Software Requirements Specification

for



(Unity 3D Game)

Version 1.1v

Prepared by

Group Name: Eliters

|  |  |  |
| --- | --- | --- |
| Rehan Mumtaz | SESE-19036 | [mumtaz4203271@cloud.neduet.edu.pk](mailto:mumtaz4203271@cloud.neduet.edu.pk) |
| Kabeer Ahmed | SESE-19028 | [ahmed4201750@cloud.neduet.edu.pk](mailto:ahmed4201750@cloud.neduet.edu.pk) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
|  |
| Instructor: Dr. Shehnila Zardarifv  *Date:* |
|  |

Contents

Role iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.1.1 SYSTEM INTERFACE……………………………………………………………………………………….

2.1.2 USER INTERFACE……………………………………………………………………………………….

2.1.3 HARDWARE INTERFACE……………………………………………………………………………….

2.1.4 SOFTWARE INTERFACE………………………………………………………………………………….

2.1.5 COMMUNITCATION INTERFACE………………………………………………………………….

2.1.6 MEMORY CONSTRAINTS………………………………………………………………………….

2.1.7 OPERATIONS……………………………………………………………………………………….

2.1.8 SITE ADAPTATION REQUIREMENTS………………………………………………………….

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 CONSTRAINTS 3

2.5 Assumptions and Dependencies 4

2.6 APPORITONING OF REQUIREMENTS 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

Appendix A – Data Dictionary 9

Role

**Rehan Mumtaz (SE-036)** and **Kabeer Ahmed (SE-028)** gathered all the requirements for their project “Car GO” which is a 3D game. To gather the requirements, they observed the gaming trends in the young generation like what type of games gain popularity and what should be the difficulty level of the game and they gathered some beautiful background sceneries to improve the UI and try their best effort to give users a best experience. Rehan Mumtaz presented the idea to associate a theme to the game and implemented the background scenery and gave player opportunity to select different locations across the globe as well as the particle effects for boosters and coins in the game. Kabeer Ahmed suggested and implemented the logic for the selection of different cars for the player ,i.e; each car have different specs based on the collection player has bought it, and also the musical theme of the game. Both the members worked together on the animations of the game. Rehan Mumtaz proposed and organized the performance requirements, external interfaces and design constraints. Kabeer Ahmed proposed the scope and perspective of the product along with all the kinds of interfaces, assumptions and dependencies of the system and the software system quality attributes. Rehan Mumtaz made the class diagram to describe the logical structure of the data while Kabeer Ahmed made the block diagram and the use-case diagram.

# Introduction

*<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>*

## Document Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.

TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>

## Intended Audience and Document Overview

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

## Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

## References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

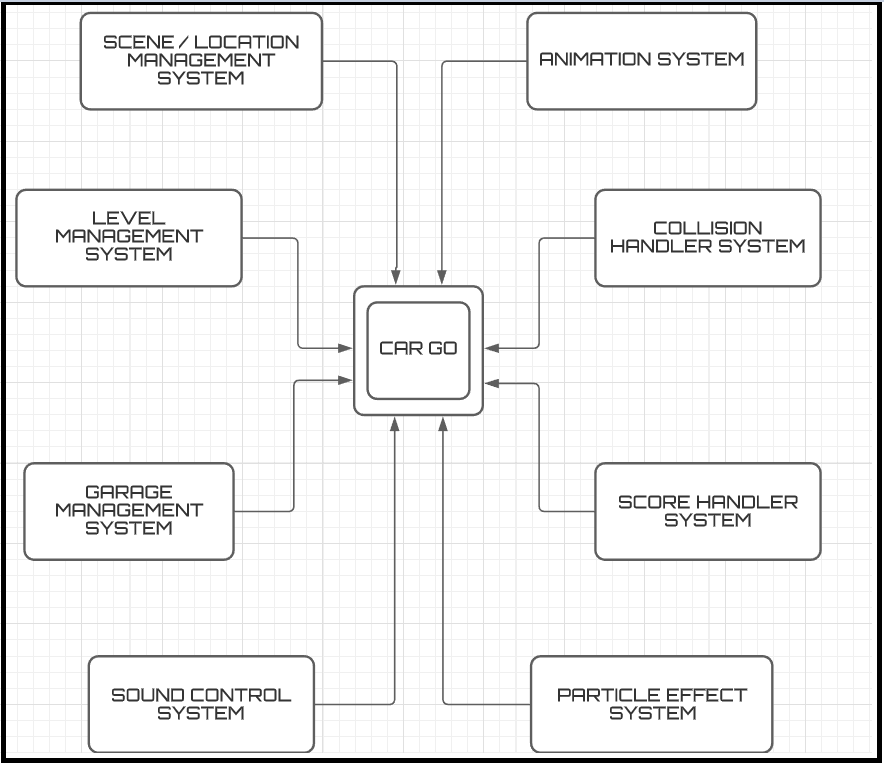
TO DO: Use the standard IEEE citation guide for this section>

