StudentWise

Project Analysis

Denis Nagayuk, Evgeniy Terziev, Jonasz Kądziela, Karina Kozarova

**Table of contents**

[**Problem definition**](#_5m0o59wl9kv5) **2**

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

[**Target users**](#_51kvfe57e5ad) **2**

[**Goals**](#_s5jlnrisv5xq) **3**

[**Technology stack**](#_16md2bvvxtox) **3**

[**Functionalities**](#_p3ir9mmyu5yl) **3**

[Must-haves](#_zc04e0jqjrdx) 3

[Nice-to-haves](#_u4hqag21lvr) 4

[Extras](#_as1fgshvyj1f) 4

[**Vision**](#_44ru0guas1wf) **4**

[**Business model**](#_sipo1th9vzou) **4**

[**Dependencies**](#_chx0rjf23sy7) **5**

[The health of team members](#_18rx7pbcqjp5) 5

[Ruby on Rails framework and gems](#_gep2efy5cdfh) 5

[Development and management tools](#_4ns22x1qr4k9) 5

# Problem definition

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.

Our solution’s main focus is tracking events in the building and keeping track of the expenses. We also think a big improvement will be monitoring the environmental conditions in the common area in a later period of the project.

# Stakeholders

Our main stakeholders can be divided into the following two groups:

* Students who are living together with their peers in the same place (building) and use shared facilities
* Landlords (companies or individuals who rent housing to groups of students)

However, according to [Karl Wieger's Book on Software Requirements](https://www.amazon.com/dp/0735618798/), a stakeholder is a person, group, or organization that is actively involved in a project, is affected by its outcome, or can influence its outcome. Thus, the developers (Denis Nagayuk, Evgeniy Terziev, Jonasz Kądziela, Karina Kozarova) are also stakeholders.

# Target users

Our main target are university users who are currently living in shared accommodation (they share a facility or have roommates). This means that our target users are on average aged between 18 and 26 so our solution should be appealing mostly to young people.

# Goals

* Improve and simplify the management of responsibilities each student has
* Track the events happening in the building and allow for planning new ones
* Introduce an easy way to keep track of expenses for shares items, for instance: dishwashing soap, paper towels, toilet paper
* Allow for reporting misbehavior or breaking the rules to the landlord - both personally and anonymously
* Incentivise students to stick to the schedule and fulfill their obligations
* Provide monitoring tools which will help reduce monthly expenses for the landlord

# Technology stack

For our solution, we have decided that our tech stack is going to be the following:

* **VCS:** GitLab (Multiple repositories: C# client, RESTful API, Arduino)
* **Main Application:** C#
* **Containers**: Docker (for the RESTful API)
* **RESTful API:** Ruby on Rails (RoR)
* **RESTful API Documentation:** Swagger
* **Database:** PostgreSQL
* **Project Management:** Trello for the major features and GitLab issues for the development
* **Communication between team members:** in-person and WhatsApp

# Functionalities

To deliver the best end product, we decided to split up our features into the following categories:

## Must-haves

* An account management system which allows for signing in and creating new accounts
* RESTful API running in the cloud, connected to the database and responding with JSON data
* A thin client which retrieves the data from RESTful API
* A system which tracks events in the building
* A system which keeps track of expenses

## Nice-to-haves

* Monitor the environmental conditions in the common area using Arduino
* Karma/points system which allows for flexible division of responsibilities

## Extras

* A notification system which after an arbitrary amount of time notifies the relevant students to return the due amount
* Provide optional 2FA with Arduino as a token

The basic idea is that we will start with implementing the must-haves. After they are finished and we have been assured that the code quality is good enough, we will continue adding the nice-to-haves features. In the end, if we have time left we will also add the extra features. However, we believe that quality should be before quantity so before every major release, we will do code reviews.

# Vision

We strongly believe our application could simplify the life of every student who will be using this system. Furthermore, it will reduce the number of communication issues between people.

While planning this project we had scalability in mind, thus, the system should work perfectly for any number of users after an appropriate configuration.

# Business model

To monetize the app, we have decided to use the freemium business model. It is one of the most favored ways of monetizing an app by offering it for free. This way the users are more likely to spend cash on the premium features of an app if they are happy with the basics. However, this would be implemented in a later phase of the project and is not our concern in the following project stage.

# Dependencies

## The health of team members

We are a small group of developers with a very limited amount of time to deliver this project. Thus health issues, even if they affect only one person, can cause significant delays.

We could mitigate this issue by doing everything to stay healthy. Additionally, in case of inaccessibility, each team member should have a substitute within our group.

## Ruby on Rails framework and gems

Furthermore, we will create the RESTful API with Ruby on Rails framework and various gems, which could potentially contain security vulnerabilities.

To avoid this problem we should inspect all of the dependencies before we include them in our project. We should make sure that they are free of any publicly-known security vulnerabilities or bugs. However, there always can be an issue which has not been discovered yet.

## Development and management tools

Last but not least, during our development process, we will be using tools such as FHICT GitLab, Trello, Google Docs. In an unlikely event, some of these services could malfunction which would slow down or potentially even break our workflow.

To mitigate this matter, we should have substitute tools, which we can use until the main tool is restored to its original state.