Student Housing B.V.

By

Matei-Cristian Mitran

Ioan-Nicolae Popa

Alexander Bogdanov

Fontys University of Applies Science

Software Solution for student houses

ICT & Software Engineering – Advanced project

INTRODUCTION

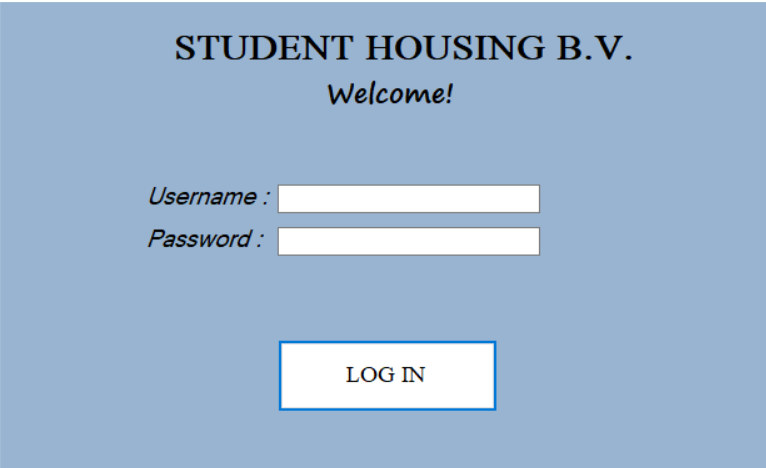
Student Housing B.V. is a housing company where students can be accommodated during their stay in the Netherlands.

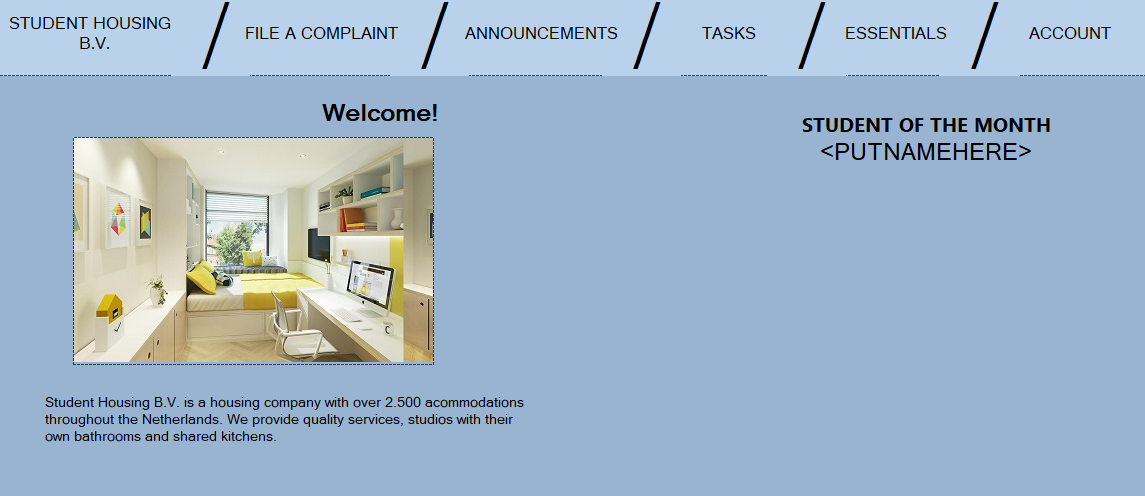
We were tasked by Student Housing B.V. to design a software application to organize and keep track of the daily activities of the tenants living in one building. Also, the application should also provide an admin side where designated admins could make changes and updates on the tenant side. Some of the complaints of the clients of Student Housing B.V. are:

* Unannounced parties
* Groceries are not done or paid for shared items
* Appointed people are not cleaning the shared facilities
* Garbage disposal not being done in time

