

Report



WeeChat

Assignment 1

Human Computer Interfaces (ICTE3002)

Curtin University

Prepared by:

Aaron Gangemi	(19447337)
Aaron Giles	(19487210)
Kevin Le	(19472960)
Terence Marcelo	(19163785)
Ryan Martin	(19514231)

Contents

Report	1
Contents	2
Introduction	5
Background	5
Program Description	5
Design Purpose	6
Team Members	7
Contribution Table	7
Design Process	8
User Involvement	8
Requirement Elicitation	9
Target Demographic	9
Survey Result and Analysis	10
Personas	12
Nathan Hamilton	12
Elissa Mill	13
Blake Grant	13
Curtis Van Dyke	14
Competitor Analysis	15
Kiwi IRC (Web Client)	15
HexChat	15
mIRC	16
Requirement Specification	17
Usability Goals	17
User Experience Goals	18
Functional Requirements	18
Non-Functional Requirements	19
User Stories	20
Use Cases	20
Hierarchical Task Analysis	24
Login	24
Register	25
Send Message	25
Add Server	26
Customization	26
Plugin/ Script Manager	27
Modify Account Details	27
Delete Account	28
Search for User	28
Edit user account information	29
Prototypes	30
Aaron Gangemi (19447337)	30
Low Fidelity	30

High Fidelity	32
Aaron Giles (19487210)	37
Low Fidelity	37
High Fidelity	41
Kevin Le (19472960)	47
Low Fidelity	47
High Fidelity	53
Terence Marcelo (19163785)	61
Low Fidelity	61
High Fidelity	64
Ryan Martin (19514231)	68
Low Fidelity	68
High Fidelity	68
Implementations	73
Aaron Gangemi	73
Implementation Discussion	73
User Involvement	87
Test Cases	89
Aaron Giles	97
Implementation Discussion	97
User Involvement	107
Test Cases	109
Kevin Le	112
Implementation Discussion	112
User Involvement	122
Test Cases	126
Terence Marcelo	132
Implementation Discussion	132
User Involvement	137
Test Cases	138
Ryan Martin	141
Implementation Discussion	141
User Involvement	147
Test Cases	148
Evaluation	154
Meetings	154
Risk Management	154
Milestones	155
Problems Encountered	156
Conclusion	157
References	157
Appendix	158
Initial User Survey Responses	158
Meeting Minutes	163

HCI Meeting 1	163
Agenda	163
New Business	163
Actions	163
Agenda for Next Meeting	164
Other Notes	164
HCI Meeting 2	164
Agenda	164
New Business	164
Actions	164
Agenda for Next Meeting	165
Other Notes	165
HCI Meeting 3	165
Agenda	165
New Business	165
Actions	166
Agenda for Next Meeting	166
Other Notes	166
Client Meeting	166
Agenda	167
New Business	167
Actions	167
Agenda for Next Meeting	167
Other Notes	167

Introduction

Contributor(s): Aaron Giles

WeeChat is an open source command line IRC client which allows communication between people who are connected to the same server and channel. WeeChat is a powerful software with many features and is nearly fully customizable. However, due to its command line design structure, it can be confusing and difficult to operate for first time users or people with little command line know-how. This report outlines a redesign of the interface for WeeChat, with the intention of improving the usability of the software whilst also keeping all the powerful features and customization that the original interface contained.

Background

Program Description

Contributor(s): Aaron Giles

WeeChat is a very basic command line only messaging platform, designed in 2003 to be fast and lightweight. Unlike many other messaging platforms it gives the user an option to create and run custom plugins, which makes it very appealing to technical users. The platform has many other features that are standard to most messaging services, such as messaging in servers, channels within servers, adding servers, custom preferences and private messaging. To perform these tasks various commands are needed to be typed and executed. Sent messages all contain timestamps and usernames. The image sending service is quite limited, the current method of doing so is to upload an 3rd party hosting service and to send the link. We wanted to stick to the program's core elements and decided to not change how it is done to avoid upsetting existing users.

The original look of Weechat can be seen below, which incorporates a very basic interface and not much “user friendliness” and direction on how to use the application.

```
10:55:24 @FlashCod+ phlux: you mean the page with screenshots?  
10:55:47     phlux FlashCode: Yeah. I'm wondering if you're accep  
just keep it the same.  
10:56:14 @FlashCod+ yes new screenshots are welcome, but.. only if  
10:56:24 @FlashCod+ if it's very similar to other screenshots, I th  
10:56:40 @FlashCod+ I try to not have too many similar screenshots  
10:56:47     phlux I don't think many people have the buffer bar :  
10:56:49     Bio m  
10:56:52     Bio same here  
10:56:57     phlux Let me fullscreen it and link you..  
  
10:57:42     Bio http://screencloud.net/v/coEI thats my setup  
10:57:42           http://[REDACTED]:46227/bC8  
10:57:44     Bio its kinda nice  
10:57:51 @FlashCod+ maybe I should do a screenshot with window spl  
10:57:53     phlux Actually, fullscreening takes away from what I  
[10:58] [20] [irc:Freenode] 12:#weechat<+nt>(391)*  
[phlux] »  
  
1.Ewnix 3.#games 5.#Alaba+ 7.#servi+ 9.#ewnix 11.#arpne+ 12.#weech+ 12.#fbsdc  
2.#ewnix 4.#dc 6.#roche 8.#SEC 10.#fbsdc 12.#weech+ 12.#fbsdc
```

(Sourced from <https://weechat.org/>)

If a user is unsure on how to use a certain feature the only way is to go onto the WeeChat website and to navigate to the help page (https://weechat.org/files/doc/stable/weechat_quickstart.en.html),

which simply states how to execute the command needed. For example, to simply start the program the following “\$ weechat” needs to be executed in the terminal. To set up the custom preferences command needs to be structured in this format of

```
/set config.section.option value
```

which seems like a bit of a headache for anyone looking as they have to stop and think to understand what they are actually changing. This is why this application is in desperate need for a graphical user interface to avoid these unnecessary headaches for users.

The scripts feature provides a range of new custom features for users. To install first the correct script must be found and researched (<https://weechat.org/scripts/>) and once the correct script is found it can then be installed. The scripts can range from all sorts of functionality, some as simple as autocomplete when typing “@” to autocomplete a user’s name, or to displaying notifications on screen.

Design Purpose

Contributor(s): Aaron Giles

The idea for picking to design a graphical interface for WeeChat was to make it appeal to other demographics, people without the technical background that most of the existing users already possess. The powerful plugins features makes this not just a basic messenger app, but something much more. However, just due to the fact that everything being in strictly command line format, that immediately deterred many non-technical users from starting or even trying the application. We attempt to bridge this gap with the use of an easy and interactive user interface that makes it easy to use all these features. Providing the user with consistent feedback and “hold their hand” through the scary unfriendly command line.

Existing users can also benefit with the new more efficient way of doing certain tasks. For example, the ease of downloading and installing a new plugin, can be researched, read about, and can all be installed in two clicks. The ease of changing servers, and switching channels, can all be done within a few clicks, instead of typing out a few commands. In addition, the visual eye candy of the new fancy graphical interface might be appreciated by some users (obviously while having the option to stick with the command line).

Team Members

Contributor(s): Everyone

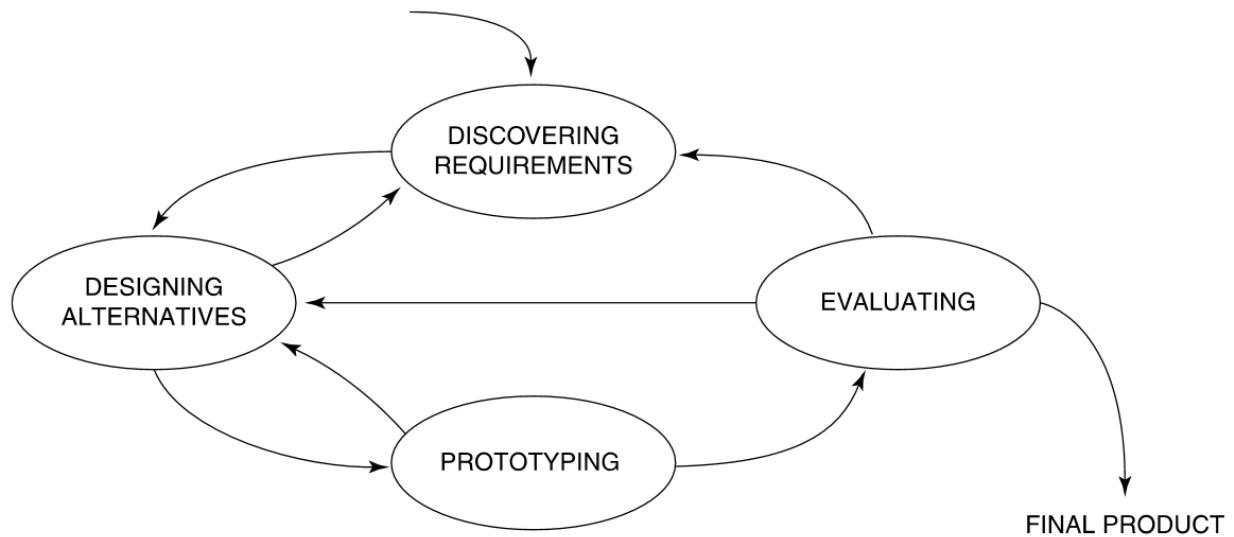
Name	ID	Role
Aaron Gangemi	19447337	Project Manager
Aaron Giles	19487210	Usability Engineer
Kevin Le	19472960	Graphic Designer
Terence Marcelo	19163785	Data Analyst
Ryan Martin	19514231	Document Controller

Contribution Table

Section	Assigned to
Background	Aaron Giles
Design Process	Terence Marcelo
Requirements Elicitation	Aaron Gangemi, Kevin Le
Requirements Specification	Aaron Gangemi, Kevin Le
HTA	Aaron Gangemi, Kevin Le
Prototypes	Everyone
Implementations	Everyone
Evaluation	Ryan Martin
References	Everyone
Appendix	Aaron Gangemi, Kevin Le

Design Process

Contributor(s): Terence Marcelo



Our team has employed the Simple Interaction Design Lifecycle Model to guide ourselves into developing a user-centric design. This model allows us to continuously improve our design throughout the design process. We continuously discover new requirements and better alternatives after evaluating and creating various prototypes therefore improving the user experience created by our design.

We began using this model by first discovering user requirements by considering our target demographic and conducting surveys and questionnaires for them. We then began our alternative designs while continually considering new requirements that we discovered. We created various prototypes for evaluation, repeating this cycle until we achieved a final design that we were confident would achieve all the requirements and expectations that we discovered throughout our design process.

Our evaluation involved a meeting with the client where we showcased our early designs for feedback and direction as well as getting opinions from potential users. For prototyping we used low-fidelity designs with wireframe interfaces which we used for the initial feedback before moving onto our high-fidelity designs and seeking further feedback. We then made our final design prototypes with Netbeans and JavaFX.

User Involvement

Contributor(s): Terence Marcelo

We began our design process by exploring our target demographic and how our design can meet their requirements. We considered what their expectations would be and conducted surveys and questionnaires to better understand them. The results of this were used to create our first alternative design which we presented to our client and potential users for evaluation, using the feedback to get a clearer picture of what the user expects from the final design.

This process was repeated multiple times to ensure we were confident that all possible requirements were explored. We decided which features were necessary and which ones would not improve or worsen the user experience.

A simple and easy-to-use interface was a huge consideration for our design. Some features we discovered through our requirements elicitation from surveying potential users but were less popular such as a news feed feature was decided against as it would add clutter to the design and worsen the user experience.

We also attempted to scope the degree of our potential user's ability to navigate our interface design by seeking out feedback regarding its ease of use from a group of people with varied ability of using technology. This information was used as a guide on how easy-to-use the interface should be without sacrificing functionality. We maximised our design's ease of use as much as we could while still satisfying the functional requirements.

Requirement Elicitation

Target Demographic

Contributor(s): Kevin Le

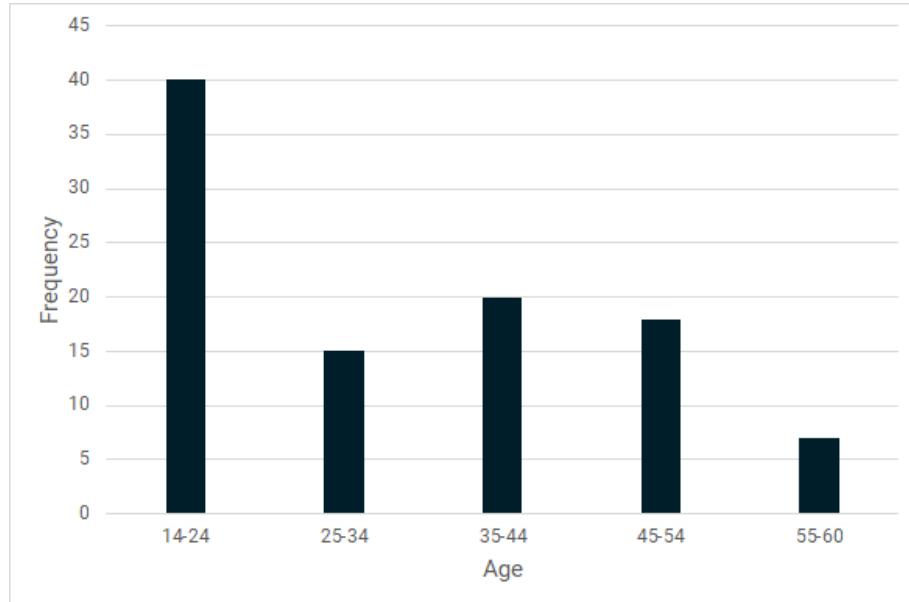
The demographic that is expected to use this program is in a relatively young to middle age range, averaging about 33 years old. They are moderately skilled in the use of technology, being able to interact with computers via mouse and keyboard input. The expected uses for the program will be both recreational and professional. With companies expected to use it and customize it to accommodate their workflow, and regular people who will be using it for staying in contact with friends and family, or discussing their hobbies and topics they are interested in.

In order to establish our target market, functionality to include or add, and what users dislike about their current to avoid, the survey was conducted with 100 people of varying contexts that use Internet Relay Chat (IRC) services. These are the results and information extracted for the following questions:

Survey Result and Analysis

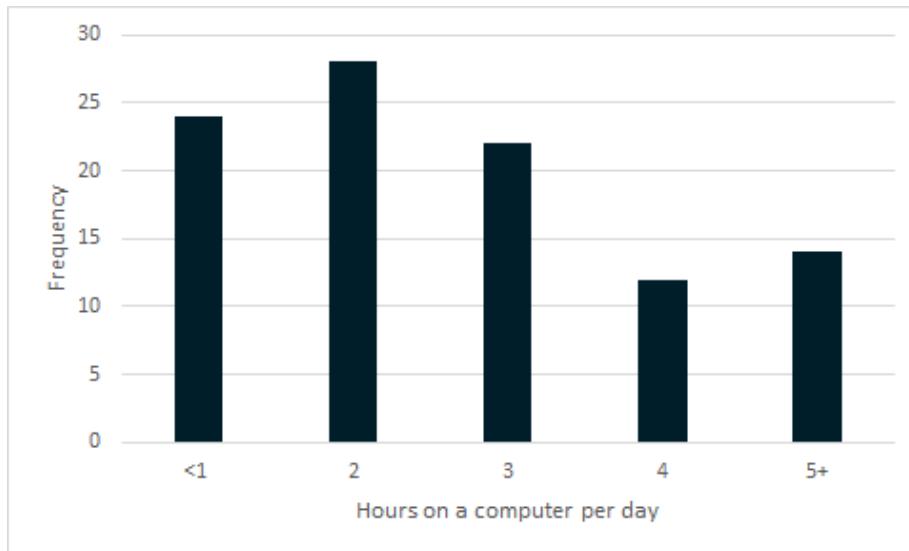
Contributor(s): Kevin Le

Q1: What is your age?



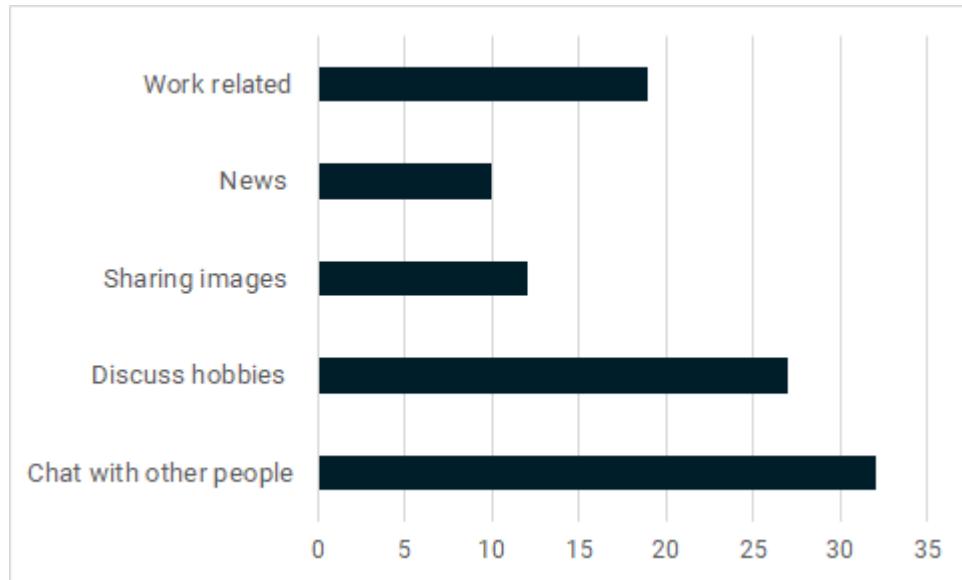
Data shows that many users of the IRC platforms fall into the 14-24 year old category, many being relatively young with a mean age of 33.17 years. This suggests that they are of moderate-high technical ability.

Q2: Roughly how many hours do you use your computer every day?



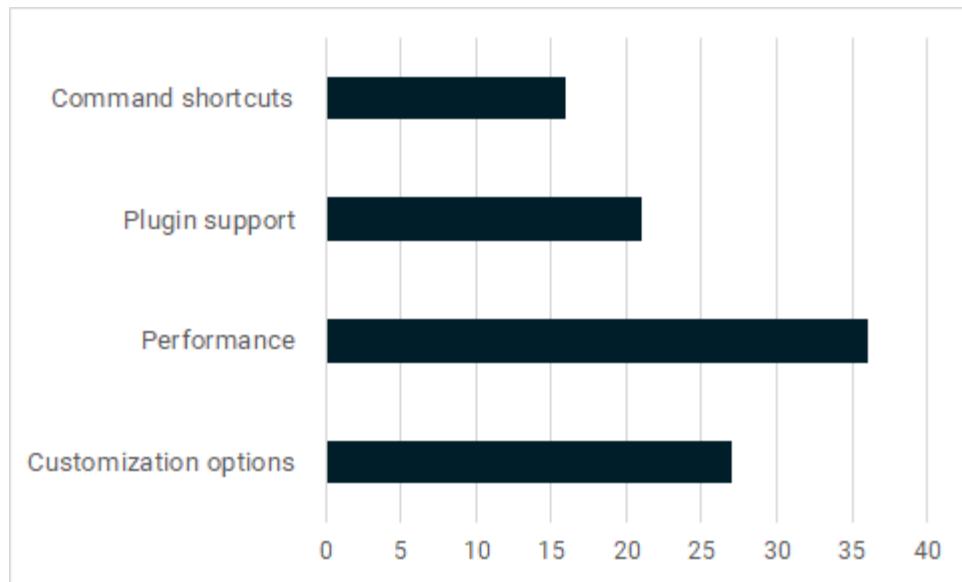
Most respondents indicated they use their computers for about 2 hours everyday, with quite a large spread. Average time spent on a computer was roughly 3 and a half hours. This suggests that they are of moderate-high technical ability.

Q3: What do you mainly use IRC for?



Respondents indicated they mostly used IRC simply to chat with other people. But a large percentage of respondents also indicated they use IRC to discuss hobbies. This is not surprising as IRC servers are self hosted and many target a variety of topics. Work also took up a large percentage also, this is likely due to some companies wanting full control over their communications, which self hosted IRC servers can allow them to and keep confidential communications in house.

Q4: What do you like most about your current IRC client?



Out of all respondents, most mentioned the performance of IRC, being lightweight and able to run on low powered machines as the most liked feature of their current IRC client, this makes sense as some other messaging applications such as Facebook messenger and Wechat on desktop are surprisingly intensive despite being text chat.

Second most liked feature of their current clients was the customization options, it provided deep configuration options in the form of JSON configuration files. Allowing the user to do whatever they wanted, even if it could break the program.

Plugin and scripting support was of major importance to a group of users, who use IRC to track project workflows or to add even more functionality to the application.

Command shortcuts include features such as: /msg User1 hello

Q5: What are the problems you have with your current IRC client?

Common complaints from respondents were:

Steep learning curve: Introducing new people to the system was difficult, as there is no help provided, and the major IRC clients are quite information dense. We can address this by providing tooltips on parts that may be confusing. New users may also need to learn commands in order to access some functionality, e.g. direct messaging.

Information dense: This was a mix between bad and good, some users expressed that they like the information density, but a majority stated that their current clients presented messages and other information too tightly packed with walls of text. We can ensure both sides are happy by allowing control over the font size and types.

Q6: What other functionality would you like to see implemented?

Not many respondents answered this question, however the majority that did expressed the need for a global notification system, their current clients did not have the ability to send notifications.

Personas

Contributor(s): Kevin Le

Nathan Hamilton

Background

- Male, 22 years old.
- Occupation: Engineering Student

Motivations

- Learn new skills from other people who share the same hobby

Frustrations

- Existing clients are difficult to use and are unappealing

Nathan is a student at Bentley University, and his hobby is working with embedded systems in his spare time outside of study, he wants to discuss various topics related to the hobby, such as soldering, Arduinos, Raspberry Pis, IoT, and programming. His university group is not easily shared with people who also enjoy the same hobbies, as it cannot handle hundreds of users at the same

time with multiple users worldwide. Having everyone in the same channel is overwhelming also, so a chat application that can separate topics into sub topics would be great, which IRC can do. However he wants an application that is easy to use. Existing IRC platforms are difficult to use and are not visually appealing.

He downloads the new Weechat GUI and joins an Engineering server, he realises it's really easy to talk to people in certain topics and the application is really easy to use, with a really clean interface.

Elissa Mill

Background

- Female, 34 years old.
- Occupation: Software Engineer at Zoomy

Motivations

- Chat with her development team to keep updated of any changes
- See updates of recent git commits in line by installing a Git plugin.

Frustrations

- Existing IRC clients and messaging apps are too locked down, don't allow much customizations, she wants full control over her messaging app.
- Learning curve with existing IRC clients
- Wants plugins

Elisa is a software engineer at Zoomy, a flight booking startup, and is currently working on the Zoomy flight booking functionality. Her development team regularly makes commits which are built and run against multiple test cases. She wants a centralized team chat platform with Git integration that provides statuses of each update made to the program but doesn't cost money to run, since as a startup, they don't have much money to pay for premium systems with inbuilt developer tools. But most importantly Elisa wants to self host the messaging app, so the data her team shares isn't located on external servers, as they hold valuable company secrets. This means she needs to set up her own IRC server and introduce her team to IRC. However, Git integration in many IRC clients isn't possible, and those that allow adding plugins are too complicated to introduce her team to. Elisa discovered an IRC client called WeeChat with a user friendly interface, and was extremely easy to use, so her team could easily move to it. It also provided full control over what plugins and scripts to add, which was perfect as she could set up her own developer tools to provide git updates. This was fantastic as it can be an all in one portal for her team to communicate and keep updated on, at near no cost, enabling frictionless collaboration.

Blake Grant

Background

- Male, 16 years old.
- Occupation: High school student

Motivations

- Chat with his friend group
- Get good grades

Frustrations

- Everyone talking over one another when trying to talk about different topics
- Unable Customize his profile

Blake is currently attending Curtis College, he is in year 11, and studying Design, Math, English and Physics Atar. He is an avid player of MOBAs such as Dota 2 and so is his friends group, but when it comes time to get down and study for a test or do his assignment, he wants to chat with his other friends that are doing the same course, but not everyone does the same subjects! So he has to talk over someone talking about League of Legends when he is trying to calculate resistance over a wire in physics... Not optimal. With his current chat platform, he could just make a new group chat, but then he would end up with multiple group chats with different combinations of people. Blake knows about IRC, but knows his friends will not want to give up Facepage Messenger as it looks much better. What he needs is a program that separates his friends group, but still allowing people to join the discussion even if they aren't in the group chat. He finds a program called Weechat that looks like Facebook messenger, but has more features that he needs, such as having certain channels for certain topics. On top of that it allows Blake to really express himself with username colours and nicknames! It is also easy for his friends to pick up.

Curtis Van Dyke

Background

- Male, 63 years old.
- Occupation: Technology Consultant at Delight

Motivations

- Stay in touch with friends and family
- Learn to use new technology in his spare time

Frustrations

- Programs have text too small
- New technology has no tool tips or help guides.

Curtis is a technology consultant at Delight, and is really close with his family, he however often travels for work and likes to stay in touch with his friends and family. As a technology consultant, he has to keep up to date with the latest technologies and also enjoys picking up new technologies in his spare time. He has used IRC in the past but has switched to more modern looking platforms such as Facepage Messenger and eyeMessage. However, these programs don't allow him to change the font sizes without scaling the entire system fonts, with his deteriorating eyesight, he would prefer if he could scale fonts in just the program up, just so it is easier on his eyes, and since scaling broke some other programs. These chat programs also lack help guides so it takes him

longer to pick up how to use it. However Curtis recently discovered a program called Weechat. The GUI it had was just as modern as the current era chat platforms, it also used IRC which he was familiar with. However anything he was unsure of, the program had tooltips on each button and help guides which was amazing! But best of all, there was an option to scale the interface font size, which was perfect for him.

Competitor Analysis

Contributor(s): Aaron Gangemi

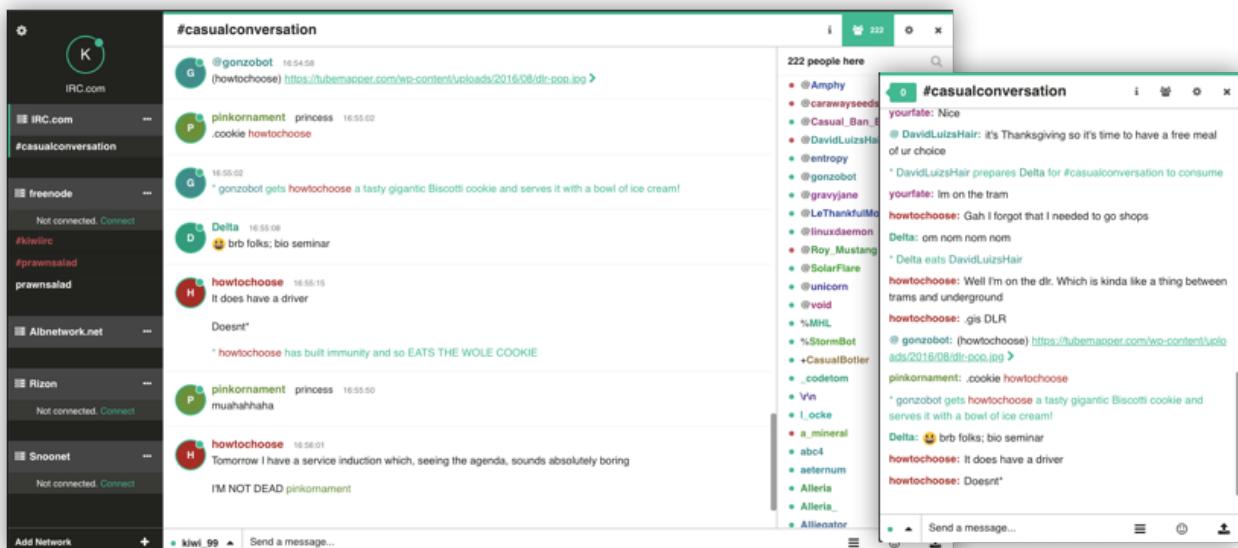
Kiwi IRC (Web Client)

Kiwi IRC (Internet Relay Client) is a free open source client that allows users to connect to both servers and channels and further communicate with other users connected to the channels in the form of text. Kiwi IRC contains a range of different features including but not limited to:

- Theme modification of the GUI
- Plugin management
- Available across multiple devices including desktop, phone and tablet
- Ensures secure communication using SSL
- Available in different languages other than English
- Text Styling
- Incorporation of User Scripts

Disadvantages of KiwiIRC:

- Notifications sent by Kiwi IRC are fixed on screen and do not time out. This means notifications must be closed manually.
- Requires additional installation and dependencies in order to run.
- Quite resource intensive because KiwiIRC runs on browser



HexChat

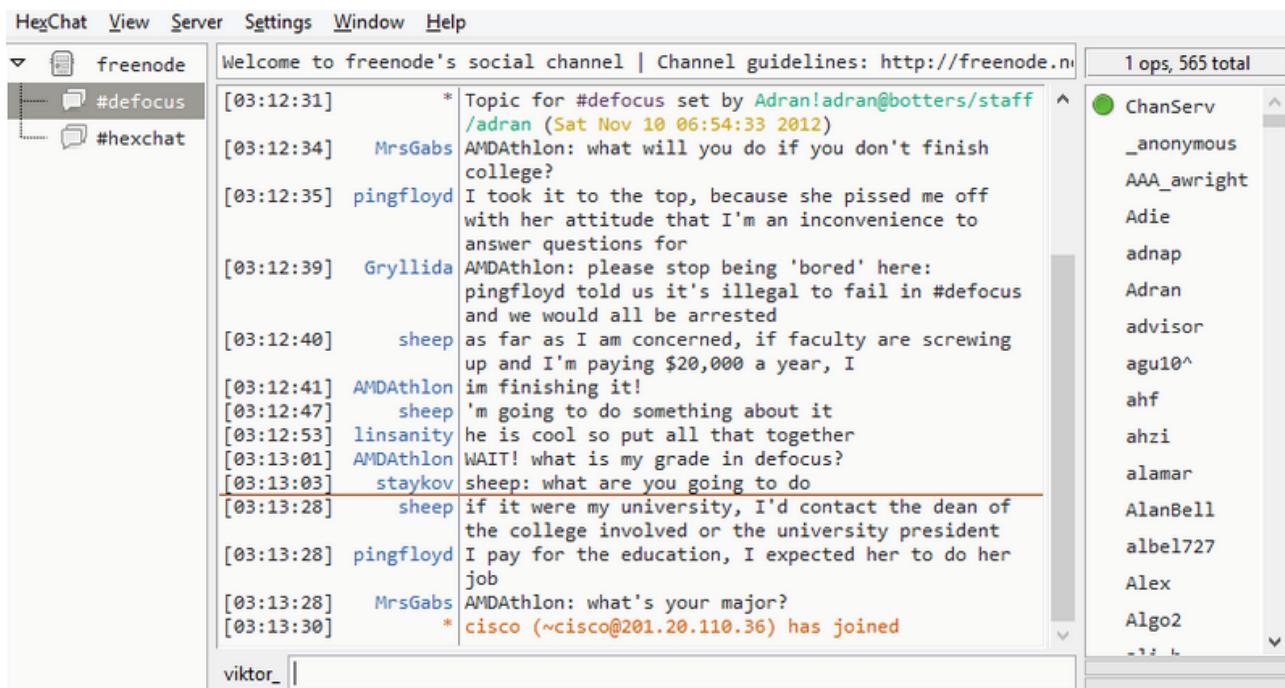
HexChat is an open-source cross-platform IRC client. It is completely based on XChat. Features of

HexChat include:

- Easy to use and customisable interfaces
- Cross-platform with Windows and Unix based Operating Systems
- Highly compatible with Lua, Python and Perl scripts
- Can be translated into multiple languages other than English
- Completely open source and is actively developed

Disadvantages of HexChat:

- Bad character support: Hexchat requires the user to have certain plugins installed in order to write certain characters
- High CPU usage on Windows 8.1 and Windows 10 operating systems
- Poor DPI display support: When the DPI is set to high, the text is blurry



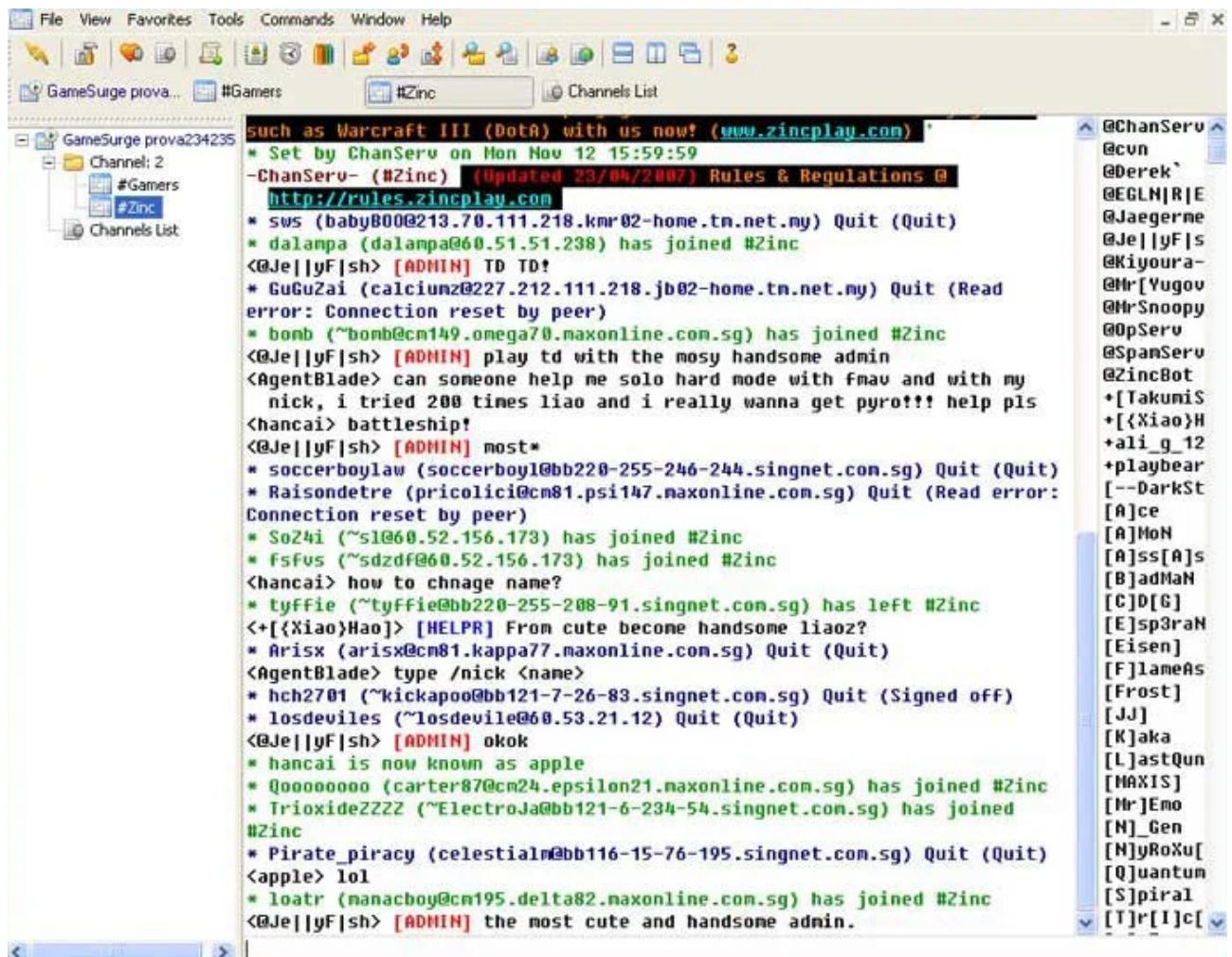
mIRC

Is an IRC client used to communicate, share, play or work with other users that are also connected to other IRC networks around the world. Features of mIRC include:

- Clean and practical interface
- Incorporates its own powerful scripting language
- Almost all actions can be executed via a graphical button which makes it easy to use
- Each server is able to contain different settings
- Contains a free trial before use

Disadvantages:

- After the free trial provided by mIRC, the software costs \$20 to use
- Does not incorporate a command line solution as well as the GUI
- mIRC was developed in 1995, and ever since, there has been a lack of support from the developer
- Dated GUI
- The developer has not provided much licensing information



Requirement Specification

Usability Goals

Contributor(s): Kevin Le

Based on our requirements elicitation and research, three key usability goals we have identified that our program should aim to meet is:

- UG-1:** Efficient to use
- UG-2:** Easy to learn
- UG-3:** Have good utility

For UG-1, the program must facilitate a frictionless method of carrying out the user's desired goal, which in this case is communicating with a group of people. It must not be obstructive in the users quest to achieve this goal. A way to achieve this efficiency is reduce the amount of clicks, taps or interactions required to perform a certain feature or reach a certain page.

We believe UG-2 is especially important, in order to attract new users and retain people, the interface must be easy to learn, where the user is able to pick it up and understand how to use it in

a reasonable timeframe. An interface that is difficult to use will never keep users, as people who are introduced to it are put off by complexity.

For UG-3, the program ultimately must have good utility, or no one will want to use it. It must fulfill its goal of providing an interface for IRC communication. With functionality that caters people with different accessibility needs, more advanced users or less technically skilled users. With all this not cluttering the program.

User Experience Goals

Contributor(s): Kevin Le

Based on our requirements elicitation and research, three key user experience goals we have identified that our program should aim to meet is:

UEG-1: Enhancing Sociability

UEG-2: Enjoyable

UEG-3: Helpful

The reasons for choosing these user experience goals were very clear to us. Starting with UEG-1, it is essentially the main and most important goal we strive for in our IRC client, as with any other messaging platforms. Which is allowing people to socialize with one another via text, and connecting people.

Regarding UEG-2, the program must be enjoyable to use, a program that makes people hate using it, will lead them to not having any motivation and desire in using it, leading to lower adoption rates.

Ultimately, with UEG-3, the program aims to be helpful, allowing people to customize it, and allowing more advanced users to add plugins or scripts to help support their goals.

By aiming to achieve these goals, we believe it would be the differentiator for our program in the market of existing IRC clients, and promote good user experiences with our program by striving to create these emotions around our product.

Functional Requirements

Contributor(s): Kevin Le

FR-1: Users shall be able to login with an existing account

FR-2: Users shall be able to register a new account providing username, email, password and username colour.

FR-3: Users shall be able to send a message to a server channel

FR-4: Users shall be able to attach a file from the file explorer in server and direct message chats.

FR-5: Users shall be able to send a direct message to a person/ individual.

FR-6: Users shall be able to view tooltips for buttons with no textual information

FR-7: Users shall be able to change channel they are currently in

FR-8: Users shall be able to change server they are currently in

- FR-9:** Users shall be able to edit their profile consisting of username, username colour, email and password
- FR-10:** Users shall be able to add an IRC server
- FR-11:** Users shall be able to customize the program interface font type, font size, font colour and interface colour.
- FR-12:** Users shall be able to search for plugins
- FR-13:** Users shall be able to upload their own plugins
- FR-14:** Users shall be able to manage current plugins by viewing information, activating/deactivating, or deleting.
- FR-15:** Users shall be able to control (activate/ deactivate) notification banners and sounds
- FR-16:** Users shall be able to view program about information
- FR-17:** Users shall be able to logout (end current session)
- FR-18:** Users shall be able to delete their account

Non-Functional Requirements

Contributor(s): Kevin Le

Performance

- P-NFR-1:** The program shall not take more than 4.0 seconds to present the login page
- P-NFR-2:** The program shall not take more than 2.0 seconds to present the register page
- P-NFR-3:** The program shall not take more than 5.0 seconds to load the main server UI on successful login/ register
- P-NFR-4:** The program shall not take more than 2.0 seconds to change channels
- P-NFR-5:** The program shall not take more than 4.0 seconds to change servers
- P-NFR-6:** The program shall not take more than 0.5 seconds to send a new message
- P-NFR-7:** The program shall not take more than 1.0 second to open the options page
- P-NFR-8:** The program shall not take more than 0.5 seconds to change to a different option
- P-NFR-9:** The program shall not take more than 0.3 seconds to change to the direct messages page.
- P-NFR-10:** The program shall not take more than 1.0 second to open the file explorer (on attach)

Security

- S-NFR-1:** The program must ensure no passwords are visible
- S-NFR-2:** The program must ensure it sanitizes all invalid characters
- S-NFR-3:** The program must provide confirmation dialogs for destructive actions

Usability

- U-NFR-1:** The program shall provide tooltips for buttons without text
- U-NFR-2:** The program shall use standard recognizable icons for certain tasks. E.g. paperclip to attach.
- U-NFR-3:** The program shall use a consistent low impact colour theme
- U-NFR-4:** The program shall provide feedback on actions or intended actions. E.g. select button on hover, and indent button on click.

Localization

L-NFR-1: The program interface shall be in English (UK)

L-NFR-2: The program interface shall comply with relevant laws regarding accessibility

Reliability

R-NFR-1: The program shall operate with 99% error free rate.

R-NFR-2: The program shall be able to handle at least 150 messages shown at once without failing.

User Stories

Contributor(s): Kevin Le

US-1: As a user, I want to be able to login with an existing account

US-2: As a user, I want to be able to register a new account

US-3: As a user, I want to be able to send a message to a server channel

US-4: As a user, I want to be able to attach a file from my computer to send to a channel or person.

US-5: As a user, I want to be able to send a direct message to a certain person

US-6: As a user, I want to be able to view tooltips for buttons that I'm not sure what it does

US-7: As a user, I want to be able to change channel they are currently in

US-8: As a user, I want to be able to change server they are currently in

US-9: As a user, I want to be able to edit my profile information

US-10: As a user, I want to be able to add an IRC server and give it a name

US-11: As a user, I want to be able to customize the program interface

US-12: As a user, I want to be able to add and manage plugins

US-15: As a user, I want to be able to control notification settings

US-17: As a user, I want to be able to logout

US-18: As a user, I want to be able to delete my account

Use Cases

Contributor(s): Aaron Gangemi

The use cases below are used to depict the core functionality of WeeChat.

UC-1

Goal: User is able to add a server to the list of servers

Primary Actor: User

Precondition(s):

1. User must be logged in successfully
2. Must be on the main server interface

Trigger: User selects the Add Server button

Flow of Events:

1. System displays main server page
2. User selects a "Add Server" button or icon
3. User displays add server page

4. User enters the Server URL and Name for the server
5. User displays back to Main Server page
6. Inputted server name and URL is added to server list on main page
7. Channels also displayed in a channel list for each server

Extensions:

1. User selects back button
2. Program displays Main Server page
3. No server is added to server list
4. Use case ends
5. User inputs invalid server address data
6. Alert is displayed
7. Use case continues from step 4

UC-2

Goal: User is able to send a message to connected server

Primary Actor: User

Precondition(s):

1. User must be successfully logged in
2. User must be on Main Server page
3. User must be connected to a server

Trigger: User enters text in text field and presses send

Flow of Events:

1. Program displays main server page
2. User selects server they would like to send text to
3. User enters text into text field
4. User clicks send button/presses Enter key
5. Text displays in server chat with username and time and date of message sent

Extensions:

1. User backspaces/deletes typed message
2. Use case continues from step 2

UC-3

Goal: User is able to send an file attachment to server chat

Primary Actor: User

Precondition(s):

1. User must be successfully logged in
2. User must be on Main Server page
3. User must be connected to a server

Trigger:

1. User selects either an image/attachment icon in main server chat

Flow of Events:

1. Program displays main server page
2. User selects server they would like to send attachment to
3. User selects the file that will be displayed in file explorer
4. File attachment is displayed in channel chat

Extensions:

1. User selects cancel in file explorer
2. Use case ends

UC-4

Goal: User is able to send a private message to another user

Primary Actor: User

Preconditions(s):

1. User must be successfully logged in
2. User must be on direct messages page
3. User must have other users in list

Trigger: User must click on either a direct messages icon or button which is located on various interfaces throughout the application

Flow of Events:

1. Program displays direct messages page
2. User selects a user in the user list that they would like to message
3. User types text that they would like to send to recipient
4. User clicks "Send" button
5. Text is successfully displayed in chat

Extensions:

1. User decides to send attachment instead
2. User selects attachment/image icon
3. File explorer opens
4. User selects file they would like to send
5. Use case continues from step 4

UC-5

Goal: User is able to upload a script to the script table

Primary Actor: User

Precondition(s):

1. User must successfully be logged in
2. User must be on Script Manager/Plugins page
3. Script must exist in a directory on the users system

Trigger: The user selects the upload button on the script manager interface

Flow of Events:

1. Program displays script manager/plugins page
2. User selects upload page
3. File explorer opens
4. User goes to directory containing script
5. User selects path of containing script
6. User selects okay
7. Script is uploaded to script table

Extensions:

1. User selects cancel in File Explorer
2. Use case continues from step 1

UC-6

Goal: User is able to search for a script online

Primary Actor: User

Precondition(s):

1. User must be successfully logged in
2. User must be on Script Manager/Plugins page
3. Script search must be a valid URL

Trigger: The user has entered a valid URL and selects search on the script manager interface

Flow of Events:

1. System displays the script manager/plugins page
2. User types in a valid URL for the script

3. User selects Search
4. Program searches for the script online
5. Program finds the script
6. Program adds script to script table

Extensions:

1. Script URL is invalid
2. Program throws an error
3. User returns back to script manager page with no script added to table
4. Use case ends

UC-7

Goal: User is able to modify their existing Weechat options and settings

Primary Actor: User

Preconditions:

1. User must be successfully logged in
2. User must be on Settings page

Trigger: The user selects a different option in the combo box and the change is applied

Flow of Events:

1. System displays options/settings page
2. User selects the combo list for a setting
3. User clicks a different option listed
4. Setting is applied

Extensions:

1. User clicks out of combo list
2. Setting is set as original setting

UC-8

Goal: User is able to search for a connected user in the server chat

Primary Actor: User

Precondition(s):

1. User must be successfully logged in
2. User must be connected to a server that contains more than 1 person

Trigger: User selects search bar icon

Flow of Events:

1. System displays Main Server page
2. User selects search for user bar
3. User types desired users name in
4. User appears highlighted in connected users table if connected

Extensions:

1. User does not appear on search
2. Search is cancelled
3. Return focus to main server page
4. End use case

UC-9

Goal: User is able to edit their account information:

Primary Actor: User

Precondition(s):

1. User must be successfully logged into their account
2. User must have an account

Trigger:

1. User selects edit icon or field to edit in order to modify account details

Flow of Events:

1. System displays user information page
2. User selects edit icon
3. Fields "unlock" so they can be edited
4. User edits fields that they would like change
5. User saves data

Extensions:

1. User enters invalid data
2. Program displays alert
3. Use case continues from step 4
4. End of use case
5. User selects back
6. Program returns to main server page

Hierarchical Task Analysis

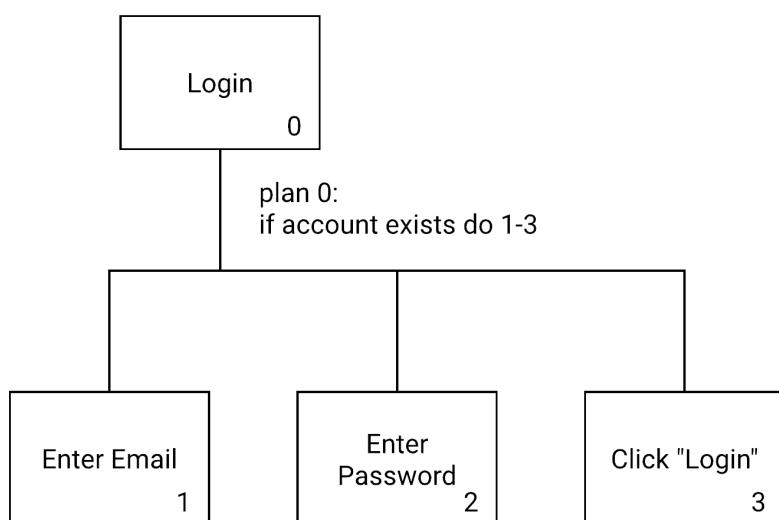
Contributor(s): Kevin Le, Aaron Gangemi

The hierarchical task analysis for Weechat was built after establishing a firm set of requirements that would be integrated in the Weechat program. These features are listed in the functional requirements for Weechat. The diagrams below will illustrate the features that are depicted in each interface.

Login

Contributor(s): Kevin Le

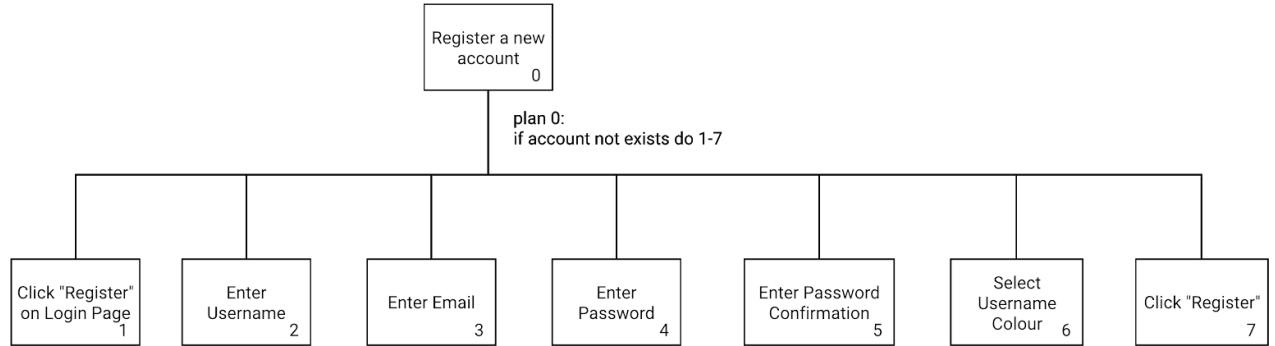
Tasks users expected to do when logging in.



Register

Contributor(s): Kevin Le

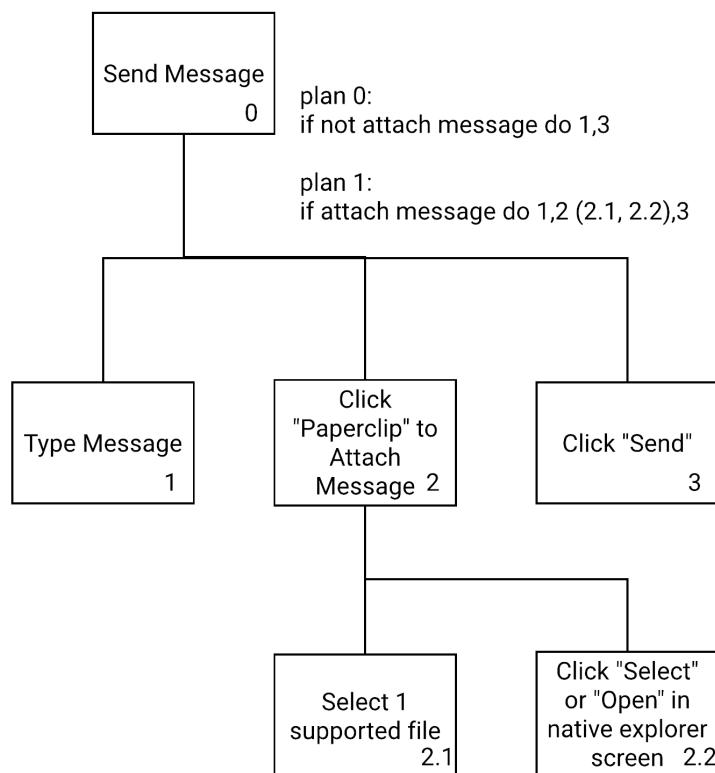
Tasks users expected to do when registering a new account. Initiated by clicking "Register" on the login page.



Send Message

Contributor(s): Kevin Le

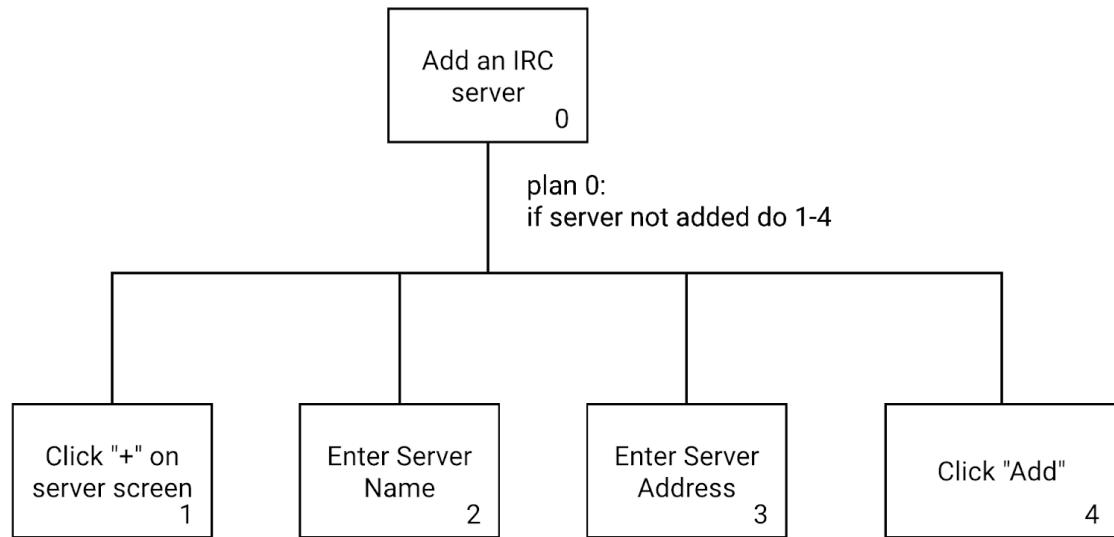
Tasks users expected to do when sending a message. Applies to both server/ channel and direct messages.



Add Server

Contributor(s): Kevin Le

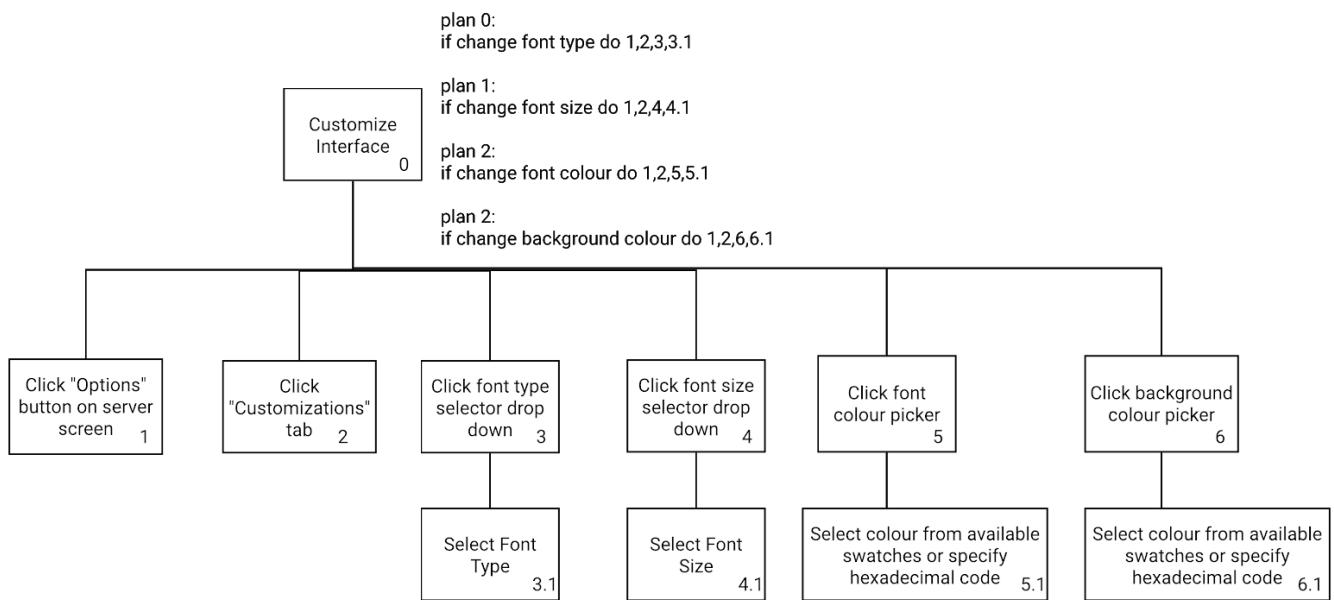
Tasks users expected to do when adding a new server. Initiated from the server screen.



Customization

Contributor(s): Kevin Le

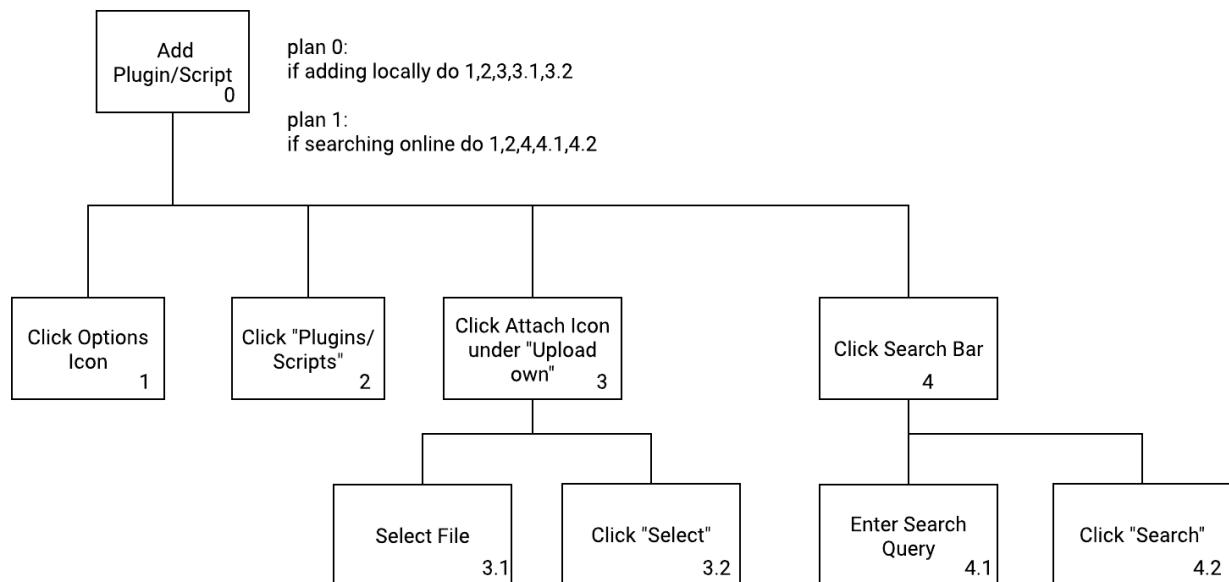
Tasks users expected to do when customizing the program interface.