# Overall Description

## This section will give information about product perspective, product functions, constraints, assumptions and dependencies, Apportioning of Requirements.

## 2.1. Product Perspective

Car GO-3D gaming application is totally independent system that is not related to any other system and not a component of a larger system. This program has only one type user, so there is no functionality differences between users which means there exists only one type user interface. Run Boy Run-3D will run on windows operating system. Moreover, it will be implemented making use of Unity3D for visualizing its features. The game seeks advantage of some built-in libraries of unity engine to include some functionalities like the scene management system and the particle effect system inside the unity engine. The game logic also uses the animation system of the unity engine to put into play all the animations of the game. The game logic interacts mainly with sub-systems and other libraries inside the unity engine but no external dependencies are on the game.

 ***Figure 1- Block Diagram***

## The diagram above shows how the main system interacts with the other system and the context of the system

## 2.1.1 System Interface

One system requirement is the required operating system i.e: Windows 7,8.1, 10 are compatible with the game , however, 64 bit is more preferred (32-bit is also workable ) The user must have a mouse and a keyboard for great user Experience other wise game wont be operable

## 2.1.2 User Interface

There will be one type of user. Therefore, there are no differences between users in terms of functionality, visualization and interface

Behind the user’s car there is a monster car which will be chasing throughout the game, the user wanna escape and should have to look carefully at the path he is passing through.

At the beginning of the program there will be 5 options;

* START - to enter the game
* SELECT Car – To select the car to play the game with
* SELECT location – To select the city user wants to play at
* HELP - to read instructions / controls on how to play
* CREDITS - to read about the team Eliters and how they developed the game
* EXIT - to quit the game

After Pressing START, the user is required to press “Spacebar” so the car actually starts to move forward. When the character starts to move forward then there are 3 cases:

* When “up arrow key” or “W” button is activated, the car JUMPS upward.
* When “rightward key” or “D” button is activated, the car moves in RIGHT direction.
* When “leftward key” or “A” button is activated, the car moves in LEFT direction.

By moving with the help of these keys, the user can collect the coins scattered all over the path. Moreover, the user can also collect boosters in same manner as the coins

## 2.1.3 Hardware Interface

Only Personal computers, be it desktop or laptop, which have keyboard to use arrow keys or WAD keys will be suitable for the application. These devices should have some limit requirements to make the application run effectively. We expect 1.5 GHz processor, 2GB minimum internal memory & 4GB RAM minimum for computers and NVIDIA 560MX OR AMD RX 560 should be required due to the 3D high end graphics it is needed to run the game effectively without lag and obviously the pixels should not be broken. For external server if found appropriate Parse servers will be used.

## 2.1.4 Software Interface

Computers or Laptop will be used for the application and they must have Windows to run the application. There will be a graphical user interface for the user to select different characters and find instructions and credits of the game

## 2.1.5 Communication Interface

There won’t be any communication required, the game is a desktop application and will work even when offline.

