Test Report

SOFT20091: Software Design & Implementation

Group 30

Hannah Ashna Jacob (N0865554)

Hassaan Naveed (N0898071)

Jarad Johnson-Bailey (N0853071)

Nicholas McCaig (N0787115)

2021

Contents

[Introduction 2](#_Toc67823842)

[Basic Functionalities 3](#_Toc67823843)

[Advanced Functionalities 16](#_Toc67823844)

[Results Analysis 20](#_Toc67823845)

# Introduction

The following document contains the alpha tests for the messaging application project.

# Summary

The following tests are alpha tests as they are acceptance testing done at the end of the implementation by internal members of the organisation testing that the application meets the requirements. Therefore, there is one test for each requirement detailed in the planning document. The testing technique used is grey-box testing, as while there is some awareness of the internals being tested and the testing is targeted to an extent, such as with white-box testing, this is not all encompassing, and many tests are somewhat generic instead. While most tests are approached with a black box method - running the application and testing a functionality - some white box techniques are used or are recommended to support the testing. Unit tests were also added to the application and test some of the basic functionality. The table seen below (Figure 1) is used to detail all the alpha tests, which have been split by their requirements priority, and are reflected in the project plan document.



Figure Testing Table Layout

# Basic Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 01 | **Description** | Users must be able to send, receive and view messages through the application. |
| **Test Type** | Qualitative | **Success Criteria** | Message can send successfully and be received by another successfully |
| **Number of attempts** | 1 | **Comments** | This test uses a combination of white and black box testing.  Both users able to send, receive and view messages without issue |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Server functionality – message passing | | |
| **Setup Instructions** | Test run the application with the MQTT server active with multiple clients to test whether it is functional.  Prior to alternative – UI based message sending functionality, may use direct input to pass the message over command line. | | |
| **Failure Correction Procedure** | Verify client and server source code for message sending and receiving for errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Both users can send and receive messages:    Figure - Test 01 - Users able to send and receive messages successfully | | |
| **Test Date** | 27/03/2021 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 02 | **Description** | Users must be able to create chat rooms (rooms with more than two contacts) |
| **Test Type** | Qualitative | **Success Criteria** | User able to create chatroom and add at least 2 other users. |
| **Number of attempts** | 1 | **Comments** | all users added successfully and able to send and receive messages within the channel. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Server functionality  Chatroom functionality | | |
| **Setup Instructions** | Create chatroom with user, add multiple other users to chatroom. Send test message, check that other users added can view the chatroom and sent message. | | |
| **Failure Correction Procedure** | Verify client and server source code for message sending and receiving for errors. Verify user and chatroom classes for errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 02 - Adding a Room    Figure - Test 02 - Adding a Channel.    Figure - Test 02 - Adding User    Figure - Test 02 – All members successfully able to see the room and channel and send messages | | |
| **Test Date** | 27/03/2021 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 03 | **Description** | The user that creates the chat room must be classified as Admin. |
| **Test Type** | Qualitative | **Success Criteria** | When a user creates a chatroom, they are classified as an admin and has access to all admin capabilities. |
| **Number of attempts** | 3 | **Comments** | Able to check if admin by viewing permissions. Gray box test as viewed the files.  Failed [Resolved] – admin file was not updated when user made room to make them admin, when refreshing after making room the user no longer had admin access. Issue has been resolved. |
| **List of equipment / requirements** | Computer running Ubuntu Image  QT Creator  Server functionality  Chatroom functionality  Admin and User classes | | |
| **Setup Instructions** | Run application and create a chatroom. | | |
| **Failure Correction Procedure** | Verify server for errors and user, admin, and chatroom classes for errors. | | |
| **Engineer(s) / Technicians** | Hannah Ashna Jacob | | |
| **Individual Results** | Figure - Test 03 [FAILED] - When making room    Figure - Test 03 [FAILED] - After Refreshing    Figure - Test 03 - Room made and user is admin in file and has permissions | | |
| **Test Date** | 26/03/2021 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 04 | **Description** | The moderator must be able to invite and remove users from a chatroom. |
