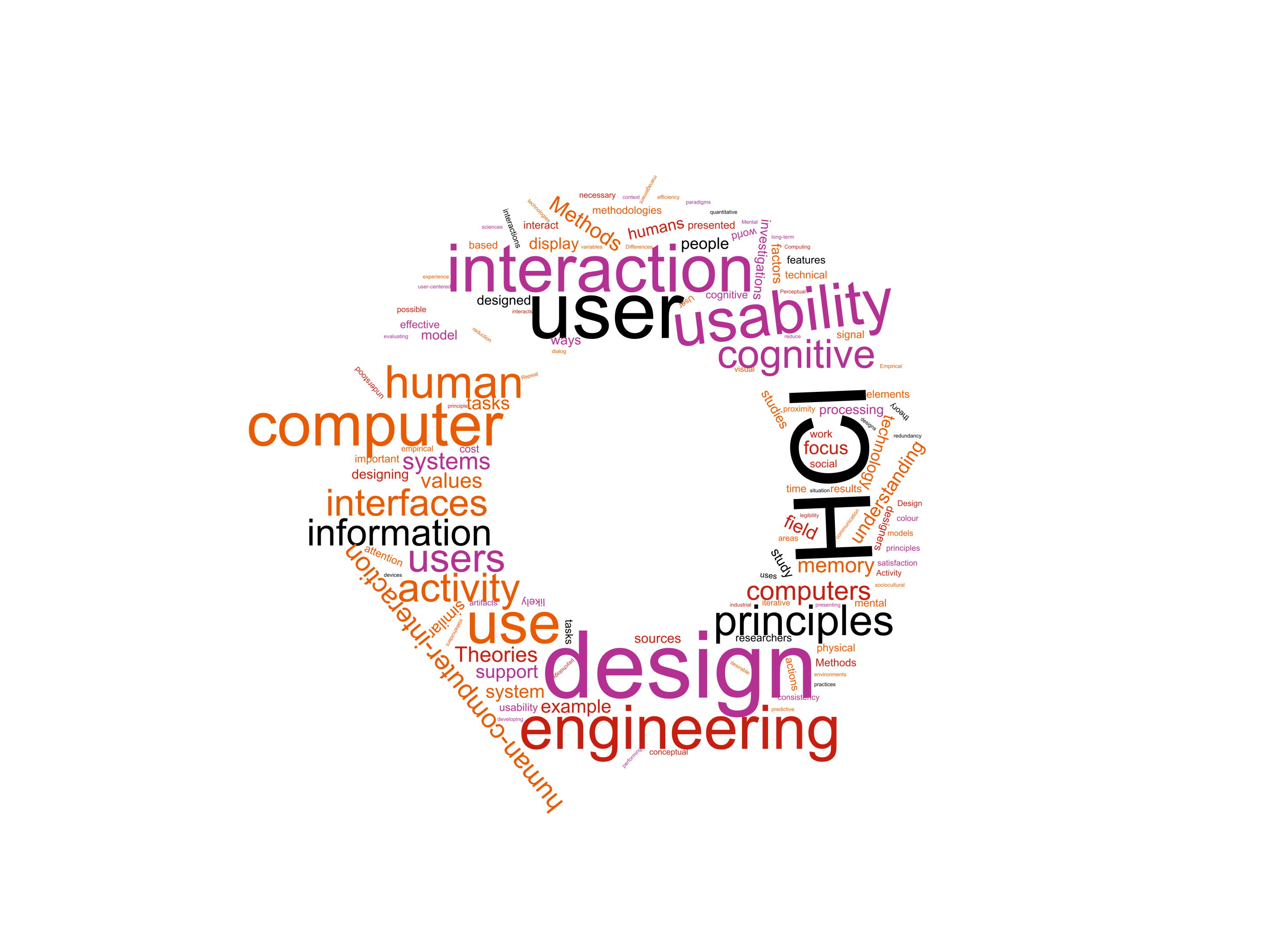
Mensch Computer Interaktion – Designspezifikation Meilenstein 1

Team

*Lukas Gobelet 11118011  
Joey Gobelet 11118012  
Erik Gödel ?  
Jonas Faßbender ?  
Hamza ? 11131184*



[Einleitung 2](#_Toc511326215)

[Szenario 3](#_Toc511326216)

[Meilenstein 1 4](#_Toc511326217)

