

**Michael Bahchevanov – Team leader (3806596)**

Aleksandar Todorov – Team member (3787842)

Kristiyan Strahilov – Team member (3807487)

Michael Groenewegen van der Weijden – Team member (3842401)

Contents

[Introduction: 3](#_Toc30153246)

[Background: 3](#_Toc30153247)

[Problem Statement: 4](#_Toc30153248)

[Process & Results: 4](#_Toc30153249)

[Evaluation/Reflection: 12](#_Toc30153250)

[Recommendation: 12](#_Toc30153251)

[Conclusion: 13](#_Toc30153252)

[References: 14](#_Toc30153253)

# Introduction:

The aim of the project is to create a software application easing the communication between landlord and tenants. The application’s goal is to enable communication and support both parties in their mission to maintain a household.

This report presents a solution on how to solve the problems with the house. The communication is solved with the help of a chatroom – for the tenants and complaint forms sent via email to the landlord. Moreover, the landlord can send announcements, preparing the tenants for an upcoming maintenance. In terms of chore separation and motivation, a calendar with the chores for the week is created. Each chore gives a certain amount of points, which in the end can be claimed to reap rewards.

# Background:

Our team examined the problems within the household and addressed the potentially significant problems, resulting in the creation of a software application solving them. The proposed solution consists of six main points we have focused on – separate accounts for the landlord and the tenants, a calendar with chores and a separate interface for the chores taken, a chat room for the tenants, a point system, a complaint system via email and an announcement system. All these points are to smoothen the communication and allow an easier track of duties per week. The proposed accounts are to enable some hierarchy in this small ecosystem. The calendar is to help with the weekly organization. The chat is there to enable communication between the tenants. The point system is there to serve as a motivation to do the appointed chores with the possibility of a reward. The complaint system is a last resort measure, since it allows the landlord to interfere with inner operation of the house. Finally, the announcement system is to serve as a way of notifying the tenants of an upcoming maintenance, rent collection, etc.

# Problem Statement:

In every student housing the tenants are expected to maintain a clean and healthy living environment, through keeping personal items away from the common areas and cleaning them whenever they are used. However, in the problem described in the project document that is not the case as the kitchen is left untidy and the toilet is left dirty for an excessive amount of time. The behavior shown by the tenants is closely connected to the lack of communication and motivation. These problems will be mitigated with the help of our proposed solution. This application will enable communication and encourage the tenants to follow pre-set guidelines. Moreover, it allows the landlord to monitor the household. It would be beneficial for the agency to implement this application as it will prevent health and economic risks in the future.

# Process & Results:

Before initializing in writing code, we held meeting in which we mapped out the schedule for deployment of all components of the solution [Figure 1]. We have decided to split the work so we can be as efficient as possible, while still having the last week to go through every component more in-depth. The separation went as follows – Michael Bahchevanov took the responsibility to archetype a database and allow a fluent information insertion and extraction. Aleksandar Todorov handled the task of creating a graphic user interface (GUI) and helped create maintainable and reusable code regarding all the GUI elements. Kristiyan Strahilov tackled the task of creating an accessible chatroom that stores everything inside the database. Michael Groenewegen van der Weijden did the work on smoothening the logic, implementation of algorithms and creation of classes.



Figure

We started the development by deciding on a product logo and naming [Figure 2], afterwards we came up with our GUI [Figure 3], which we kept until deployment. Later, we discussed the implementation of several features, but ultimately keeping a calendar [Figure 4], a chat room [Figure 5], a reward system[Figure 3] and a complaint system [Figure 6]. Moreover, we decided on having an account system [Figure 7], meaning that the tenants and landlord would have their own separate account with different features. In order to be able to implement all these features we went with a solution including a database, hosted on the HERA server[[13]](#_References:), and a database management system – Microsoft SQL Server Management Studio (**SSMS****)** [[1]](#_References:)**.**

****

Figure

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a cell phone

Description automatically generated

Figure

A screenshot of a cell phone

Description automatically generated

Figure

A screenshot of a computer

Description automatically generated

Figure

A close up of a screen

Description automatically generated

Figure

For the design of the logo and initial GUI, we used Adobe Photoshop [[2]](#_References:). Within Visual Studio and with the help of the .NET framework[[12]](#_References:), we created a Windows Form Application (WFA) [[3]](#_References:) and the first implementation was made by Michael Groenewegen van der Weijden and later changes were made by Alexander Todorov, keeping the core design the same throughout the whole process which consists of a calendar tab [Figure 4], a chatroom tab [Figure 5], a scoreboard tab [Figure 8], a complaints tab [Figure 6] and an announcement pop-up [Figure 9]. For the landlord we went with an overview tab of the scoreboard [Figure 10], a tab where he could alter the accounts [Figure 11] and an announcement section [Figure 12] where he could real-time message to the tenants.