DESIGN

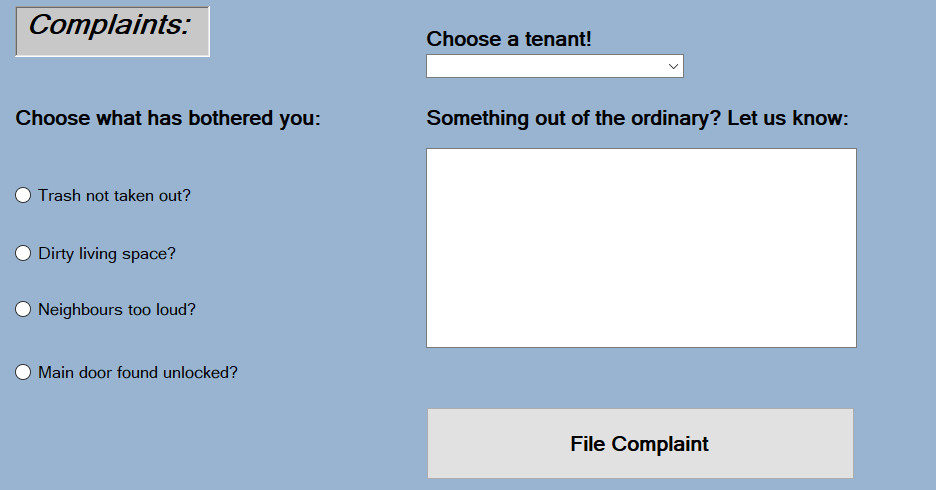
We decided that the best course of action would be to start working on the application by creating the design first. The resources for the design of a Windows Form Application are limited in Visual Studio C# but we tried our best to make it feel as professional as possible.

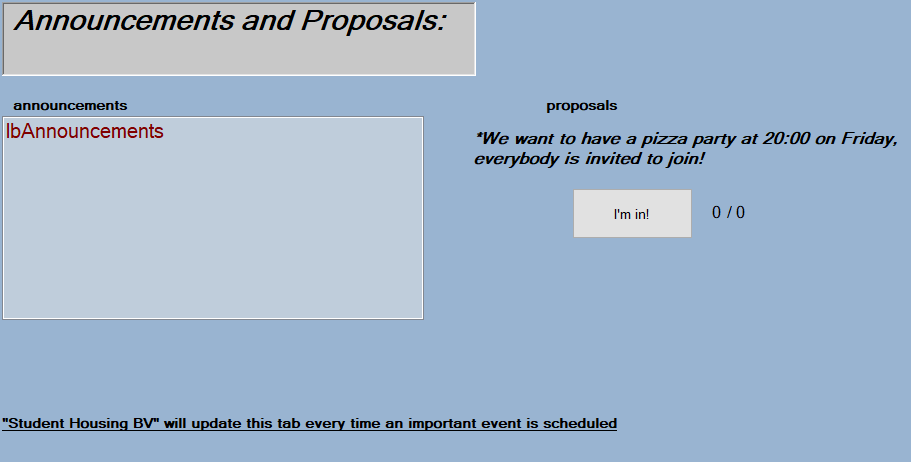
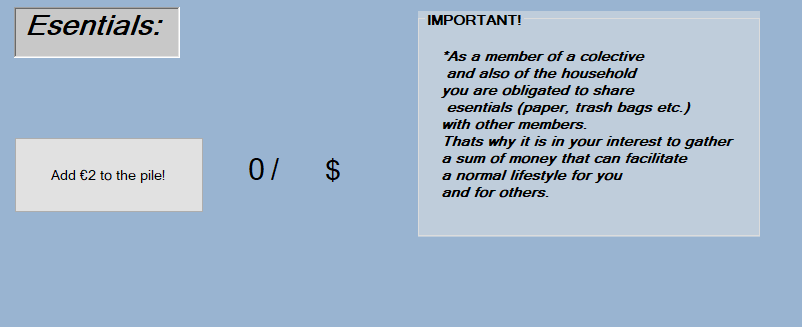
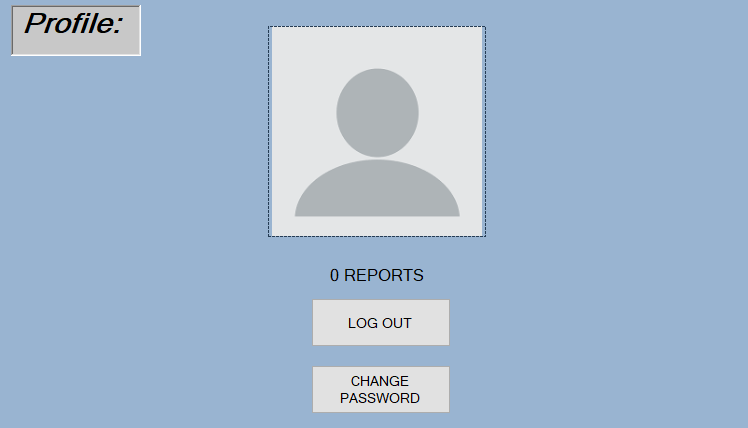
 First, we designed the Log-In Form where we added a label containing the name of the company and another label welcoming the user to the form. We added two textboxes, one for the username and one for the password. The characters in the password textbox are hidden. Finally, we added a button to log in:

