Software Requirements Specification

for

Survival VR(Virtual Reality game)

Version 1.0 approved

Author

Sameer Kazmi-SE-35

Muhammad Ali-SE-12

Humza Imaran-SE-38

Ahsan Khan-SE-21

Mahnoor Rahim-SE-02

Organization

<Sir Syed University of engineering and technology>

Created

<March-27-2017>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## 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.>

/\*

Our motivation is the boom of this VR technology in the current era

and how its storming the market leading us to a whole new world. Major game

engines such as Unity and Unreal are now supporting VR, and the computer

power required to drive VR is \_nally a\_ordable. Today we have some great

platforms to work with including Oculus Rift, Google Project Cardboard, and

Samsung Gear VR. .

\*/

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. 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.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

/\*

The scope to build such game is to learn new technology that is escalating

quickly into the world and step into the zone of creating games on VR which is

the trend in the 21st century. The product will be motivation for youth to work on such technologies which is going to be game changer in this era.

\*/

## References

<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. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

/\*

* VR with using VR properties
* Bluetooth mobile controller
* Playable on Android lollipop
* Playable on IOS 10.0

\*/

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

/\*

Characters

1. Main player:-

2. A father

3. Little ghost girl

1. Surviving test unit:-

2. Scientist

3. Doctor

4. Army man

Ghost

1. Children

2. Little ghost girl

3. Nurse

4. Security guard

5. Doctor

6. Mad patient

Object

1. Flash light

2. Helicopter

3. Lock

4. Screwdriver

5. Shot gun

6. Chain cutter

\*/

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

/\*

* SURVIVAL VR is a science fiction survival horror video game which will be available on Android and IPhone .OS.
* 3d game
* Camera setting
* High performance
* Real life like experience

\*/

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

/\*

* unity frameworks,
* Markup languages and Programming languages
* Photoshop/illustrator for UI/UX design
* 3D Maya for charater design
* Bluetooth mobile controller
* Cardboard
* Samsung gear VR,

\*/

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

/\*

Manuals will be provided to audience in order to overcome any problem while using the product and will provide Youtube videos for guidance and promotions

\*/

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# 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., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

/\*

Controls:-

Use y axis and x axis to move the player in to the direction it's looking.

Press A to set the direction.

Press B to interact to things.

Press C to turn ash light on/o\_

Press X to run/crouch (sit)..

Save and load game

Game will be auto saved and loading will be chapter vise and load the

saved game . . . .

\*/

## 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, and communication protocols to be used.>

/\*

Our product is supported by devices such as Cardboard and Samsung gear VR.

Run on : android lollipop/IOS 10.0 above

\*/

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, 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. Refer to documents that describe detailed application programming interface protocols. 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.>

/\*

Operating Systems: Windows 7, 8.1, 10

Integrated development Environment : Mono development

Markup Software : Photoshop. Maya3d.

Programming Languages : javascript, C-Sharp

Game Engine : Unity 5.5

\*/

## 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.>

/\*

The game product as its on youtube channel for help and guidance for using product and also for game promotion

* Input device : Bluetooth controller

\*/

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional 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.>

/\*

Performance will be tested on different stages of development and when finally development will be completed will be tested on different devices(Android, IOS) and different user’s experience

\*/

## Safety 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.>

/\*

All the possible security concerns will be make to ensures the quality of the desired product.

\*/

## Security Requirements

<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. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## 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.>

/\*

it will be checked on Android and IOS devices as to know the game is running.

1. Functionality Testing : Test all components in your game are working

correctly and make sure there no object is un readable. Test SCRIPT are

working on game. Test the controls as it runs

* Checking for Syntax Errors.
* Test user interface to ensure that all the user interface
* things are there.
* Readable color schema's.
* All the buttons are working correctly

2. Usability Testing :

Test the game menu :

* Menus, buttons or Links to di\_erent pages on your game should be
* easily visible and running.

3. Interface Testing : Three areas to be tested here are game scene user

interface game controller

* Game Scene : Test the game obj are all there.
* User Interface : Menu is working.
* Game Control : Is player controlling the game according to the
* script.

4. Performance Testing :

* The game should run on 60fps.

\*/

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

/\*

* Each member of group is assigned a task who will be responsible for his task

to be completed on given dates.

* Completion on every phase encourages our team mates and will make queries

and planning according to the situations

* Phase 1: basic idea on our game to be made once the basic game is made then

we move on to the next phase.

* Phase 2: After completing phase 1 we will focus on making our game beautiful
* using Photoshop and Maya3d and add few extra thing to make game more fun

to play

\*/

# Other Requirements

<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: Glossary

<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.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>