## 2.1.6 Memory Constraints

It should be keep in mind that the Game engine which will be using is the Unity 3D , as it has lots of dependencies and packages came with it as it requires some amount of memory. For performance a RAM of 4GB is required and storage of 2GB is required

## 2.1.7 Operations

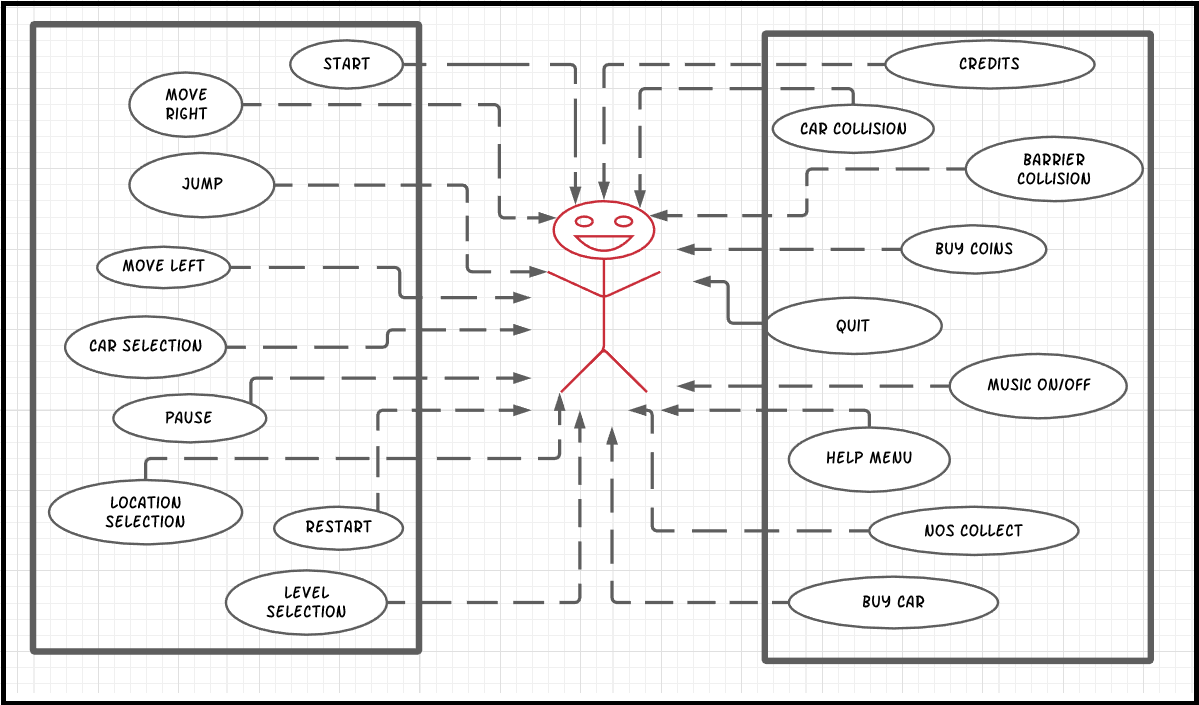
There are not so many operational constraints. Most of the operations are visible to user if he/she navigates through the graphical user interface. Regarding the operations, user can find help menus also. After the user and system interactions, some operations will take place at the background implemented through coding, such as the score and coin increments after player collects each coin or nitro boosters.Users can buy coins via the payment method implemented at the backend which can be used to buy new cool cars

## 2.1.8 Site Adaptation Requirements

There won’t be any site adaption requirements because the game is a desktop application and and in case the user wants to buy coins or purchasing then internet is required.

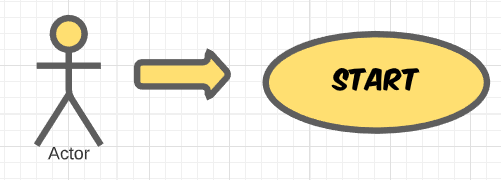
## Product Functions

Use case diagram of the Run Boy Run-3D application is revealed in Figure 2. Steps are gathered in distinct entities, the functions of which are stated in further subsections.



**Figure 2- Use Case Diagram**

#### Start

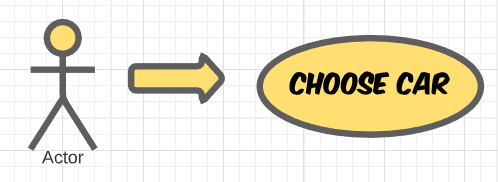


## 

*Brief Description:*

This functionality becomes active just after clicking on Run Boy Run-3D application icon or when the user enters the application. After starting user will be directed to a introductory screen and to proceed further, the user has to press CONTINUE button. Then the user will be asked to select one of the available options from the main menu. After which the user can start the game based on his/her choice of option

**2.2.1.2** **Choose Car**



*Brief Description:*

This functionality becomes activated when the user presses “CHOOSE CHARACTER” button on the Main menu. This function makes the user to choose between different characters to play with.

## 2.2.1.3 Help

*Brief Description:*

## This functionality becomes activated when the user presses “HELP” button on the Main menu. This function takes the user to HELP menu where the user can read instructions about.

## 2.2.1.4 Credits

*Brief Description:*

## This functionality becomes activated when the user presses “CREDITS” button on the Main menu. This function takes the user to CREDITS menu where the user can read about the developers of the game.

## 2.2.1.5 Quit

*Brief Description:*

## This functionality becomes activated when the user presses “QUIT” button on the Main menu. This function exits the gaming application

## 2.2.1.6 Resume

*Brief Description:*

This functionality becomes activated when the user clicks on “RESUME” button on the PAUSE menu. This function takes the user back to its original play. In other words it resumes the game.

## 2.2.1.7 Pause

*Brief Description:*

This functionality becomes activated when the user clicks on the “PAUSE” button on the screen while the character is running. This function pauses the game and displays PAUSE menu.

## 2.2.1.8 Restart

## 

*Brief Description:*

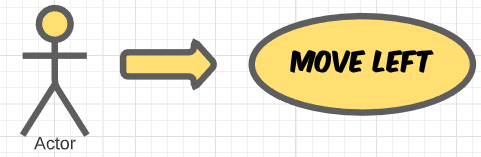
## This functionality becomes activated when the user clicks on “RESTART” button on the PAUSE Menu. This function restarts the level for the user.

## 2.2.1.9 Move Right

*Brief Description:*

This functionality becomes activated when the user press ‘rightward key’ or ‘D’ button on the keyboard. This function makes the user move right.

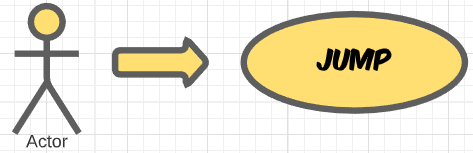
## 2.2.1.10 Move Left

**

*Brief Description:*

This functionality becomes activate when the user press “leftward key” or “A” button on the keyboard. This function makes the user move left.

## 2.2.1.11 Jump

**

*Brief Description:*

This functionality becomes activate when the user press “forward key” or “W” button on the keyboard. This function makes the user Jump upward.

## 2.2.1.12 Obstacle Collision

*Brief Description:*

## This functionality becomes activated when the car collides with an OBSTACLE ( Moving Cars,and Barriers). This function makes the game over for the user and the user dies thus the level restarts.

## 2.2.1.13 NOS Collect

*Brief Description:*

This functionality becomes activated when the character touches a Nitro BOTTLE. This function makes the Cars speed fast for the user and thus it increases score of the user.

## 2.2.1.14 Music On/off

*Brief Description:*

This functionality becomes activated when the user wants to SWITCH ON or OFF the music

## 2.2.1.15 Buy Coins

*Brief Description:*

This functionality becomes activated when the user wants to purchase the coin which the user can use to buy new cars and bought some cool specs for their cars

## 2.2.1.16 Buy Car

*Brief Description:*

This functionality becomes activated when the user wants to purchase the cars , user can buy any car of the budget he/she owns

## 2.2.1.17 Selection of car

*Brief Description:*

This functionality offers the user to select any car from the garage he/she owns .Moreover, user can apply differenct colours and change differenct specs of the cars

## 2.2.1.18 Location Selection

*Brief Description:*

This functionality offers the user to select different locations across the globe which are unlocked only. The user will have to exprrience each map in order to unlock different locations

## 2.2.1.19 Level Selection

*Brief Description:*

This functionality lets the user to change level diffulty in order to complete the game . It offers user three levels in each map i.e: EASY , MEDIUM & DIFFICULT

## Users Characteristics

The user should be familiar to using a computer and windows environment, and how to navigate to and startup an application. Since the language of the application is English, so, the user should be familiar to basic level of English language.

