Requirements Analysis Document

Cinema Ticket Reservation System

Version 2

Authors	Miriam Anschütz, Ivana Zucic, Georg Groh
Date of creation	01.07.2020
File name	Requirements_Analysis_Document_Cinema_Case.docx
Number of pages	13

Contents

1 Introduction

This chapter gives an introduction into the project and describes what purpose it has.

This document describes the cinema ticket reservation system and requirements of the project. The goal is to provide a general overview of the project.

1.1 Purpose of the System

Briefly describe who might use the system and for which purpose. What is the target group of the application?

The users will use the system for cinema tickets reservation. The user should be able to see what is playing in each of the cinemas that are available in the application and book the desired amount of tickets for a certain playtime. They should see their list of reservations and be able to cancel a reservation.

The target group of the product represents any person who wants to browse available cinemas, movies and playtimes or who wants to reserve tickets in one of the cinemas that are participating in the application.

1.2 Scope of the System

Explain the problem your system is solving. How wide is the area you can apply your project to?

The system can provide a reasonably large number of cinemas and a large number of movies that the users can look through. Since there is no option to search for a cinema, the user must look it up manually by scrolling in a list. For that reason, the application would be suitable for hosting the cinema program of a city or a region, such that the user experience is still good because the list of cinemas would be easy to go through by hand.

1.3 Objectives and Success Criteria of the Project

Explain the project objectives and the criteria you'd use to judge if it was successfully executed.

The goal of the contractors is to deliver an application that serves the purpose that was laid out above. All requirements described below should be met and the product owner should accept the product.

The project will be accepted if:

The user can make an online reservation for a specific movie at a specific date and time for a selected cinema. This way, the user makes sure they will have a place to watch the desired movie and has no need to be afraid that they will walk to the cinema and be unable to buy any tickets.

1. The user can manage the previously made reservations and either modify the reservation, i.e. select another amount of reserved tickets, or completely delete specific reservations.
2. The user can only perform the previously described action if they are logged in. This means the user has to create an account in order to utilize the services. At first this might seem as a drawback, however this is the only way to make possible that the user can safely modify or delete a reservation which was previously made.
3. The user can also view and edit their current profile information. This way it is easy for the customer to check if all information is still correct and possibly update it.

1.4 Definitions, Acronyms and Abbreviations

This part should prevent any wrong interpretations of the terms used in this document. For example, when we say cinema, we mean one of the cinema branches and not the whole cinema company or one of the movie projecting halls. Such explanations are helpful in designing the system correctly. In this document, a few examples are given and not all the possible terms are defined, since this is a toy application used to give you an example. ***When you'll be writing your document, it's enough to define only a few (3 - 4) terms as well, just to prove that you understand what kind of content should go here.***

Here are the definitions of the terms, acronyms and abbreviations that are necessary to understand to read this document.

Term / Acronym / Abbreviation	Definition
Cinema	Cinema is considered to be a branch of some cinema company, situated at a certain address. Cinemas are projecting different movie program depending on the date.
Movie	Movie is a video that is played at a cinema in one of the movie halls. The same movie can be played multiple times. A movie has a name, length and a genre.
Reservation	Reservation is done by one user for one movie that is playing at one cinema at a certain time. A reservation has to keep track of the number of the tickets that the user reserved, since there can be multiple for one reservation.

1.5 References

Here one should provide references to the development context (e.g. references to existing systems with a similar purpose). The project described in this document is an independent project and has no dependencies to any other project. It is part of the course IN0827 and as such is a standalone project. This means, no further projects will be built upon the here described project. The same is correct for your project.

There is no previous system, it's a greenfield engineering project.

1.6 Overview

Give a ***very short*** *overview for the reader, introducing the content which follows in the document.*

The following chapters will describe the current way the tasks supported by the cinema ticket reservation system are done and define the requirements that the system should fulfil.

2 Current System

This chapter is used to describe the current state of affairs. If your system replaces an existing system, you should describe the functionalities and problems of the old system. Otherwise, you should describe how the tasks supported by the new system are accomplished now (e.g. they are done manually on paper).