A screenshot of a computer

Description automatically generated

Figure

A screenshot of a cell phone

Description automatically generated

Figure

A screenshot of a cell phone

Description automatically generated

Figure 10

A screenshot of a computer

Description automatically generated

Figure 11

A picture containing screenshot

Description automatically generated

Figure 12

Once the GUI was completed we proceeded to setup the database on the Hera server of Fontys, for which in order to connect, we had to use the Cisco any connect VPN (Virtual Private Network) [[4]](#_References:). Michael Bahchevanov and Alexander Todorov had the duty to set up the database connection, create stored procedures, using SSMS and a class with methods, inside Visual Studio[[3]](#_References:) for accessing the information within the database. With the help of Dapper[[8]](#_References:), Michael Bahchevanov linked the database with the WFA[[7]](#_References:) and he created stored procedures inside SSMS with the help of SQL[[10]](#_References:). Next Michael Groenewegen van der Weijden proceeded to test out the methods in small chunks to ensure proper work and report for improvement if needed.

Parallel to that, Kristiyan Strahilov was testing a chatroom, using VPN with Hamachi[[9]](#_References:) with the help of a UDP Connection[[5]](#_References:). Unfortunately, this did not go as intended so he resorted to using the database to implement the functionality of the chatroom.

Continuing forward, the announcement system [Figure 9] was implemented by Michael Groenewegen van der Weijden. The duo of Aleksandar Todorov and Michael Groenewegen van der Weijden implemented the scoreboard [Figure 8] and the complaint system [Figure 9], which utilizes Gmail[[11]](#_References:) as a way of communication from the application. Following that, the initial chatroom [Figure 5] was implemented by Kristiyan Strahilov and continued to work quality of life improvements. Lastly, the calendar system [Figure 4] was implemented by Alexander Todorov. After developing all the primary functionalities, we began the long process of code refactoring, with the help of Clean Code[[14]](#_References:), adding minor functionalities and overall easy to follow code.

In the end, the application had all the initial features, as well as additional features like the password retrieval, account manipulation, etc. The team managed to meet all the deadlines, as said, and deploy a functioning application.

# Evaluation/Reflection:

To look back as a team/group at the project we are surprised how fast this whole process went from just having the idea to deploying a project of such scale amazes us. Moreover, we are all satisfied with how things progressed. The team had outstanding communication and a steady implementation process. A key aspect that contributes greatly to this project’s success is that we worked as often as possible together. The only thing that could be considered a drawback was our unrealistic expectation at the start of the project, this impaired the process and for the next project we could be more realistic. It did not affect our project but made things a little more challenging for us.

# Recommendation:

We faced some problems with the platform – Windows Forms Application (WFA) is a simple for beginner-friendly way of creating coherent applications but it is limited when it comes to design options. For future projects, the team would use a more open and flexible platform like Windows Presentation Foundation (WPF)[[6]](#_References:). Moreover, there was a period of two weeks of holidays, where we could have done more work. That would have helped meeting the deadlines in a better pace.

# Conclusion:

Overall, by taking everything into an account, that has happened these last four weeks, we are quite happy on how things have turned out – no major communication issues, no fights between us and in general a very professional and productive environment. Moreover, the application turned out outstanding, with all the functionalities, In the end, the lessons we have learned are not only about how to code or have a good insight and algorithmic thinking, or how to set up a database and implement the necessary procedures, but life and personal skills were also learned, communication skills were improved and friendships were formed.

# References:

No. 1: SQL Server Management Studio (SSMS), Link: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>, Release date: 09/11/2019.

No. 2: Adobe Photoshop, Link: <https://www.adobe.com/nl/products/photoshop.html?promoid=RL89NFBP&mv=other>, Release: Unknown.

No. 3: Visual Studio, Link: <https://visualstudio.microsoft.com/vs/>, Release: 2019.

No. 4; Cisco any connect vpn, Link: <https://www.cisco.com/c/dam/en/us/products/collateral/security/anyconnect-secure-mobility-client/at_a_glance_c45-578609.pdf>, Release: 06/17/2017.

No. 5: UDP Connection, Link: <https://en.wikipedia.org/wiki/User_Datagram_Protocold>, Release: 02/12/2019.

No. 6: Windows Presentation Foundation, Link: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/> Release: 01/25/2018

No. 7: Database connection with Dapper, Link: <https://www.youtube.com/watch?v=Et2khGnrIqc&t>

Release: 02/28/2017

No. 8: How to use Dapper, Link: <https://www.infoworld.com/article/3025784/how-to-use-the-dapper-orm-in-c.html> Release: 07/22/2019

No. 9: Hamachi, Link: <https://www.vpn.net/> Release: 12/31/2019

No. 10: SQL, Link: <http://trafodion.apache.org/docs/sql_reference/index.html> Release: 05/01/2017

No. 11: What is Gmail, Link: <https://whatis.techtarget.com/definition/Gmail> Release: 09/2005

No. 12: The .NET framework, Link: <https://docs.microsoft.com/en-us/dotnet/api/?view=netframework-4.8> Release: UNKOWN

No. 13: Server – host HERA, Link: UNKOWN Release: UNKNOWN

No. 14: Clean code – Robert Martin-Lei Han - Publishing House of Electronics Industry - 2012