Then, we designed the Tenant Home Form, which is the first thing the tenant should see after he logs in. This is also the hub that makes the connection between each of the tenant forms an allows a user make decisions:

On the upper side of the screen there is a range of buttons that allows the user to do the following:

* File a complaint: pressing this button will open a new form that allows the user to file a complaint based on the problems.

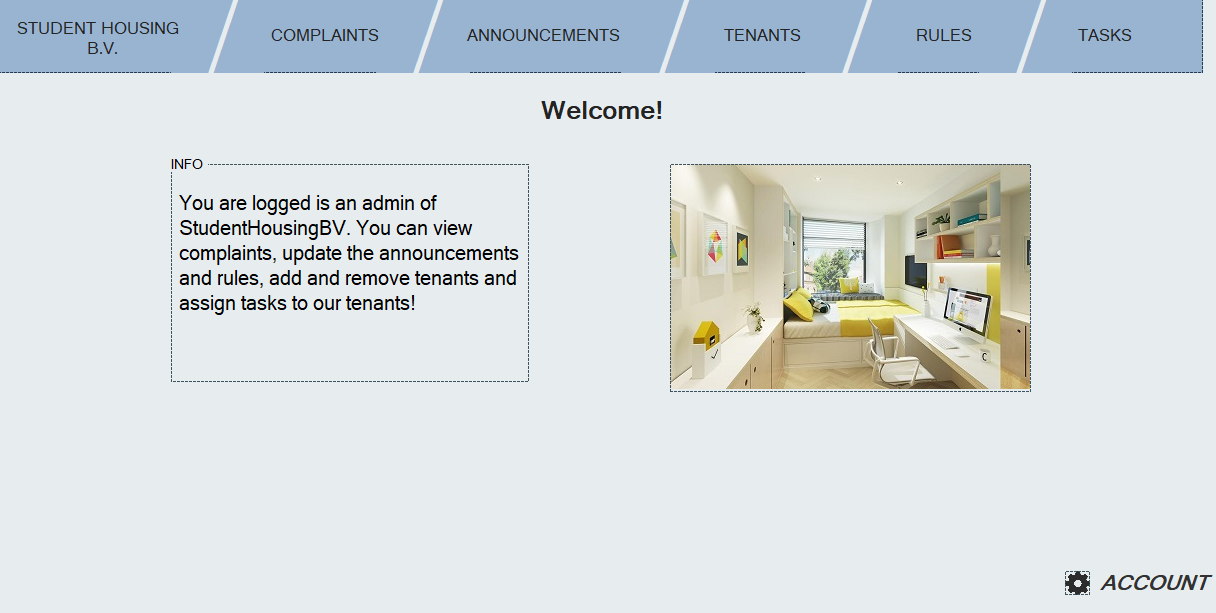
The user can select from a predefined category of complaints that are more commonly encountered in the house or can create a personalized complaint that better suits the situation.

