Student house management system

Georgi Nikolov: 3774163

Daniel Vaswani: 3782689

Khoa Doan: 3603407

[**1. Project Description**](#_iog3ooue2yli) **1**

[**2. Problem Definition**](#_82ingskk68ip) **2**

[**3. Assumptions**](#_v7zoh4a6e7bl) **2**

[**4. Solution**](#_qt0gtktr22jp) **2**

[4.1 Goals](#_kt1okpx7wiwh) 3

[4.2 Expected result](#_3h3g748jv8k2) 3

[**5. Deliverables**](#_x62zur8gefxc) **3**

[5.1 Documentation](#_g47fyk9x15ur) 3

[5.2 UX/UI Design](#_2lreux5te3pj) 3

[5.3 Prototype and testing](#_iyjvd3h7lxen) 3

[5.4 Backend](#_pyroz4mpp286) 4

[5.5 Debugging](#_y050ak4ri0i4) 4

[**6. Way of working**](#_w8dzjol0o69u) **4**

[**7. Risk Assessment**](#_fl965jvelmbm) **4**

#### 1. Project Description

**Student Housing BV** is a housing agency that owns different buildings in which students can stay during their academic programmes in the Netherlands. Each one of the residences has a certain number of rooms available for renting with shared facilities such as toilet, kitchen, basement, bathroom, living room etc. Lately, they are having issues with their tenants, as a result of continuous misunderstandings and people hosting events without the knowledge of their roommates. So far the employees have been receiving the different complaints via post or email, however as it turns out this has no effect on the situation. They are in serious need of a software system that takes care of the organizational matters within each shared accommodation that supports a display of all the chores, house rules, complaints and events in order to maintain the peaceful cohabitation of all tenants.

#### 2. Problem Definition

The fundamental problem within **Student Housing BV** is regarding the flow of up-to-date information about the inner organization of each shared accommodation. The students need to be aware of the responsibilities they have to commit to as per the rental agreement and any other in-house arrangements. Also, a “dashboard” of scheduled events(such as parties, gatherings etc.) should be available to them in order to prevent future surprises and misunderstandings. The landlord, on the other hand, needs to be aware of all the chore agreements the students have between one another so that they can enforce it. Moreover, he should have an updated view on other problems happening in the house such as maintenance issues or it could be that one of the students is doing something that disobeys the house rules/contract.

As part of the analysis process we identified the following bottlenecks:

1. Misinformation and miscommunication
2. Frequency of garbage disposal
3. Facilities not being maintained by the assigned person
4. Time taken to buy depleted items

#### 3. Assumptions

We take into consideration that each building has a terminal for students to use the application as intended. Each house is managed by at least one admin(employee of Student Housing BV).

Another point is that the employees cannot view events that students have announced, however if there is a problem they can view the complaints tab. Moreover, it is only possible for tenants to file their issues only a year in the past.

Additionally, when the students arrange schedules for the chore task, we should consider the fact that the same student should not do the same chore on the same day. Except for that case, any other permutations is valid.

When the employee types in the information about the new tenants, there is a text box called Age where the age of the new student is stored. We have validated this box to make sure that only numbers can be accepted. Moreover, the user can only input the numbers ranging from 0 to 100, which is the range that we found the most logical.

#### 4. Solution

HouseMate is meant to be a robust management system to help students and landlords uphold their respective responsibilities indicated in the rental agreement. Moreover, by creating a functioning GUI application to display the appropriate information when needed, this decreases the probability of future conflict between users.

##### 4.1 Goals

Our technical goals are to create a C# windows form application that makes the distinction between students and employees. By utilizing object-oriented programming principles, we aim to make a readable and understandable codebase for collaboration and future evaluation.

Therefore, our plan is to develop a dynamic application that contains the following:

1. A tab for the ***chores*** that the students are assigned to do.
2. A tab for the ***complaints*** that each students has filed anonymously.
3. A tab for the ***rules*** that every tenant has to obey in order to continue living in the building.
4. Special ***tenants management tab*** for an employee of Student Housing BV that allows him/her to add/remove house members.
5. Special ***events tab*** for tenants(students) that enables them to announce to others that they have an event planned for a particular date.

##### 4.2 Expected result

What we anticipate for this project is to alleviate and strengthen the communication between housemates and ensure their successful cohabitation. With the windows form application developed, we offer a new communication protocol, which is more modern, interactive and powerful between all the housing stakeholders. Additionally, with the solution we came up with, we also effectively help solve the long-lasting issues in the housing industry.

#### 5. Deliverables

Within a time frame of six weeks our team will strive to perform the following actions to the best of our ability.

##### 5.1 Documentation

A detailed project planning is performed, clarifying the problem at hand. We describe the solution and outline our social and technical goals.

##### 5.2 UX/UI Design

Design the UI while keeping in mind that it should be simple, responsive and according to the 10 usability heuristics. It has to be functional and efficient in displaying the appropriate information when needed.

##### 5.3 Prototype and testing

Testing the flow of the program without the code behind the tasks that require processing.

##### 5.4 Backend

This is the main objective of the project. We will be making a hierarchical and understandable code structure. Due to the fact that we show different forms to students and employees, we need to put most of our code into methods so the code is reusable. We need to use object oriented paradigms to make the code as modular and collaborative as possible.

##### 5.5 Debugging

In this stage we have to test the functionality extensively in different ways by anticipating every state of the program.

#### 6. Way of working

The project is executed by our team which is composed of three people in total. Every member has dedicated tasks assigned in a way that guarantees work cohesion and conflict-free environment. Moreover, during the arranged weekly meetings we are able to discuss the challenges that will occur and inform each other of the progress made up until that point.

The main tools that we will be using throughout the project is Visual Studio 2017/2019 IDE, Mockflow, Google Drive and Git for version control. As regards to the techniques, we will employ PERT(program evaluation and review) for project management and OOP techniques for the technical part of the solution.

#### 7. Risk Assessment

* **Skill gap**: During the project, it is inevitable that there will be some knowledge gaps between the members in the group. As the project gets more difficult, a more involved and collaborative attitude is required among all the members to resolve this.
* **Merging problems**: As mentioned in the method section, we use Gitlab as a remote working platform in which we can simultaneously improve the app program separately and securely. Nevertheless, there might be unexpected overlaps, in which case, two team members work on the same part and then try to merge into the master branch. This problem usually happens with git beginners or misinformation during the task division, which is called “merge conflict”.
* **Encapsulation problems**: By applying the object-oriented programming paradigm, we can effectively deliver the most optimal solutions to the problems the Student Housing BV is facing. For security reasons, we also use encapsulation for the program by setting the scope of all the instance variables to private with the intention of preventing someone mistakenly changing the value of any of these variables. However, we can never make sure that there will be no problems or bugs in the code, which is the case that we also need to take into account.
* **Miscommunication between team members**: As it is the first time we have worked together on a project, there might be times when we misunderstand the information from each other. The skill gap and communication skills are also counted as the reasons leading to this miscommunication.
* **Failure to keep up with the indicated progress**: It is required and important that we have to make a list of things to finish every week in order to keep track of the progress we have made and avoid any unnecessary task overdue. However, there will always be problems that might happen throughout the project that we can not predict. Therefore, it is a good idea to build a back-up plan to make sure that the project solutions will be delivered on time. For example, it could happen that the program can have bugs or errors that can slow down the speed of the team.