

# Final Report

**Team: g1t2**

Member 1: (Christina Brückl, 01219643)

Member 2: (Leonie Katharina Geyr, 11778679)

Member 3: (Thomas Krabichler, 11906966)

Member 4: (Kevin Rimml, 12036261)

Member 5: (Ioannis Deligiannidis, 11913541)

**Proseminar group: PS-703080-1, Team2**

**Date: 23.06.2022**

## 1. Analysis of the project process

- *Comparison of the planned milestones with the actual course of the project and the issues. Comparing the set of milestones with the actual project course, we were almost 100% on the planned time. We started very motivated by setting down issues for our programming part and which milestones must be done, but in the late phase of the project we neglected the writing issues down a lot.*
- *Analysis of the hours worked (share of the respective design activities in the total number of hours)*

PRIMARY ACTIVITY	DURATION (HOURS, MINUTES)	PERCENTAGE %
Implementation	~205 Hours	~47%
Proseminar Units	~68 Hours	~15%
Workshops	~30 Hours	~7%
Software Concept	~24 Hours	~6%
System Testing (Own and Others)	~35 Hours	~8%
Final Report	~10 Hours	~2%
Configuration and Deployment	~19 Hours, 30 Minutes	~5%
Coordination and Project Management	~40 Hours	~9%
Final Presentation	~5 Hours	1%
<b>ACTIVITY SUMMARY DURATION</b>	<b>~433 Hours, 30 Minutes</b>	<b>100%</b>

## 2. Analysis of the implemented system

- *Was the initial concept (including use cases, functional class diagram) stable throughout development? If not, which parts have changed?*

*To realize the models in the backend for the database an ER-Diagram was made at the very beginning. That helped a lot to initialize our models, so that they were very stable throughout development.*

*The system is based on our use case diagram, where in the use case diagram only the basic functionality of each actor is shown. More functionalities, like validations, email and password changing were added in the future, they were not part of our use case diagram.*

*Therefore, some small things were missing in our use case diagram, e.g. login/logout, the responsible person to add/edit and remove rooms, etc... which we implemented later on.*

- *Was the component structure planned in the SW architecture stable during implementation? If not, which parts have changed?*

*At the beginning we had to simplify a little bit our component diagram, that was a good tip from our PS Leader.*

*The initial component structure planned was based on a H2 database, that we had to change it to MySQL or another persistent Database. We've chosen MySQL.*

*We also had to add the facility management in our component plan.*

- *Could the entire planned functionality of the system be implemented?*

*So far, we've implemented and tested the planned functionality of the system. We also tested and implemented strict validation mechanisms for some of the cases, as well as mechanisms for some edge cases.*

- *What experiences have you gained with the quality management of your system?*

*We used the Sonar Qube features from the very beginning to keep our system clean of bugs and to avoid a major issues, which may occur at the end.*

*But in order to produce a complete functional, secure and qualitative program (including documentation, testing stuff etc..) in a short time period can be very tough, stressful and difficult.*

### 3. Root cause analysis

- *If blatant discrepancies and problems were found under the first two points, what could have been the causes?*

*The most cause from our point of view are private causes. A critical root cause is the time management with the other subjects of the semester. Another could be the "real" job in-between the studies.*

- *What could your team do better next time?*

*Better communication between the team members.*

### 4. Experience with the tools used

- *Did the tools used to offer adequate support during development?*

*External tools we used are, Sonar Qube to improve the quality of our code and Docker to containerize the Web App and the MySQL database and to ensure that the data are stored persistently. Post-Man was also involved to test the exchange of the data between the Rest API and our backend.*

*Sonar Qube was very helpful to find and eliminate duplicated code to increase the performance and to reduce the complexity of the program.*

*The experience using Docker to deploy our system was a good one! The simplicity of running a whole application (webapp and database) with just one command and nothing else being installed can't be explained. A very nice tool to use.*

– *Which tools did you have problems with? Which worked well?*

*At the beginning Sonar Qube was working perfectly without problems. After few days, some problems occurred because of the pipelines, and we could not use that tool in the proper way. Some days later the problems were fixed. We had to temporarily test our program by installing Sonar Lint, a local and a bit weaker version of Sonar Qube.*

*About the Docker, at the beginning it took us a lot of time to understand how the docker containers are working and how to program the files in order to run it successfully. After some hours the work was done and the docker was running perfectly.*

## 5. Feedback about Proseminar-Organization

– *Did you find the way the project was carried out useful?*

*The way the project was carried out was useful, because that was a nice real-life and very practical project to work with it.*