**Software Requirements Specification**

**For**

**CS4398 Online Multiplayer Game**

**Version 1.0**

Prepared by Group 4398\_SM\_5

Raul Zuniga

Dayana Joseph

Denise S. Gan

Firdaus Botirzoda

Jie Zhou

2/14/2016

**Table of Contents**

[Revision History 4](#_Toc256000002)

[1. Introduction 5](#_Toc256000003)

[1.1 Purpose 5](#_Toc256000004)

[1.2 Product Scope 5](#_Toc256000005)

[1.3 Document Conventions 6](#_Toc256000006)

[1.4 Intended Audience and Reading Suggestions (Roy Working) 6](#_Toc256000007)

[1.5 References 6](#_Toc256000008)

[1.6 Product Perspective [Roy working] 7](#_Toc256000009)

[1.7 Product Functions 7](#_Toc256000010)

[1.8 Operating Environment 7](#_Toc256000011)

[1.9 Design and Implementation Constraints 7](#_Toc256000012)

[1.10 Assumptions and Dependencies 7](#_Toc256000013)

[2. External Interface Requirements 8](#_Toc256000014)

[2.1 User Interfaces 8](#_Toc256000015)

[2.2 Communications Interfaces/ 8](#_Toc256000016)

[3. Functional Requirements 8](#_Toc256000017)

[3.1 Use Case 1 8](#_Toc256000018)

[3.1.1 Selection Screen 8](#_Toc256000019)

[3.1.1.1 The selection screen shall have a button displayed to create a room. 8](#_Toc256000020)

[3.1.1 Creating 1 player game 9](#_Toc256000021)

[3.1.1.1 There shall be a button displayed to create a room. 9](#_Toc256000022)

[3.2 Use Case 2 9](#_Toc256000023)

[3.2.2 Creating 2 player game 9](#_Toc256000024)

[3.3 Use Case 3 9](#_Toc256000025)

[3.4 Use Case 4 10](#_Toc256000026)

[3.5 Use Case 5 11](#_Toc256000027)

[3.6 Use Case 6 [DENISE IS WORKING ON THIS] 11](#_Toc256000028)

[3.7 Use Case 7 11](#_Toc256000029)

[3.8 Use Case 8 11](#_Toc256000030)

[3.9 Use Case 9 12](#_Toc256000031)

[3.10 Use Case 10 12](#_Toc256000032)

[3.11 Use Case 11 12](#_Toc256000033)

[4. Nonfunctional Requirements 12](#_Toc256000034)

[4.1 Performance requirements 12](#_Toc256000035)

[4.2 Security Requirements 12](#_Toc256000036)

[4.3 Software Quality Attributes 12](#_Toc256000037)

[4.4 Business Rules 12](#_Toc256000038)

[5. Traceability Matrix 13](#_Toc256000039)

[6. Appendix A: Use Cases 18](#_Toc256000040)

[7. Appendix B: Glossary 18](#_Toc256000041)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Raul Zuniga | 2/14/2016 | Initial version. | 1.0 |
| Firdaus Botirzoda | 2/15/2016 | Completed section 5.6 | 1.1 |
| Raul Zuniga | 2/15/2016 | Added Section 1.1 | 1.2 |
| Firdaus and Jie | 2/15//2016 | Added section to section 3 | 1.3 |
| Denise Gan | 2/15/2016 | Added sections 3.3-3.5 to section 3. | 1.4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The reason for this document is to present a detailed description of a new online first-person shooter multiplayer game. It will explain the purpose and features of the system, its interfaces, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for the stakeholders, which consists of Dr. Rodion Podorozhny, the project manager and software engineers who will develop the system.

## Product Scope

The system shall provide a fun online multiplayer first-person shooter experience for one or two players. The game will be free to play. The game will be provided by a host server which will contain all the project files, a game engine, and game controller using a real-time connection. Players will play in a map in which there are enemies roaming around that can kill the player. The players will also have the ability to shoot each other and cause health damage. The game will end for the player when the exit for the map is found. The player will be able to play multiple levels. As the player progresses through the levels of the game the difficulty of the game will get progressively harder. The system will not save the state of the game when the player quits the game. When the user plays the game again, the player will start at the beginning of the first level.

## Document Conventions

Every detailed requirement statement will have its own priority. This priority will not be inherited by higher-level requirements.

## Intended Audience and Reading Suggestions (Roy Working)

The audience for this SRS is Dr. Rodion Podoronzhny, the project manager, software and test engineers of the system. It is written in natural English language so a broad audience

of stakeholders can understand it. Section 1 of the SRS is an introduction to the overall system.

