Software Requirements Specification

for

Life Tree

Document 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: | 12/11/2014 |

Contents

Revisions 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 2

1.5 Document Conventions 2

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 4

2.4 Operating Environment 4

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 8

3.3 Behaviour Requirements 9

4 Other Non-functional Requirements 20

4.1 Performance Requirements 20

4.2 Safety and Security Requirements 20

4.3 Software Quality Attributes 20

Appendix A – Data Dictionary 21

Appendix B - Group Log 24

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Choo Ting Wei | Complete the requirement for the Life Tree Game project. | 12/11/2014 |

# 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 0.5. Life Tree version 0.5 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 used 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

SFX - sound effect

GUI -Graphical User Interface

SRS – Software Requirement Specification

## 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**

1. All constants are CAPITALIZED.
2. All variable representing members are prefixed with m.
3. All system variables are prefixed with s.
4. All texture regions are postfixes with TR.
5. All tiled texture region postfixes with TTR.
6. All sprites are postfixes with \_s.
7. All texts are postfixes with \_t.

## References and Acknowledgments

Sommerville, I. (2011). Software Engineering (9th ed.). Pearson: Massachusetts.

# Overall Description

## Product Perspective

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.



**Figure 1**: *Life Tree gameplay scene*

## Product Functionality

Functionality of Life Tree:

* Can choose game world/level
* Can view the gameplay instruction
* Can mute the background music
* Click mute the sound effect
* Can buy item
* Can switch gameplay mode

command

item

gameplay currency

result

Shop

Player

**Figure 2**: *Life Tree* *Data Flow Diagram (context level)*

## 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 1200 x 800 WXGA pixels(216ppi) 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

The user interface shall consists of three main GUI screens. These interfaces are: Main Menu, World/Level Select, and Item Shop. All interfaces shall have consistent component such as button, text and image which is corresponding to the device screen display.

**3.1.1.1 Main Menu**

The Main Menu screen shall serve as the main UI for the game. Once the game is run, the main menu shall be in the standby state.

Main Menu shall display the following components:

1. The item shop button
2. The start game button
3. The endless button
4. The help button
5. The setting button
6. The exit button
7. The Neo’s eyes winking animation

**Figure 3**: A screenshot of Main menu Screen



The button shall be able to perform the following operation once there is an input from the player:

* Item shop button-switch the screen from main menu screen to Item shop screen.
* Start game button-switch the screen from main menu screen to level selection screen.
* Endless button –switch to endless gameplay mode.
* Help button-pulls down the panel and shows the general tutorial.
* Setting button- pulls down the panel and shows the setting that is available.
* Exit button-close the game.

**3.1.1.2 World/Level Selection Screen**

World/Level Selection Screen shall be appearing once the start button is clicked. It serves as the second main screen in the entire game. In this screen, player is asked to choose the map in order to proceed to the next screen. The variable mCurrentScreen in the main menu class will be assigned to level selection screen.

World/Level Selection Screen shall display the following components:

1. Map button
2. Back button
3. Map Tag

**Figure 4:** A screenshot of World/ Level Selection Screen



The button shall be able to perform the following operation once there is an input from the player:

* Map button-enter the game world with the chosen map.
* Back button-allow player to switch back to the main menu screen.

**3.1.1.3 Item Shop Screen**

Item shop screen serve as a platform that sell the in game item to the player. The item only can be bought using the game currency. The variable itemAmount\_t will record the amount of item that bought by the player and store it into the shared preferences (SHARED\_PREFS\_ITEM).

Item Shop Screen shall display the following components:

1. The price tag of the item.
2. The buy button.
3. The item button.
4. The back button
5. The enlarge Image of the item if it is clicked.
6. The amount of item.
7. Game currency.
8. The item name.

**Figure 5**: A screenshot of the Item Shop Screen



The button shall be able to perform the following operation once there is an input from the player:

* Item button-once the item is clicked the price tag will be appear.
* Buy button- once the item is clicked the player is allow to buy the item.
* Back button-allow player to go back to main menu screen

### Hardware Interfaces

* Minimum Requirements:
* Processor: Nvidia Tegra 3 (Quad-core 1.2 GHz Cortex-A9)
* RAM: 1GB
* Disk Space: 400MB free space
* Screen Display: 7-inch ,1280×800 WXGA pixels (216 ppi)
* Recommended Requirements:
* Processor: Nvidia Tegra K1 (Dual-core 2.3 GHz Denver)
* RAM: 2GB
* Disk Space: 800MB free space
* Screen Display: 10.1 inch, 1280×800 pixels (149 ppi)

### Software Interfaces

Android version: v4.0 or higher

### Communications Interfaces

N.A. – LifeTree does not requires Internet Connection or any online feature.

