StudentWise

Project Report

Denis Nagayuk, Evgeniy Terziev, Jonasz Kądziela, Karina Kozarova

**Table of contents**

[**Introduction**](#_5m0o59wl9kv5) **2**

[**Background**](#_k4smhff6113p) **2**

[**Problem statement**](#_51kvfe57e5ad) **3**

[**Process & Results**](#_p0sudoinvrq) **3**

[Denis Nagayuk](#_9002qfch5laq) 4

[Evgeniy Terziev](#_ih17dg31h7y9) 4

[Jonasz Kądziela](#_3170ldbell7z) 5

[Karina Kozarova](#_iaxyn5qebcme) 5

[**Conclusion & Recommendations**](#_ttntzf2e6dhs) **6**

[**Individual Evaluation & Reflection**](#_16tq1xp31xgn) **6**

[Denis Nagayuk](#_fs4g2qlvoz15) 6

[Evgeniy Terziev](#_fcy3qghitaa) 7

[Jonasz Kądziela](#_v0wj88ru633a) 7

[Karina Kozarova](#_e3zdlk3qx85j) 8

# Introduction

The aim of this report is to showcase all of our workflow and material that we did throughout the entire advanced phase of our first semester. The report outlines our ideas about how to make the living in a shared facility not so much of a burden for the students and it also mentions all the reasons that backed up those visions about this project. Furthermore, the report provides a clear overview of everyone’s work on this task and everyone’s own reflection of the work process within the team.

# Background

Nowadays, many students live in shared facilities with other colleagues, which is not always the most pleasant experience. Before starting this project, many students complained about someone from their facility, who has not done their chores or is making a huge mess around the house.

Generally speaking, nobody wants a messy roommate or someone who doesn’t do anything in the house, so we had to take many factors into consideration. One of the most important factors was spreading the household chores equally amongst all the tenants so that there are no quarrels between them. Another major factor for us was to make sure that everyone from a given accommodation is able to see upcoming events and is able to agree or disagree on them, as some unexpected parties can sometimes ruin your day. Furthermore, we wanted to make sure that expenses for shared items, for example toilet paper, paper towels and others, were spread evenly across all tenants.

Considering the fact, that usually students tend to be a bit lazy we wanted to find a way how to stimulate them to do their chores, so we came up with some ideas, for instance, a karma system that tracks the points earned by the tenants for doing their chores, that we might implement as extra features later on.

Lastly, there are sometimes students that are a bit wild and tend to break some things, that’s why we thought of implementing a section where you can complain if somebody has done something wrong, like breaking a piece of furniture or not completing his tasks.

# Problem statement

When dealing with shared housing, people encounter lots of different problems - from someone not taking the trash out, to your roommate throwing parties the night before your biggest exam and even problems as insignificant as not being able to report when one of your flatmates is misbehaving or giving you a hard time.

All of this reflects a lot also on the owners of the buildings - they are supposed to deal with all the problems that arise. To simplify this problem, our client (a student housing business owner) needs an ICT solution that will help them manage all of this easier so that they can focus on other parts of their business.

Our solution’s main focus is tracking events in the building and keeping track of the complaints. To achieve this, our features include having accounts for the users. Every user is able to create an account and login with it. We also support having multiple sessions on the same device. For example, all the roommates can log in to their accounts from a single device.

Once they are logged in, they can create events. For convenience, we decided that the event creator should also be a participant of the event by default. The creator of the event can also delete it. When you delete an event, all unnecessary relationships such as participants and votes are also removed. We also have a listing of all the events a user is participating in, which makes it easier for the user to keep track of his upcoming tasks and another view where the user can see all the events they have created.

To prevent anybody from spending too much money on shared items, we also implemented a list that contains all the expenses from a given accommodation. This way everyone can see how much money they should give or receive at the end of the month. Lastly, we decided to add a tab where the tenants can make agreements between each other so that there are no misunderstandings. Those agreements cannot be deleted or edited for safety reasons.

# Process & Results

To achieve our goals, we decided to split the work horizontally - Rails REST API, thin client and GUI client. We also decided to have separate repositories - one for the API and one for the GUI. We also have an extra one for embedded solutions which would be more widely used in the future. For project management, we choose using a Trello board - it was a simple yet elegant solution in our case. At the beginning of the project, we also did an estimate to see if we can fit with the deadlines, got some feedback on our ideas from our fellow classmates and prioritized the tasks. Overall, we as a team think that our job was very successful even though not fully finished because of all the extra features we still want to implement.

## Denis Nagayuk

My role in the project development was to create a level of abstraction on the C# side of the project that isolates the front-end development team from the details of the API implementation and client-server interaction. To achieve this goal, I needed to understand both the limitations of the API and the needs of the user interface developers. I worked in tight contact with the cloud server developer, discussing diverse approaches to solve problems arising while we add new features or introduce more intricate interactions between existing concepts.

The major part of my job was to expose a set of classes/objects that represent well-known entities and provide a set of methods/properties to retrieve and update data on the server seamlessly. The exposed interfaces are supposed to be minimalistic and simple to use, yet flexible when required. I believe that the result of my work meets all the specified requirements.

## Evgeniy Terziev

My main role in this project was developing the C# application. I was working alongside Karina on the GUI, where with the help of Denis’s thin client we had to receive information from the server as well as send the necessary data to the server through Denis’s client.