Section 2 describes the external interfaces of the system. Section 3 gives a description of the functional requirements of the system. Section 4 gives a description of the informal requirements of the system. Section 5 is the traceability matrix. Section 6 and 7 are the Appendixes.

## References

## Product Perspective

The software product will be interface with the game's graphical user interface where users can operate can operate all the provided functionality. A domain and host provider will host the website where all the files will be stored. A script on the server will act as a controller of the flow of data between the client(s) and the server. A player will be able to join and quit the game whenever they want. Up to two players will be able to join a game session at a time.

## Product Functions

## Operating Environment

## Design and Implementation Constraints

## Assumptions and Dependencies

There are no assumptions and dependencies.

# External Interface Requirements

## User Interfaces

The game shall work and be tested on the following browsers: Internet Explorer, Firefox, Google Chrome, Apple Safari, and Opera. The game shall work and be tested on different devices such as a desktop, tablet with 10-inch screen, and different mobile phones.

## Communications Interfaces/

Transfer Protocol (HTTP), which is designed to enable communications between clients and servers. The method used for request-response between client and server will be POST. This method submits data to be processed to a specified server in a secure manner because data is not displayed in the URL. The controller that we use will be Node.js, which will be connected to a game engine Impact.js. Node will be transferring information between the browser back and forth in real-time using socket.io.

# Functional Requirements

## Use Case 1

**Title:** Create a 1P game

**Actor:** Customer

Scenario:

1. Create a game session (room) in a room selection screen.
2. The system will generate the name of the room or session ID.
3. Character selection screen appears
4. User selects 1P
5. User selects character
6. User can input character name or use default name
7. User tells the system that he is ready to start the game.
8. Game session starts

## User Screen

User Name

Figure 1 User screen

### A User screen shall be the first screen displayed in the browser when the user types in the game URL and presses enter.

### In the character select screen there shall be an input box for a username.

### In the User screen the input box for username will have system created username by default.

### The User screen shall have a selector to create a room for a one-player game.

### The User screen shall have a selector to join a room for a two-player game.

### Creating a room takes the player to the Selection screen

## Creating a Room takes the player to the Selection screen

### In the User screen the player starts a one-player game by selecting the selector for creating a room.

### After the selector for creating a room is selected the Selection screen appears.

## Joining a Room takes the player to the Selection screen

### In the User screen two players can start a two-player game by selecting the selector to join a room.

### After the selector for creating a room is selected the Selection screen appears.

## Selection Screen

1P 2P

Character Image

Character Image

Character Character

name name

Figure 2 Selection Screen

### The Selection screen shall have a character select screen for each player

### The character select screen shall have an image of each character that can be selectedby using an arrow button

### In the character select screen the image of the character will appear in color if there exists another player.

### In the character select screen the image of the character will appear in black and white if there doesn’t exist another player.

### In the Selection screen there shall be a button called “ready” to start the game

### In the Selection screen there shall be a “ready” button that will light up green after it’s clicked

## Starting 1 player game

### In the User screen the player starts a one-player game by selecting the selector for creating a room.

### In the Selection screen the user press the ‘ready’ button.

## Use Case 2

**Title:** Create a 2P game

**Actor:** Customer

1. Create a game session (room) in a room selection screen.
2. The system will generate the name of the room or session ID.
3. Character selection screen appears
4. Users selects 2P
5. 1st player selects character
6. 2nd player selects character
7. Users can input character name or use default name
8. Users tell the system that they are ready to start the game.
9. Game session starts

## Creating 2 player game

### The weapons and all the rules for creating a one player game apply to a two player game.

### The number of enemies will double.

## Use Case 3

**Title:** Create game session

**Actor:** Player

1. The game must load the map
2. The game must load all sprite classes
3. The game must load dependent game libraries (2D, sound, etc)
4. The user should see player character(s) appear to start the game.

## Create a game session

### The game loads the first level map.

### Dependent game libraries loaded (i.e. sound).

### The main game screen will display the current level of the game.

### The main game screen will display health indicator of the player.

### The player character(s) appear at the start of the map with default weapons

### (pistol).

### Player(s) start with full bars of health. The health indicator will display three

### hearts, meaning the player has three lives.

### Sprites are loaded and await to be triggered by player movement.

## Use Case 4

**Title:** Killing an enemy

Prerequisite: game has started

**Actor:** Player

1. The default enemy can be killed by one bullet.
2. The default enemy can only run.
3. The shooting enemy can be killed by one bullet.
4. The shooting enemy can also shoot bullets.

