This report and the software it documents are the result of my own work. Any contributions to the work by third parties, other than tutors, are stated clearly below this declaration. Should this statement prove to be untrue I recognise the right and duty of the Board of Examiners to take appropriate action in line with the university’s regulations on assessment.

Name: Jacob Matt Fright.…………………………………………………………….ID No: N0663809

Name: Allan Moses Fernandes…………………………………………………….ID No N0979708

[Introduction 2](#_Toc87490781)

[Background Research 2](#_Toc87490782)

[Design 2](#_Toc87490783)

[Requirements Gathering 2](#_Toc87490784)

[Functional Requirements 2](#_Toc87490785)

[Information Oriented 2](#_Toc87490786)

[Process Oriented 2](#_Toc87490787)

[Non-Functional Requirements 2](#_Toc87490788)

[Operational 2](#_Toc87490789)

[Performance 2](#_Toc87490790)

[Security 2](#_Toc87490791)

[Cultural and Political 2](#_Toc87490792)

[Task List 3](#_Toc87490793)

[Time Planning 3](#_Toc87490794)

[Coding Standards 3](#_Toc87490795)

[Implementation Details 3](#_Toc87490796)

[Use Case 3](#_Toc87490797)

[Activity Diagram 3](#_Toc87490798)

[Class Diagram 3](#_Toc87490799)

[Testing Details 3](#_Toc87490800)

[Unit Tests 3](#_Toc87490801)

[Integration Tests 3](#_Toc87490802)

[Acceptance Tests 3](#_Toc87490803)

[Results 3](#_Toc87490804)

[Conclusions 3](#_Toc87490805)

[Future Work 3](#_Toc87490806)

[References 3](#_Toc87490807)

# Introduction

# Background Research

## Ubuntu 20.04 LTS

## QT

## Netbeans

## GIT

## BOOST

## Doxygen

## MVC (Model, View, Controller)

# Design

## Requirements Gathering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Priority | Requirement | Implications | Tasks |
| 1 | Must | Have friendly user interface | Allows the user to navigate the application | T1.1: Create login screen  T1.2: Create home screen  T1.3: Create user search screen  T1.4: Create message screen |
| 2 | Must | Users must be able to create an account | Gives user access to the private account | T2.1: Create user information sheet  T2.2: Create password verification  T2.3: Create scripts for generating accounts  T2.4: Create scripts for encrypting password  T2.5: Create scripts for saving user details to database |
| 3 | Must | Users must be able to log in | Give user access to their account | T3.1: Create scripts to pass login request  T3.2: Create scripts for decrypting password  T3.3: Create script to verify login details  T3.4: Create script to update users' activity status in the database |
| 4 | Must | Users must be able to log out | Allows users to log out of their account  Protects account from 3rd party use | T4.1: Create script for updating user profile to show they are offline  T4.2: Create script for taking user back to the login page |
| 5 | Must | Search for other users | Allows the user to contact other users | T5.1: Create script to compare search parameters to database  T5.2: Create script to display user icons matching the search results |
| 6 | Must | View active users | Allows a user to view which other users are online | T6.1: Create script to receive users' activity status from database  T6.2: Display icon next to username to show logged in status |
| 7 | Must | Initiate chat | Allows users to send messages | T7.1: Create script to handle clicking user profile to take you to messaging screen  T7.2: Create script to handle making TCP link  T7.3: Create script to handle passing message to server  T7.4 Create script for handling receiving message from the server |
| 8 | Must | Run in a Linux OS | Client has specifically requested this | T8.1: Design the system with Linux  T8.2 Implement the system in a Linux environment |
| 9 | Must | Communicate between C++ and Java | Client has specified this | T9.1: Build client/server in C++  T9.2: Build client/server in Java  T9.3: Check communication between clients |
| 10 | Should | Add multiple users to chat | Allows users to make group chats | T10.1: Create script to add user IDs to group chat  T10.2: Create scripts to handle multiple concurrent user inputs to chat log |
| 11 | Could | View chat history | Allows users to view previous messages | T11.1: Create script to return chat log  T11.2: Create scripts to display chat log in chat screen |
| 12 | Could | Send messages offline | Allows users to send messages to offline users | T12.1: Test messages get saved to chat log when the other user is offline  T12.2: Create script to highlight unread messages when user logs in  T12.3: Create scripts to move unread messages to the top of the user list |
| 13 | Could | Send emojis | Allows the users to send emojis as part of their chat | T13.1: Upload emoji library  T13.2: Link key combinations to emoji library items |
| 14 | Should | Update details | Allows the user to update their details and profile picture | T14.1: Create user profile screen  T14.2: Create scripts to update details  T14.3: Create scripts to upload/update image |

## Task List

## Time Planning

## Coding Standards

# Implementation Details

## Use Case

## Activity Diagram

## Class Diagram

# Testing Details

## Unit Tests

## Integration Tests

## Acceptance Tests

## Results

# Conclusions

# Future Work

# References

[How to Prepare Functional Requirements for Your Messaging App (yalantis.com)](https://yalantis.com/blog/writing-software-requirements-for-your-messaging-service/)

Project Plan: - Requirement's list -

Project Risks - Mitigation plan -

Use of monitoring tools. -

Gantt Chart

Audience-Oriented Requirements:

* Business requirements
* User requirements
* Functional requirements
* Non-functional requirements
* Implementation

FURPS:

* **F**unctionality
* **U**sability
* **R**eliability
* **P**erformance
* **S**upportability

FURPS+:

* Design constraints
* Implementation requirements
* Interface requirements
* Physical requirements

Common Requirements

* Screens
* Menus
* Navigation
* Workflow
* Login
* User types
* Audit tracking and history
* Archiving
* Configuration