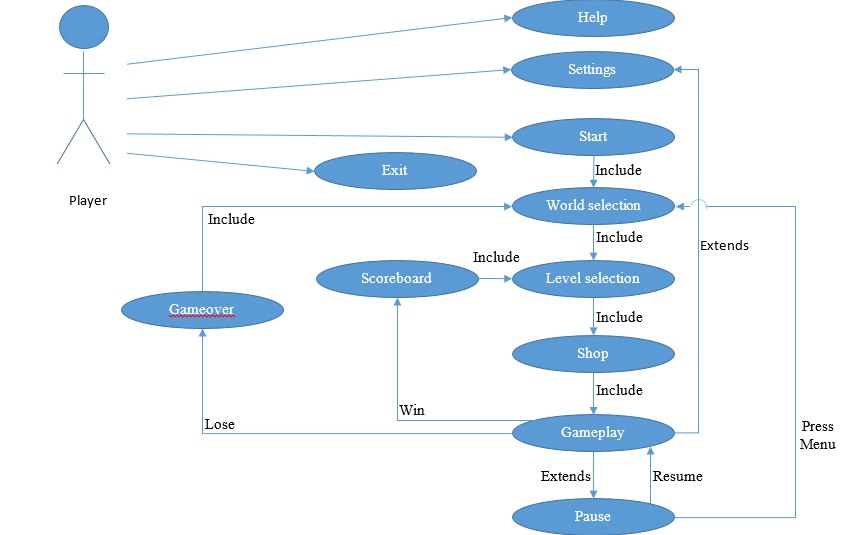
## Functional Requirements

### User Interface Operations

* Click start button to enter game world/level screen: Upon the start button is clicked, the player redirect to the world/level selection screen.
* World/level is chosen: Once the world/level is chosen, the player will be redirect to the item shop screen.Item bought by the player will store into the itemAmount variable.After that,player will sent to the game world.
* Click the help button: Once the help is triggered, there will be a panel pop out and show up the general tutorial.
* Click the setting button: Once the setting is clicked, the mute music button and mute sound effect button will be shown. The operation of mute button will set the initial value of mMusicMuted and mSoundsMuted.
* Click the shop button: The item shop screen will bring to the player and allow player to buy the item that can be used in the gameplay.
* Click the endless button: switch to endless game mode.

## Behaviour Requirements

### Use Case View



### Sequence Diagram

Sequence diagram (Help)

Player

Help

Application

Open help

Parse help details

Retrieve help notes

Display help details

Close help

Destroy help menu

Return to Homescreen

Sequence diagram (Settings)

Player

Settings

Application

Open Setting

Setting details

Update game settings

Display settings menu

Close settings

Destroy setting menu

Return to Homescreen/Gameplay

Change settings

Display updated settings

Change game settings

Sequence diagram (Exit)

Exit

Application

Start game

Request exit Application

Dispose objects/variables

Display warning

Exit confirmation

Close application

Decline exit

Return game Homescreen

Player

Alt

(If confirm exit)

(If decline exit)

Sequence diagram (Start)

Player

Start

Application

Start game

Request world selection

Close Homescreen

Open world selection

Sequence Diagram (World selection)

Player

World selection

Application

World selection details

Display world selection

Selected world

Request level selection

Close world selection

Open level selection

Back

Return to Homescreen

Open world selection

Load world details

Alt

(If a world is selected)

(If a back button pressed)

Sequence diagram (Level selection)

Player

Level selection

Application

Level selection details

Display level selection

Selected level

Request level selection

Load level difficulty

Open shop

Back

Return to world selection

Open level selection

Load level details

Alt

(If a level is selected)

(If a back button pressed)

Sequence diagram (Shop)