## Killing an enemy

### Enemies can be killed by one bullet.

### Killed enemies will drop their weapon.

### The character will have the ability to pickup the dropped weapon.

### The list of weapons include: shotgun, assault rifle, laser gun.

### pistol – can kill one enemy per bullet and only travel 1/2 of the screen, one shot per

### 1.5 second.

### The shotgun – can kill five enemies per shot, but can only travel 1/3 of the screen

### distance. One shot per second is the limit.

### The assault rifle can travel 2/3 of the screen, can kill one enemy per bullet, 4 shots per

### second is the firing rate.

### The laser – can travel the whole screen and can kill all of the enemies in the screen that

### line up with the laser.

## Use Case 5

**Title:** Killing a player

Prerequisite: game has started

**Actor:** Player

1. A player can be killed by one bullet
2. A player can be killed by an enemy touching him.
3. A player can be killed by falling into a pit.

### Killing a player

3.5.1.1 The enemy can kill the character with one shot.

3.5.1.2 Direct contact with an enemy can kill a player.

3.5.1.3 Falling into hazards (i.e. pits) can kill a character.

## Use Case 6 [DENISE IS WORKING ON THIS]

**Title:** Player actions

**Actor:** Player

1. Player can run
2. Player can shoot
3. Player can jump (onto objects)
4. Player can pick up a dropped weapon
5. Player can duck

### The player can run (set speed).

### The player can jump (set distance).

### The player can crouch with the option to shoot.

### The player can shoot and will have unlimited ammo.

### The player can pick up dropped weapons but have no more than one in

### possession at any time.

## Use Case 7

**Title:** Enemy actions

**Actor:** Enemy

1. Enemy can walk
2. Enemy can run
3. Enemy can shoot
4. Enemy can jump (onto objects)

## Use Case 8

**Title:** Player kills another player

**Actor:** Player

1. Player can shoot at other player
2. Player can be killed by three bullets from other player
3. A health bar will be displayed for each player during the game.
4. Health bar will decrease when shot.

## Use Case 9

**Title:** Player completes mission

**Actor:** Player

1. Player reaches the end of the map.
2. The end of the map is designated by a door or portal that he walks through.

## Use Case 10

**Title:** User action

**Actor:** User

1. User can move using a touch screen on a tablet
2. User can move using a keyboard on a personal computer.

## Use Case 11

**Title:** User can pause a game

**Actor:** User

1. User can pause a game for an unlimited time
2. User can then restart the game

# Nonfunctional Requirements

## Performance requirements

The product will be web-based and will require powerful server machines with high band Internet access so it can handle multiple users at the same time. Expected number of simultaneous users shall be at least two players’ dependent on how connections are established. This game will be online and multi-player. The HTML5 web-based game requires users to have an up-to-date browser that has HTML5 capability. The game should not require a lot of processing power.

## Security Requirements

There will not be a substantial amount of data going back and forth in this game. Therefore, the game should be secured by the browser.

## Software Quality Attributes

## Business Rules

### This game is built strictly for demonstration purposes. This game is not licensed for use in any commercial, non-profit, or revenue-generating business activities by its developers.

