|  |
| --- |
|  |
| TaskBerry |
| The smart scheduler |
|  |
|  |
|  |

|  |
| --- |
|  |

Contents

[2 Overview 3](#_Toc26560941)

[2.1 Project Objectives 3](#_Toc26560942)

[2.2 Project Constraints 4](#_Toc26560943)

[2.3 Project Risks 4](#_Toc26560944)

[3 Proposed Solution 4](#_Toc26560945)

[3.1 Architecture 4](#_Toc26560946)

[3.2 Development 7](#_Toc26560947)

[3.3 Testing 7](#_Toc26560948)

[4 Project Resources 7](#_Toc26560949)

[4.1 Roles and Responsibilities 7](#_Toc26560950)

[4.2 Project Materials 7](#_Toc26560951)

[5 Project Approach 7](#_Toc26560952)

[5.1 Communication Management 7](#_Toc26560953)

[5.2 Documentation 8](#_Toc26560954)

[6 Schedule 8](#_Toc26560955)

# Overview

The aim of our company is to create a smart scheduler for realizing and helping fellow students maintain a more organized and clean-living environment. In order to aid the tenants with our service, we have come up with a simple but effective approach.

* Task distributor
* 2 application – tenant and landlord
* Announcement panel
* Chat room
* Pointing system
* Complaint emails
* Database

**Need for project**

We are creating this application because of the need of organizing the chaos that exists in the usual student house, as well as the lack of communication and inability to interfere from the side of the landlord. The stakeholders in the project are the landlord and Student Housing BV.

**Challenges**

The challenges may include the following:

* Lack of knowledge
* Components not delivered on time from some parties
* Time management and efficiency of work

**Opportunities**

By implementing the project, we will be able to achieve the following things:

* Ease of communication between parties
* More motivating environment
* Suffice the expected workload
* External control over the property

## Project Objectives

This section should specifically list project objectives. These are the criteria which will be used to measure project success. For example:

* Complete application implementation by the end of 14/01/2020
* Provide a fully functioning application for both the tenants and the landlord
* Provide a way to communicate between parties
* Provide motivation for the users
* Implementation of a database
* Implementation of a seamless connection between 5 users
* Simple and interactive user interface

## Project Constraints

The primary constraint we are going to encounter, is the limit in time. Also, we can face sudden health problems or unexpected events regarding the team members.

## Project Risks

The main project risk is not meeting the requirements in the timeframe.

**Risk Probability**

The chances of this happening are moderately low. As we are a motivated and dedicated team, we strive for the best results, which we are reaping until this point.

**Risk Impact**

If we do not meet the timeframe, we will not be able to deliver the project, meaning we are going to fail the assigned work.

**Risk Mitigation**

We could mitigate the risk by finishing deliverables on time and going the extra mile to provide the expected results. Also planning and scheduling consistent sessions is another way of reducing the chance of delivering an unsatisfactory result.

**Contingency Plan**

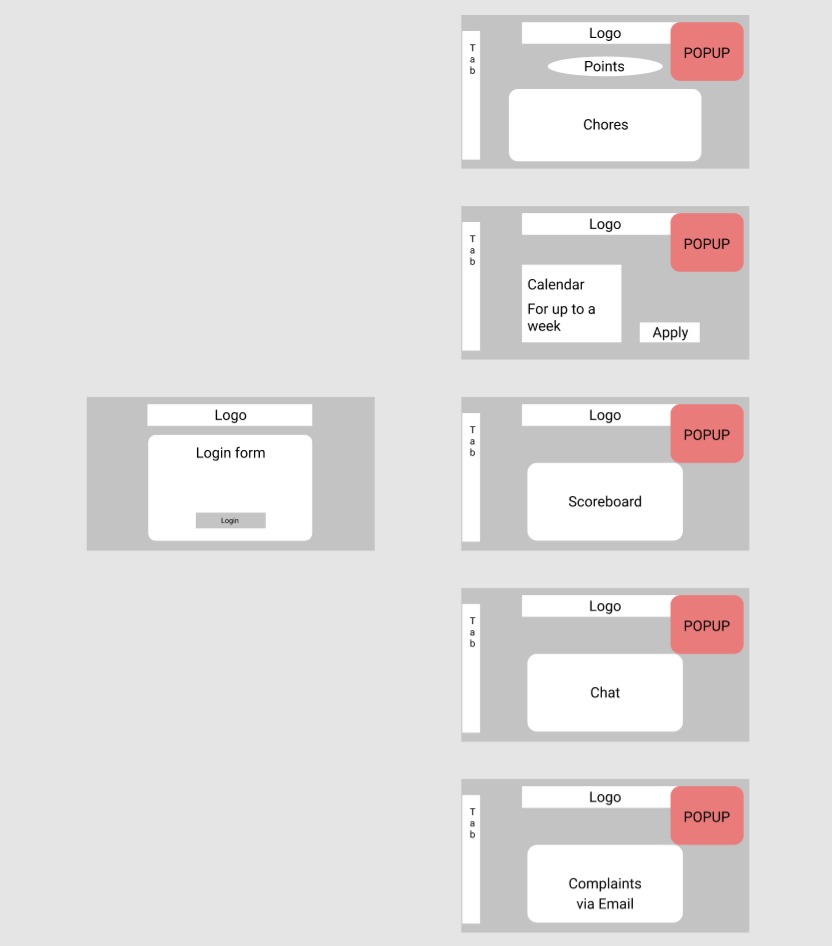
Our contingency plan is to create a simple more lightweight version which has only the basic functionality (a single application with a central screen for everyone to use).