* Announcements: once here, the user can see the latest announcements placed by the owners of Student Housing BV and can also show their opinion based on a proposal made by one of the tenants. Depending on the opinions of others, the event takes place or not.
* Tasks: from time to time, Student Housing BV will attribute tasks to tenants. Along side a task, a due date is also issued to one of the tenants. The user can see here if they were assigned to a task or not. If the user completed the given task, it should be able to press the finish button and show that the task was completed.
* Essentials: once a month, the company will call for a common fund in which the students must put a sum not bigger that a few euros. That sum will later be used to buy the items required for a normal lifestyle (i.e., paper, paper towels, detergents, soap, and other consumables).
* Account: here the user can see the number of times it has been reported, can change the password of their account, and can also log out.

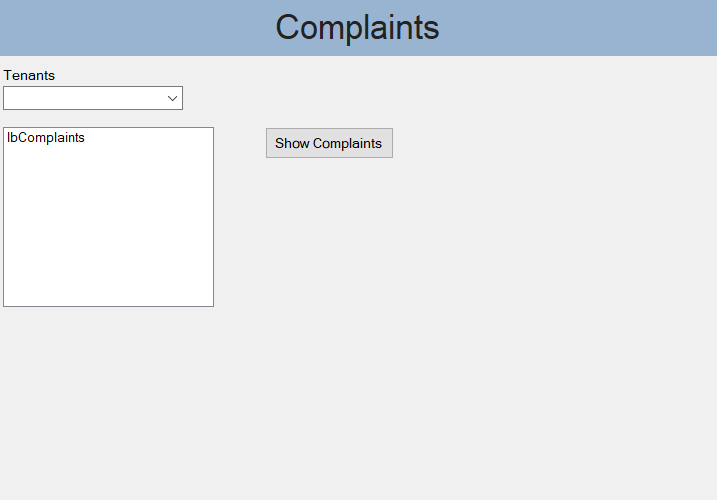
Once the “LOG OUT” button is pressed the user will be presented with the log in tab where it can input other credentials or the same ones as before.

Pressing the “CHANGE PASSWORD” button will allow

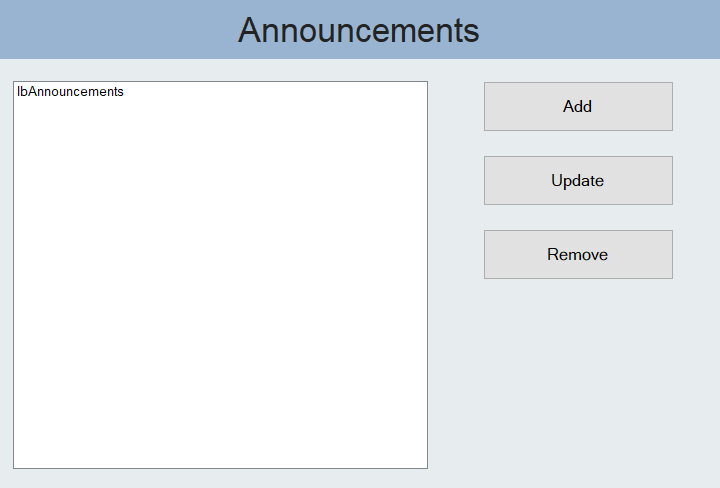
The application has in mind not only the tenant but also the company itself, Student Housing B.V. With this we managed to create a separate interface that allows easy access for admins to log in and manage the tenants.

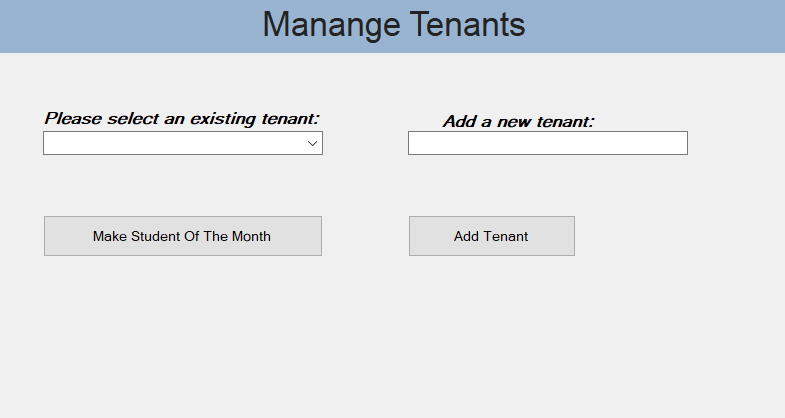
* Log in: the login page is identical to all, what really sets the difference sis the inputted data. If the user is an admin and logs in accordingly the screen that will be shown will be this:

Similar to the tenant side, the admin sees the same categories.

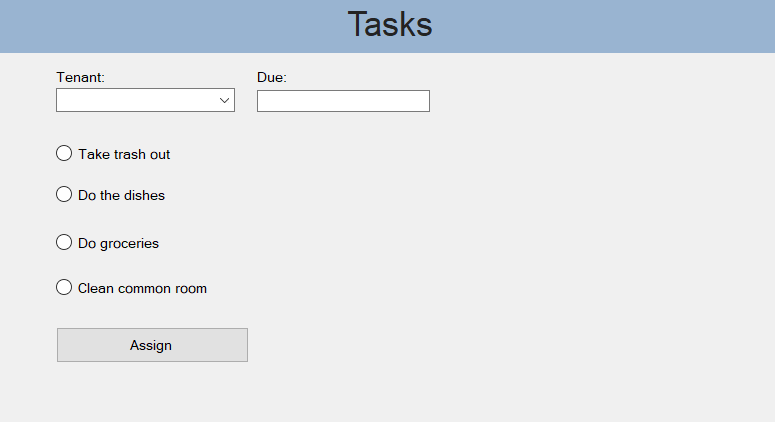


* Complaints: once here, the admin is presented with a combo box full of tenant names, a list box and a button. The list box is used after a tenant is selected. It shows all the complaints attributed to that certain tenant. Pressing the button facilitates that action:



* Announcements: this tab links up with the tab that the tenants can see. With the help of a list box the admin sees all the announcements that are available for the tenants to see. The admin can also update an already existing rule and can remove it as well:
* Tenants: since this is the admin view, the admin has to manage the students that are housed or will be. That’s why we added the “Tenants” tab. Here, the admin can select an already existing tenant and make it “Student of The Month”. It can also add a new tenant to the house by imputing the name of said tenant in the text box:

Rules: rules must be put in place by people in charge, that’s why we added the “Rules” tab. Here the admin can select from all the rules that have been put in place, can add a rule for the tenants to follow, can update an already existing one and they can also remove it. We wanted to keep it versatile and easy to manage for the admin so we decided to keep it similar to the “Announcements” tab:

* Tasks: this tab is assigned for the attribution of tasks to one or multiple tenants living in the house. The admin chooses from the combo box which tenant is needed. From here, 4 options are available. The tenant is tasked with either taking out the trash, washing the dishes, buying groceries or with cleaning the common room. The possibility to assign multiple tasks to one tenant is also possible. Not only that but a due date is also issued for each task:

COMPOSITION

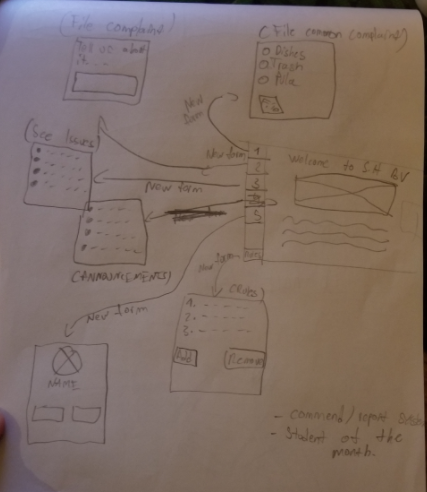
We are dealing with a house full of individual students. That means that we had to get creative and to make each interaction with the app unique.