One of the challenges I personally had was that the level of difficulty of the project plan that we came up with was higher than my skill level, so I tried my best not to fall behind and work alongside the others. Furthermore, I tried participating in every discussion concerning the project and I gave feedback whenever necessary.

## Jonasz Kądziela

During the advanced project, I undertook the implementation of a layer between C# backend and the database. I decided to develop RESTful API using a very popular web development framework named “Ruby on Rails”. This decision vastly contributed to the security and reliability of the application.

The most significant challenge I faced during this project was JWT (JSON Web Token) integration, as I have never worked with it before. Moreover, I was unable to develop anything else until the JWT feature was finished, which created an undesirable bottleneck in the whole team workflow.

I am convinced the final result of the RESTful API is astonishing. During work, I followed the best practices of web development, such as: sharing as little sensitive data as possible with the API consumers, writing automated tests, dockerizing the whole application or even implementing a CI/CD pipeline to the GitLab repository.

## Karina Kozarova

My main role in the project was working on the C# app. I think I contributed a lot to the GUI. I was crafting the GUI with the help of Evgeniy and we had to add all the needed validations and receive information from the server using Denis's thin client and then sending the needed information to Denis's client who will then send the appropriate request to the server. The level of abstraction we had was very nice but it sometimes slowed down the development part of the GUI because we needed to have both the API and the thin client ready to work on most parts of the GUI.

However, I also did a lot of project management - I spent a lot of the time working on keeping the Trello board up to date, making new tasks and assigning them to the appropriate people. I also had to do some planning in advance to set deadlines for the tasks. Another thing I did was making mockups in the beginning which from the feedback I got from Evgeniy was very useful.

# Conclusion & Recommendations

Although we were very ambitious at the beginning, coming out with a lot of useful and creative ideas, due to time constraints we had to pull back and focus on finishing the requirements of the task before everything else. We wanted to implement features like two-factor authentication done with Arduino and point system, which allows for flexible division of responsibilities amongst tenants, but unfortunately, we didn’t find the time to implement them. Of course, these are great ideas for the future development of the app. We understand that we can’t solve every problem there is in such a short time so we focused on those features that would significantly increase the overall satisfaction of the users. However, we truly believe that by implementing all the extra features we have thought of, our app will definitely be very attractive to the end-users.

After all, we managed to finish all the must-have requirements that we set out to do thanks to our teamwork and good communication amongst the team and we ended up with an outstanding app that we believe the users will love.

# Individual Evaluation & Reflection

## Denis Nagayuk

Although I have practical experience with programming and a fair understanding of various architectural approaches and paradigms, some pieces of this assignment were considerably novel for me. I have never tried to write a reliable and robust system on top of network communications with a cloud API. Unsurprisingly, a lot of things can go wrong, so you need to try your best to consider all possibilities and perform error handling appropriately. Especially, if you don't want to expose internal errors to the caller unprocessed since they might look quite mysterious out of context.

Anyway, my part of the project went great: I implemented everything the way I wanted to. I am satisfied with the overall architecture of my abstraction layer, and, in particular, the way I handle JSON and filter events. They both are minimalistic, yet powerful and extensible.

## Evgeniy Terziev

Going into this project I knew I would have to work more than usual because my teammates are much more knowledgeable than me in the ICT field. So, I expected to face some issues during the making of this project. Although I struggled here and there while working on the project, I tried my best and with the help of my teammates most of the time I managed to finish my task.

One thing that I would definitely do again is meeting with my teammates before starting work on the given task and discussing our workflow. For instance, that’s what me and Karina did before starting work on the C# application and it helped a lot further on in the creation of the application. Of course, we faced some issues throughout the process, but we managed to solve them with ease. One thing I am definitely proud of is that while working on the C# app with Karina, our communication was always on point and both of us knew what the other was doing so that there are no misunderstandings between us.

## Jonasz Kądziela

I consider the project a great success. Not only did we manage to implement all of the fundamental features, but also a few additional ones. The work in our group was well-organized and divided evenly. Every person knew what our vision is for the project and what has to be done by them. This was mainly the result of Karina's management skills.

Should I do a similar undertaking in the future, I would try to use well-established technology to develop the project again since it simplifies work a lot and makes it more enjoyable. On the other hand, next time I would unquestionably research certain concepts and technologies before the start of the project as it would save me a lot of time and frustration.

I am most proud of the documentation of the API which is dynamically generated from automated tests. This was my first time working with Swagger, and I have to admit that it is immensely useful. Furthermore, all of the tests are automatically run when someone pushes changes to the GitLab repository. That way, we always know whether the API is in a fully-functional condition and nothing stopped working after the implementation of new features.

## Karina Kozarova

I have a lot of experience in the ICT field but the part of application development I got to work the least on in my career was frontend and especially with C#.

Because of that, I expected a very bumpy road on this project and so I prepared in advance. Before we actually started writing code, I made a meeting with Evgeniy to plan our work, talk about conventions we are going to follow and I also made an estimate on all the tasks we wrote down. I think it was very useful and I would definitely do it again. Even though it takes some time at the beginning of the project, I'm sure it saved us a lot of time in the long run.

During the development we encountered some of the most usual problems in software development - merge conflicts. However, our attitude towards them was positive because our git workflow was using feature branches so having conflicts was logical in contrast to some of our fellow students who only used the master branch. Another thing I am very proud of was that me and Evgeniy were very consistent in our work. We didn't have many sleepless nights because we planned well. We also had great communication on sharing progress and giving each other feedback.