Player

Shop

Application

Item details

Display items and prices

Select item to buy

Request item

Check availability

Open shop

Load shop items

Notification

Check player coins

Update player inventory

Notify player purchase complete

If insufficient coins

Notify player purchase failed

Back

Return to level selection

Done

Request gameplay

Close shop

Open gameplay

Alt

(If player buys item)

Alt

(If sufficient coins)

(If insufficient coins)

(If back is pressed)

(If done is pressed)

Sequence diagram (Gameplay)

Player

Gameplay

Application

Open gameplay

Load level & monster

Spawn monsters

Show monsters

Update monsters’ position

Updated monsters’ position

Loop

Gameplay details

Show game screen

Choose orb

(If hit and orb element == enemy element)

Eliminate monster

Update monster count

Update score

Alt

Alt

(If enemy count == total enemy)

Updated scores

Show updated scores

Destroy level & monster

Show selected orb

Release orb

Alt

(If select orb == existing orb)

Choose orb

Remove orb selection

(If select orb != existing orb)

Choose orb

Combine orb element

Show combined orb

Alt

Alt

Open scoreboard

Alt

(If player choose orb)

(If player use item)

(If monster reach lifetree)

Alt

(If player press settings)

Monster reach lifetree

Ref

Item used

Ref

Settings

Ref

Sequence diagram (Item used)

Player

Gameplay

Application

Selected & use item

Lock item temporary

Update inventory

Update gameplay details

Apply effect details

Item effect & lock timeout

Sequence diagram (Monster reach Lifetree)

Player

Gameplay

Application

Monster reach lifetree

Deduct player health

Destroy monster

Player details

Updated life bar & lifetree condition

Alt

(If health <= 0)

Destroy level & monster

Open gameover

Sequence diagram (Gameover)

Player

Gameover

Application

Display gameover screen

Next

Request world selection

Close gameover screen

Open world selection

Close gameplay

Sequence diagram (Scoreboard)

Player

Scoreboard

Application

Display scoreboard

Next

Request next level

Close scoreboard

Close gameplay

Load level selection

Sequence diagram (Pause)

Player

Pause

Application

Pause

Request pause

Display pause screen

Request quit game

Resume

Halt gameplay

Request resume

Resume gameplay

Menu

Stop & close gameplay

Open world selection

Class Diagram



# Other Non-functional Requirements

## Performance Requirements

* + Game will not crash over 10 time average
  + The RAM usage of the game will not exceed 512MB
  + The game will not be lower than 60 frame per seconds.
  + During gameplay, the CPU utilization should not exceed 40%.
  + The application will not take more than 10 seconds to start.
  + Any item bought within the shop will not take more than 5 seconds.

## Safety and Security Requirements

* The application should be secure enough to prevent the corruption of codes from unauthorized users.
* The application shall minimise the number of permissions in order to mitigate risk of unintentionally misusing those permissions.
* The application shall be built by using Android SDK, instead of using native code with Android NDK. Applications built using Android NDK are complex, less portable and more likely to have memory corruption errors.

## Software Quality Attributes

### Availability

The application must be runnable when requested. The application should be runnable 90% of the time.

### Maintainability

All code shall be documented. Each function and parameters are given reasonable names to avoid confusion.

### Usability

The application shall be easy to understand so that users know how to play the game entirely on their first try.

### Flexibility

The application shall be runnable on any Android devices with version 4.0 and above.

