Use Cases:

I. Sign Up – Alexander Ramharter

II. Login - Alexander Ramharter

III. Look at and search for films/screenings - Oliver Schweiger

IV. Buy ticket – Utz Nisslmüller

V. Look at and cancel bought tickets - Utz Nisslmüller

VI. Manage films - Oliver Schweiger

VII. Manage screenings – Oliver Schweiger

VIII. Employee administration - Yasin Ergüven

IX. Hall administration – Yasin Ergüven

X. User administration - Alexander Ramharter

XI. Role Based Access Control - Alexander Ramharter

More detailed description of work distribution:

Alexander Ramharter: I was very interested in how Login and Sign Up actually works so I chose these two use cases. It took me a little to get back into PHP, but after a while I got pretty confident with using it. After seeing the implemented use cases we thought it would be nice that different users have different abilities on the webpage. And since I already used global session variables for log in I implemented that use case as well. In the end we decided to add additional operations for the user such as changing the password or delete the account.

Oliver Schweiger: I was responsible for the Use Cases regarding the interactions with our movies and screenings. I was also responsible for setting up the MySQL connector in PHP and building the basis of our pages that interact directly with the database. I implemented the CRUD functionality for films and screenings as well as some comfort functions. For MS3 I plan to add even more of those, so that interacting with the movie database and searching for screenings will be even easier.

Yasin Ergüven: For MS2, I was responsible for the realization of the use cases "Employee administration" and "Hall administration". I have added the CRUD operations for these two use cases and tried to demonstrate all information as simple as possible. The great challenge was the realization of the unary relationship for employees inside PHP and the representation of the generated data in an understandable way to provide a functionality for an efficient administration of information.

Utz Nisslmüller: In our initial team meeting session I chose the use case of buying tickets. This meant purchase (entry into the ticket table of our MySQL schema), display of purchased tickets on some sort of overview page as well as their deletion. However, I realized pretty early, that this was infeasible and unrealistic (simulation of transaction, extra table for credit/payment info, abstraction of the process via the web interface…). Hence, I decided to change this use case into reservation only – purchase would take place over the counter at the cinema itself, e.g. half an hour prior to the screening. This also made sense, since discount eligibility (student and pensioner) would have to be verified on-site anyway. Via the web interface, it is now possible to reserve tickets in varying quantities and discount types (reserveticket.php), to view your reservations (user.php). The user can also delete his ticket on user.php. I didn’t include an update operation, since the user should only be possible to make new reservations if there is room available and therefore go via the process on reserveticket.php again. A logged-in employee is able to view all tickets and insert, update and/or delete an arbitrary table entry (ticket.php & updateticket.php).