Plugin/ Script Manager

Contributor(s): Kevin Le

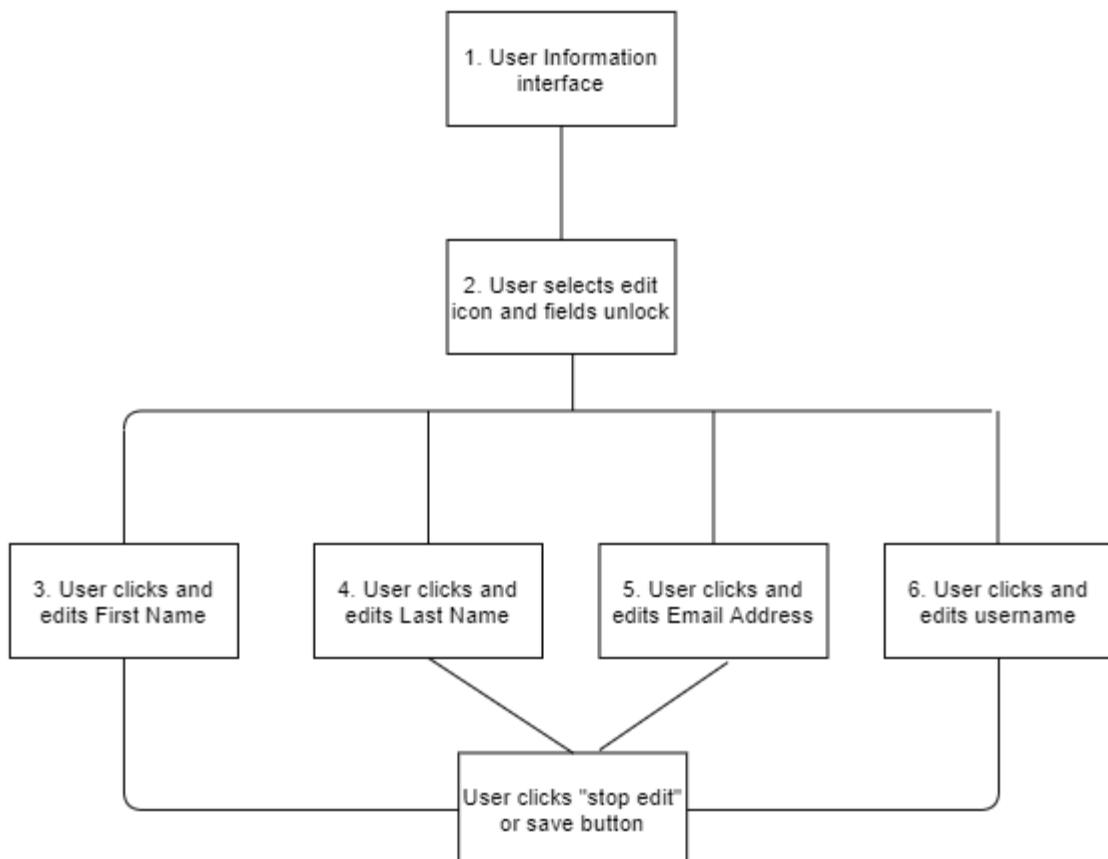
Tasks users are expected to do when uploading a script to the script manager/plugins



Modify Account Details

Contributor(s): Aaron Gangemi

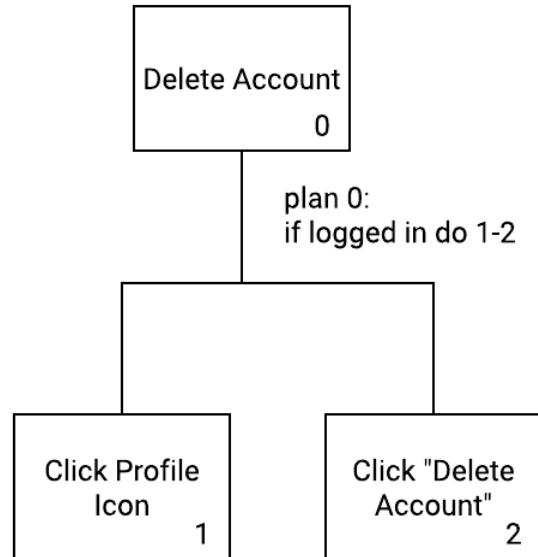
Tasks user is expected to complete when user modifies account details



Delete Account

Contributor(s): Kevin Le

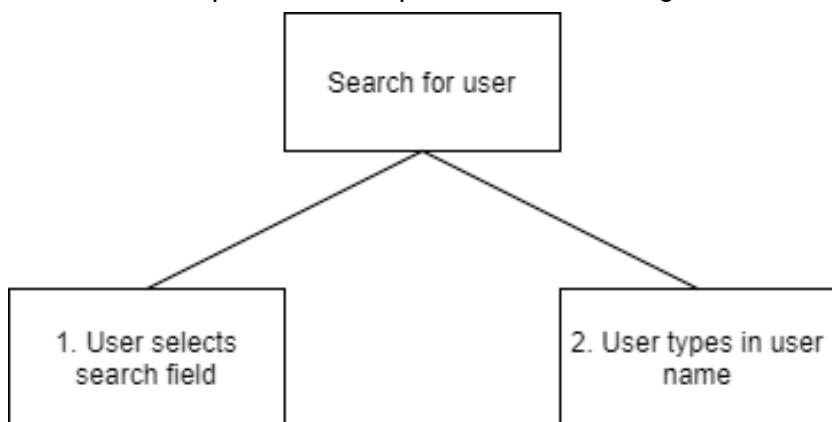
Tasks user is expected to complete when user modifies account details



Search for User

Contributor(s): Aaron Gangemi

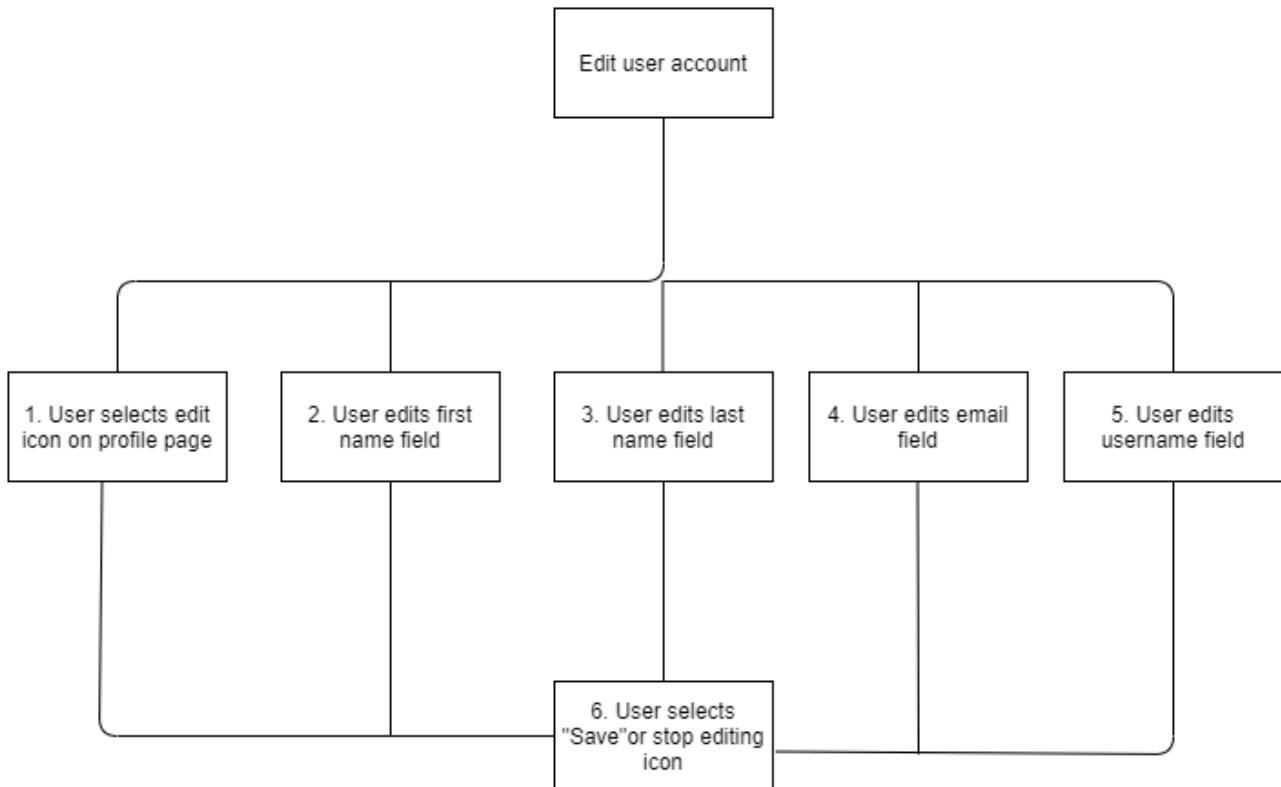
Tasks user is expected to complete when searching for username



Edit user account information

Contributor(s): Aaron Gangemi

Tasks user is expected to complete when a user wants to edit their account information



Prototypes

Contributor(s): According to names

Aaron Gangemi (19447337)

Low Fidelity

Direct Messages	
① <username>	<user1.username>
<user 2>	② me: mm
<user 2>	mm: user1
<user 3>	me: mm
<user 4>	mm: user1
:	me: mm
:	mm: user1
:	mm: user1
<input type="text" value="Type here"/> ⑥ ▶	

Options & Settings

- User
- Information
- Options & Settings
- Script Manager
- ← Logout
- ↓ Exit Weechat

V About (28)

V Notifications

- Receive Mentions
- Direct Messages

V Customisations

- Font Size: 22 ▾
- Background Color: Blue ▾
- Text Color: Black ▾

V Date / Time

- Perth (C +0:800) ▾

(28) "About", "Notifications", "customisations" are dropdown text which are settings than can be modified

Script Manager				
① Use Information	② Options & Settings	③ Script Manager	④ Search	⑤ Command
Script Manager	Logout	Script Name	Date	Description
Exit Wechat		~	~	~
		~	~	~
		~	~	~
		~	~	~

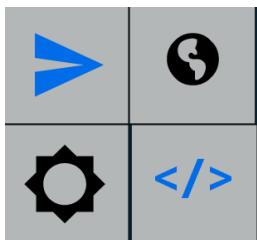
High Fidelity

Common features across interfaces:

All icons from <https://www.xd guru.com/adobe-xd-icons-material/>

Weechat logo from: <https://en.wikipedia.org/wiki/WeeChat>

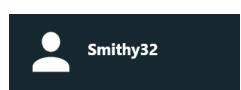
GLM Logo created by GLM Group members



These icons in the top corner take the user to 4 different interfaces. The messenger symbol will take the user to the direct messages interface ([Figure 7](#)). The globe interface takes the user to the Main Server page ([Figure 8](#)). The settings icon will take the user to options and settings ([Figure 4](#)). The “</>” symbol will take the user to the script manager ([Figure 5](#)). This will usually be located in the top right hand corner



The icon on the left will sign the user out, in which they will be redirected to the logout page ([Figure 1](#)). The icon on the right will close the WeeChat application. These can be found in the left panel of the applications interfaces.



This icon is presented in most interfaces after the user logs in. If clicked, the user will be able to view their profile information.



This arrow will redirect the user to the previous page.

Figure 1: Login Page

The below image is the first page that will be displayed in my prototype of WeeChat. It presents a username and password input field to be entered. The user also has the option of ticking “Remember Me” which will automatically sign them in after the first attempt. In addition, if the user forgets their username/password, it can be retrieved. Following this, they click login, in which they will be redirected to the main server page ([Figure 8](#)). However, if the user does not have an account, they are able to click “Create an Account” in which they will be redirected to [Figure 2](#).

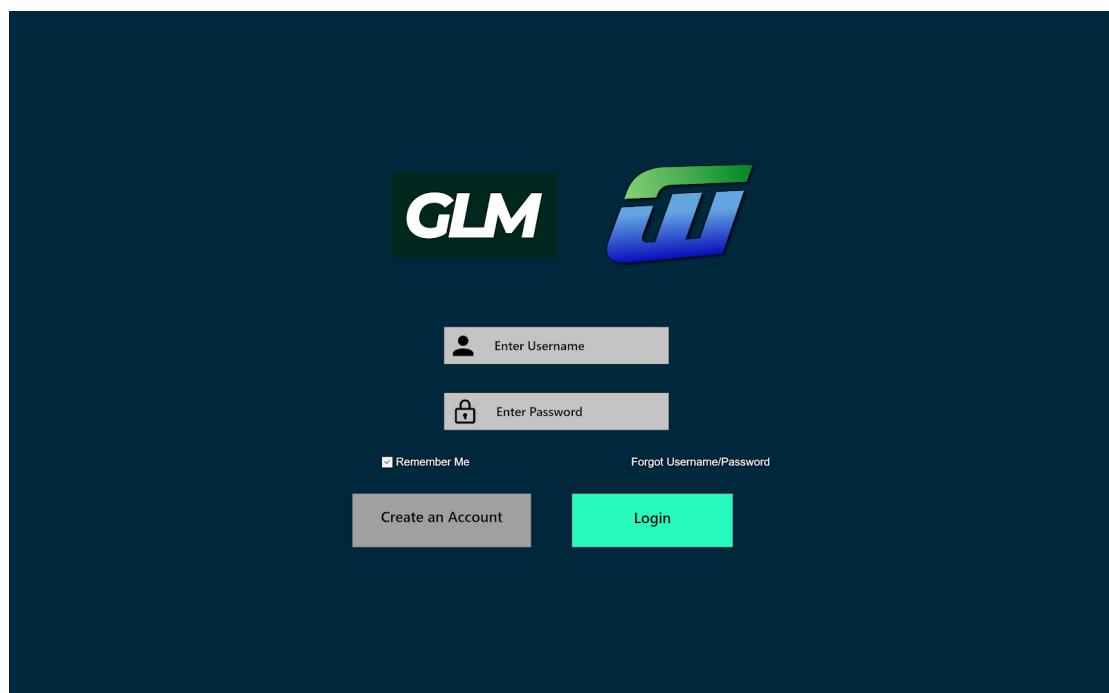
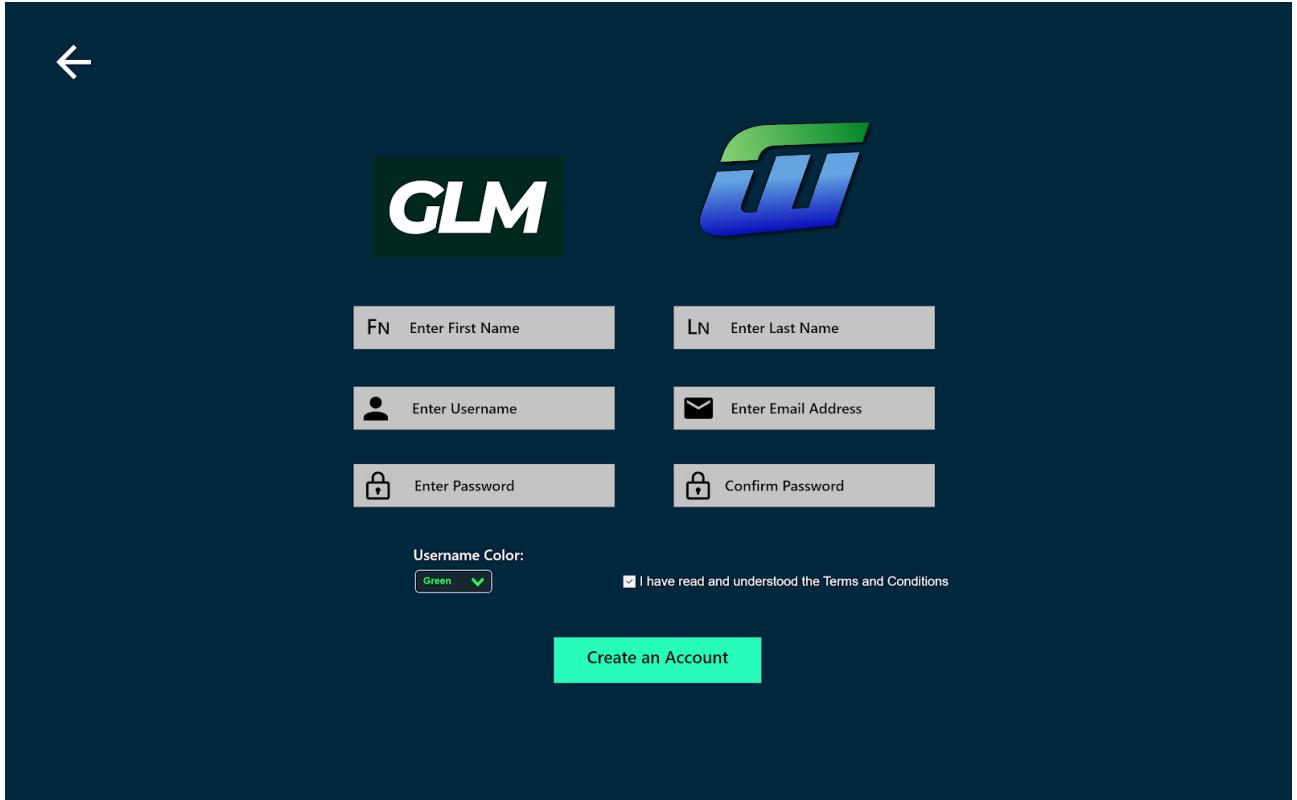


Figure 2 : Create an Account

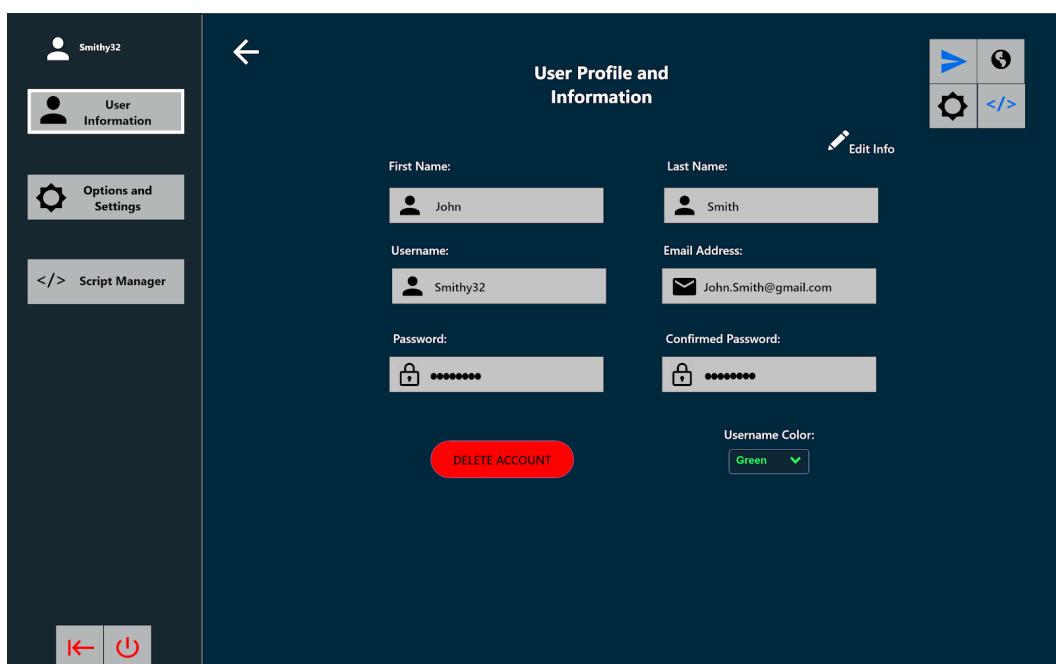
The below image illustrates the prototype for the “Create an Account.” The page contains fields for First Name, Last Name, Username, Email Address, Password and confirmation of Password. The user will also be required to select their Username Color. The user is also required to agree to the Terms and Conditions before being able to login to the application. Upon successful creation of a users account, they will be redirected to the server page [Figure 8](#).



A screenshot of a mobile-style "Create an Account" form. At the top left is a back arrow icon. In the center is a logo consisting of the letters "GLM" in white on a dark green rounded rectangle, with a stylized blue and green swoosh graphic to its right. Below the logo are six input fields arranged in two columns of three. The first column contains "FN Enter First Name" with a person icon, "Enter Username" with a person icon, and "Enter Password" with a lock icon. The second column contains "LN Enter Last Name" with a person icon, "Enter Email Address" with an envelope icon, and "Confirm Password" with a lock icon. Below these fields is a "Username Color:" dropdown menu set to "Green" with a green checkmark. To the right of the dropdown is a checkbox labeled "I have read and understood the Terms and Conditions". At the bottom is a large teal-colored "Create an Account" button.

Figure 3 : User Profile and Information

Figure 3 illustrates the User Profile and Information page. This page displays the user data that the user entered on login or the user data that has been most recently modified. If the user clicks “Edit Icon”/Pencil Tool, all fields will unlock and the user will be able to edit their information. The options in the left panel allow the user to view the associated interfaces ([\(Figure 3\)](#), [\(Figure 4\)](#), [\(Figure 5\)](#)).



A screenshot of a "User Profile and Information" form. On the left is a vertical sidebar with a user profile icon and the name "Smithy32", followed by three buttons: "User Information", "Options and Settings", and "</> Script Manager". At the bottom of the sidebar are red "←" and "⟳" navigation icons. The main content area has a back arrow at the top left. The title "User Profile and Information" is centered above the form fields. There are two columns of fields: "First Name:" with a person icon and "Last Name:" with a person icon, both currently showing "John" and "Smith"; "Username:" with a person icon and "Email Address:" with an envelope icon, both currently showing "Smithy32" and "John.Smith@gmail.com"; "Password:" with a lock icon and "Confirmed Password:" with a lock icon, both currently showing masked password entries. To the right of the fields are four small square icons: a blue arrow, a gear, a settings gear, and a double slash. Above the "Edit Info" icon is a pencil icon. At the bottom left is a red "DELETE ACCOUNT" button. At the bottom right is a "Username Color:" dropdown menu set to "Green" with a green checkmark. The entire interface has a dark theme.

Figure 4 : Options and Settings

Figure 4 illustrates the options and settings menu for the user. All fields are drop down lists. The "About" field contains a description regarding what WeeChat is about. "Notifications" gives the user the options of which notifications they wish to receive. "Customisations" allow the user to change the font size and text color of their text sent and the background color of WeeChat. The "Date/Time" field will contain a drop down list in which the user is able to select their desired Date/Time. This will modify the Date/Time displayed in the Direct Messages interface ([Figure 7](#)).

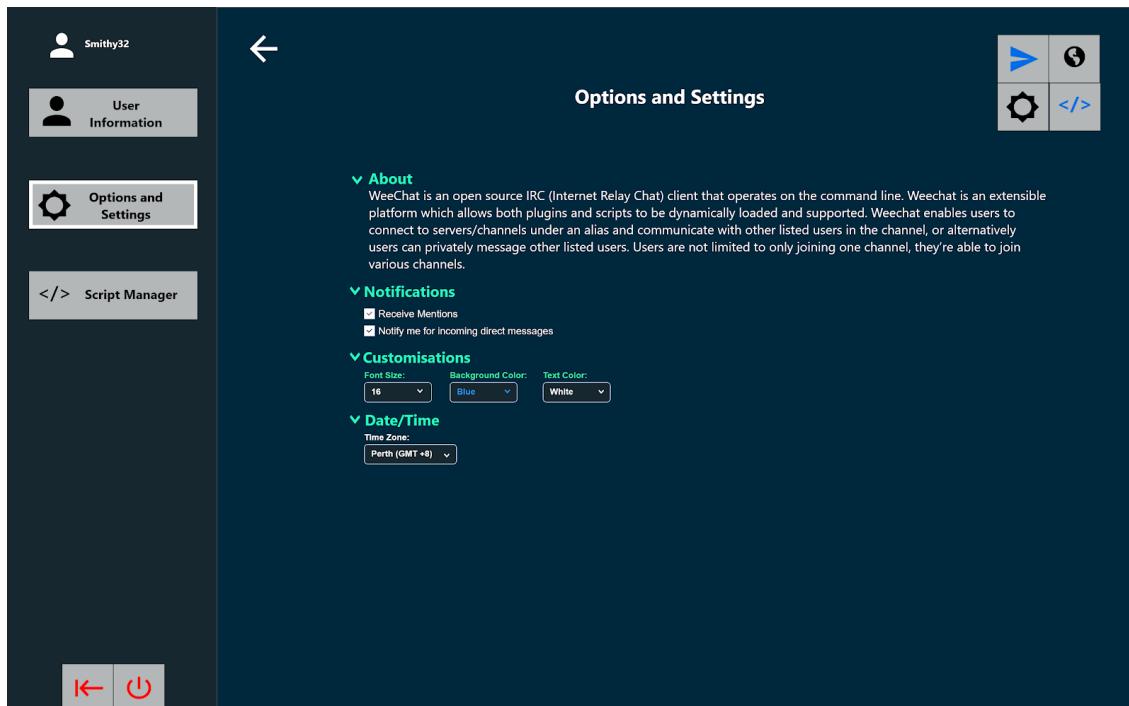


Figure 5 : Script Manager

The image below depicts WeeChat's script manager. Using the textboxes on display, the user will be able to either upload a script from the computer they are currently using to access WeeChat, or enter a link to find a script online. The table below those fields displays information regarding previously uploaded scripts.

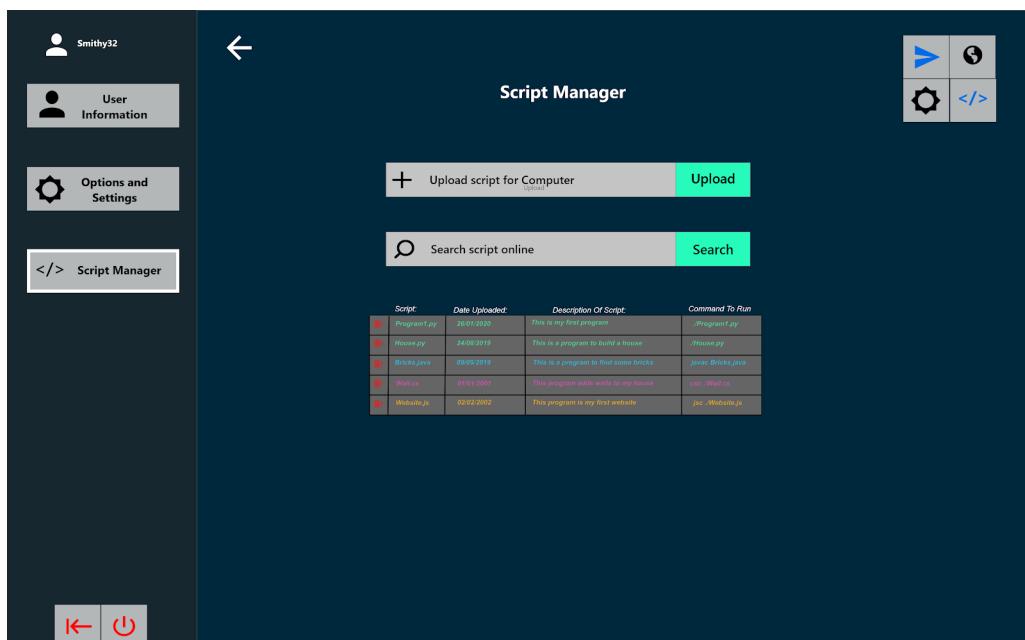


Figure 6 : Add Server

The below interface is used to add a server to the main server page ([Figure 8](#)). The user is required to enter a server URL and a server name which will be displayed for the user, as depicted in [Figure 8](#). Once the user has completed filling this form in, they can select “Add Server” to progress successfully.

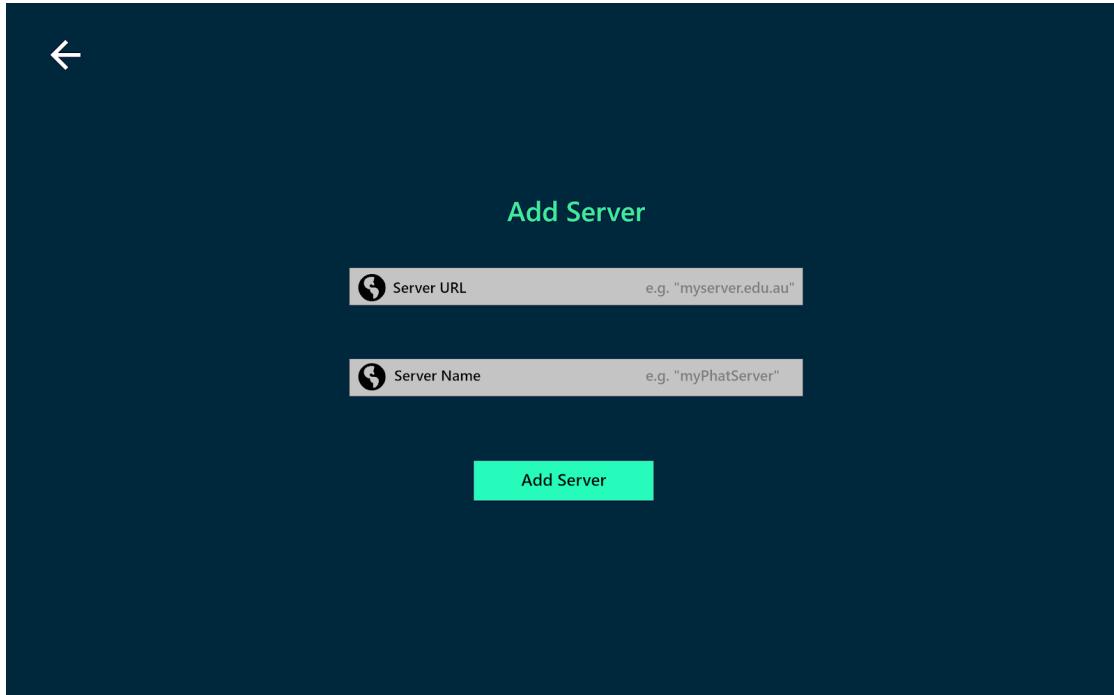


Figure 7 : Direct Messages

The image below illustrates WeeChat's direct messages interface. The username listed at the top indicates the user that the currently logged in user is messaging. After a message is sent from one user to another, the message will be displayed in the chat (text in white). Each username is highlighted based on their profile preferences. The icons in the enter text field allow the user to attach files and images and send emoji's. The user can reach this page by clicking on a user's name in the main server page ([Figure 8](#)) or click the direct message symbol in the top corner.

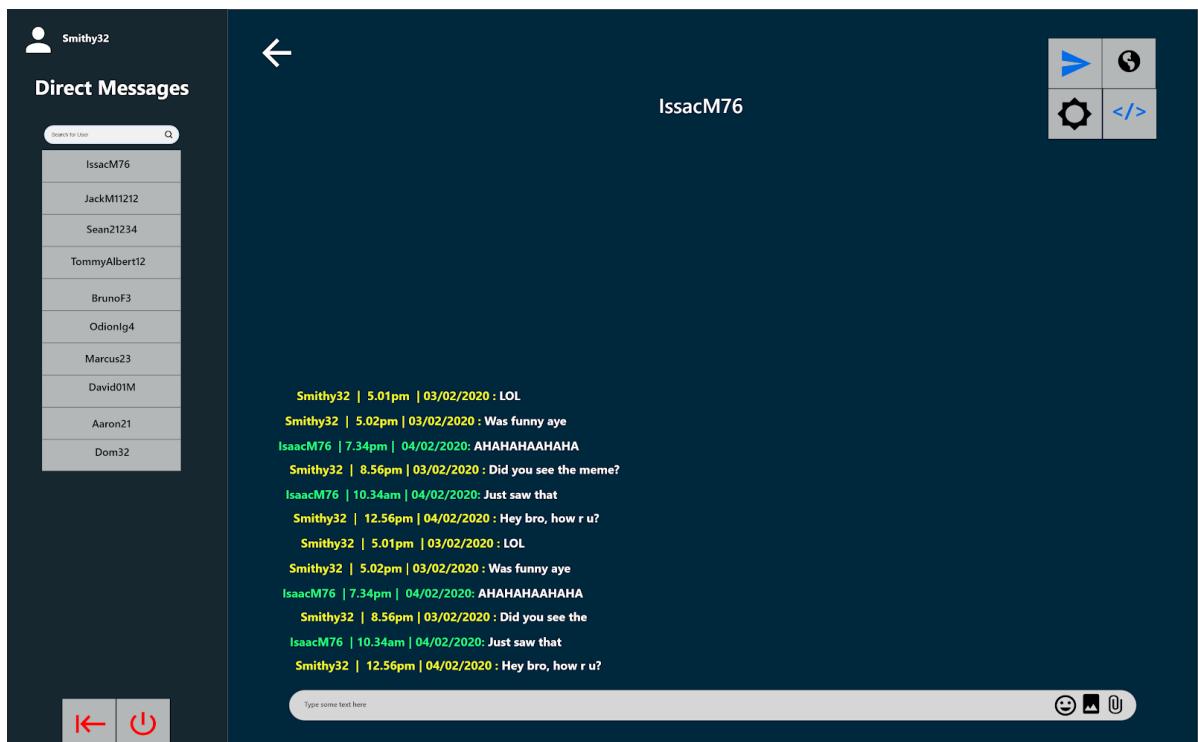
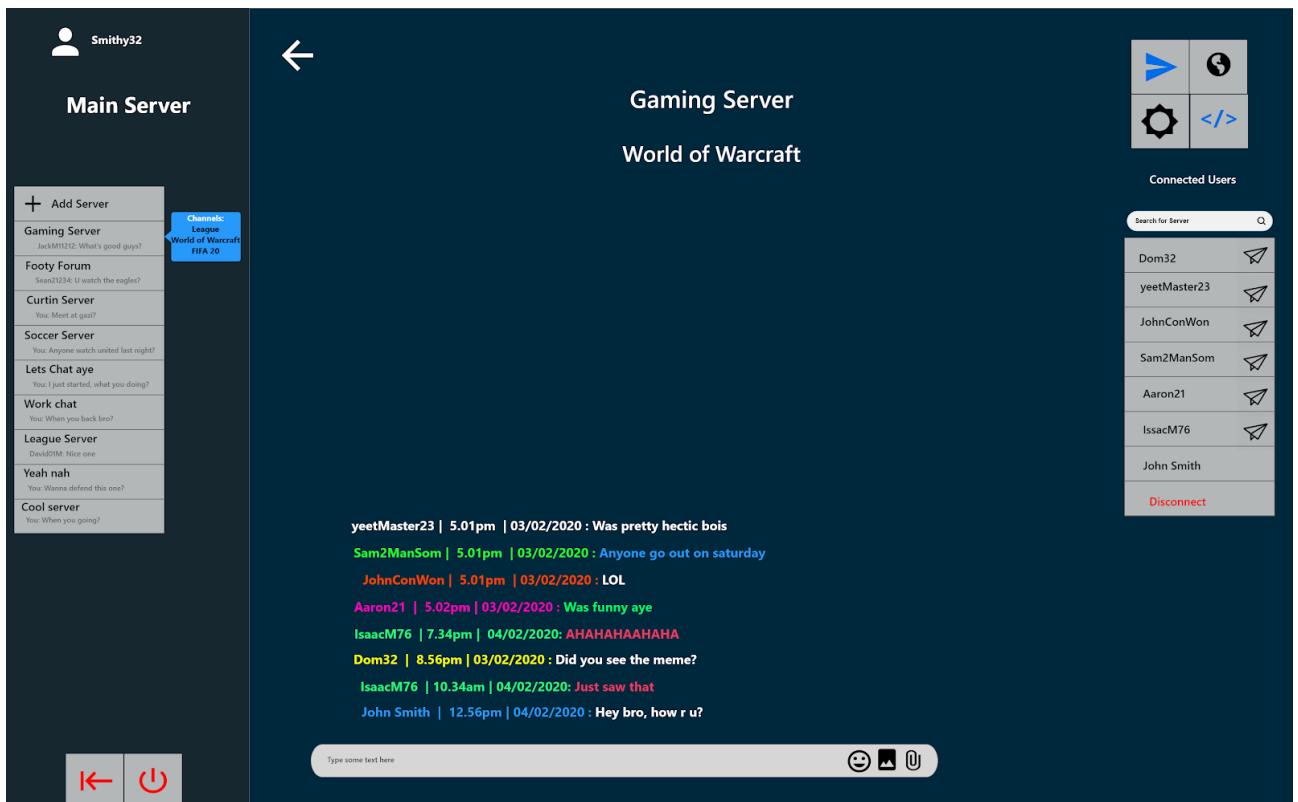


Figure 8 : Main Server Page

The below image is the prototype for the Main Server page of WeeChat. Many users are able to connect to a server by entering a valid URL in the Add Server interface ([Figure 6](#)). The server name that the user enters in the Add Server page will be displayed in the left panel. All users will be able to communicate using the text bar at the bottom. The links at the end of the text bar allow the users to attach images and files, as well as emojis. Each username displayed in the chat has a different color, which is chosen by the user on login. On the right hand side, there is a list of all users who are connected to the current server. If the user clicks the send symbol next to the user's name, they will be able to direct message that user, and will be redirected to the Direct Messages interface ([Figure 7](#)). Furthermore, each server may have various channels. The example is displayed when the user hovers over a server name and the associated channels appear in a blue box. Each of these channels are different chats that correspond with the server and the user can connect to.



Aaron Giles (19487210)

Low Fidelity

The image shows two hand-drawn wireframes side-by-side.

Create User: The title "Create User" is at the top. It features a "Weechat logo" icon, followed by input fields for "Email", "Password", "Confirm Password", and "Display name". Below these is a checkbox for "Accept terms conditions" and a "Register" button at the bottom.

Login: The title "Login" is at the top. It features a "Weechat logo" icon, followed by input fields for "Username" and "Password". There are links for "Forgot username?" and "Forgot password?". A "Stay signed in" checkbox is next to a "Sign in" button. At the bottom are links for "Create user" and "for create user".

The image shows a hand-drawn wireframe for a "Preferences" dialog box.

The title bar includes tabs for "Preferences", "Scripts", "Profile", and "Options (Preferences)".

The main content area contains a list of checkboxes:

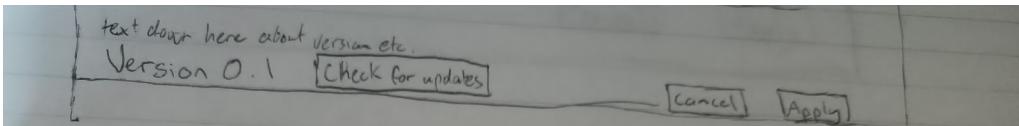
- Allow notifications
- Start with computer
- Other options?
-

Below this is a section titled "About" with placeholder text: "text about here about version etc."

At the bottom left is the text "Version 0.1" and a "Check for updates" button. At the bottom right are "Cancel" and "Apply" buttons.

On the far right, there are settings for "Theme: Default", "Font: Arial", and "Size: Small".

At the very bottom, it says "Ctrl + Alt + Shift = (client manager)"



Preferences Scripts Profile Options (Script manager)

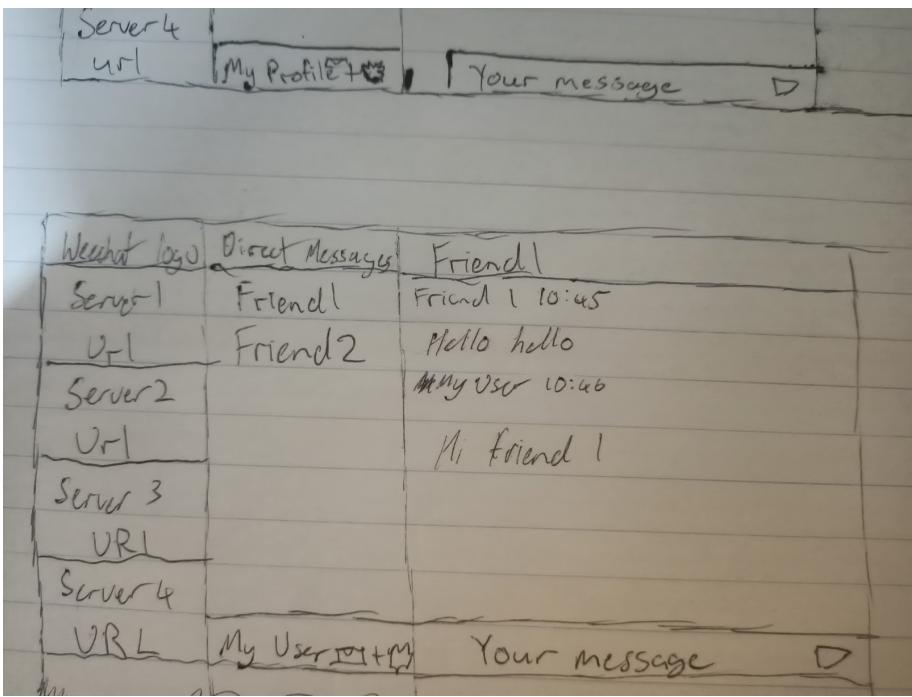
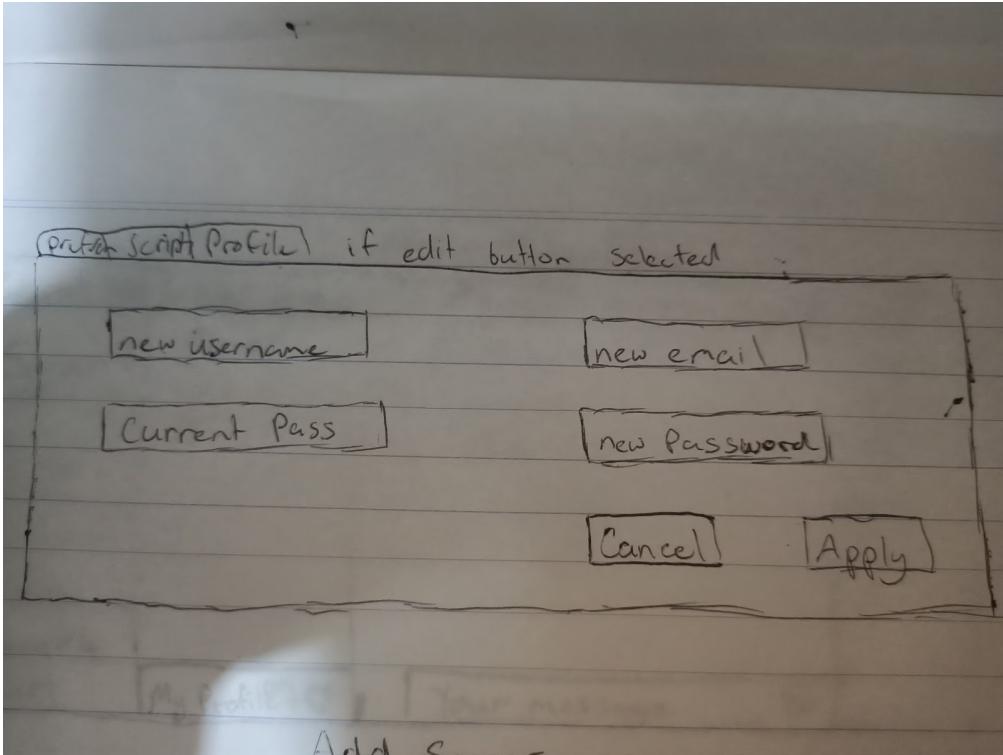
Plugin	Version	Author	Description
Plugin 1	1	Me	Really cool
Plugin 2	2	Me	good
Plugin 3	1	Me	..

Editing Script Profile Options (User Customisation)

Editing Script Profile Options (User Customisation)

Username Colour

Email



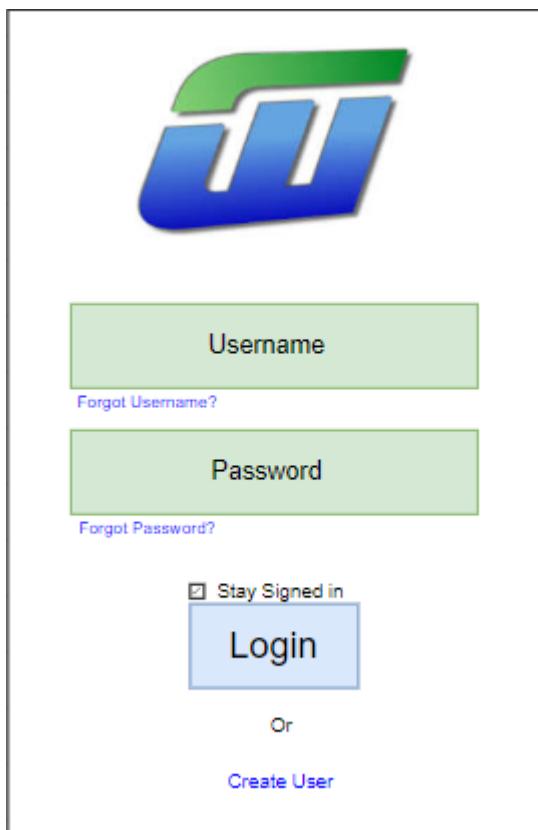
Main Interface				
Weechat logo	Selected Server	Channel name : Description	Members	
Server 1 url	Channel 1 Channel 2 Channel 3	2:43 user ¹ message 2:44 user ² message 2:55 user ³ HAHHA	User1 User2 User3 My User	
Server 2 url				
Server 3 url				
Server 4 url	My Profile	Your message		

High Fidelity

Login Page

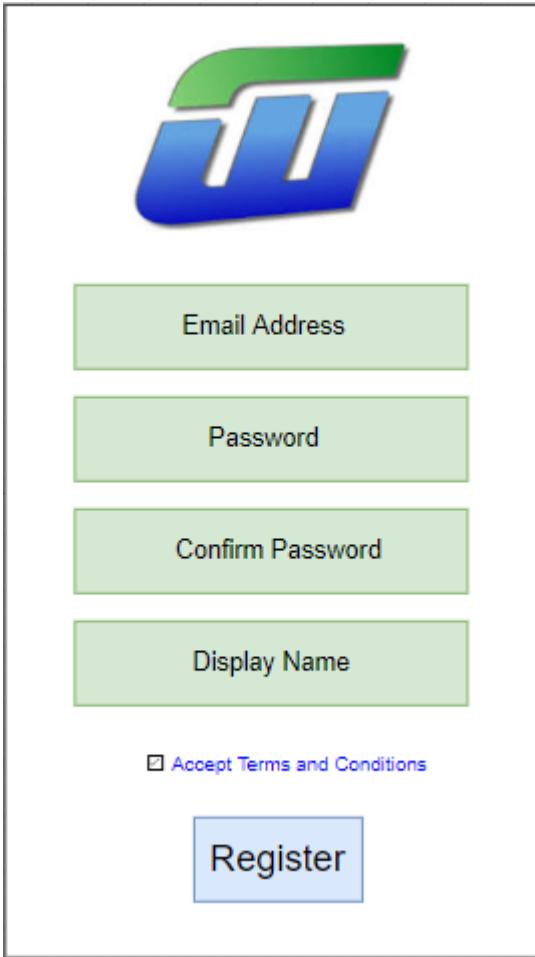
The idea was to be as simple as possible to avoid as much confusion as possible.

With large buttons and only two input boxes, and an option to stay signed in. Options to be redirected to some sort of password/username recovery system, external from our GUI. This will be promoted whenever the application is opened.



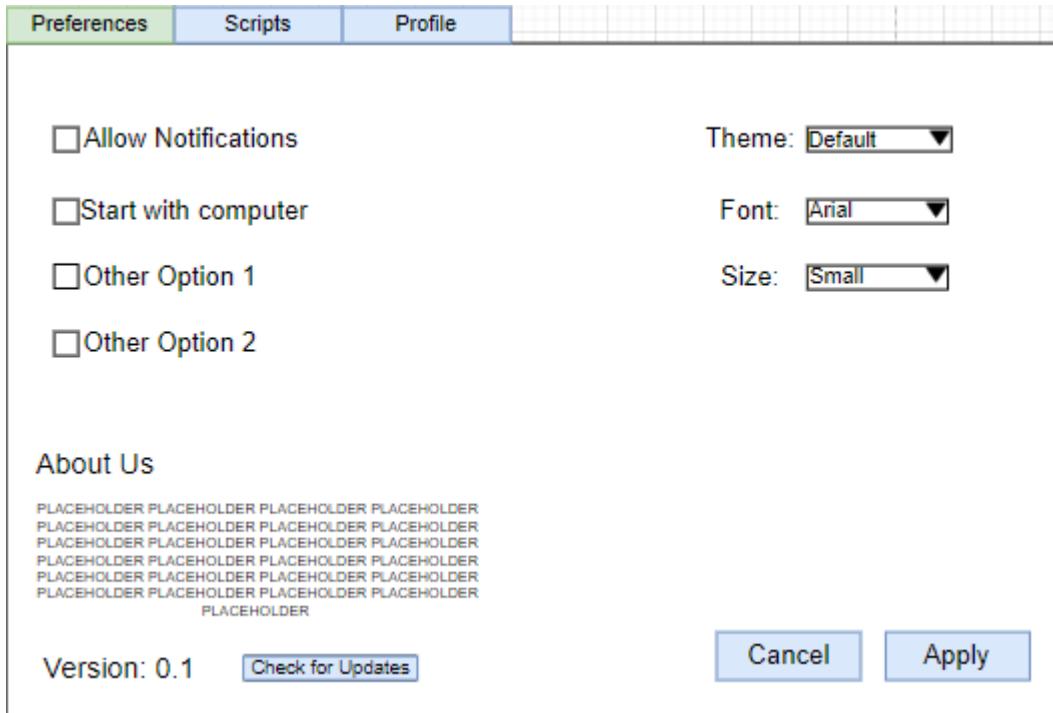
Register

Whenever the "Create User" option is selected on the login page you will be redirected to the register page to create an account. Only the essential information is required here, further info can be added later to the profile page etc... Not much on screen to avoid confusion. If the terms and conditions are clicked you are automatically redirected to a page on the website stating the terms and conditions etc...



Options(Preferences)

Pop out screen given when clicked on the settings button down near the my user part of the main interface. From here all necessary customastions for looking and feeling of the application can be changed. With options to change font, change theme (yet to be designed but for future), can allow notifications and other various options, yet to be decided. "About us" to describe what we are all about us and why we developed this(currently placeholder text), and a place to display what version of Weechat app you are currently on and to check if there are any updates available.



Options (Script Manager)

Pop out screen given when clicked on the settings button down near the my user part of the main interface. This is the part where all the scripts/plugins for the application can be managed, installing and removing etc... The box next to name should be ticked to determine actions. The name, version, author and brief description are all on display for easy and quick decision making. There is a search function which will search through the names of plugins until a match is found. Exit button at the bottom to exit once everything is complete.

Plugin Name	Version	Author	Description
<input type="checkbox"/> Plugin 1	1	Me	Basic Plugin that does action 1
<input type="checkbox"/> Plugin 2	2	Me	Basic Plugin that does action 3
<input type="checkbox"/> Plugin 3	3	Me	Basic Plugin that does action 3
<input type="checkbox"/> Plugin 4	1	Me	Basic Plugin that does action 4 but longer text t...
<input type="checkbox"/> Plugin 5	1.2	Me	Basic Plugin that does action 1
<input type="checkbox"/> Plugin 6	1.01	Me	Basic Plugin that does action 1
<input type="checkbox"/> Plugin 7	1	Me	Basic Plugin that does action 1

Search Remove Install Exit

Options (Profile)

Pop out screen given when clicked on the settings button down near the my user part of the main interface. This interface allows users to view and change their profile data. There is not much to this, but it allows for future expansions if more options become available. The colour of the name can be changed and email and username are displayed, not for changing until a further action is taken. To edit data, the “Edit” button is needed to be clicked, in which it will take you to the 2nd screen, where username, password and email can all be changed. With the “Apply” and “Cancel” buttons to finalise decisions.

The image displays two screenshots of a software interface, likely a mobile application, showing profile management screens.

Top Screenshot:

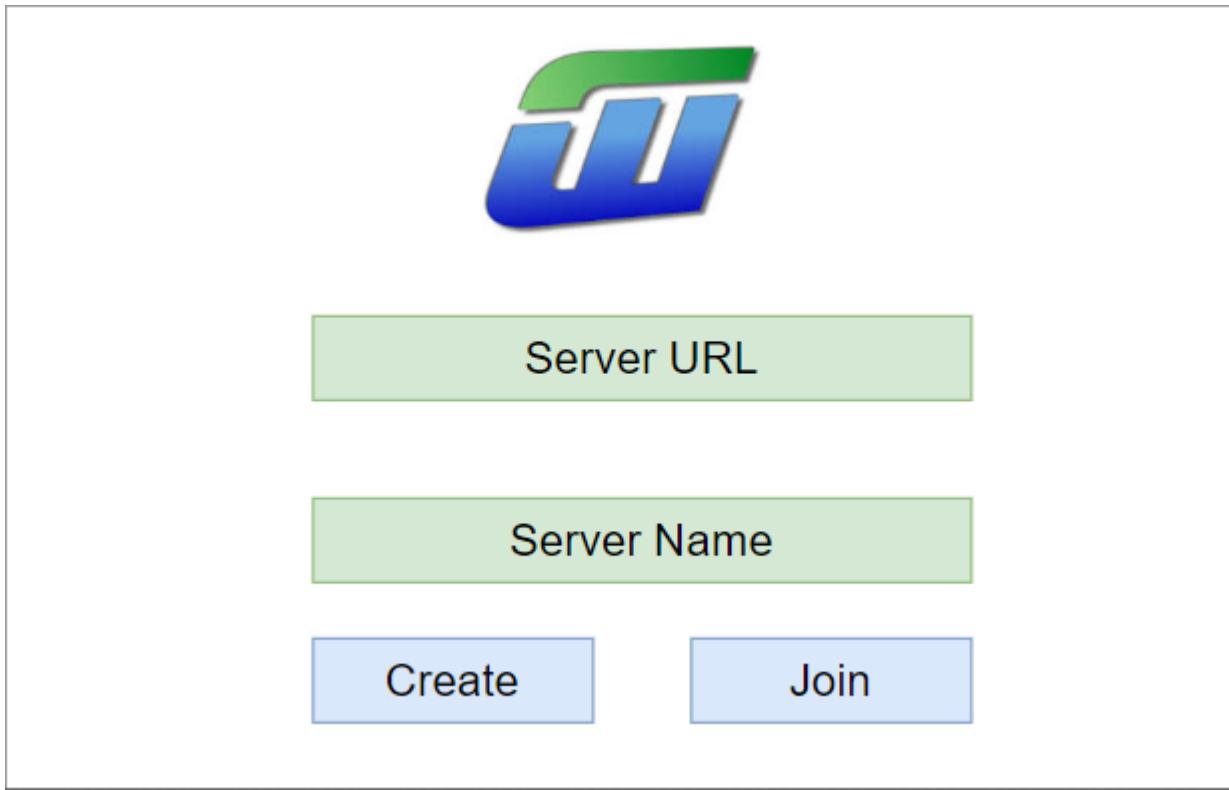
- At the top, a navigation bar with tabs: Preferences, Scripts, and Profile (selected).
- The main area contains a "Username" field with the value "fake@email.com".
- A "Colour" dropdown menu is set to "Red".
- At the bottom right are "Edit Profile" and "Exit" buttons.

Bottom Screenshot:

- At the top, a navigation bar with tabs: Preferences, Scripts, and Profile (selected).
- The main area contains four input fields arranged in a 2x2 grid:
 - New Username
 - New Email
 - Current Password
 - New Password
- At the bottom right are "Cancel" and "Apply" buttons.

Add Server

Pop out screen given when clicked on the plus icon down near the my user part of the main interface. This interface is simple and just allows users to add a server given the url, or create a server with the url (not sure how this would work in practice). The option to add a name is what the user will see on their side under their server lists. Assumption that there is an exit button(like default windows) so that it can be exited.



Main Interface

This is the screen that when logged in and application is running will be the main screen. From this interface, all settings, messages and add server functions can be accessed. On the right is the server lists on which servers you have added, and what you have named them, and displaying their URL underneath. The selected server/channel is highlighted green and others blue. Next is channel lists for the server, ontop displaying which server selected. Furthermore, when the channel is selected the text will display for channel and on top a brief description of the channel and name. On the far left a list of members, displaying everyone in the server. Each name is coloured based on what that user has chosen in their menu. All messages are time stamped. If the message is larger than usual it will be given a large box for the message.

Server 1 11.11.11.11	Server 1	Channel 1: Use this channel to talk about things relevant to Channel 1	Members
	Channel 1	User 1 2:43 pm 19/04/2020 User 1's message about channel 1	User 1
	Channel 2	User 2 2:44 pm 19/04/2020 User 2's response about channel 1	User 2
	Channel 3	User 3 2:46 pm 19/04/2020 User 3's input about channel 1	User 3
	Channel 4	User 1 2:53 pm 19/04/2020 User 1's really long message about message about channel 1	My User
Server 2 22.22.22.22	Channel 5		
Server 3 33.33.33.33			
Server 4 44.44.44.44			
	My User  + 	Your message... 	