[Nutzermodellierung 4](#_Toc511326218)

[Stakeholder Analyse 4](#_Toc511326219)

[User Profiles 5](#_Toc511326220)

[Personas 6](#_Toc511326221)

[User Storys 8](#_Toc511326222)

[Top Level User Tasks 9](#_Toc511326223)

[Hierarchical Task Analysis 9](#_Toc511326224)

[Rapid Prototyping 13](#_Toc511326225)

[Story Board 14](#_Toc511326226)

[Quellenverzeichnis 16](#_Toc511326227)

# 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 Pflegeintensiv 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 1

## Nutzermodellierung

### Stakeholder Analyse

#### Gatekeeper:

He lends the transponders to the lenders.

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

He is also holding a register about who lends 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.

**Person in charge of rooms:**

He grants the lenders the timely limited authorizations for the transponders.

He has to give these to the gatekeeper.

he is interested in automating the system of delivering the lists to the gatekeeper.

#### Lenders

The lender can be any person. He can lend the transponder for which he has the authorization granted.

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

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

The lender 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]*

### User Profiles

#### Gate staff

*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 gate staff has a hard time managing all these lists and the process occupies accordingly a lot of time.

The gate staff is 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 are in charge of rooms and are able to give permissions for transponders that open their rooms to other people.

*Characteristics:*

They are very busy and they 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).

#### Lenders

That can be basically everybody. But only the ones with permissions can lend transponders. The persons in charge of the rooms trust them but they are responsible for any loss of the transponder.

*[Erläuterung]*

### Personas

#### Andreas Fischer (gatekeeper)

Andreas Fischer works as a doorman 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 lend 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 lend the corresponding transponders that can open the rooms. So, he needs to go down to the doorman, where he needs to authorize and give the doorman 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 directly from his desk .

Prof. Dr. Siegelbud wants:

* to give permissions for transponders (rooms) to people immediately 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 way

#### 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 in the 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 doorman in the lobby. There she waits for the doorman 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 it there. The lending of transponders also takes a while. She feels sorry for the doorman, 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 lending a transponder is more quicker
* that she can look beforehand whether a transponder is lent or not

*[Erläuterung]*

Here I chose to make from each different user role for the system exactly one representative persona. So we have the gatekeeper, Andreas Fischer, the professor Siegelbud and the student Sandra Meyer. I tried to tie the stories of them 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.

## User Storys

#### Gatekeeper

1. As a gatekeeper, I can check the availability of a specific transponders, so that I can give the person, who wants to lend 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 lend 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 check status (lent or not, lender, date,...) of all transponders, so that I know which are missing and can perform the necessary actions if transponders are missing for a too long time.
4. As a gatekeeper, I can lend transponders to people without too much administrative effort, so I don't have to manage huge lists and the persons who want the transponders are happy.

#### Persons in charge of rooms

1. As a person in charge of a room, I can give/remove permissions to other people for lending transponders to open the rooms I am in charge of, 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.

#### Lenders

1. As a lender, I can lend a transponder to open a room without much effort, so I can open rooms by using it.
2. As a lender, I can look beforehand whether a transponder is already lent or not, so I save time and I only lend a transponder if it is available.

*[Erläuterung]*

## Top Level User Tasks

## Description

The Top Level User Tasks should give us a comprehensive insight into the main tasks our users will try to fullfill 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.

## Top level user tasks

* lend transponder
* give/remove permission
* verify lender's permission
* check for available transponder
* check if I have permission to room/transponder
* ask for permission

## Conclusion

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

*[Erläuterung]*

## Hierarchical Task Analysis

## Description

Our Top Level User Tasks decomposed in smaller, hierarchical tasks/subtasks and actions a user will perform using our application. Furthermore it allows us to specify plans, which helps us to model state (e.g. transponder not available, lender has no permission), conditions (e.g. only if lender has permission he can lend a transponder, else the gatekeeper reject the request) and descisions (e.g. lender 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 on in the development proccess.

## 0. Lend transponder

**graph in ../../static/hta-graphs/lend\_transponder**

**Plan 0.3.1:** 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:** 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 lender does. He/she wants to use this system to gain access to a room he/she has permission to use.

