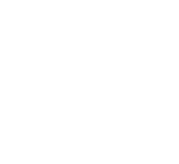
**ADVANCED PROJECT  
STUDENT HOUSING SOFTWARE SOLUTION**

****

Ivan Z. Ivanov (S.N. 3782409)

Martin R. Georgiev (S.N. 3782484)

Aleksandar H. Ivanov (S.N. 3781488)

Fontys University of Applied Sciences

# Introduction

This report is an overview of a developed coursework project aimed to resolve certain issues for a housing agency called “*Student Housing BV”* and further improve the user experience for both the tenants and the administrative staff. The purpose of the report is to reflect on the group workflow and to analyze the reason for the decisions made during the period of development.

This document will go through:

* The contextual background of the project;
* The problem statement, intended to highlight how the user of the application can find a solution to the problems they are facing using the built-in functionality of the software;
* The group project workload – the process & results; a conclusion;
* Future improvements;
* Individual self-evaluation & reflection for each group member;

The developed software solution serves as a platform where each group member sought self-improvement and tried to innovate by coming up with valuable ideas. Each member shared their vision of what a perfect student housing application might look like and worked hard to achieve a fully working product.

# Contextual Project Background

Student Housing BV – a housing agency, operating in the Netherlands, started receiving complaints from their clients related to the miscommunication between the tenants in their housing units. Those complaints mainly revolved around the maintenance of said units – keeping each facility clean, resupplying the open-area’s necessities, garbage disposal, unannounced gatherings, and parties, etc.

Consequently, the housing agency is determined to find a better way to operate said housing units. Their idea is to provide each tenant with a software solution as part of their extra services with the goal to improve the communication between tenants.

Before the group started work, there were a couple of factors that had to be considered. The agency wanted each tenant to be able to: see the house rules at any given time; file complaints anonymously; record and see agreements between the other tenants in the same housing unit. Moreover, each staff member ought to be able to change the house rules easily.

The group was fully committed to providing a fully working application that will solve all complaints while keeping in mind the occurring problems & factors. After several discussions, the team came up with different ideas on how to proceed with the assignment. One of the things we used was the *MoSCoW* *(Must have, Should have, Could have, and Won't have)* method of planning and prioritization in order to meet all project deliverables dates.

|  |  |
| --- | --- |
| Must have | * Cleaning schedule that will include a client’s name and the facilities he has to clean in upcoming weeks. * A grocery list with predefined shared products. Info whether they are available and whether it has been paid for them * Garbage takeout schedule with a possibility of either mixing and displaying it on the grocery list schedule or a standalone window. * Anonymous report screen so that each student can inform the company of any issues or complaints that they may be having. * Events list in which everyone can create their own event be it a party/ gathering/ teambuilding/ etc. Each user will have a limit for a certain amount of time so that the system does not get over-spammed. * House rules screen * Simplistic and intuitive design |
| Should have | * Admin (employee) window with the ability to see each event, scroll/read through all of the reports made, list apartments with their tenants (possibly add/remove ones) and the option to update house rules so that no employee has to manually go to each building and do it. * A social tab with a chat where tenants can directly communicate with each other. |
| Could have | * Preferably an option for each client to be able to re-schedule stuff/record agreements made between him and other member(s). * Progress bar to the “Garbage takeout schedule” which using Arduino will determine how full is the trash can and indicate it 24/7. * Ability to add/remove housing units. * Added a database to the software. * Ability for tenants to request a new password (in case they forgot their current one). |
| Won’t have | * Credits system. Ability to buy credits and pay groceries with them. * Ability to change the appearance of the application using themes (Dark mode). * Pictures for each tenant. * Communication with remote MSSQL server. (The group opted for a local database instead). |

# Problem Statement

This section of the report is here to showcase how the client’s potential issues could be resolved using the implemented software solution.

**ISSUE:** As a tenant, I don’t have an easy way to tell the other tenants about the events (party, gathering, etc.) I want to organize. Even if I tell them, they might forget.

**SOLUTION:** With the built-in events system, each tenant can create a new event at any time. The other tenants will be able to see them on a calendar, displaying all events for the month.

**ISSUE:** As a tenant, I don’t have an easy way to arrange the cleaning schedule accordingly with the other tenants. There might be disagreements as well.

**SOLUTION:** The software application offers а built-in cleaning scheduler that will keep track of who needs to do what instead of you.

**ISSUE:** As a tenant, there are often disagreements on who needs to do the shopping next as we forget. A shopping list is rarely made.

**SOLUTION:** The software application offers а built-in groceries system where tenants will be able to add items to the shopping list, be able to see the current and next person who needs to do the shopping.

**ISSUE:** As an administrator, I waste too much time going to each housing unit to change the house rules.

**SOLUTION:** Administrators will be able to change the rules within the application.

**ISSUE:** As an administrator, if no complaints are sent via e-mail, it is often too late to act on time as tenants make their complaints only when I come to gather them. That also takes valuable time of my workday.

**SOLUTION:** The application has a built-in report system where administrators can easily view reports and be able to reply.

# Process & Results (partially individual)

## Like every other project at the start, you must have a general idea of what you would like to accomplish. The team’s vision was quite clear and yet, in the end, there were quite a lot of changes. The process of creating the application involved a lot of planning beforehand which for the most part was established through the usage of “Trello”. Every day there were discussions on what could be improved, bugs and problems in the program and new ideas that were/could be added as extra.

## Every individual in the group contributed to this project almost as equally as the other did. The group made an effort to split the workload and ensure that not only would each person do coding but also focus on the part that they feel most comfortable about doing. Below you can find a brief description of the contributions. Processes such as bug fixing, adding quality of life features, etc. apply to everyone and are not separately mentioned.

## **Martin Georgiev**

Shortly after finding that the group would not manage without a Database he was already working on researching how to do it and implement one. To make life easier for the others he made all of the necessary classes and a UML diagram so that they could use them to their full potential. He was very active in researching everything and decided to create his very own Calendar for the application. That and the database were very time-consuming and somewhat hard to do.

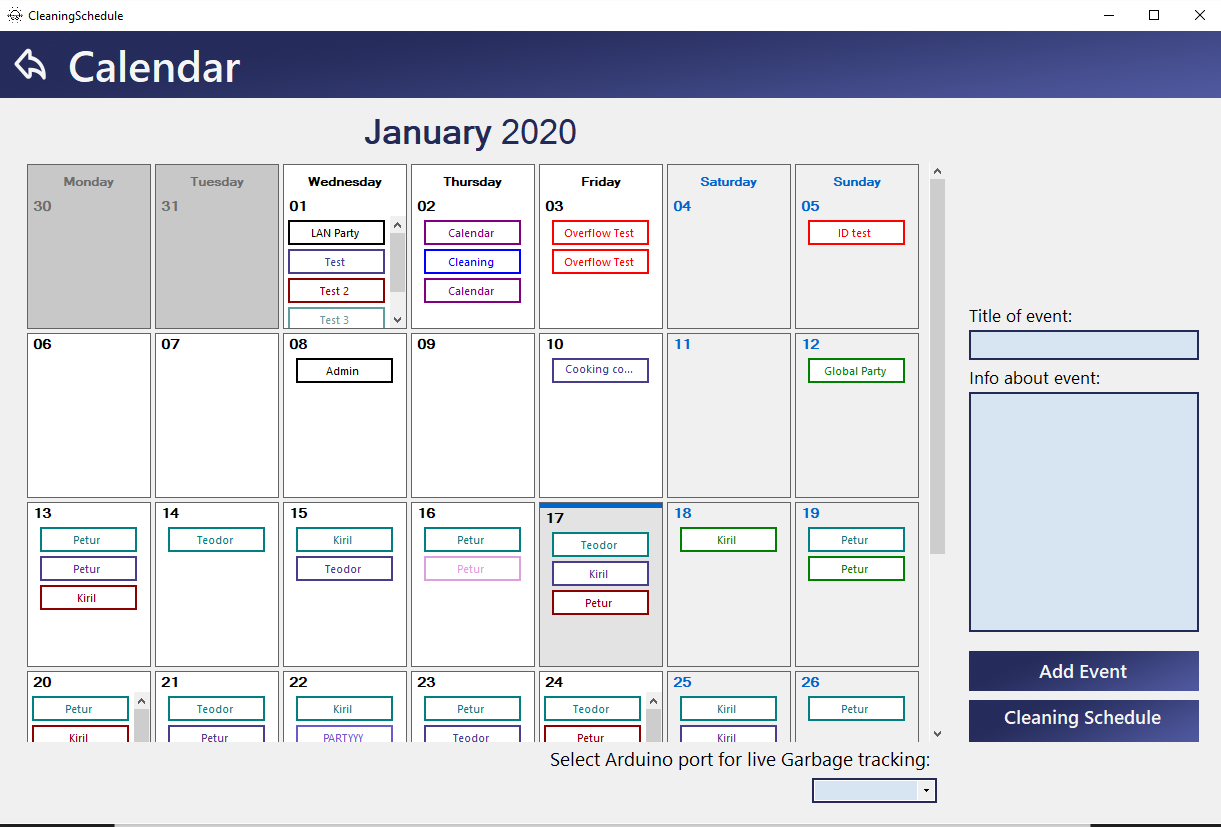
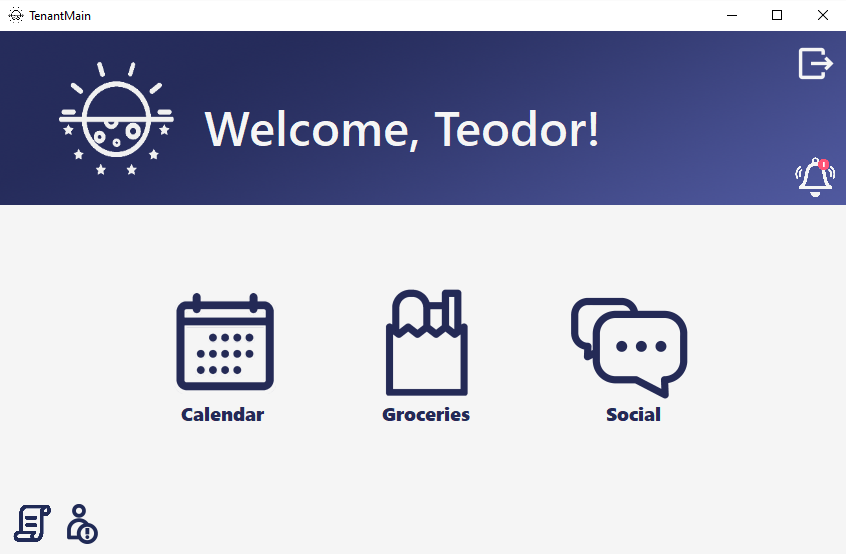
## **Ivan Ivanov**