## Constraints

The gaming application requires a computer system (Desktop or Laptop) with decent specifications as described in section 2.1.3 i.e. Hardware Interfaces. The operating System on the computer must be windows

## Assumptions and Dependencies

Apart from the operating system, the game can also run on Unity 3D engine so it will be our dependency. The game runs offline so no servers are required. The users are assumed to be familiar with basic computer skills and English language to navigate to the game and also select and perform different operations inside the game.

**2.6 Apportioning of Requirements**

The additional requirements for future versions of this gaming application are;

* At least 10 different cars for the player to select.
* More than 10 different locations of the game with different themes
* Endless runner game mode selection option.

# Specific Requirements

## External Interface Requirements

### User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Those who will be able to provide optional Graphical User Interface screenshots, will be rewarded by extra marks.>

### Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

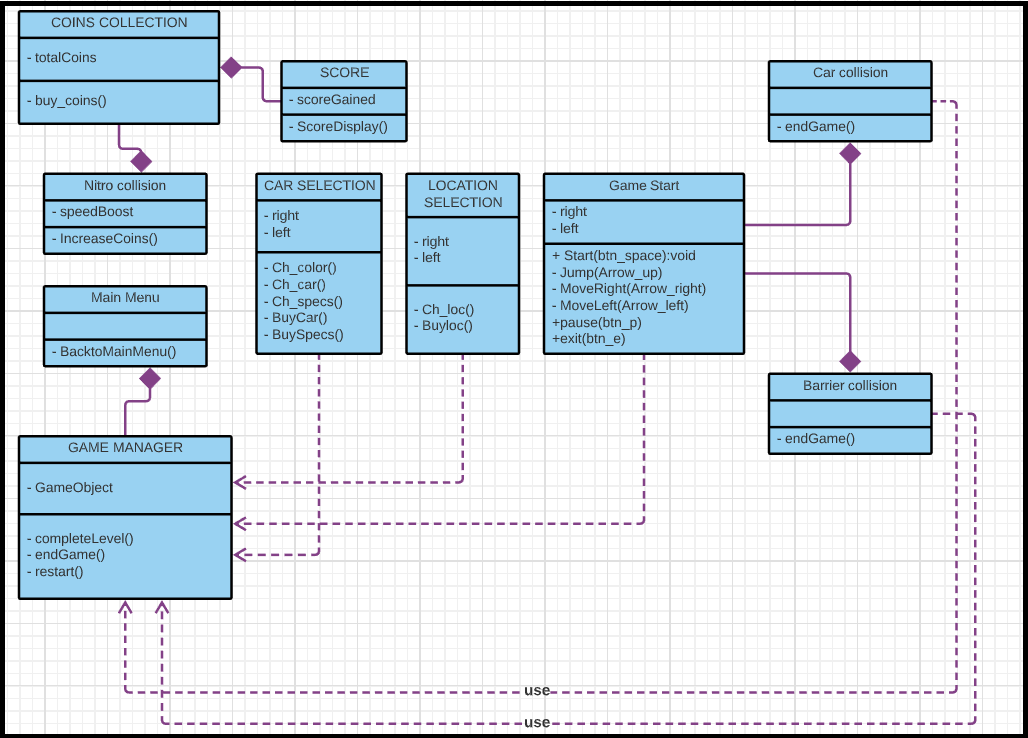
*TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

## Behaviour Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.

TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>



**Figure 3- Class Diagram**

# Other Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide at least 3 different safety requirements based on your interview with the client, and again you need to be creative here.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

# Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*