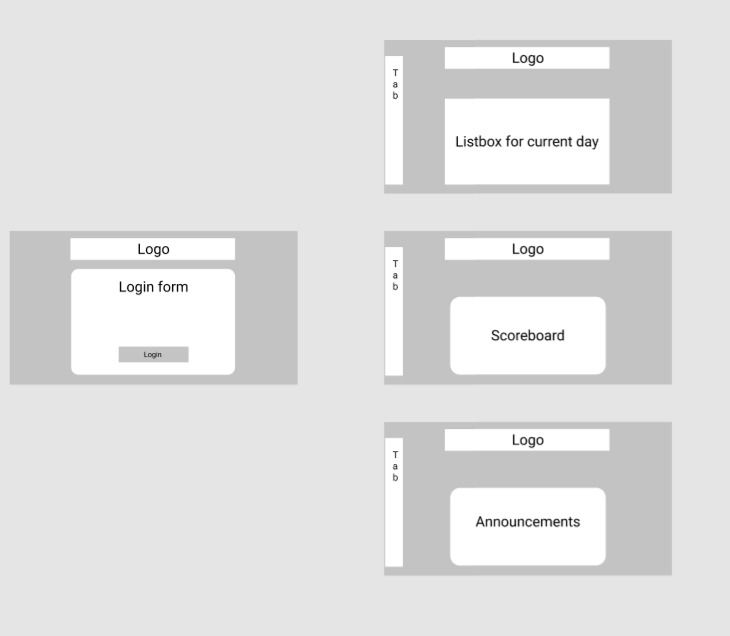
# Proposed Solution

We are going to develop 2 applications – 1 for the landlord and 1 for the tenants. This application is going to allow smooth communication between the tenants via a chat room. We are also going to develop a job/chore distributing system. Tenants are also able to file complaints if chores are not completed. A point system is also going to be implemented, motivating the tenants into doing the chores. Additionally, the landlord has a full overview of the points and behavior of the tenants, with the use of a generalized scoreboard. On top of that, the landlord can deduce points from the tenants whenever he receives a complaint towards them. We are going to store the information (accounts, chores, announcements) inside a database.

## Architecture

Wireframe of the first iteration of the product:





**Functional Specifications**

Let us begin with the communication between applications. For that we are going to use a software called ‘Hamachi’, used for creating a VPN in a close environment (5 machines). We are going to send the data from the chat with a UDP (user datagram protocol) for every tenant to see. Secondly, we are going to implement a database with MySQL, hosted on a local machine (a server), to store the information of the accounts, chores and announcements. We are also going to link the database to the WFA (Windows Forms Application) for authentication purposes and have the application up-to date. We will handle the communication using a class inside the WFA, with C# (an object-oriented programming language), with the appropriate methods.

**Technical Specifications**

Technical specifications include the following:

* C# programming language
* MySQL relational database management system
* Hamachi (VPN)

## Development

We are separating the development of the application into 3 phases: Initial, Interim and Final. The ‘Initial’ phase includes brainstorming and gathering ideas, creating a basic interface and layout of behavior of the application. The ‘Interim’ phase includes the basic functionality of the developing product and more work on the interface, as well as incorporating the database and communication between machines. Lastly, the ‘Final’ phase is about optimizing and improving the interim version of the product, as well as bug-fixing and fool-proofing the application.

## Testing

We are going to give constant iteration of the product to a selected group of users. In order to fix all the possible bugs with each iteration, we are going to improve on the feedback we have been given. Additionally, we are going to use the integrated debugging tool inside Visual Studio to debug our iterations.

# Project Resources

## Roles and Responsibilities

The positions taken in the team are as follows:

* Software Architect – Michael Groenewegen van der Weijden
* Network Engineer – Kristiyan Strahilov
* Design and Interface Developer – Aleksandar Todorov
* Database Administrator – Michael Bahchevanov

## Project Materials

We use the following services/hardware/equipment:

* Laptop
* Server
* Visual Studio
* MySQL
* Hamachi

# Project Approach

## Communication Management

We are using a few ways to maintain the workflow of the application. These include software-based methods, namely:

* Discord
* WhatsApp
* GIT (with the GitKraken GUI)

## Documentation

Testing documentation details will be delivered during both the course of the project and at project end. We are going to provide it via a document or a video.

# Schedule

The project schedule follows these guidelines:

Initial phase: Completion – 08/12/2019

* The two user interfaces (interim version) are going to be our main priority
* ‘Dummy code’ and simple functionality for overviewing purposes (classes, lists and simple event handling)

Interim phase: Completion – 12/12/2019

* Implementation of database and connection
* Implementation of chat room and connection
* Interface improvement
* Implementation of proper code

Final phase: Completion – 14/01/2020

* Optimization of code
* Final touches on interface
* Preparations for final presentation
* Final document ready