Direct Messages

This can be accessed from the main interface by clicking on the mail icon down in the profile section. The interface is similar to the main interface however lacking a members list. Instead of a channels list it is instead just different friends and people you share messages with. The idea is similar to server chats, but just between two people. All servers can be accessed here and if selected will redirect you back to the main interface. All settings and Add server can be accessed from it.

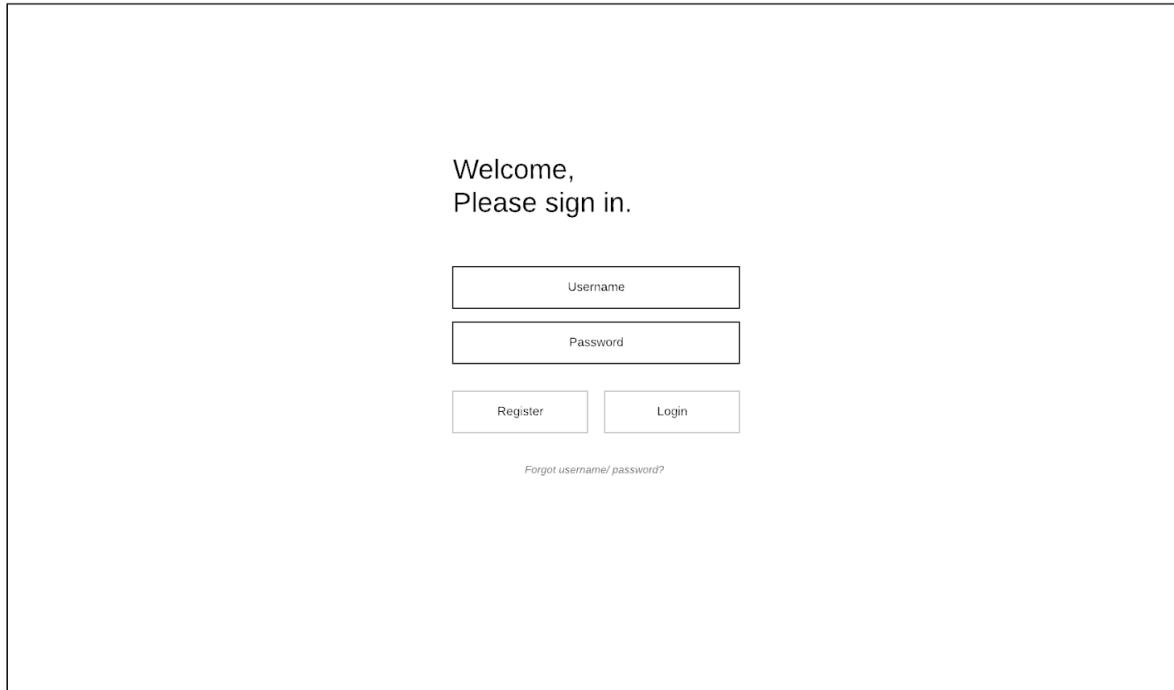
	Direct Messages	Friend 1	
		Friend 1	Friend 2
Server 1 11.11.11.11	Friend 1	Friend 1 3:43 pm 19/04/2020 Hello there My User! Glad to meet you	
		My User 3:44 pm 19/04/2020 Hello Friend 1, nice to meet you too	
	Friend 2	Friend 1 3:48 pm 19/04/2020 really good convo we having here teehee	
		Friend 1 3:53 pm 19/04/2020 Where did you go? ive been waiting for you response? Hello i dont mean to spam chat but hello? you ok buddy? dont seen me	
	Friend 3	My User 4:58 pm 19/04/2020 Yep back	
Server 2 22.22.22.22			
Friend 4			
Server 3 33.33.33.33	Friend 5		
Server 4 44.44.44.44			
My User   		Your message... 	

Kevin Le (19472960)

Low Fidelity

My low fidelity was designed trying to keep everything simple, and following the styles of many popular, well established programs. This helps ensure the user experience is excellent, as they can recall their previous use with other similar multilayered applications in order to learn how to use this application.

Login



Everything located in the centre to capture users' gaze and attention.

Register

Username

Email

Password

Password Confirm

Username colour: #00BFFF

I accept the terms and conditions

Register

Input fields ordered in a way that the user is expected to fill it in, located in the centre.

Servers

Server 1	#main
Server 2	#study
Server Folder 1	#engineering
Server 4	#computing
Server 5	#3dprinting
	#memes
DM	
+	

User 1 - 9:42AM 24/3/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

User 2 - 9:45AM 24/3/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

User 4 - 9:49AM 24/3/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

User 2 - 9:45AM 24/3/20
@User 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

User 1 - 9:42AM 24/3/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

User 1 - 9:42AM 24/3/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

User 2 - 9:45AM 24/3/20
Quam elementum pulvinar etiam non quam. Sed vulputate mi sit amet mauris commodo quis imperdiet massa. Pharetra diam sit amet nisl suscipit adipiscing bibendum.

Connected to:
irc.curtin.edu.au
as: User1

Users online:

- Tillie
- Meredez
- Hedwig
- Gus
- Dong
- Winfred
- Felecia
- Marianna
- Rachelle
- Stella
- Lenora
- Maricela
- Barton
- Markita
- Noelle
- Elisha
- Siobhan
- Chere
- Tran
- Debra
- Rebecca
- Kena
- Jose
- Hermine
- Lasandra
- Jennie
- Marita
- Raven
- Marylou
- Madelaine

The server UI was designed in order to follow standard drop down/ cascading menu styles. Starting from a server, which is a large community of people, down to channel, which is a topic *in* a server to be discussed. The server and channel buttons were made relatively large and visible, so they could not be missed or misclicked.

So it made sense to put servers, then channels. Following traditional IRC design, the text would be sent all aligned to the left (in non right hand reading regions), as unlike popular messaging applications such as Facebook messenger which aligns text based on senders.

The 2x2 grid of icons at the bottom left are the menu options for the program, “DM” ought to be replaced with a message icon, as it is more commonly associated with “Messaging”. The “person” icon was chosen for the profile settings page as it is often associated with profile, or themselves. The “+” icon was chosen as it is associated with “Add” which leads to “Add server” functionality, when the user is first exploring the program, they will likely try that button to add a server, and learn that it does. The “settings” icon is a symbol with multiple sliders, indicating “control”, hence why it was chosen.

Direct Messages

The screenshot shows a messaging interface titled "Direct Messages". On the left, there is a sidebar with a back arrow icon and a list of users: User 1, User 3, and User 4. The main area displays messages from User 1 and User 2. Each message includes the sender's name, timestamp, and a placeholder text block. At the bottom right, there is a large input field and a "Send" button.

User	Message
User 1	User 1 - 9:42AM 24/3/20 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
User 2	User 2 - 9:45AM 24/3/20 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
User 3	User 1 - 9:49AM 24/3/20 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
User 4	User 2 - 9:45AM 24/3/20 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
	User 1 - 9:42AM 24/3/20 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
	User 2 - 9:45AM 24/3/20 Quam elementum pulvinar etiam non quam. Sed vulputate mi sit amet mauris commodo quis imperdiet massa. Pharetra diam sit amet nisl suscipit adipiscing bibendum.

Follows the same style as server messaging UI.

Add Server

←

Add an IRC server

URL	Add
-----	-----

The goal of the add server interface was to keep it as simple as possible, upon first glance it was expected the user understands what to do. Where the URL is to be typed/ pasted into the text field and then click "Add".

Profile

← Profile

user1
you@email.com

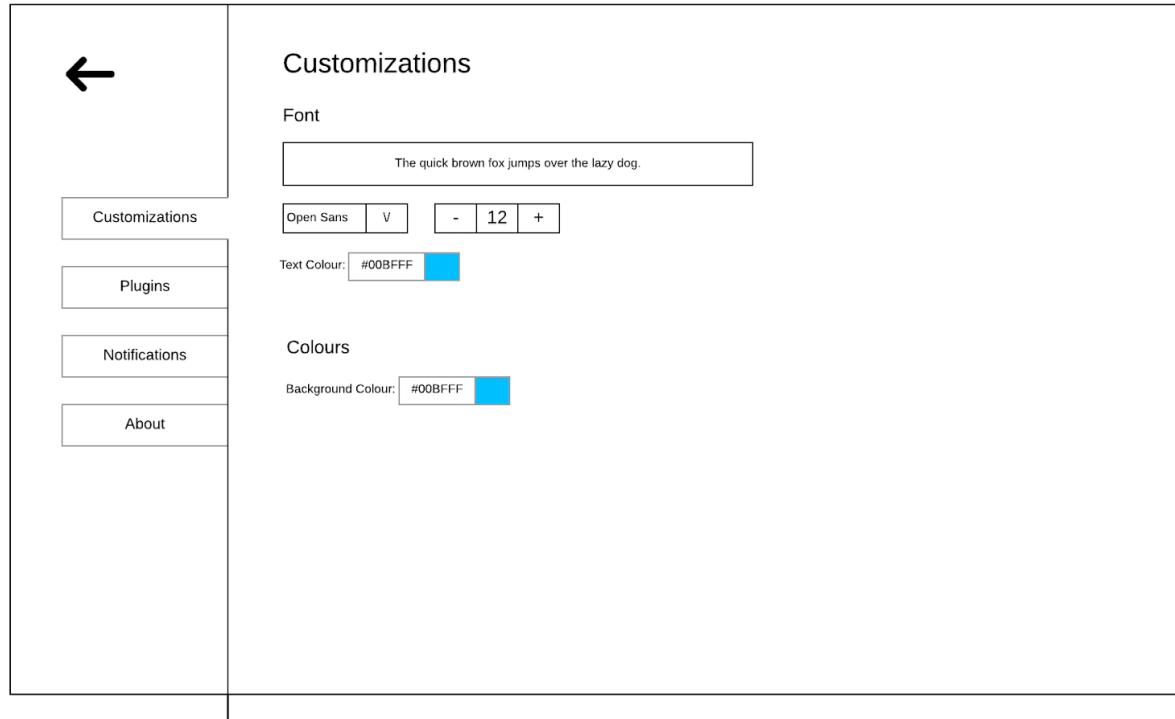
Old Password
New Password
New Password Confirm

Username colour: #00BFFF

Save	Delete Account
	Logout

The profile page continued the simple design and presented text fields in a hierarchical order that makes sense.

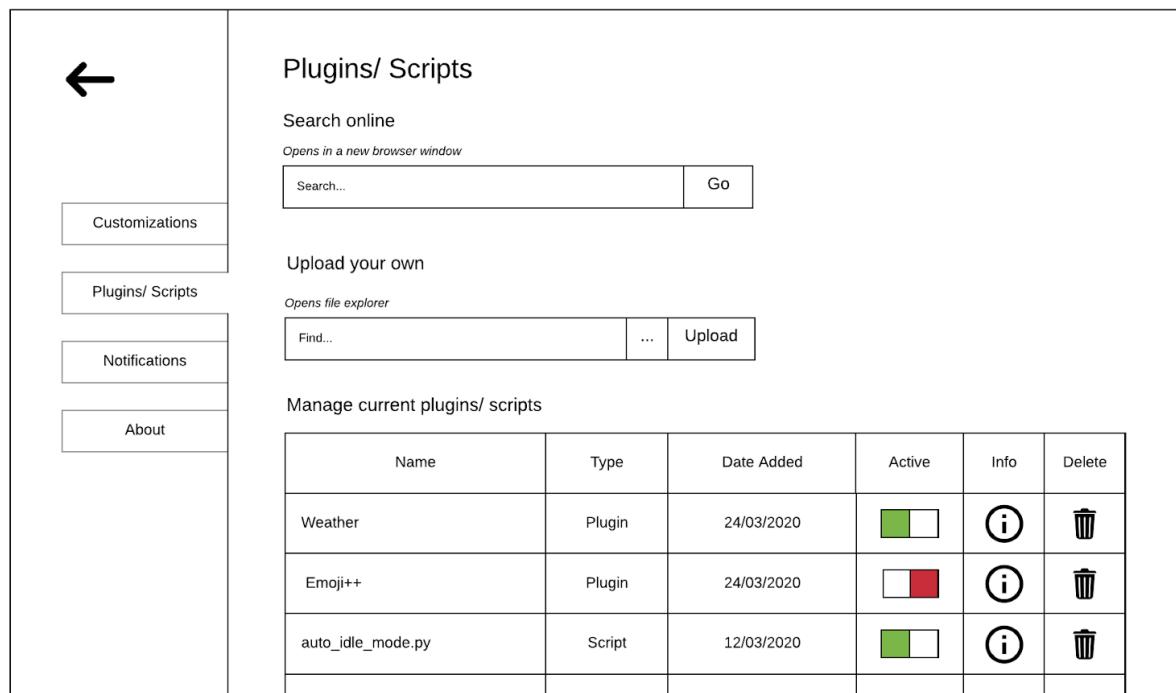
Options - Customizations



Options page with “tabs”, like a multi layered binder, intuitive as users can identify what sub options the options page has, back button on the top left as many other programs also use that location.

Customizations has a dropdown menu for font list selection, and decrementer/ incrementer for font size.

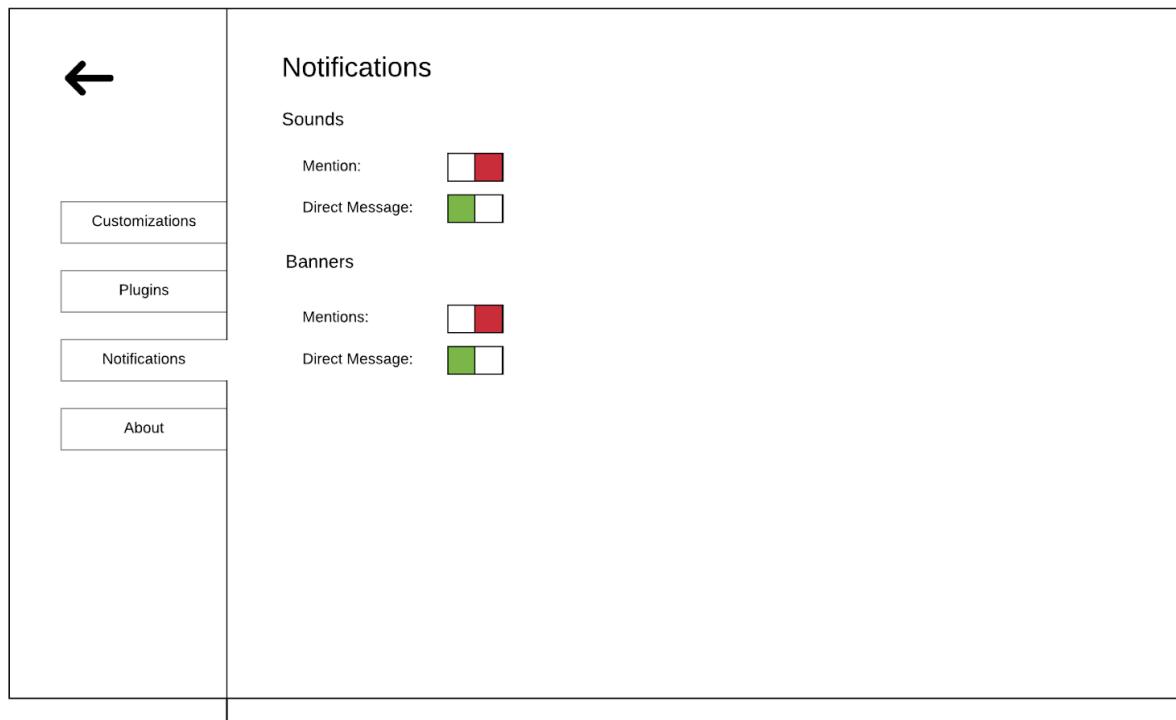
Options - Plugins



Sub options are indicated using hierarchical font structure, large for title, medium for sub title and normal for any additional text. Switches were chosen to activate/ deactivate plugins as they are

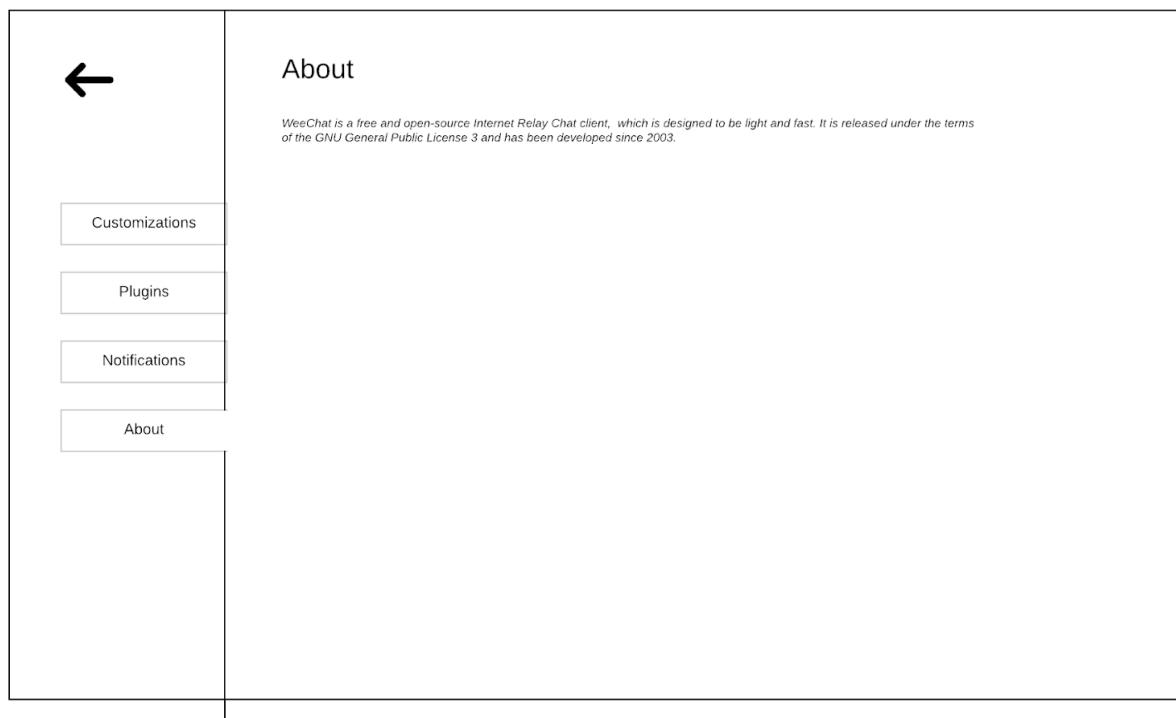
intuitively associated with on and off. The information icon was chosen as it is quite a universal symbol for information, and the bin icon also.

Options - Notifications



Switches chosen for the same reasons outlined above

Options - About



High Fidelity

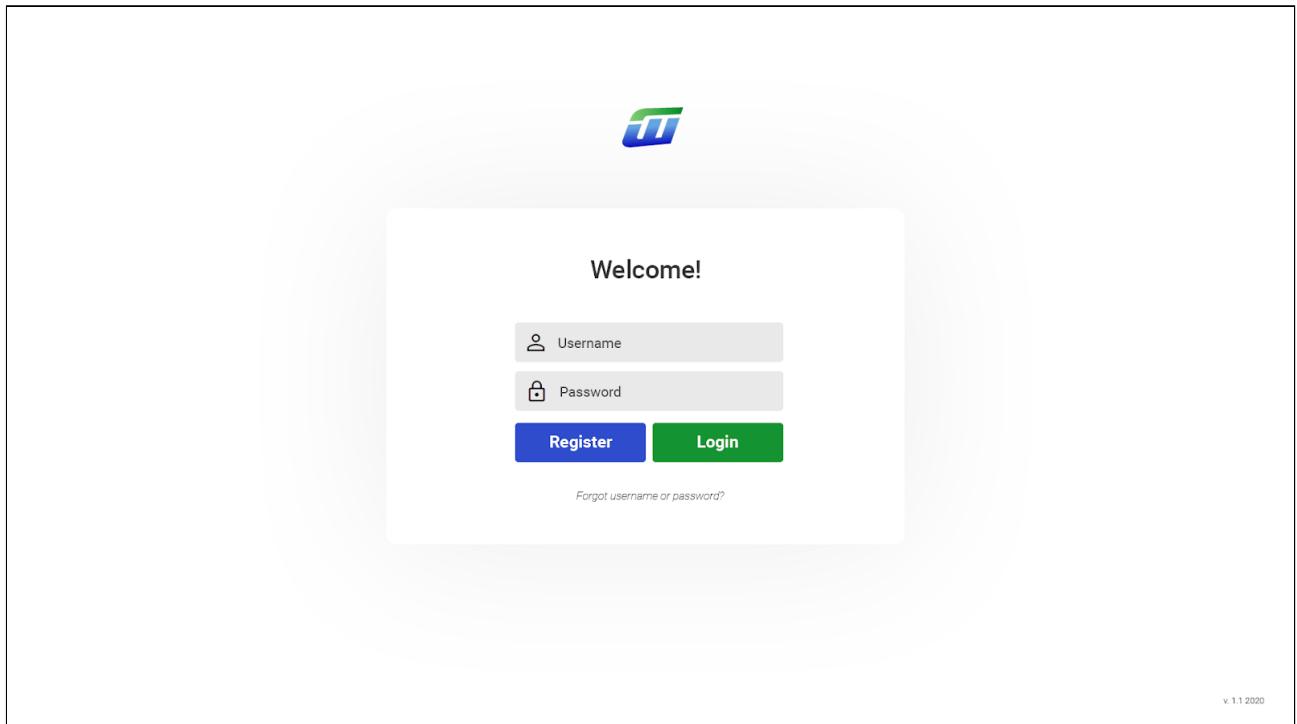
In the development of the program, I utilized the main colour themes derived from the Weechat logo, this is to create a brand identity for the program that is recognizable. The buttons that proceed to the next step aim to attract the users attention, and hence are marked with green, associated with "Success".

The entire program follows a rounded design, where all elements have sharp corners eliminated, this is done to create a more organic and friendly interface. Text fields, buttons, checkboxes for example are all rounded at roughly 15px.

Each button has attributes on hover and on click, when the user moves their cursor to hover over the button, a border is shown that is of darker colour than the button, this is to provide visual feedback to what action the user will execute on click. When the user clicks the button, the entire button will "indent", to simulate an actual click of a button in real life. This responsive behaviour allows the user to determine what actions will or have taken place.

Text fields have text prompts and icons to indicate what purpose the field serves. E.g. an email field will have the prompt text "Email" and an icon that looks like a letter to symbolize and enhance the meaning of the field, as to avoid confusion.

Login



Upon starting the program, the login screen will be presented if there is no existing session. On successful login, the server UI will be presented. Everything is centred to attract the users attention, with a light grey shadow guiding their gaze to the centre.

Register



Register

Username:

Email:

Password:

Password Confirm:

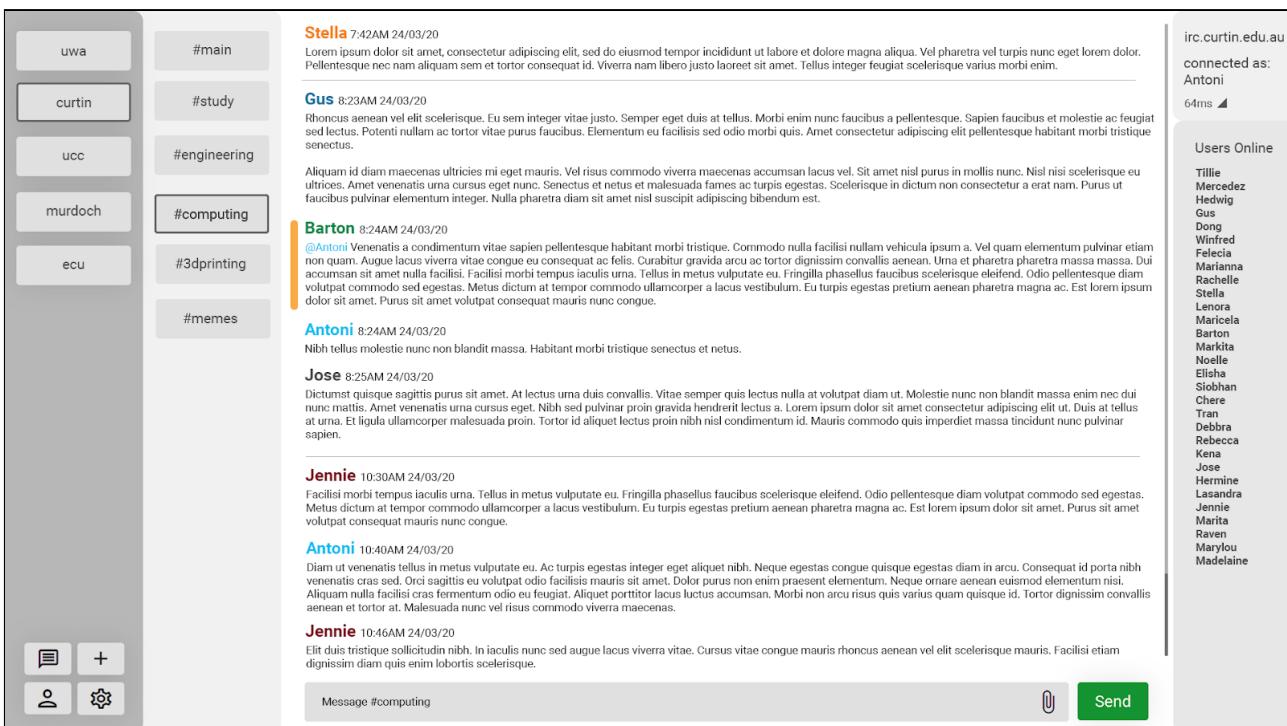
Username Colour: #149332

Accept the Terms & Conditions

Register

Register screen is presented when the “Register” button on the Welcome/ Login screen is clicked. Upon filling in the fields and clicking register, they will be returned back to the Login screen to enter their new login. This form format was chosen as it is a standard format seen on many register pages.

Servers



The interface shows a sidebar with server names: uwa, curtin, ucc, murdoch, and ecu. Below them are channels: #main, #study, #engineering, #computing, #3dprinting, and #memes. At the bottom are icons for a list, a plus sign, a user, and a gear.

Stella 7:42AM 24/03/20
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut pharetra vel turpis nunc eget lorem dolor. Pellentesque nec nam aliquam sem et tortor consequat id. Viverra nam libero justo laoreet sit amet. Tellus integer feugiat scelerisque varius morbi enim.

Gus 8:23AM 24/03/20
Rhoncus aenean vel elit scelerisque. Eu sem integer vitae justo. Semper eget duis at tellus. Morbi enim nunc faucibus a pellentesque. Sapien faucibus et molestie ac feugiat sed lectus. Potenti nullam ac tortor vitae purus faucibus. Elementum eu facilisis sed odio morbi quis. Amet consectetur adipiscing elit pellentesque habitant morbi tristique senectus.

Alliquam id diam maeccenas ultricies mi eget mauris. Vel risus commodo viverra maeccenas accumsan lacus vel. Sit amet nisl purus in mollis nunc. Nisi nisi scelerisque eu ultrices. Amet venenatis urna cursus eget nunc. Senectus et netus et malesuada fames ac turpis egestas. Scelerisque in dictum non consectetur a erat nam. Purus ut faucibus pulvinar elementum integer. Nulla pharetra diam sit amet nisl suscipit adipiscing bibendum est.

Barton 8:24AM 24/03/20
@Antoni Venenatis a condimentum vitae sapien pellentesque habitant morbi tristique. Commodo nulla facilisi nullam vehicula ipsum a. Vel quam elementum pulvinar etiam non quam. Augue lacus viverra vitae congue eu consequat ac felis. Curabitur gravida arcu ac tortor dignissim convallis aenean. Urna et pharetra pharetra massa massa. Dui accumsan sit amet nulla facilisi. Facilisi morbi tempus iaculis urna. Tellus in metus vulputate eu. Fringilla phasellus faucibus scelerisque eleifend. Odio pellentesque diam volutpat commodo sed egestas. Metus dictum at tempor commodo ullamcorper a lacus vestibulum. Eu turpis egestas pretium aenean pharetra magna ac. Est lorem ipsum dolor sit amet. Purus sit amet volutpat consequat mauris nunc congue.

Antoni 8:24AM 24/03/20
Nih tellus molestie nunc non blandit massa. Habitant morbi tristique senectus et netus.

Jose 8:25AM 24/03/20
Dictumst quisque sagittis purus sit amet. At lectus urna duis convallis. Vitae semper quis lectus nulla at volutpat diam ut. Molestie nunc non blandit massa enim nec dui nunc mattis. Amet venenatis urna cursus eget. Nih sed pulvinar proin gravida hendrerit lectus a. Lorem ipsum dolor sit amet consectetur adipiscing elit ut. Duis at tellus at urna. Et ligula ullamcorper malesuada proin. Tortor id aliquet lectus proin nih nisl condimentum id. Mauris commodo quis imperdiet massa tincidunt nunc pulvinar sapien.

Jennie 10:30AM 24/03/20
Facilisi morbi tempus iaculis urna. Tellus in metus vulputate eu. Fringilla phasellus faucibus scelerisque eleifend. Odio pellentesque diam volutpat commodo sed egestas. Metus dictum at tempor commodo ullamcorper a lacus vestibulum. Eu turpis egestas pretium aenean pharetra magna ac. Est lorem ipsum dolor sit amet. Purus sit amet volutpat consequat mauris nunc congue.

Antoni 10:40AM 24/03/20
Diam ut venenatis tellus in metus vulputate eu. Ac turpis egestas integer eget aliquet nibh. Neque egestas congue quisque egestas diam in arcu. Consequat id porta nibh venenatis cras sed. Orci sagittis eu volutpat odio facilisis mauris sit amet. Dolor purus non enim praesent elementum. Neque ornare aenean euismod elementum nisi. Aliquam nulla facilisi cras fermentum odio eu feugiat. Aliquet porttitor lacus luctus accumsan. Morbi non arcu risus quis varius quam quisque id. Tortor dignissim convallis aenean et tortor at. Malesuada nunc vel risus commodo viverra maeccenas.

Jennie 10:46AM 24/03/20
Elit duis tristique sollicitudin nibh. In iaculis nunc sed augue lacus viverra vitae. Cursus vitae congue mauris rhoncus aenean vel elit scelerisque mauris. Facilisi etiam dignissim diam quis enim lobortis scelerisque.

Message #computing Send

irc.curtin.edu.au
connected as: Antoni
64ms

Users Online
Tillie
Mercedez
Hedwig
Gus
Dong
Winfred
Felecia
Marianna
Rachelle
Stella
Lenora
Maricela
Barton
Markita
Nelle
Elishe
Siobhan
Chere
Tran
Debra
Rebecca
Kena
Jose
Hermine
Lasandra
Jennie
Marita
Raven
Marylou
Madelaine

The server interface contains all the messaging and is first presented on start up if the user has an existing session. This contains buttons in the bottom left leading to, direct messages, server adding, options, and profile. The icons chosen for each button was done by researching what existing programs used to lead to similar functionalities. The message icon made the most sense for Direct Messages, the “+” for Adding a server, the “Person” icon for profile, and the gear cog for settings/ options.

The sidebar closest to the left contains the servers, and the 2nd sidebar from the left contains the channels in the selected server. The main white section contains the messages from various users, and the right sidebar shows the connection information, and members in the session. Personal connection information is shown in the square on the top right, and the list of users are shown below.

This setup draws a natural path with the users, in terms of categories from Servers which contain channels, larger to smaller.

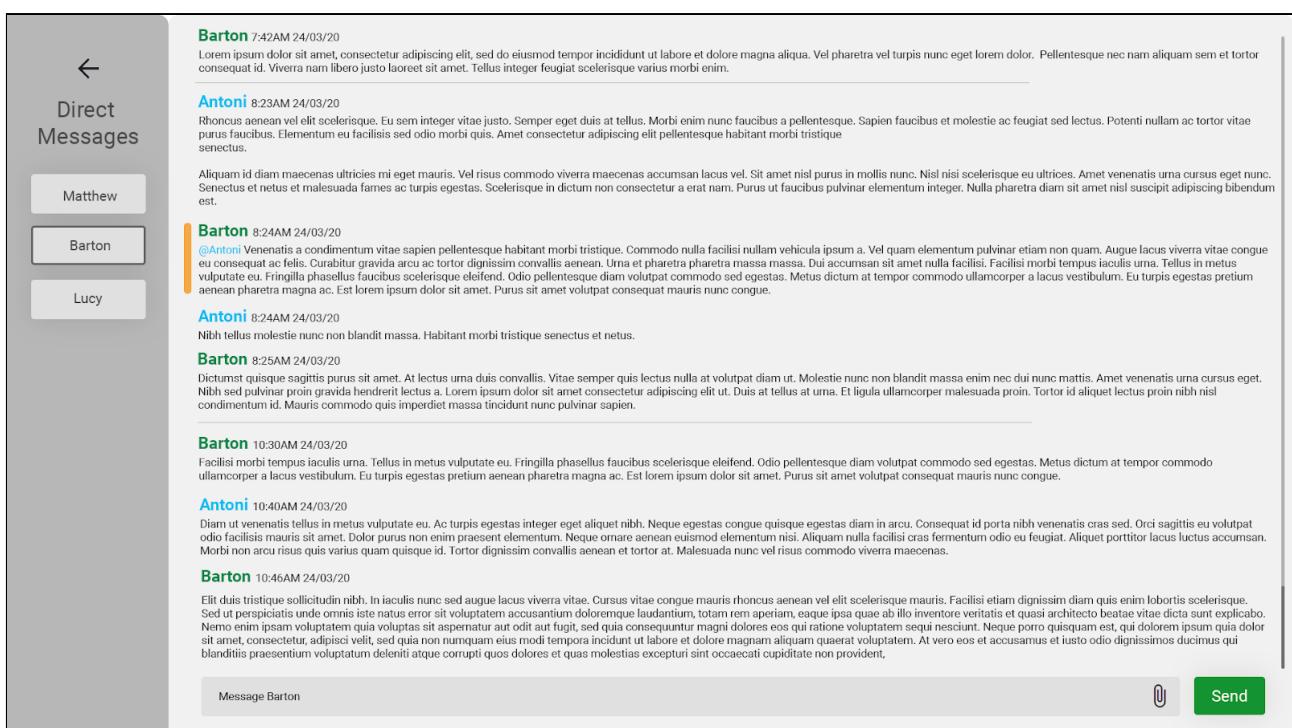
Upon clicking on a server, the server button will be highlighted with a border, indicating the selection, this also applies for the channel selection.

Messages which mention the current user will be indicated with an orange bar for the length of the message. Orange was chosen as it is often associated with warning/ alerts.

All buttons on hovering over, a border will be shown to indicate current cursor position.

Sending a message is done by clicking the send button. The paper clip allows the user to attach files when clicked, this icon was chosen as the paperclip is often associated with “attaching” as it has meanings of attaching the real world, it opens the default file explorer of the OS.

Direct Messages



This interface is shown when the “message” button in the server UI is clicked. Contains individual one to one messages. Left sidebar displays all the users the current user has communicated with and the currently selected user with a grey border. It utilizes the same messaging UI as servers (above). The back arrow will return them back to the server UI.



Add Server



A screenshot of a mobile application interface titled "Add Server". At the top left is a back arrow. Below it is a text input field with a placeholder "Add an IRC server" and a globe icon labeled "URL". To the right of the URL field is a green button with a white right-pointing arrow. The background is white with a thin black border around the main content area.

This interface is shown when the “+” button in the server UI is clicked. Contains an input field for URL and an arrow pointing to the right, indicating continue, which will return back to the server UI with the new server (if valid) added. Arrows pointing right symbolize going forward, whilst left symbolizes going back. A globe is used to support the meaning of the text field, as a globe is associated with the WWW, which an irc url is.



Profile

The screenshot shows a profile editing interface. At the top left is a back arrow and the word "Profile". At the top right is a red "Logout" button. Below these are two input fields: "Username" (with placeholder icon) and "Email" (with placeholder icon). Underneath the email field is a "Username Colour" section with a color swatch labeled "#149332". Below this are three password-related fields: "Old Password", "New Password", and "New Password Confirm", each with a lock icon. At the bottom center is a blue "Save" button, and to its right is a grey "Delete Account" button.

This interface is shown when the “person” button in the server UI is clicked. Contains the currently logged in users information allowing them to modify. Upon finishing editing, the save button will save the updated information (if valid), and update the information. The back arrow will return them back to the server UI, indicated by an arrow.



The interface also has the option to logout or delete their account. Which will return them back to the login screen. The logout button was set to red as it indicates an action the user may take that is destructive, as the session is destroyed, and they will have to login again. This also draws attention, so if they are about to click it, they would notice it.

The delete account button is grey in order to not attract attention, if the user wanted to delete their account, it had to be a deliberate action. It was also separated far away from any other buttons to avoid misclicks. Previously was next to the logout button in the low fidelity. Text fields are again supported with text prompts and iconography. With the user details being separated from the password fields.

Options - Customization



Options menu is presented when the user clicks on the settings cog in the main server chat screen.



The tabbed design seen in the low fidelity prototypes was forfeited for simple buttons. This was done to have a consistent selection style, as seen in the server UI with the server and channel buttons, and the direct messages users.

Subcategories are sorted hierarchically with font sizing for the subtitles to indicate what the option belongs to. E.g. "Customizations" title is large font, "Font" subcategory is medium sized font and any other text is normal sized.

Sub menus are presented when the user clicks on the sub menu button in the left sidebar. Which subsequently highlights the current selection they are on, with a grey border and shows the specific options in the right panel.

The back arrow will return the user back to the server UI. As an arrow pointing left in the top left is associated with returning/ back in many other programs.

The “-” and “+” serve to indicate numerical increments used for the font, as it symbolizes increment and decrementing. The colour picker uses both text (hexadecimal colour code) and visual colour prompts to indicate the current selection, seen in colour pickers found in other programs.

Options - Plugins

The screenshot shows the 'Plugins' section of the application's options. On the left, a sidebar has buttons for 'Customizations', 'Plugins' (which is selected and highlighted in green), 'Notifications', and 'About'. The main area is titled 'Plugins/ Scripts'. It includes a 'Search Online' section with a search bar and a 'Go' button. Below it is an 'Upload Your Own' section with a 'Find...' button, an ellipsis button, and an 'Upload' button. A table lists current plugins and scripts, with columns for Name, Type, Date Added, Active (with a switch), Info, and Delete. The table entries are:

Name	Type	Date Added	Active	Info	Delete
Weather	Plugin	24/03/2020	<input checked="" type="checkbox"/>	<i>info</i>	<i>trash</i>
Emoji++	Plugin	24/03/2020	<input type="checkbox"/>	<i>info</i>	<i>trash</i>
auto_idle_mode.py	Script	12/03/2020	<input checked="" type="checkbox"/>	<i>info</i>	<i>trash</i>

At the bottom left is a logo and the text 'v. 1.1 2020'.

Green buttons symbolize “going”, so when users see it they ought to associate them with the next step. Under the “Upload your own” category, the text field has a button labeled “...” this is often associated with finding a file path, and users often expect a file explorer to be opened. Under the table listing the current scripts, it was decided a table representation would be the best approach to show the current plugins, since each entry has quite a lot of detail. Switches were used to switch between the plugin being activated and deactivated. This is also reinforced by the colours red and green, which are associated with off and on respectively in many contexts.

Options - Notifications

The screenshot shows the 'Notifications' section of the WeeChat options. On the left, a sidebar lists 'Customizations', 'Plugins', 'Notifications' (which is selected and highlighted in blue), and 'About'. The main area is titled 'Notifications' and contains two sections: 'Banner' and 'Sounds'. Under 'Banner', there are two switch controls: 'Mentions:' (red switch) and 'Direct Messages:' (green switch). Under 'Sounds', there are two speaker icon controls: 'Mentions:' (red switch) and 'Direct Messages:' (green switch). A note below the speaker icons says 'Click on the speaker icon to preview the sound'. At the bottom left is the WeeChat logo, and at the bottom center is the text 'v. 1.1 2020'.

Switches were chosen here for the same reasons as above. The speakers may not provide sufficient meaning so a textual prompt was decided to be added, to ensure that users understood the purposes.

Options - About

The screenshot shows the 'About' section of the WeeChat options. On the left, a sidebar lists 'Customizations', 'Plugins', 'Notifications' (which is selected and highlighted in blue), and 'About'. The main area is titled 'About' and contains text about WeeChat: 'WeeChat is a free and open-source Internet Relay Chat client, which is designed to be light and fast. It is released under the terms of the GNU General Public License 3 and has been developed since 2003.' Below this, a link 'Learn more at: <https://weechat.org/>' is provided. At the bottom left is the WeeChat logo, and at the bottom center is the text 'v. 1.1 2020'.

Terence Marcelo (19163785)

Low Fidelity

The image displays three separate wireframe mockups for a WeeChat application, arranged vertically.

Top Mockup: Login Screen

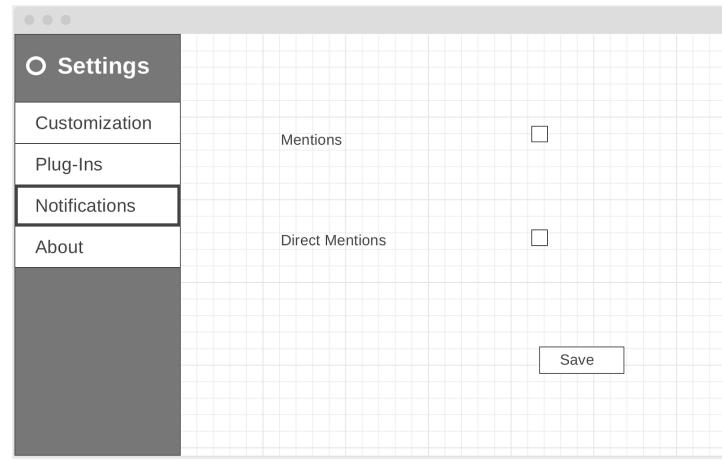
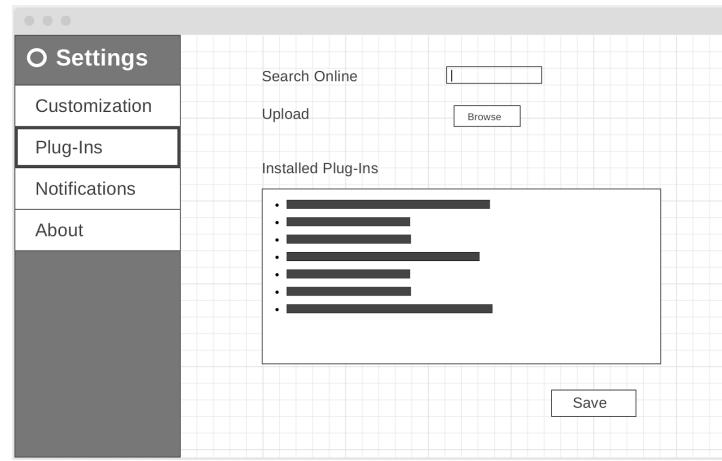
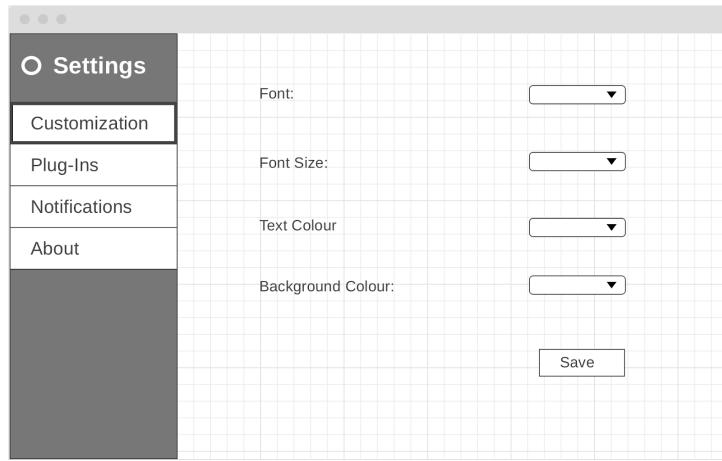
This screen is titled "Welcome to Weechat". It features fields for "User Name" and "Password", each with an associated text input field. Below these is a "Log-In" button. To the right of the password field is a "New User?" link, and below it is a "Forgot User Name/Password?" link. To the right of the "New User?" link is a "Sign-Up" button, and below it is a "Reset" button.

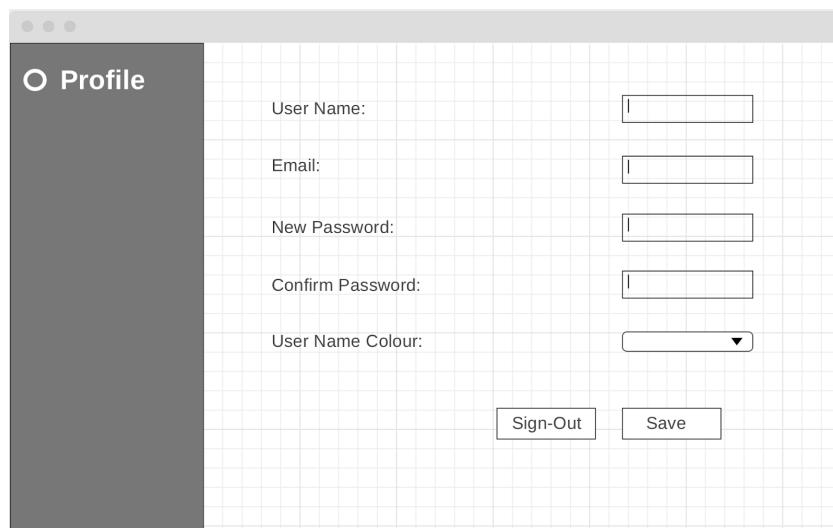
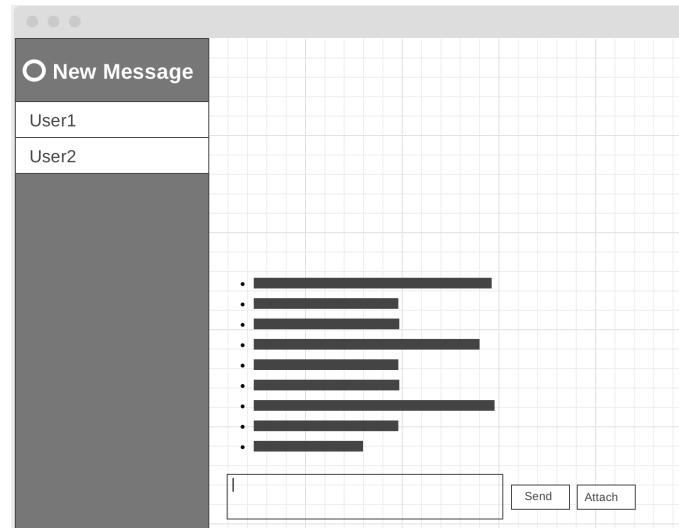
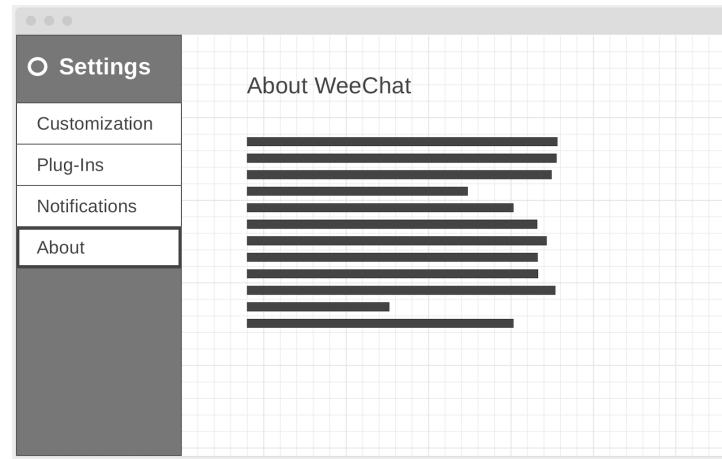
Middle Mockup: Registration Screen

This screen is titled "Register your WeeChat account". It includes fields for "User Name", "Email", "Password", "Confirm Password", and "User Name Colour" (a dropdown menu). Below these fields is a checkbox labeled "I agree to the Terms and Conditions". At the bottom are "Cancel" and "Create Account" buttons.

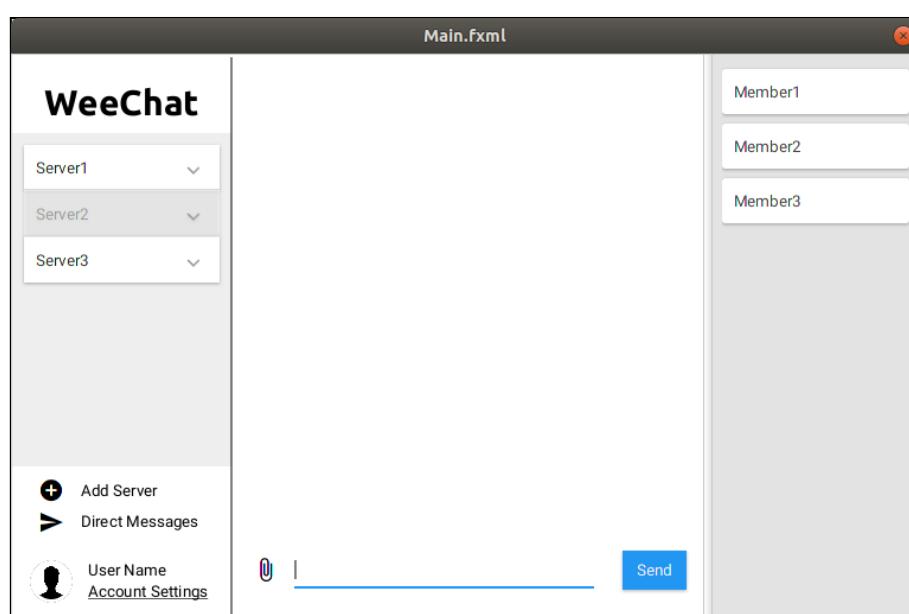
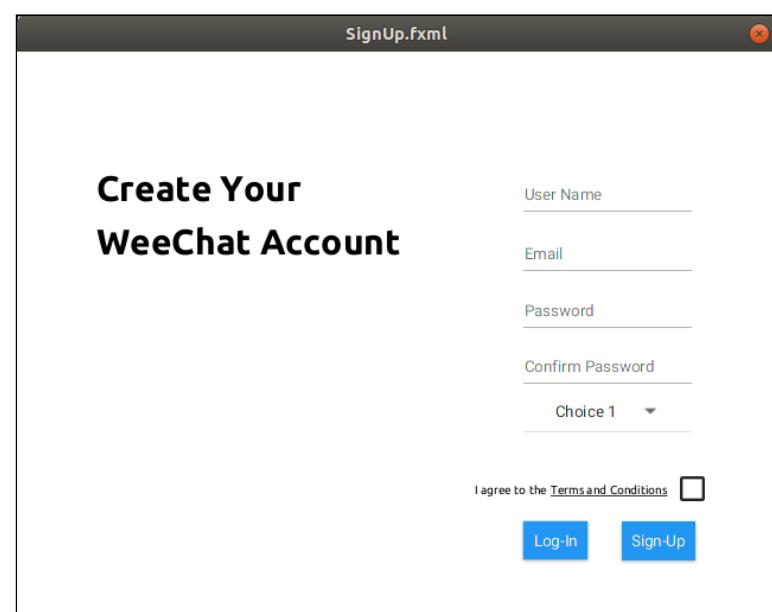
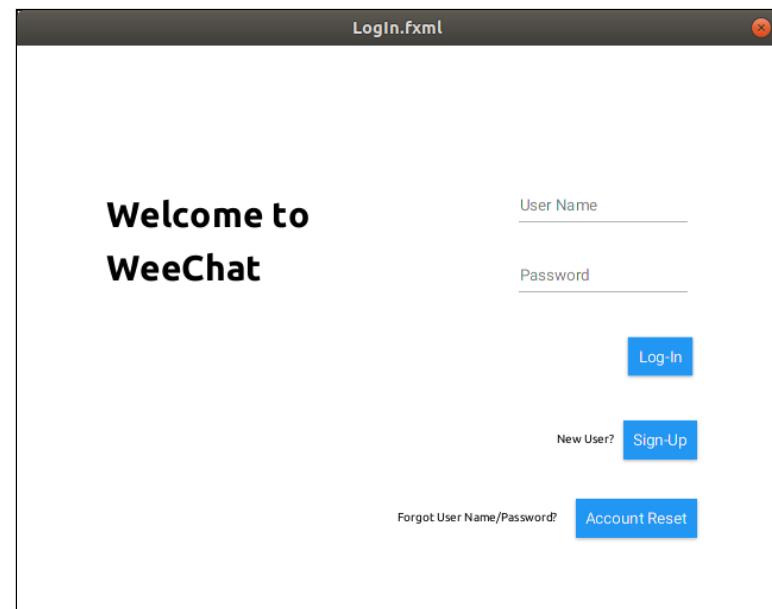
Bottom Mockup: Main Chat Window

This screen is titled "WeeChat". On the left, there is a sidebar with a tree view showing "Server1" (with "Channel1A", "Channel1B", and "Channel1C" listed), "Server2", "Server3", and "Server4". A "Settings" button is located at the bottom of this sidebar. The main area contains a list of messages represented by horizontal bars of varying lengths. On the right, there is a "Members" list containing "User1" and "User2". At the bottom is a text input field with a placeholder, and "Send" and "Attach" buttons.

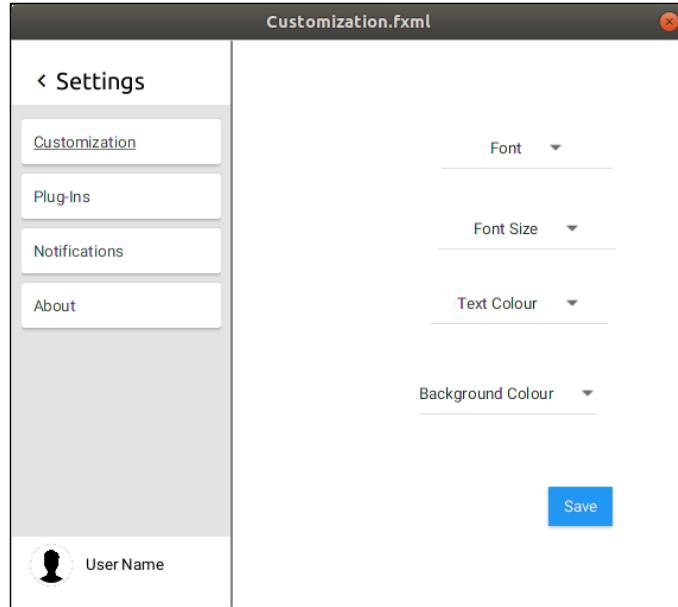




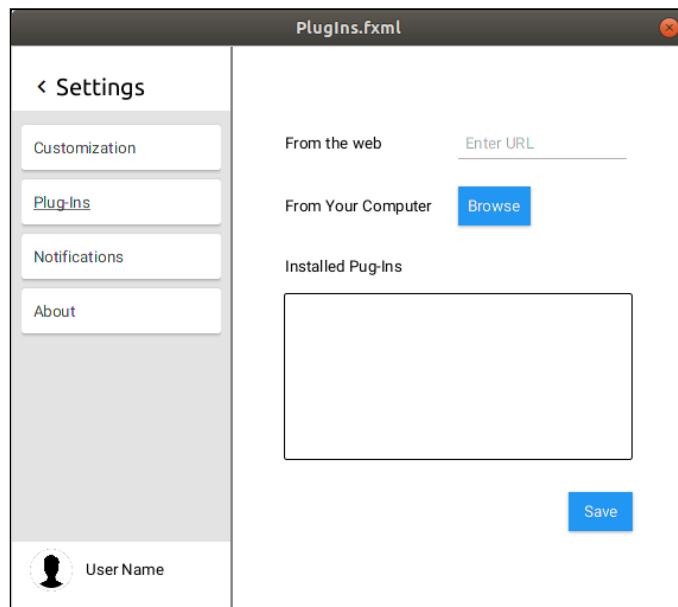
High Fidelity

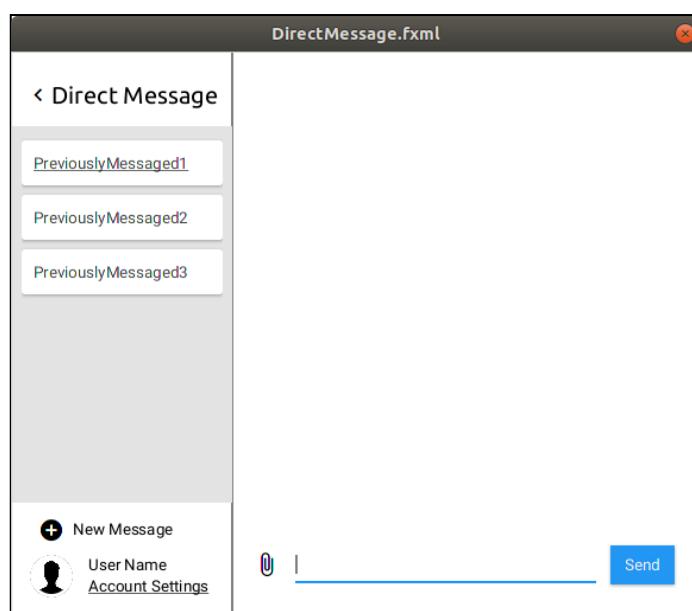
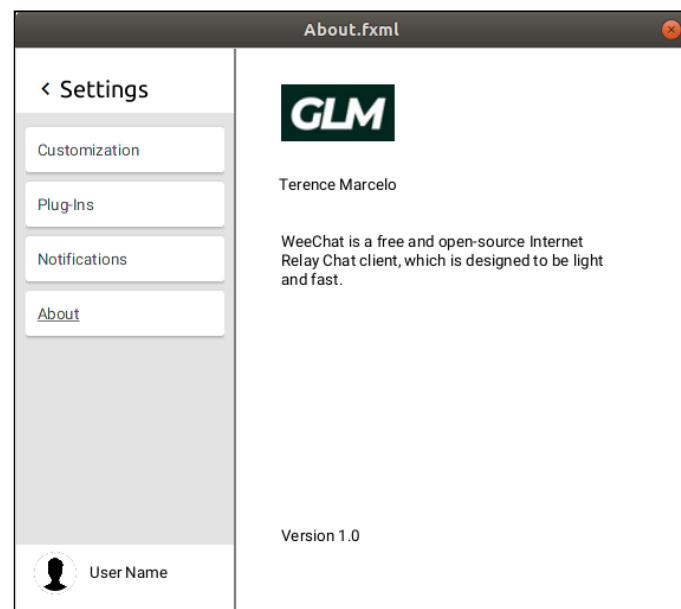
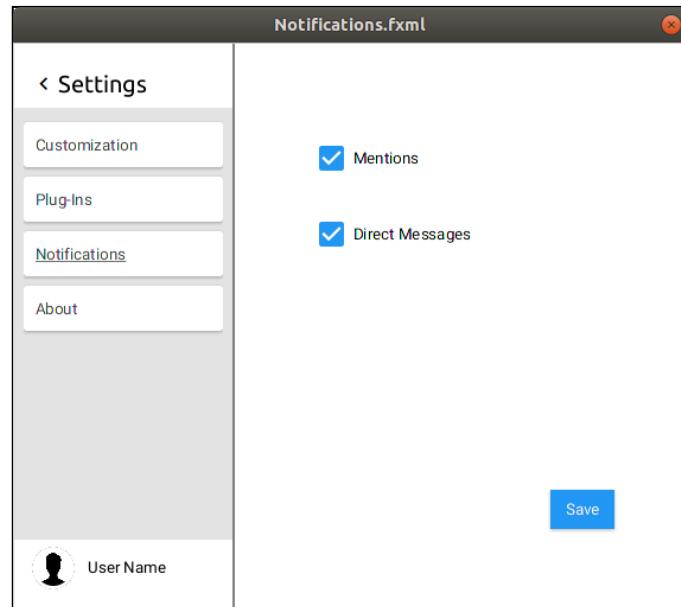


The Main UI shown once logged in. The left side contains tabs to different servers. The tabs can be expanded to show the channels. The right side contains the members of that server. The bottom left contains links to add a new server, to access the direct message window, account settings and profile. Selecting 'Add Server' opens up a pop-up with a text field for the new server's URL.

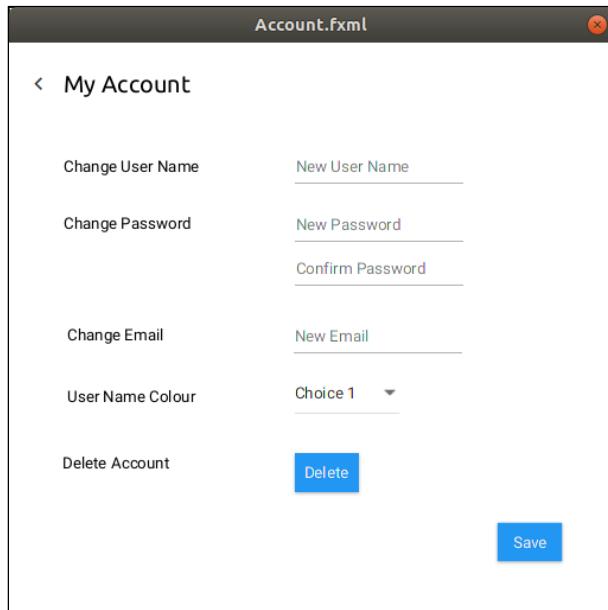


Top left contains a back button to return to the main interface.





Left side contains previous direct message threads.



Selecting 'Delete' opens up a pop-up asking to confirm the deletion.

Ryan Martin (19514231)

Low Fidelity

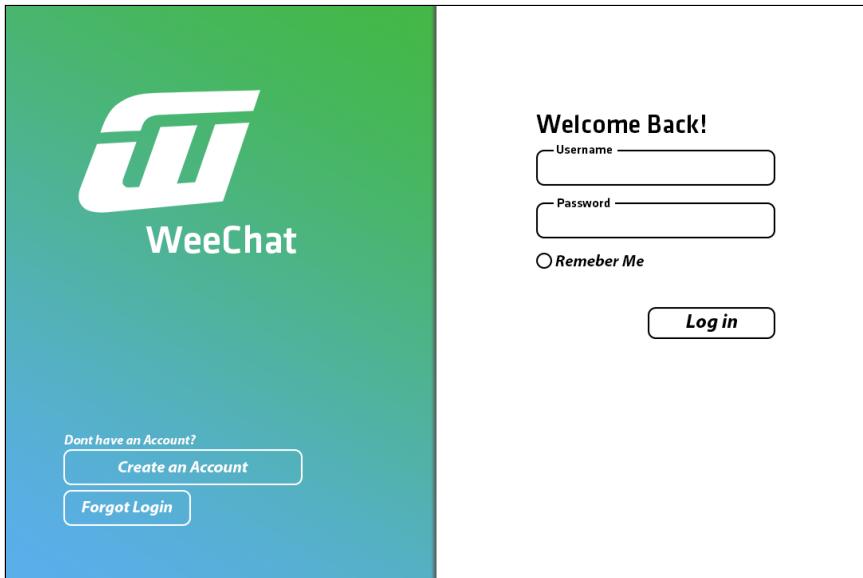
<p>Welcome Back!</p> <p>Username. Password.</p> <p>Log in</p> <p>Dont have an account? Create an Account</p> <p>Forgot Login</p>	<p>Registor</p> <p>Username Email address Password Confirm</p> <p>User Colour terms and conditions</p> <p>Join the Community</p> <p>Already have an account? Signin</p>																														
<p>User Profile</p> <p><username here> <current email here> Change Email</p> <p>User Colour Change Password</p> <p>Sign Out Delete Account</p>	<p>Settings Plugins</p> <p>URL Search Upload from PC</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Added Date</th> <th>ON</th> <th>Remove</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td><input type="radio"/></td> <td>X</td> </tr> <tr> <td></td> <td></td> <td></td> <td><input type="radio"/></td> <td>X</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Description	Added Date	ON	Remove				<input type="radio"/>	X				<input type="radio"/>	X															
Name	Description	Added Date	ON	Remove																											
			<input type="radio"/>	X																											
			<input type="radio"/>	X																											
<p>Servers</p> <table border="1"> <thead> <tr> <th>Username</th> <th><Server Name></th> <th>Active Users</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>MAN Chicken</td> </tr> <tr> <td></td> <td></td> <td>DoB</td> </tr> <tr> <td></td> <td></td> <td>Random</td> </tr> <tr> <td></td> <td></td> <td>Names</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>MAN (19:51 xx/04/2020): the sky is blue Chicken (19:52 xx/04/2020): is it? Sure is :)</p> <p>(Add server) (Settings)</p>	Username	<Server Name>	Active Users			MAN Chicken			DoB			Random			Names										<p>Direct with User Chicken</p> <p>Username</p> <p>MAN (19:51 xx/04/2020): the sky is blue Chicken (19:52 xx/04/2020): is it?</p> <p>Sure is :)</p> <p>(Add server) (Settings)</p>						
Username	<Server Name>	Active Users																													
		MAN Chicken																													
		DoB																													
		Random																													
		Names																													
<p>Settings Plugins</p> <p>Customization Font Size Font Type Background Colour Text Colour</p> <p>Notifications <input type="radio"/> Direct Message <input type="radio"/> Mentions</p> <p>About Coz we are the best!</p>	<p>Add Server</p> <p>Server Address Server name</p> <p>+</p>																														

High Fidelity

WeeChat Logo Retrieved From;

https://upload.wikimedia.org/wikipedia/commons/a/a3/Weechat_logo.png

Login Interface



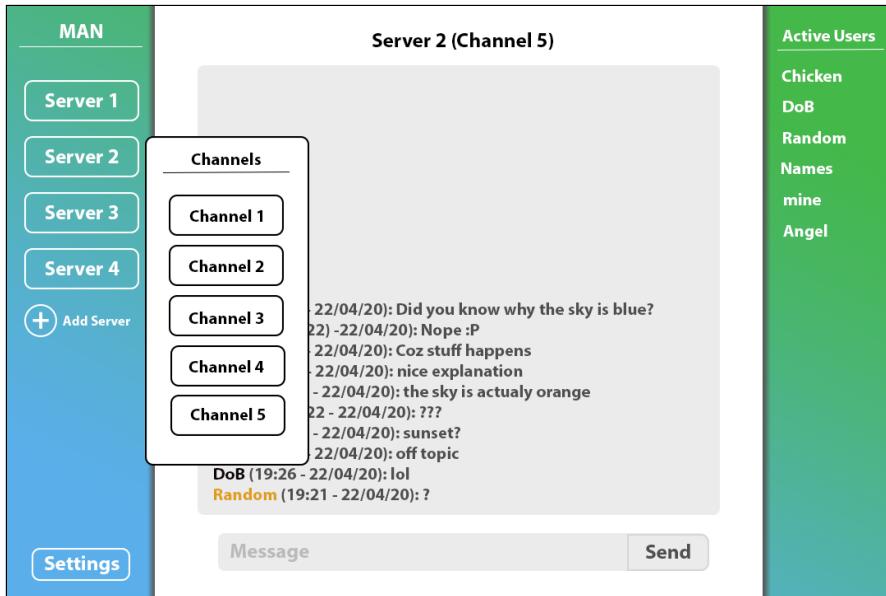
Upon starting WeeChat the user is faced with the login screen, if the user is not already logged in. The user has the option of logging into an existing WeeChat account or to create a new one. When logging into WeeChat the user needs to enter the WeeChat username and password, they also have the option of selecting the 'Remember Me' box for ease-of-use when connecting to WeeChat the next time. In case the user forgets their login details, they can be easily reset with the forget login button or they can create a new account with the 'create an account' button.

Register Interface



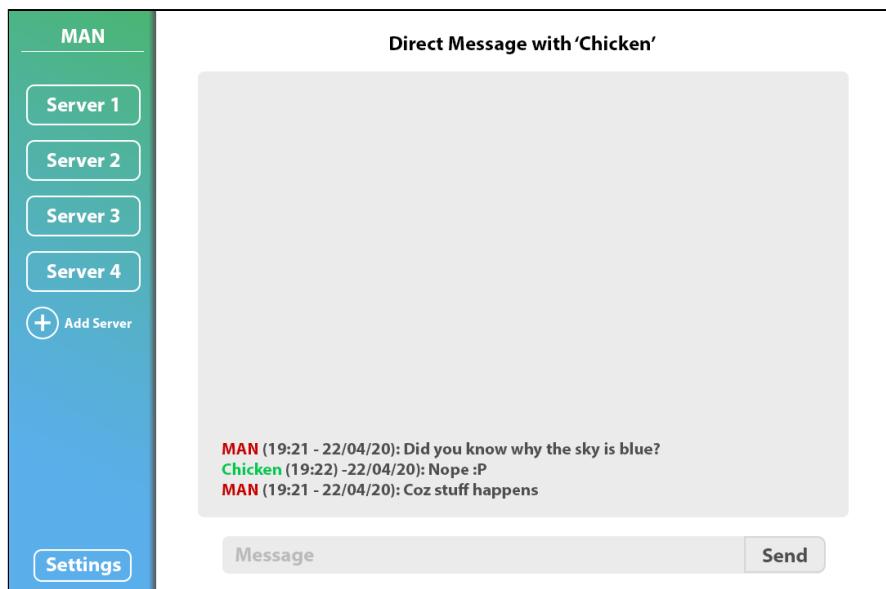
If the user does not have a WeeChat account and selects the 'Create an Account' button displayed on the login interface, they will be redirected to the register interface. Upon arriving at this interface the user is asked to enter a username, email address and a password. After the user's details have been entered the user then can select their username colour by entering the colour code into the marked input box and the colour selected will appear in the circle next to the box to show the entered colour. If the user accidentally clicked the 'create an account' button the 'sign in' button will return the user to the login page.

Server UI Interface



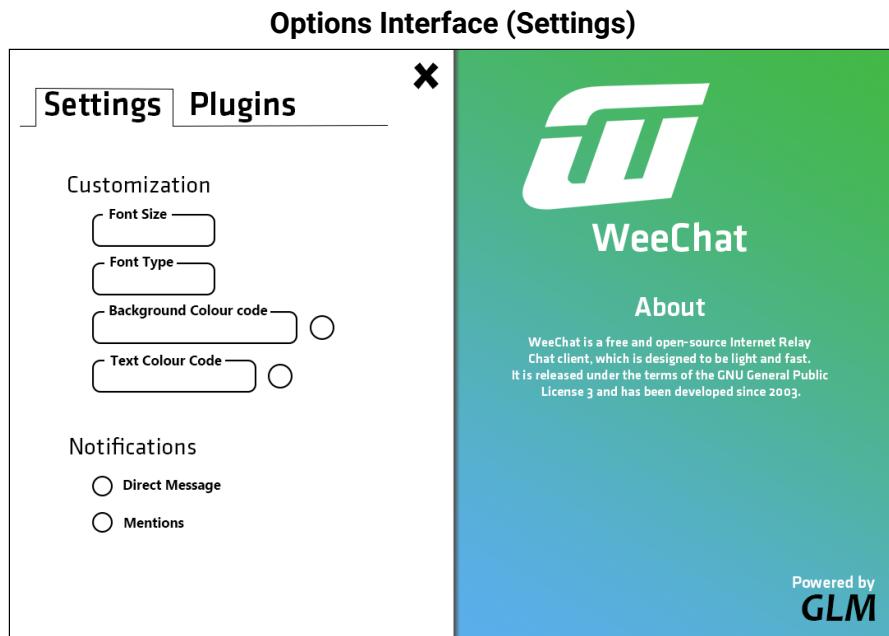
The Server UI interface is the main interface of WeeChat. The last used chat will be displayed on WeeChat when it is opened. The current user's username is displayed in the top left corner and when clicked will open the user's profile. The settings interface can be accessed by the 'settings' button at the bottom left corner. The user can easily choose from all their connected servers from the list on the left. When the user clicks/ hovers over a server all the channels connected to that server will be displayed. When the mouse is no longer hovering over the server the list of channels will disappear. This maximises the chat space while also still giving the user all the available options. On the right side displays all the currently active users on the selected server and channel, this can be scrolled through if many users are connected at once. If a user's name is clicked from the list of active users then the direct message interface is opened. The server name and channel name are labeled at the top to clearly show the user which one they are connected to.

Direct Message Interface

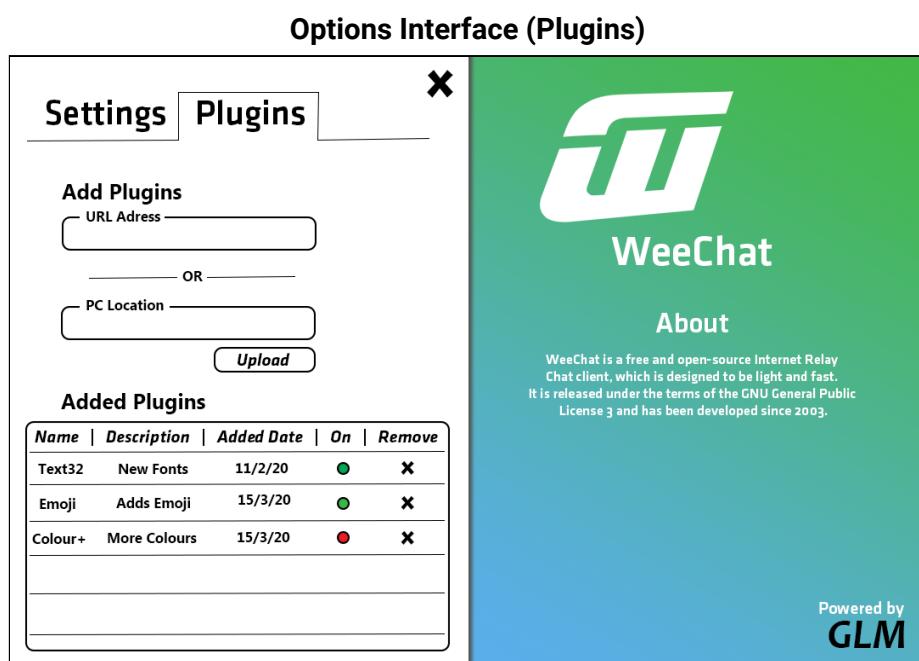


When another active user is selected from the active user list, then the interface is switched to direct message. Here the user can communicate with another user directly. The user can exit this interface by selecting another server and channel and the current direct message will be

suspended. The user can also access the settings, user profile and adding servers from this interface.



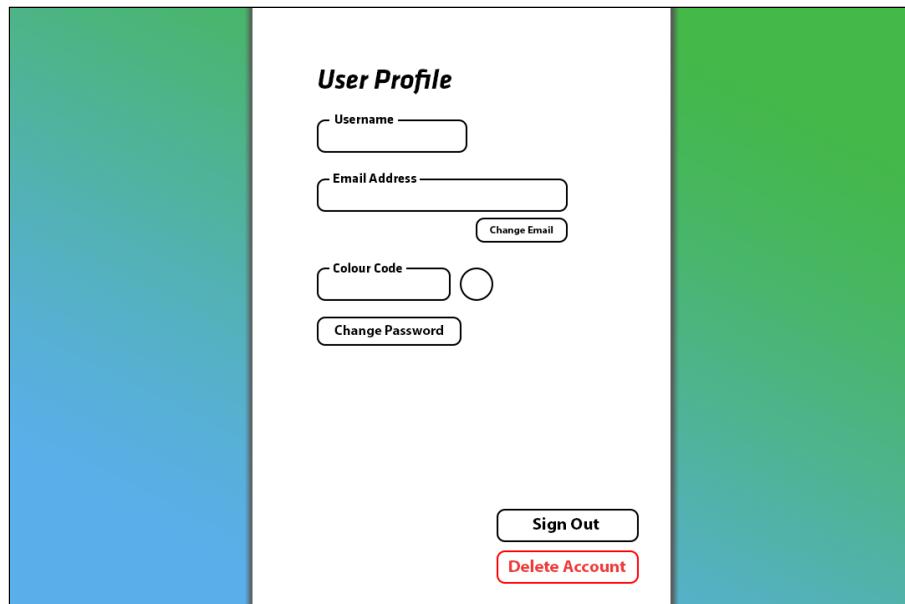
When the 'Settings' button is clicked the options interface is launched, this is where the user can change the font style and size as well as the text and background colour. The options interface can simply be exited by clicking the 'X' in the top right corner at any time. The colour boxes accept colour codes at which the colour selected will then be displayed in the corresponding circle. This interface increases the ease-of-use for the user as before with the old interface this would have to be done with input commands which can be hard to remember and can be entered incorrectly. As well as the new notifications buttons which can quickly and easily change the settings of WeeChat without having to remember the corresponding commands.



By clicking on the 'Plugins' tab in the Options interface the plugin settings are opened. This is where the user can edit, add or remove plugins for WeeChat. The options interface can simply be exited by clicking the 'X' in the top right corner at any time. The plugins interface makes viewing all current active and inactive plugins easier by scrolling through the plugin table. The plugins can also

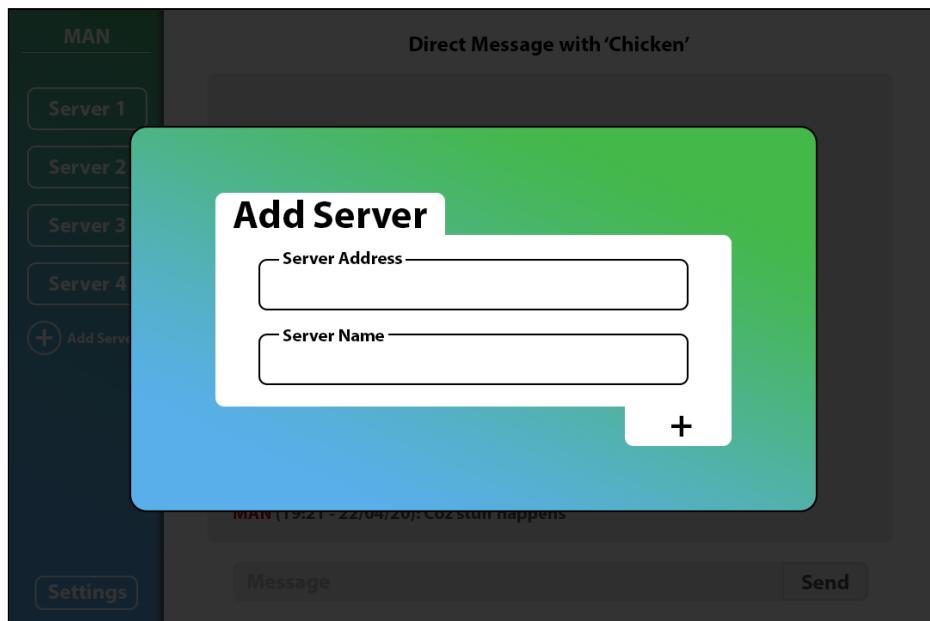
now be sorted by clicking the corresponding title, disabled and enabled using the corresponding button and be easily uninstalled with the corresponding 'X'. Adding plugins to WeeChat can be done quickly by either using the URL address for online plugins or the PC Location for plugins saved to the pc.

User Profile Interface



By Clicking on the username displayed on the server ui and direct message interface the user's profile is opened. The user can easily and quickly change their email, password and/or their selected username colour. The user also has the option of signing out or deleting the current user's account.

Add Server Interface



The User can easily and quickly add a new server to their list by clicking on the add server button on the direct message or server ui interface which then prompts the user with the following interface. The user then can enter the server address, give the server a name then add it to the server list by clicking the '+' button. This is an easier and more simplistic way to add a server to the list over the previous commands needed with the old command line interface.

Implementations

Contributor(s): According to Names

Aaron Gangemi

In this section, I will discuss the NetBeans implementation of my program, including any changes I have made from my prototype, functionalities that have been implemented and any problems that occurred during implementation. Furthermore, following this, I will be discussing the user involvement to confirm that the usability goals have been met, and that my implementation is able to provide a good user experience. Finally, I will end my section with the test cases for my implementation to confirm it is functioning as planned.

Implementation Discussion

After demonstrating my prototype to the client and receiving valuable feedback that can be applied to my implementation, I was able to transition to the implementation phase of the project. I took the clients feedback and applied it to the final implementation. I was successfully able to implement my prototypes and I decided to update certain design choices along the way.

Unfortunately, due to time restrictions for the product, I was not able to construct an SQL database to store data such as login and server details. Instead, my supporting solution was to pass through a user object on each transition from one interface to another, and for each page to set any data it requires from that object being passed through. In addition, from my original high-fidelity prototype submitted to the client, I have updated it to incorporate most of the feedback that will be discussed below and updated some of my design choices.

I will further discuss any functionality that has been implemented successfully and correspondingly, any other functionality that has not been implemented. Any functionality not implemented will be accompanied with a valid reason as to why it was not implemented. I do not believe that any missing functionality will significantly impact my final product as it is a proof of design concept.

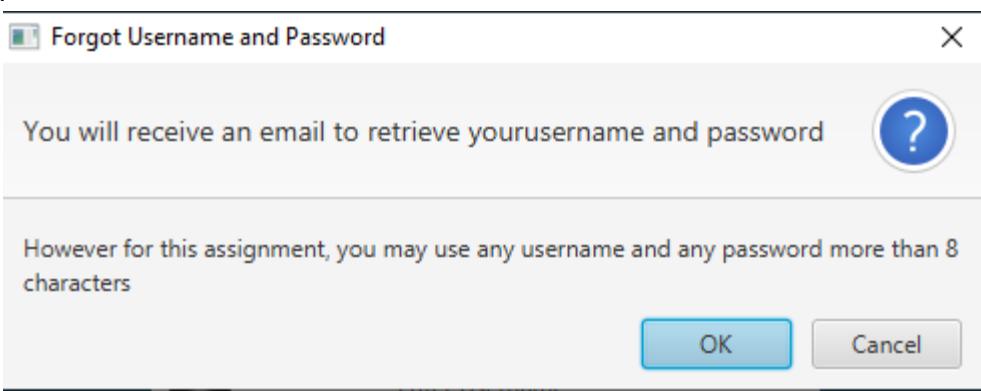
The application can be run using NetBeans. Upon startup, the following page will appear. In order to access the application, the user will be required to either login with an existing account, or create a new account. (**File: Login.FXML, Controller: LoginController.java**)



The password string is required to be greater than characters long and is hashed out as the user is typing. If the user fills in a password that is less than 8 characters, the following alert is displayed. Note that once the search fields are clicked, they clear the placeholder text for the user.

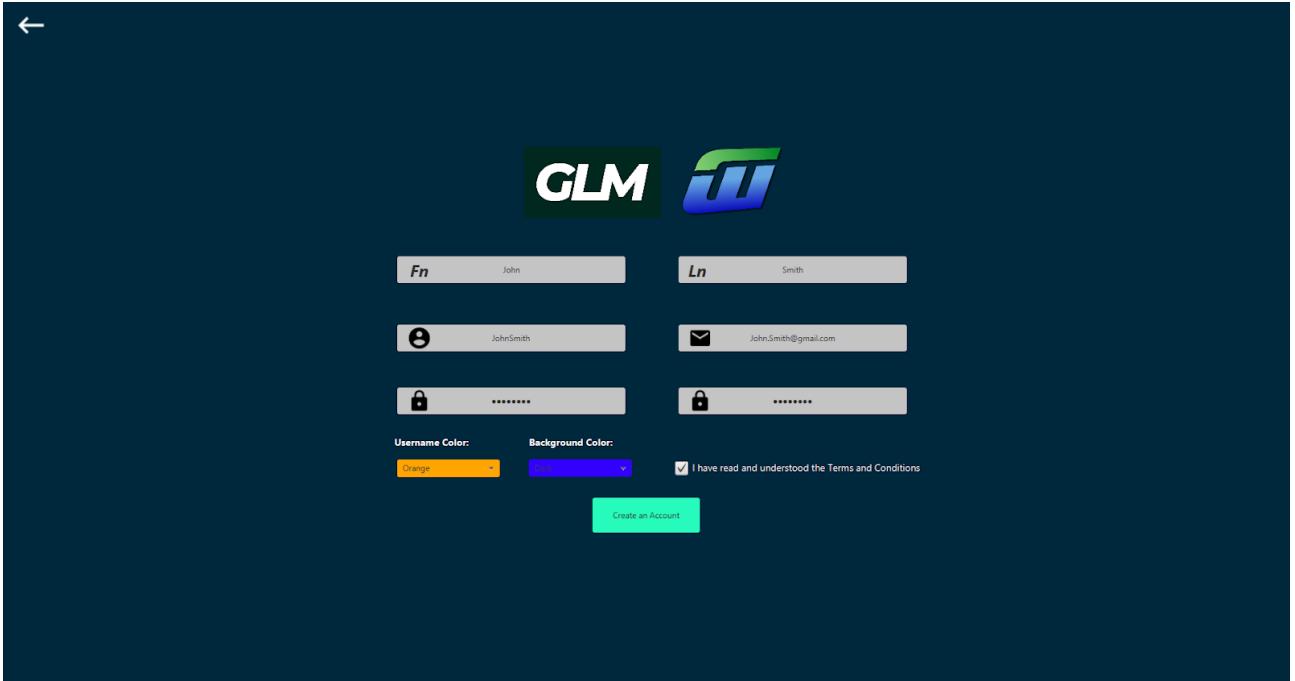
Without a proper account due to no existing database to store accounts, the user can login with any username that is not empty, and any password greater than 8 characters.

The user is able to tick the “Remember Me” checkbox which has not been implemented. However, if it was to be implemented, the users account details would be stored and the user would not be required to login every time the application is run. In addition, the page also incorporates a forgot username and password features which has also not been implemented. If this was to be implemented, an email would be sent to the user asking them to reset either their username or password.

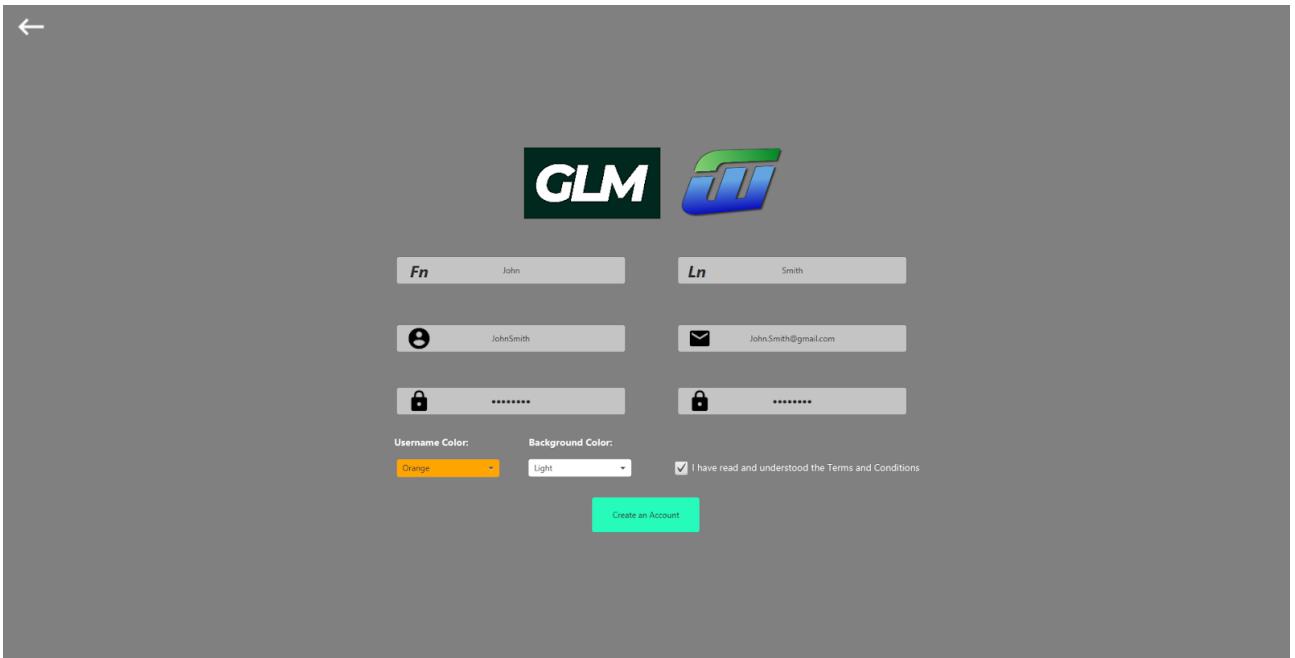


The second way that the user is able to access login successfully is by creating a new account. They can do this by selecting the “Create an Account.” If they choose to, they program will display the below interface:

(File: **CreateAccount.FXML**, Controller: **CreateAccountController.java**)



The create account page contains a username, first name, last name, password, confirm password and email fields for the user. The user is also able to use the list pickers to be able to change the username color used in the program and the background color of the program. The user can either choose between a dark theme and light theme. These have both been chosen as throughout the program they have been tested to provide good contrast with other design features and elements which encourage a good user experience. The light version is shown below



The Create Account page also includes a range of validation ensuring that the password and the confirmed password match, all fields are not empty, the email is a valid format and the password field is greater than 8 characters. If all these criteria match, then the user can login successfully. The user is also required to tick the terms and conditions box in order to proceed. The user is not able to view any Terms and Conditions page as I have not written up any document. The back button in the top left corner is used to take the user back to the login screen. Other than the Terms and Conditions, all features of the create account page have been implemented successfully.

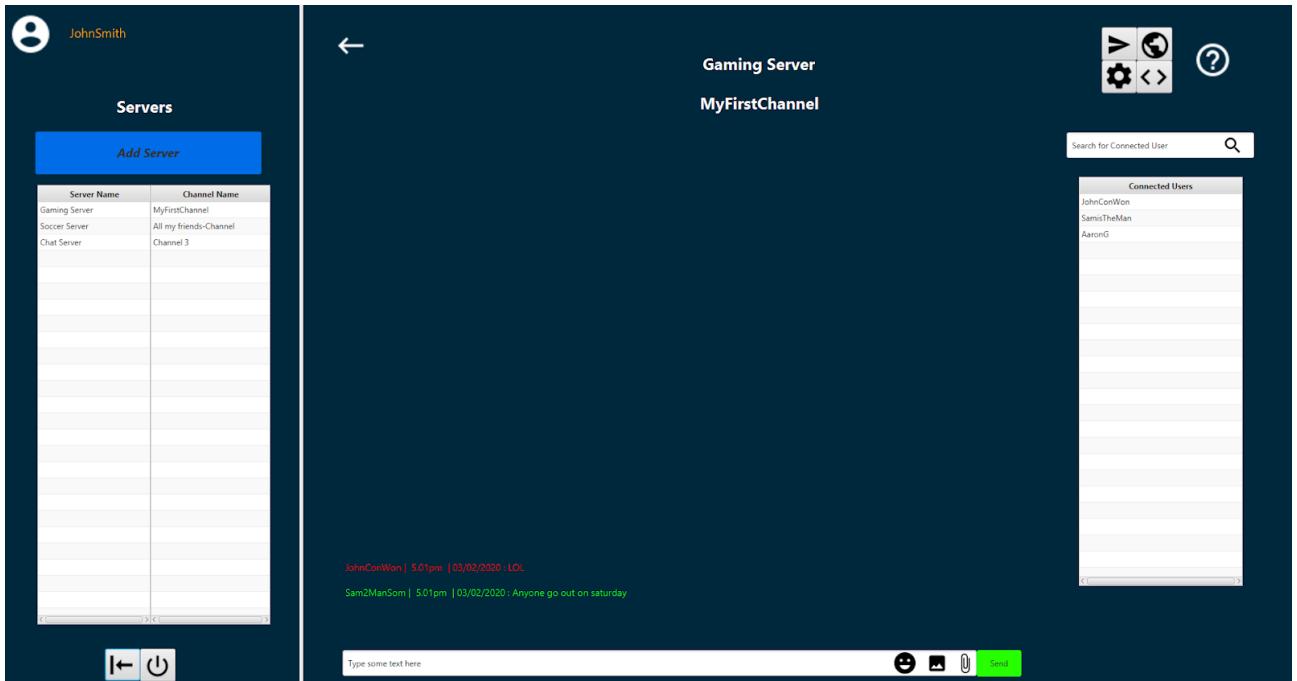
Once the user fills in the Create Account page validly, my program creates a user object which is then passed around the application on every transition, and each interface uses the object to set any data such as the username field in the top corner. They will be able to proceed to the Main Server page which is shown below (with dark theme background):

(File: Server.fxml, Controller: ServerController)

(Server Table: File: Server.java)

(Channel Table: File: Channel.java)

(Connected User table: File: ConnectedUser.java)



The Main server page acts as the central or main page for the application as it is the first interface displayed after login. In the top corner, the username “JohnSmith” that was entered upon creation of a user account is displayed. This data is obtained from the account controller page to the main server page. The Main Server page is also split into 3 sections: the listing of servers and channels, the communication forum, and the list of connected users.

Across all interfaces in my program, a common structure is implemented. The interfaces are divided into sections to ensure that the user is not overwhelmed with clustered information. In addition, there will be the 4 icons in the top corner on every page which provide quick and easy navigation for the user. The icons also incorporate tooltips to assist the user. On the left hand side, the next to the username is the account icon. This provides the user quick access to the account information page. In addition, the logout and exit buttons are also consistent across every page which will allow the user to exit or logout of the application at any time. By providing these features on each interface, the user experience is enhanced as the user does not have to navigate through a maze of pages to reach their desired page and they can easily get from one page to another.

The first feature of this page is the server list on the left side of the page. It displays a list of servers that have been previously added by the user. I have added some pre-existing servers with associated channels. The channels have not been properly implemented in the program, and therefore I have stubbed them in the table. If a server is clicked, the channel list will not change. However, if a user selects a server or a channel, the text at the top of the screen stating “Gaming

Server" will change to the server name that was clicked, and if a channel name is selected, the channel name that is currently "MyFirstChannel" will also change on selection.

The servers are listed in the table with the header "Server Name" and the channels are listed under the table with the header "Channel Name." Initially in my prototypes I was going to display the channels in a context menu that the user was going to need to right click on to view, however it is a lot easier for the user if all the channels and servers are displayed at the forefront of the interface.

One thing I was not able to implement throughout the application that was asked on client feedback was the minimise and maximize feature. This feature was requested by the client, however the user is able to minimise and maximize the page using the provided buttons in the top left corner of the Ubuntu interface and therefore I decided not to go ahead with it. However, this design choice has not impacted my overall functionality of my product.

If the user right clicks on a server, an context menu will appear, which if clicked will allow the user to delete the server and associated channels.

The icons in the bottom left corner are used to logout or exit from WeeChat. If either of these are selected, an alert will occur and the user will be asked to confirm their decision.

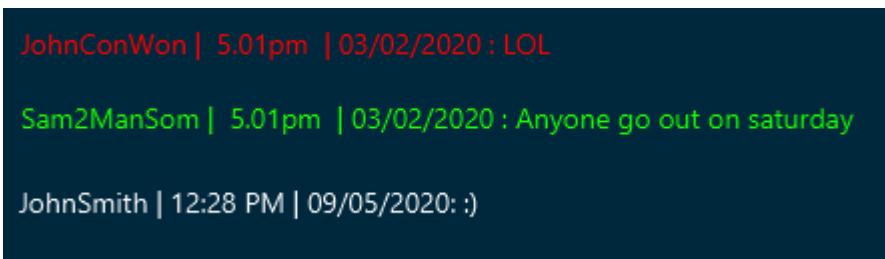
The account information icon in the top left hand corner also acts as a button. If the account information is clicked, the user will be redirected to the user information page which is depicted later in this discussion.

Furthermore, the second part of the server interface is the forum that allows users to communicate with each other by sending either an emoji, text, file attachment or image. I have pre populated the chat with a couple of messages from other users. If the user enters text such as "Hi, my name is John" and clicks send, the following occurs:



The text typed appears in chat.

If the user selects the emoji icon, the ":" text will appear as a placeholder. The following occurs if the user clicks the emoji icon:



The ":" appears.

If the user selects the image icon, the operating systems file explorer will open and the user will be asked to select an image file. The file explorers filter will be set for “Image Files.” For example, if I select an attachment called “Capture.png”, although I have not implemented this feature, the file path will be displayed as text in the chat to demonstrate a proof of concept.

JohnConWon | 5.01pm | 03/02/2020 : LOL

Sam2ManSom | 5.01pm | 03/02/2020 : Anyone go out on saturday

JohnSmith | 09:07 PM | 08/05/2020 | C:\Users\61459\OneDrive\Desktop\Capture.PNG was uploaded

The filepath of the image in file explorer was displayed in text.

If the user selects the attachment icon, the same process as sending an above image will occur, however, when file explorer is opened, there will be no filter, meaning the user can upload a file of any type.

JohnConWon | 5.01pm | 03/02/2020 : LOL

Sam2ManSom | 5.01pm | 03/02/2020 : Anyone go out on saturday

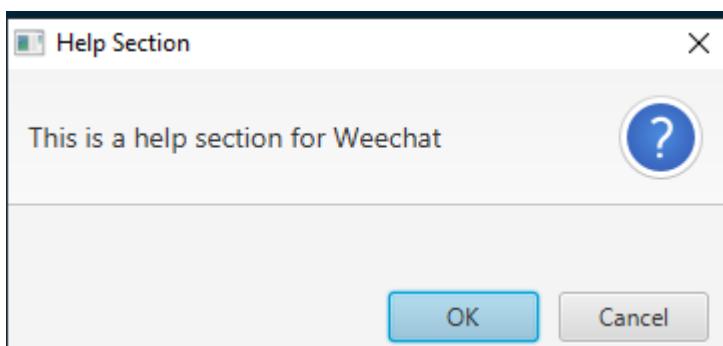
JohnSmith | 09:11 PM | 08/05/2020 | C:\Users\61459\OneDrive\Desktop\AccountController.cs was uploaded

The user has successfully uploaded a C# file.

The final component of the middle part of the server page is the back button. If the user clicks the back button, they will be taken to the options and settings page in the program.

The final part of the main server page is the right panel. The panel displays a list of users that are connected to the channel. If the user selects any of these usernames, they will be directed to the direct messages page in which they will be able to select a user. This feature has not been implemented as I would require a server to implement and this feature is used to demonstrate a proof of concept for this implementation. The search for user also has not been implemented due to time constraints, but its purpose if it was implemented is to allow the user to search for a username, and later highlight any users with a match to the inputted string.

The help icon in the top corner was a feature requested by the client to be implemented after submission of the prototype. If the user clicks on the help icon, an alert will pop up to represent a help message.

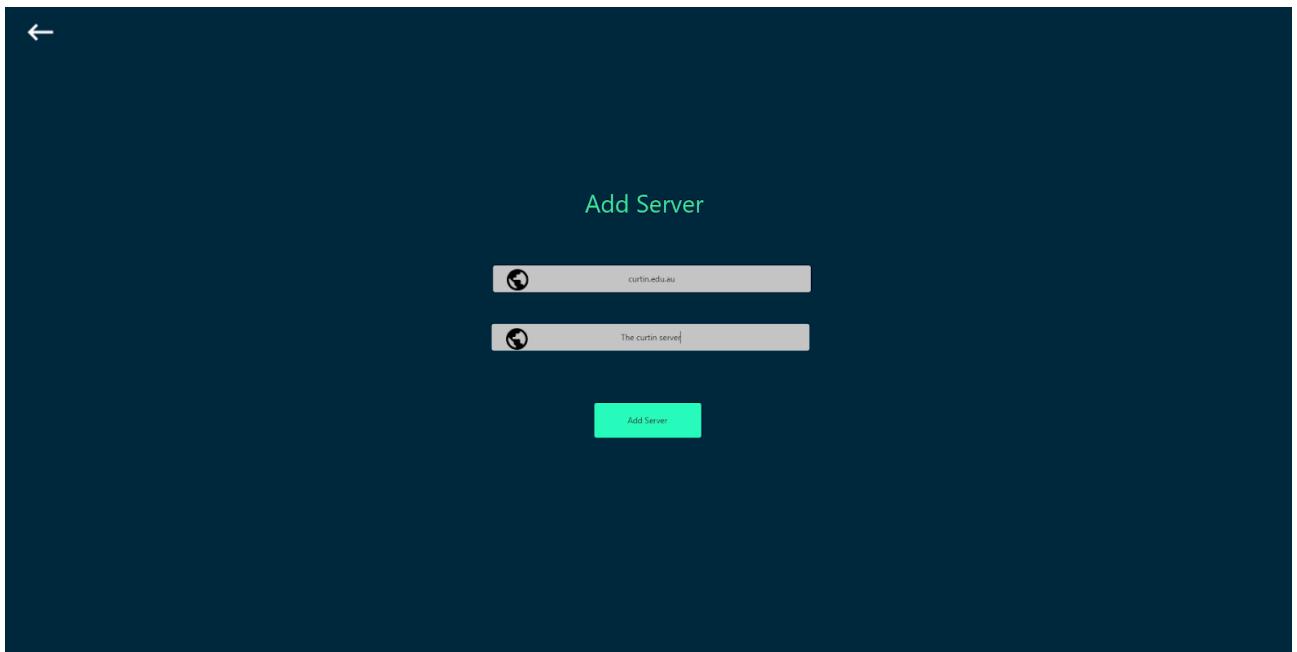


Not implemented as it is a proof of design concept, however it is used to demonstrate a help section implemented in WeeChat.

Finally, the 4 buttons in the top corner are displayed on all pages and are used to provide quick and easy access to other pages within the application. They contain links to the direct messages page, the main server (Current) page, options and settings page and the script manager page. By incorporating this feature, the user is able to easily navigate to the page they want, regardless of what interface they are on. This helps provide a good user experience as the 4 buttons are consistent across most interfaces in the application. In addition, the client requested that I add tooltips to my implementation. If the user hovers over either over these icons, the tooltips will be displayed.

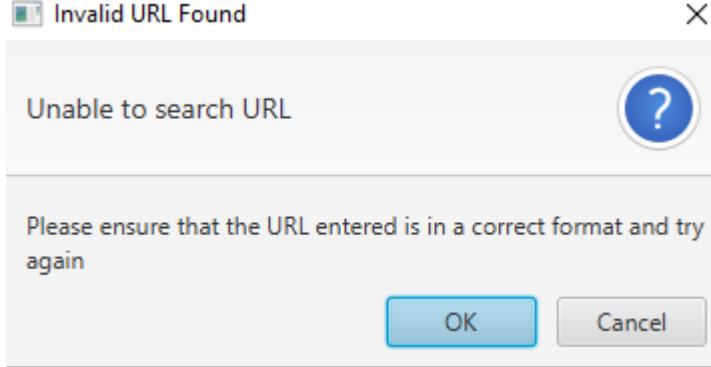
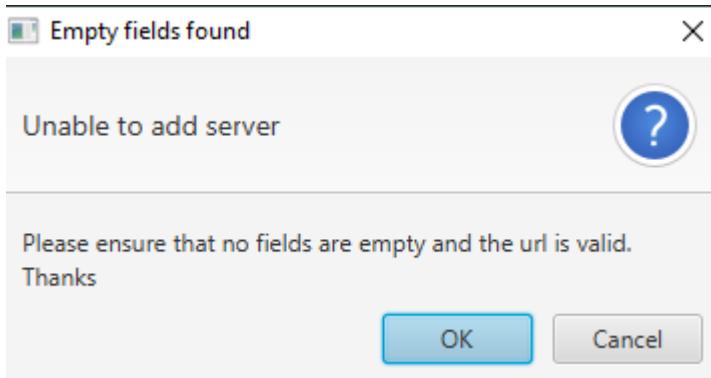
The next interface discussed will be the Add Server interface:

(File: **AddServer.fxml**, Controller: **AddServerController.java**)

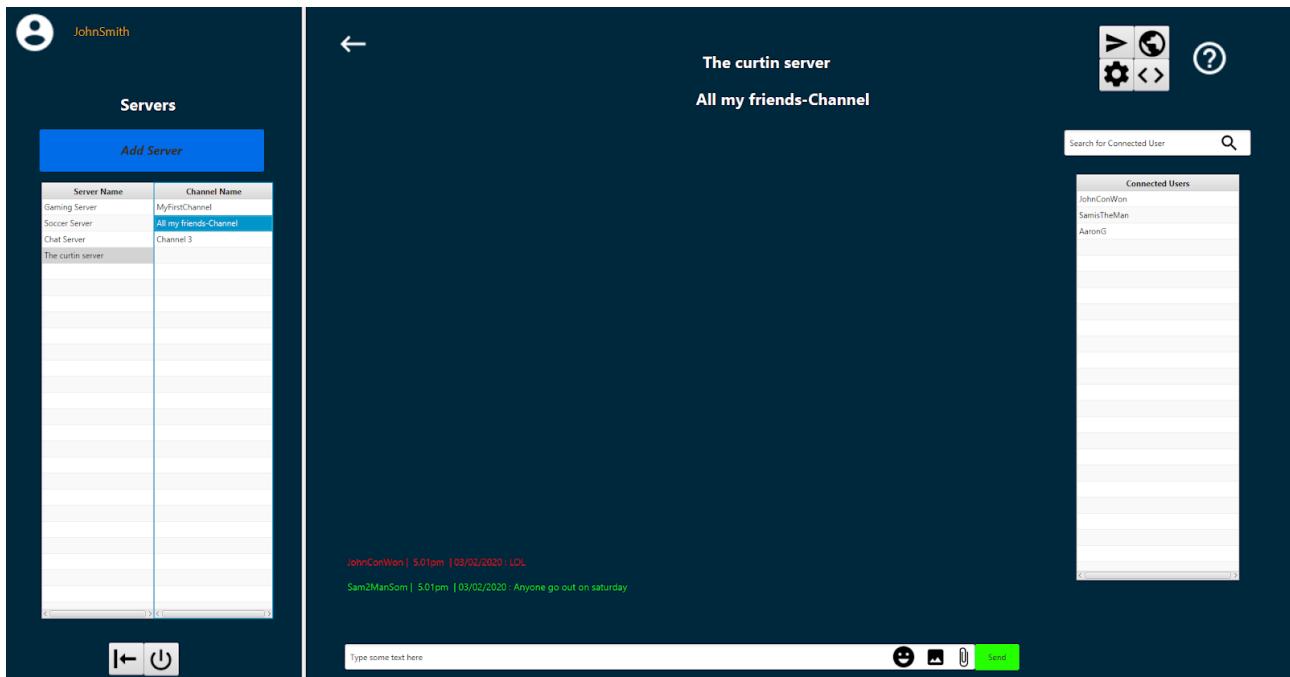


The user is required to type in a valid server URL and a server name. The program then performs validation checks to determine if the URL is a valid string and both fields are not empty. The server fields also provide example URLs in the right hand side of each text field with lower opacity. Once the user clicks on the input fields to enter their text, the placeholder text will disappear, making way for the user.

This gives the user an example of the type of data that will be accepted by the field, which will help them to use the program. If the validation fails then the following alert will appear and the fields will reset to their default strings.



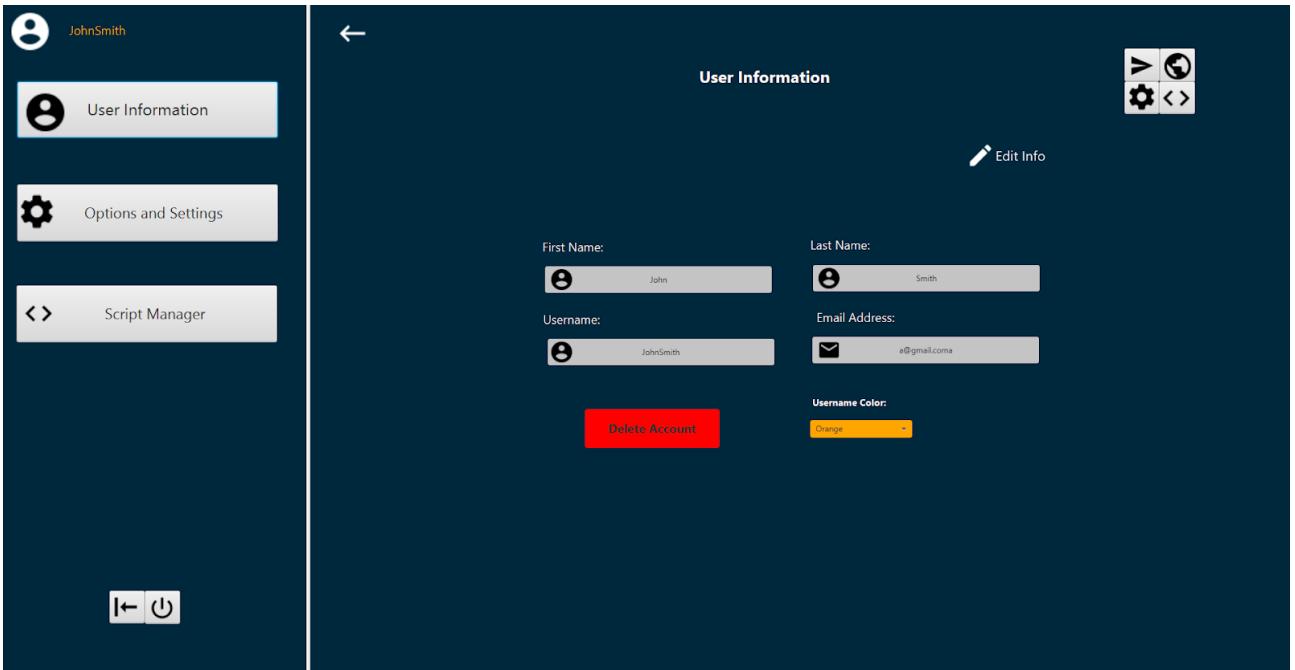
Once the user enters a valid URL, such as the web URL for “The curtin server”, the server will be added to the Main server table as depicted below:



Demonstrates adding “The curtin server” to the main page

The next interface I will discuss is the user information page which can be accessed from the main server when the user selects the account icon in the top left corner. It can also be accessed from the options and settings and script manager pages.

(File: **UserInformation.fxml**, Controller: **UserInformationController.java**)



The user information is split into two sections. The left section introduces the use of large buttons that are easily visible to the user. The buttons demonstrated in my program are grey as they provide good contrast with the dark background and make the text easy to read which enhances the usability of the program. This is also demonstrated through the red delete account button and the option lists to change username color as the red and orange in the above image create good contrast with the background.

Each tab in the left hand side, if clicked transitions the user to the corresponding page. From here, using the tabs, they will be able to visit either the options and settings interface where the application settings are displayed or the script manager will be displayed to manage plugins.

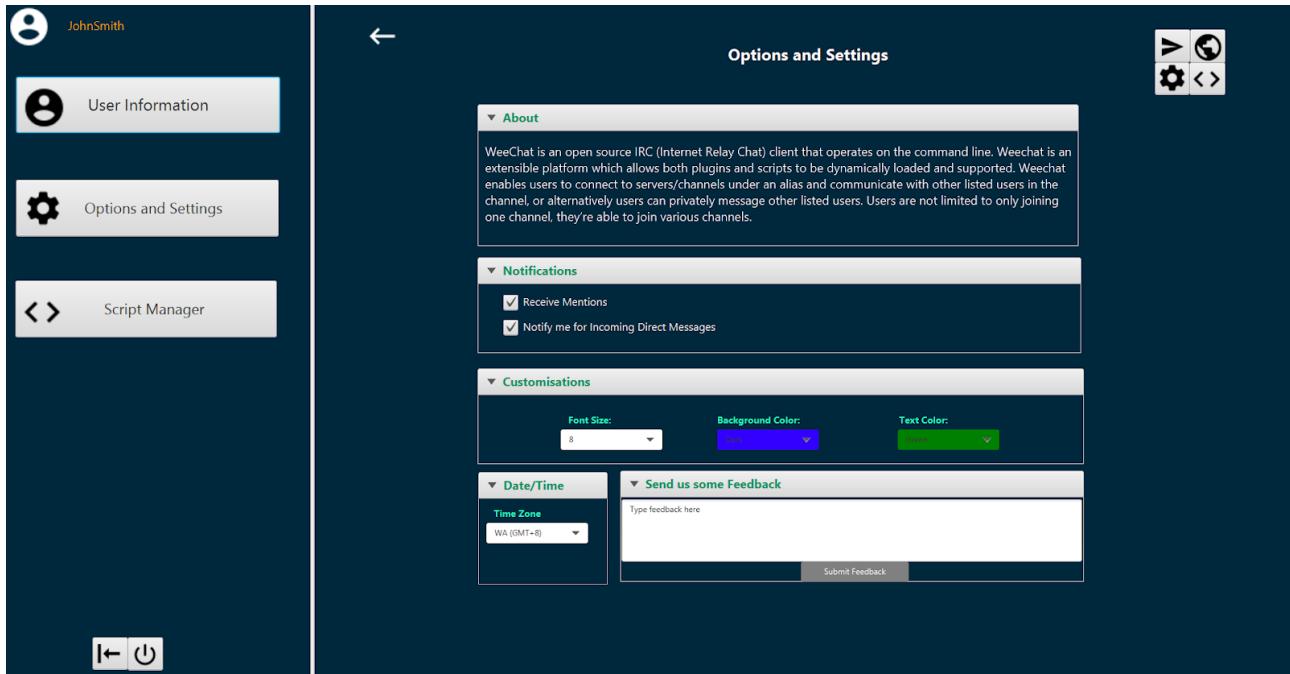
The right hand side of the user information page displays the fields that the user entered on creation of their account. The fields are set using the user object that is passed around the application on each transition. All fields initially are not editable. Once the user clicks the pencil next to "Edit Info", the fields will unlock and the user will be able to edit them. On change of the username color using the color list, the text in the top left hand corner will also change. When the user has finished making changes, they can click the edit icon again, and the changes will be saved.

Finally, the 4 icons are once again displayed to provide a familiar and consistent design across all pages.

If the user selects delete account, an alert will be displayed asking them to confirm that they wish to delete their account. If they choose to proceed with deleting their account, they will be taken back to the login screen. Because there is no SQL database, there is no way to delete the account, however once they log back in from there, the user object containing the user data will be overwritten.

From here, the user may wish to select the options and settings button in which they will be displayed the page below:

(File: OptionsAndSettings.FXML, Controller: OptionsAndSettingsController.java)



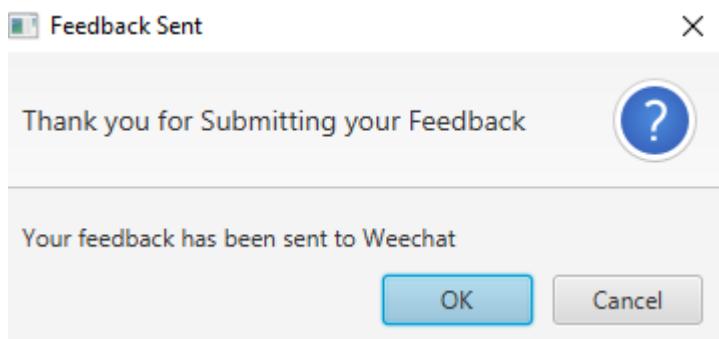
The options and settings page is also broken down into 2 sections like the other user information page. The tabs linking to other pages on the left, and the main content on the right.

In the right section of the interface, each section of content is divided using drop down boxes, meaning the content is expanded, but if the arrow in the top left corner of each section of content is clicked, the content will collapse.

