

# Test Plan Document

## 1. Introduction

The introduction provides context for the project.

- Identify the software product(s) to be produced by name.
- Fleet Destroyer (a battleship game online)
- Explain what the software product(s) will, and, if necessary, will not do
- This product will facilitate 2 player battleship, hold accounts passwords, maintains leader boards, allows for in game chatting, and provides an appealing GUI with background music and highly interactive button employment.

## 2. Quality Control

### 2.1 Test Plan Quality

How are you going to ensure the quality of the testing process? What is going to drive the selection of your test cases? How do you know whether you have a good test plan?

- Our tests will encompass the entire project through the evaluation of nearly every process testable. We aim to show that the main menu is working through testing the help and leader board. We will also main function of the game board which is to set the ship.

### 2.2 Adequacy Criterion

Define the termination criteria for your testing activity. In other words, define how to decide when testing is done. Typically, this involves some form of branch, statement, or path coverage.

- Testing is determined to be done when all main methods can compile alone as well as work together to accomplish a goal, both offline and online.

## 3. Test Strategy

Every major function of this game from sending a message to sending a hit to logging in is tested to ensure everything works together. The goal of us testing is to see how effective our implementation is and where flaws are (if any).

## 4. Test Cases

Provide a table of test cases, one per row. For each test case, the table should provide its purpose, the steps necessary to perform the test, the expected result, the actual result, and any additional information that you think

is relevant. **Note that all groups must implement their test cases using Travis, so that they are invoked on each git commit. You may use unit testing frameworks specific to your language or write your own drivers and stubs.**

Test Case ID	Purpose	Steps	Expected Results
<b><u>GAME BOARD INITIALIZATION</u></b>			
1	Game board ship coordinates compilation	Click on start Game and set up user ships  Wait for ship locations to print out	Tester will generate game board and will wait until user has picked all ships  Returns final coordinates
2	Game Board Initialization	Click start game from main menu and wait for result	Tester will assume game board set and attempt to compile successfully
<b><u>LEADERBOARD FUNCTIONALITY</u></b>			
3	LeaderBoard functionality	Click on LeaderBoard in main menu	Leaderboard will read from txt file Leaderboard.txt and display on leaderboard GUI
4	LearderBoard functionality	Invoke return LeaderBoard on command line	Prints Leaderboard onto command line
<b><u>CHAT IMPLEMENTATION</u></b>			
5	Chat functionality	Overwrite the get username call with stock name  Type message	Stock name should appear with chat message
6	Chat functionality	Append message directly to chat box instead of through client	The chat will compile and messages will only appear on the users window
<b><u>DATABASE IMPLEMENTATION</u></b>			
8	Testing new Sign up	Invoke new user form with new user name	The new user name should be added and

			printed into the database file
9	Testing new sign up with old user name	Invoke new user form with a taken username	Player database should reject the request
10	Verify that database was updated	Invoke the new playersToString method	The new username should be added at the bottom of the list and the taken username should not be duplicated
<b><u>INITIAL SCREEN IMPLEMENTATION</u></b>			
11	Verify initial screen gui loads with proper media	Initial screen	Login in and sign should appear with ocean gui in background
12	See if the login screen can be called from the initial screen	Invoke login	Login screens should appear
13	See if sign up screen can be called from initial screen	Invoke sign up	Sign up screen should appear
<b><u>SIGN UP FUNCTIONALITY</u></b>			
14	Verify that sign up method works	Attempt to sign up with username and pass	User successfully created, Prints new List
15	Verify that you cannot take someone else's credentials	Attempt to sign up with already taken username and pass	Added = false; Prints list with no new update
<b><u>HOME SCREEN IMPLEMENTATION</u></b>			
16	Corroborating that you can invoke the start game option from the home screen	Invoke game start event from the test method	Gameboard should initialize
17	Show that invoke leader board works	Invoke leaderboard event from the test method	Leaderboard should initialize

18	Show that invoke chat works	Invoke chat event from test method	Chat should initialize
19	Show the invoke options	Invoke options event from test method	Options should initialize
<u>HELP OPTIONS FUNCTIONALITY</u>			
20	Load the Help Option Menu	Invoke the main method from tester	The load GUI should open along with displaying instructions
<u>LOG IN FUNCTIONALITY</u>			
21	Load LogIn Menu	Invoke the main method of LogIn from tester	The LogIn menu should load with Graphics+Music
<u>CHATCLIENT-MAINCHATGUI IMPLEMENTATION</u>			
22	ChatClient message transaction	Invoke the GUI chat main method and then overwrite the ChatClient display message	MainChatGUI should show text recieved from ChatClient.

When testing our individual classes, we attempted to find any light mistakes that could throw our code off, as well as deeper ones. We ensured that we covered multiple required bases to deliver a level of functionality that is both successful and appealing. Our test cases compiled in 151.7 seconds using JUnit on Eclipse, with 0 errors.