### The game may not be distributed through any commercial online app marketplace.

### The game may not be copied by anyone without authorization by its creators.

# Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical Requirement ID** | **Test Case ID** | **Functional Requirement Description** | **Priority** | **Type** |
| 4.1.1.1 | 1 |  |  | Interface |
| 4.1.1.2 | 2 |  |  | Functional |
| 4.1.1.3 | 3 |  |  | Functional |
| 4.1.1.4 | 4 |  |  | Functional |
| 4.1.1.5 | 5 |  |  | Functional |
| 4.1.1.6 | 6 |  |  | Functional |
| 4.1.1.7 | 7 |  |  | Functional |
| 4.1.1.8 | 8 |  |  | Functional |
| 4.1.1.9 | 9 |  |  | Functional |
| 4.1.2.1 | 10 |  |  | Interface |
| 4.1.2.2 | 11 |  |  | Functional |
| 4.1.2.3 | 12 |  |  | Functional |
| 4.1.2.4 | 13 |  |  | Interface |
| 4.1.2.5 | 14 |  |  | Functional |
| 4.1.2.6 | 15 |  |  |  |
| 4.1.3.1 | 16 |  |  | Functional |
| 4.1.3.2 | 17 |  |  | Functional |
| 4.2.1.1 | 18 |  |  | Interface |
| 4.2.1.2 | 19 |  |  | Interface |
| 4.2.1.3 | 20 |  |  | Interface |
| 4.3.1.1 | 21 |  |  | Interface |
| 4.3.1.2 | 22 |  |  | Interface |
| 4.3.1.3 | 23 |  |  | Functional |
| 4.4.1.1 | 24 |  |  | Interface |
| 4.4.1.2 | 25 |  |  | Interface |
| 4.4.1.3 | 26 |  |  | Interface |
| 4.4.1.4 | 27 |  |  | Interface |
| 4.5.1.1 | 28 |  |  | Interface |
| 4.5.1.2 | 29 |  |  | Functional |
| 4.5.1.3 | 30 |  |  | Functional |
| 4.6.1.1 | 31 |  |  | Functional |
| 4.6.1.2 | 31 |  |  | Interface |
| 4.6.1.3 | 32 |  |  | Functional |
| 4.6.1.4 | 33 |  |  | Interface |
| 4.6.1.5 | 34 |  |  | Functional |
| 4.6.1.6 | 35 |  |  | Interface |
| 4.6.1.7 | 36 |  |  | Functional |
| 4.6.1.8 | 37 |  |  | Interface |
| 4.6.2.1 | 38 |  |  | Functional |
| 4.6.2.2 | 39 |  |  | Functional |
| 4.6.2.3 | 40 |  |  | Functional |
| 4.6.2.4 | 41 |  |  | Functional |
| 4.6.2.5 | 42 |  |  | Functional |
| 4.6.2.6 | 43 |  |  | Functional |
| 4.6.2.7 | 44 |  |  | Functional |
| 4.6.2.8 | 45 |  |  | Functional |
| 4.6.3.1 | 46 |  |  | Interface |
| 4.6.4.1 | 47 |  |  | Interface |
| 4.6.5.1 | 48 |  |  | Interface |
| 4.6.6.1 | 49 |  |  | Interface |
| 4.7.1.1 | 50 |  |  | Functional |
| 4.7.1.2 | 51 |  |  | Functional |
| 4.7.1.3 | 52 |  |  | Functional |
| 4.7.1.4 | 53 |  |  | Functional |
| 4.7.1.5 | 54 |  |  | Functional |
| 4.7.1.6 | 55 |  |  | Functional |
| 4.7.1.7 | 56 |  |  | Functional |
| 4.7.1.8 | 57 |  |  | Functional |
| 4.7.1.9 | 58 |  |  | Functional |
| 4.7.1.10 | 59 |  |  | Functional |
| 4.7.1.11 | 60 |  |  | Functional |
| 4.7.1.12 | 61 |  |  | Functional |
| 4.7.1.13 | 62 |  |  | Functional |
| 4.7.1.14 | 63 |  |  | Functional |
| 4.7.1.15 | 64 |  |  | Functional |
| 4.7.1.16 | 65 |  |  | Functional |
| 4.7.1.17 | 66 |  |  | Functional |
| 4.7.1.18 | 67 |  |  | Functional |
| 4.7.1.19 | 68 |  |  | Functional |
| 4.7.1.20 | 69 |  |  | Functional |
| 4.7.1.21 | 70 |  |  | Functional |
| 4.7.1.22 | 71 |  |  | Functional |
| 4.7.1.23 | 72 |  |  | Functional |
| 4.7.1.24 | 73 |  |  | Functional |
| 4.7.1.25 | 74 |  |  | Functional |
| 4.7.1.26 | 75 |  |  | Functional |
| 4.7.1.27 | 76 |  |  | Functional |
| 4.7.1.28 | 77 |  |  | Functional |
| 4.7.1.29 | 78 |  |  | Functional |
| 4.7.1.30 | 79 |  |  | Functional |
| 4.7.1.31 | 80 |  |  | Functional |
| 4.7.1.32 | 81 |  |  | Functional |
| 4.7.1.33 | 82 |  |  | Functional |
| 4.8.1.1 | 83 |  |  | Functional |
| 4.8.1.2 | 84 |  |  | Interface |
| 4.8.1.3 | 85 |  |  | Interface |
| 4.8.1.4 | 86 |  |  | Interface |
| 4.8.1.5 | 87 |  |  | Interface |
| 4.8.1.6 | 88 |  |  | Interface |
| 4.9.1.1 | 89 |  |  | Interface |
| 4.9.1.2 | 90 |  |  | Interface |
| 4.9.2.1 | 91 |  |  | Interface |
| 4.9.2.3 | 92 |  |  | Interface |
| 4.9.3.1 | 93 |  |  | Interface |
| 4.9.3.2 | 94 |  |  | Interface |
| 4.9.3.3 | 95 |  |  | Interface |
| 4.9.4.1 | 96 |  |  | Interface |
| 4.9.4.2 | 97 |  |  | Interface |
| 4.10.1.1 | 98 |  |  | Functional |
| 4.10.1.2 | 99 |  |  | Functional |
| 4.10.1.3 | 100 |  |  | Functional |

# Appendix A: Use Cases

# Appendix B: Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Field | A cell within a form. |
| Member | A member of the Exe Club listed in the Exe database. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |