T02 Group 5 Deliverable #1

SE 3A04: Software Design II – Large System Design

March 4, 2017

1   Introduction

The following is a description of the product to be developed, as well as an overview of the SRS.

1.1   Purpose

The purpose of this document is to provide a general description of this project and to specify the requirements for the project. The focus of this SRS is the development of the software product. The SRS is also meant to be used as a reference by the developers to ensure that the specified requirements have been fulfilled. The intended audience of this document is the project's clients, i.e. Dr. Ridha Khedri and the teaching assistants for SFWR ENG 3A04, as well as future engineers working on the project.

1.2   Scope

SpaceBase Ephemeris (SBE) is a mobile simulation game that models the overall behavior of a space settlement on a celestial body, i.e. a real-life system. The game represents how different sub-systems interact with each other and affect the overall system. It replicates how each of the sub-systems of a space base react to different stimuli from outside of the system. Therefore, the player must ensure that the sub-systems are working in the desired fashion to keep the base operating. The game mimics the operations of a space settlement, and places the user as the head of the base. Hence, the key duty of the player is to ensure that the base remains operating. The player does this by keeping an eye on all the sub-systems and maintaining them whenever there is a need. This will be done by assigning tasks to all the members of the base, which could be anything from fixing a breach in the walls of the base to interaction with alien flora and fauna. If there are multiple stimuli happening concurrently, then the player must prioritize the tasks, based on how critical they are and on what level they affect the safety of the community. The main objective of this game is to provide entertainment to its users. However, it will also develop time management skills of the users, as well as improve their multitasking abilities.

1.3   Definitions, Acronyms, and Abbreviations

1. SBE: SpaceBase Ephemeris, the title of the game

1.4   References

No references were used for this document.

1.5   Overview

This document describes the product software that is to be developed as well as the requirements specifications. The second section of this document provides an overall description of the game. It provides product perspective when compared to another related product. It describes major functions of the product as well as the user characteristics. It also explains the constraints of the system as well as the assumptions for this project. The third section of this document specifies the functional software requirements of this game. It

provides sufficient details to design a system with the specific functionality specified in the requirements and to test that the design has fulfilled those requirements. The final section of the document is the non-functional requirements. A Division of Labour section is also placed at the end of the document which details the contributions of each team member.

2 Overall Description

The following is a general description of the product and its requirements. For more specific requirements, see the requirements sections.

2.1 Product Perspective

SBE is largely independent and self-contained, but does interact with social media systems in the case that the user decides to post a snap-shot of their gamestate. It does not require network communication for other aspects. As the game is an Android app, it may be distributed on the Google Play Store, but the Store is not part of the system. The game will contain elements of various existing simulation games, but is not intended to completely emulate any of them. The simulation is intended for entertainment purposes only, so bears only a thematic relationship to some scientific simulation software.

2.2 Product Functions

The user will be able to view compositions of various sub-views of the system. Each sub-view corresponds to a different subsystem. The 3 sub-systems of the application are the atmospheric system, power system, and the station's population. The user will be able to adjust the atmospheric pressure in their station, and control the flow of gasses with airlocks and other devices which the user can operate. The user will be able to generate power in the station, control the flow of power with wires, and control which devices draw power from the supply. In the event of failures with atmospheric contain or power transfer, the subsystems will respond appropriately. They will be able to issue orders to the population of the station, who will follow those order autonomously. The station population will react to changes in the atmospheric pressure, and operate devices connected to the power network. The user will be able to save the state of their game, and share the corresponding file with other users, who will in turn be able to load it. **(Creative/Innovative)** The user will be able to post a snapshot of their game-state to social media from within the application.

2.3 User Characteristics

Users are expected to be fluent in English, and at least 10 years of age. Users are expected to be generally familiar with the Android operating system and Android apps. Users are expected to have only a cursory understanding of the subject matter, as SBE is intended for entertainment rather than scientific simulation. Most users are assumed to have appropriate social-media account, but it is not a requirement to use the system.

2.4 Constraints

The following are constraints on the development of the system.

1. The system must be produced as an Android app.

2. The system must consist of several separate subsystems. (at least 3)

2.5 Assumptions and Dependencies

The following are assumptions that affect the requirements for the system.

1. It is assumed that the device running the application will have the Android operating system available, with sufficient resources available.

2. It is assumed that the device running the application will have internet access.

2.6 Apportioning of Requirements

Section N/A as development has not begun.

3 Functional Requirements

1. BE1. The user launches the application

   (a) User

      i. The system provides a view of all the sub–systems in their present state to the user
      ii. The system must allow the user to select a sub-system to interact with

2. BE2. The user wants to adjust an atmospheric or power controlling device within the station.

   (a) User

      i. The system must allow the user to select the device.
      ii. Once an atmospheric device is selected, the system must display the atmospheric/power state of the station.
      iii. The system must allow the user to select an operation for the device and any appropriate values.
      iv. The appropriate subsystem must react to any operations input by the user, which must cascade to other subsystems.

3. BE3. The user wants to issue a work order the station population.

   (a) User

      i. The system must allow the user to select various aspects of the station to be manipulated.
      ii. The system must display the available population to carry out the order.
      iii. The system must allow the user to specify the order to be issued.
      iv. The system must display the progress of the order as it is completed and correctly react the final result.

