

Project Report

student housing apllication

Brandon Lodowica, Dobri Trifonov, Hristo Hristov |

Advanced Software Course |

12 January, 2021

Table of Contents

[Introduction 2](#_Toc61442474)

[Project Analysis 3](#_Toc61442475)

[Who are our Stakeholders? 3](#_Toc61442476)

[Who are our end-users? 3](#_Toc61442477)

[Problems that the residents face 3](#_Toc61442478)

[Which functionalities will our software have? 4](#_Toc61442479)

[Sketch Gui 4](#_Toc61442480)

[The Process and Challenges 5](#_Toc61442481)

[Week 13 5](#_Toc61442482)

[Week 14 5](#_Toc61442483)

[Week 15 6](#_Toc61442484)

[Week 16 6](#_Toc61442485)

[Week 17 6](#_Toc61442486)

[Reflection 7](#_Toc61442487)

[Minutes 7](#_Toc61442488)

# Introduction

Student Housing BV own different buildings which students can rent while studying in the Netherlands. Because the facilities are shared, Student Housing BV has been receiving complaints about: unannounced parties, people do not clean up after them self, garbage disposal is not done in time and more. That is why we have been tasked to create an application to better arrange day to day situations and reduce the number of complaints.

# Project Analysis

## Who are our Stakeholders?

* The owner of the student house (represented by our professor Vucht, Mieke C.M. van)
* The people living in the student house.
* Us (Our 3 software engineers)

## Who are our end-users?

**Our end users are:**

* The residents of the student house.
* The owners of the student house.

## Problems that the residents face

* Appointed persons not cleaning the shared facilities. (Schedules feature)
* Groceries are not done or paid for shared items such as toilet paper, dish soap, etc. (Complaints feature)
* Garbage disposal is not done on time (Schedule feature)
* Unannounced parties, gatherings etc. (Rules feature)
* Agreements between students for compromises. (Discussion feature)

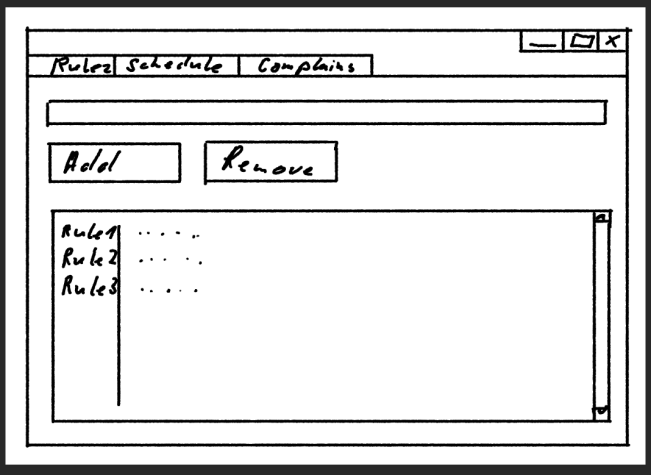
## Which functionalities will our software have?

Me discussed and in the end made the following criteria:

* Allow residents to make a schedule of chores and arrangements between each other.
* Must have a pre-written house rules, which can be seen by the guests but **only** employees should be able to make changes to the rules (make new ones or remove old ones)
* Employees should also be able to gather complaints and print them (change their status).
* Residents can file complaints anonymously with date time stamps.

## Sketch Gui

for the look of our app, we made a simple sketch of how it may look in the end.



# The Process and Challenges

The main challenges that we as a group faced during this project were using Static to access information between different forms, figuring out our final GUI layout. We changed the layout of our application quite a bit going from 3 forms to 2 and then back to 3.

## Week 13

In the first week we go the task to form a group, divide the work, and make a project analysis. After first meeting and then discussing, we had an idea of how we wanted the GUI to look like. So, we started designing the initial version for the GUI together. After the GUI was done, we divide the remaining work among ourselves. Later that week we had a meeting with our project mentor about the project and implemented the needed changes according to the given feedback.

Challenges:

There were no real challenges the first week everything was relatively easy, there were some minor problems when uploading to GitHub.

## Week 14

Week 14 we added a third form, the login form that that opens the Administrative panel or the tenant form, depending on the inputted username and password. There was a Discussion tab added to the form where the tenants can communicate and come to an agreement with each other. After we decide how the classes would be structured and connected, we started to work on our classes individually.

Challenges:

The main challenges this week were figuring out how to work with multiple forms and transferring data (in our case a list) from one form to the other from, because we started early, and it was not explained yet. Another challenge was creating and uploading to a branch on GitHub. We decide to wait until static was explained properly to continue with our project. However, we did solve the Git problem by using google.

## Week 15

We made some minor changes after receiving feedback from our project mentor there was a tenant tab added to the owner form where the owner can add tenants fill in their personal detail (Name, age, gender, Rent) And calculates the total revenue of all the rent of the tenants.

Challenges:

Main challenge was brainstorming what extra functions would be useful to add to the application and figuring out how static works.

## Week 16

After some testing, we figured out how to “transfer” the information from form to form. And finish the main functionalities of the application. There was a time stamp added to the discussion tab. Schedule tab uses a calendar instead of a textbox to plan in a chore and the owner can sort complaint by date.

Challenges:

Main challenges of week 16 were continue figuring out how static works so the list can be seen in the student and owner from. And making changes to make the application as user friendly as possible.

## Week 17

After another meeting with our project mentor. We added timers to refresh the list-box automatically. We implemented that when the administrator form opens the tenants, form closes and the other way around. Added a status for the complaints. We divided the work of making the presentation, changing the design of the GUI, and making the report.

Challenges:

We had minor issue with the timer messing up prior functions which were fixed by slowing down the timer and adding more timers to divide the responsibility.

# Reflection

**What did we learn?**

Working on the project helped us gain more experience about how and when to use Static while working with multiple forms. We also learned how to merge a project using Gitlab and how to Divide responsibilities between Classes . We were able to solidify the material that we have discussed in the Software course by implementing it into a project.

**What would we do different next time?**

Because we started designing and programming the application early and had the goal to finish the main application as fast as possible. On occasion while working some of the needed material was not discussed yet so we would hit roadblocks, where we would have to wait until the material was properly explained. Which also lead to us not using some of the theory we learned later in the course as much like properties. Next time we will try to make a better schedule, where we separate the work evenly between the weeks and gradually finish the application. With the goal of making the best application rather than being done as fast as possible.

# Minutes

**Week 13**

* December 1st : divided the work and started with analysis Document.
* December 2nd : Finished the analysis document and finished design GUI template.
* December 9th : Decided the structure of the classes, how there are connected and who would work on what Class.

**Week 14**

* December 12th  : Finished Classes and Shared our individual progress. Merged all work in Git.

**Week 16**

* Jan 6th : Shared ideas about extra function for application and discussed static. Merged in Git.
* Jan 8th : Shared individual progress again and discussed how to change application according to the feedback , same day show progress extra functions. Merged in Git.

**Week 17**

* Jan 11th  : Discussed PowerPoint, Presentation, and report.
* Jan 15th : edit and finish report.