This is a toy system and we imagine that only the physical way of selling tickets exists and we're the first ones implementing the software system. You can assume the same for your toy systems. ***Here we imagine a world where there are no other cinema ticket reservation systems. You should also assume that there is no pre-existing system for your application.***

The system is an independent project and doesn't serve as a replacement of any other existing system.

Here is a short overview of the current approach that cinemas use to and sell tickets on their cash registers.

Each customer queues for the tickets at the cinema cash register. The customers must come during the cash register opening hours and ask for a ticket for a certain movie. The payment is done by cash or by card.

Pros	Cons
There is no need to create a user account.	The queue for the tickets might become very long, and the waiting time also.
There is no need to have a bank card.	The customer has to come during the cash register working hours.
	The customer doesn't know if all the tickets are sold out, so they could be waiting in the queue just to get to know that there are no tickets left.

From the overview above, it's visible that the users who would have a way to conduct cinema tickets reservations online could benefit from it.

The cinema tickets reservation system will make online ticket reservations possible and also enable a digital way to look for a ticket for a desired movie.

3 Proposed System

This chapter documents the requirements elicitation and the analysis model of the new system.

3.1 Overview

Present a functional overview of the system.

The general procedure of a user is that the user browses the available cinemas and the corresponding movies they offer. After deciding for a specific movie in one cinema, the user needs to select a suited date and time. Following, the user can make a reservation if the user is logged in to his/her account. If the user is logged in, he/she can always view the made reservation and can modify or deleted them as desired.

3.2 Non-functional Requirements

In this section, describe user-level requirements that are not directly related to functionality. Dedicate one short section to each of them.

In this section each of the non-functional requirements will be listed i.e. all the additional requirements that are not contained in the functional requirement but are still relevant and further constraints will be described.

3.2.1 Usability

The complete application should be intuitive to use and the user interface should be easy to understand and to operate for all people of any class of age. This specifically means that no complex and nested graphical components should be used.

3.2.2 Reliability

The list of available movies for a cinema should be displayed within 5 seconds after the user clicked on the cinema. The same time limit is applied to other actions, such as: performing a reservation, changing a reservation or changing the profile data.

3.2.3 Performance

The system should provide the desired output almost instantly. This means that any action that the user performed should have an immediate effect. The user should, for example, get a list of movies for the desired cinema almost immediately after choosing a cinema.

The application should not crash when the user inputs some wrong data. All the data the user inputs should be quickly validated and the user should be notified if they entered some invalid data.

3.2.4 Supportability

The implemented software should be well structured and should have a good code style throughout the project. Existing features are easy to modify and the system can easily be extended with additional requests and features.

3.2.5 Implementation

No implementation details *should be provided in the Requirements Analysis Document, but it makes sense to give some* ***very rough*** *information telling the developers about the* ***target OS and platform*** *for the system that needs to be implemented. This is needed simply because different developers specialize for working on for example Android mobile applications and*

they would have to learn to develop some Windows stand-alone application. To hire the right people for the project, we need to know what their rough area of work needs to be.

The implemented system needs to be a stand-alone Java application able to run on normal PC or laptop running Windows.

3.2.6 Interface

The users are all the people interested in cinema, who want to either see what's playing or reserve a ticket. This also includes elderly people and thus the interface of the application should be relatively simple to operate and should not contain layouts that are too complex and hard to understand.

3.2.7 Packaging

The system should be delivered to the user as a Java package.

3.2.8 Legal

No costumer data should be available without logging in. No customer should be able to view or alter the data of other customers. Here the data includes both personal information and the list of reservations.

3.3 System Models

This section should contain the complete functional specification.

3.3.1 Scenarios

Write the scenarios for your application in the form of short paragraphs. Give concrete examples (a specific user does a specific action).

3.3.1.1 Scenario 1: View Items

Max wants to do something for the weekend so he is exploring the cinema tickets reservation system to see what's playing. He sees a list of all the cinemas and picks the Museum Lichtspiele cinema, because it's the closest to him, according to the address that he sees in the cinema description. He looks at the list of movies for that cinema and picks Titanic. He looks at the available dates and times, but there is no screening of Titanic for the weekend, so he goes back to the list of movies playing at Museum Lichtspiele and clicks on Avatar.

3.3.1.2 Scenario 2: Login and Registration

Max wants to reserve a ticket, but he doesn't have that option since he didn't log in. He also didn't use the system before, so he doesn't have an account and needs to register. He clicks the "Sign Up" button and gets a form to fill in. He types his first and last name, e-mail and a password that he just invented (1234). After that he clicks the "Sign up" button in the form to confirm his entry, he gets a message saying that he should create a longer password. He thinks of a better password and clicks "Sign Up" again. This time, a success message is shown and he is already logged in.

3.3.1.3 Scenario 3: Simple Login

Max likes the cinema tickets reservation system very much so he uses it every week. He already has an account that he always uses. Now he wants to reserve some tickets for the next Sunday so he clicks the "Log In" button in the reservation system interface and logs into the system by typing his e-mail with the corresponding password in a form. After the he clicks the "Log In" button in the form, a success message is shown and Max continues to browse the system, ready to reserve some tickets when he finds an interesting movie.

3.3.1.4 Scenario 4: Reserve Item

Max decided that he wants to see Avatar this weekend. He found it playing at Museum Lichtspiele at 19:00 on Sunday. He is already logged in, so he can start the reservation process by clicking the button "Reserve". Now he gets to choose the amount of tickets to reserve. Max choses three tickets for him and his friends and clicks on the "Reserve" button to confirm his choice. He sees a success message and he is directed to his profile page he also sees that his new reservation was successful.

3.3.1.5 Scenario 5: Show Profile

Max wants to check if he reserved enough tickets for his night out with friends, so he logs in the cinema ticket reservation system and looks at his profile page. There he sees his personal information (his first and last name and e-mail address). He also sees all his past and future reservations in a table. He can easily see his reservation because all of the past reservations are greyed out.

3.3.1.6 Scenario 6: Modify Reserved Item

Max notices on his profile page that he forgot to count himself for the Avatar tickets when he was reserving them, so he reserved 3 instead of 4 tickets. He selects the reservation for Avatar on the profile page and clicks the "Modify" button. He can now modify the amount of reserved seats for this reservation. He changes the number of tickets from 3 to 4 and clicks the "Change reservation" button. Then he sees the success message is led back to the profile page where the modification of the reservation is visible.

3.3.1.7 Scenario 7: Delete Reserved Item

Max got sick, so he needs to cancel his plans this weekend and also delete his cinema ticket reservation for the movie "Titanic". He logs in the cinema ticket reservation system and on his profile page he selects the reservation for the movie "Titanic". Then he has an option to delete the reservation by clicking on a "Delete" button. Max clicks and the reservation for "Titanic" is no longer shown in the reservations table on Max's profile.

3.3.1.8 Scenario 8: Log out

Max wants to be sure his younger sister doesn't use his cinema tickets reservation account while she's borrowing his laptop for school, so he logs out by clicking the "Log Out" button.

3.3.2 Use Case Model

Place your use case diagrams here. We will skip the detailed textual description of the use cases here, you can do the same in your document.

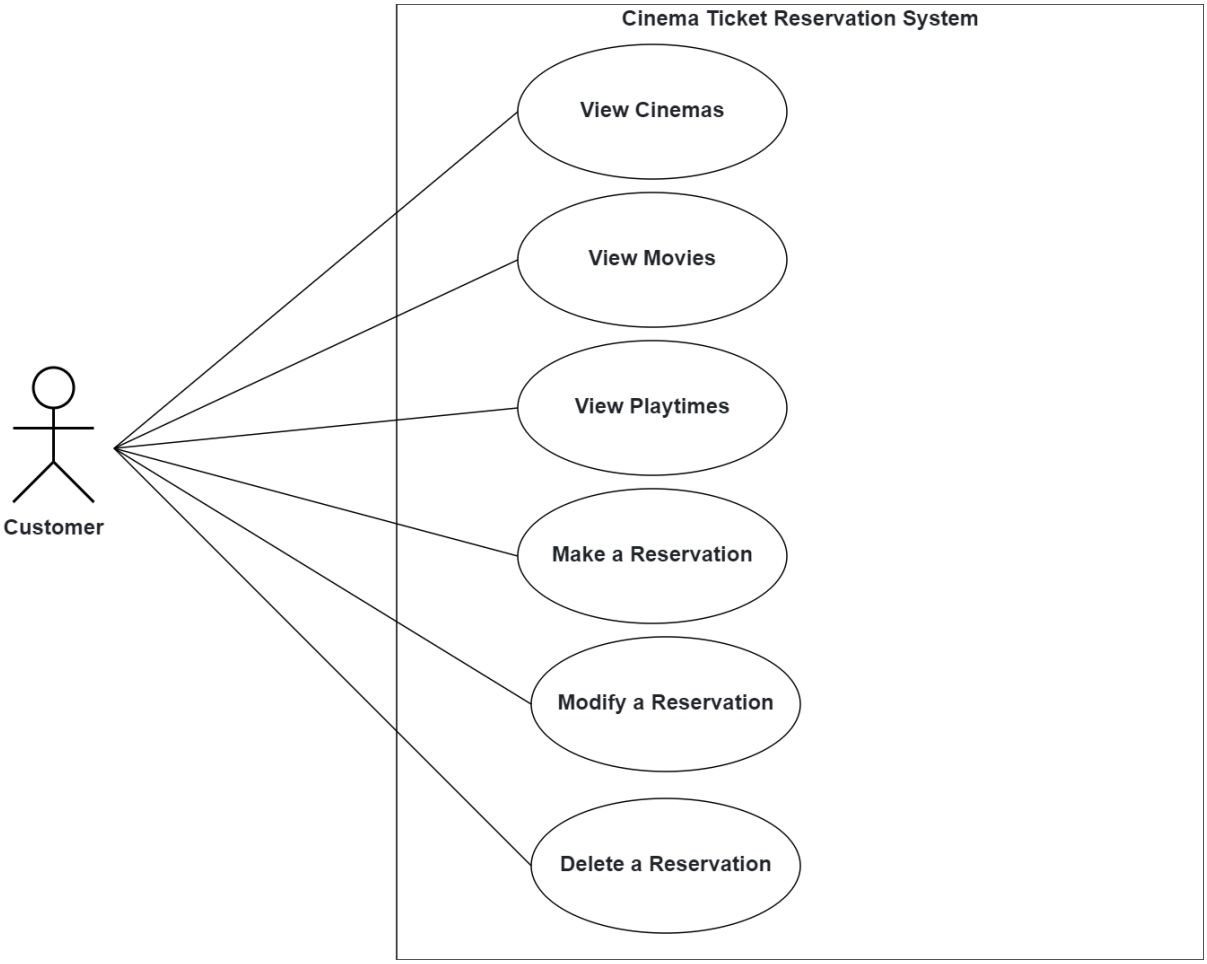

Figure 1: Use case diagram.

3.3.3 Object Model

Place your class diagram showing the application domain here. No textual description is required.

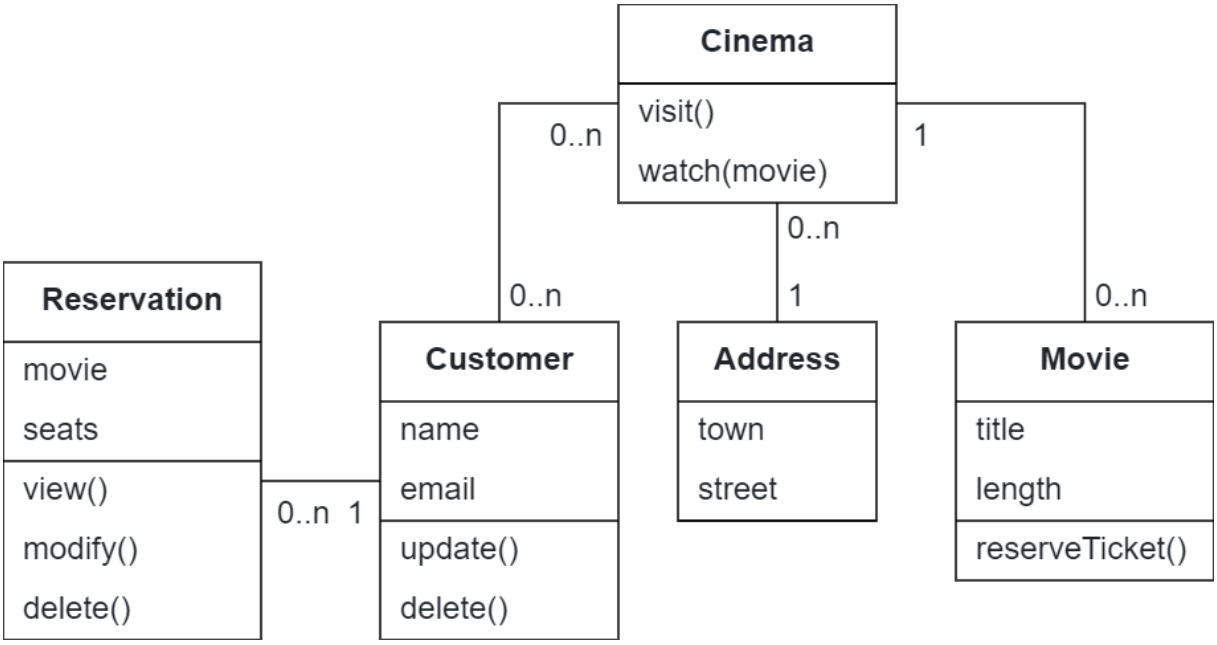

Figure 2: Class diagram from the application domain.

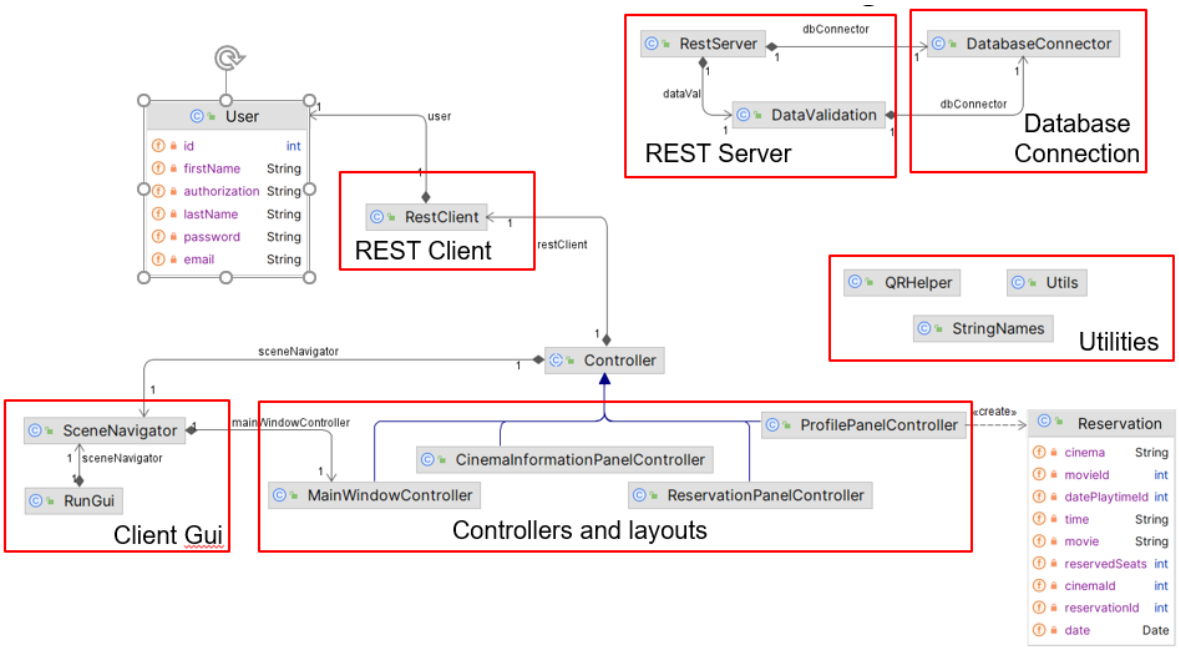

Figure 3: Class diagram from the solution domain.

3.3.4 Dynamic Model

Place your communication diagrams here. No textual description is required.

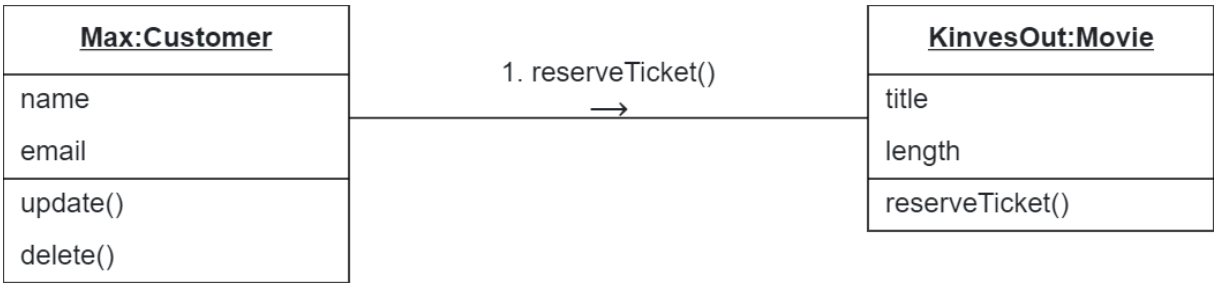

Figure 4: Communication diagram showing ticket reservation.

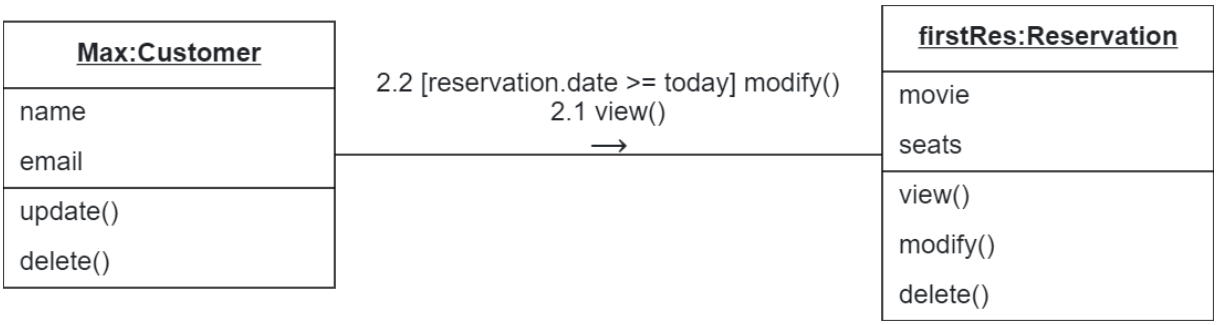

Figure 5: Communication diagram showing reservation modification.

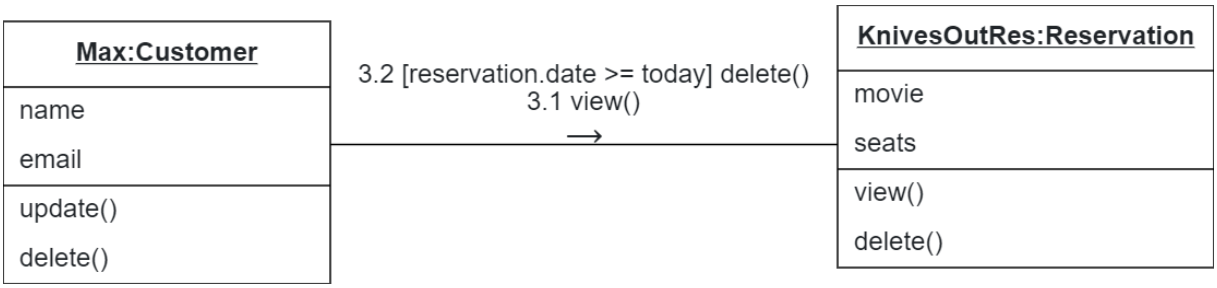

Figure 6: Communication diagram showing reservation deletion.

Figure 7: Communication diagram showing a visit to the cinema to watch the movie.

3.3.5 User Interface – Navigational Paths and Screen Mock-ups

This part of the document should illustrate the user interface of the system and navigational paths represented by the sequence of screens.

***You can leave this part out** from your document or leave it empty, since you are only working on a small toy project for the purpose of the course. **This part will not be graded, but feel free to put mock ups here if you like**, especially if you created them during your discussions with your teammates. Here some mock ups are provided such that you get a clearer picture of what kind of content should go to this section. Mockups don't need to be perfect. They should just provide an idea about what element goes where in the interface. There are many online tools for creating them, but you can also draw them in any way you like as long as they serve their purpose.*

4 Glossary

One of the obstacles between developers and users is differing terminology. To establish a clear terminology, developers identify the participating objects for each use case in a glossary.

In our case, the glossary would mostly duplicate the chapter called "Definitions, Acronyms and Abbreviations", so we leave it blank and you can leave it blank for your Requirements Analysis Document.

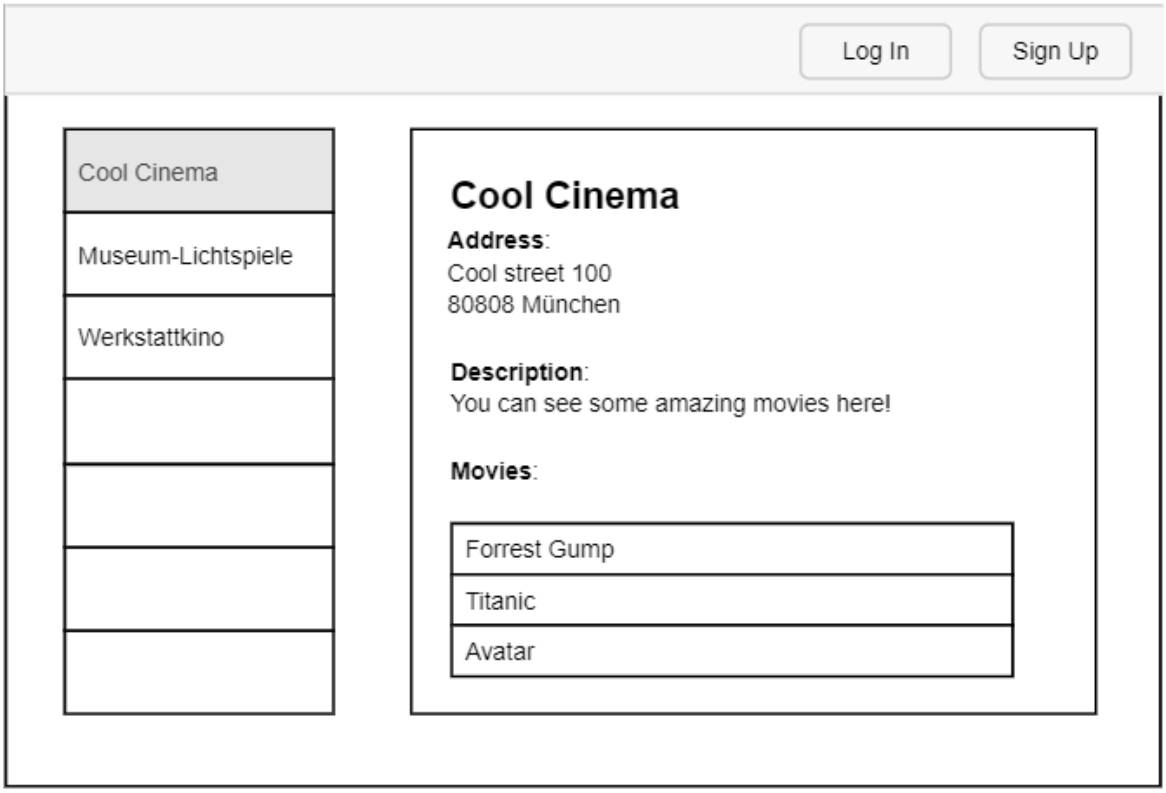

Figure 8: The interface that shows the list of cinemas: cinemas are clickable and the details about the cinema that was clicked is shown to the user.

Figure 9: The interface that shows the profile page of the user.