We created different classes for each of the main objects of the app, the tenant being the most important. This class has in its composition the most important attributes that one tenant should hold, from the name and the password of the account to the list of tasks and complaints attributed to said tenant.

The admin also has a dedicated class but since there are only so little things that need to be stored for a specific admin, it’s not as intricate as the tenant class.

Announcements, rules, and tasks also have simple classes that are used mostly in composition with the admin and the tenant classes.

ROADMAP AND ISSUES

When we were assigned with the creation of the application, we decided to start with the design first. As we’d later find out, Visual Studio doesn’t allow for a more complicate designs when creating such user-friendly apps. So, we had to get creative.

After creating the first page we were not sure how we would arrange our buttons for the user to interact with. At first, our layout of the home page had its buttons on the side, as shown in our sketches from the early stages of development.

After some thought, we changed the button layout to be displayed on the top of the screen for better visibility, since the name of the buttons would have looked unprofessional placed on the side.

When that was done, we moved on to the design of the tabs. We decided to go for a more detailed look for the tenant side, since the app was going to be used mostly by the tenants rather than the admin who will only hop in from time to time.

In less than a week we had our design ready and we were already making plans about the features that we will soon be adding to the app.

Shortly after the general design for the tabs was done, we also added the aforementioned classes for the tenants and the admin. We made the tabs communicate with the home screen through the press of each of the buttons that redirects you to a designated tab (i.e., File a complaint, Announcements, etc.).

We were not yet sure how to save all the data of all the tenants that will be using the app. At first, we thought that we will have to get everything stored in the classes themselves but than we also thought we could also use files to store data such as the lists of tenants and all they’re passwords.

It was hard implementing something we barely had any knowledge about but with the help of other experimented programmers in our group, we managed to get it working nice and easy.

We than focused on each of the tabs, making sure that each of them was doing its job properly. The tab that we had some problems with was the “Profile” tab. We were not yet sure what we could add to make this tab stand out. We decided that it is best to leave it simple, with just a “log out” and a “change password” button.

The real problem came when we had to implement the changing of that password of an account. Since Visual Studio doesn’t allow you to read and to write a text file at the same time, so we had to create a new file that would copy everything that was written in the other file plus the extra info and delete the old one but not before renaming the new one it matches the previous one.

Passing user information from form to form was made possible by making the current user a static object so that its data is always available.

This matter presented more problems for us. Before we could figure this out, the app would crash once you would log in as an admin, log out, and log back in as a user. This happened because the “Current Tenant” would be set as nonexistent.

As a tenant you can vote and can also add money for various groceries. Here is where we used our new understanding of “static” variables. We managed to make the total sum of money necessary to always be displayed on the screen no matter the tenant that accesses the tab.

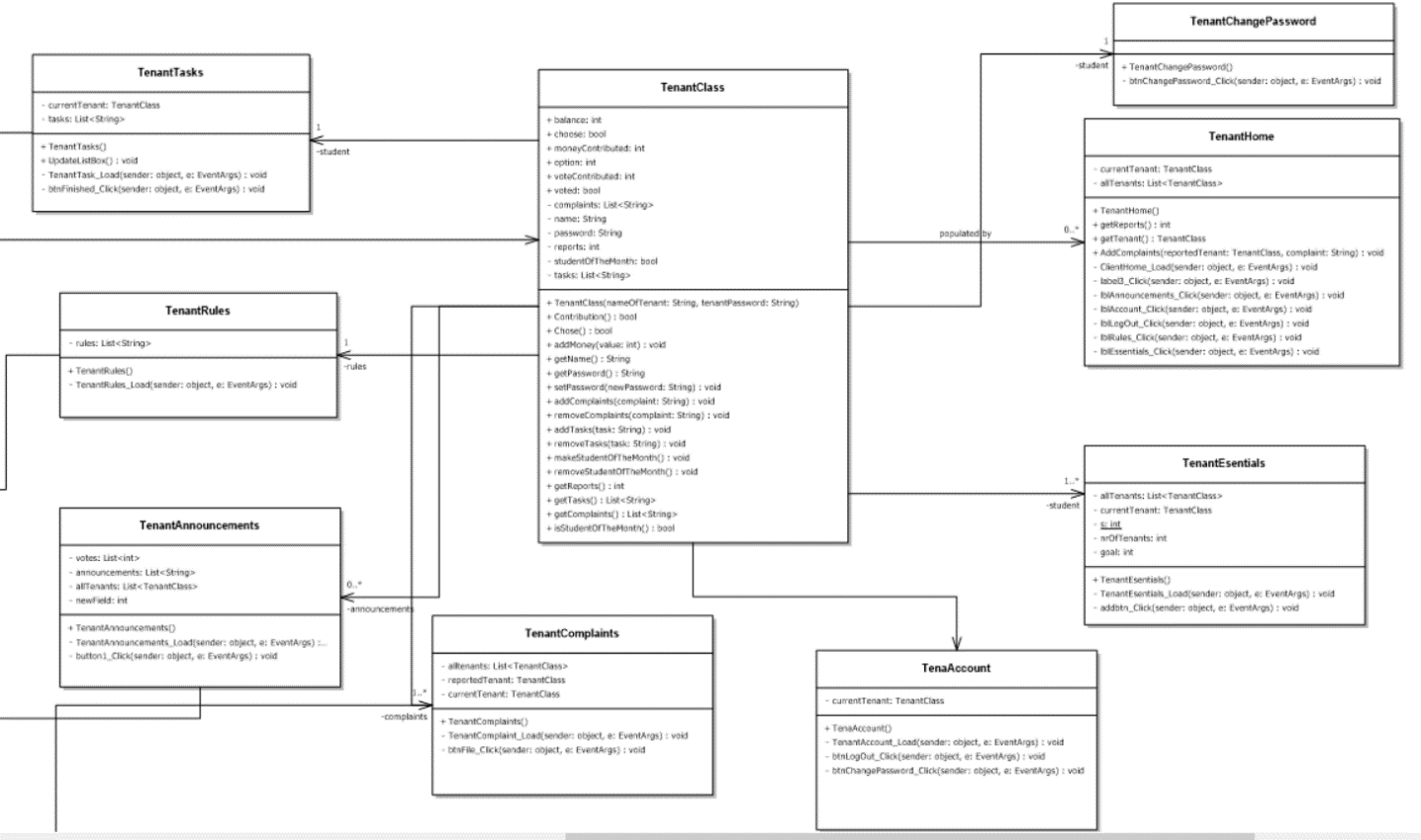
We took this system and we also implemented it on the voting system.

POSSIBLE IMPROVEMENTS

* We could see this app being pared up with a navigation app like “Google Maps” that could show all of the housings available by “Student Housing B.V.”. By selecting a certain house from the map, you can log in as a tenant of that house and can partake in daily activities.
* Pairing this app with a social media app would make it easier for the company to keep track of all of the tenants by storing their information, making it easier to log in and change accounts.

FEEDBACK

This project is, for most of us, our first demo of real-life work that we, as to-be Software Engineers, must be able to. This thought gave us the motivation to persevere even when faced with intimidating tasks that made us research and made us ask questions that we haven’t asked before, like: “How could we use files to store information?”, “Is there a way to store permanent information without databases?”, “What would be the best way to make our application be relevant when compared to modern day apps?” and more.

We hope to be able to dive deeper into such topics and we also hope that we will be given more opportunities like this, opportunities that allows us to be very creative, practical and productive.

