

# Purpose of this document

This document helps entail the various jobs and aspects that the team will face during development, this document will also keep all members on a set path to a completed project/program. This program could be considered a '“timeline” of the events that will occur during the next 3 months (Development), of course there will be unforeseen bugs/issues/occurrences that will arise in future, but this document will keep us on the straight and narrow giving this report a sense of meaning and direction.

# Client details

The client is an employee working at an educational institution, Nicola Dingel. Nicola needs to communicate with other staff in different rooms of the institution. This will enable Nicola to have a far more connected workplace and a general means for communication which is both useful and is perfectly appropriate.

# Problem outline

The client needs a solution that will allow him to communicate with other colleagues over a network, and has asked for the team to develop a chat program that operates with a theoretically unlimited number of users, where each user can send and receive messages from one another. Provided there is a medium in which they can communicate through (The server). The client aspect of the solution will entail the connected users, the time in which they posted a message and what they had said. This is essentially a bare bones chat program with a few bells and whistles.

# Requirements of software solution

## Input requirements

The user needs to input a Username, and an IP address the program can connect to, the user may then input messages to send to other users.

## Output requirements

The Program needs to output on a server screen, a message that lets the host know when the server has started, a user has joined, sent a message, or left. The program also needs to output to all users when a user inputs a message, as well as when a user has connected and disconnected.

## Performance requirements

In order for the chat program to function properly, the user must have a windows computer with low-end specs. The user must have basic computer peripherals such as a keyboard, mouse, and monitor. Finally, the user must have access to a network to connect with other users with.

## Social and ethical requirements

The program must protect the data being sent from client to client, while blocking racist, sexist, and generally offensive language. The program must recommend not to share personal or sensitive information.

# Development Team members

## List members

Eilia - Software Developer

William - Software Developer

## Member skills and knowledge

Eilia:

- basic knowledge of TCP networking.

- Has advanced knowledge on all basic functions of visual studio and C#.

- knowledge of XAML for creating User Interface.

- knowledge of debugging.

Will :

* Use of XAML for creating a User Interface.
* Has advanced knowledge on all basic functions of visual studio and C#
* Has a keen eye for detail when reading criteria
* Knowledge of debugging.

## Member responsibilities

Eilia : 60/40 Server Vs Client

Eilia will be focusing on the Networking part of the project and ensuring that clients can effectively communicate with one another. While also editing some aspects of the client to ensure that they can work with the server.

Will : 40/60 Server Vs Client

William will be focusing more so on the functionality of the client and will be ensuring that the client is both pleasant to operate and has some features to spice it up. However will also be working on the server of the solution as the secondary objective to gain knowledge and experience in that aspect as it will be crucial in future.

# Potential development approaches

{Problems can be solved in lots of different ways. So what software development approach will be used? What tools will be used to assist in development, what platform(s) could this exist on, etc}

## Approach #1

The first approach we could use to complete the solution is a Structured Approach. We may use Visual studio as a platform to develop the solution. The development team will use GitHub to keep track of the versions of the solution and Google Docs to create documents during the development of the software. The team may also use Draw.io and fluidui.com for creating storyboards and screen designs.

## Approach #2

The second option we could use as the development team, would be a combined approach of Structured and the Agile Approaches. The team would still use the same software to assist in the development of the solution as listed in the first Approach.

## Conclusion

In conclusion we choose the combined approach due to a handful of reasons, one being that an approach that consists of both Structured and Agile suited our needs best. For example we don’t believe that we will be able to identify all aspects of the structured approach, that being said we believe we will be able to identify most but not all. This is where the Agile approach arises and this takes care of any issues / unforeseen “roadblocks” that will be in our path.