

IdentiSky: SRS

Group 2 for SFWR ENG 3A04

Alex Guerrero, guerreap, 1133763
Jabrayil Malikov, malikoj, 1302641
Matt Franceschini, francemj, 1310437
Sam Hamel, hamels2, 1321692
Vicky Bilbily, bilbilv, 1317465

February 8th, 2016

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2	Overall Description	4
2.1	Product Perspective	4
2.2	Product Functions	5
2.3	User Characteristics	5
2.4	Constraints	5
2.5	Assumptions and Dependencies	5
2.6	Apportioning of Requirements	6
3	Functional Requirements	6
4	Non-Functional Requirements	7
4.1	Look and Feel Requirements	7
4.1.1	Appearance Requirements	7
4.1.2	Style Requirements	7
4.2	Usability and Humanity Requirements	8
4.2.1	Ease of Use Requirements	8
4.2.2	Personalization and Internationalization Requirements	8
4.2.3	Learning Requirements	8
4.2.4	Understandability and Politeness Requirements	8
4.2.5	Accessibility Requirements	8
4.3	Performance Requirements	8
4.3.1	Speed and Latency Requirements	8
4.3.2	Safety-Critical Requirements	8
4.3.3	Precision or Accuracy Requirements	8
4.3.4	Reliability and Availability Requirements	8
4.3.5	Robustness or Fault-Tolerance Requirements	8
4.3.6	Capacity Requirements	9
4.3.7	Scalability or Extensibility Requirements	9
4.3.8	Longevity Requirements	9
4.4	Operational and Environmental Requirements	9
4.4.1	Expected Physical Environment	9
4.4.2	Requirements for Interfacing with Adjacent Systems	9
4.4.3	Productization Requirements	9
4.4.4	Release Requirements	9
4.5	Maintainability and Support Requirements	9
4.5.1	Maintenance Requirements	9
4.5.2	Supportability Requirements	9
4.5.3	Adaptability Requirements	9
4.6	Security Requirements	9
4.6.1	Access Requirements	9

4.6.2	Integrity Requirements	9
4.6.3	Privacy Requirements	10
4.6.4	Audit Requirements	10
4.6.5	Immunity Requirements	10
4.7	Cultural and Political Requirements	10
4.7.1	Cultural Requirements	10
4.7.2	Political Requirements	10
4.8	Legal Requirements	10
4.8.1	Compliance Requirements	10
4.8.2	Standards Requirements	10
5	Division of Labour	10

1 Introduction

1.1 Purpose

The purpose of this documents is to establish a basis for a agreement between the users and developers on what the application will do. It outlines the program's requirements, both functional and non-functional. This document provides a foundation for development and will be used for verification and validation. This document is to be used as a reference for the development team as well as the professor and teaching assistants of SFWR ENG 3A03.

1.2 Scope

IdentiSky is a mobile application that identifies objects in the night sky. These objects are limited to stars and their associated **constellations**, planets, planes, and noteable **satellites** such as the **ISS**. Search results are based on the user's current location, date, time, and weather in the local area, as well as the user's answers to questions queried by the application. The goal of IdentiSky is answer the question, "What is that?" when **sky gazing** or simply looking up at the sky.

1.3 Definitions, Acronyms, and Abbreviations

Constellation - a group of stars forming a recognizable pattern that is traditionally named after its apparent form or identified with a mythological figure

Satellite - a celestial body orbiting or an artificial body placed in orbit around the earth

ISS - International Space Station

Sky Gazing - gazing at the sky for pleasure or with an astronomical interest

1.4 References

- Grabianowski, Ed. "The Ten Most Important Satellites Orbiting Earth Now". Internet: <http://io9.gizmodo.com/391538/the-ten-most-important-satellites-orbiting-earth-now>, May 19, 2008 [Feb. 4, 2016].

1.5 Overview

The rest of the document is organized as follows: 2 - overall description of the software, 3 - functional requirements, 4 - non-functional requirements , and lastly 5 - division of labour.

2 Overall Description

2.1 Product Perspective

IdentiSky is a standalone android application, and therefore is completely self-contained and is in no way part of a larger system.

2.2 Product Functions

IdentiSky's purpose is to identify objects that are currently in the sky in the user's relative vicinity, it accomplishes this through two main functions:

1. The application will track and process information about the user, such as finding the user's location, the direction they are currently facing, altitude, and current weather in their location in order to accurately identify objects in the sky that they are potentially looking at.
2. The application will identify objects by taking user inputs and also querying the user with questions based on the possible object they wish to identify.
3. The Application shows a map of all the notable objects current in the sky, the user can select items on the map for further info, and can filter out certain items using categories. They can also select a specific location, date, and time in order to see what the sky will contain in another setting.
4. The Application is able to notify the user when they are in a place where they can see notable objects in the sky. The app can also tell the user if optimal **sky gazing** areas are nearby

The application will be able to identify the following:

- * Planets
- * Notable stars
- * **Constellations**
- * Notable **satellites**
- * Airplanes

2.3 User Characteristics

The general users of the application are going to be people of any age above six that have the ability to see. The users will need little to no background in astronomy, however, most likely, a majority of the users will be astronomy enthusiasts who have a more in depth knowledge of Space.

2.4 Constraints

The application must be developed for Android, and it must interface with Google maps. There are non-adjustable time constraints for several deliverables and the overall project must be complete by the 8th of April. In addition, since the budget for this project is zero dollars, all technology must be free and/or open source material.

2.5 Assumptions and Dependencies

There are three main factors that might change overtime such that it would require revising the SRS. The first of these factors is the likelihood that updates to the operating systems can affect the UI and implementation of the program such that changes will have to be made. Another factor is if Google changes aspects of their Maps API such as charging developers for it, or removing features, then the SRS must change since the use of the Google Maps API is one of the constraints of this

project. The third possible factor is any laws that can change or appear that pertain to the tracking and processing of user information. Since the application takes part in these actions, any changes can become new constraints and/or requirements that must be added or removed.

2.6 Apportioning of Requirements

There are no plans to delay any of the requirements thus far, as a formal project timeline has not been prepared yet.

3 Functional Requirements

VP1. User

BE1.1 User wants to open application

- i. The system must provide a way to be initiated by the user from the main screen of their device.

BE1.2 User wants to identify an object in the night sky.

- i. The system must provide an option to begin an object identification process.
- ii. The system must be able to take in gesture-based user-input.
- iii. The system must be able to identify the user's location, for the Location Expert to use.
- iv. The system must be able to identify the weather at the user's current location.
- v. The system must be able to identify the time and date at the user's location.
- vi. The system must provide multiple-choice questions for the user to answer, to gain more information about the object in question.
- vii. The system must have a database of objects to refer to and compare the user's information to.
- viii. The system must be able to score each possible object on how likely it is that they are the correct object.
- ix. The system must be able to list the 10 most likely objects according to their score.

BE1.3 User wants more information about an identified object.

- i. The system must provide a way for the user to select one object from a list of identified objects.
- ii. The system must display extra information about an object to the user when prompted.

BE1.4 User wants to know what is currently visible in the sky.

- i. The system provides an option to view all objects currently visible in the sky.
- ii. The system must provide an interactive sky-map of all objects currently visible in the sky.