The about section gives the user a description of WeeChat. The notifications section includes tick boxes which allow the user to change their notification settings. These notifications have not been implemented in the application due to time constraints and the fact that the main chat has also not been implemented. The next customisations section allows the user to change the background color, text color and font size. They are all implemented as drop down lists that allow the user to pick another option. The background color is implemented successfully, however the text color and font size have not been implemented due to time constraints.

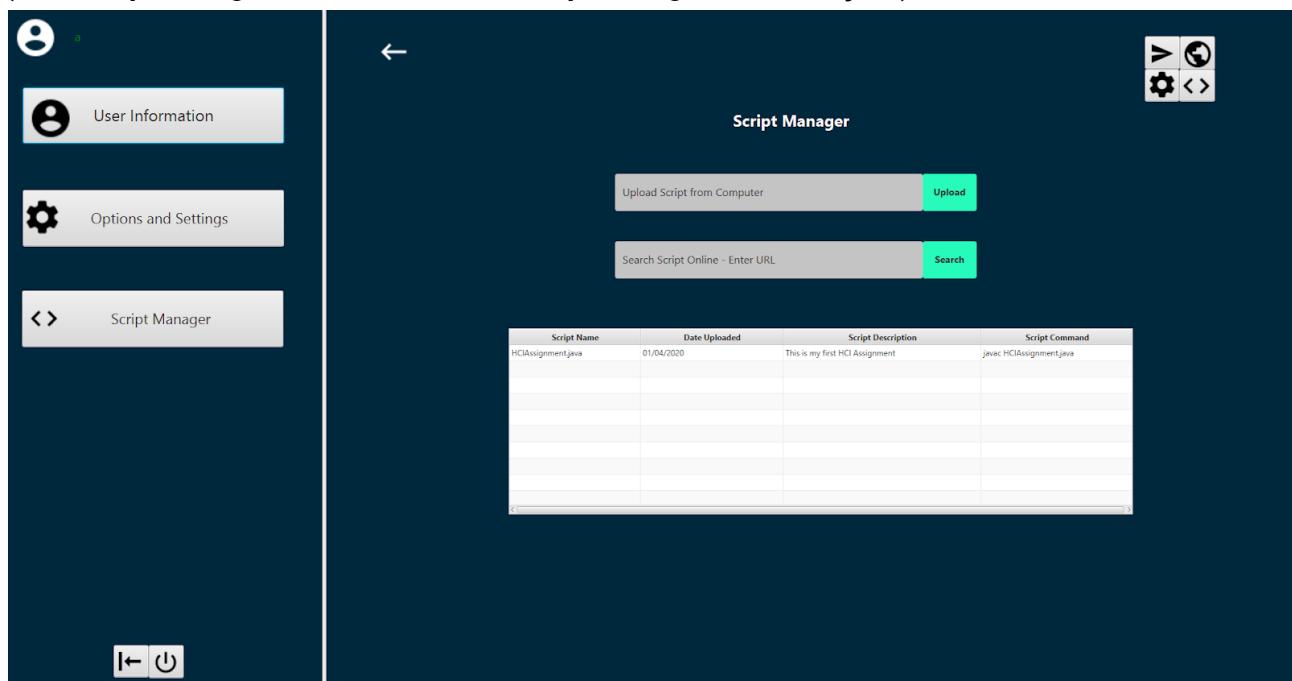
The date/time section has also not been implemented, but its purpose is to display the time of the selected time zone with the message that is sent in either the main server or direct messages interface.

Finally, the feedback section was a request from the client. This allows the user to provide feedback to Weechat regarding the application. As the feedback button has not been implemented, once the user submits their feedback using the "Submit feedback" button, the text field is cleared and the following alert will appear:



The next interface discussed is the script manager.

(File: ScriptManager.fxml, Controller: ScriptManagerController.java)

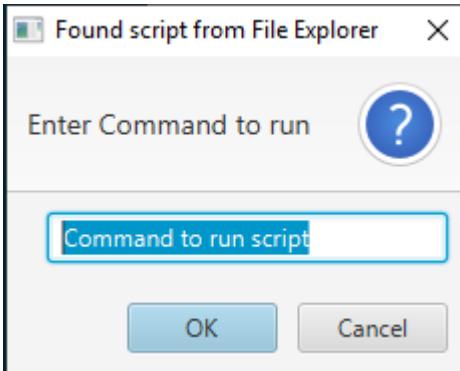
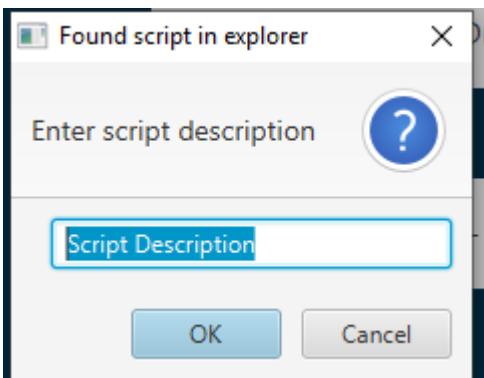


The left panel is the same across the settings pages and has been discussed above. The right hand side panel incorporates a range of features. The script table is used to display scripts that are uploaded either through URL search or locally using the above fields and buttons. I have pre populated the script table with a script for the user to indicate the purpose of the table.

If the user selects the upload button, the program will open a file explorer and ask the user to choose a script.

If the user chooses to enter a script using the URL, and clicks search, the script will be added to the table, however this has not been implemented as I encountered problems with ubuntu unable to open the web browser. To replace this functionality, I have just displayed the script URL as the script name.

Following either of these methods to upload a script, once the user selects a valid script from file explorer or enters a URL, the following alerts will be displayed:



After these have been successfully entered, they will be displayed in the table as follows:

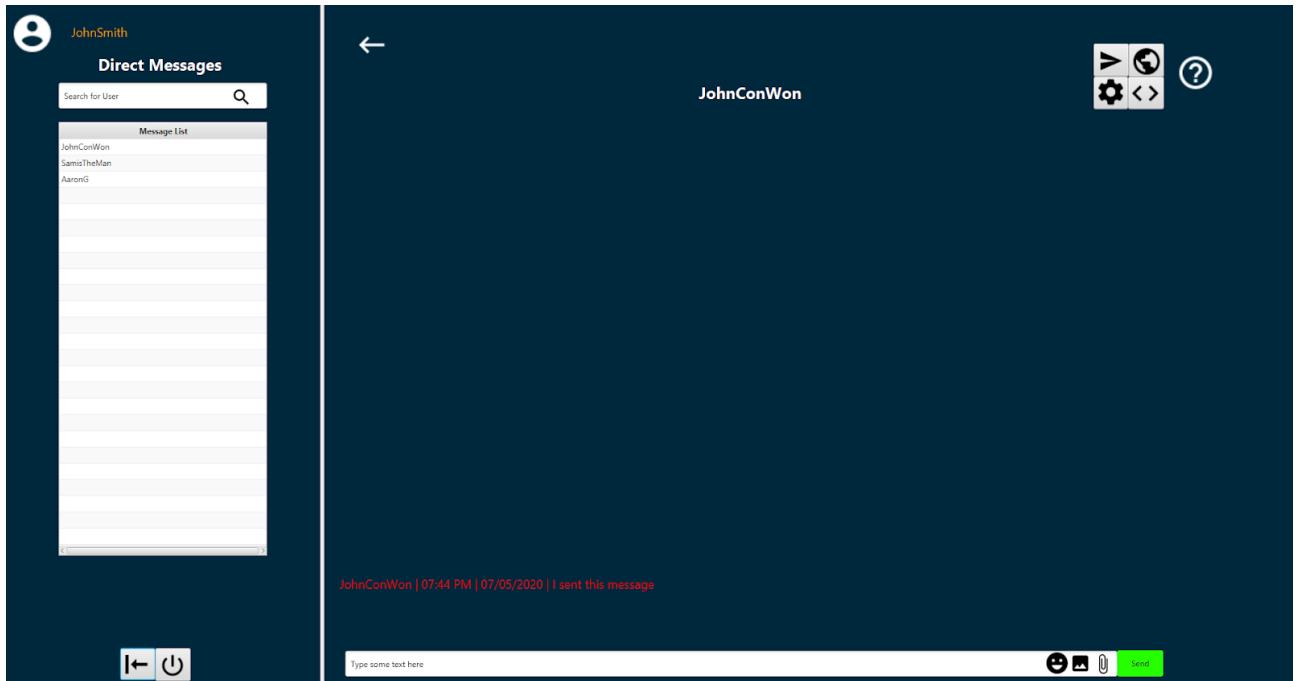
Script Name	Date Uploaded	Script Description	Script Command
HCIAssignment.java	01/04/2020	This is my first HCI Assignment	javac HCIAssignment.java
AccountController.cs	09/05/2020	A C# app run in java	javac main.java

The script has successfully been added to the table.

The program does this by creating a script object with each field.

The final page is the direct messages page.

(File: DirectMessages.fxml, Controller: DMController.java)



The direct messages page contains a similar format to the main server page. This is to provide a consistent and familiar feel for the user for interfaces that allow them to send communication. This will allow it to be more easy to learn and use as they would've seen and used a page similar to this previously in the program. As seen previously, the page is split into 2 divisions, maintaining the same structure throughout the application.

The left hand side contains a list of users that the user has direct messaged previously. I have mocked this as it has not been implemented due to the chat communication amongst users also not being implemented. The table has been implemented using the connected users object. Therefore, there are pre populated users in the table. If the user clicks on a specific name, the recipient name at the top of the direct message page will change to the selected user page. Secondly, the search for a user search bar has also not been implemented due to time constraints, however its purpose would be to allow the user to search a username, and highlight any matching results.

In the right hand side of the application, I have pre populated the chat with a user called "JohnConWon" who has sent a message, and the date time he sent it. Once the user enters a message and clicks send, the message will appear similar to the message appearing in the main server, including the data and time that the message was sent..

```
JohnConWon | 12:31 PM | 09/05/2020 | I sent this message
JohnSmith | 12:31 PM | 09/05/2020 | Hi there
```

As the emoji, image and attachment has not been implemented, like the server page, I have included text to represent if it was implemented as demonstrated below.

JohnConWon | 12:31 PM | 09/05/2020 | I sent this message

JohnSmith | 12:34 PM | 09/05/2020: :)

The emoji ":" was sent to represent an actual emoji sent.

JohnConWon | 12:31 PM | 09/05/2020 | I sent this message

JohnSmith | 12:34 PM | 09/05/2020 | C:\Users\61459\OneDrive\Desktop\Capture2.PNG was uploaded

The file path was sent to represent an image being sent to chat.

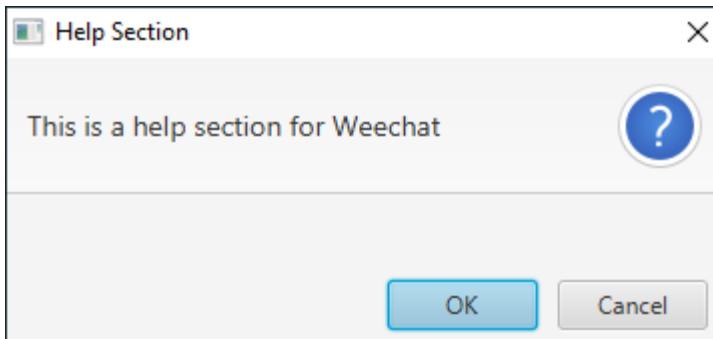
JohnConWon | 12:31 PM | 09/05/2020 | I sent this message

JohnSmith | 12:35 PM | 09/05/2020 | C:\Users\61459\OneDrive\Desktop\TransactionController.cs was uploaded

The file path of the uploaded attachment was successfully uploaded to chat.

The direct messages chat has not been properly implemented as the messages do not move up after being sent. Instead the user message is just overwritten by the next message sent.

Finally, the client requested a help function to be implemented for the application. Like the server page, I have also included it in the direct messages page. If the user clicks on the question mark icon, the following help alert will be displayed.



Help alert is displayed.

No help function has been implemented, however help text would be displayed in this alert.

In conclusion, after taking the client feedback on board, I have made some changes that I believe will improve the user experience for the project. The only feedback I have not implemented is the minimize and maximize buttons. There is a fair bit of functionality missing, however, as a proof of design concept, my project has successfully been implemented to demonstrate a graphical design for WeeChat.

Overall, my implementation includes an easy to navigate, consistent and familiar design which provides quick access throughout the application. As mentioned earlier, this significantly enhances the applications usability. Correspondingly, with the design features combined and discussed throughout my discussion, I believe that my implementation provides a good user experience as the program is easy to learn and understand for a first time user, as all elements are at the forefront of the applications and labelled quite clearly for the user to understand.

User Involvement

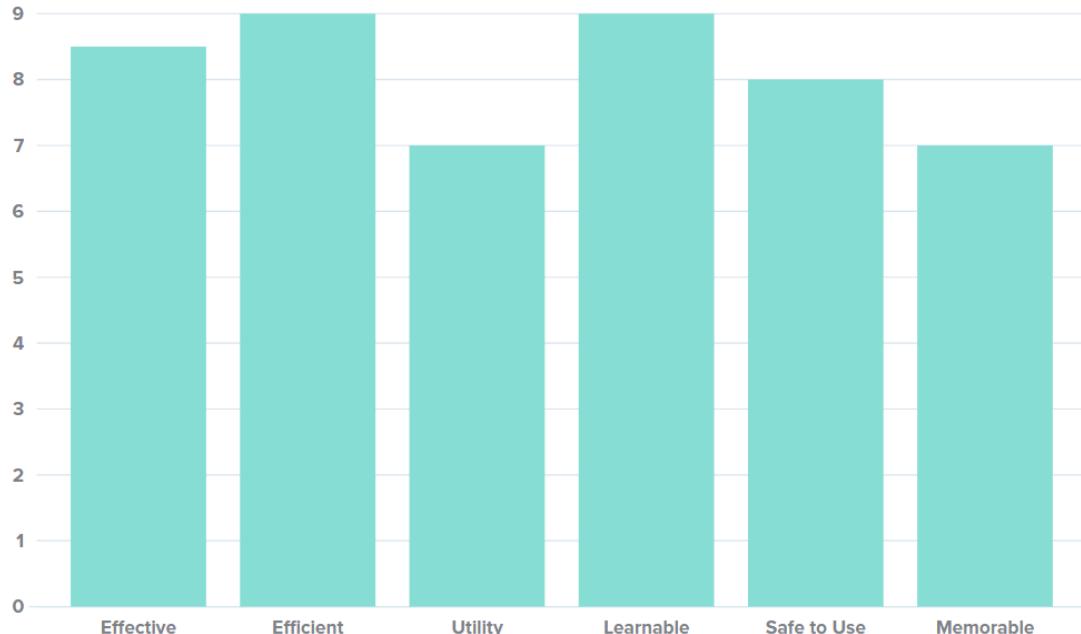
Contributor: Aaron Gangemi

After I completed and tested my implementation for WeeChat, I consulted with users to get their valuable insight and feedback on the new WeeChat system. I have asked the users to give their feedback on the overall user experience and how well the following usability goals mentioned by Preece, Rogers and Sharp:

1. Effective: Effective to use
2. Efficient: Efficient to user
3. Utility: have good utility
4. Learnable: Easy to learn
5. Memorable: easy to remember how to use
6. Safe to use: Safe to use

I later compiled the results of the provided by the group of users into a bar chart as demonstrated below. The users found that the program was very effective, efficient and learnable, however in achieving this, the program lacked utility and wasn't as easy to memorise. This was due to features that had not been implemented due to time constraints or lack of resources utilized such as the SQL database implementation.

Usability of WeeChart



However, on a more promising note, it is pleasing to know that the group of users found the program to be highly effective, effective and learnable. This was due to the consistent interface and the use of labels for functions throughout the program.

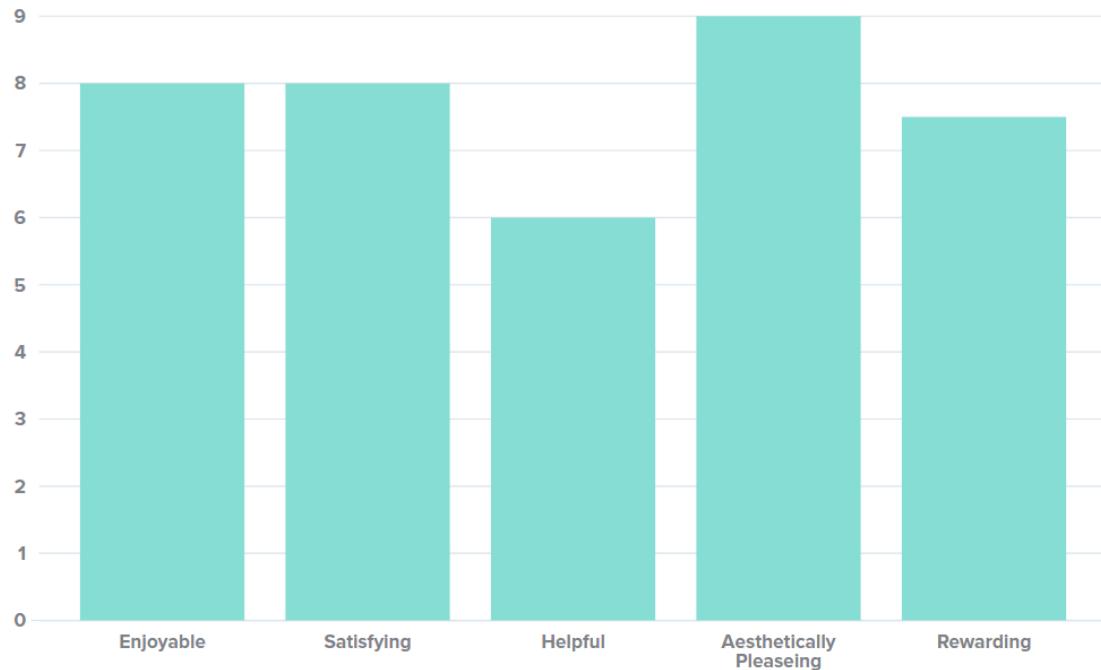
Overall from the above graph, the group of users deemed WeeChat as highly usable.

Following this, the user was asked to provide a rating from 1 to 10 for user experience goals that are stated by Preece, Rogers and Sharp including:

1. Enjoyable
2. Satisfying
3. Helpful
4. Aesthetically pleasing
5. Rewarding

The bar graph below depicts the results provided by the user:

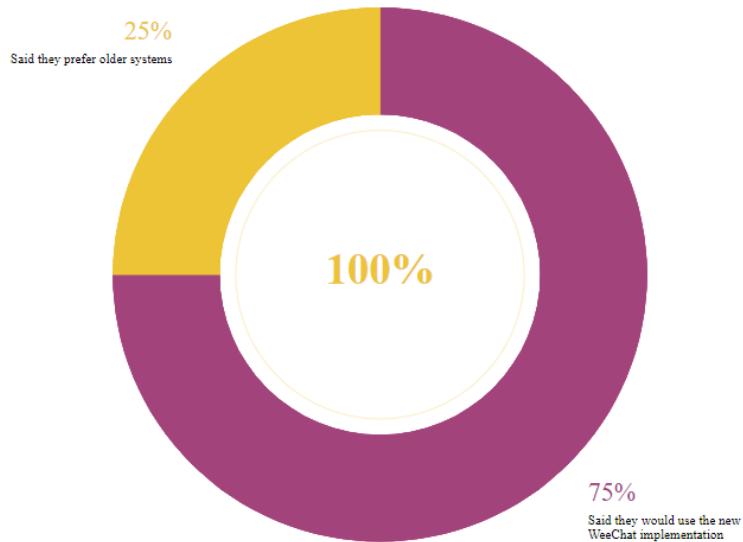
User Experience of WeeChat



The results show that the implementation was highly enjoyable to use, quite satisfying, very aesthetically pleasing and quite rewarding. Users stated this was due to the good contrast of the background with the content, the layout and divisions involved in the design and the immediate response they received when they performed a task. They also found it quite enjoyable to use with the implemented functionality. However, in search of achieving these goals, they found that there was a lack of help functionality as it is only implemented briefly using tooltips, and help icons implemented on the main server page and the direct messages page. In addition, the lack of functionality when they were displayed affected this result.

I also asked a number of users on whether they would rather use my current implementation, or an older IRC chat, and the results show 75% of people said they would rather use my implementation and 25% would rather use a different and older implementation.

WeeChat implementation vs Competitors



Overall, the results given by the group of users are considered positive and that their feedback is an indication that the implementation, without some incorporated functionality is considered a success and on the right track if any future development was to take place.

Test Cases

Contributor: Aaron Gangemi

Following my implementation of WeeChat, I have run a range of tests to test the implementations functionality.

Test Case ID: Case000

Test Case Name: Successful login into WeeChat.

Prerequisite: Username field is not empty and password field is greater than 8 characters

Steps:

1. Enter a valid username
2. Enter a valid password
3. Click "Login" and proceed to Main Server page

Expected Result: Upon successful login, the user will be taken to the Main Server page

Status: Pass

Test Case ID: Case102

Test Case Name: Successful creation of use account

Prerequisite:

1. User must enter a non-empty first name, last name, username
2. Email address must be in correct format
3. Password must be greater than 8 characters
4. Password must match confirmed password
5. Terms and Conditions box must be ticked

Steps:

1. User enters valid first name and last name
2. User enters valid username
3. User enters a valid email address in the correct format
4. User enters a password greater than 8 characters
5. User enters a matching password in the confirm password field.
6. User selects the Terms and Conditions field
7. User successfully chooses the username color of their choosing

Expected Result: User successfully proceeds to main server page

Status: Pass

Test Case ID: Case001

Test Case Name: User successfully able to send text to server channel

Prerequisite:

1. The main server interface is displayed
2. User is connected to a server and channel

Steps:

1. User types text they wish to send to chat
2. After finished typing user clicks "Send" button

Expected Result: Typed text appears in chat

Status: Pass

Test Case ID: Case002

Test Case Name: User successfully sends image file to channel

Prerequisite:

1. User must be on the Main Server page
2. User must have image files stored on device

Steps:

1. User selects the image icon in the text field
2. File explorer opens
3. User navigates to desired image file to send
4. User selects image file
5. File path appears in chat to represent image file sent

Expected Result: The current file path does appear in the channel chat

Status: Pass

Test Case ID: Case003

Test Case Name: The user is able to successfully send an attachment to the server channel

Prerequisite:

1. User must be on the Main Server page
2. User must have files stored on device

Steps:

1. User clicks attachment icon in main server interface
2. File explorer opens

3. User navigates to desired file
4. User selects file

Expected Result: Attachment file path appears in Main Server channel

Status: Pass

Test Case ID: Case004

Test Case Name: User successfully sends emoji to Main Server chat

Prerequisite: User is on main server page

Steps:

1. User clicks on emoji symbol in main server chat

Expected Result: “:)” appears in channel chat

Status: Pass

Test Case ID: Case005

Test Case Name: User clicks on server or channel name and name in server/channel changes

Prerequisite:

1. User must currently be on the main server page
2. Current servers and channels must exist in list

Steps:

1. User selects server name
2. User selects channel name

Expected Result: Server name successfully changes and channel name successfully changes

Status: Pass

Test Case ID: Case006

Test Case Name: User is able to successfully delete the server from the server list

Prerequisite:

1. User be on Main Server interface
2. Server must currently exist in order to delete

Steps:

1. User right clicks on server to delete
2. User selects “Delete” in context menu

Expected Result: The server should be removed from the list

Status: Pass

Test Case ID: Case007

Test Case Name: User is able to successfully add a server to the list

Prerequisite:

1. User has clicked the “Add Server” button in the main server interface
2. Server URL is not empty
3. Server URL is a valid URL
4. Server name is not empty

Steps:

1. User inputs valid server URL in correct URL format
2. User inputs non-empty server name to be displayed

Expected Result: User will be transitioned back to main server page and server name will be added

Status: Pass

Test Case ID: Case008

Test Case Name: User is able to visit any one of the pages when ¼ buttons in the top corner of any interface is clicked

Prerequisite: User must be on a page with the 4 icons in the top right corner

Steps:

1. User selects an icon in the top corner of a page
2. User is redirected to the corresponding page

Expected Result:

1. Script icon redirects to script manager page
2. World icon redirects to main server page
3. Send icon redirects to messages page
4. Settings icon redirects to options and settings page

Status: Pass

Test Case ID: Case009

Test Case Name: If the user clicks the account icon in the top left corner, they successfully transition to User information page

Prerequisite:

1. User must be on a page with account icon
2. User must be successfully logged in

Steps:

1. User clicks on icon

Expected Result: The user successfully transitions to the user information page

Status: Pass

Test Case ID: Case010

Test Case Name: The user is successfully able to edit their associated account information

Prerequisite:

1. User is currently on the user information page
2. Email should be in valid format
3. First name should not be empty
4. Last name should not be empty
5. Username should not be empty

Steps:

1. User selects the edit icon
2. User edits any desired information leaving all fields valid and not empty
3. User clicks edit icon again to stop editing

Expected Result: User fields are successfully updated. If username has been edited, then the username in the top corner is updated

Status: Pass

Test Case ID: Case011

Test Case Name: For all settings pages, all the left hand side buttons transition the user to the corresponding pages.

Prerequisite:

1. User must be on a settings page

Steps:

1. User selects a different settings button different to the page they are on

Expected Result:

1. User is transitioned to the corresponding settings page

- a. Script Manager transitions user to script manager page
- b. User information transitions user to user information page
- c. Options and settings transitions user to options and settings page

Status: Pass

Test Case ID: Case012

Test Case Name: User is able to delete account

Prerequisite:

- 1. User is on the user information page

Steps:

- 1. User selects "Delete Account"
- 2. Alert is displayed asking user to confirm deletion of account
- 3. User selects "Ok"

Expected Result: User is successfully transitioned to login screen

Status: Pass

Test Case ID: Case013

Test Case Name: User is able to logout successfully

Prerequisite: User is on a page with logout button

Steps:

- 1. User successfully clicks logout button
- 2. Alert appears asking user to confirm logout
- 3. User selects "Ok" to alert

Expected Result: Program displays login screen

Status: Pass

Test Case ID: Case014

Test Case Name: User is able to exit weechat successfully

Prerequisite:

- 1. User is on a page with exit WeeChat buttons

Steps:

- 1. User clicks exit icon
- 2. Alert appears to confirm exit
- 3. User selects "Ok" to exit

Expected Result: Program closes

Status: Pass

Test Case ID: Case015

Test Case Name: User selects connected username in main server

Prerequisite:

- 1. User must be on main server page
- 2. User must be connected to a server and channel

Steps:

- 1. User clicks on other connected user

Expected Result:

- 1. User is transitioned to direct messages page
- 2. Connected username that was clicked on is displayed as recipient name

Status: Pass

Test Case ID: Case016

Test Case Name: User is able to upload script locally from device

Prerequisite:

1. User is on the script manager page
2. User has files to upload to table

Steps:

1. User clicks upload button
2. File explorer opens
3. User navigates to script and selects file
4. Alert appears asking for user to enter command
5. User enters the command name and selects the "Ok" button
6. User enters the description name and selects the "Ok" button

Expected Result: Script name, date, inputted description and command is successfully uploaded to table.

Status: Pass

Test Case ID: Case017

Test Case Name: User is able to search script online

Prerequisite:

1. User is on the script manager page
2. User has a URL to search script in correct format

Steps:

1. User types in valid script URL
2. User clicks search button
3. Alert appears asking for user to enter command
4. User enters the command name and selects the "Ok" button
5. User enters the description name and selects the "Ok" button

Expected Result: Script name, date, inputted description and command is successfully uploaded to table

Status: Pass

Test Case ID: Case018

Test Case Name: Collapse and Expand content tabs

Prerequisite: User must be on the options and settings page

Steps:

1. User clicks any of tabs to collapse/expand

Expected Result: Tab successfully collapses/expands

Status: Pass

Test Case ID: Case019

Test Case Name: Tick and Untick notification boxes

Prerequisite: User must be on options and settings page

Steps:

1. User clicks on notifications boxes

Expected Result: Boxes are no longer ticked

Status: Pass

Test Case ID: Case020

Test Case Name: Customisations options change

Prerequisite: User must be on the options and settings page

Steps:

1. User clicks on a list box that they would like to modify
2. User selects desired option

Expected Result: Change is displayed in option list

Status: Pass

Test Case ID: Case021

Test Case Name: Send Feedback

Prerequisite: User must be on the options and settings page

Steps:

1. User expands feedback tab
2. User types feedback in text field
3. User clicks submit

Expected Result: Alert appears stating feedback has been sent

Status: Pass

Test Case ID: Case022

Test Case Name: Click on username

Prerequisite:

1. User must have existing users in message list
2. User must be on Direct Messages page

Steps:

1. User clicks on username

Expected Result: Recipient name successfully changes

Status: Pass

Test Case ID: Case023

Test Case Name: Send direct message to chat

Prerequisite:

1. User is on direct messages page
2. User is connected to another user

Steps:

1. User types text into chat
2. User clicks send

Expected Result: Text appears in chat with date and time sent

Status: Pass

Test Case ID: Case024

Test Case Name: User successfully sends image file to chat

Prerequisite:

1. User must be on the direct messages page
2. User must have image files stored on device

Steps:

1. User selects the image icon in the text field
2. File explorer opens
3. User navigates to desired image file to send
4. User selects image file
5. File path appears in chat to represent image file sent

Expected Result: The current file path does appear in the chat

Status: Pass

Test Case ID: Case025

Test Case Name: The user is able to successfully send an attachment to the chat

Prerequisite:

1. User must be on the direct messages page
2. User must have files stored on device

Steps:

1. User clicks attachment icon
2. File explorer opens
3. User navigates to desired file
4. User selects file

Expected Result: Attachment file path appears in the chat

Status: Pass

Test Case ID: Case026

Test Case Name: User successfully sends emoji to chat

Prerequisite: User is on direct messages page

Steps:

1. User clicks on emoji symbol in chat

Expected Result: “ :) ” appears in channel chat

Status: Pass

Test Case ID: Case027

Test Case Name: Help icon clicked

Prerequisite: User must be on page with help icon

Steps:

1. User clicks on help icon

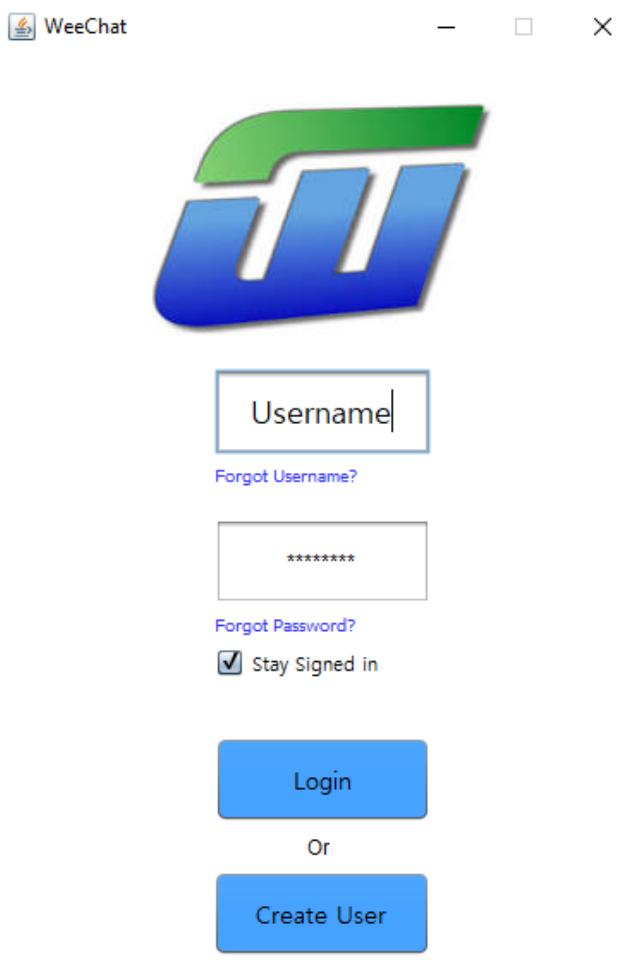
Expected Result: Help alert appears

Status: Pass

Aaron Giles

Implementation Discussion

Upon first opening the WeeChat application a simple login screen will be prompted. The idea and reasoning behind the simple login was to minimise clutter and make for easy navigation through the process. After feedback was received the colour scheme of the application has been changed since the high fidelity prototypes. The blue buttons were kept to simply add a small bit of colour, however, for the most part everything kept white. Ideally in the future, a dark theme would be an option, however, for the mean time only the light theme is available. The resizable option has been disabled, since the user should not need anything larger or smaller than what is given.



Based on feedback basic tool tips have been added to the password and username box to avoid any confusion. The tooltips avoid unnecessary clutter labels displaying what eachbox does, as it should be relatively easy to work out.



Username

[Forgot Username](#)

[Forgot Password?](#)

Stay Signed in

Or

If the login option is selected you will be automatically redirected to the main interface (currently no authentication for login), however if the create user option is selected be brought to the register page.



Accept Terms and Conditions

The register page has once again had the option to resize the window disabled due to the user not really needing much else on the screen and to avoid awkward sizes of boxes. The layout has been given tooltips for all of its text boxes to display what is needed in the boxes incase of any confusion.



Email Address

Password

Display Name

Accept Terms and Conditions

Register

The terms and conditions box needs to be clicked to continue onto the next layout which is the main interface, if the user clicks the register button without a text will display directing the user to accept it before continuing.



Email Address

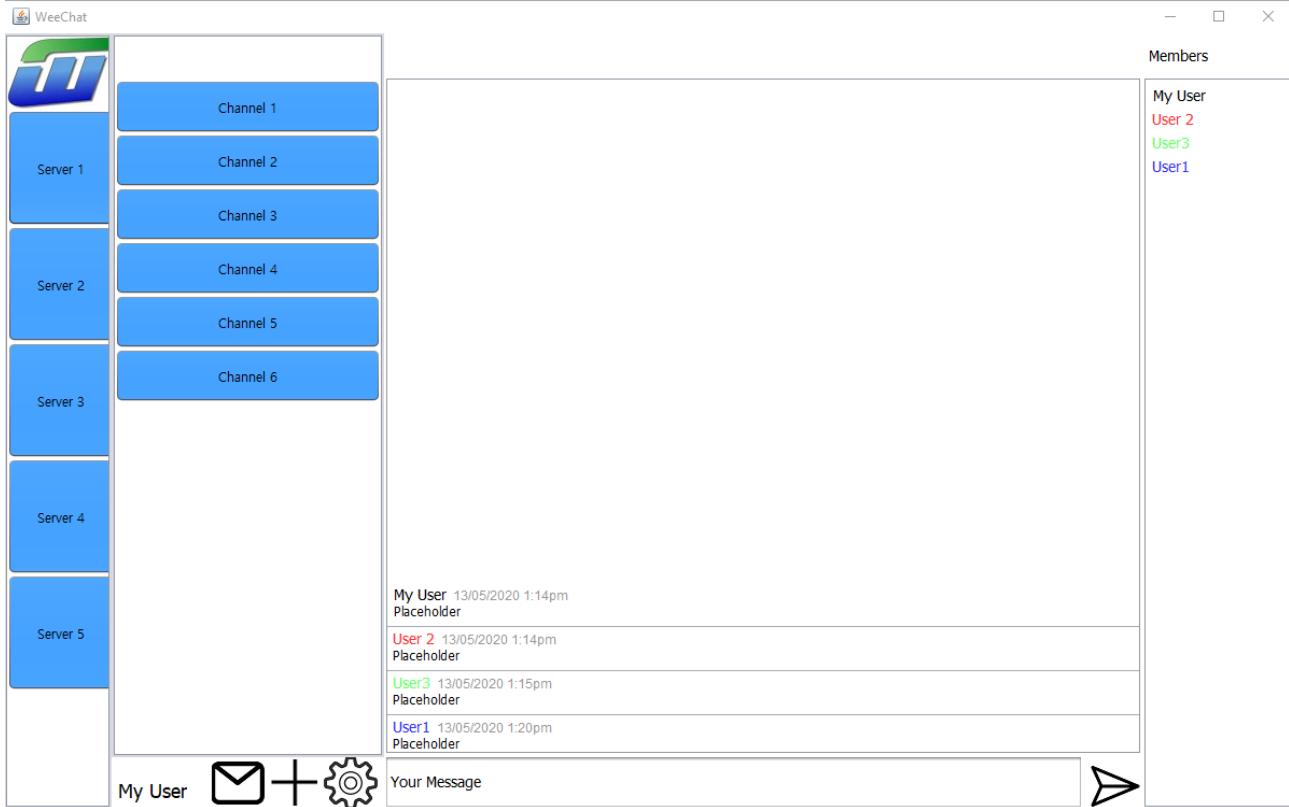
Display Name

Accept Terms and Conditions

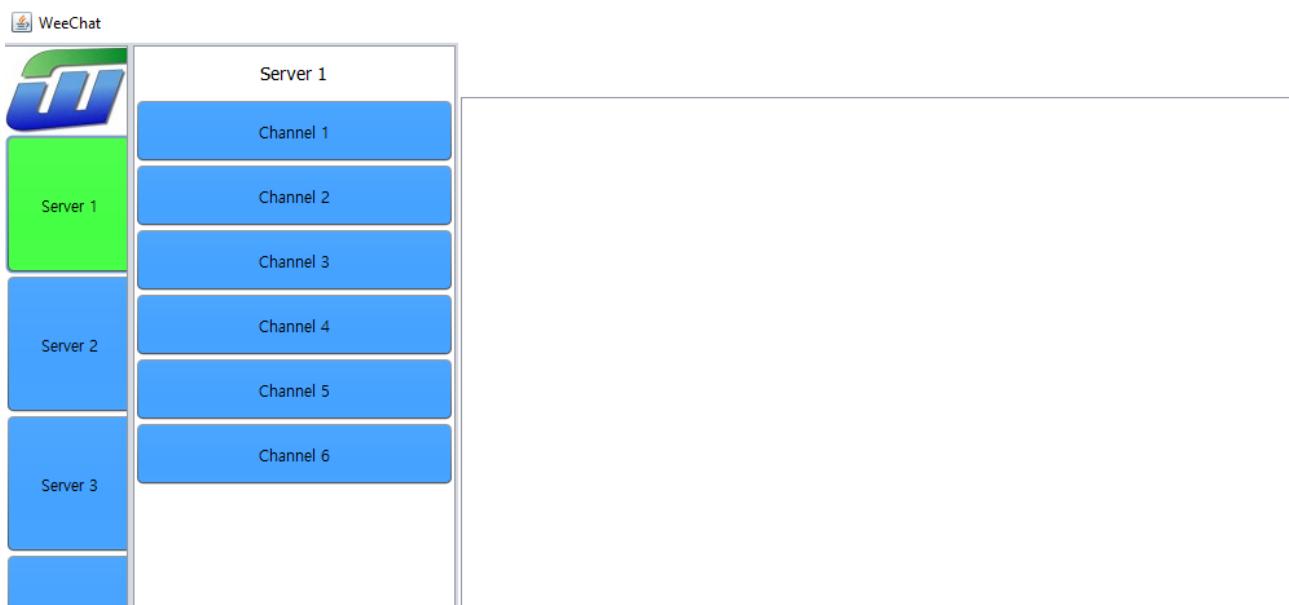
Register

Accept Terms and Conditions first

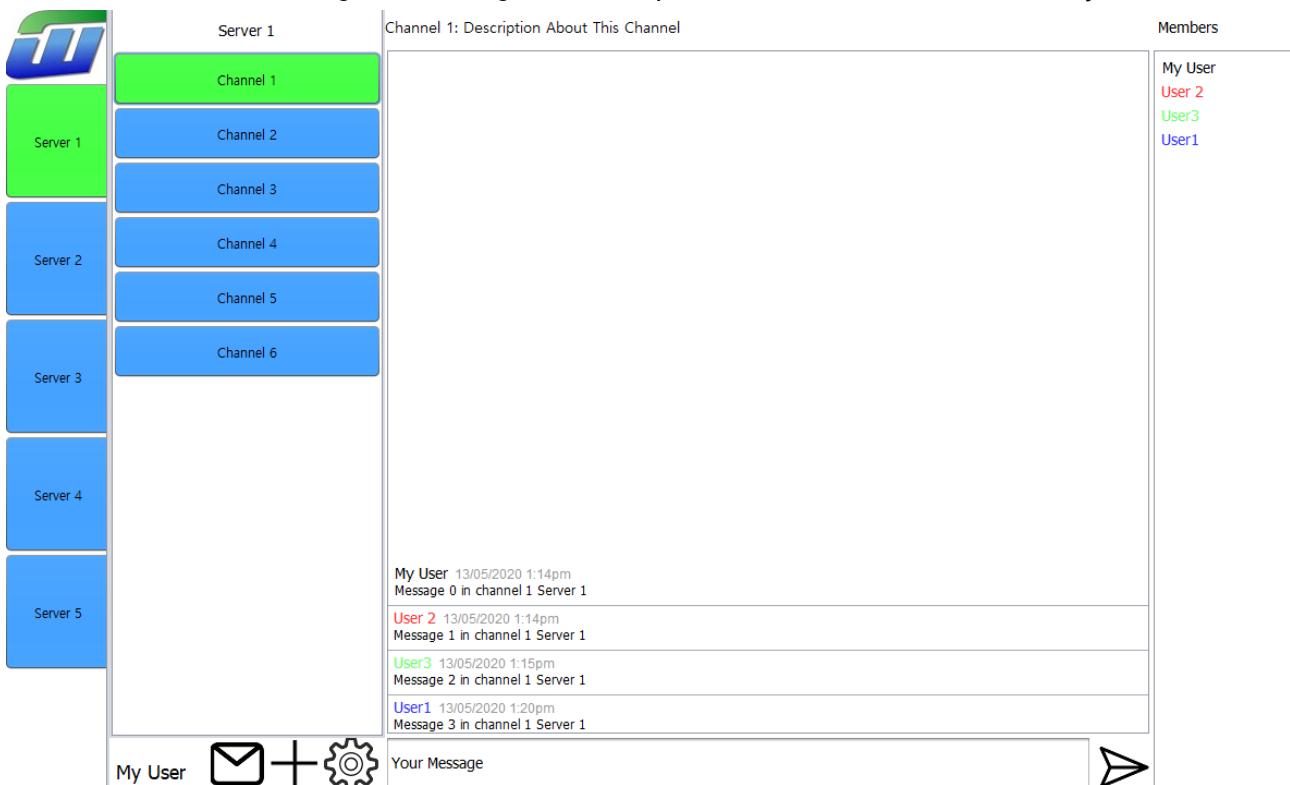
Once the login and/or register page has been navigated through the user will be directed to the main interface, which displays all messages and contains the functionality of the program. The colour scheme currently is a slight modification on the prototypes with a lighter blue and green to reduce contrast on the text. When initially arriving no server or channel is selected and currently just given a placeholder. The window is resizable and all elements will readjust accordingly.



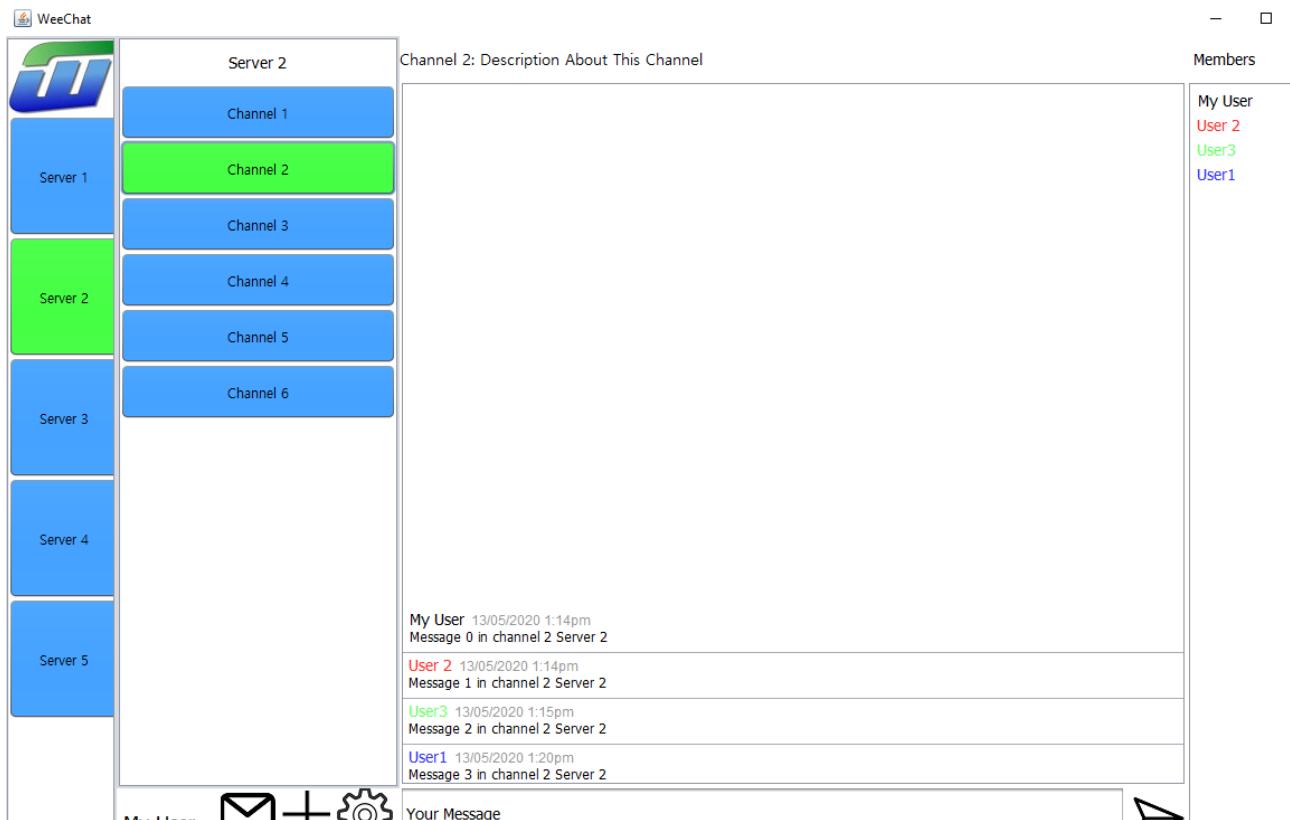
Feedback to the user has been added based on the feedback, whenever a user clicks on a server the currently selected server will be in green. The server will also be displayed in text above the channel list.

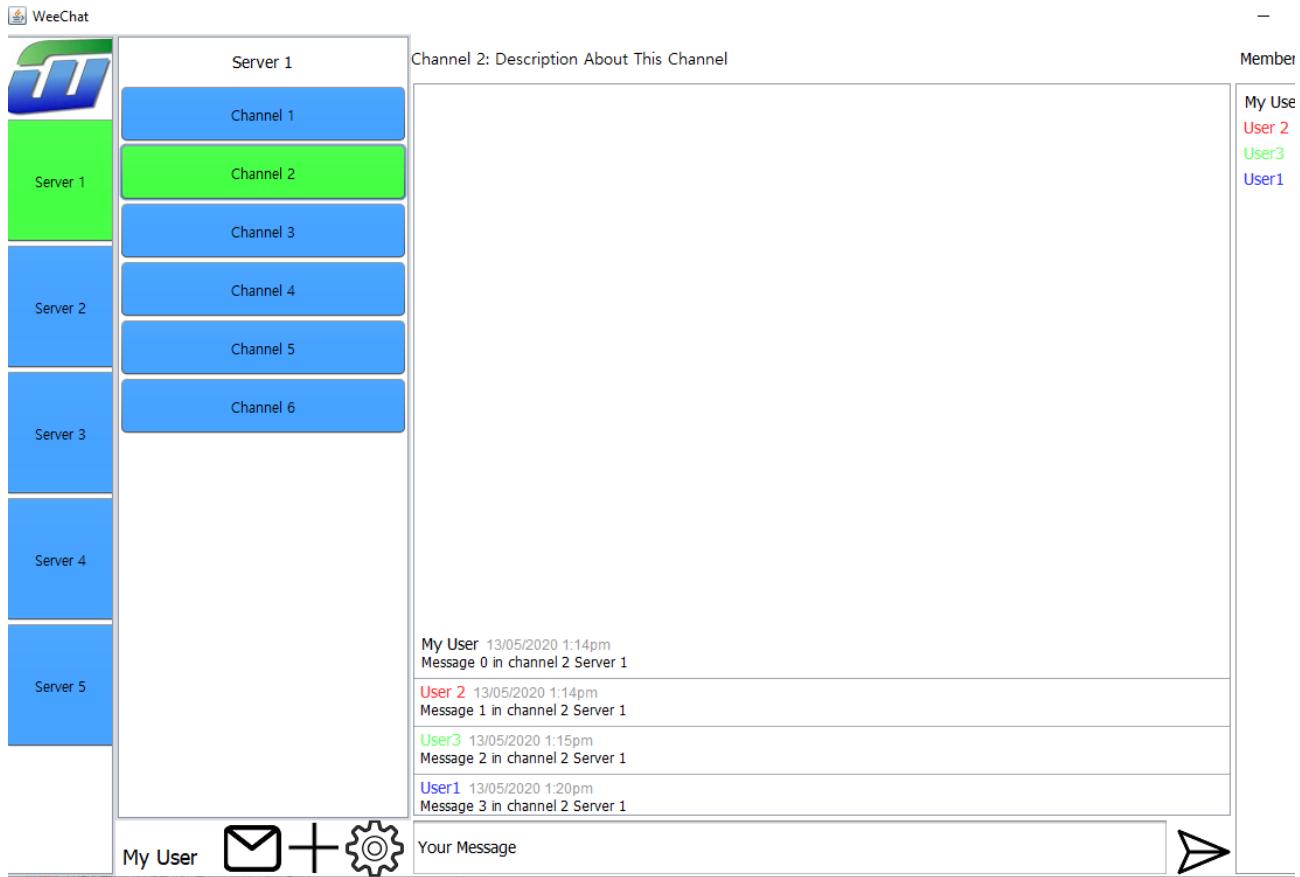


Similarly, once a channel is selected, the selected channel will highlight and display in text what channel the user is viewing. All messages will be updated and text/font sizes and styles.



When a server/channel is changed shown below.

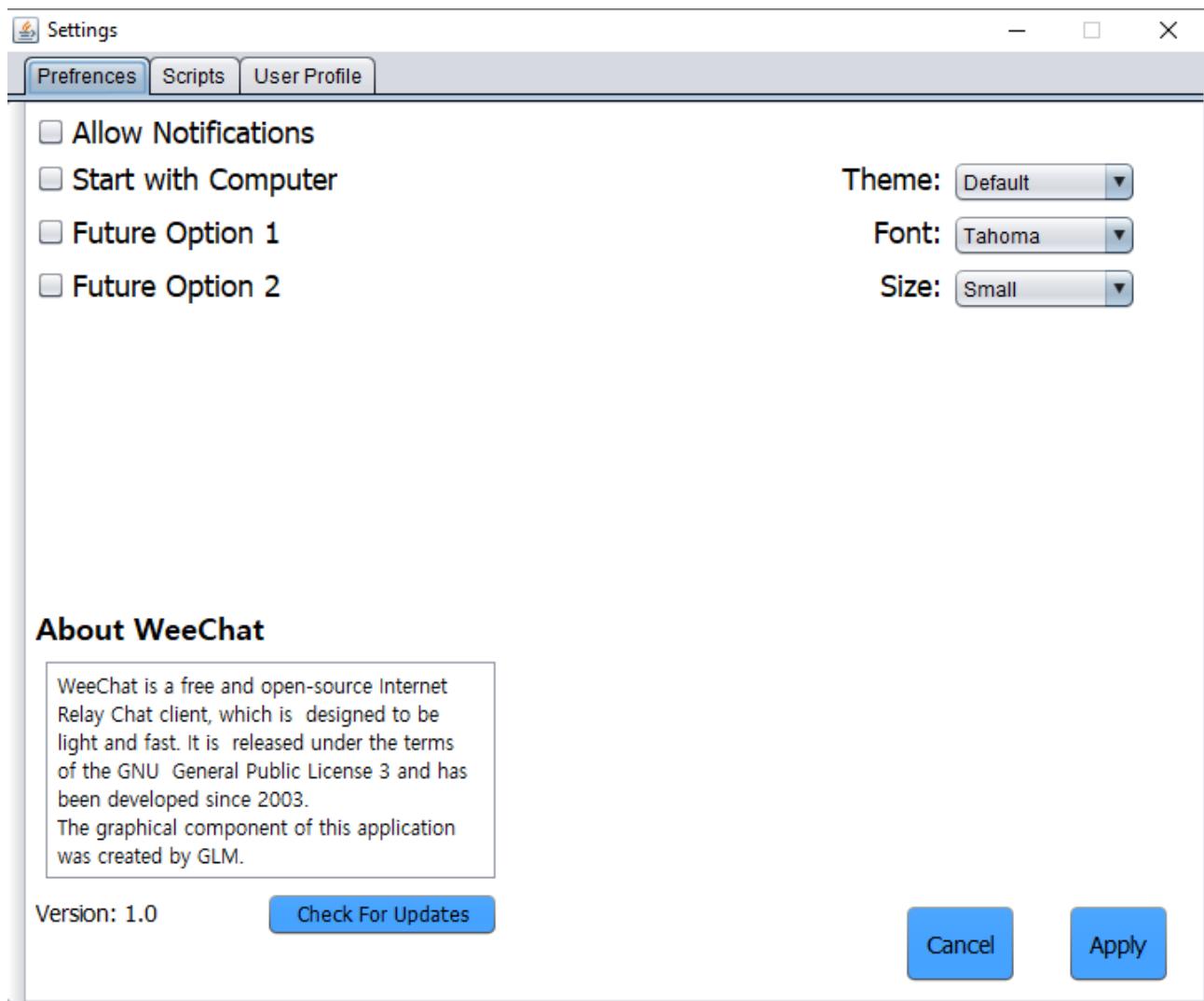




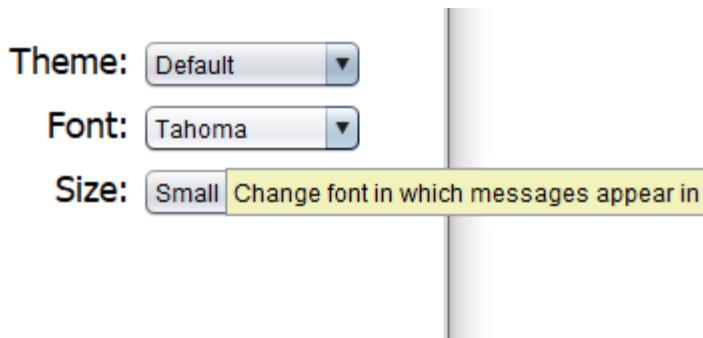
Based on feedback, tooltips were given to all the options/buttons down in the bottom left, if any of them are hovered over, the corresponding tooltip telling the user where it leads to.



If the settings option(the cog) is selected, a new pop up window will come onto the screen. The idea of this was to keep it as little clutter as possible for easy navigation of the settings menu. There are 3 sections the settings menu has been broken up into. Preferences, everything relating to how the look and feel of the application feels, which unfortunately the theme feature did not receive and implementation. Scripts, which is essential for adding and removing custom scripts. Currently not functionally although it provides what it would look like in future implementations. Finally the user settings, relating to the user's personal account settings, such as password and email. It was decided to split these up to avoid conflicting/not similar information on the same page. They are split up using tabs, the tab selected will be different and clearly visibly to be selected to provide the user feedback on they actually clicked something.

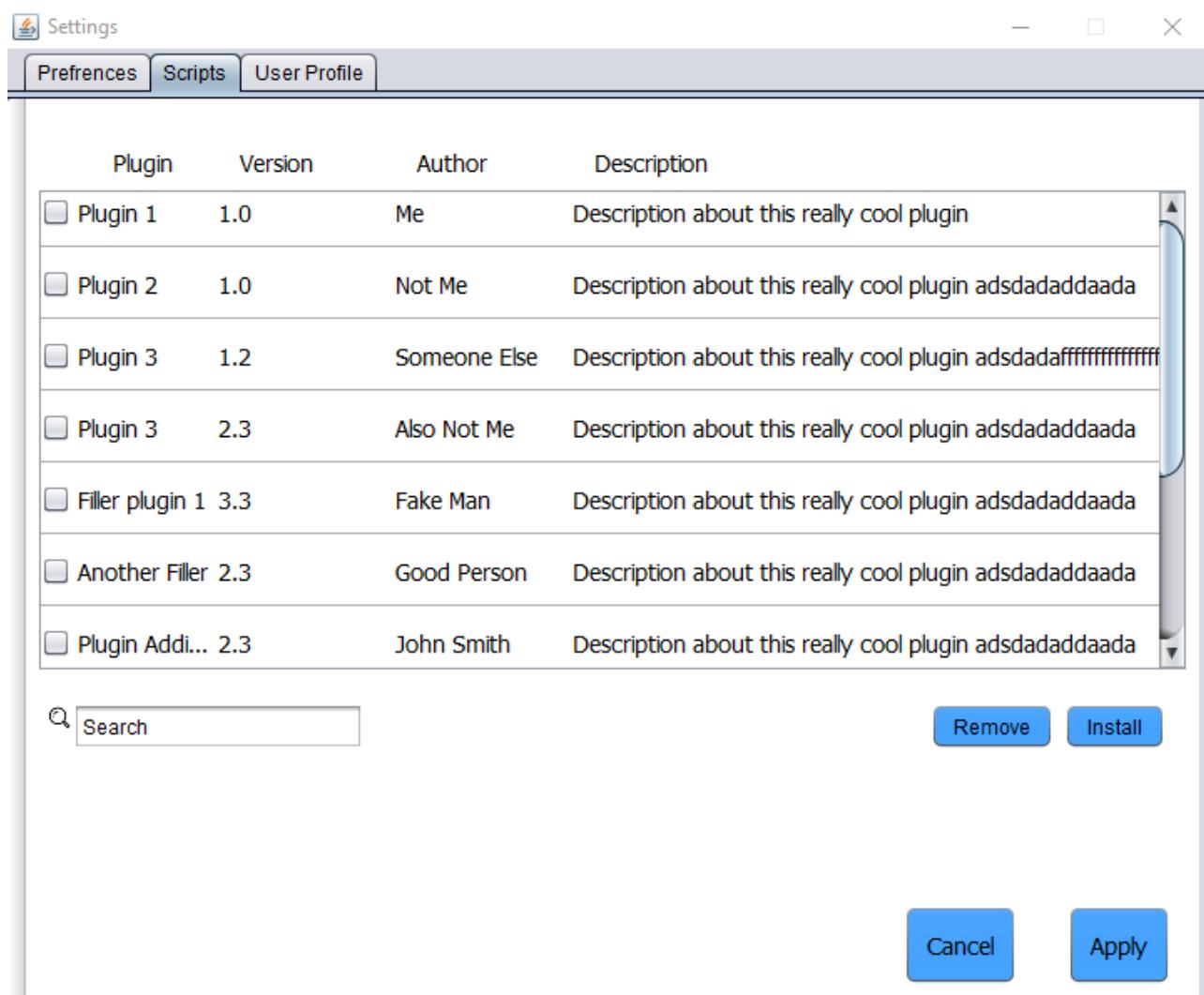


All of these options have tooltips as suggested by the feedback from the prototypes. The tooltips provide a brief description of what the setting does. The font and font size features are working however the others are not.