| **Test Type** | Qualitative | **Success Criteria** | Moderator able to add user. Added user able to see and interact with the chatroom, when the removed however they are unable to see it |
| **Number of attempts** | 2 | **Comments** | This test may use white box techniques prior to the UI’s implementation. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Server functionality  Admin, Moderator, User classes | | |
| **Setup Instructions** | Enable MQTT server, run application with moderator user and standard user. Moderator adds user, user checks their chatroom access, moderator removes user, user checks their access again. | | |
| **Failure Correction Procedure** | Verify client and server source code for errors. Chatroom source code for permission errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 04 - Users after being removed from chatroom    Figure - Test 04 - Error message due to crash  Failure: Users added fine, but they can see the chatroom when they are removed from a chatroom until they select a new one. If a message is sent to the chatroom in the meantime, a bug causing them to crash occurs. | | |
| **Test Date** | 26/03/2021 | **Result** | Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 05 | **Description** | Moderators must inherit all the admin permissions; however, Moderators cannot demote the Admin. |
| **Test Type** | Qualitative | **Success Criteria** | Moderator able to add users, create new chatrooms and promote and demote users that are not Admin. |
| **Number of attempts** |  | **Comments** | This test may use white box techniques prior to the UI’s implementation. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Server functionality  Admin, Moderator, User classes | | |
| **Setup Instructions** | Enable MQTT server, run application as three users; Admin, Moderator and User. Test all basic admin functions as moderator, including adding and removing users and adding and removing channels. Attempt to remove admin. | | |
| **Failure Correction Procedure** | Verify client and server source code for errors. Chatroom source code for permission errors. | | |
| **Engineer(s) / Technicians** |  | | |
| **Individual Results** |  | | |
| **Test Date** |  | **Result** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 06 | **Description** | Application must provide a friendly User Interface (UI) |
| **Test Type** | Quantitative | **Success Criteria** | User response is at least 60% positive based on numerical feedback. |
| **Number of attempts** | 11 | **Comments** | Since user acceptance is challenging to evaluate, a judgement call will be made. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Test Users  User Feedback form  UI functionality | | |
| **Setup Instructions** | Begin application and give to user to evaluate.  Have at least one test user evaluated the interface based on usability provide feedback. | | |
| **Failure Correction Procedure** | Using collected user feedback as a guide, re-evaluate UI design and make usability improvements, recollect feedback. | | |
| **Engineer(s) / Technicians** | Hassan Naveed | | |
| **Individual Results** | Feedback Form: <https://forms.gle/zKvRikNPT3wieRED7>    Figure - Test 06 - User Feedback on clarity of UI    Figure - Test 06 - User Feedback on ease of user of GUI  Average = 8.7    Figure - Test 06 - User Feedback on Aesthetics of GUI  Average = 6.5    Figure - Test 06 - User Feedback on enjoyment of GUII  Average = 7.9    Figure - Test 06 - User Feedback whether they would recommend the app.  Average = 6.3  Overall average of 73.5% positive | | |
| **Test Date** | 27/03/2021 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 07 | **Description** | Users must be able to see the active users in the chat room. |
| **Test Type** | Qualitative | **Success Criteria** | All active users are visible through the chatroom interface. |
| **Number of attempts** | 2 | **Comments** | Passed but requires user to select their status first to then show whether they are online or not and can be inconsistent across users. Passable but not ideal. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  UI functionality  Server functionality  Chatroom functionality | | |
| **Setup Instructions** | Begin application, create a chatroom, add multiple users, have some be online and some offline. Check the active users updates when users go online and offline. | | |
| **Failure Correction Procedure** | Verify client and server source code for errors. Verify UI for errors and chatroom source code for errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 07 - users online shown to be online across all users, users offline shown to be offline across all users | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 08 | **Description** | Users must be notified when a new notification is received. |
| **Test Type** | Qualitative | **Success Criteria** | All notification worthy events trigger a notification. |
| **Number of attempts** | 1 | **Comments** | Notification worthy events include direct messages, chatroom messages or being added to a chatroom; Notification received when a new message is received. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  UI functionality  Server functionality  Chatroom functionality  Notification system | | |
