Advanced Project Report

3793672 Andrei Stefan

3781917 Georgi Manev

Contents

[Introduction 3](#_Toc30193924)

[Background 3](#_Toc30193925)

[Problem Statement 3](#_Toc30193926)

[Process & Results 3](#_Toc30193927)

[Proposed Solutions 3](#_Toc30193928)

[Work we did on the project 4](#_Toc30193929)

[Georgi’s part 4](#_Toc30193930)

[Andrei’s part 4](#_Toc30193931)

[Code implementation 4](#_Toc30193932)

[Admin 4](#_Toc30193933)

[User 4](#_Toc30193934)

[Conclusions and Recommendations 5](#_Toc30193935)

[Evaluation/Reflection 5](#_Toc30193936)

[Georgi 5](#_Toc30193937)

[Andrei 6](#_Toc30193938)

# Introduction

The topic of this document is the project for the advanced part of the orientation phase, semester 1.

In this document we’ll go over the steps we undertook in order to create the advanced project, but also how we did it, the problems we faced along the way and many more.

# Background

The projects consists of an student housing application, that is managed by an admin, and is used by the students. At the start of the project we didn’t have the required knowledge to complete it, and during the project we learned a lot of things in order to help us build the project. When considering the approach and the promised deliverables there where a lot of options presented. Implementing an application that only showcases functionality but can’t save its instance for the next use, Using a basic saving method through .txt files, or using a database system that could be hosted either locally or in the cloud. We chose to implement a local database, as we figured we would be able to do that in the allotted timeframe.

# Problem Statement

The client which we built the app for had multiple problems that had to be addressed by the app. The most major issues we were told about were: the appointed person not cleaning the shared facilities, the garbage is not being taken out on time by the appointed person, groceries are either not done or not paid for by tenants and last but not least unannounced parties and gatherings which disrupt the other tenants. Some not as pressing problems that the tenants are faced with include: not being able to issue an anonymous report of the disturbing individuals.

# Process & Results

## Proposed Solutions

In order to fix the problems that our client is facing, we came up with a few solutions. In order to help the client put an order to cleaning the shared facilities we decided an admin can implement a schedule inside the application, which the user can view, letting him know when he is due for cleaning. Taking out the garbage is fixed in a similar way, the admin being able to add persons to a schedule , which then the user sees when he is due for taking out the garbage. Both the garbage and the cleaning schedules are interactive, meaning that once the date of cleaning passes the tenant will not see it anymore. Parties can be announced by a tenant, which can be voted for by the other tenants. Once more than half (including half) the tenants voted yes on a party then it’s approved and will display an appropriate message. Once a party date passes it’s not going to be displayed anymore.

## Work we did on the project

## Georgi’s part

I connected the C# application to the SQL database, using tutorials I found on youtube. I also worked on the finishing touches of the application’s GUI. The bulk of the programming was done alongside my teammate.

## Andrei’s part

I did the first iteration of GUI and the following versions, working alongside my teammate on the bulk of the programming.

## Code implementation

We implemented an Access database with multiple tables: Users, Schedule, Party, Reports and Groceries. Inside the C# applications we have 2 windows, one that we log in, and one with the actual GUI.

Inside the first window there’s 2 textboxes with in which you input your Username and Password, and a Login button. When you press the button, the application checks the inputted data against the database, and if it’s correct it opens up the second form, if not it let’s you know the data you inputted is not correct.

The second window is composed by a tab control with multiple tab pages each representing some functionality of the application. Overall it can be seen that it has 2 modes of operation: either the user logged in with an admin account, or the user used a normal account.

### Admin

In the first case, the application will only display the admin tab in which the admin can manage the databases through. It has 3 controls. One which can be used to add a new user to the database by providing the Room, Password, First Name and Last Name, the username is a combination of the First Name and Room number, thus making it unique. It can also edit a user, by providing the room number and changing the other details or just delete a filed based on room number. The second control manages the schedule of the tenants, by allowing the admin to assign an user to either garbage or cleaning on a specific date. The third control let’s the admin view the user reports by selecting the report he wants to view from a combo box and clicking the view button. He can also delete the report once he finishes dealing with it.

### User

The user is greeted with the schedule of his garbage and cleaning, and the party announcements, in 3 separate listboxes. In the same tab we have a button that changes the date of the application, in order to showcase the functionality of our application, which automatically hides parties and cleaning/garbage dates that are in the past.

The groceries tab lets the user pay for the common groceries and add groceries to the common groceries. Each user has a balance that the money is subtracted from and that can be topped up.   
When you pay for a grocery you can’t pay for it twice, and once everyone pays that grocery is removed from the list. If you buy a grocery, once everyone pays for it you get the money that you spent of it back. We do this feature by having the first person who pays for the grocery be the buyer, when he adds the grocery to the list.

The party announcement tab is made to let the user announce a party, but also vote for the current parties. The user can select the party from a combobox and then clicks either a yes or no button or a view details of the party. Each user can only vote for a party once and the person who created the party is automatically counted as a yes vote. In order to add a party you must provide a name, a description and select a date from the date viewer.

Reporting a problem is anonymous and is done by providing a name and a description for the problem. The application automatically adds the date to the database, which the admin can see when reviewing the problems.

The bathroom check tab allows the user to see if the common bathroom is in use with the use of an Arduino, which checks the light level in the bathroom, and if it detects that the light is turned on then sends the message that the bathroom is occupied at the moment. This feature saves users time, as they don’t have to check if the bathroom is currently in use.

In my account tab the user can manage his balance by providing the amount he wants to add to his account in a textbox and clicking button and change his password.

# Conclusions and Recommendations

I think that the project managed to hit its requirements. In order to maintain our software for the future I think that a better database system should be implemented and a in cloud database, in order to make it easier to access and use. I think that more features can be added such as a chat app, being able to pay for other person’s groceries and to specify your time away, so the admin knows how to make the schedule. The schedule could also be automated.

We managed to hit the must haves of our project starter which were a login system, a cleaning and garbage schedule and a party announcement system. Should have: has been implemented in the form of a database. Could have: is realized by the bathroom availability with the use of an Arduino and the balance which you can use to pay for groceries.

# Evaluation/Reflection

## Georgi

I think that the project went well, as we managed to deliver it on time, while also delivering what we promised, learning new skills on the way. In the next project I would continue working on the features at the same time with my teammates, as 2 pair of eyes catch mistakes more easily, but I would also start working on it earlier as some challenging parts took a lot longer than expected, making us short for time.

I’m proud of the fact that we managed to use a database, and we didn’t have to resort to some other kind of data saving.

## Andrei

This project was a blast! I believe that it was challenging and that it required a lot of work but it felt great to work on it. In the next project I would use Git, as it makes it easier to share code around a bigger group and I think that it makes the work easier. I would not leave the easiest parts for last, as doing them first means I don’t spend idle time with no progress done on the project.

I’m proud because this project took a while to make but in the end we managed to finish it.