Testing Plan Document

1. Introduction

1.1The purpose of this document is to provide a comprehensive test plan for the software project being developed by Team Clack. This test plan outlines the scope of the testing effort, the objectives of testing, the testing methodologies and techniques, and the test execution procedures and reporting. The document also includes an overview of the software being tested, as well as any assumptions and dependencies related to the testing effort.

1.2 The testing effort for this project is critical to ensure that the software meets the functional and performance requirements, as well as to identify and resolve any defects or issues that may impact the user experience. The testing effort will involve multiple levels and types of testing, including unit testing, integration testing, system testing, functional testing, and performance testing.

1.3 The software project being developed is a client-server-based application designed to provide users with a platform to chat and make friends either individually or in groups. The application includes features such as logging in and out, password recovery, and the ability to manage group chats among other things. The software is being developed using an Agile development methodology, with regular releases and iterations throughout the development process.

Test Plan Overview

2.1 Objectives of Testing

The objectives of testing for this project are to:

\*Validate that the software meets the functional and non-functional requirements

\*Verify that the software is stable and reliable

\*Ensure that the user interface is intuitive and user-friendly

\*Identify and report any defects or issues found during testing

\*Verify that the software is compatible with Windows 10 and above

2.2 Assumptions and Dependencies

The following assumptions and dependencies have been identified for the testing effort:

\*The testing team will use Python 3.10 for the build

\*The testing environment will be set up on Windows 10 or higher

2.3 Test Environment

The following test environment will be used for the testing effort:

\*Hardware: Standard desktops and laptops with Windows Operating systems, 10 or higher

\*Software: Latest versions of the software being tested, as well as any necessary test automation tools and frameworks

\*Test data: Test data will be created and maintained by the testing team, and will include realistic scenarios and use cases

\*Test infrastructure: The testing environment will be set up and maintained by the development team, and will include any necessary servers or databases for testing.

3. Testing Methodology and Techniques

3.1 Testing Methodology

The testing methodology for this project will follow an Agile testing approach. This approach emphasizes continuous testing throughout the software development lifecycle, with an emphasis on collaboration and communication between the development and testing teams.

The Agile testing methodology involves the following key practices:

\*Early and continuous testing

\*Collaborative testing

\*Exploratory testing

\*Continuous feedback and improvement

3.2 Testing Techniques

The following testing techniques will be used to ensure comprehensive testing coverage:

\*White box testing: This technique involves testing the software with knowledge of its internal workings. It will be used to verify that the software is stable and reliable, as well as to identify any defects or issues related to the software architecture or code.  
\*Black box testing: This technique involves testing the software without the knowledge of its internal workings and will simply check that the classes and methods produce the right output with a given input. These tests are a select small, manageable set of test cases that maximize the chances of detecting a fault, while also minimizing the chances of wasting a test case.

\*Regression testing: This technique involves re-testing previously tested functionality to ensure that it still works as expected after changes have been made to the software.

\*Exploratory testing: This technique involves ad-hoc testing of the software, with an emphasis on creativity and spontaneity. It will be used to identify any defects or issues that may not be discovered through scripted testing.

3.3 Test Data Management

Test data will be managed by the testing team, with a focus on creating realistic and diverse data sets that cover a wide range of scenarios and use cases. Test data will be stored and maintained in a secure and accessible location, with appropriate access controls and backups in place. The testing team will also ensure that test data is properly cleaned and reset after each test cycle to ensure the integrity of the testing environment.

4. Test Requirements

4.1 Functional Requirements

The following functional requirements will be tested to ensure that they meet the specified requirements:

\*The software shall allow users to sign up and log in with a valid username and password.  
\*The software shall allow users to log out

\*The software shall allow users to reset their password.

\*The software shall allow users to send messages

\*The software shall allow users to view their own messages and messages from other users.

\*The software shall allow users to send and view messages in group chats.  
\*The software shall allow group admins to kick users or transfer their adminship to another group member.  
\*The software shall allow users to leave group chats and invite other users to group chats

\*The software shall allow users to see which of their friends are online.

\*The software shall allow users to add other users as friends and remove them

4.2 Non-Functional Requirements

The following non-functional requirements will be tested to ensure that they meet the specified requirements:

* We will use Kanban for our software development process
  + Using a Trello board
  + We will have 2 weekly meetings (one on Wednesday at 7:30pm and another on Friday at 1pm), which group members will be expected to participate
* The program must be able to run on Windows 10 and above
* The program must be made in Python 3.10
* Chat history must be stored indefinitely.
* Another user’s messages must be received within a second.
* The app can be used anytime anywhere.
* Files and images can be shared easily within a few clicks.
* All the private information must be kept strictly confidential.
* User account must be linked to a valid email address.
* The system must be able to handle up to thousands of online users at a time
* Viewing the online status of friends should take no more than 1 click to view
* The system must not let users have the same username as other other users

4.3 Test Scenarios

The test scenarios will follow the use case scenarios, which are listed in section 4.1

4.4 Test Cases

The test cases will follow the use case scenarios, which are listed in section 4.1, and will be used to validate the test scenarios

5. Conclusion

In conclusion, this test plan document has outlined the test strategy and approach for testing the software. The functional and non-functional requirements have been identified, and test scenarios and cases have been developed to ensure that the software meets these requirements.

Additionally, it should be noted that there are some features that may be developed in the future, such as the following:

1. Upload images
2. Upload files
3. Delete messages
4. Change username
5. Blocking user
6. Editing a message
7. Voice chat
8. Servers (aka group chats with multiple text and voice chats with a role hierarchy)
9. Pin chat messages (so they are easier to access)
10. Delete account

These features are not included in the current version of the software and will not be tested as part of this test plan document. However, they shall be taken into consideration for future testing and development efforts.

Thank you for reading this test plan document. We look forward to successful testing and a successful release of the software.