| **Setup Instructions** | Set up two clients, one to send messages and make a chatroom, and one to receive the notifications when the events occur. | | |
| **Failure Correction Procedure** | Verify notification method for errors, verify that all event that trigger a notification are in place. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 08 - User receiving notification pop-up when message received | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 09 | **Description** | Clients must not connect directly to other clients without a server or a broker. |
| **Test Type** | Qualitative | **Success Criteria** | A connection error message is displayed when a message is attempted to be sent while disconnected. |
| **Number of attempts** | 1 | **Comments** | If clients can connect directly this would pose a security risk. Uses white box techniques. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  Server functionality | | |
| **Setup Instructions** | Manually disable MQTT server broker and attempt to connect client. | | |
| **Failure Correction Procedure** | Check for bugs and implement error handling procedure for client/server connection. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 09 - Application Settings prior to connecting to server.    Figure - Error message shown when trying to access chatroom while disconnected.    Figure - Test 09 -Application Setting after connecting to server | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 10 | **Description** | A server or a broker must allow multiple authorised clients to connect to it |
| **Test Type** | Qualitative | **Success Criteria** | Multiple clients can successfully connect to the server simultaneously |
| **Number of attempts** | 1 | **Comments** | May use white box techniques.  Multiple clients successfully connected. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  Server functionality | | |
| **Setup Instructions** | Enable MQTT server and attempt to connect at least 3 clients. | | |
| **Failure Correction Procedure** | Verify client and server source code for errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 10 - Multiple clients connected to the server at once | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 11 | **Description** | Users must only access their space after the login |
| **Test Type** | Qualitative | **Success Criteria** | User is only able to access their space after login |
| **Number of attempts** | 1 | **Comments** | Only able to sign in when both username and password are correct (note: only correct username/incorrect password combination displayed an error message, but only correct username and password allowed login) |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Login Procedure | | |
| **Setup Instructions** | Start application and ensure that a test account with username and password is available. Test that incorrect username/password combo, incorrect username / correct password, correct username / correct password combinations don’t work, before using correct username/password. | | |
| **Failure Correction Procedure** | Verify login source code for errors. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 11 - Incorrect username + incorrect password – Unable to log in    Figure - Test 11 - Correct username + incorrect password - unable to log in + error    Figure - Test 11 - Incorrect name + correct password - unable to log in    Figure - Test 11 - Correct username + password - able to log in | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 12 | **Description** | Passwords must be saved securely locally. |
| **Test Type** | Qualitative | **Success Criteria** | Passwords are inaccessible or otherwise not easily hackable. |
| **Number of attempts** |  | **Comments** | This might requires some white box techniques to check everything is secure |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Test User  Login System | | |
| **Setup Instructions** | Attempt to hack application by trying to decrypt passwords using rainbow table. | | |
| **Failure Correction Procedure** | Verify source code for errors, implement additional error handling and validation procedures. | | |
| **Engineer(s) / Technicians** | Hassaan Naveed | | |
| **Individual Results** | Figure - Test 12 - User Credentials File    Figure - Test 12 - Hash checking  User credentials text file has encrypted passwords, so they are secure. | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 14 | **Description** | Application must list all the personal contacts in the contacts pane. |
| **Test Type** | Qualitative | **Success Criteria** | Contact’s pane shows all user contacts. |
| **Number of attempts** | 1 | **Comments** | Requires classes and UI implementation |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Contacts System  UI Functionality | | |
| **Setup Instructions** | Set up a user with a few dummy contacts. | | |
| **Failure Correction Procedure** | Verify source code and UI for errors, implement additional error handling. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 14 - Contacts Dropdown/Pane showing all users except current | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

# Advanced Functionalities

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 16 | **Description** | The moderator should be able to create and delete channels in the chatroom |
| **Test Type** | Qualitative | **Success Criteria** | Moderator can add and remove channels. |
| **Number of attempts** |  | **Comments** | Should also test this ability is available to Admin.  Failed [Resolved] – Room config file failed to update on deletion, has since been resolved. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Chatroom functionality with channels  Moderator/admin/user classes | | |
| **Setup Instructions** | Set up a chatroom and use a moderator account, add a channel to chatroom. Check the channel is viewable to other users and messages can be sent received. As moderator, remove channel. Check this is updated for users. | | |
| **Failure Correction Procedure** | Verify chatroom and moderator source code for errors, implement additional error handling. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Adding Channel:    Figure - Test 16 - Adding Channel    Figure - Test 16 - New Channel available in channels dropdown    Figure - Test 16 - New channel added to room config file.  After Deleting Channel (Failed):    Figure - Test 16 [Failed] - Added Channel still in file after deletion.  After Deleting Channel (Resolved):    Figure - Test 16 - Channel removed from dropdown and file after deletion | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 18 | **Description** | Users should be able to change their status |
| **Test Type** | Qualitative | **Success Criteria** | User can change status and their status is visible as the same for other users. |
| **Number of attempts** | 1 | **Comments** | How status is displayed is UI dependant and may change.  Partial Passed – Requires user to select their status first to then show whether they are online or not and can be inconsistent across users. Passable but not ideal. |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  UI functionality  Status settings  Contact’s functionality | | |
| **Setup Instructions** | Open two users, who have each other as contacts. Have one change their status, check the change is visible to the other. | | |
| **Failure Correction Procedure** | Verify source code for errors, implement additional error handling. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 18 - User1 status set to Online.    Figure - Test 18 - User1 in online panel    Figure - Test 18 - User1 status set to offline    Figure - Test 18 - User1 in offline panel | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 19 | **Description** | Messages should be sent and received within 5-10 seconds |
| **Test Type** | Quantitative | **Success Criteria** | Time between messages being sent and received is <10 seconds, on average |
| **Number of attempts** | 10 | **Comments** | May use some white box techniques to get the most accurate time |
| **List of equipment / requirements** | Computer running Ubuntu Image  MQTT  QT Creator  Timer  Message Sending | | |
| **Setup Instructions** | Set up two users with each other as contacts, one to send and one to receive. Send a message and time how long it takes to receive a response. | | |
| **Failure Correction Procedure** | Verify client and server source code and message sending procedure for efficiency issues. | | |
| **Engineer(s) / Technicians** | Jarad Johnson-Bailey | | |
| **Individual Results** | Figure - Test 19 - Set-up prior to sending message    Figure - Test 19 - Result with timer after sending/receiving message    Figure - Test 19 - Results Table and Average time | | |
| **Test Date** | 27/03/21 | **Result** | Passed |

# Results Analysis

While most tests passed, some errors were raised in the testing process. Two of the tests that failed; the issue of the admin file failing to update for test 03 and the issue with the room config file failing to update on the deletion of channels, while being able to be patched, show a clear issue with the file system’s interaction with the application. There are often issues with the files, these were just highlighted during the testing process. It is recommended that the interaction between the files and system be refactored to hopefully uncover and remove any bugs that have may have been missed moving forwards. There is also a standing bug seen in test 04 that caused a crash in very specific circumstances and needs more time to be investigated further before any message saving functionality may be developed. Finally, the partial passes for tests 07 and 18 are for the same issue of the online/offline users’ system being buggy. It is currently at a state of being passable but needs serious reworking to make it consistent. A potential solution is a timer to update the online status of users, which may resolve the issues, and could be implemented alongside the automatic sign out requirement. Aside from errors, Test 06’s survey responses may be used later in development to help provide some guidance on how to best improve the application, and it should be noted that specifically the feedback on the aesthetics of the application was relatively low.