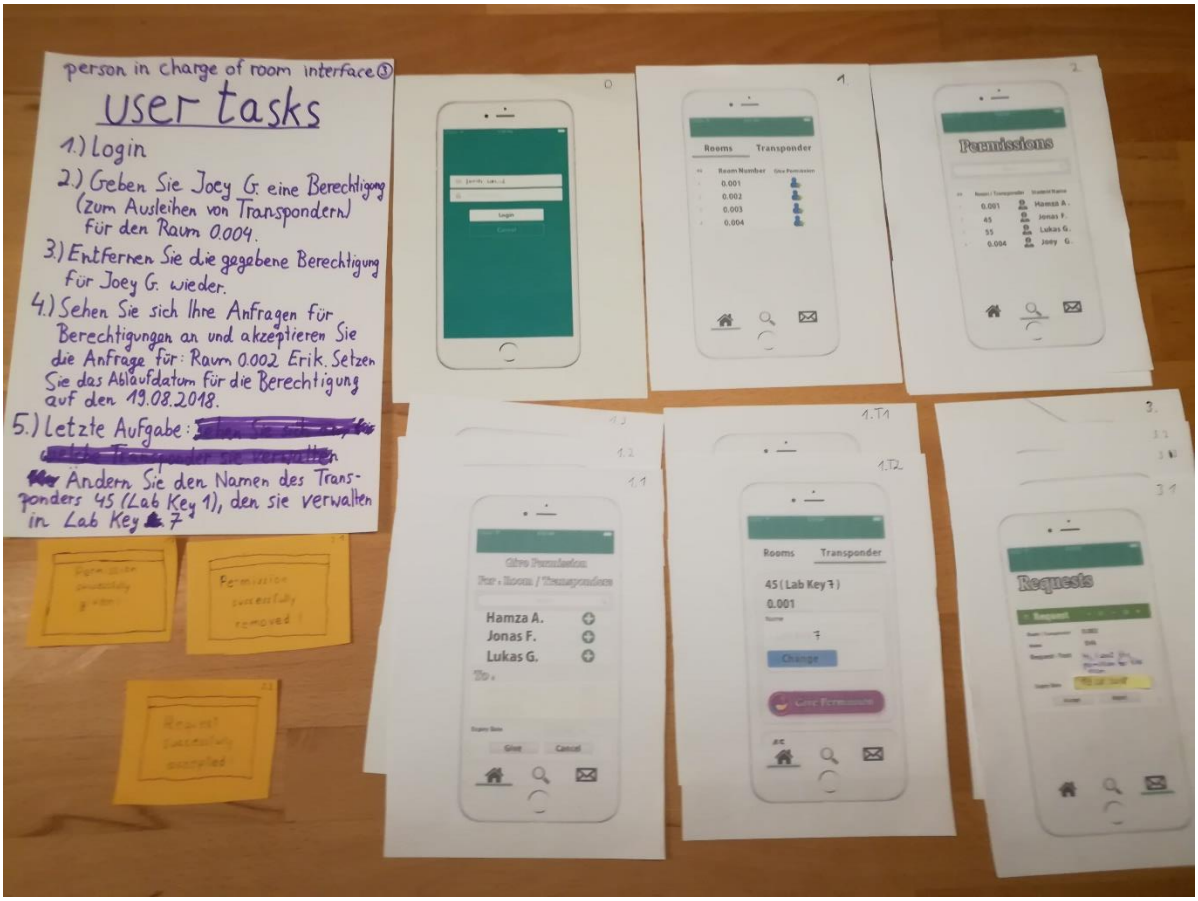
2. gatekeeper interface



User tasks:

1. Log in.
2. Max Mustermann would like to borrow a transponder for the room 0503. What do you need to do?
3. Look which transponders are lent at the moment and if you can also see Max Mustermann in the list.

3. person in charge of room interface



User tasks:

1. Log in.
2. Give Joey G. a permission (for borrowing a transponder) for the room 0.004 and set the expiry date on the 19.08.2018.
3. Remove the just given permission.
4. Look at you requests for permissions and accept the request for “room 0.002 Erik”. As the expire date, put there “19.08.2018”.
5. Change the name of the transponder “45 (Lab Key 1)” to “Lab Key 7”.

[Erläuterung]

Ergebnisse des Paperprototypings

[Artefakt/e]

Here we will describe the problems the test users had during testing and which insights we gained during testing for each single interface. We will also propose possible solutions for the problem.

1. borrower interface

All test users had problems with the bottom menu to navigate through the app. The menu icons are not clear enough. To solve this problem we could use text descriptions for the menu icons and fit the icons better to the view. We should also consider to change the type of menu into a sidebar or something similar.

We observed in the view where the borrowers can see their transponders, which they own at the moment, that the test users could not associate the header “My Keys” with the actual transponders they own. They just thought that this were the keys they could borrow and not had borrowed. To solve this the best option would be to change the header “My Keys” to “My borrowed transponders” or something like that.

2. gatekeeper interface

Test user 2 gave us feedback about the “Result”-label that is above the list of possible transponders, that a borrower can borrow. The test user said it is unclear what “Result” means and that we should change it to something like “Available transponders for <Person> and room <Room>”, where in the placeholders the data should be automatically inserted.

The user also noted that it would be better to write “TR-NR: <Number>” instead of just the transponder number.

This feedback was very helpful for us and we accept the proposals of the test user.

We also observed, that the users were very confused when the list of possible transponders appeared and didn’t know the next step. The next step would be, that the borrower signs and the gatekeeper selects a transponder, holds it over the NFC-Pad and then give the transponder to the borrower. But we don’t think this is a big problem for the system later, because the gatekeeper is an experienced user, who use the system daily and knows exactly what to do.
The test users were only introduced very briefly in the context of the gatekeeper, so they obviously didn’t know what to do. Therefore we decide to leave the interface int this regard roughly as it is and not add any helping instructions when lending the transponder, because for the gatekeeper these would be just annoying.

3. person in charge of room interface

There is also the problem with the bottom menu. (see borrowers)

One observation was, that the test users didn’t know how to remove the permissions they gave one step before. They thought that they could remove the permission in the same view, where they also gave it. But they would need to go to another view. We think we should meet

the expectation of the users and add the option to remove the permission to the same view. One possible solution could be to change the design in that way, that the users can click on an item of their room/transponder list. Then a view shows up with all the permission for the room/transponder. When the user then clicks on a permission they see the see the details and can remove it just like in the view, where they can see all their permissions. In this overview there is then a button to add permission for rooms/transponders, that links to the view for adding transponders, that we have already designed.

[Erläuterung]

Quellenverzeichnis

Literatur: [Autor; Tite;, Verlag/Veröffentlichung]

Alan Cooper, Robert Reimann, David Cronin; About Face. Interface und Interaction Design, 1. Aufl. – Heidelberg [u.a.] : mitp, 2010

Web: [Autor/HomePage; URL: ; Stand: Datum]

Wikipedia: Persona (user experience);

https://en.wikipedia.org/wiki/Persona_(user_experience); cited 23.05.2018