4. BE4. The user wants to update the application to the latest version

   (a) User

      i. The system must prompt the user to update the system when the application is launched
      ii. The system must allow the user to open the Play Store and download the latest version.

5. BE5. The user wants to change overall functionality settings of the system

   (a) User

      i. The system must provide an interface for editing values of how the system works based on customizable aspects of the system
      ii. The system must store these new settings and incorporate them into the systems functionality

6. BE6. The user wants to share a snapshot of the system

   (a) Social Media Application

      i. The system must be able to provide data that is compatible with the specified social media framework

   (b) User

      i. The system must provide a means of sharing users system data via social media

7. BE7. The user wants to save the game-state

(a) User

      i. The system must display an option for saving the game.

      ii. The system must allow the user to specify a save file name and location.

      iii. The system must save the game at the requested location.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

1. LF1. The application must have graphical interface (not just textual) for the users to use.

2. LF2. The graphical interface must be spannable, zoomable, and rotatable.

3. LF3. The majority of the intended users must find the product attractive.

4.1.2 Style Requirements

1. S1. The system must use the native Android application styling.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

1. UH1. The application shall be easy for anybody of age 13 and up to use.

2. UH2. The application shall display the current game-state clearly.

4.2.2 Personalization and Internalization Requirements

1. PI1. Personal settings or game files shall be able to be saved, and loaded at user's will.

4.2.3 Learning Requirements

1. LER1. The application shall include a brief tutorial or manual to guide new users which introduces the game concepts, acronyms, terms, and settings step-by-step.

4.2.4 Understandability and Politeness Requirements

1. UPR1. There will only be English option, however easy to understand symbolic icons will be used in the interface when possible.

4.2.5 Accessibility Requirements

1. ACCR1. This application shall be accessible to individuals with sufficient visual and mechanical capabilities to operate the base Android operating system.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

1. SOLR1. Any interaction between the user and the application must have a maximum response time of 1 second.

2. SOLR2. The view of the application must update in real-time (.1 second delay) to the state of the station.

4.3.2 Safety-Critical Requirements

1. SCR1. Any graphics used must not induce dizziness, tiredness, or seizures.

4.3.3 Precision or Accuracy Requirements

1. PAR2. Any critical values the system depends on must be accurate to six significant digits.

4.3.4 Reliability and Availability Requirements

1. RAR1. The application must have at least 95% uptime.

4.3.5 Robust or Fault-Tolerance Requirements

1. RFT1. In the event of a crash within the system, the system must save the user's data before exiting.

2. RFT2. No data must be lost even in case of system failure.

4.3.6 Capacity Requirements

1. CAPR1. The application must meet the performance requirements in any station with ¡= 50 popuation of the default square area.

4.3.7 Scalability or Extensibility Requirements

1. SER1. N.A.

4.3.8 Longevity Requirements

1. LR1. The application is expected to operate as long as the hardware supports it.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

1. EP1. The application is expected to be used in a lighting environment consistent with the physical device's display brightness

2. EP2. The application is expected to be used in both noisy and quiet physical environments.

4.4.2 Requirements for Interfacing with Adjacent Systems

1. RIAS1. The application is expected to interface with the Twitter authentication and posting API.

4.4.3 Productization Requirements

1. PRR1. This application shall be distributed as an APK file.

2. PRR2. The application must be available on Android Play Store.

4.4.4 Release Requirements

1. RR1. At release the application must be free of significant bugs known by the developers.

2. RR2. The development team will produce 2nd patch release several weeks after the initial, fixing any discovered bugs.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

1. MSM1. A new Module Interface Specification must be made within a week of the date when the requirements are agreed upon.

2. MSM2. Future developers should be able to access the values of each separate subsystem indepedently for debugging purposes.

4.5.2 Supportability Requirements

1. MSS1. An email system where users can interact with the developing team must be set up.

2. MSS2. This application is to run under Android.

4.5.3 Adaptability Requirements

1. ADR1. The application must support the addition or removal of subsystems by developers with minimal effort.

4.6 Security Requirements

4.6.1 Access Requirements

1. SAR1. Any user with a Google account has access to the application from the Google Play Store.

2. SAR2. Any user with valid social-media credentials has access to the social media portions of the application.

4.6.2 Integrity Requirements

1. IR1. The application shall ensure that any attempts to breach the integrity of the user's social media accounts through the application be deterred.

2. IR2. The application will maintain the integrity of stored save files.

3. IR3. The application will not require excess privileges on the Android device (for example the camera).

4.6.3 Privacy Requirements

1. PR1. The application shall not collect any personal information beyond the social media credentials necessary.

2. PR2. The application will notify the user if any privacy policy changes.

4.6.4 Audit Requirements

1. AR1. The development team will search for bugs within the application with both automating and manual testing before release.

4.6.5 Immunity Requirements

1. IMR1. N.A.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

1. CR1. This application shall not contain any symbols or messages that may offend any ethnic or religious groups.

4.7.2 Political Requirements

1. PP3. N.A.

4.8 Legal Requirements

4.8.1 Compliance Requirements

1. LCR1. N.A.

4.8.2 Standards Requirements

1. LSR1. N.A.

A Division of Labour

1. Arfa: Introduction

2. Ian: Overall Description

3. Areeb: Functional Requirements

4. Nishanth: Functional Requirements

5. Steven: Nonfunctional Requirements