|  |
| --- |
| East London Science School |
| A-Level Computer Science Project |
| JarChat |

|  |
| --- |
| Adam Tazul  [Date] |

Contents

[1: Analysis 2](#_Toc86858955)

[1.1: Aims of JarChat 2](#_Toc86858956)

[1.2: Target Audience 2](#_Toc86858957)

[1.3: Research 2](#_Toc86858958)

[1.3.1: Existing IRC Clients 2](#_Toc86858959)

[1.3.2: Features of JarChat 4](#_Toc86858960)

[1.3.3: Limitations of JarChat 4](#_Toc86858961)

[1.4: Requirements 4](#_Toc86858962)

[1.4.1: Software 4](#_Toc86858963)

[1.4.2: Hardware 4](#_Toc86858964)

[1.5: Success Criteria 5](#_Toc86858965)

[2: Design 5](#_Toc86858966)

[2.1: UI Design 5](#_Toc86858967)

[2.1.1: Start-up Screen 5](#_Toc86858968)

[3: Development & Testing 5](#_Toc86858969)

[3.1: Stage 1: Connection to IRC 5](#_Toc86858970)

# 1: Analysis

## 1.1: Aims of JarChat

JarChat will aim to be a cross-platform IRC client which will support Windows, MacOS, Linux, and maybe more operating Systems not already listed. This will have the benefit of native cross-platform support as it is built-in to Java. Java’s JDK version 8 (Java version 1.8.0 release 301) will be used to compile all binaries as most computers still run JDK version 8, even though Oracle has moved their LTS (Long Term Support) version to JDK 11.

## 1.2: Target Audience

I expect everyone who uses a computer to be able to take advantage of JarChat. While most services have moved their IM services to other platforms (Microsoft Teams, Skype, Discord, TeamSpeak, etc.), various FOSS (Free and Open-Sourced Software) still use IRC as their support protocol, taking advantage of IRC’s decentralized nature.

## 1.3: Research

### 1.3.1: Existing IRC Clients

#### 1.3.1.1: Windows

Existing Windows clients include mIRC and XChat.

##### 1.3.1.1.1: mIRC

mIRC is a paid and proprietary software which costs £17.94 (though it offers a 30-day free trial). It provides a slightly dated User Interface and was compiled in 32-bit form. This makes it compatible with 32-and-64-bit versions of Windows from Windows XP up to Windows 11. The UI makes use of a template too store different tabs as windows on a canvas. These windows cannot be moved or used in other parts of Windows. The settings menu leaves much to be desired and leaves few customization options. There is no option for IRC’s VOIP feature included with the software at any point.

Figure 1: Default look of mIRC, running on Windows 11 build 22449.1000 (Dev Insider Preview)

I intend to take inspiration from mIRC's raw functionality as it just plain works (albeit only on Windows). I do plan, however, to expand this functionality to take advantage of all of the features that are included on IRC’s featureset

##### 1.3.1.1.2: XChat

XChat is another IRC client. Written in C, the Linux version is free and open-sourced under the GNU GPLv2 license. The client is precompiled for Windows and Fedora GNU/Linux, with forks available for Arch Linux (in the AUR), and with a precompiled \*.deb file in the official Debian and Ubuntu repositories. The Windows version of the software is closed-sourced and proprietary, costing users US$19.99 (equivalent to £14.42 in September of 2021), though it offers a 30-day free trial. The Windows version officially supports all Windows versions from Windows 2000 up to Windows 10 (all in 32-bit).

Figure 2: The first screen of the Windows version of XChat, running on Windows 11 build 22449.1000 (Dev Insider Preview)

I tried to use XChat to connect to my favorite IRC server, but the Windows version fialed to do so on Window sversions 7, 8.1, 10, and 11. This lack of ease of use is an issue that I would like to resolve with JarChat.

#### 1.3.1.2: Linux

Existing Linux IRC clients include Konversation, WeeChat, and HexChat.

##### 1.3.1.2.1: Konversation

Konversation is a Linux IRC client which is tightly integrated with the KDE Plasma Desktop Environment. It is fully free & open-sourced under the GNU GPLv2 license. As a part of the KDE Applications suite. Konversation is precompiled for all major distributions of Linux & GNU/Linux, with source code hosted on the official KDE git repository ([https:/invent.kde.org/network/konversation](https://invent.kde.org/network/konversation)).