## After having discussed the envision of roughly how the team would like the final application to look like, he took the task of creating sketches of different UI’s. The first version of the Groceries menu was also done by him and every single part of the GUI. The challenges that he met were understanding how the database works and customizing controls to make them look better.

## **Aleksandar Ivanov**

Since the very beginning, he challenged himself that he would make the trash can work with Arduino and so he did. Made work on the second version of Groceries which involved moving the data to a database, finished the Social and Report menu. In the end, as an extra, he also made the Notifications menu, which proved to be another hard and full of research part.

Картина, която съдържа екранна снимка

Описанието е генерирано автоматичноThe final version of the application exceeded the expectations of everyone. While collaborating on creating it, more and more extra features came out of the blue, ready to be implemented. Upon spending so much time on it, with heads held high, the group can be very proud of the accomplishment. Tenants will no longer face any problems with organizing and communicating with their housemates or the company itself and neither side will need to spend more resources to achieve that. The complete application was made while keeping in mind that everyone should be able to understand it and make use of it without needless effort.

# Testing

## Introduction

After finishing the functionality and the GUI of the Project, it was time to continue with testing the app for various bugs. We conducted a test on our own as well as 3 with people, who haven’t seen our Project. We found various design and user experience flaws as well as some functionality issues, some of which are mentioned below.

## Own testing

* When adding a house unit, the unit ID can be written with an 0 in the beginning and saved as a 2-Digit number, which ruins our concept of having only 3-Digit numbers as house iD’s*. Fixed with a simple check.*
* Notification Panel felt too big when there were not that many notifications in there.

*Fixed by adding a responsive panel size when there are fewer notifications*

## Test 1

* You could make an event in the calendar without actually giving it a name. There was text in that textbox, so it was taking it as a name. *Fixed with a simple Boolean check*
* Sending messages in the chat took your name twice. *Fix: found a mistake in the string that was getting sent as a message and fixed it there.*
* Announce button was ugly. Fix: *Icons with transparent background*

## Test 2

* You could write in a lot of combo boxes in the Project, making the program crash due to invalid input. *Fix: Made all of them read-only.*
* The menu was hard to navigate through. *Fixed by adding labels in the menu.*
* “Tenants” with empty entries were found. *Deleted manually.*

## Test 3

* There was a $ symbol on both sides of the Payment history list.*That was a bigger issue than we thought, as we were using some string. Split functions by that symbol, so we had to change some functionality to fix that.*
* Notifications icon doesn’t change whenever you close all notifications. *Fixed*
* You could put symbols in your user’s first name that would bug out the non-SQL Database. – *Fixed by making a check for those symbols on admin account creation.*

## Testing conclusion

Taking the time to make a user test proved itself to be extremely useful in such an app with so much functionality, as we found a lot of small bugs that we wouldn’t have if we submitted the app directly. Fixing them made for a way more robust application, which is our main goal.

# Conclusion & Recommendations:

At last, the application was developed to be very close to professional software and that is a big accomplishment. It fulfills all requirements of the client and on top of that adds even more. Testing and polishing brought to light new improvements that allowed smoother user experience. An important quote kept in mind was that simplicity rules over all. It contributed by bringing an easier to understand software and to please the majority of people.

For the future, if we were to extend and maintain the app we would definitely need to reconstruct some of the coding to make the application work faster and better than it already does. After having access to data from users who tested the application we can reap the benefits of it and make new assumptions on what works good, what could be changed and what could be kept the way it is.

Before starting to maintain the application we would first need to bring the local database to a server hosted one. Furthermore, additional good to have features would be the ability to change password, improved personal identification through the form of pictures to each tenant, private messaging with people, the hit of the generation – dark mode, etc.

# Self-evaluation & Reflection

## **Martin:**

Personally, I think that the project went very well. If given the opportunity, I would like to work with the same group for my next coursework project. The only thing that the group had issues with are the git conflicts. They took time to resolve and on occasion lines of code were reverted to an earlier state, resulting in loss of information.

I am most proud of the amount of work we were able to do in the time period we had. A lot of the systems required extra research to develop. Nevertheless, the application has many well-built features.

## **Ivan:**

The development of the project was way better than expected. I liked with how much passion everyone did their part of the project and spent days working on it. My teammates’ enthusiasm was very contagious and got me to work constantly on what was expected of me, although it can always be more. Git was a bit tough to deal with but we managed.  
 In my next project, I would definitely follow and improve the steps that were needed to finish this one and would be glad to keep the same group. As much as I’d like to work with new people and experience their workflow, the Introduction part showed me that being in different groups more or less can be rather terrific. There is also a saying that if it works, don’t change it. For the next coursework, I would try to do even better at coding and deliver stuff even earlier.

It is a really hard decision on what I would be most proud of. The user interface looks really good, the options that the application provides are of a wide range and everything works as expected. The amount of time that we had was short and during holidays no one worked so I am quite happy with what we managed to get done. To conclude, I learned a lot because of my two teammates and the research I had to do, so those two things would be what I am most proud of.

## **Aleksandar:**

I am very happy with the way our project developed and the time I spent on it - it was a lot, but it felt very motivating to work alongside my 2 team members. Researching a lot of the functionality with Martin was a really fun process and I did not mind staying up until late night (or early morning) doing that, and I've clicked with the way Ivan works ever since our introduction project, I'm very happy of his work, as he was the person who had to call the project "finished", and that is a task on its own. I would gladly work with them again on my next projects in Fontys.

We've already discussed that we need to start developing new techniques for the next projects that we research ourselves - WPF forms, Web DB instead of a Local one, etc. As for what I would change, I'd like to see even more planning ahead, although it was good now with the fact that we divided tasks and used Trello, I'd like it to happen at planning stage instead of the working one, as I personally am way more productive when I know exactly what needs to be done by me.

I'm most proud of the fact that for the project we researched a lot of new concepts that were not taught to us in university (User Controls, Flow Layout Panels, Databases to name a few). I like adopting that type of mentality - you need to work with something, you just start doing it and find a way to make it happen. I'm also very happy about the fact that this project is just a few steps to a finished product, which seems very satisfying, it being only 20 weeks into programming.