Appendix A – Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Data Type** | **Access modifier** | **Description** |
| Entity:LifeTreeActivity | | | |
| sCameraWidth | float | public | System variable. Camera width will initial with the value that measured by the program itself. |
| sCameraHeight | float | public | System variable. Camera height will initial with the value that measured by the program itself. |
| mCamera | Camera | public | In-game camera view for in-game camera use |
| SHARED\_PREFS\_MAIN | String | public | Constant Variable.To stores the name of shared preferences. |
| SHARED\_PREFS\_ITEM | String | public | Constant variable.To stores the data of item. |
| SHARED\_PREFS\_MUSIC\_MUTED | String | public | Constant variable.To stores the music status. |
| SHARED\_PREFS\_SOUNDS\_MUTED | String | public | Constant variable.To stores the sound effect status. |
| Entity: GameManager | | | |
| mGameLevel | GameLevel | public | To get instances of the GameLevel class. |
| mGameLevelGoal | GameLevelGoal | public | To use the interface of the GameLevelGoal class. |
| Entity: ResourceManager | | | |
| INSTANCE | Resource  Manager | public | Constant Variable.Use by the other class to perform some specific action. |
| mCameraWidth | float | public | To store the value of sCameraWidth in LifeTreeActivity. |
| mCameraHeight | float | public | To store the value of sCameraHeight in LifeTreeActivity. |
| mCameraScaleFactorX | float | public | To store the value of Scale factor. |
| mCameraScaleFactorY | float | public | To store the value of Scale factor. |
| menuBackgroundTR | TextureRegion | public | To store the image region. |
| menuBackground2TR | TextureRegion | public | To store the image region. |
| menuBackground3TR | TextureRegion | public | To store the image region. |
| ttNextButtonTR | TextureRegion | public | To store the image region. |
| map1TR | TextureRegion | public | To store the image region. |
| map2TR | TextureRegion | public | To store the image region. |
| map3TR | TextureRegion | public | To store the image region. |
| map4TR | TextureRegion | public | To store the image region. |
| map1TitleTR | TextureRegion | public | To store the image region. |
| map2TitleTR | TextureRegion | public | To store the image region. |
| map3TitleTR | TextureRegion | public | To store the image region. |
| map4TitleTR | TextureRegion | public | To store the image region. |
| itemTR | TextureRegion | public | To store the image region. |
| itemTR2 | TextureRegion | public | To store the image region. |
| itemShowTR | TextureRegion | public | To store the image region. |
| itemShow2TR | TextureRegion | public | To store the image region. |
| cloudTR | TextureRegion | public | To store the image region. |
| musicToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| soundToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| shopToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| startToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| exitToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| endlessToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| helpToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| settingToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| generalTutorialTTR | TiledTextureRegion | public | To store the tiled image region. |
| buyToggleTTR | TiledTextureRegion | public | To store the tiled image region. |
| Entity: SFXManager | | | |
| mMusic | Music | private | To store the music directory and the music name. |
| mlevelSelectMusic | Music | private | To store the music directory and the music name. |
| mClick | Sound | private | To store the sound effect directory and the sound effect name. |
| mMusicMuted | Boolean | public | To store the music status |
| mSoundsMuted | Boolean | public | To store the sound effect status |
| Entity: SceneManager | | | |
| mCurrentScene | ManagedScene | public | To store the current scene. |
| mNextScene | ManagedScene | private | To store the next scene to be display. |
| mEngine | Engine | private | To store the engine setup . |
| mNumFramesPassed | Integer | private | Use to indicate the number of frame has passed. |
| mLoadingScreeen  HandlerRegistered | boolean | private | To check the loading screen handler. |
| Entity: MainMenu | | | |
| mCurrentScreen | MainMenuScreens | public | To store the current scene. |
| mHomeMenuScreen | Entity | private | A new screen which can attach child on it. |
| mLevelSelectScreen | Entity | private | A new screen which can attach child on it. |
| mShopScreen | Entity | private | A new screen which can attach child on it. |
| lifeTreeBg\_s | Sprite | private | To store the main game background image. |
| lifeTreeLevelSelectBG\_s | Sprite | private | To store the level selection background image. |
| lifeTreeShopBg\_s | Sprite | private | To store the item shop background image. |
| map1\_title\_s | Sprite | private | To store the name tag of the map. |
| map2\_title\_s | Sprite | private | To store the name tag of the map. |
| map3\_title\_s | Sprite | private | To store the name tag of the map. |
| map4\_title\_s | Sprite | private | To store the name tag of the map. |
| itemS01\_s | Sprite | private | To store the showed item. |
| itemS02\_s | Sprite | private | To store the showed item. |
| itemAmount\_t | Text | public | To store the amount of an item. |
| priceTag\_t | Text | public | To store the price of an item. |

***Table 1.0****: Data Dictionary*

Appendix B - Group Log

|  |  |
| --- | --- |
| Meeting No. | 1 |
| Date | 15th September 2014 (Monday) |
| Time | 12pm – 2pm |
| Venue | Image Processing Lab |
| Agenda | Discussion on Game Project and Platform Selection |
| Topic of Discussion | 1. Decision on develop project either build from scratch or modify existing game. 2. Decision on which platform should use when develop the game project. 3. Decision on which type of game should develop. |
| Result | 1. Develop game project from scratch. 2. Use Android platform to develop the game project. 3. Develop a tower defense android game. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 2 |
| Date | 22nd September 2014 (Monday) |
| Time | 12pm – 2pm |
| Venue | Image Processing Lab |
| Agenda | Assign Role for Each Member and Prepare Proposal |
| Topic of Discussion | 1. Assigning role for each member. 2. Name of game project. 3. Specific game details and requirements. 4. Project scope. 5. Reference game to use. 6. Dateline for project proposal draft. |
| Result | 1. Role is assigned for each members. 2. The name of project is Life Tree. 3. Game specification and requirement is recorded. 4. Project scope defined. 5. Defender II as our reference game. 6. Proposal draft dateline is on 29th September 2014 |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 3 |
| Date | 24th September 2014 (Wednesday) |
| Time | 6pm – 10.30pm |
| Venue | Image Processing Lab |
| Agenda | Design Coding of Life Tree |
| Topic of Discussion | 1. Programming method. 2. Use case design. 3. Coding task allocation. |
| Result | 1. Pair programming. 2. Use case diagram draft. 3. Coding task allocated. |
| Participation | 1. Chai Min Chun 2. Lai Ted Liang |

|  |  |
| --- | --- |
| Meeting No. | 4 |
| Date | 26th September 2014 (Friday) |
| Time | 6pm – 10pm |
| Venue | Image Processing Lab |
| Agenda | Discuss Proposal Draft |
| Topic of Discussion | 1. Background for Life Tree. 2. Limited knowledge on pollution index. 3. Objectives of Life Tree. 4. Team and project goal. 5. Differences between Defender II and Life Tree. 6. Additional features to make Life Tree more attractive. |
| Result | 1. Background of Life Tree is about pollution. 2. A complete version of proposal |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 5 |
| Date | 29th September 2014 (Monday) |
| Time | 12pm – 1pm |
| Venue | Image Processing Lab |
| Agenda | Review Proposal and Prepare Presentation Slides |
| Topic of Discussion | 1. Go through complete version of proposal. 2. Prepare presentation slide. |
| Result | 1. Final version of proposal is produced. 2. Presentation slide is produced. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 6 |
| Date | 1st October 2014 (Wednesday) |
| Time | 6pm – 9.30pm |
| Venue | Image Processing Lab |
| Agenda | Enhance Proposal |
| Topic of Discussion | 1. Discuss commend given by lecturer 2. Mistakes on proposal. |
| Result | 1. Corrected version of proposal. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 7 |
| Date | 6th October 2014 (Monday) |
| Time | 6pm – 9.30pm |
| Venue | Image Processing Lab |
| Agenda | Develop use case diagram |
| Topic of Discussion | 1. Develop use case diagram. |
| Result | 1. Use case diagram completed. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 8 |
| Date | 16th October 2014 (Thursday) |
| Time | 3pm – 5pm |
| Venue | Image Processing Lab |
| Agenda | Follow-up meeting |
| Topic of Discussion | 1. Report progress for each team members. |
| Result | 1. Review progress. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 9 |
| Date | 16th October 2014 (Thursday) |
| Time | 12pm – 1.30pm |
| Venue | Image Processing Lab |
| Agenda | Design Coding of Life Tree |
| Topic of Discussion | 1. Programming method. 2. Use case design. 3. Coding task allocation. |
| Result | 1. Pair programming. 2. Use case diagram draft. 3. Coding task allocated. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 10 |
| Date | 3rd November 2014 (Monday) |
| Time | 12pm – 1.30pm |
| Venue | Image Processing Lab |
| Agenda | Follow-up meeting |
| Topic of Discussion | 1. Report progress for each team members. |
| Result | 1. Review progress. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |

|  |  |
| --- | --- |
| Meeting No. | 11 |
| Date | 10th November 2014 (Monday) |
| Time | 6pm – 11.30pm |
| Venue | Image Processing Lab |
| Agenda | SRS |
| Topic of Discussion | 1. Develop SRS. |
| Result | 1. SRS is completed. |
| Participation | 1. Chai Min Chun 2. Choo Ting Wei 3. Lai Ted Liang 4. Lim Ban Yong 5. Willis Fung Shin Choi |