I use Konversation myself frequently as my main desktop setup is Artix Linux with KDE Plasma, and its tight integration means accurate and appropriate theming in line with the global QT5 theme, no matter what theme is used. It also uses Kcolorchooser to allow for the user to create a custom colour for Konversation’s theme, even if the user is not using a QT5-based theming engine.

Using Konversation, I found that it is a fully-featured client with support for every part of IRC, displayed in a user-friendly layout that is easy-to-use. I intent to take inspiration from this and remake these features in JarChat.

##### 1.3.1.2.2: WeeChat

WeeChat is a Terminal User Interface IRC client for Linux which can be used on any Linux Distribution without the need for any Desktop Environment or Window Manager. It is also FOSS under the GNU GPL v3 license.

While strictly TUI and Keyboard Interaction only, WeeChat still provides a fully-featured experience with support for every part and situation possible in IRC, with every customization as needed by anyone who would use it. Since WeeChat is TUI, it is aimed at people who are willing to learn how to use it, which can take some time. For this very reason, there are a lot of people who have installed it, tried to use it, and then immediately uninstalled it as they found it extremely difficult to get started in. This is completely understandable as WeeChat was designed with minimalism at the forefront of the developers’ minds

##### 1.3.1.2.3: HexChat

HexChat is a cross-platform IRC client which supports all Linux Distributions via a Flatpak, its source code, .deb file, the official Arch Linux repositories, and more. It is also FOSS under the GNU GPL v2 license and features full support for Windows 7, 8, 8.1, 10, and 11. While it has Windows support, I am testing it on my install of Artix Linux.

#### 1.3.1.3: MacOS

Even though there are one or two IRC clients for MacOS, I do not have the means to test this. Creating virtual machines with MacOS installed on them have proved to create bugs at best, and hackintoshing is also problematic, with stringent hardware requirements I do not meet. In any case, trying to do either will result in me breaking Apple’s EULA, which is legally binding.

### 1.3.2: Features of JarChat

JarChat is intended to include:

* Basic functionality as an IRC Client
* Cross-compatibility between various different operating systems without compromising on features
* Vast configurability through an easy-to-use UI
* GNU GPLv3 Open-Sourced license to allow for the community to better help shape the future of JarChat

### 1.3.3: Limitations of JarChat

The main limitation of JarChat is the fact that it will be written in Java, which means that JarChat will require and expect the user to have a JVM compatible with Java version 1.8.0 release 301 already installed on their system. While Java is generally installed on millions of computer systems globally, it cannot be guaranteed that everyone will be able to use JarChat for lack of the ability to install the software.

## 1.4: Requirements

### 1.4.1: Software

* Any Desktop OS made from the year 2000 onwards
* Java version 1.8.0 release 301

### 1.4.2: Hardware

* Any CPU + RAM + GPU + Motherboard + PSU configuration able to handle the OS’s System requirements
* At least 512**MiB** free RAM (TBD)
* TBD secondary storage space free

## 1.5: Success Criteria

In order to be successful, JarChat will need to:

* Function as a fully working IRC client
* Be usable for everyone who would need to use it
* Be able to use SSL encryption for servers which use it
* Run on any OS that Java version 1.8.0 release 301 can be installed on

In order to check that each criterion is met, JarChat will need to

* Be tested in every aspect that is needed of any IRC client
* Be distributed to various users with various skill levels in computer usage
* Be tested on every OS JarChat is likely to be run on
* Be testing in connections with servers that use SSL encryption

# 2: Design

## 2.1: UI Design

### 2.1.1: Start-up Screen

On startup, I intend for JarChat to show the main window where server output will go, with a message in there if the user has not added any IRC servers. Alongside the main window,

Figure 3: First sketch of how the UI is going to look like, as drawn in Paint 3D

# 3: Development & Testing

## 3.1: Stage 1: Connection to IRC

The actual connection to IRC is going to be handled by the pircbotx library for Java (an open-sourced library hosted on GitHub here: <https://github.com/pircbotx/pircbotx>). This library has extensive documentation on GitHub (at <https://github.com/pircbotx/pircbotx/wiki/Documentation>, <https://github.com/pircbotx/pircbotx/wiki>, and <http://pircbotx.github.io/pircbotx/latest/apidocs/>), and has been actively maintained since 2015 so it is a good choice to use and still has support.