Software Requirements Specification

for

Life Tree

Version <1.0>

Prepared by

Group Name: D2D

|  |  |  |
| --- | --- | --- |
| Chai Min Chun | 35657 | 35657@siswa.unimas.my |
| Choo Ting Wei | 35769 | 35769@siswa.unimas.my |
| Lai Ted Liang | 36579 | 36579@siswa.unimas.my |
| Lim Ban Yong | 36709 | 36709@siswa.unimas.my |
| Willis Fung Shin Choi | 39324 | 39324@siswa.unimas.my |

|  |  |
| --- | --- |
| Instructor: | Madam Nurfauza Binti Jali |
| Course: | TMP 3413 Software Engineering Laboratory |
| Lab Section: | Teaching Lab 1 |
| Teaching Assistant: | Madam Nurfauza Binti Jali |

|  |  |
| --- | --- |
| Date: | <place the date of submission here> |

Contents

Revisions ii

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

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 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 6

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 7

Appendix A – Data Dictionary 8

Appendix B - Group Log 9

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Choo Ting Wei | Complete the requirement for the Life Tree Game project. | 00/00/00 |

# Introduction

This document provide a complete details of Software Requirement Specification (SRS) document for an Android game which name as Life Tree. This is an educational game that increase the awareness of pollution through interesting gameplay. In the following section of the SRS, we specify the purpose of this documents, project scope, intended audience, and included all sources that used to complete this document.

## Document Purpose

The purpose of this document is to give a detailed description of functional and non-functional requirement and functional requirement associated with Life Tree (Version). Life Tree (Version) SRS will act as the mainstay for any other document to be developed for this project in the future. Furthermore, it will serve as the guideline for the future software verification and testing by stating the behaviour requirement of the system in the form of a Use Case Diagram and Sequence Diagram.

## Product Scope

Life Tree shall be an Android based game which will allow user to play suing Android-powered smartphone / tablets version 4.0 and above. The game optimized for screen resolution of 800 x 480. The game user interface includes shop to buy items, setting scene, world / level selection screen, gameplay scene, pause scene, game complete scene, game over scene and tutorial scene. Apart from the scene, the game shall allow user to buy items, select level, adjust game setting, pause game function, and tutorial function for user.

Life Tree will be of the most benefit to teenagers and children. This game can educate the young generation about the consequences of pollution during the gameplay. Furthermore, the game shall has an easy to use interface and tutorials which can help user to more understand and attracted to the game. On the other hand, Life Tree can also use as a teaching material for students.

## Intended Audience and Document Overview

This SRS document is produce after the mutual agreement between development team and client through interviews to bind a contract. The details of SRS document should read and approved by both parties.

Section 2. This section of the SRS document will produces a full description of the product by specifying the product perspective, high level functionality of product and also general requirement of the product.

Section 3. This section of SRS describes the more specific requirement of the software. Use case diagram and state diagrams are use to describe the External Interfaces, Functional Requirement, and Behavior Requirements.

Section 4. This section of SRS explains non-functional requirements and focuses on software attributes such as maintainability, security, performance, reliability, usability and others.

## 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

**1.5.1 Document Conventions**

Several formatting conventions have been followed throughout the entire document:

1. Section titles are 18pt Arial font.
2. Subsection titles are 14pt Arial font.
3. Any further subsection breakdown is 12pt Arial font.
4. All text contained in this document is 11pt Arial font.
5. All sections and subsection are numbered using the X.X.X… format, where X represents numbers.
6. **Introduces terms are in bolded Times New Roman italics.**
7. Any further repetition of these terms is in Times New Roman italics.

**1.5.2 Naming Conventions**

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. An example citation guide is posted for you on the website.>

# Overall Description

Life Tree is a Windows phone’s game built on XNA games engine which founded by Althen and his teammates to join windows Imagine Cup. After the competition, the game passed to us for conversion and extension. After reverse reengineering the windows version Life Tree, the game is converted to Android platform and using AndEngine game engine due to previous game engine is incompatible with Android platform.

<General Diagram>

## Product Perspective

## Product Functionality

<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 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, will be effective.

TO DO:

1. Provide a bulleted list of all the major functions of the system

2. **(Optional)** Provide a Data Flow Diagram of the system to show how these functions relate to each other.>

## Users and Characteristics

There will be only one user type playing the game. The user known as Player. No special knowledge or skills shall be assumed on the user.

The Player shall comfortable with Android smartphones / tablets and has no difficulties on English as the game shall operate in English language.

## Operating Environment

The Life Tree shall able to execute on Android-powered smartphone / tablet. The Android version should be version 4.0 and above. The optimize screen resolution is 800 x 480. No network connection is needed during the gameplay.

## Design and Implementation Constraints

After the requirement analysis process the following constraints were identified:

* Life Tree is built on Android platform using Java Programming language, therefore the project must be implement on Java programming language.
* Life Tree is a standalone application.
* Life Tree does not support multi-touch screen
* Life Tree only available for Android Operating System with version 4.0 or higher
* Only one player are available at a time

## User Documentation

No user documentation is available for the game currently. The in-game tutorials feature shall implement inside the gameplay. The tutorials will guide the user step by step during the gameplay.

## Assumptions and Dependencies

List of assumed factors that could affect the requirement of the game product.

* The Android OS must be in place.
* The minimum Android version must appropriate.
* Screen resolution must be in 800 x 400 to achieve the best resolution.
* The smartphone / tablet hardware resources such as RAM and must be adequate for the game application.

# 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: Brake 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. For more information please refer to your UML guide and the MiniThermostat SRS example file.>

# 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 or, on your ABM related research, 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.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>