Josh Evans

# Term Project: *IRRC*
Test Plan Document

# Table of Contents

# 1 Introduction

## 1.1 Purpose and Scope

The purpose of this document is to outline the test strategy and overall test approach for the IRRC project. This includes test methodologies, traceability, resources required, and estimated schedule.

## 1.2 Target Audience

The audience of this document is the project team and the project management team. This document is also written for the extended test team. The test lead, testers, and any outsourced testers should be able to utilize this document to understand the scope of work that must be accomplished by the test team. The document is intended to accomplish its purpose only for the intended audiences.

# 2 Test Plan Description

## 2.1 Scope of Testing

The test plan will extensively test the visible functions of the framework from the client's perspective, and therefore test the back end of the server as well. Library methods from javafx or any other native methods will not need to be tested.

## 2.2 Testing Schedule

Functional prototype completion – client/server pair testing

Functional server completion – netcat testing with fake client

Functional client completion – netcat testing with fake server

Total completion – pair testing with real client/server

## 2.3 Release Criteria

The IRRC system must implement all design requirements with no exceptions or errors by release.

# 3 Unit Testing

This section will outline the units of the infrastructure that will require testing.

## 3.1 Server

The server and its backend features will need to be thoroughly tested before the completion of the client. Features that will need to be tested include the user infrastructure as well as the connection protocol. This user infrastructure includes the online userlist, channels, allchat messages, and direct messages between users. The connection protocols include correctly receiving and handling connections from clients, and correctly handling disconnects/reconnects.

## 3.2 Text Client

The text client is the prototype to the GUI client – a fully functional client to connect to the server without the javafx GUI wrapping. Since this is fully featured, all the features of the client excluding GUI-specific features must be tested.

## 3.3 GUI Client

The GUI client must prove that it implements all of the features of the text client, as well as show that it also implements the GUI-specific features from the requirements.

# 4 Integration Testing

## 4.1 Connecting

Use case 1 – the client attempts connection to the server.

## 4.2 Basic Chat

Use case 2 – chat messages are received from the client to the server.

## 4.3 Channels/Users

Use case 3 – the server sends channel and user data to the client to be displayed by the UI.