## 1. Give/remove permission

**graph in ../../static/hta-graphs/give\_remove\_permission**

**Plan 1.4.1:** do 1.1-1.3. If you want to give permission do 1.4.1 and continue with 1.5.

**Plan 1.4.2:** do 1.1-1.3. If you want to remove permission do 1.4.2 and continue with 1.5.

### Description

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

## 2. Verify lender's permission

**graph in ../../static/hta-graphs/verify\_lenders\_permission**

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

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

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

**Plan 2.6.2:** do 2.1-2.5. If the lender 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 lender's task "0. lend transponder".

Since one of our main goals is to keep the university's property safe, this task of verification is totally 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 lending 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 lender 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 lent 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 lender used his ID for identification, the gatekeeper presses a button on his interface which tells the app who is the lender. 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 avaiable transponder

**graph in ../../static/hta-graphs/cfat**

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

### Description

A lender can check if his desired transponder or a transponder to his desired room is available using our mobile application. The lender can look up his permissions and filter them. Since our system knows all the time if a transponder is available or not, it can give the lender this information.

## 4. Check if I have permission to room/transponder

**graph in ../../static/hta-graphs/cptrpt**

### Description

In case a lender wants to check if he has permission to a room or transponder. Furthermore he can see when his permission expires.

## 5. Ask for permission

**graph in ../../static/hta-graphs/ask\_for\_permission**

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

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

### Description

A lender can use our application for making permission requests. That means he/she can look up a room or transponder and ask for a permission, which the person in charge of this room or transponder can either grant or refuse. Either way the lender gets informed by our application about whether the request is granted or refused.

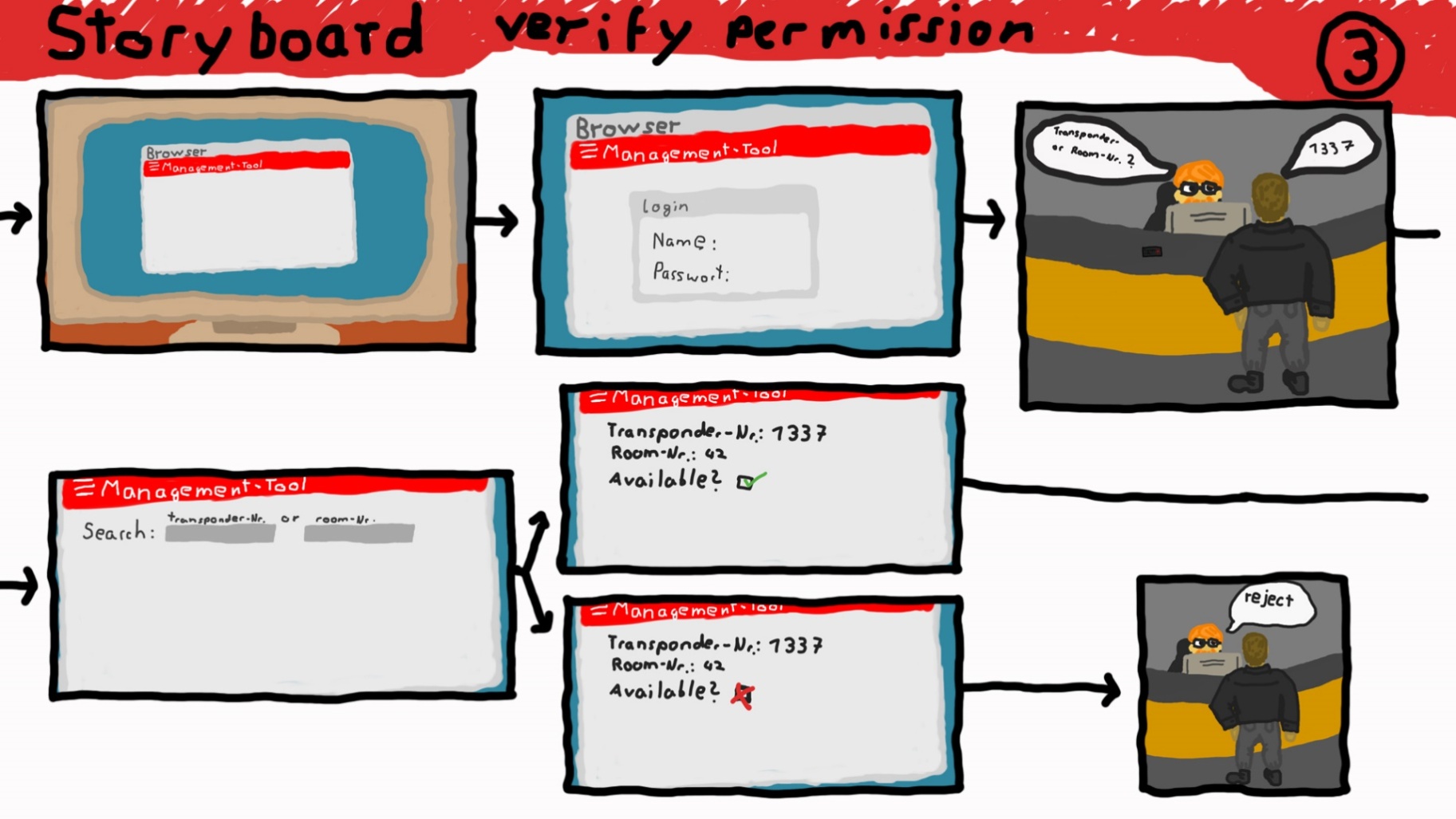
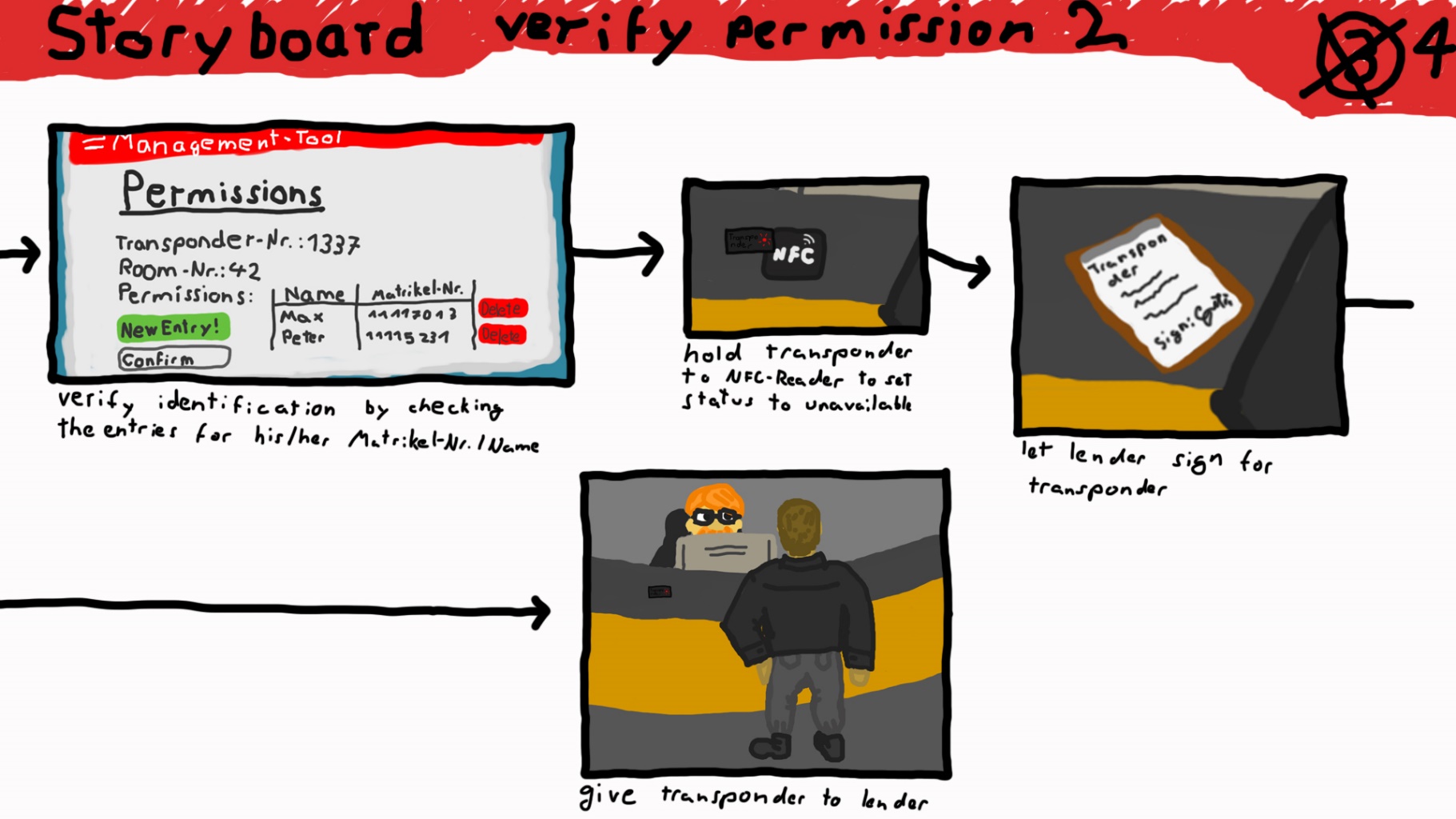
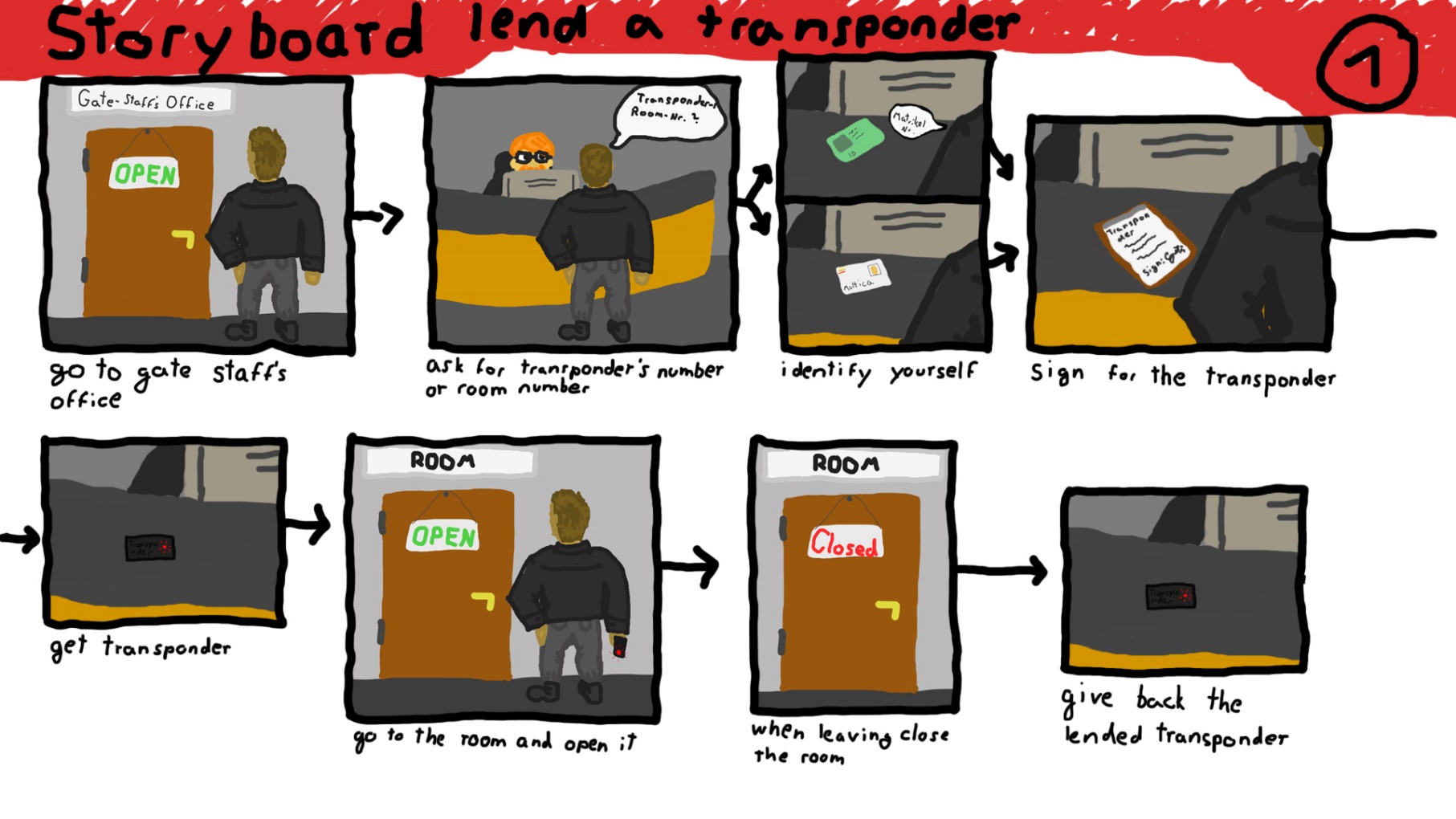
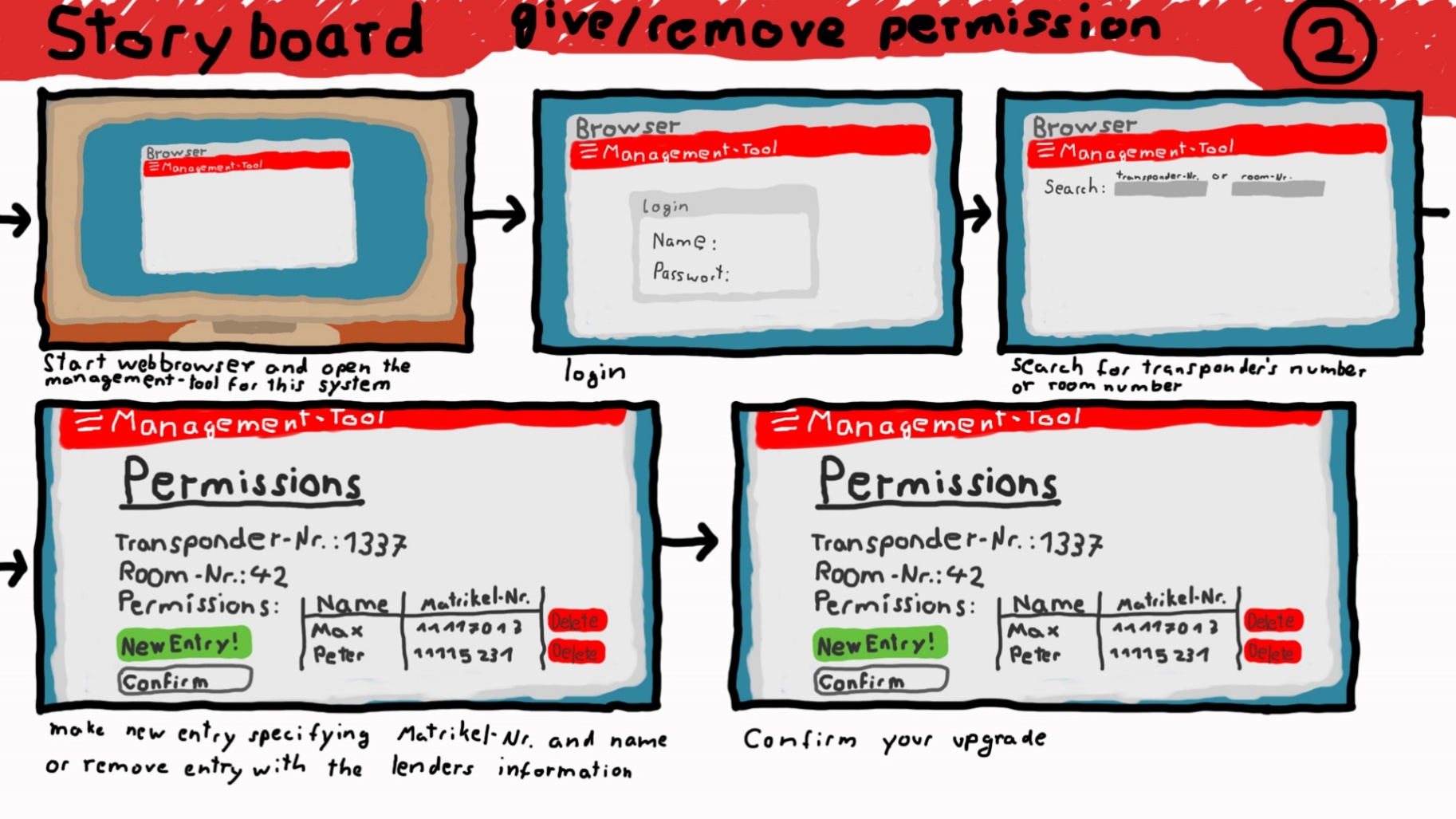
## Conclusion

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.).

*[Erläuterung]*

## Rapid Prototyping

### Story Board



*[Erläuterung]*

# Quellenverzeichnis

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

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