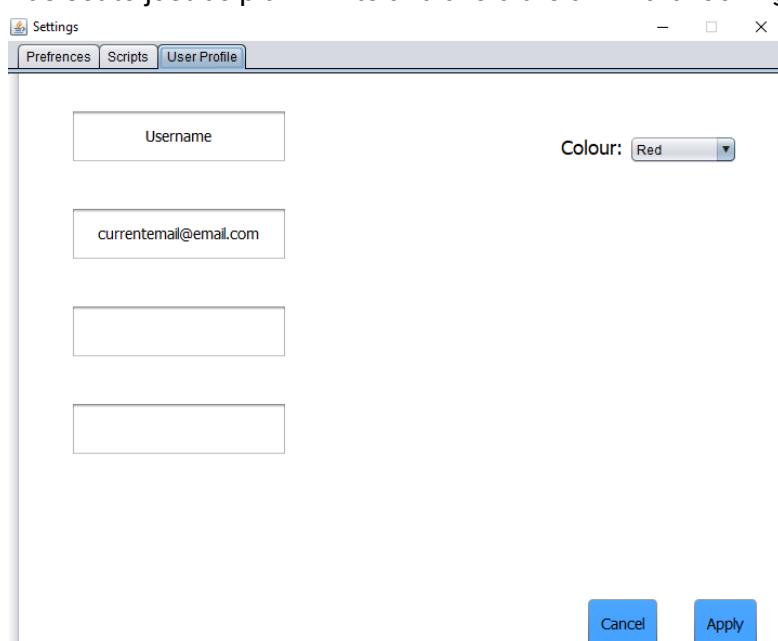


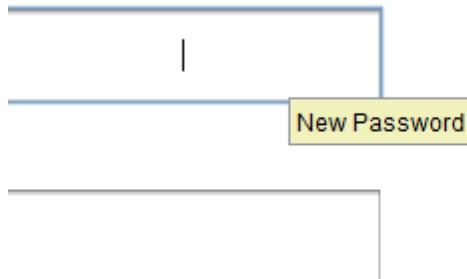
The check for updates when clicked will display a message/feedback to the user saying it is currently updated.

If the scripts tab is selected a scroll window displaying all the available scripts will appear, the scripts tab is currently not functional however provides the framework on which it would look like. Providing a brief description, author, version and name, saving the user time on researching and all being available in the click of a button.

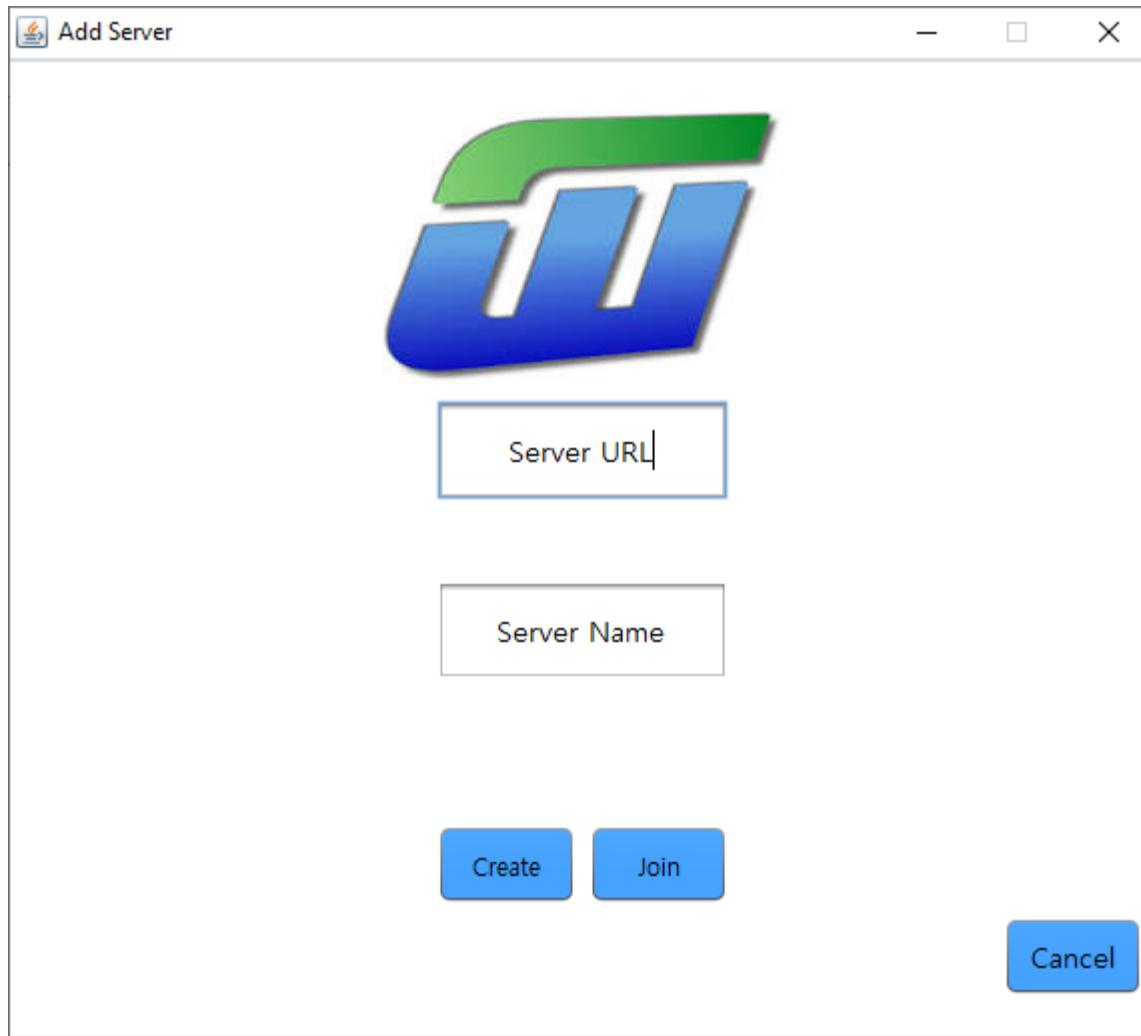


The user profile section provides the user options to change settings and details. With tooltips and easy navigation. No unnecessary information is visible. Also, based on feedback colour scheme was set to just be plain white and avoid the awkward looking green and blue where unnecessary.

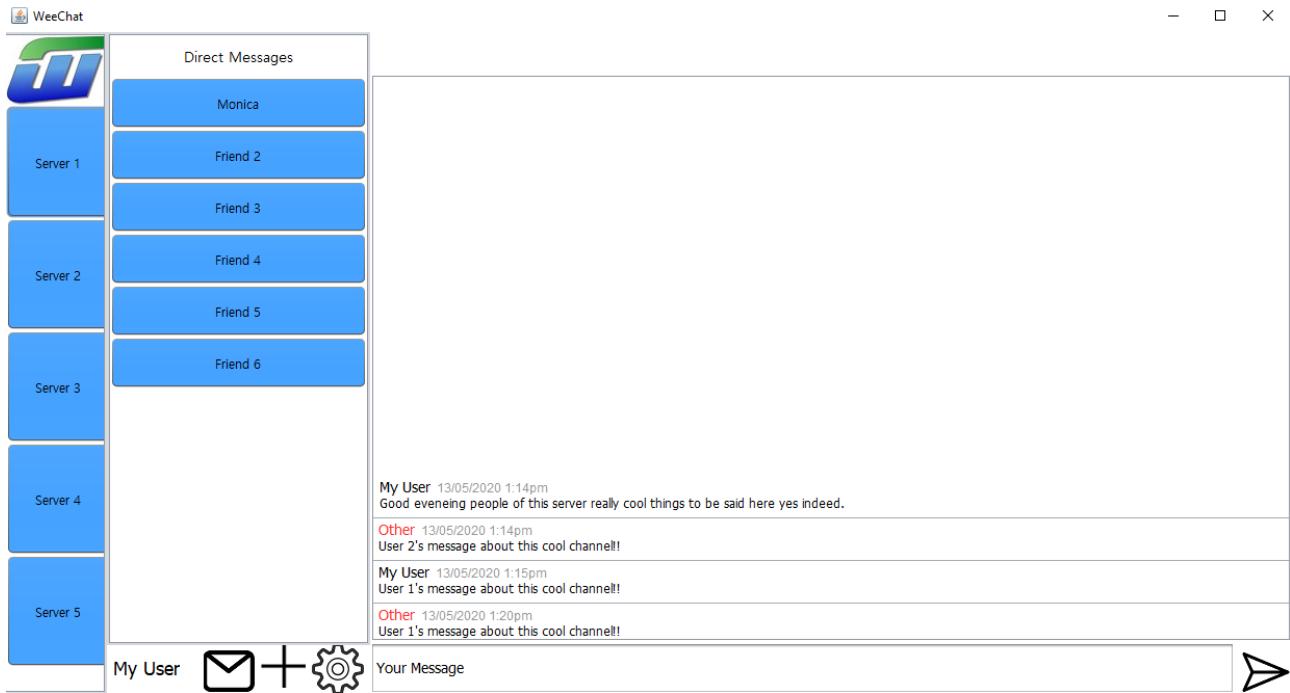




When the settings menu is closed, via clicking apply or close, the window will close, while still having the main interface open in the background. If from the main interface the user then selected the "Add Server" button(the plus icon) they will be directed to a currently unfunctional add server page. Very simple interface, that has had the resizable option disabled to avoid awkward scaling. The interface is simple and should be relatively straight forward. Currently the only button that does anything is the cancel, which will close the window, the window opens separately from the main interface/direct messages so after closing you'll be back into the previous window.



The direct messages button will direct the user to messages that only that user can see, independent to any server. The interface is very similar to the main interface, however, lacks a members list, due to the irrelevance of it and to avoid clutter. The server menu is still available on the side, this keeps the look and makes easy navigation back to the servers. All menus are blue indicating none are selected, however, if they are selected the user will be redirected to the main interface with the servers. When arriving, no friend is selected, hence no messages visible, some placeholder text is there for no until a better implementation can be completed. The colour scheme/theme was decided to keep similar through the program, even though the bright colours may cause eye strain, a future implementation would have a darker theme.

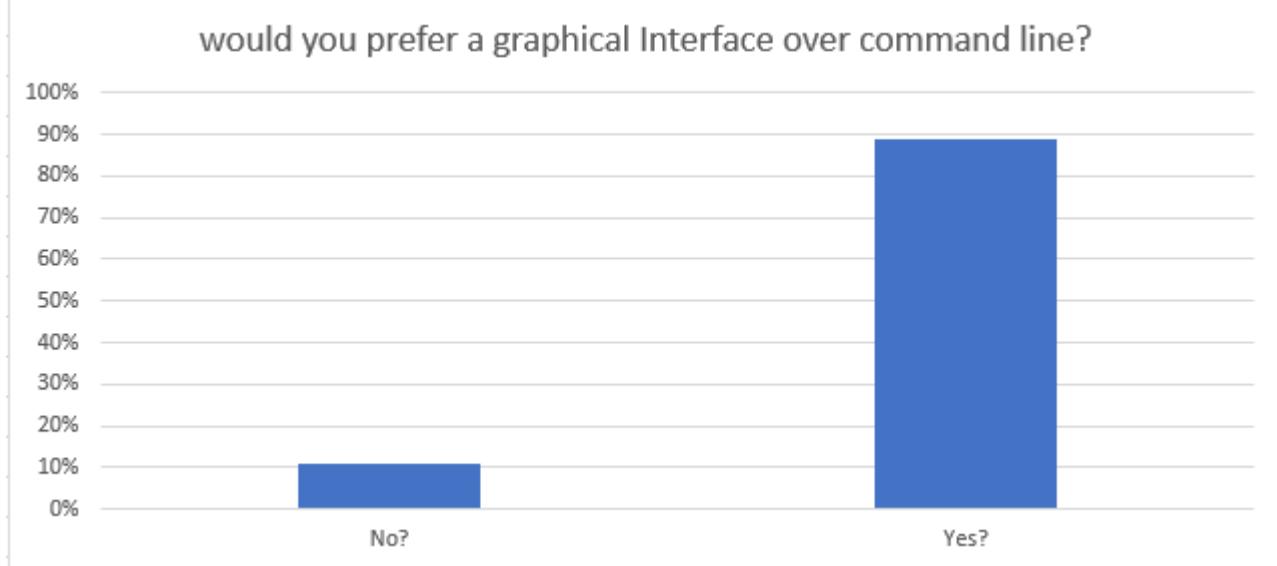


Once a friend is selected the usernames will change and messages accordingly. The functionality of this is limited although the idea of how the basic messaging platform would work.

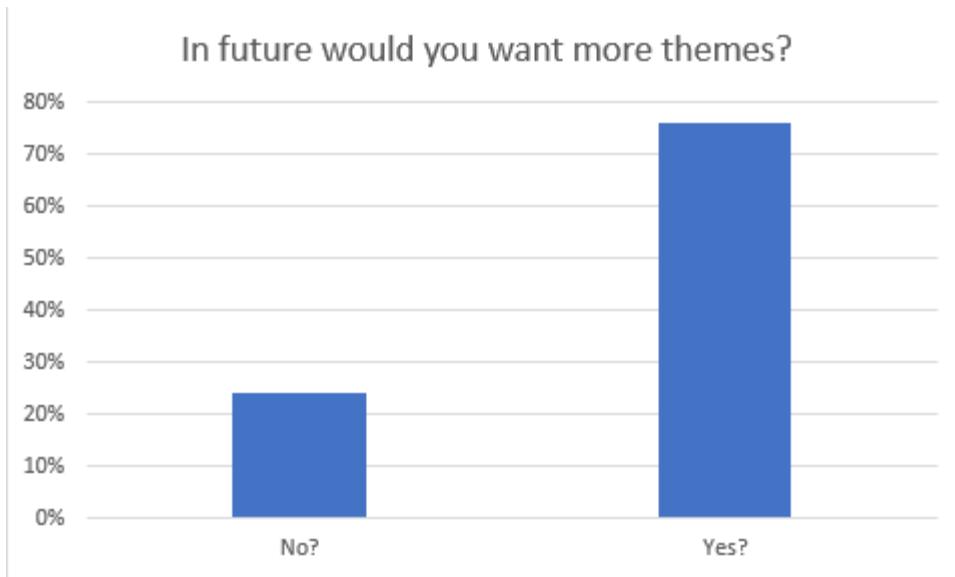


User Involvement

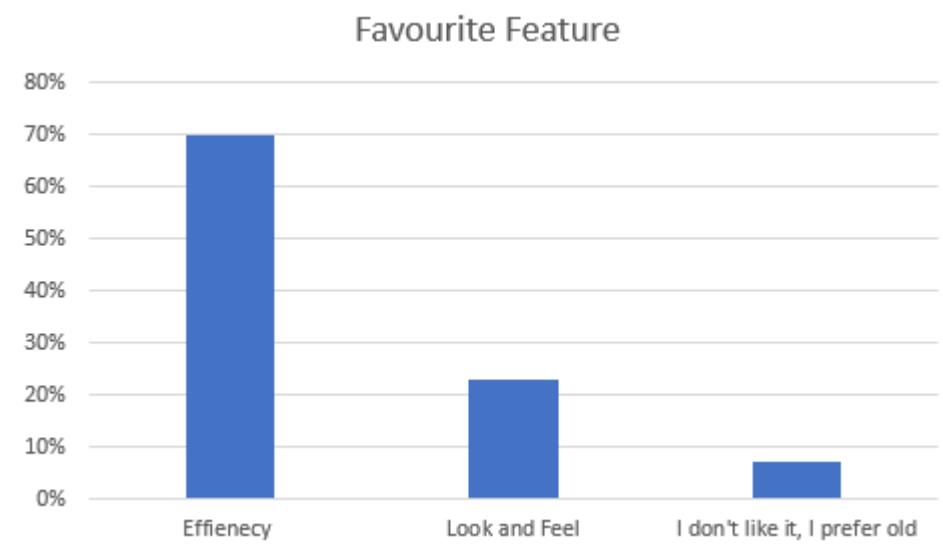
Throughout development constant user involvement and feedback was gathered. Through surveys and simple questionnaires to improve the design and appeal to new users. Simple yes or no questions were asked to stakeholders. Firstly we asked if users would even prefer a graphical interface for their application, and with a relatively large majority the decision was that it was wanted/needed.



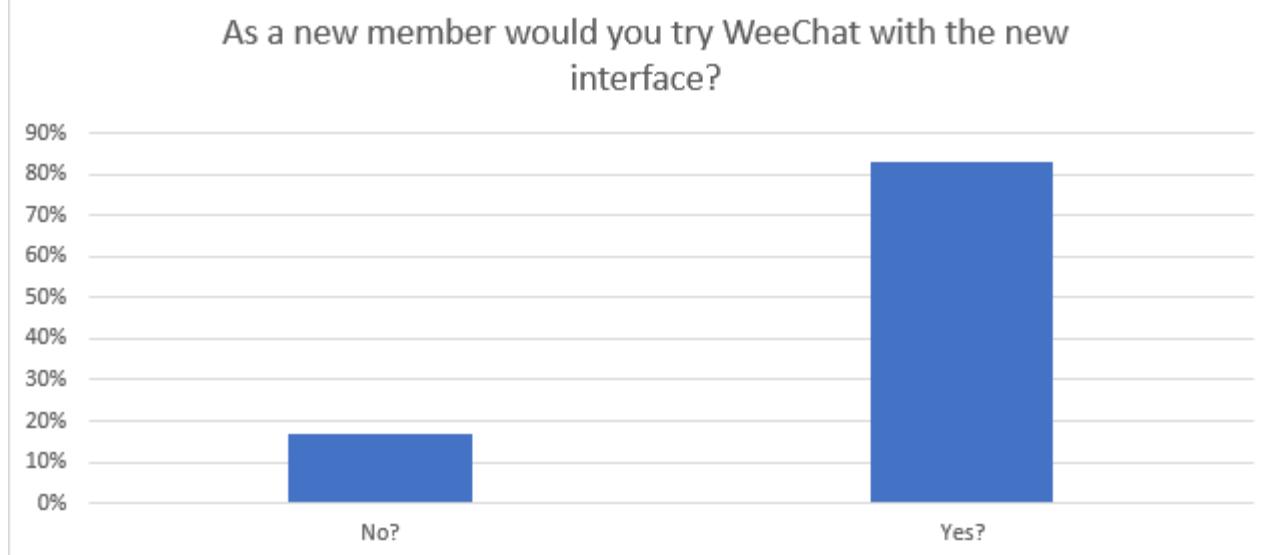
Secondly we asked if users would prefer new themes in the future, and with the majority of the feedback saying yes they would, hence the theme feature was added, without an implementation just yet. Further investigation into what themes would be needed however, a good starting point.



Once the final implementation was designed and completed, we asked those same users, what was their favourite feature about the new graphical interface. With the large majority agreeing on the efficiency of the new interface feature is a great benefit. Other options such as the look and feel were slightly favoured in 2nd, some users really liked the new look. However, a small portion said they did not like the new interface at all and prefer the older one, which is understandable, not everyone will agree, the option to leave the old command line interface is still available.



A new sample population was selected for the next survey, with every person not having used WeeChat at all in the past. Each person was shown the new graphical interface, and asked a simple yes or no question, if they would use this application, not knowing there is another command line only version of the app. The large majority said yes, they would use the application.



Those same members were shown the original command line version of the application and the results significantly changed based on it. Due to the confusion to non-technical users the command line had brought on.

As a new member would you try WeeChat with the new interface?



Test Cases

Name: 1

Details: When the login button is selected you will be brought to the main interface screen.

Steps:

1. Open the login window
2. Click the login button

Expected Result: The main interface should pop up.

Pass/Fail: Pass

Name: 2

Details: When the register/create user button is clicked the register interface should pop up.

Steps:

1. Open the login window.
2. Click the register/create user button.

Expected Result: The register interface should pop up.

Pass/Fail: Pass

Name: 3

Details: Register button clicked and terms and conditions accepted should go to the main interface.

Steps:

1. Tick the TOS box.
2. Click register.

Expected Result: The main interface will pop up.

Pass/Fail: Pass

Name: 4

Details: Clicking the register button without terms and conditions ticked

Steps:

1. Click the register button without ticking the box.

Expected Result: A text should appear directing users to accept terms and conditions.

Pass/Fail: Pass

Name: 5

Details: Clicking a server should light up in green and the rest should be blue.

Steps:

1. Click on the server you want selected.

Expected Result: Server selected box will turn green, all others blue. The server name should also appear above the channels.

Pass/Fail: Pass

Name: 6

Details: Selecting a channel

Steps:

1. Select the server you want.
2. Select the channel you want.

Expected Result: Channel selected box should be green, all others green and text messages for that channel should appear. The channel name/description should appear at the top of the messages box.

Pass/Fail: Pass

Name: 7

Details: Clicking buttons.

Steps:

1. Navigate to the main interface/direct messages.
2. Click on buttons in the bottom left.

Expected Result: Corresponding window should open based on which button clicked. Eg. settings button clicked, settings should appear.

Pass/Fail: Pass

Name: 8

Details: Navigate to direct messages.

Steps:

1. Click on the mail icon in the bottom left.

Expected Result: The new direct messages interface should appear.

Pass/Fail: Pass

Name: 9

Details: Change settings, font and size.

Steps:

1. Navigate to the settings page.
2. Select new font and font size.
3. Click apply.
4. Switch between channels to refresh text.

Expected Result: The text/size should change as stated.

Pass/Fail: Pass

Name: 10

Details: Navigating between settings tab.

Steps:

1. Navigate to settings.
2. Click on other tabs to go to different screens.

Expected Result: Other settings menus should appear, such as scripts and user profiles.

Pass/Fail: Pass

Name: 11

Details: Clicking cancel/apply

Steps:

1. On the settings click the cancel button.

Expected Result: The window should close, although keep the main interface/direct messages in the back.

Pass/Fail: Pass

Name: 12

Details: On the direct messages if the server is clicked it should redirect the user out to the main interface not direct messages.

Steps:

1. In direct messages click on any of the servers in the list.

Expected Result: The main interface with the servers should appear.

Pass/Fail: Pass

Name: 13

Details: Clicking the update button

Steps:

1. In the settings, preferences, click on the check for updates button.

Expected Result: Message should display about current availability of the update.

Pass/Fail: Pass

Name: 14

Details: Accessing buttons from direct messages page.

Steps:

1. Click the buttons in the bottom left of the direct messages page.

Expected Result: Various windows should open based on which button clicked.

Pass/Fail: Pass

Name: 15

Details: Text/size changes in direct messages after settings applied.

Steps:

1. Click on the settings button.
2. Change the text sizes and hit apply.
3. Refresh text by going in between channels.

Expected Result: Text should be new size/style.

Pass/Fail: Pass

Name: 16

Details: Navigating between friend's direct messages

Steps:

1. Click on another friend to direct message.

Expected Result: The new friend selected should light up in green and all others blue and have their name appear at top.

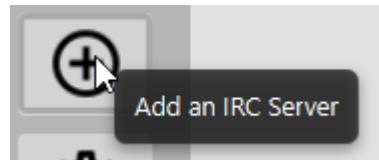
Pass/Fail: Pass.

Kevin Le

Implementation Discussion

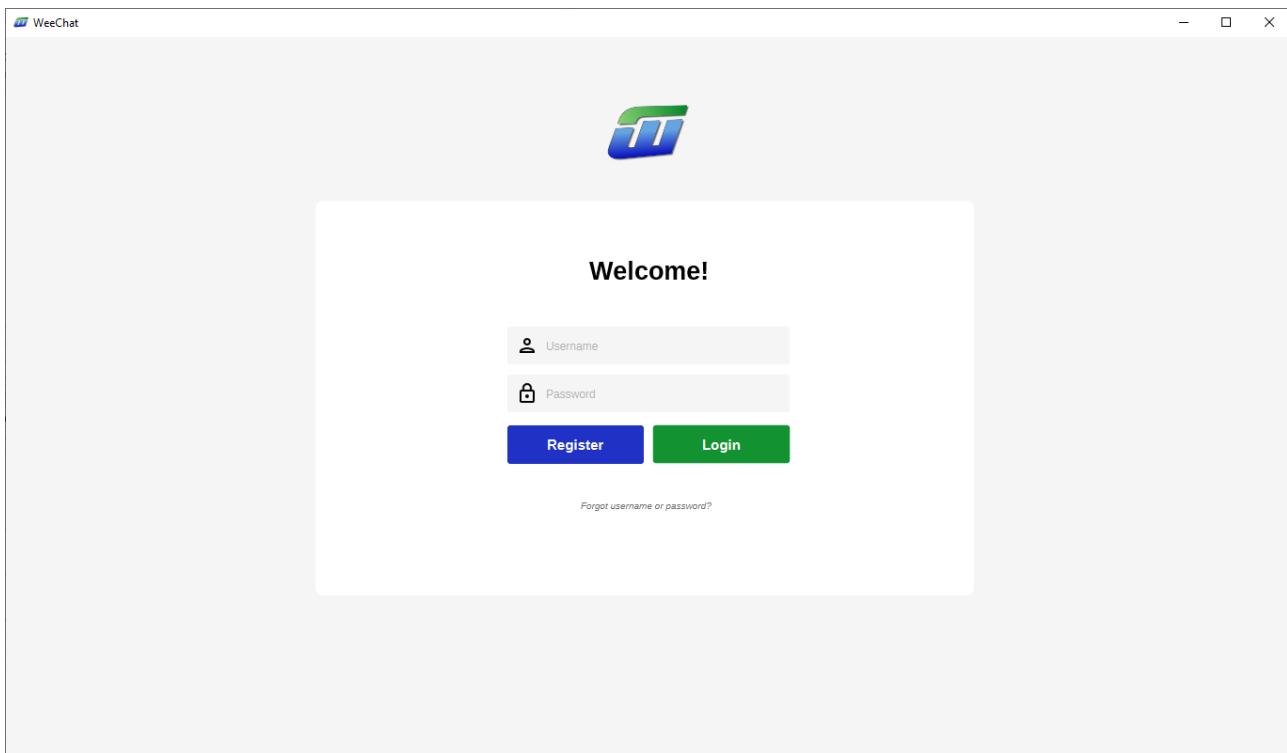
After discussion with the clients, they gave the team the green light to proceed with the implementations, with a few suggested modifications. I was able to accommodate most of them, with some being out of reach given the project deadlines. The program does not have actual functionality. The font used in the prototypes was Roboto, but to ensure multi OS compatibility, it was changed to Liberation Sans.

I was able to incorporate an application icon, title and the noted missing minimize, maximize, and exit buttons. I additionally provided tooltips on all the non textual buttons as suggested.



Suggestions that I was not able to incorporate was a dark theme and foldable sidebars due to time constraints.

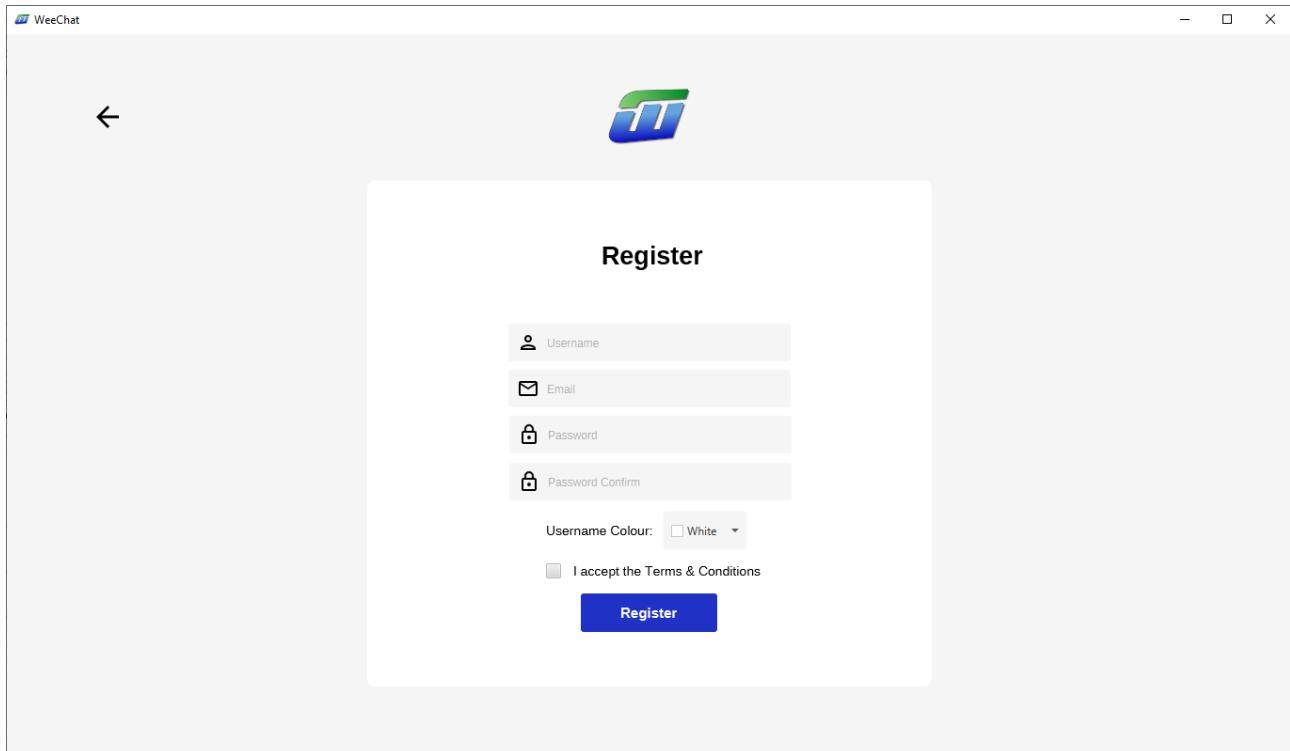
Upon loading the program, the user is presented with the login screen. This remained mostly unchanged from the high fidelity prototypes as the client expressed no issues with it. The “shadow effect” from the high fidelity was not incorporated in order to make the program more “flat”, for easier viewing. If “Login” is clicked, they will be taken to the server page. Forgot username/password not implemented as that is expected to open a website.



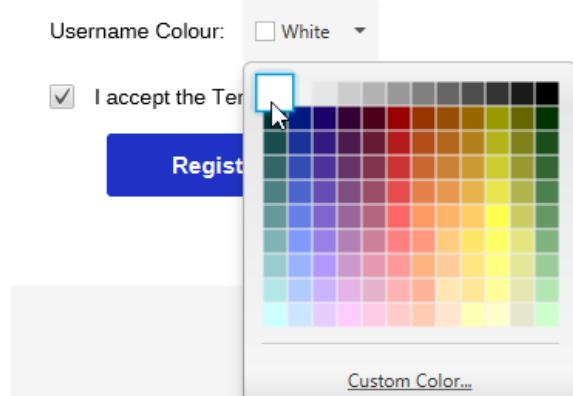
All buttons have on hover and on click attributes, to help the user understand where their cursor is positioned, and what button will be clicked or has been clicked. This button feedback is important in order to provide a more interactive experience.



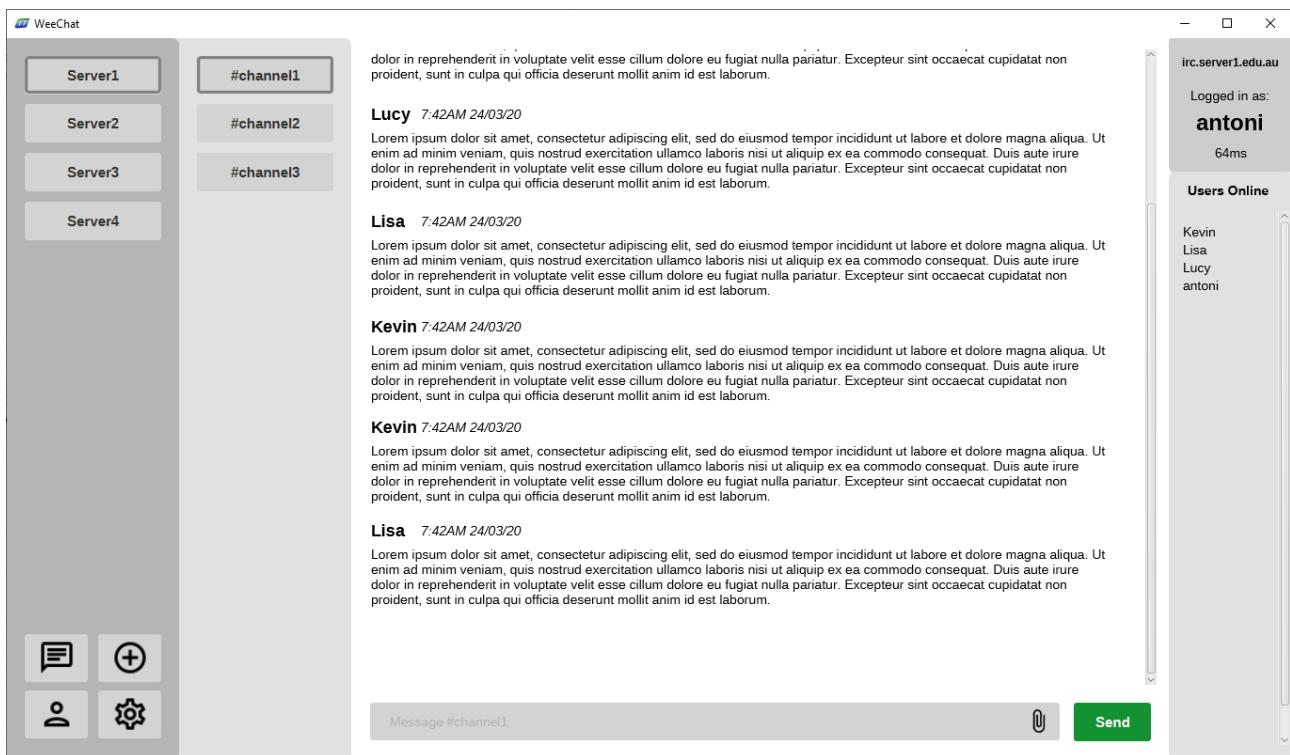
The left button has the cursor hovering over, and a border is shown, while the right button has been clicked, and the entire button darkens to indicate so. All buttons will darken based on their current colour. E.g. blue will have a dark blue border on hover, and be fully dark blue on click. Only 4 button colours were chosen to keep things simple. Blue, Green, Red and Grey. Red is for destructive actions, Blue is for actions that are secondary to a Green button, which is the most important indicating a next step or “to proceed”.



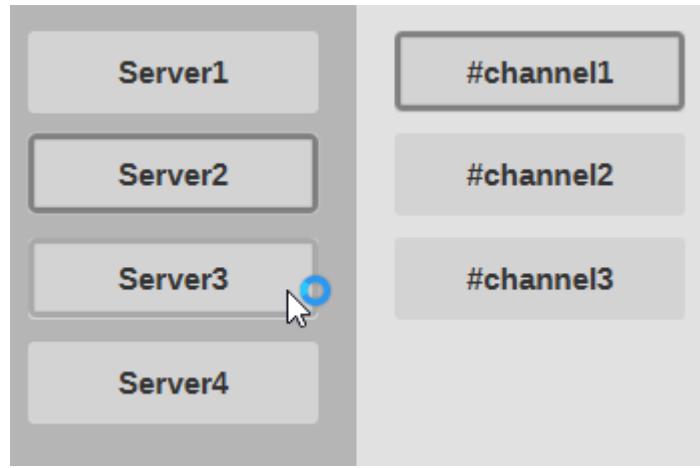
The register page also remained mostly unchanged, apart from the elimination of the shadow in favour of a more flat design. Like the login, the text fields have the same colour as the background and the Register button blue is taken from the shade of blue in the “Weechat” logo, this was done to reduce the amount of colours and clutter on the page, keeping a clean design. When “Register” is clicked, they will be taken to the server page.



The colour picker is the standard JavaFX inbuilt picker.



The server UI also remained mostly unchanged, however, the connection information and current user logged in located in the top right underwent some modification, the text is now centred, and instead of a connection bar, the ping to server is shown in milliseconds. The menu buttons and their corresponding icons in the bottom left were kept the same from the high fidelity prototypes.



The currently selected server is marked by a grey border, but if the user were to select a different server, the mouse hovering over the button would mark what server they will select if clicked.

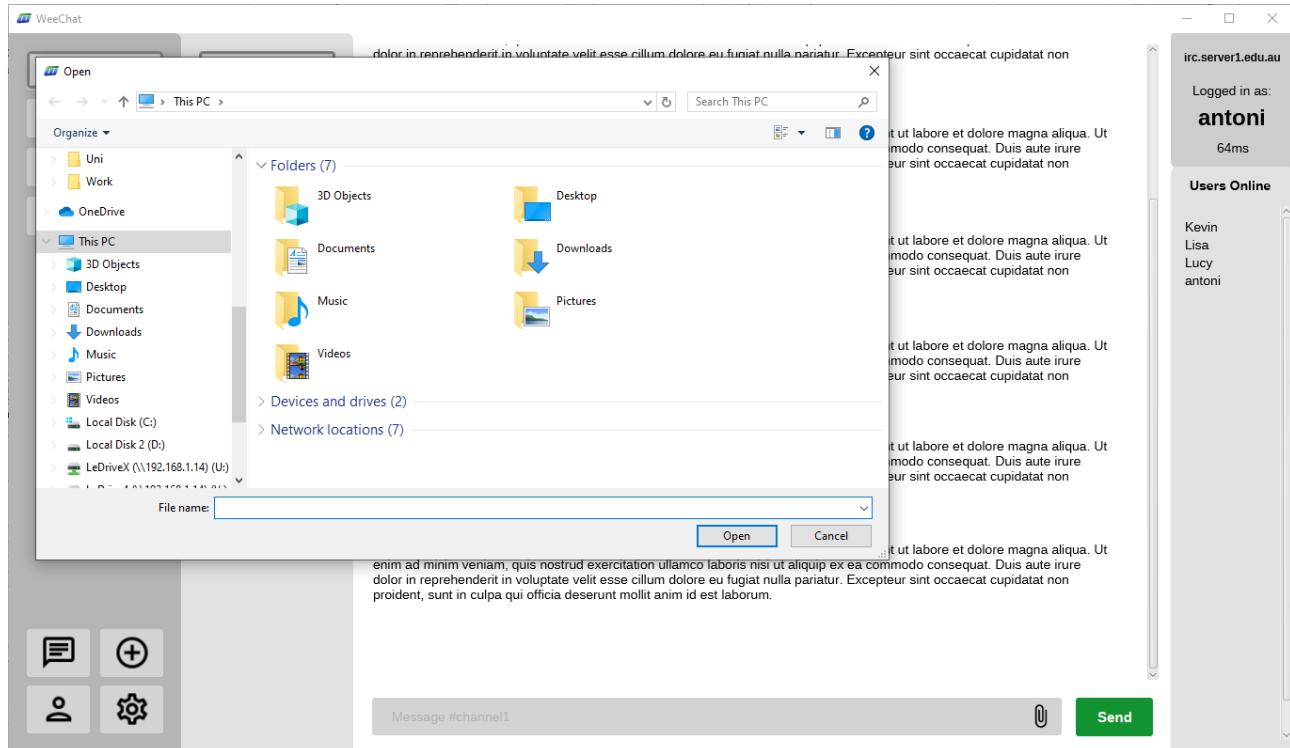
However when changing servers, the messages will not change as this is just to demonstrate how servers are switched.

anton1 8:42AM 24/03/20

hello

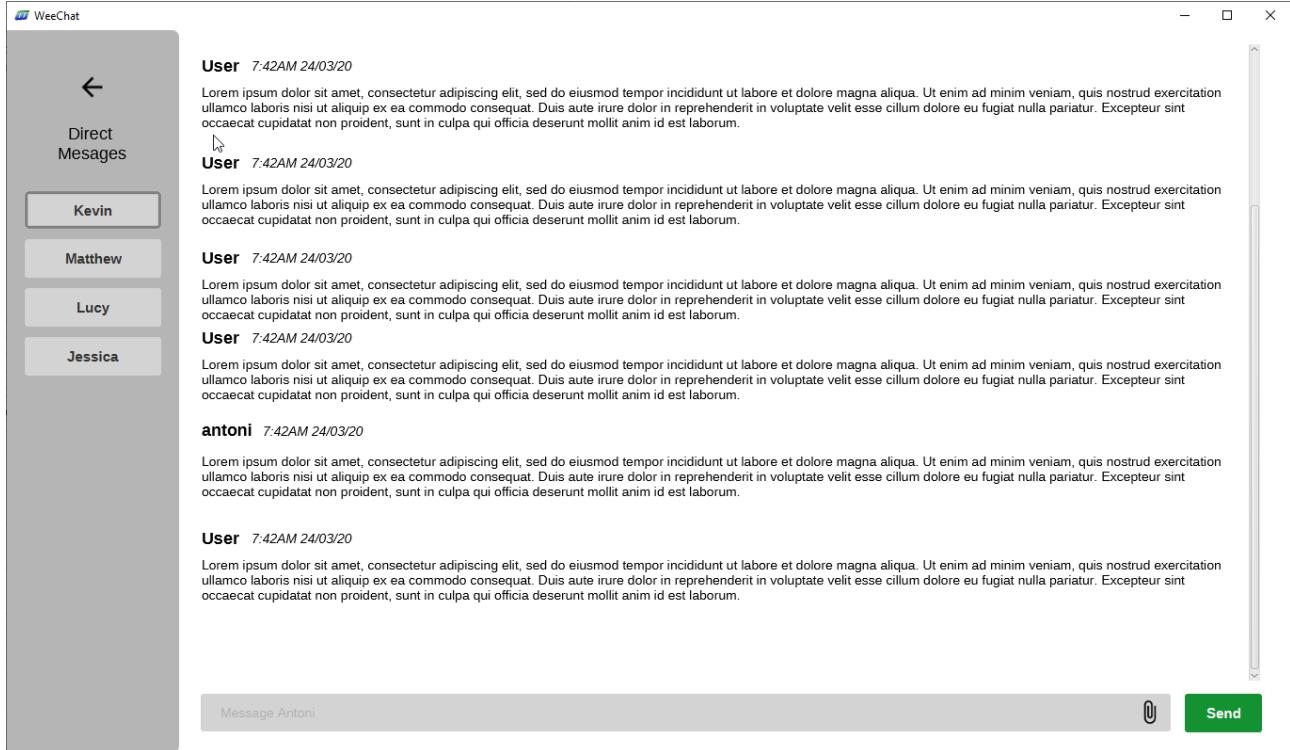
hello (paperclip icon) Send

In the messaging UI, a paperclip was chosen to symbolize the “attach” functionality, as it is used commonly in many programs and also associated with attaching something in the real world.



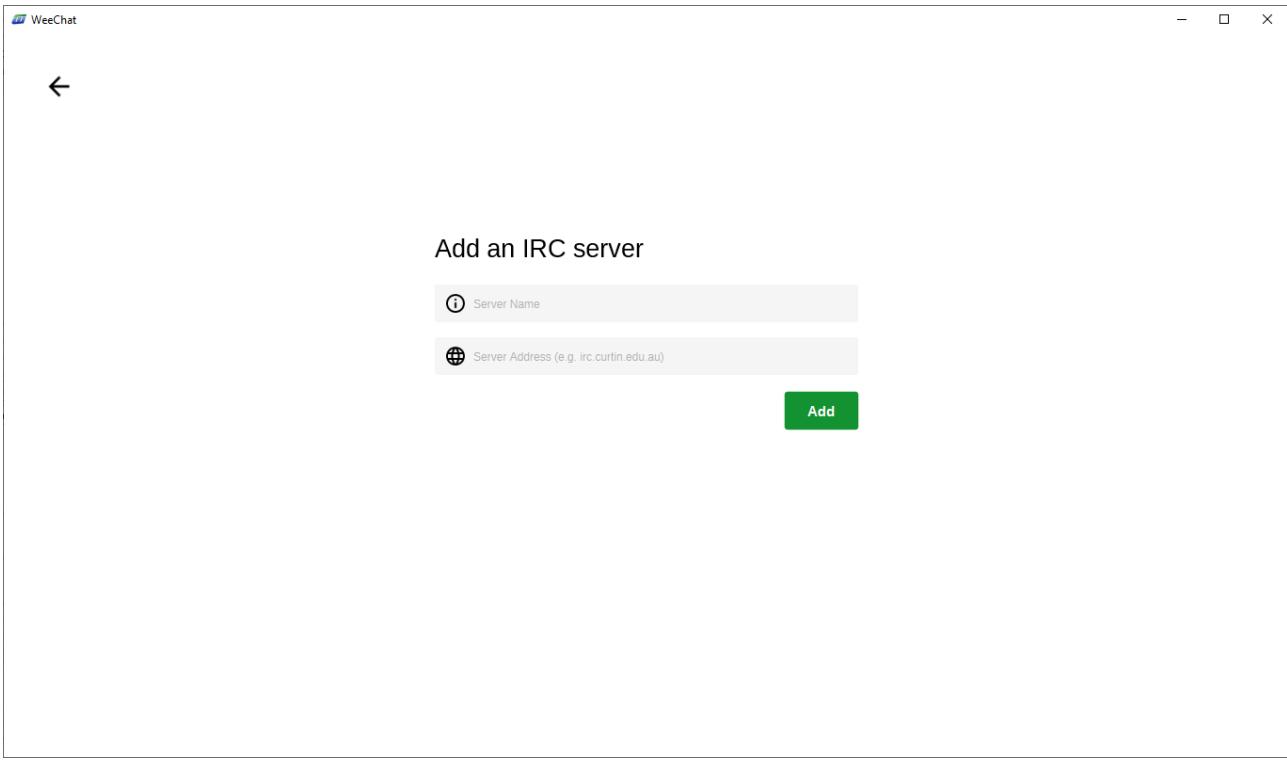
When clicking the paperclip to attach a file, the file explorer is opened, this is to represent the traditional steps taken to attach a file. However upon selecting a file nothing will happen, as this is just a demonstration.

When the user clicks on the “message” icon in the bottom left, they will be taken to the Direct Message page.



The direct message page follows the same design as the server messaging page, but instead of servers that contain channels, it is just other users.

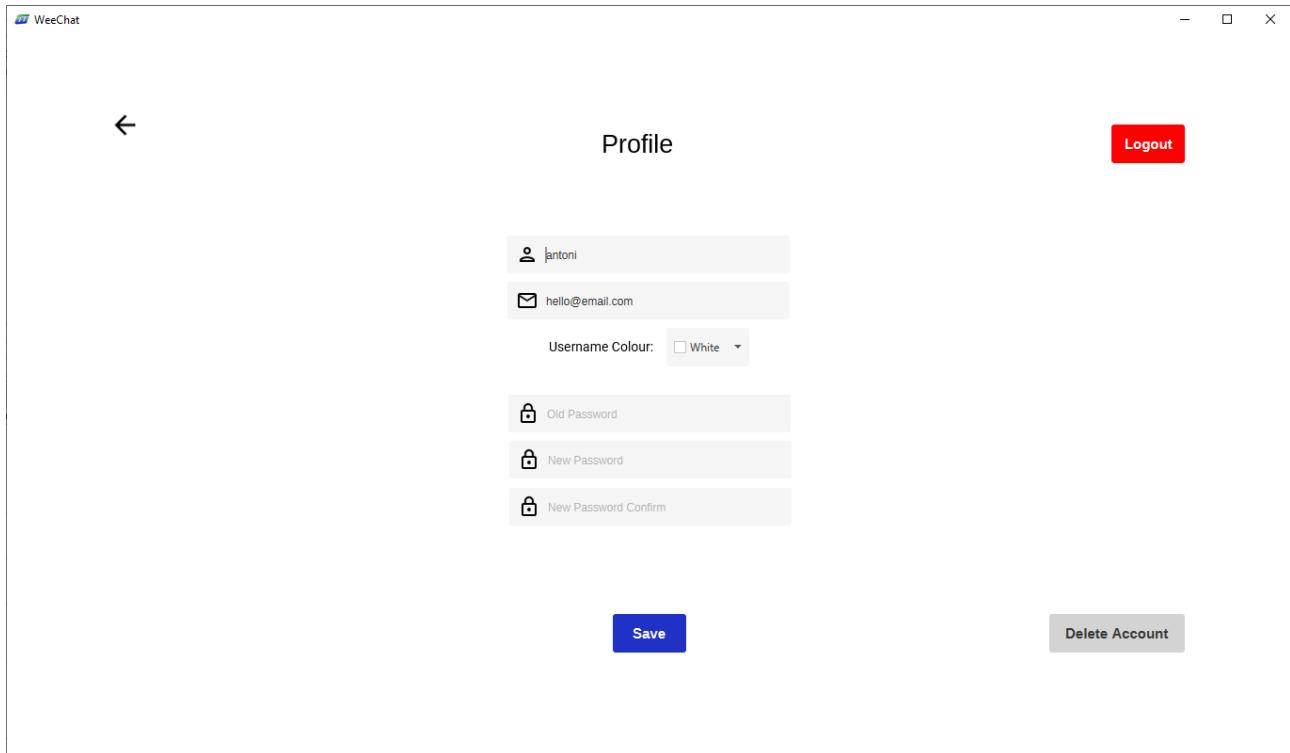
Back in the server UI, if the user clicks on the “+” button, they will be presented with the add server page.



This page was kept very simple to avoid clutter and any other complexities. The user should be able to determine what they need to do in order to add a server, which in this case is just give it a name and provide the URL. The green button uses the word “Add” instead of the right pointing arrow in the high fidelity prototypes in order to be more explicit and avoid confusion.

An oversight in the high fidelity prototypes was the field “Server Name”, as the buttons in the server UI use the name as the label. So it was added in the implementation.

Back in the server UI, if the person icon is clicked on, it will present the Profile page.



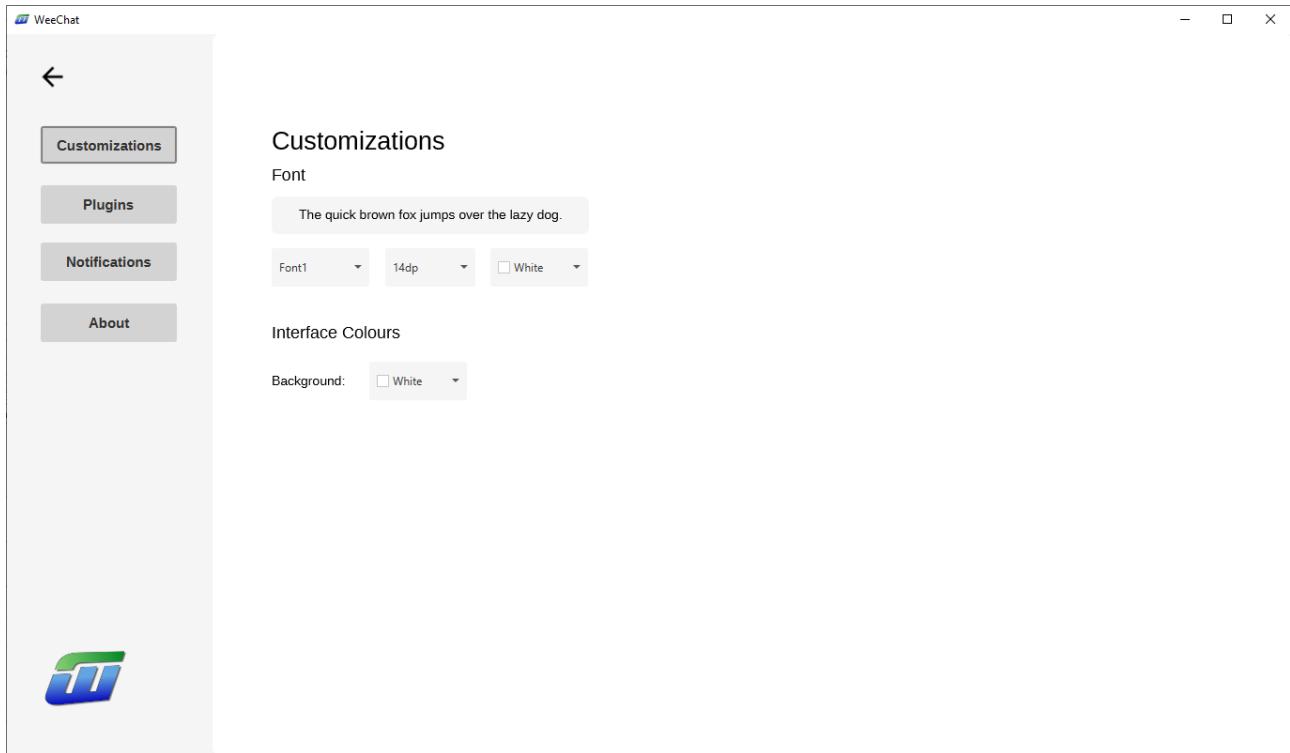
This page remains unchanged from the high fidelity apart from the “Profile” title being moved to the centre, as previously it was next to the Back button, which might lead to misunderstanding that clicking the back button would return to the profile. The “Logout” button is red to attract the users attention, if they were to click on it, it would be intentional. Again delete account is separated from other buttons to avoid misclicks. However in a more proper implementation, there would be a confirmation dialogue.

The save button is positioned under the form, where the user might usually find a submit button, so they know what actions it would perform.



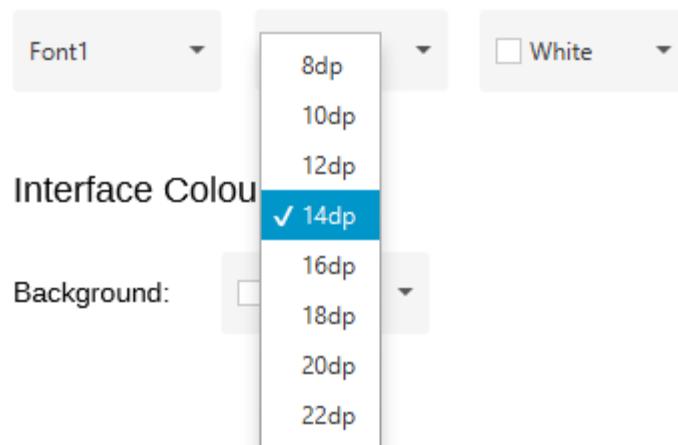
Upon clicking on “Save” the text will be changed to “Saved” in order to provide feedback and confirmation to the user.

Back in the server UI, if the gear (settings icon) is clicked, it would open the options page, presenting customizations.

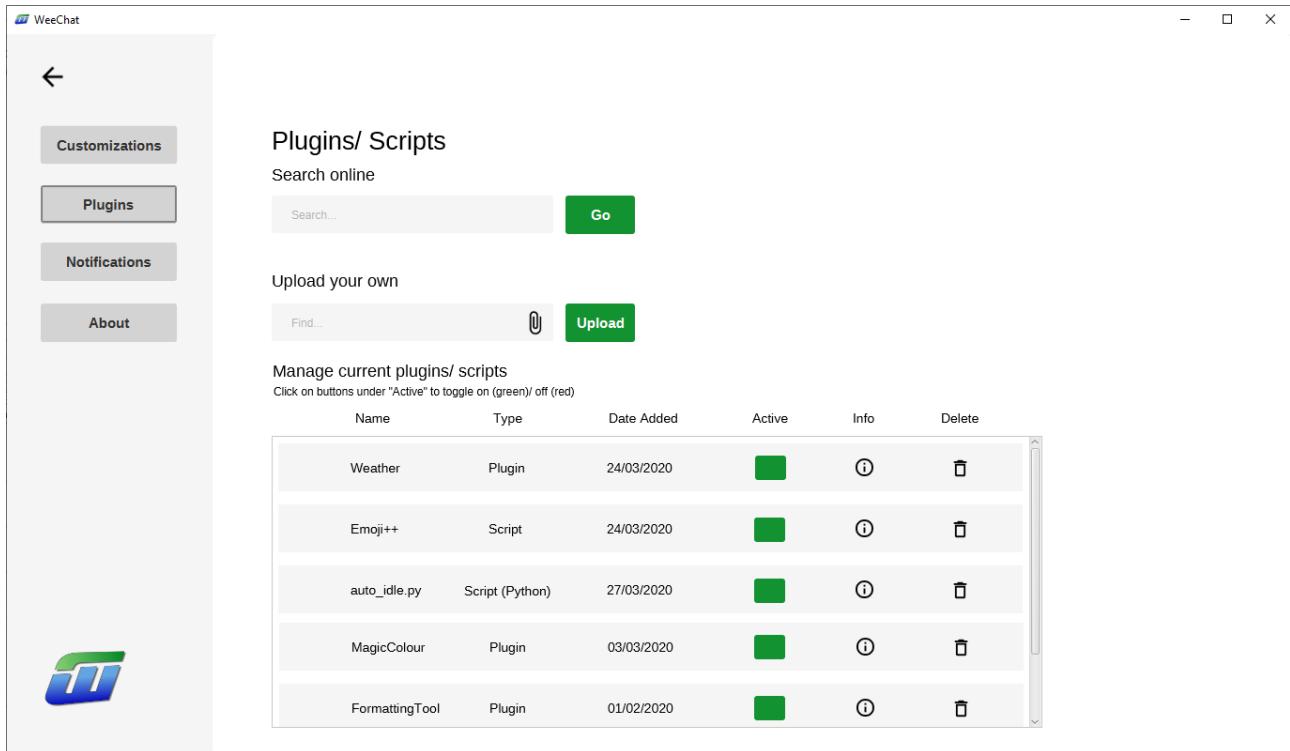


This also remained mostly the same from the high fidelity, but the font sizes have been changed to a drop down list in increments of 2pts instead of a decrementer and incrementer, which may take a long time to change font sizes, especially if they are changing from small to large sizes, and getting users to type font sizes in wasn't extremely user friendly.

Modifying these options don't actually change the actual sizes of the UI fonts as it is just for demonstration.



Upon clicking the "Plugins" button it will lead to the Plugins/ Scripts option page.



Remains mostly unchanged from the High Fidelity, however the attach button for “Upload your own” has been changed to a paper clip instead of “...”, to keep it consistent with what the program uses for attaching files.

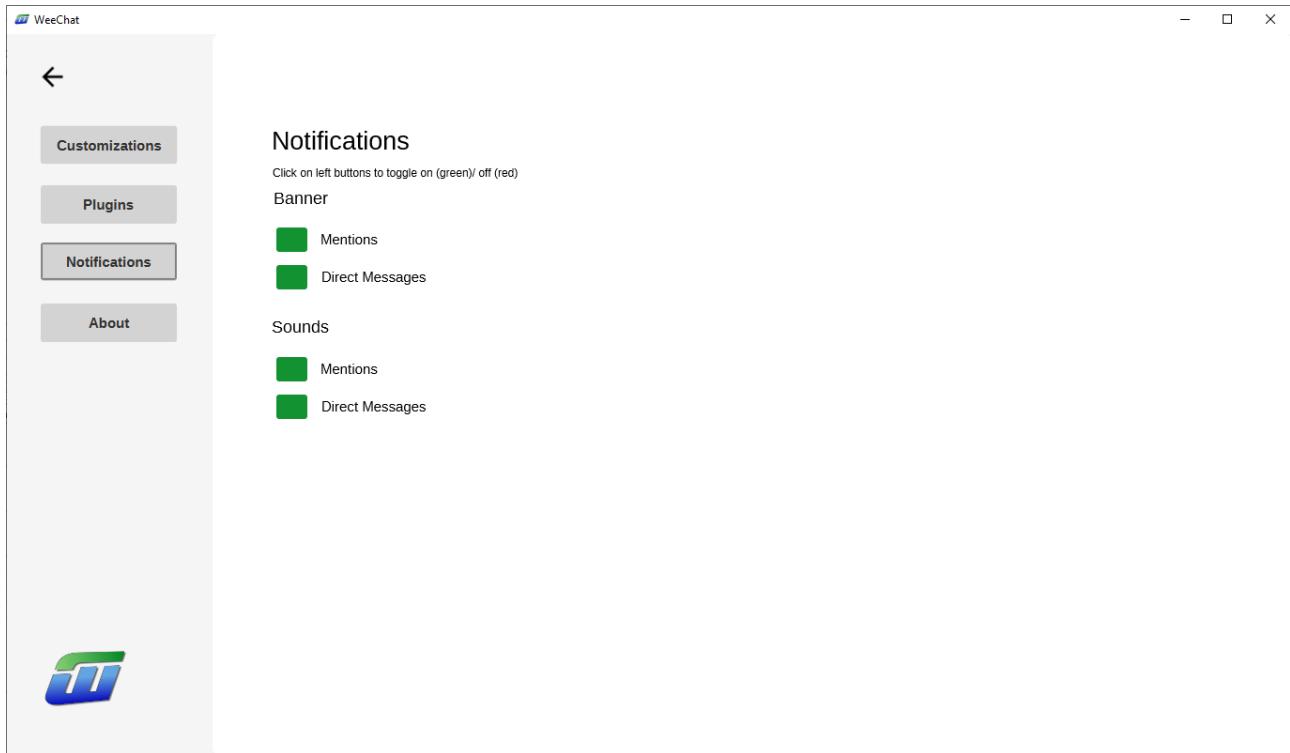
All items on this page are for demonstration only, the “Search Online” field should typically open the browser to the WeeChat plugin search page and uploading your own plugin makes it appear in the plugins list.

Due to limitations in JavaFX, there was no switch that existed as default, without needing to import a library as far as i know. This was therefore substituted for a ToggleSwitch, which changes between states on (active) and off (inactive). So this was a suitable alternative. In this implementation, the off state was represented by the colour red, and the on state with green.



Addition text prompts were also provided: “Click on left buttons to toggle on (green)/ off (red)”

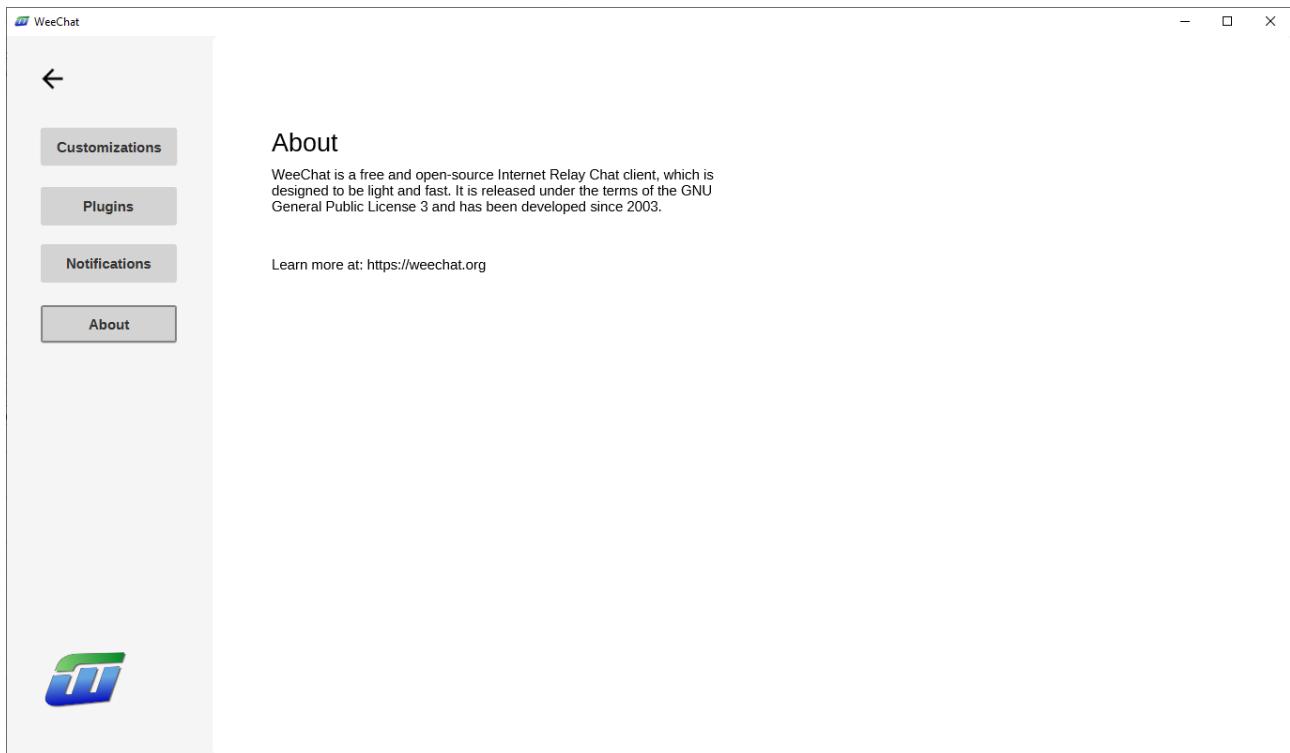
Upon clicking the “Notifications” button it will lead to the Notifications option page.



This page remains unchanged apart from the notification sound preview functionality being removed due to JavaFX limitations.

Toggle buttons were also used here in place of switches for the same reasons as above (in Plugins).

Upon clicking the "About" button it will lead to the about option page.



This simply provides information about the program from the Wikipedia entry. Unchanged from the prototypes.

User Involvement

The implementation was segmented into 4 milestones, and at the end of each milestone the current status of the project and progress so far is presented to potential end users of the product.

Each milestone consisted the program parts:

Milestone 1

- Login
- Register
- Add Server

Milestone 2

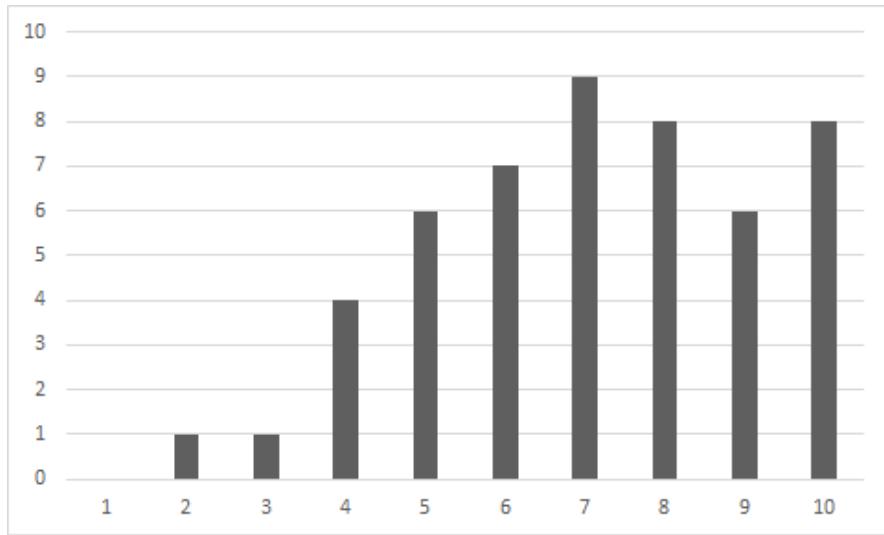
- Server
- Direct Messages
- Profile

Milestone 3

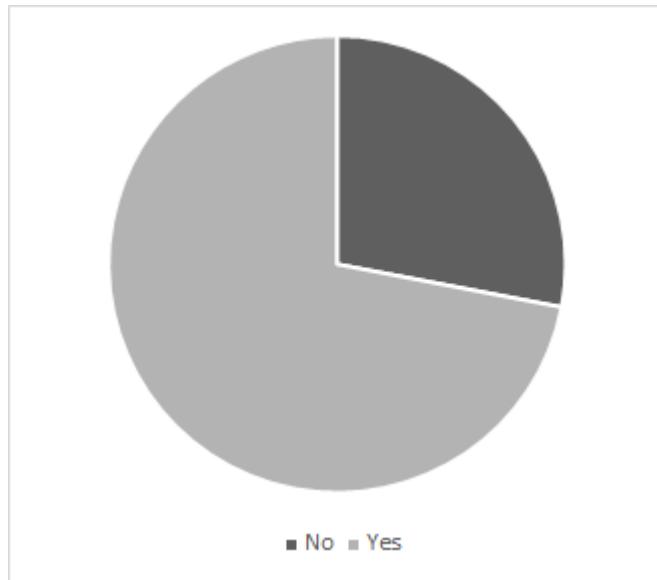
- Options
 - Customizations
 - Plugins
 - Notifications
 - About

Implementation finished

At each milestone, users were asked if they were satisfied with the product so far on a scale of 1-10 and compared to a competitor, which one they would choose. Those who were unsatisfied expressed some missing functionality, which was then added if it was reasonable and desired by multiple people.



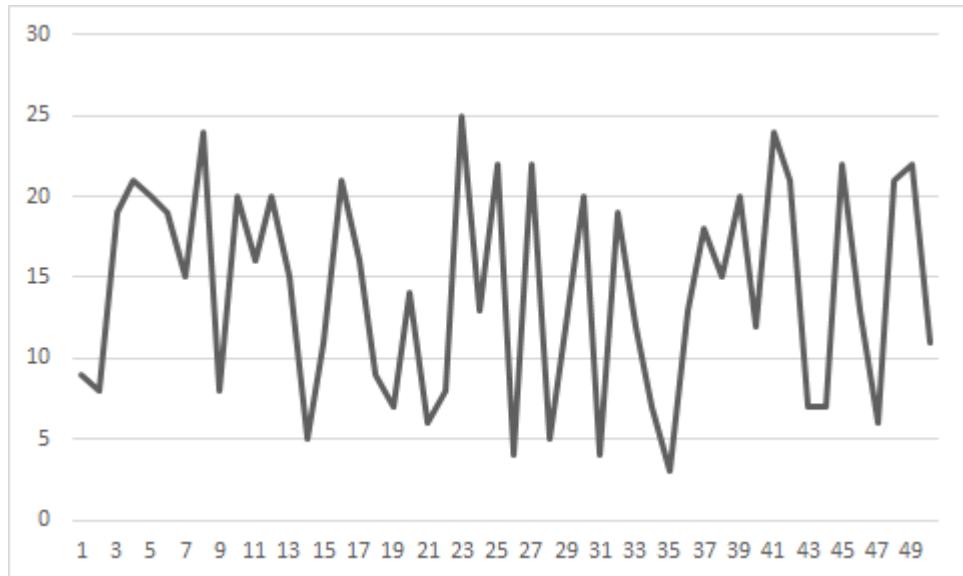
Competitor vs Current System:



72% of users said they would likely switch to the new system when compared with a competitors system.

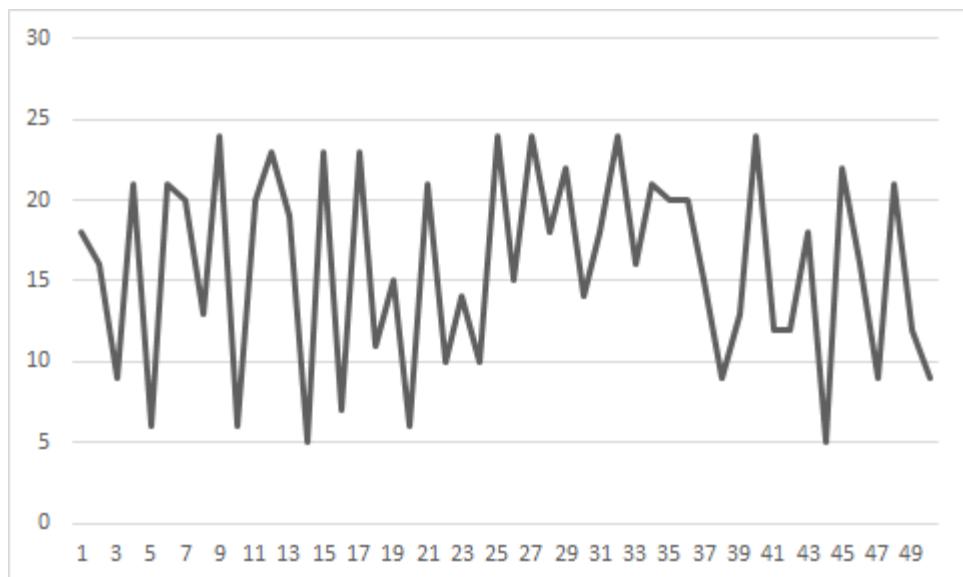
When the implementation development was completed, the final product was presented to a new group of 50 users. They were tasked with completing a certain goal in the program, and timed how long it took to complete the action. The tasks were:

Change username and password:



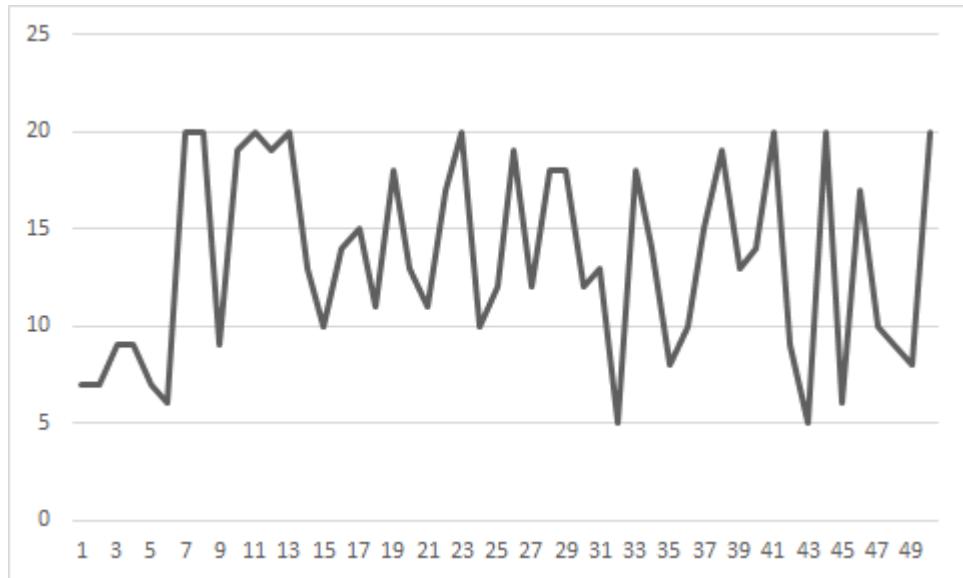
On average it took users new to the system 14.22 seconds to change their username and password. This involved clicking on their profile, editing fields, and clicking save.

Send direct message “Hello Antoni”:



On average it took users new to the system 15.88 seconds to send a direct message. This involved locating the direct message button, choosing the right user, typing the message and clicking send.

Add server “irc.curtin.edu.au” with name “Curtin University”:

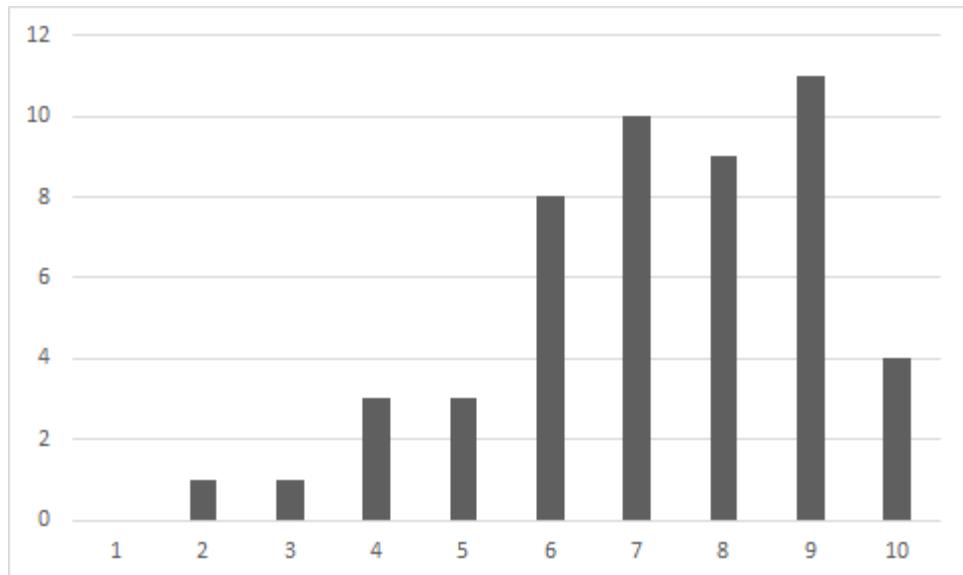


On average it took users new to the system 13.36 seconds to add a new IRC server. This involved locating the add server button, typing in the name and address, and clicking add.

When users took more than 20 seconds on average to perform a certain action, it had to be rethought out. (Not including the time it takes to login). In this case, there were no issues uncovered with navigation around the program.

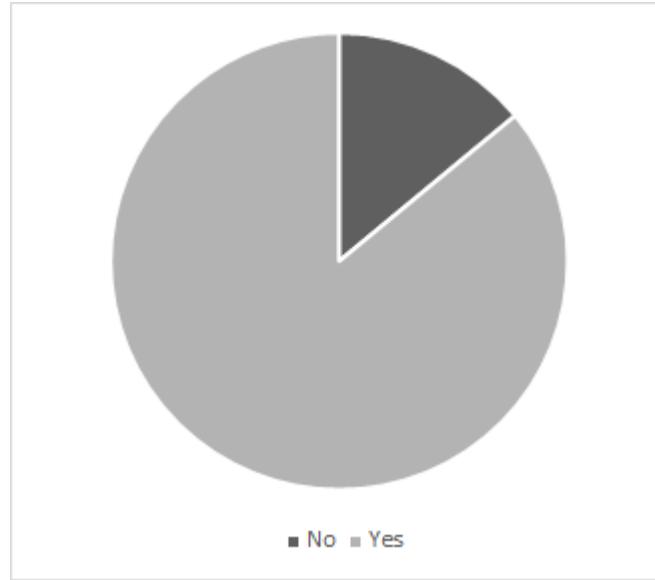
These users were also asked to rate the program on a scale of 1-10 in terms of how easy it was to use, and if they would continue to use it.

How easy was it to use?



On a scale of 1-10, users on average rated the program "Very Good" with an average score of 7.22. This indicated that a large majority of users found the program easy to use, but those that found it difficult expressed the need for more help documentation, which will be followed up on.

Continue to use?



86% of users expressed they will continue to use the program.

User quality of life features were added as requested such as tooltips, button feedback, etc.

Test Cases

Test Case	1
Name	Login
Expected Result	Opens chat upon valid login entered
Steps	<ol style="list-style-type: none"> 1. Fill in username and password fields 2. Click "Login"
Pass/Fail	Pass

Test Case	2
Name	Register
Expected Result	Opens chat upon valid fields entered to register
Steps	<ol style="list-style-type: none"> 1. Fill in username, email, password, colour fields 2. Terms & Conditions checkbox ticked 3. Click "Register"
Pass/Fail	Pass

Test Case	3
-----------	---

Name	Send Message to Server Channel
Expected Result	New message appears when sent
Steps	<ol style="list-style-type: none"> 1. Fill in message 2. Click send button 3. Message appears sent by the currently logged in user
Pass/Fail	Pass

Test Case	4
Name	Attach file
Expected Result	File explorer opened for user to attach file
Steps	<ol style="list-style-type: none"> 1. Click paperclip icon in message field 2. Select file

Test Case	5
Name	Send Direct Message
Expected Result	New message appears when sent
Steps	<ol style="list-style-type: none"> 1. Click “message icon” to open Direct Message page 2. Fill in message 3. Click “send” button 4. Message appears sent by the currently logged in user

Test Case	6
Name	Tooltips
Expected Result	Tooltip shown for every button without text description
Steps	<ol style="list-style-type: none"> 1. Hover over tooltip enabled button 2. Rectangle shown with about information

Test Case	7
Name	Change Channel
Expected Result	New channel selected highlighted
Steps	<ol style="list-style-type: none"> 1. Click a channel that isn't currently selected 2. Newly selected channel highlighted with grey border
Pass/Fail	Pass

Test Case	8
Name	Change Server
Expected Result	New server selected highlighted
Steps	<ol style="list-style-type: none"> 1. Click a server that isn't currently selected 2. Newly selected server highlighted with grey border
Pass/Fail	Pass

Test Case	9
Name	Edit Profile
Expected Result	User able to modify all aspects of their data
Steps	<ol style="list-style-type: none"> 1. Click "person" icon in chat page 2. Profile page is displayed 3. Edit any fields (Username, Name Colour, Email, Password) 4. Click "Save"
Pass/Fail	Pass

Test Case	11
Name	Add Server
Expected Result	User able to add a new server
Steps	<ol style="list-style-type: none"> 1. Click "Add" icon in chat page 2. Add server page is displayed 3. Fill in Server name and URL 4. Click "Add"
Pass/Fail	Pass

Test Case	12
Name	Customize program
Expected Result	User able to customize program font and colours
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Customizations" button 3. Modify Font Type, Colour and Size. 4. Modify Interface Colour
Pass/Fail	Pass

Test Case	13
Name	Search Plugins/ Scripts
Expected Result	User able to search plugins/ scripts online
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Plugins" button 3. Click "Go" with Search field having values <p>NOTE: Web search functionality not implemented</p>
Pass/Fail	Pass

Test Case	14
Name	Upload Plugins/ Scripts
Expected Result	User able to upload own plugins/ scripts
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Plugins" button 3. Click "Paperclip" icon to open file explorer 4. Click "Upload" with file path having values
Pass/Fail	Pass

Test Case	15
Name	Manage Plugins/ Scripts
Expected Result	User able to search, add, update and view plugins/ scripts

Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Plugins" button 3. Search online: click "Go" with Search field having values 4. Upload own
Pass/Fail	Pass

Test Case	16
Name	Manage Plugins/ Scripts
Expected Result	User able to toggle active plugins/ scripts, view date added, information and delete
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Plugins" button 3. Table under subtitle "Manage current plugins/ scripts" shown 4. Name, Type, Date added, Active, Info, and Delete columns. 5. Toggle switch 6. Click "i" icon for more information 7. Click "Bin" icon to delete
Pass/Fail	Pass

Test Case	17
Name	Control Notifications
Expected Result	User able to toggle notification banners and sounds
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "Notifications" button 3. Use toggle switches to change between "Green" on and "Red" off for banners and sounds.
Pass/Fail	Pass

Test Case	18
Name	View About
Expected Result	User able to view about information of program
Steps	<ol style="list-style-type: none"> 1. Click "Settings" icon in chat page 2. Click "About" button 3. About page with information shown

Pass/Fail	Pass
-----------	------

Test Case	19
Name	Logout
Expected Result	User able to logout of current session and returned to login page
Steps	<ol style="list-style-type: none"> 1. Click “Person” icon in chat page 2. Click “Logout” button
Pass/Fail	Pass

Test Case	20
Name	Delete Account
Expected Result	User able to delete account and be returned to login page
Steps	<ol style="list-style-type: none"> 1. Click “Person” icon in chat page 2. Click “Delete Account” button
Pass/Fail	Pass

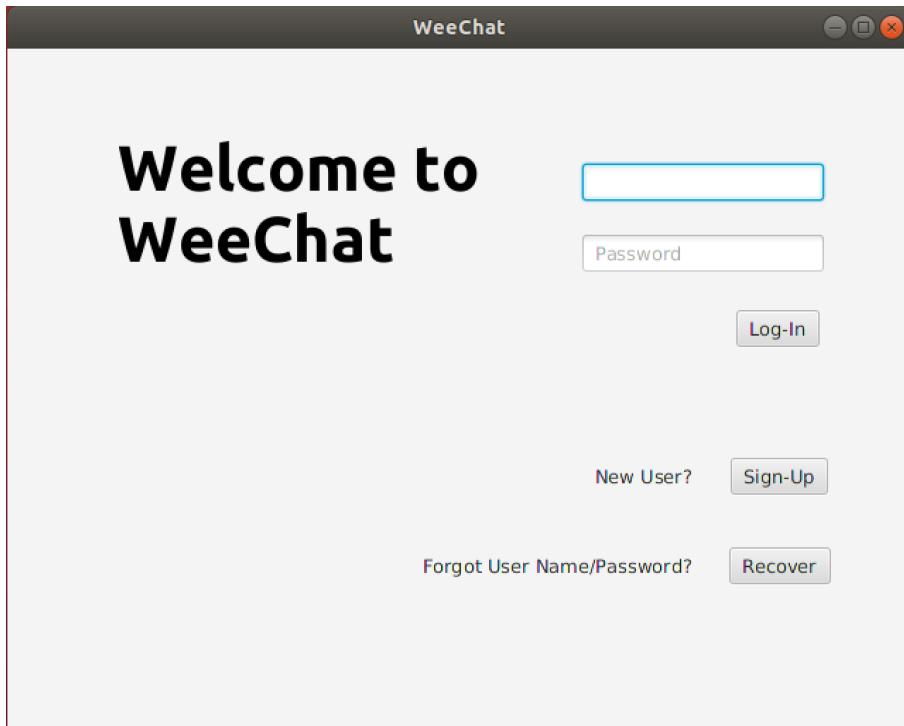
Terence Marcelo

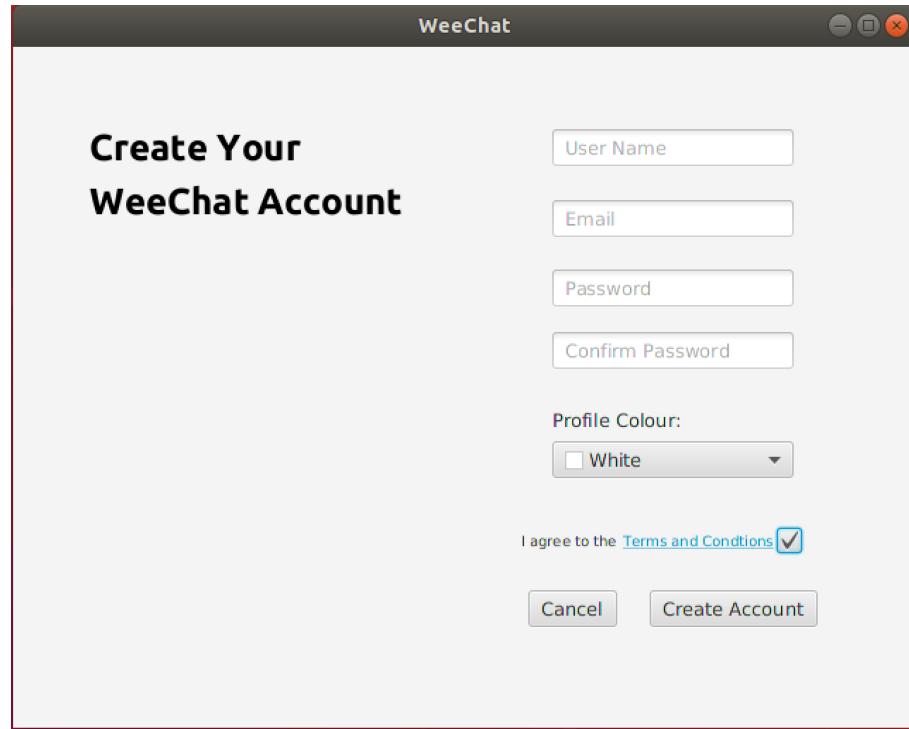
Implementation Discussion

After a meeting with the client, I received feedback regarding areas in the design that need improvement and additional features I could implement. Some suggested features included tooltips to guide users what certain buttons in the interface are responsible for, a help section, and a dark user interface mode for users who may want an interface that's easier on the eyes. These were then added after my high-fidelity prototype. Due to time constraints some suggestions were not implemented. These include making the side bars collapsible and to have more UI feedback. The option for dark mode is present in the settings menu it's actual implementation is not shown. It is also not connected to an actual SQL database and a sample message in a message thread is hard-coded.

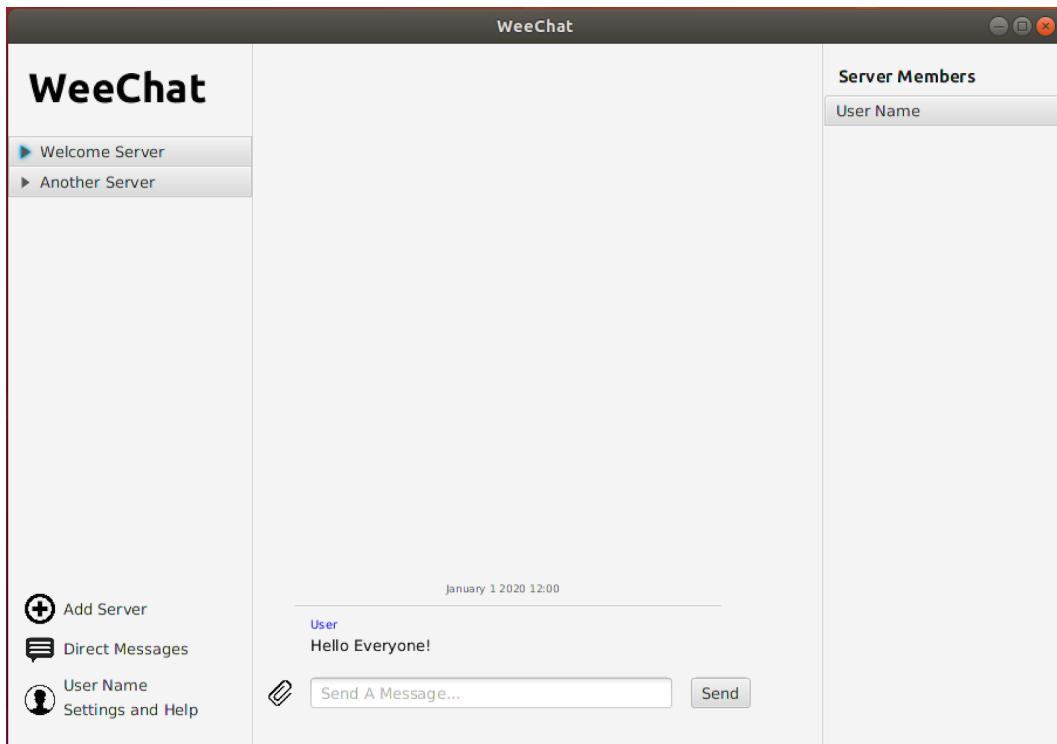
The final design does give a general idea of how the application would appear if it were functional. I used Netbeans Scene Builder for this implementation. All UI elements such as text boxes, buttons, font, etc. used were just the default elements available in the scenebuilder.

When the application is first started, the user is presented with a clean and simple Sign-In/Sign-Up window. For this assignment simply pressing Sign-In will take you to the main interface, sign-up would open a window with fields to enter details, and the recover option just displays a message saying a recovery email has been sent.

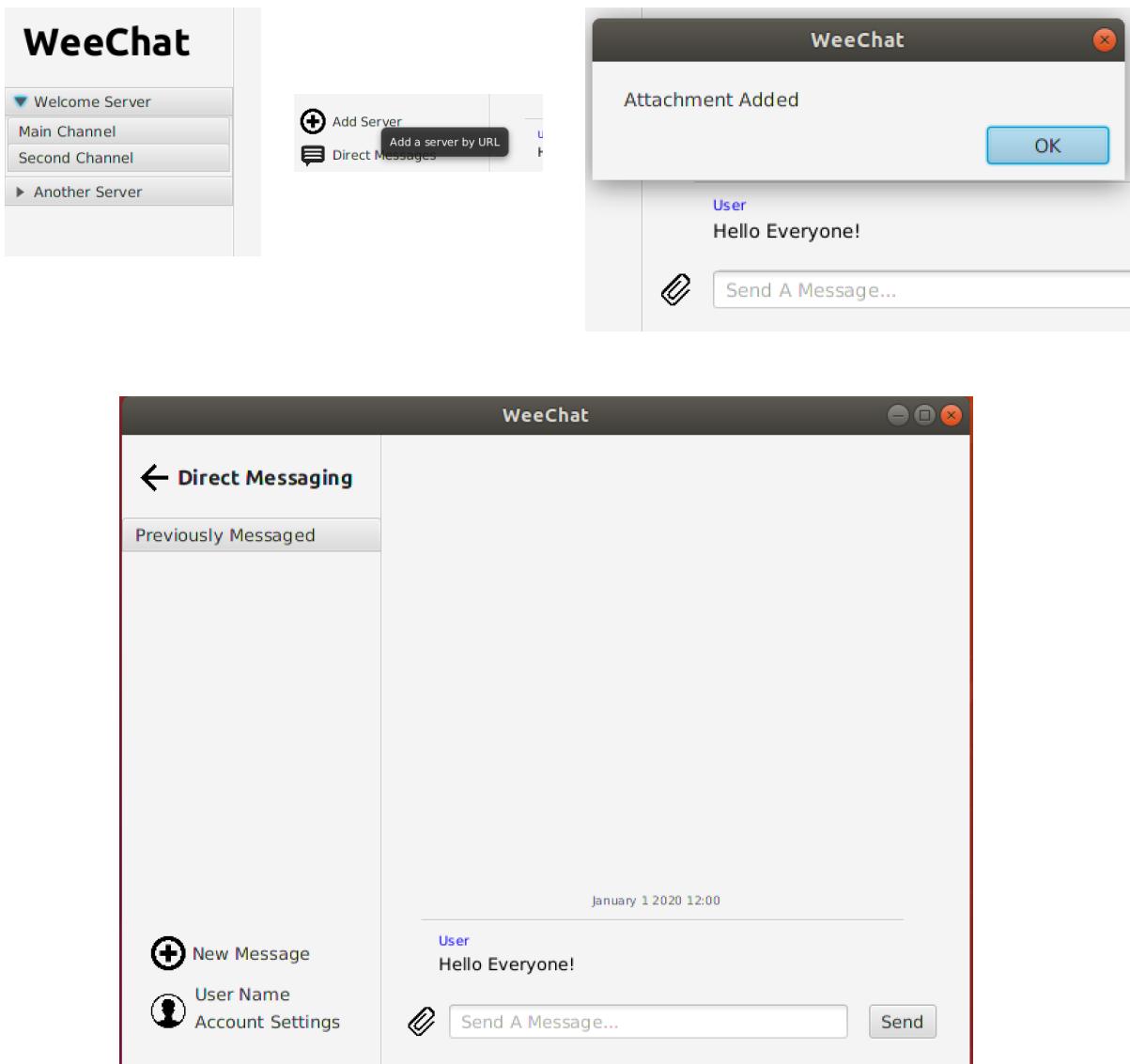




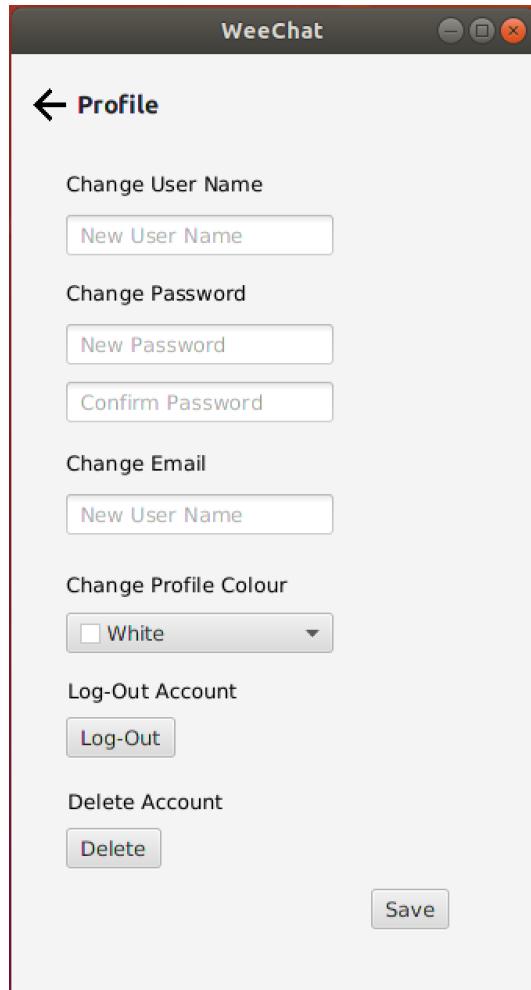
Selecting both Cancel or Create Account will take you back to the sign-in window but a message saying "Account Created" shows when selecting Create Account.



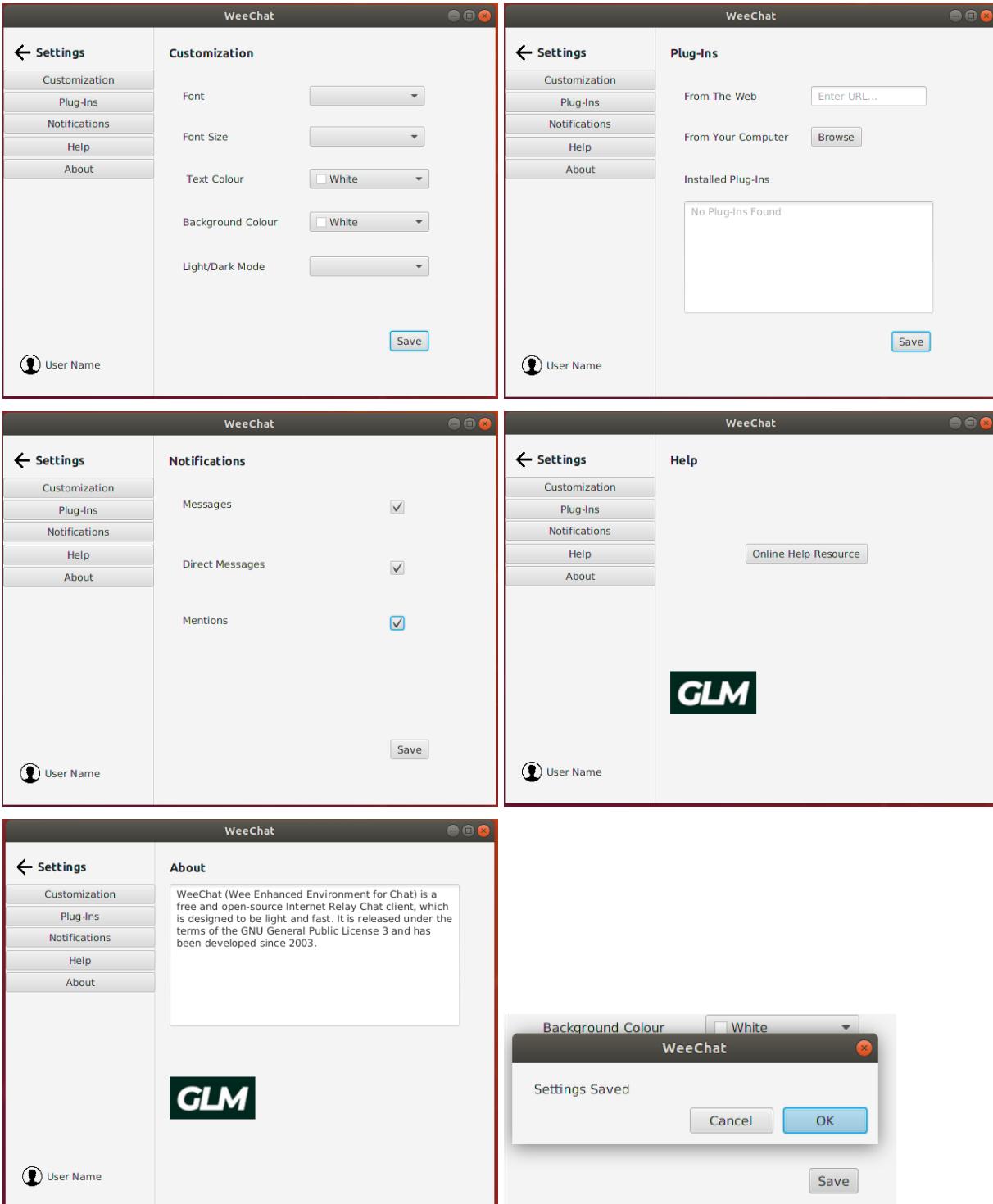
This is the main window of the program. On the left there is a sidebar that contains the server which can be expanded to show the channels. Hovering over any element shows a tooltip to guide the user. The paperclip icon is for adding an attachment to send but for this assignment it just displays a message saying attachment added. Add Server is similar with a message saying “Server Added”. Selecting Direct Messages, Username/Icon, and Settings opens their respective windows. Selecting on one of the server members opens the direct message window to send them a direct message.



The back button returns to the main interface. For this assignment, selecting “New Message” just displays a message saying you don’t have any contacts yet. The other elements that also exist in the main window function the same way.



This window is shown when selecting the Username/User icon. Log-out just returns you to the Sign-in/Sign-up window. Selecting delete account displays a message asking for confirmation and for this assignment “cancel” is the only option.

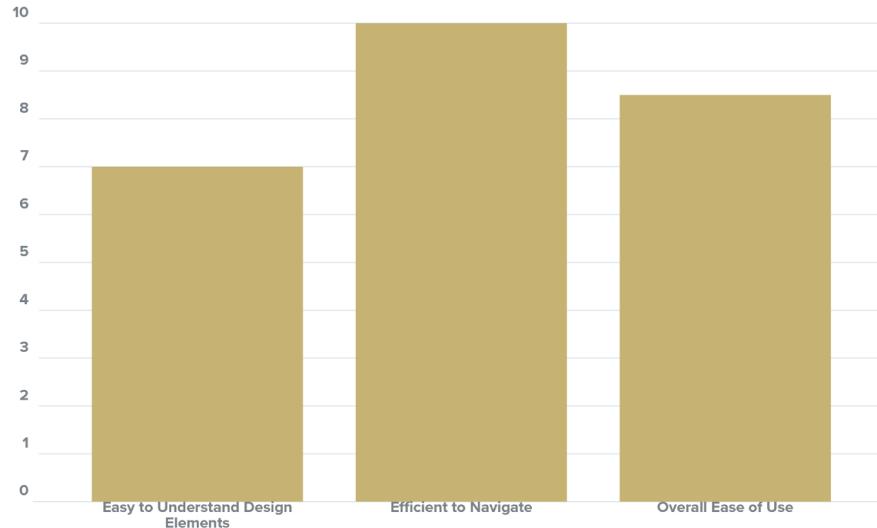


These are the various settings windows. For this assignment it is just for display and doesn't actually change anything. Selecting "Save" just displays a message saying that the settings have been saved. The Online Help Resource would link to the online help website but isn't implemented.

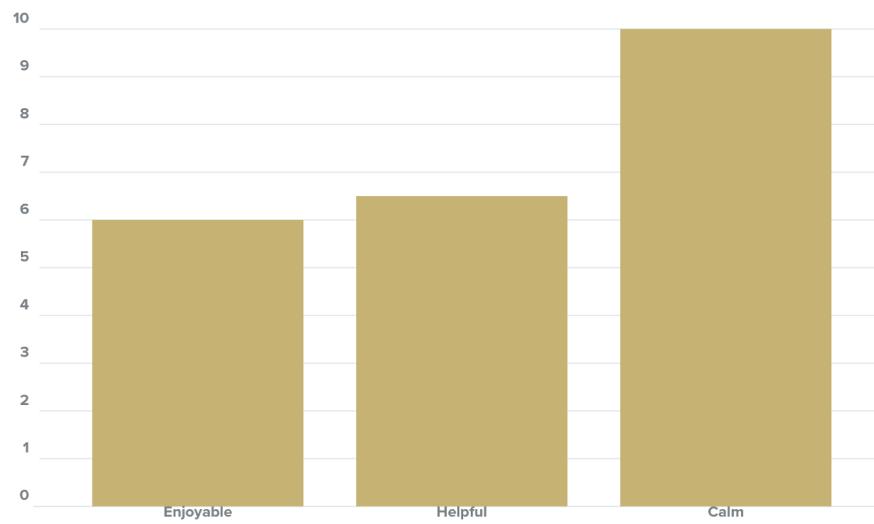
User Involvement

At each significant stage in the design process, I presented my design to potential users for feedback. Information regarding ease of use, user experience, and which design the prefer was then gathered. Users overwhelmingly prefer the simple GUI compared to the cluttered command-line interface of the previous design.

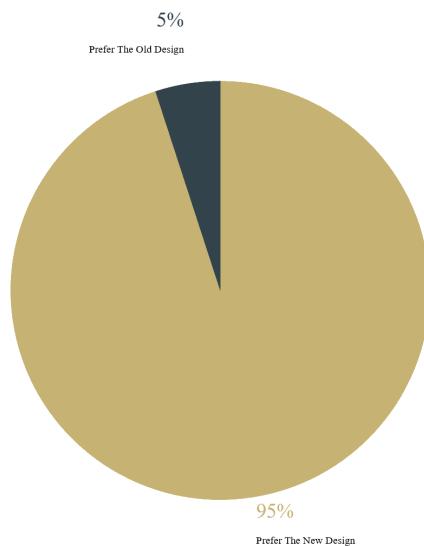
Average Rating from Users



User Experience



User Design Preferences



Test Cases

Test Case Number: 01

Test Case Details: Navigate the Log-In Screen to get to the main user interface.

Steps Required:

1. Enter User Name and Password.
2. Select Log-in.

Expected Result: Main window will be displayed.

Outcome: Pass

Test Case Number: 02

Test Case Details: Navigate the Log-In Screen to get to the Sign-Up interface and create an account.

Steps Required:

1. Select Sign-Up.
2. Complete every text field.
3. Select Profile Colour from Colour Picker.
4. Select Terms and Conditions checkbox.

Expected Result: Message displaying confirming account creation. Log-in window will be displayed.

Outcome: Pass

Test Case Number: 03

Test Case Details: Navigate the main window to send a direct message.

Steps Required:

1. Select Direct Messages.
2. Select a previously message thread or select "new message".
3. Enter message in text field below.
4. Select send.

Expected Result: Message text is displayed in the message thread.

Outcome: Not Implemented

Test Case Number: 04

Test Case Details: Navigate the main window to send a message to a server channel.

Steps Required:

1. Select Server to expand the list of channels.
2. Select a channel.
3. Enter message in text field below.
4. Select send.

Expected Result: Message text is displayed in the message thread.

Outcome: Not Implemented

Test Case Number: 05

Test Case Details: Navigate to the profile window to make changes.

Steps Required:

1. Select either the user name or the user icon.
2. Enter text into the corresponding text field that you want to make changes to.
3. Select save.

Expected Result: Message saying settings have been saved is displayed.

Outcome: Pass

Test Case Number: 06

Test Case Details: Navigate to the settings menu to make changes to the application settings.

Steps Required:

1. Select settings and help.
2. Select relevant area that you'd like to change (Customization, Notifications or Plug-Ins).
3. Make changes by selecting either the checkboxes, colour picker, or dropdown selector.
4. Select Save.

Expected Result: A message saying confirming that the settings have been changed should be displayed.

Outcome: Pass

Test Case Number: 07

Test Case Details: Get Help with the application.

Steps Required:

1. Select settings and help.
2. Select help in the sidebar.
3. Select the online help resource.

Expected Result: You're browser open to the webpage containing help.

Outcome: Not implemented.

Test Case Number: 08

Test Case Details: Adding a new server.

Steps Required:

1. Select new server from the main window.
2. Enter the URL of the new server in the text field that is displayed.
3. Select save..

Expected Result: The new server will be added to the sidebar.

Outcome: Not implemented.

Test Case Number: 09

Test Case Details: Log-out of your account.

Steps Required:

1. Select either the user name or the user icon.
2. Select Log-out.

Expected Result: You will be returned to the sign-in screen.

Outcome: Pass

Test Case Number: 10

Test Case Details: Delete your user account.

Steps Required:

1. Select either the user name or the user icon.
2. Select Delete.
3. Select yes when asked to confirm deletion.

Expected Result: You will be returned to the sign-in screen.

Outcome: Not Implemented

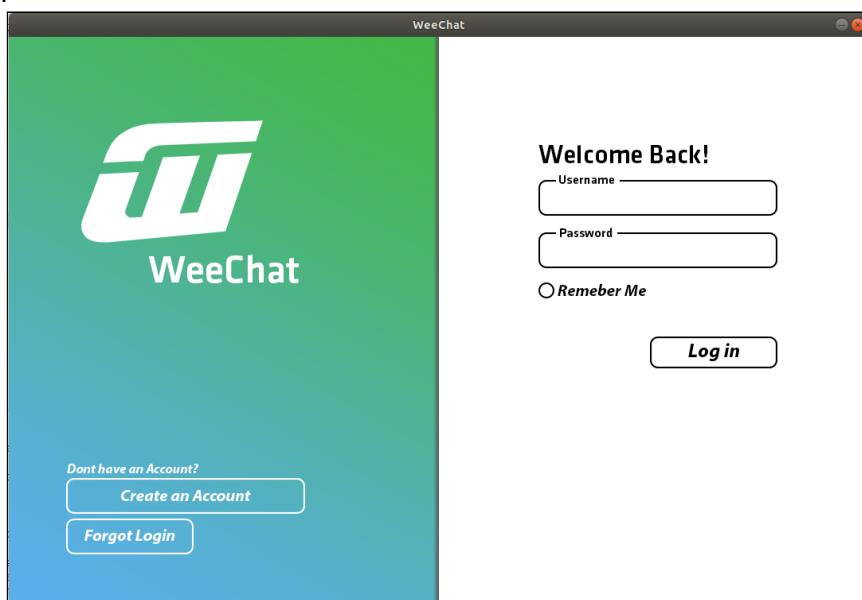
Ryan Martin

Implementation Discussion

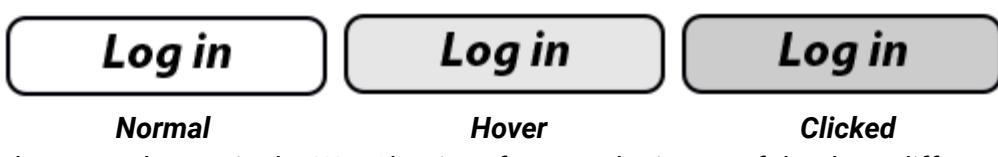
Upon completing the High Fidelity drawings for a new WeeChat interface the client has approved of the overall design, however they have asked for a few changes that they would like to see in the finished version of the application. Follow the feedback i gathered from the client i was able to implement many of the functions that the client wanted to see in the new interface which included:

- Adding an exit and minimize button.
- Added foldable sidebars for the active users.
- Added Icons and Fixed the “Add Server” and “Settings” button.
- Added a more descriptive UI feedback.
- Added a “Help” button and tool tips

With the feedback given to me by the client I was able to adjust the design of the interface to better fit the clients expectations.



Upon starting WeeChat the user is faced with the “login” interface. I have improved the UI feedback, upon the client's request, giving the user better responsiveness when using the UI.



As shown above, any button in the WeeChat interface can be in any of the three different states.

- Normal: When the user is not using the button, this is the button's default state.
- Hover: When the user has the mouse hovering over the button.
- Clicked: When the user has interacted with the button and expects a result.

This clearly and simply lets the user know if the user input is being acknowledged by the program.

Welcome Back!

Username _____

Password _____

Remember Me

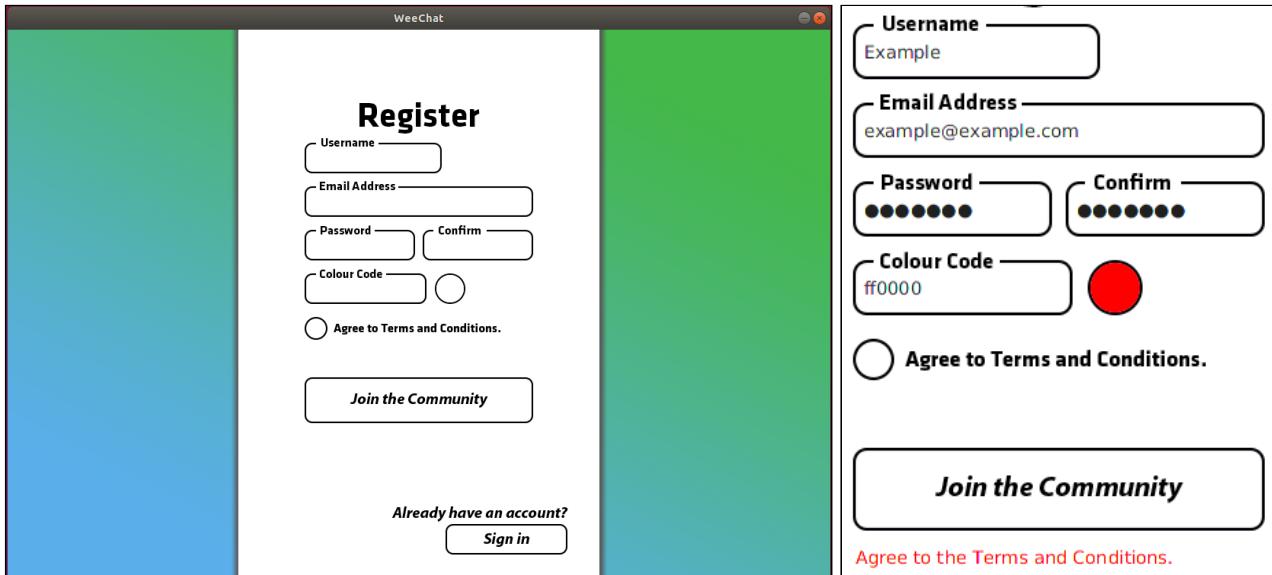
Username or Password is Incorrect.

Log in

I have also added an login error display that informs the user if the ‘username’ or ‘password’ that has been entered is incorrect. If the input is incorrect a red message will be displayed above the ‘login’ button informing the user of the incorrect user input.

I have also added the title bar at the top of WeeChat where the exit and minimize buttons are located. Other than the changes listed above the login interface is the same as the High Fidelity drawings.

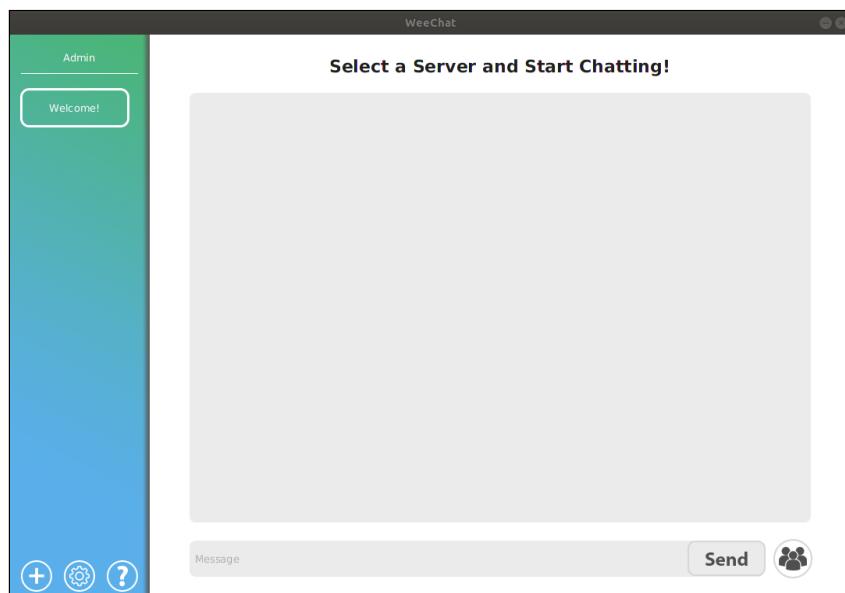
Upon the user selecting the 'Create an Account', they are taken to the registration interface.

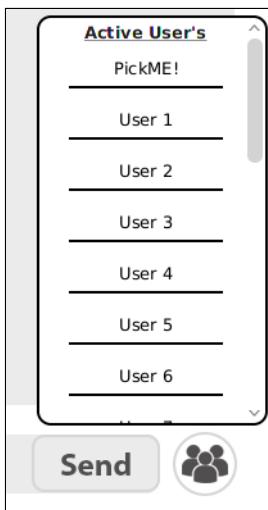


The "registration" interface is very similar to the one from the High Fidelity drawings however, all the buttons in this interface have the three state responsive UI feedback, similar to the buttons from the "login" interface.

The "registration" interface now has a registration error display message to inform the user if any details entered are either incorrect or does not match the input requirements. The error display message is red to easily grab the users attention and is also located under the 'Join the Community' button. All input fields have to be filled in for the 'join the community' button to work, if not all of them are filled an error message will be displayed similar to the above registration example image. The colour code input works by accepting a six-digit colour code and once the 6th digit is entered the colour display next to the input will automatically change to the desired colour. If the code is not valid or more/less digits are entered nothing will happen and an error will be displayed if the user tries to make an account. If the user already has an account, they can simply click the 'sign in' button to be taken back to the 'login' interface.

Once the user has logged in or created an account they will be immediately taken to the 'Server UI' interface which acts as the main interface for this redesign. The title at the top of this interface changes depending on which server and channel the user is connected too.





The client believed that having a dedicated sidebar for active users would take up too much space, the client's request the active user's sidebar should be foldable to maximise the space. This was done by adding an 'active user' button which when clicked will display a foldable menu list, above the button, that shows all the active users currently connected to that server and channel. Using the foldable display rather than the fixed sidebar has given the 'server UI' improved messaging space.

The 'server UI' interface now has three distinct usable buttons at the bottom left side opposed to the single 'settings' button the prototype contained. The first button is the new and improved 'Add Server' button, which per the client's request, the 'add server' caption was removed and then the button was placed into a more convenient and



constraint position rather than changing position every time a new server was added to the list. The second button is the modified settings button, the caption 'Settings' was removed and an easily identifiable icon was used in its place. The last button is a 'help' button which will take the user to the WeeChat support page when clicked. Along with the new easily identifiable icons for the buttons, tooltips have also been added so that if the user doesn't know what a certain button does the tooltips will help.

Each message contains the user's display name, the current time and date, followed by the message that the user wishes to send. As the user changes the font family, font size or username colour code the chat box will update with the nearly new data. The default colour code for the account

Admin (22:19 - 16/MAY/2020): This is a test
Admin (22:20 - 16/MAY/2020): this is what a message looks like

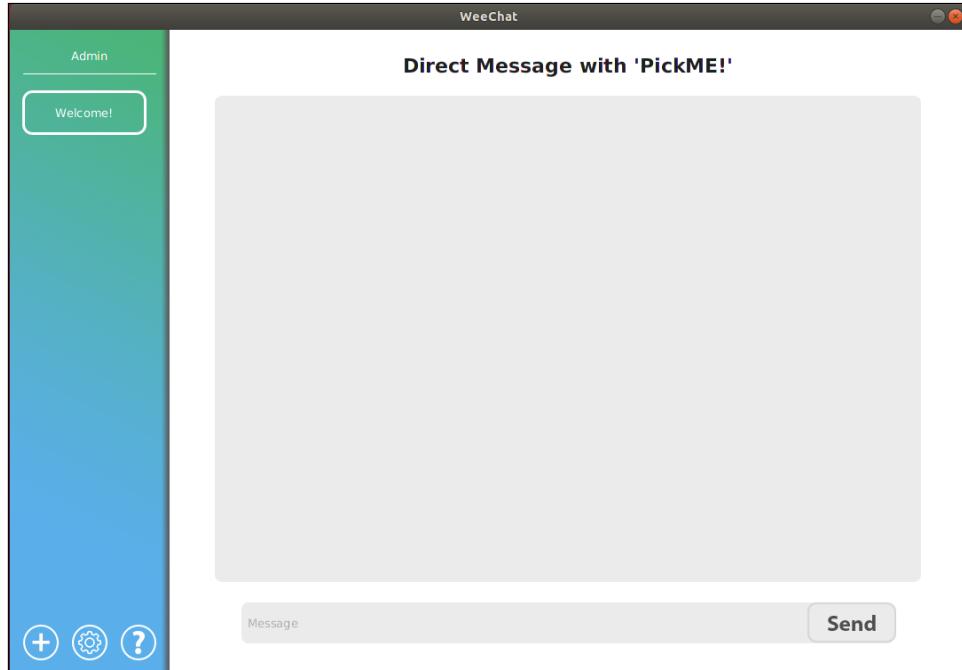


'Admin' is red, which is shown by the above image, the default font family is 'System' and the font size is '13'.

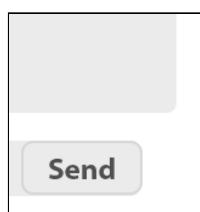
Similar to the prototype when the server is clicked, the foldable channels menu is displayed which shows all channels connected to that specific server. The channels menu will disappear when the mouse is no longer in the confines of the channels menu. The user is able to scroll through the channels list easily until the desired server is found. The servers are also in a list which can be scrolled through if needed and when the desired server is found can be clicked to get

the list of all channels connected to it. The default 'welcome' server is already added to welcome new users to WeeChat and act as a help page if users need extra advice.

Once the user selects a user from the active user's list a direct message will begin between the two users. The title will change informing the user its a direct message and with whom they are communicating with.

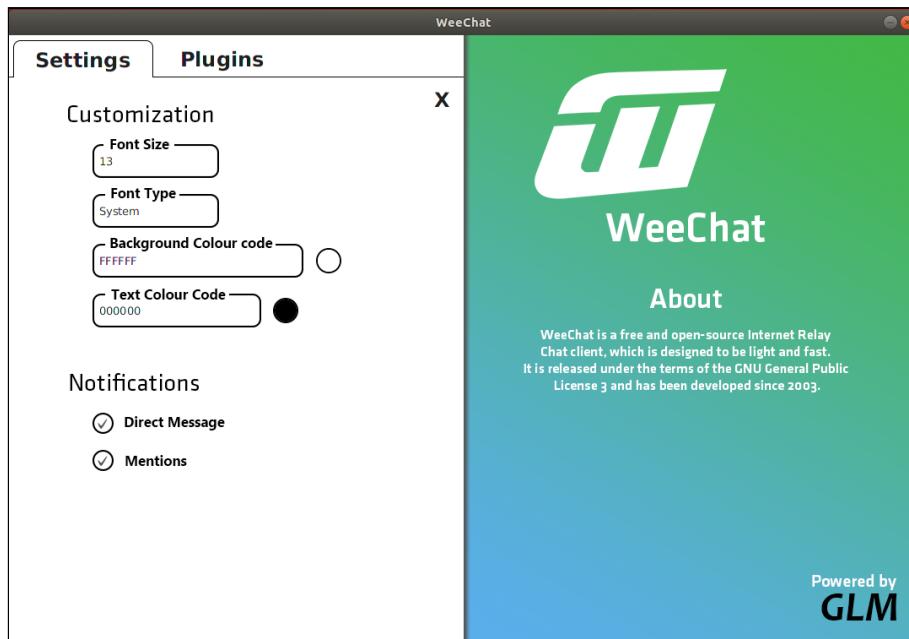


The 'Server UI' and 'Direct Message' interfaces are very similar in appearance and functionality.



The main difference between the two interfaces is that the 'active users' button has been removed, as shown in the image on the left, and the message bar at the bottom has been centered to the chat window. To exit the 'Direct Message' interface all the user has to do is select a server and channel from the server list and join it. After the server is joined then the direct message is automatically suspended.

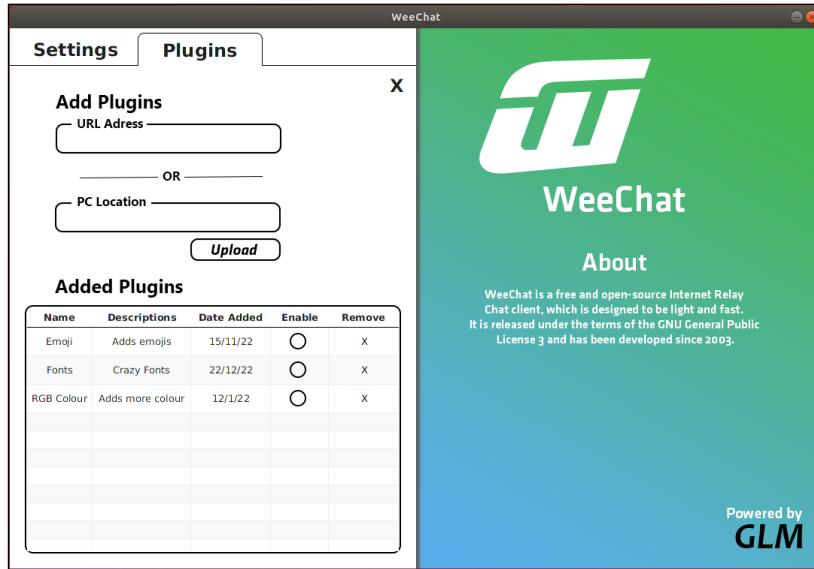
When the user selects the 'Settings' button they redirected to the settings interface. To leave the settings interface the 'x' button, which also uses the 3 state responsiveness, will return the user to the previous interface.



The 'settings' interface is similar to the prototype, however the UI of this interface has been improved to simplify the usability. Similar to the 'registration' interface, the background and text colour code need a 6-digit input and if valid the circle next to the corresponding code input will automatically change to the desired colour. The default background colour is white and the default

text colour is black, for easy readability. The font size and type can also be easily changed by entering in the new size or font family and it will be automatically saved and all changeable text will now display the new desired font type and size.

When the user decides to add plugins to WeeChat and clicks the 'Plugins' tab they then will be faced with the following interface. Here user's can add and remove plugins easily. To leave the settings interface the 'x' button, which also uses the 3 state responsiveness, will return the user to the previous interface.

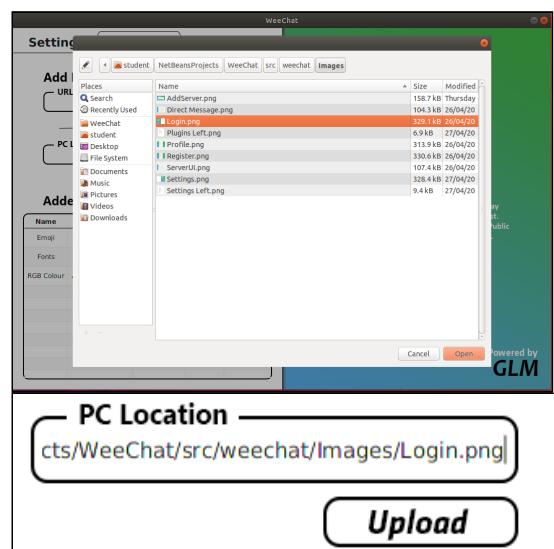


When the user wants to add a new plugin to WeeChat they have two ways of doing so, first is by uploading one from a URL address and the other is by uploading one from the users computer. If the user decides to use a URL address, they simply need to enter it and then click 'Upload'. On other hand if the user selects to upload one from their computer, it can be done by clicking the marked pc location bar and then a file searcher will open letting the user select the desired file. Once the file is selected and opened then the files pc location will then be displayed in the pc location input section, as shown on the right.

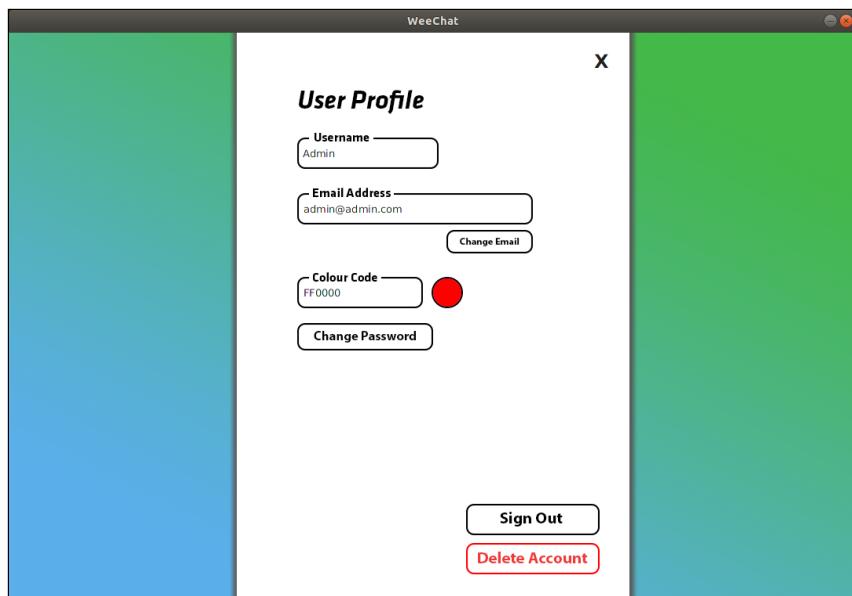
The plugin table displays all added plugins and consists of the plugins name, description, date the plugin was added, enable or disable checkbox and a remove plugin button. Once a plugin has been added to the table the user has the option to either enable or disable a plugin, meaning it doesn't have to be removed to be disabled. Enabled plugins will indicate so with the checkbox having a green fill, while the disabled will remain with the transparent background fill. The table also contains a remove column with a button in each cell that will

remove the corresponding plugin. The data can also be sorted by clicking on the header, while the order of columns can be shuffled around to meet the users requirements.

Name	Descriptions	Date Added	Enable	Remove
Emoji	Adds emojis	15/11/22	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
Fonts	Crazy Fonts	22/12/22	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
RGB Colour	Adds more colour	12/1/22	<input type="checkbox"/>	<input type="button" value="X"/>

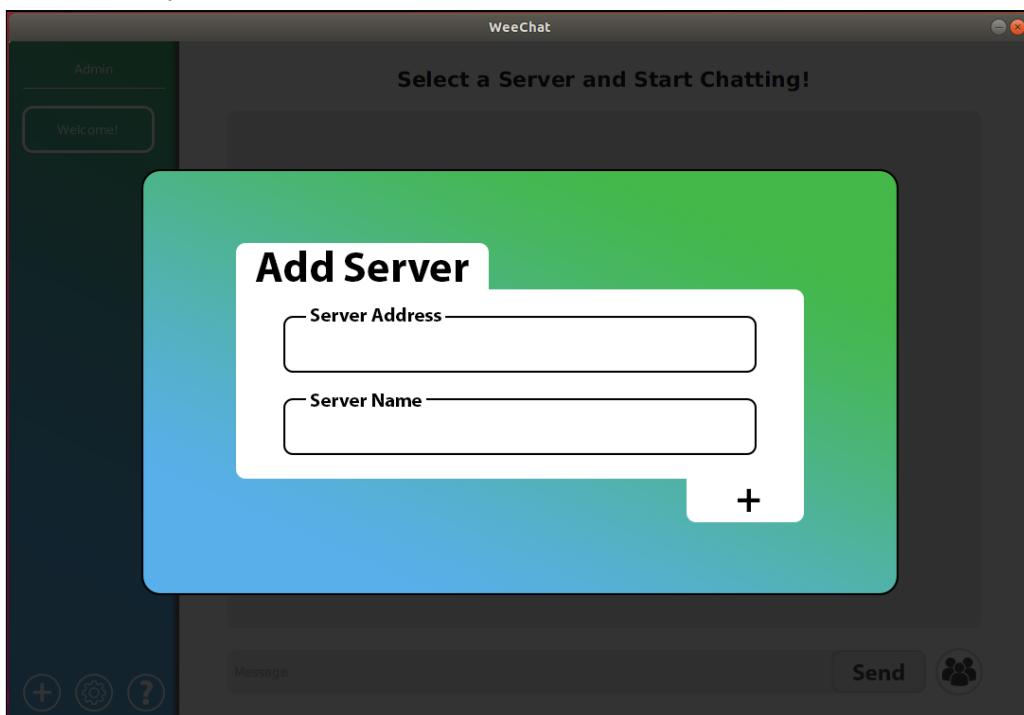


When the user clicks on their username at the top left, the interface will be redirected to the user's WeeChat profile. To leave the user profile interface the 'x' button in the top right, which also uses the 3 state responsiveness, will return the user to the previous interface.

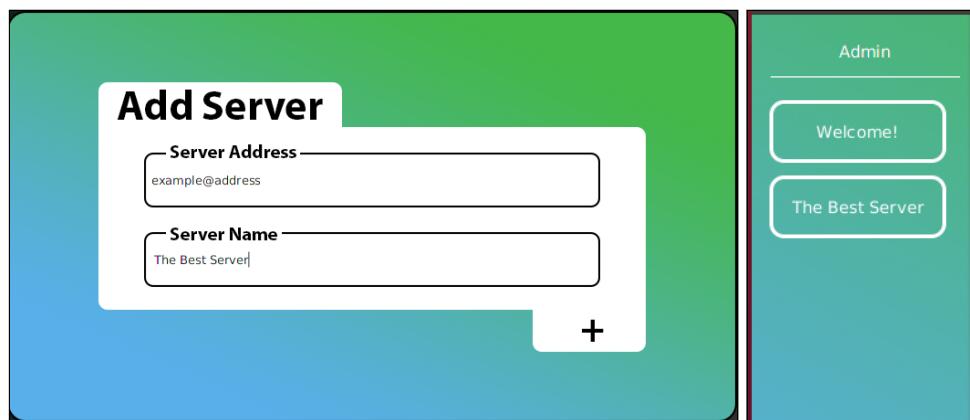


This interface has had little change to the final prototype however, the interface now implements improved UI making it more responsive for the user. This interface lets the user change their email address, password and username colour code. Similarly to the 'register' and 'settings' interfaces, the colour code needs a 6-digit input and once entered will automatically update showing the desired colour in the corresponding circle. The user can update their email address by entering in the new address then clicking the 'change email' button. The options to sign out and delete accounts are located in the bottom right and once clicked will return the user to the login page. If the user signs out they can re-login with their created account while if the user selects delete account that account's login details are lost.

To add a new server the user must click on the '+' icon on the bottom left and then the 'add server' interface will be displayed.



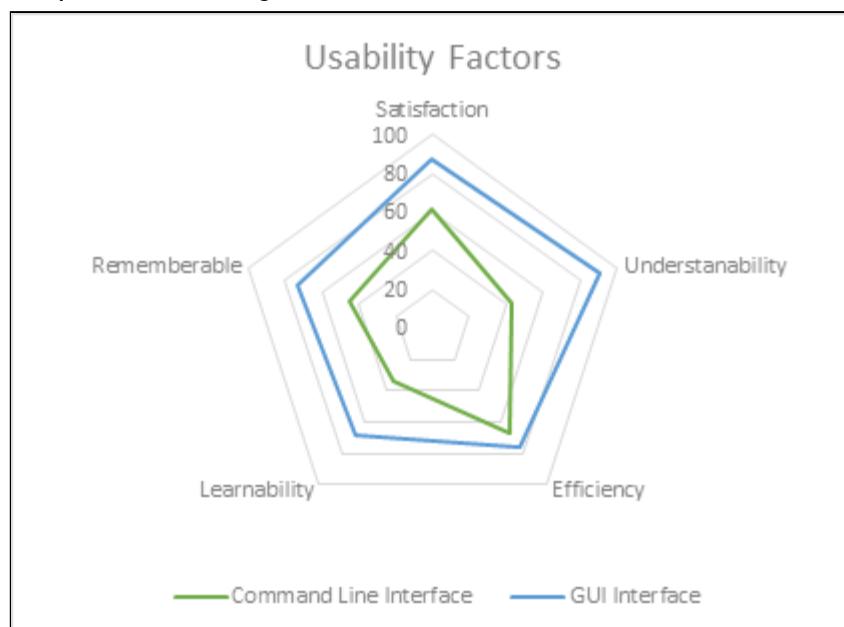
To leave the 'add server' interface all the user must do is click anywhere that is tinted black and then the user will be returned back to the previous interface. The user must fill both the server address and server name text fields, once done then the user must click the '+' button and then the new server will be added to the server list on the left as shown below.



User Involvement

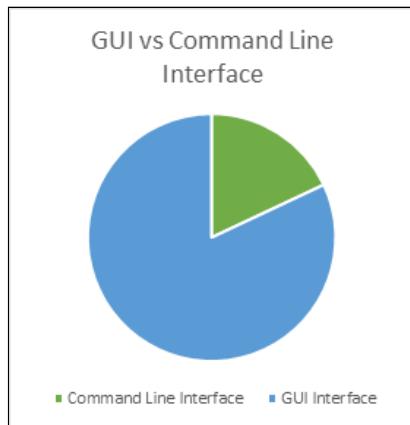
Throughout the design process potential users were asked to test and supply feedback on what they wanted to see in the final design. Changes were made to the final design based on the feedback from the user involvement. Feedback was either directly added to the design, modified to better fit the design constraints or rejected due to being insensible or difficult to implement.

Upon the creation of the final design potential users were asked to rate, out of 100, how they felt about the usability of the new interface compared to the original command line structure. All the scores were tallied up and the averages from each section were added to the below graph.



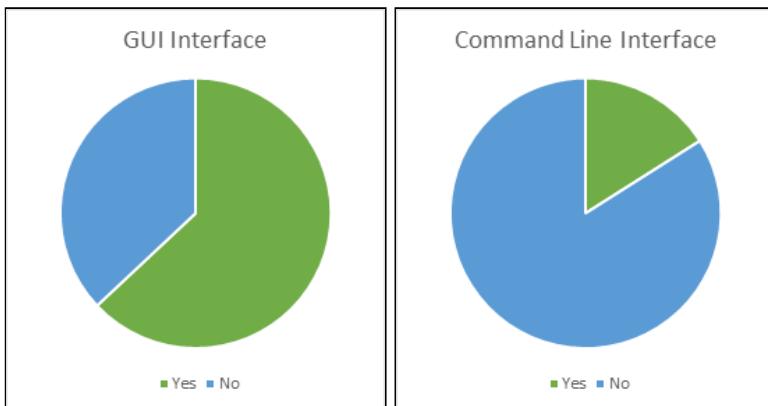
1. Learnability: The system must be easy to learn for both novices and users with experience from similar systems.
2. Efficiency: The system must be efficient for the frequent user.
3. Rememberable: The system must be easy to remember for the casual user.
4. Understandability: The user must understand what the system does.
5. Subjective Satisfaction: The user must feel satisfied with the system.

As the 'Usability Factors' graph shows, the new GUI interface has improved the usability of WeeChat over the original command line interface in all five factors. However, due to improved usability of the interface the sacrifice was the functionality. Not all functionality was able to be implemented into the new interface as doing so would decrease the simple and easy to use structure, so in turn only the main functionalities were added.



When the potential users were asked which interface they preferred the majority said the new GUI interface, due to its simple and straightforward layout and easy to use functionality. On the other side there were a few potential users that performed the original command line interface over the new GUI. When explored why, the main reason was they preferred the functionality of the command line interface over the ease-of-use of the new one, however the majority of potentials users who performed the old interface had some background in command line operation, this meant the ease-of-use for both interfaces were similar and the next important factor is functionality.

When the potential users were asked if they would switch to WeeChat as their new daily messenger platform, 64% of users said that they would make the switch if WeeChat was using the new interface. On the other hand 16% said they would make the switch if WeeChat was using the Command Line interface. Data is shown in the below graphs.



Test Cases

(ID) Name	(1) Login to Existing Account
Prerequisite	- At the 'Login' interface
Steps	<ol style="list-style-type: none"> 1. Enters a eligible username 2. Enters eligible password 3. Clicks the 'LogIn' button
Expected Result	Logs the user in and changes interface to 'Server UI'
Final Result	Pass

(ID) Name	(2) Register a New Account
Prerequisite	- At the 'register' interface
Steps	<ol style="list-style-type: none"> 1. Enter a Username 2. Enter a Email Address 3. Enter a Password 4. Confirm Password 5. Enter valid colour code 6. Agree to terms and conditions
Expected Result	Creates an Account and changes interface to 'Server UI'
Final Result	Pass

(ID) Name	(3) User can Send Message to Server
Prerequisite	- At the 'Server UI' interface
Steps	<ol style="list-style-type: none"> 1. Type message into the message input box 2. Click 'Send' or press enter
Expected Result	Sends a Message to the Server Chat
Final Result	Pass

(ID) Name	(4) Enter Direct Message with Another User
Prerequisite	- At the 'Server UI' interface
Steps	<ol style="list-style-type: none"> 1. Clicks the 'Active Users' button 2. Selects a active user 3. Click on selected user's username
Expected Result	Begins a Direct Message with the Selected User
Final Result	Pass

(ID) Name	(5) Open WeeChat Support Page
Prerequisite	- At the 'Server UI' or 'Direct Message' interface
Steps	<ol style="list-style-type: none"> 1. Clicks the 'Help' button
Expected Result	Opens the WeeChat Support Web Page
Final Result	Fail

(ID) Name	(6) Open WeeChat Settings
Prerequisite	- At the 'Server UI' or 'Direct Message' interface
Steps	<ol style="list-style-type: none"> 1. Clicks the 'Settings' button
Expected Result	Opens the 'Settings' Interface
Final Result	Pass

(ID) Name	(7) Open the User Profile
Prerequisite	- At the 'Server UI' or 'Direct Message' interface
Steps	1. Clicks the user's username/'Profile' button
Expected Result	Opens the 'Profil' Interface
Final Result	Pass

(ID) Name	(8) Add a New Server
Prerequisite	- At the 'Server UI' or 'Direct Message' interface
Steps	1. Clicks the 'Add Server' button 2. Enters a server address 3. Enters a server name 4. Clicks the '+' button
Expected Result	Adds a New Server to the Server List
Final Result	Pass

(ID) Name	(9) Change Server
Prerequisite	- At the 'Server UI' or 'Direct Message' interface
Steps	1. Click on selected server 2. Click on selected Channel
Expected Result	Changes Server
Final Result	Pass

(ID) Name	(10) Change Email Address
Prerequisite	- At the 'Profile' interface
Steps	1. Enters the new email address in the email input field 2. Clicks the 'Change Email' button
Expected Result	The Users Email Address Changes
Final Result	Pass

(ID) Name	(11) Change Username Colour Code
Prerequisite	- At the 'Profile' interface
Steps	1. Enters a valid colour code
Expected Result	The Username Colour Code Changes
Final Result	Pass

(ID) Name	(12) Change User Password
Prerequisite	- At the 'Profile' interface
Steps	<ol style="list-style-type: none"> 1. Click 'Change Password' button 2. Enter old password 3. Enter new password 4. Confirm new password 5. Click 'Change Password'
Expected Result	Changes the Users Password
Final Result	Fail

(ID) Name	(13) Sign Out of Current User
Prerequisite	- At the 'Profile' interface
Steps	1. Click the 'Sign Out' button
Expected Result	User Returns to the 'LogIn' Interface
Final Result	Pass

(ID) Name	(14) Delete Current Account
Prerequisite	- At the 'Profile' interface
Steps	1. Click the 'Delete Account' button
Expected Result	User returns to the 'LogIn' Interface
Final Result	Pass

(ID) Name	(15) Leave the 'Profile' Interface
Prerequisite	- At the 'Profile' interface
Steps	1. Click the 'X' button
Expected Result	Returns User to Previous Interface
Final Result	Pass

(ID) Name	(16) Change Font Size
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' tab
Steps	1. Enters valid font size in font size input field
Expected Result	Changed the Default Font Size
Final Result	Pass

(ID) Name	(17) Change Font Family
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' tab
Steps	1. Enters valid font family in font type input field
Expected Result	Changes the Default Font Family
Final Result	Pass

(ID) Name	(18) Change BackGround Colour
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' tab
Steps	1. Enters valid colour code in background colour code input field
Expected Result	Changes Default Background Colour
Final Result	Fail

(ID) Name	(19) Change Text Colour
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' tab
Steps	1. Enters valid colour code in text colour code input field
Expected Result	Changes Default Text Colour
Final Result	Fail

(ID) Name	(20) Change Selected Notification Options
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' tab
Steps	1. Click selected notification option
Expected Result	Changes Selected Notification State
Final Result	Pass

(ID) Name	(21) Leave 'Settings' Interface
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Settings' or 'Plugins' tab
Steps	1. Clicks the 'X' button
Expected Result	Return User to Previous Interface
Final Result	Pass

(ID) Name	(22) Add Plugin Using a URL Address
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Plugins' tab
Steps	<ol style="list-style-type: none"> 1. Enters URL in URL address input field 2. Clicks the 'Upload' button
Expected Result	Adds Plugin to 'Added Plugins' Table
Final Result	Pass

(ID) Name	(23) Add Plugin Using a PC Location
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Plugins' tab
Steps	<ol style="list-style-type: none"> 1. Clicks on 'PC Location' input field 2. Selects plugin 3. Opens the selected plugin 4. Clicks the 'Upload' button
Expected Result	Adds Plugin to 'Added Plugins' Table
Final Result	Pass

(ID) Name	(24) Change Plugin Enabled State
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Plugins' tab - Added Plugins table is not empty
Steps	<ol style="list-style-type: none"> 1. Clicks the corresponding enable box for selected plugin
Expected Result	Enable Box Changes States
Final Result	Pass

(ID) Name	(25) Remove Plugin From 'Added Plugins' Table
Prerequisite	<ul style="list-style-type: none"> - At the 'Settings' interface - Under the 'Plugins' tab - Added Plugins table is not empty
Steps	<ol style="list-style-type: none"> 1. Clicks the corresponding remove 'X' button
Expected Result	Removes Plugin from 'Added Plugins' Table
Final Result	Fail

(ID) Name	(26) Send a Direct Message
Prerequisite	<ul style="list-style-type: none"> - At the 'Direct Message' interface
Steps	<ol style="list-style-type: none"> 1. Type message into the message input box 2. Click 'Send' or press enter
Expected Result	Sends a Direct Message to Selected User
Final Result	Pass

(ID) Name	(27) Leave 'Add Server'
Prerequisite	<ul style="list-style-type: none"> - At the 'Server UI' or 'Direct Message' interface - Adding a new server
Steps	<ol style="list-style-type: none"> 3. Clicks anywhere in the tinted black
Expected Result	Returns User to Previous Interface
Final Result	Pass

Evaluation

Contributor(s): Ryan Martin

In this evaluation of the design process, factors such as risk management, milestones and the problems encountered will be considered to gage the overall completion of the project to the clients specifications as well as the performance of the team members throughout the project's timeline.

Meetings

Contributor(s): Ryan Martin

Throughout the entire design process the team members kept in close contact with each other to ensure that if any problems, questions or clarification arise they can be quickly answered by other members of the team. The team members also held periodic meetings to confirm important details about the design process like the HTA and the requirement specifications, as well as dividing up new tasks that needed to be completed while also setting or changing the deadlines for current and future tasks.

During the first team meeting it was decided that '*Facebook Messenger*' would be the simplest platform to stay in contact as all members of the team were experienced or had experience with the application which eliminated the need to learn any new programs. Through the use of '*Messenger*' to communicate, it made it easier for team members to provide feedback on someone's work, while also giving the whole team an easy platform to send out announcements that could signal the change of a deadline or for redistributing tasks.

While the face-to-face team meetings made the classification of larger problems and questions as well as the distribution of tasks easier to sort out, organizing times where all members could be present was difficult as all members had different schedules. However, through the use of '*Messenger*' we were able to sort out many of the smaller queries leaving only the larger ones for the team meetings that require face-to-face interaction to find the solutions. With a mixture of both communication types, face-to-face meetings and '*Messenger*', the team was able to stay well connected and sort out any queries that arised.

Risk Management

Contributor(s): Ryan Martin

During the early stages of the design process the team members quickly realized and collaborated on a few risks that could do extensive damage to the project if not managed correctly at an early enough stage.

A noticeable risk that could affect the project's progression is loss of data, which could be due to a number of circumstances such as but not limited to program crashes and file corruption. To help alleviate the risk of data loss it was determined in addition to continuously saving any nearly completed work, but to also either upload any updated work to one drive or to copy all new work

onto a flash drive so that there is always a secure backup increase problems that could arise. This included but is not limited to all low and high fidelity drawings, Netbeans implementations including all the previous versions and any drafts of the report or other work completed.

In addition to each team member's individual data backed up, the main copy of the final report was kept saved in google docs where each team member could access it and make the necessary changes. Each team member was required to download the document, including the data and time in the name, after editing any part of the report to make sure that there was always an up to date backup on at least one team member's computers.

To reduce the risk of an uneven workload for any team member all tasks were divided up based on the complexity of the task and even distributed to each member of the team. By dividing up the tasks based on complexity, every team member would then spend the same amount of time working on the report. If each team member was given the same amount of tasks, this could possibly lead to some team members spending more time on the report due to the variation of complexity which would result in an uneven workload.

Due to the limited time to complete the design process, the team ran the risk of running out of time if no time management was created and managed. Through the use of task deadlines and milestones the team was able to confirm that each member was able to complete all tasks in allotted time so as to not push back any future deadlines which could impact the completion date. While the team was able to keep up with all the allocated tasks, due to some members of the team also having other assignments and tests to complete, less time than what was allocated was spent working on implementation. Due to this some of the features that the client wanted to see in the final implementation had to be either skipped or modified to better fit the remaining time. The team discussed each feature case by case, to determine if the feature should be skipped or if it could be modified and implemented into the design.

Milestones

Contributor(s): Ryan Martin

Milestones were used throughout the design process to keep all the team members on track while also creating an estimated completion date for the client. The start and completion date for every task was collected to improve the estimated time of completion of the overall project. A task/milestone is only marked completed once all members of the team have completed their individual assignments and have incorporated them into the whole. The below table shows all milestones (MS) throughout the project including all tasks (T) that needed to be completed as well as any meetings (MM).

	Start date	Milestone Details	Date of Completion
MM	6th March 2020	Team Meeting	6th March 2020
MS	9th March 2020	Project Proposal	13th March 2020
MM	13th March 2020	Team Meeting	13th March 2020
T	14th March 2020	Survey on Popular Messaging Applications	14th March 2020
MS	14th March 2020	Confirmed Requirement Specifications	15th March 2020

MM	20th March 2020	Team Meeting	20th March 2020
T	21st March 2020	Low Fidelity Prototypes	1st April 2020
T	2nd April 2020	High Fidelity Prototypes	19th April 2020
MS	24th April 2020	Client Meeting	24th April 2020
T	25th April 2020	Incorporate Client Feedback into Design	27th April 2020
MS	27th April 2020	Confirmed HTA	27th April 2020
T	28th April 2020	NetBeans Prototypes	4th May 2020
T	5th May 2020	NetBeans Implementation	8th May 2020
T	9th May 2020	User Experience Feedback	11th May 2020
T	12th May 2020	Project Evaluation	13th May 2020
MS	13th May 2020	Finalise WeeChat Project Report	14th May 2020

Problems Encountered

Contributor(s): Ryan Martin

Throughout the design process very few problems were encountered by the team. This could have contributed to great risk management and foresight from each team member concerning each other's and individual problems.

A notable issue that the team encountered was time constraints on the completion of certain tasks for the prototypes. After the client meeting, each member of the team was given feedback from the client about what they would like to see in the final redesign, however due to the limited time not all of the clients requests have been implemented into the final redesign while some requests had to be slight modified to better fit between what the client wanted and the time reaming.

A significant problem that the team encountered was due to the spread of COVID the client and team were unable to meet for the face-to-face client meeting about discussing the additions that the client would like to see in the final redesign. It was decided by the client and team that an online video conference would be the most optimal solution to overcome this issue.

A minor problem that occurred during the design process was that each member of the team had a different scheduler which made finding a time that all team members could meet for a face-to-face meeting was difficult. This issue of face-to-face meetings was made more difficult with COVID, however the team used '*Messenger*' to cover all the points that would have been brought up in the meetings.

Conclusion

Contributor(s): Ryan Martin

WeeChat, a command line messenger client which allows communication between people connected to the same server and channel, is a powerful software with many features and is nearly fully customizable. The aim with the WeeChat redesign was to improve the user friendly aspect of the program while also keeping all the customizable options and features that WeeChat comes with.

Through the work done by the team as a whole and through the individual efforts, the team was able to design and create five prototypes from the requirements supplied by the client, refined them with user's involvement and later finalized with the final feedback given to the team during the client meeting. All aspects of the design process were explored and documented to help create prototypes that matched the clients specifications for a new interface, as much as possible.

All five implementations of WeeChat have successfully redesigned the interface giving the user a more simplistic and streamlined version of the application. With these redesigns first time or experienced user's now have a more user friendly interface without sacrificing any of the powerful features or limiting any customization that the original Weechat interface came with.

References

Contributor(s): Everyone

The referencing style used is Chicago referencing.

- **GLM** logo made by Kevin Le, for everyone to use.
- **Google Material Design**. 2020. "Material Design Icons".
<https://material.io/resources/icons/> - Kevin Le, Aaron Gangemi - Icons used in prototypes and implementations:
- **Iconfinder** Open source icons used - Terence Marcelo
- **Hexchat.Github.io**. 2020. "Home – Hexchat". <https://hexchat.github.io/>.
- **KiwiIRC**, July 12, 2017 ."KiwiIRC". <https://kiwiirc.com/>.
- **Mirc.Com**. 2020. "Mirc: About Mirc". <https://www.mirc.com/about.html>.
- Sharp, Helen, Yvonne Rogers, and Jenny Preece. 2015. **Interaction Design**: Beyond Human Human-Computer Interaction. 3rd ed. John Wiley & Sons Ltd.
- **Slant**. 2020. "Slant - Hexchat Review". <https://www.slant.co/options/4192/~hexchat-review>.
- **Slant**. 2020. "Slant - Kiwi IRC Review". <https://www.slant.co/options/4194/~kiwi-irc-review>.
- **Slant**. 2020. "Slant - Mirc Review". <https://www.slant.co/options/4185/~mirc-review>.
- **WeeChat**, 2020. "WeeChat, the extensible chat client". <https://weechat.org/>
- WeeChat image obtained from: **Wikipedia**. 2020. "WeeChat".
<https://en.wikipedia.org/wiki/WeeChat>

Appendix

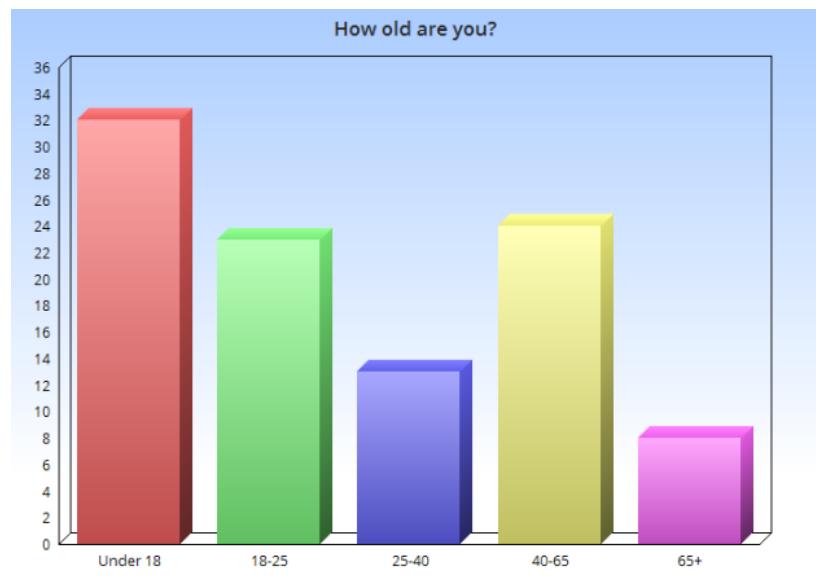
Initial User Survey Responses

Contributor(s): Aaron Gangemi

Our group conducted a survey to ask users about their current form of communication. The survey questions and results are shown below

What is your age? *

- Under 18
- 18-25
- 25-40
- 40-65
- 65+

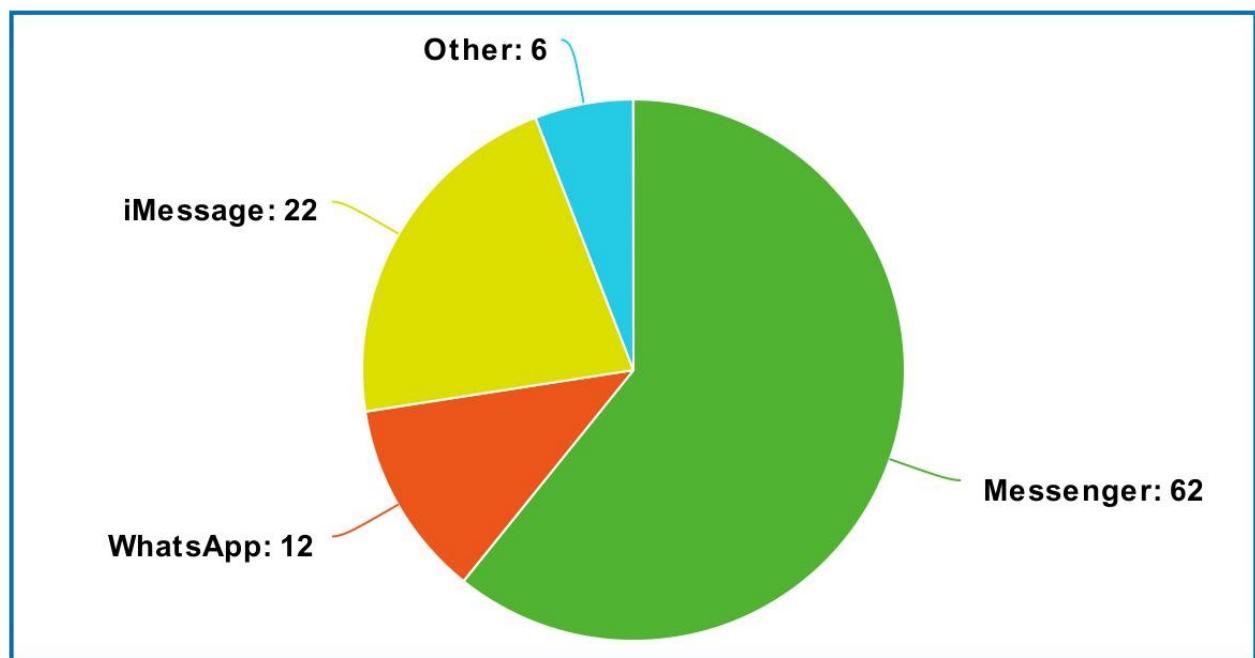


X Data	Y Data
Under 18	32
18-25	23
25-40	13
40-65	24
65+	8

What is your primary form of communication?

- Facebook Messenger
- WhatsApp
- iMessage
- Other

What is your primary form of communication?



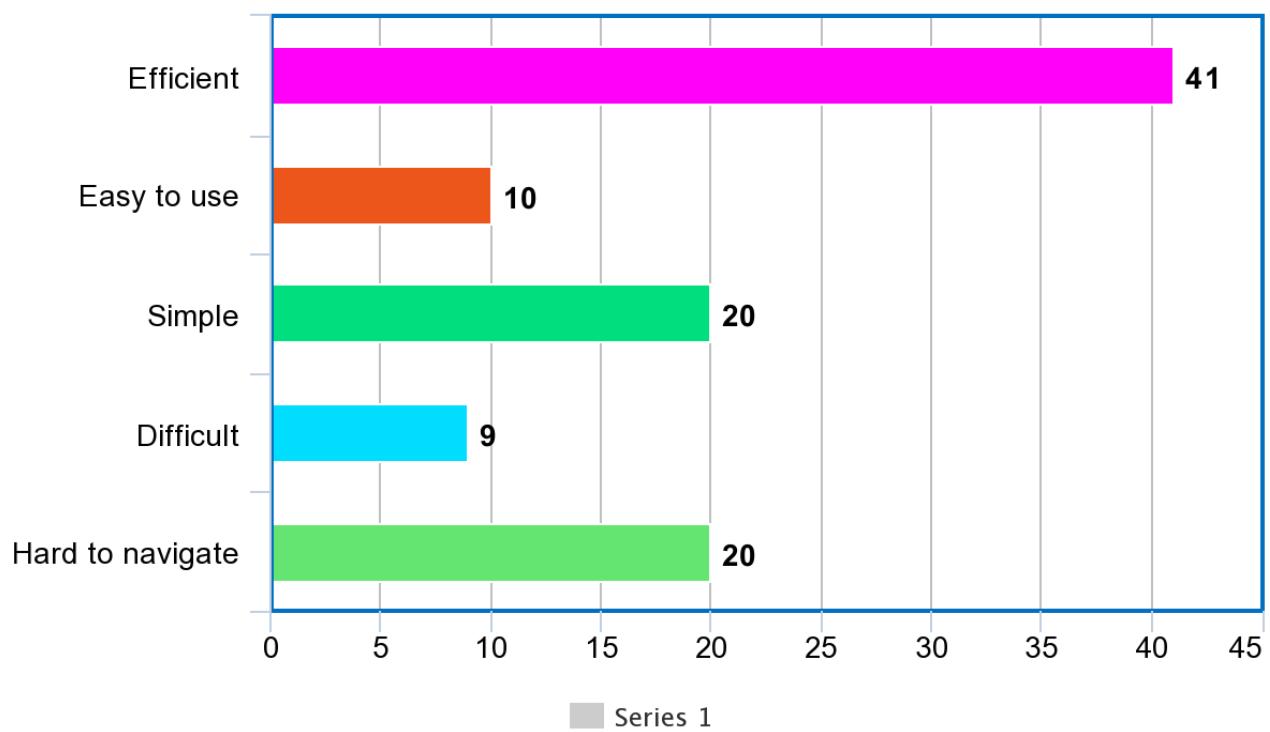
■ Messenger ■ WhatsApp ■ iMessage ■ Other

meta-chart.com

Which statement would best describe your current form of communication?

- Efficient
- Easy to use
- Simple
- Difficult
- Hard to navigate through

Which statement would best describe your current form of communication?

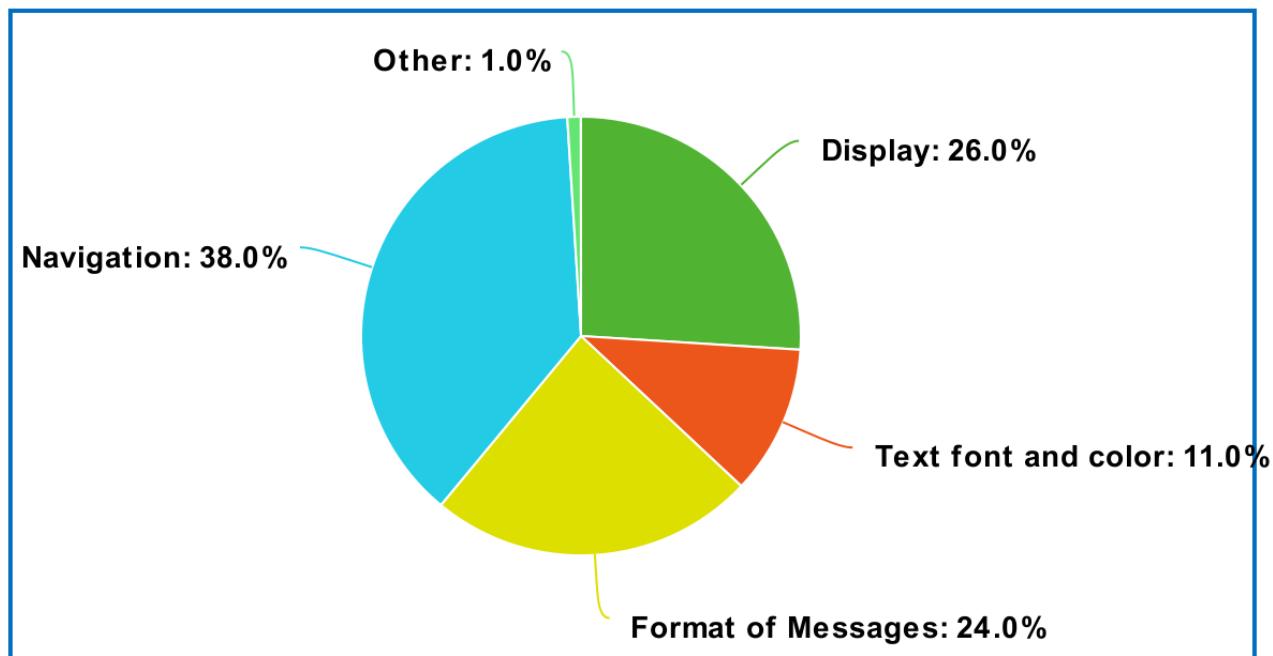


meta-chart.com

What could be improved about your current form of communication? *

- The display
- Text font and color
- Format of messages
- Navigation
- Other: _____

What could be improved about your current form of communication?



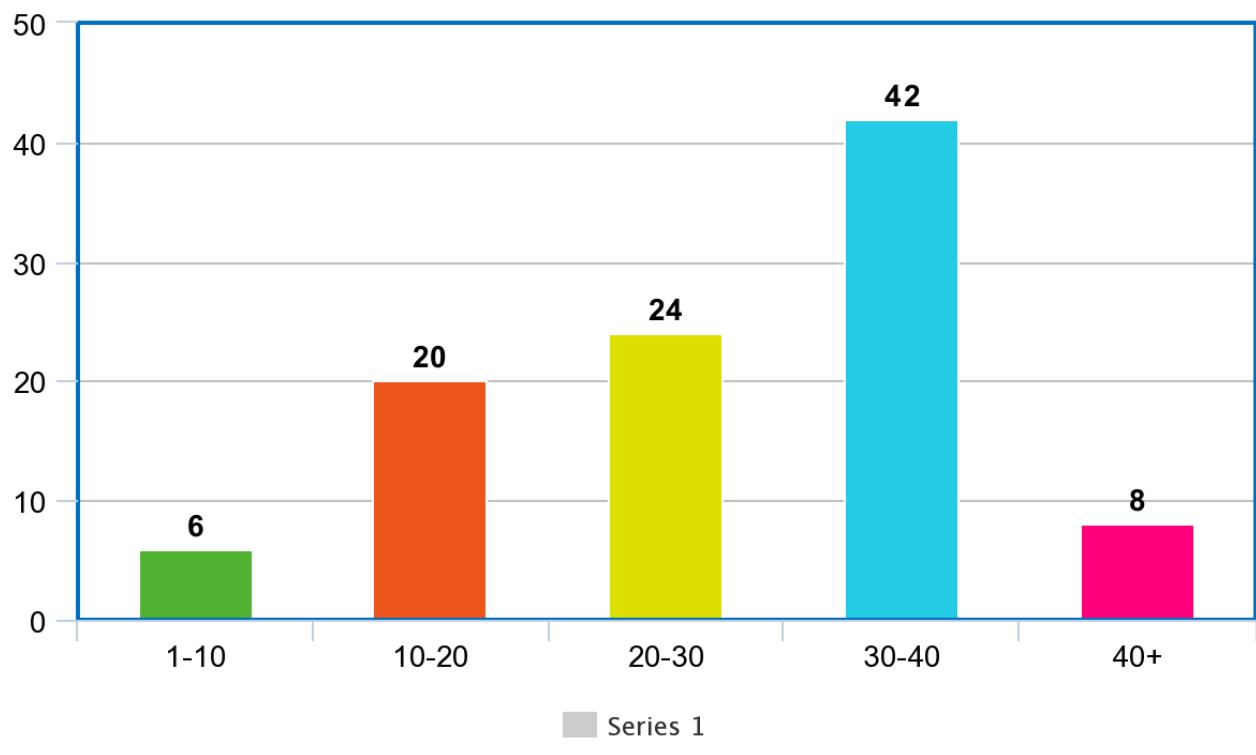
■ Display ■ Text font and color ■ Format of Messages ■ Navigation ■ Other

meta-chart.com

How many text messages do you send per day?

- 1-10
- 10-20
- 20-30
- 30-40
- 40+

How many text messages do you send per day?



Meeting Minutes

Contributor(s): Aaron Gangemi, Kevin Le

Note: Date below in format, MM-DD

2020-01-01 - 2020-12-31

Total 03 h 18 min

Date	Description	Duration
03-06	HCI Meeting 1	1:02:46
	Human Computer Interface Meetings	12:58-14:01
03-13	HCI Meeting 2	1:14:00
	Human Computer Interface Meetings	13:00-14:14
03-20	HCI Meeting 3	1:01:49
	Human Computer Interface Meetings	12:57-13:59

Note: Date in below meeting minutes in format, DD/MM/YYYY

HCI Meeting 1

Meeting Date: 06/03/2020

Meeting Time: 12:58-14:01

Location: Curtin University - T.L. Robertson Library

Group Attendees: Aaron Gangemi, Aaron Giles, Kevin Le, Ryan Martin, Terence Marcelo

Absent: N/A

Agenda

- Meet the HCI group
- Discuss action items.

New Business

- Members introduced themselves to the team
- Discussed potential programs to use

Actions

- Each member contributes ideas to the document:
 - https://docs.google.com/document/d/1aF59_enWrvwHomQSZ8_VPCP6RS-NIB1QB_a4oT3mqef8/edit?usp=sharing
- Will decide on a program to use using Strawpoll once everyone has submitted entries

Agenda for Next Meeting

- Decide on program

Other Notes

- Nil

HCI Meeting 2

Meeting Date: 13/03/2020

Meeting Time: 13:00-14:01

Location: Curtin University - T.L. Robertson Library

Group Attendees: Aaron Gangemi, Aaron Giles, Kevin Le, Ryan Martin, Terence Marcelo

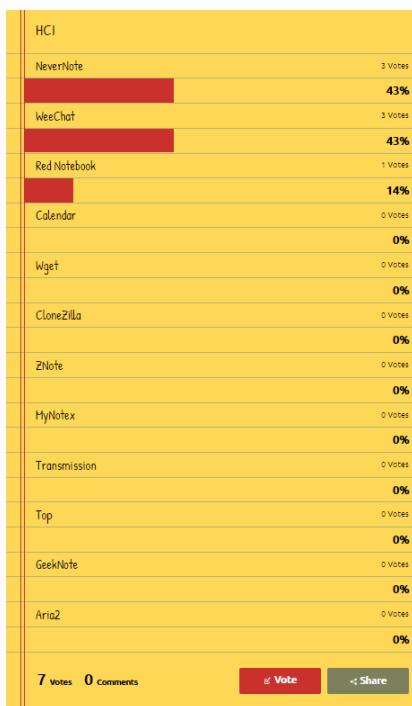
Absent: N/A

Agenda

- Make a final decision on the program we are going to redesign

New Business

- Prepare NetBeans following practical task
- Discuss stalemate: Decided on WeeChat



Actions

- Nil

Agenda for Next Meeting

- Decide on interfaces to design

Other Notes

- Nil

HCI Meeting 3

Meeting Date: 20/03/2020

Meeting Time: 12:57-13:59

Location: Curtin University - T.L. Robertson Library

Group Attendees: Aaron Gangemi, Aaron Giles, Kevin Le, Ryan Martin, Terence Marcelo

Absent: N/A

Agenda

- Decide on 5-10 interfaces

New Business

- Interfaces settled on:
 - Login
 - 2 input fields
 - Remember login
 - Forgot Username/ Password
 - Register Button
 - Register
 - Username
 - Email
 - Password
 - Password Confirm
 - Box T&C
 - Pick username colour
 - Options
 - Customizations
 - Font size/ type
 - Background Colour
 - Text colour
 - Plugins
 - Search online
 - Upload
 - Plugin/ Script manager
 - List of:

- Script name
 - Script Date Uploaded
 - Script Description
 - Command to run
- Notifications
 - Mentions
 - Direct Messages
- About
 - Text
- Server UI
 - Server List
 - Channel List
 - Chat
 - Username
 - Time
 - Message
 - User List on Right
- Direct Message
 - Same as Server UI chat without channels, user list
- Profile
 - Username
 - Email
 - Password
 - Username Colour Picker
 - Sign Out
 - Delete Account
- Add Server
 - Server Name
 - Field

Actions

- Each member goes and develops their prototypes, and following client approval, their implementations.
- Following that, the report will be worked on.

Agenda for Next Meeting

- Nil

Other Notes

- Nil

Client Meeting

Meeting Date: 24/04/2020

Meeting Time: 13:00-13:59

Location: Remote via Collaborate Ultra

Group Attendees: Aaron Gangemi, Aaron Giles, Kevin Le, Ryan Martin, Terence Marcelo, Antoni Liang

Absent: N/A

Agenda

- Client (Antoni) feedback on prototypes we present
- Any questions for client

New Business

- Received feedback for each member regarding:
 - Layout/ structure
 - Information cluster
 - Navigation
 - Icons
 - Colour choices/ Themes
 - Font Size/ Colour
 - UI Feedback
 - Help (Regarding how program is used)
- Identified what the client believes is missing
- Team didn't have any questions for client
- Feedback was overall positive

Actions

- Team goes off and refine their prototypes based on client feedback

Agenda for Next Meeting

- Nil

Other Notes

- Nil