- iii. An option is provided to filter the objects on the map by category: Planets, Stars, **Constellations**, **Satellites**, and Airplanes.
- BE1.5 User wants to know what will be visible in the sky at a particular time and location.
 - i. On the interactive sky-map screen, the system provides an option to change the time and/or location of the sky-map.
- BE1.6 User wants a better vantage point of the sky.
 - i. The system must provide an option to request nearby **sky gazing** locations.
 - ii. The system must list nearby **sky gazing** locations to the user, arranged by distance or altitude, according to the user's discretion.
 - iii. The system must provide a map-view of all nearby **sky gazing** locations.
- BE1.7 User wants to modify the application settings.
 - i. The system provides an option to navigate to a settings page.
 - ii. The system settings page allows the user to modify settings such as notifications, volume, and text size.
- VP2. Developer
 - BE2.1 User reports a bug.
 - i. The system must have a "developer mode" for assessing performance and bugs.
 - BE2.2 A new identifiable object has been discovered/created.
 - i. The system must allow for updates to be made to the database.
- VP3. System
 - BE3.1 There is an astronomical phenomenon currently viewable near the user.
 - i. The system must be able to send the user a notification when there is a visible astronomical phenomenon taking place, if the user has this notification enabled.
 - BE3.2 It is an excellent day for **sky gazing**.
 - i. The system must be able to send the user a notification informing them of the clear weather for **sky gazing**, if the user has this notification enabled.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

LF1. The system shall have a simple interface.

4.1.2 Style Requirements

LF1. The app will be styled in a modern way, in order to attract all audiences.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

- UH1. The system should be usable by anyone from the age of 7 years old to 80 years old.
- UH2. The system shall be used by people with no previous training.

4.2.2 Personalization and Internationalization Requirements

- UH1. The app should not include any idioms which are not transferable between languages.

4.2.3 Learning Requirements

- UH1. The product shall be able to be used by members of the public who will receive no training before using it.

4.2.4 Understandability and Politeness Requirements

- UH1. The product shall use symbols and words that are naturally understandable by the user community.
- UH2. The product shall hide the details of its construction from the user.

4.2.5 Accessibility Requirements

- UH1. N/A

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

- PR1. The app must be as quick as possible.
- PR2. The app must not have very many load screens.
- PR3. The system must load all data in 5-10 seconds.

4.3.2 Safety-Critical Requirements

- PR1. N/A

4.3.3 Precision or Accuracy Requirements

- PR1. The system must be accurate enough to identify an item in the night sky.

4.3.4 Reliability and Availability Requirements

- PR1. The system must be offline as little as possible as it is always night somewhere.

4.3.5 Robustness or Fault-Tolerance Requirements

- PR1. The system must be robust enough to withstand 100s of users accessing the data at any one time.

4.3.6 Capacity Requirements

PR1. N/A

4.3.7 Scalability or Extensibility Requirements

PR1. N/A

4.3.8 Longevity Requirements

PR1. N/A

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

OE1. Outside on a clear day with connection to the internet.

4.4.2 Requirements for Interfacing with Adjacent Systems

OE1. N/A.

4.4.3 Productization Requirements

OE1. N/A.

4.4.4 Release Requirements

OE1. The system shall be released on the Google Play Store.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. Shall be updated regularly via Google Play store.

4.5.2 Supportability Requirements

MS1. Support shall be provided via comments on the Google Play store.

4.5.3 Adaptability Requirements

MS1. Shall be able to adapt to sudden changes in external databases.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. Anyone with an Android phone shall be able to use this app.

4.6.2 Integrity Requirements

SR1. There shall be some encryption within the app.

4.6.3 Privacy Requirements

SR1. No personal information shall be required.

4.6.4 Audit Requirements

SR1. N/A.

4.6.5 Immunity Requirements

SR1. N/A.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1. The app will not mention any specific culture so as to avoid offense.

CP2. The app shall only make reference to any country in reference to ownership of satellites.

4.7.2 Political Requirements

CP1. The app will not mention any current political standings or issues so as to avoid offense.

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1. The app shall adhere to Compliance laws of Canada, Ontario.

4.8.2 Standards Requirements

LR1. The app shall adhere to Standards laws of Canada, Ontario.

5 Division of Labour

Section 1: Alex

Section 2: Sam

Section 3: Vicky

Section 4 (subsections 4.1-4.4): Matt

Section 4 (subsections 4.5-4.7): Jabrayil

Signatures: