Software Requirements Specification

for

Kepler III

Version 1.0 approved

Prepared by Alexander Schwartz, Sayan Chatterjee, Aaron Morgado, Morgan Black, Harrison Dinius

Embry-Riddle Aeronautical University

February 10, 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document contains details regarding the Kepler III program. The System Requirement Specification (SRS) will be elaborating functional and nonfunctional requirements of Kepler III. The hopes of this document is to explain how Kepler III will function. This document has the ability to completely explain the Kepler III project.

1.2 Document Conventions

This document did not use any special conventions.

1.3 Intended Audience and Reading Suggestions

As mentioned before, this document contains the functionality of the Kepler III program. The primary interest for this document is for the team, the project manager, and the team's prospective professor. After reading the SRS, the reader should be have proficient usability of the program. If there are any questions to remain, a diagram is located in the appendix section for more clarification.

1.4 Product Scope

The Kepler III program is a simulation that has the ability to take in NASA's exoplanet data in order to make accurate models. The main features of the program is to visualize recently discovered planets and then predict possible timelines for each solar systems. The program should also give insight on each exosolar system. In creation of the program, the hope is to encourage people to have interest in space.

1.5 References

No special references were used at the time.

2. Overall Description

2.1 Product Perspective

The program Kepler III interprets data collected by NASA and uses it to create simulations of exosolar systems. These exosolar systems will be accurate to the data, containing calculated gravitational orbits for each planet in the system. The program will also feature a sandbox that allows the user to create their own exosolar system/exoplanets.

2.2 Product Functions

- User may view a simulation of any imported exosolar system.
- User may view a simulation of Earth's solar system.
- User may view individual data of each planet and star in the system.
- User may create an exportable exosolar system.

2.3 User Classes and Characteristics

User
A user is anyone who desires to view simulations of celestial bodies and their solar systems within the program, whether they are imported or created in the sandbox. The user is not required to have any background knowledge in astrophysics to operate the program.

2.4 Operating Environment

Kepler III will be a Java application running on Java Virtual Machine Version 8 or later. In later versions, the program will be able to run on Android Phones.

2.5 Design and Implementation Constraints

- All scripts will be in Java.
- Any diagram will be virtually constructed.

2.6 User Documentation

UD-1: The program shall provide a help menu to allow users to learn how to upload their own data.
UD-2: The program shall provide a help menu to allow users to learn how to create their own exosolar system in sandbox mode.

2.7 Assumptions and Dependencies

DE-1: The user must have access to the NASA kepler data online if they wish to import any recorded exoplanets/exosolar systems.

3. External Interface Requirements

3.1 User Interfaces

UI-1: The program shall permit navigation of menus via mouse clicks and keyboard input
UI-2: The program shall allow users to click the "Import" button to load data into the program

UI-3: The program shall allow users to click the “Earth” button which will display a simulation of the Earth solar system.
UI-4: The program shall allow users to click the “Exit” button to exit the program
UI-5: The program shall allow users to click the “Options” button to edit settings of the program.
UI-6: The program shall display an error message if an error has occurred
UI-7: The program shall be a fixed 720p resolution.
UI-8: The Kepler III display shall be similar for all users.

3.2 Hardware Interfaces

HI-1: The program shall be able to run on a system that has a 32 gigabyte operating system and 2 gigabytes of ram.

3.3 Software Interfaces

SI-1: CSV file import system
SI-2: The program shall be able to import CSV files and parse them for solar system, planet, or star data.
SI-3: The program shall be compatible with Windows and Mac OS x operating systems.

3.4 Communications Interfaces

No communication Interfaces have been identified at this time.

4. System Features

4.1 Home Screen

4.1.1 Description and Priority

When the user opens the program, the user is brought to a main menu screen. This menu consists of 6 selections: import, sandbox, Earth, options, help, and exit. Requirements 1-5 are intended to be completed in the first sprint. Requirements 6-10 are intended to be completed in the second sprint.

4.1.2 Stimulus/Response Sequences

Stimulus: User will run startup file
Response: Program will provide the user with the menu options
Stimulus: User selects import.
Response: Program takes user to selection screen with default and upload.
Stimulus: User selects default
Response: Program displays the first solar system from the included data file.
Stimulus: User selects upload
Response: Program takes user to the screen where they can select their own data file

Stimulus: User selects sandbox
Response: Program take user to sandbox screen as described in 4.4
Stimulus: User selects Earth
Response: Program displays Earth's solar system
Stimulus: User selects options
Response: Program takse user to options menu
Stimulus: User selects help
Response: Program takes user to help screen
Stimulus: User selects exit
Response: Program closes

4.1.3 Functional Requirements

REQ-1: The import button shall bring up a screen with two options: default and upload.
REQ-2: The default option from the import button shall allow the user to display solar systems from a predetermined selection of Kepler data included with the program.
REQ-3: The upload option from the import button shall allow the user to display solar systems from their own data file.
REQ-4: The Earth button shall automatically display Earth's solar system for the user.
REQ-5: The exit button shall close the program.
REQ-6: The sandbox button shall bring up a screen that allows the user to input values to create their own solar system.
REQ-7: The options button shall bring up a screen with options for the user.
REQ-8: The program shall have adjustable volume for the music playing in the background accessible from the options button.
REQ-9: The program shall have a toggleable button to start or stop music accessible from the options button.
REQ-10: The help button shall bring up a help screen.

4.2 Data Upload

4.2.1 Description and Priority

After the user selects the import button and either chooses the default data or uploads their own data. Once chosen, the program will start performing calculations with the data. All requirements for this section are intended to be completed in the first sprint.

4.2.2 Stimulus/Response Sequences

Stimulus: User has an option between the Kepler data or their own data.
Response: System will start running calculations on said data and project the results on another screen.

4.2.3 Functional Requirements

REQ-11: System will import Kepler data from a csv file into an array list
REQ-12: Program shall display a list of exoplanets that the user can select from.
REQ-13: Program shall display a list of exosolar systems that the user can select from.

REQ-14: The program shall take the data from the file and calculate accurate scaling for planet size, star size, and distances between objects.
REQ-15: The program shall be capable of handling file I/O errors.

4.3 Solar System Display

4.3.1 Description and Priority

With either user created data or data taken from a file, the program takes the data and displays the solar system in a GUI. Requirements 16-21 are intended to be completed in the first sprint. Requirements 22-28 are intended to be completed in the second sprint.

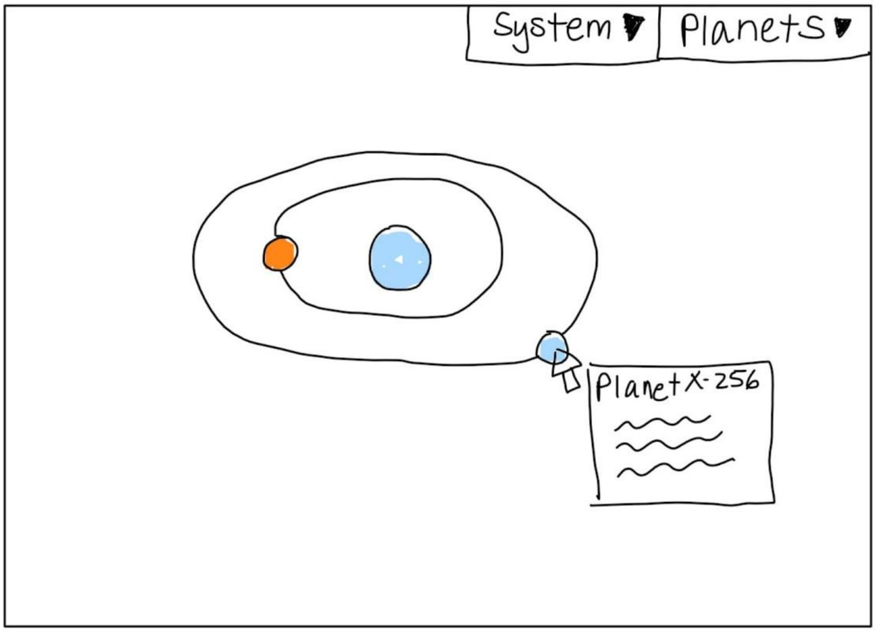
4.3.2 Stimulus/Response Sequences

Stimulus: The user chooses to display a solar system
Response: The program displays the solar system in the GUI

4.3.3 Functional Requirements

REQ-16: The program takes the calculations from REQ-14 to accurately display all planets and stars in the chosen solar system.
REQ-17: The program displays orbital lines for all planets.
REQ-18: The program shall allow the user to click on a planet/star and zoom to have the planet/star fill the entire screen.
REQ-19: The program shall have a drop-down menu of exoplanets contained in the data file that the user can choose from.
REQ-20: The program shall have a drop-down menu of exosolar systems contained in the data file that the user can choose from.
REQ-21: The program shall be color accurate; a planet's color will reflect its physical properties (gas, rock, ice); star colors will be true to data collected.
REQ-22: The program shall have a zoom feature, accessed through mouse wheel scrolling.
REQ-23: The program shall give planet or star specific information when being hovered over with mouse.
REQ-24: The program shall have a gravitational model to display dynamic orbits
REQ-25: The program shall have a user controllable time scale that shows important events in a pop-up window.
REQ-26: The program shall have a button that allows the user to take a screenshot of the current window.
REQ-27: The program shall have it's window fixed to a TBD[1] size.
REQ-28: The program shall have soft jazz music playing in the background by default.

4.3.4 Display Diagram



4.4 Sandbox Function

4.4.1 Description and Priority

When the user chooses the sandbox from the home screen, the program takes the user to a page where they are prompted for information which allows the user to create their own solar system from scratch. All requirements for this section are intended to be completed in the second sprint.

4.4.2 Stimulus/Response Sequences

Stimulus: User enters correct data and clicks calculate
Response: Program takes the user to the Solar System Display displaying the user entered data
Stimulus: User enters incorrect data and clicks calculate
Response: Program displays an error message telling the user what data they have entered incorrectly and returns to the screen to allow the user to enter new data points

4.4.3 Functional Requirements

REQ-29: The program shall have data entry locations for all required data.
REQ-30: The program shall be able to take user entered data and put it in an array list.
REQ-31: The program shall be capable of handling user input errors.

4.5 Program Calculations

4.5.1 Description and Priority

The program will make use of physics equations (namely Newton's Law of Gravitation) to predict unstable solar systems, and perform calculations on any uploaded exosolar system data. Other calculations to draw comparisons between the properties of the currently displayed data and the general Earth Solar System can also be made. All requirements in the section are intended to be completed in the second sprint.

4.5.2 Stimulus/Response Sequences

Stimulus: Program will display loading screen and then draw the new solar system.
Response: User has the option to receive more information from clicking the stars or planets.
Response: Newton's Law of Gravitation will be used to calculate and display the orbits of the planets in the solar system.
Stimulus: If the user selects a planet, a pop up will appear displaying information and comparisons.
Stimulus: If user selects a star, a pop up will appear displaying information and comparisons.
Response: User has the ability to press a button labeled show calculations.
Stimulus: A pop up will appear listing out the calculations for the current solar system and determine any potential collisions.
Stimulus: The user adjusts the time scale while viewing a solar system.
Response: An alert will appear to inform the user if a collision happened during that time frame, and which planets collided.

4.5.3 Functional Requirements

REQ-32: The program shall use Newton's Law of Gravitation to return a value indicating whether or not a collision will occur between planets.
REQ-33: The program shall display a notification in an alert window anytime a significant event (such as a collision between planets) has been predicted to occur.
REQ-34: The program shall use classical orbital mechanics to describe solar system models.
REQ-35: The program shall use Newton's Law of Gravitation for solar system calculations to display planet orbits.
REQ-36: Program will be able to compare current data within the program and the general Earth Solar System.
REQ-37: Program will list temperature and luminosity on current star, and compare it with the general Earth Solar System.

4.6 Data Export

4.5.1 Description and Priority

Upon creation of a new solar system within the program, the data will be exported to a csv document. Screenshots taken within the program will be saved to the desktop. All requirements in this section are intended to be completed in the second sprint.

4.5.2 Stimulus/Response Sequences

Stimulus: User has the option to select a button that saves data after creating a new solar system.
Response: A pop up window will allow the user to select where they want the csv file to be saved (desktop or in another directory).
Stimulus: User takes a screenshot within the program of the current window.
Response: The screenshot is saved to the desktop.

4.5.3 Functional Requirements

REQ-38: The program shall be able to export a custom solar system in the form of a csv document.
REQ-39: The program shall save exported csv documents to the user's folder of preference.
REQ-40: The program shall be able to save taken screenshots to the user's desktop.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The program shall be able to run on a 32 bit 2 gigabyte ram system. This is to ensure that an individual does not need a specialized system in order to run the program.
PE-2: The program shall be able to generate a TBD[2] number of stars and planets without significant lag or crashing.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

No security requirements have been identified.

5.4 Software Quality Attributes

Efficiency 1- The program shall have prompt the user if they are still using the program and close the program after 1 minute of no response to the prompt. This is to keep the program from running in the background for long periods of time and tying up the RAM and the CPU.
Efficiency 2- The program shall allow the user to select a timeout period: 5 minutes, 10 minutes, 30 minutes, 1 hour, or never.
Usability 1- All user inputs, buttons, and menu options shall be clearly labeled to facilitate ease of use.
Usability 2- If a user inputs out of bounds values, the program shall display an alert which states the acceptable range of values to assist the user in completing the task.
Usability 3- All text displayed in the GUI shall be at minimum 10 point font to ensure readability.
Usability 4- The menu shall contain a help page which describes the functions of the program to assist users.
Portability 1- The program shall run on any computer with java.
Reliability 1- The program shall not crash during normal use.

5.5 Business Rules

No business rules have been identified.

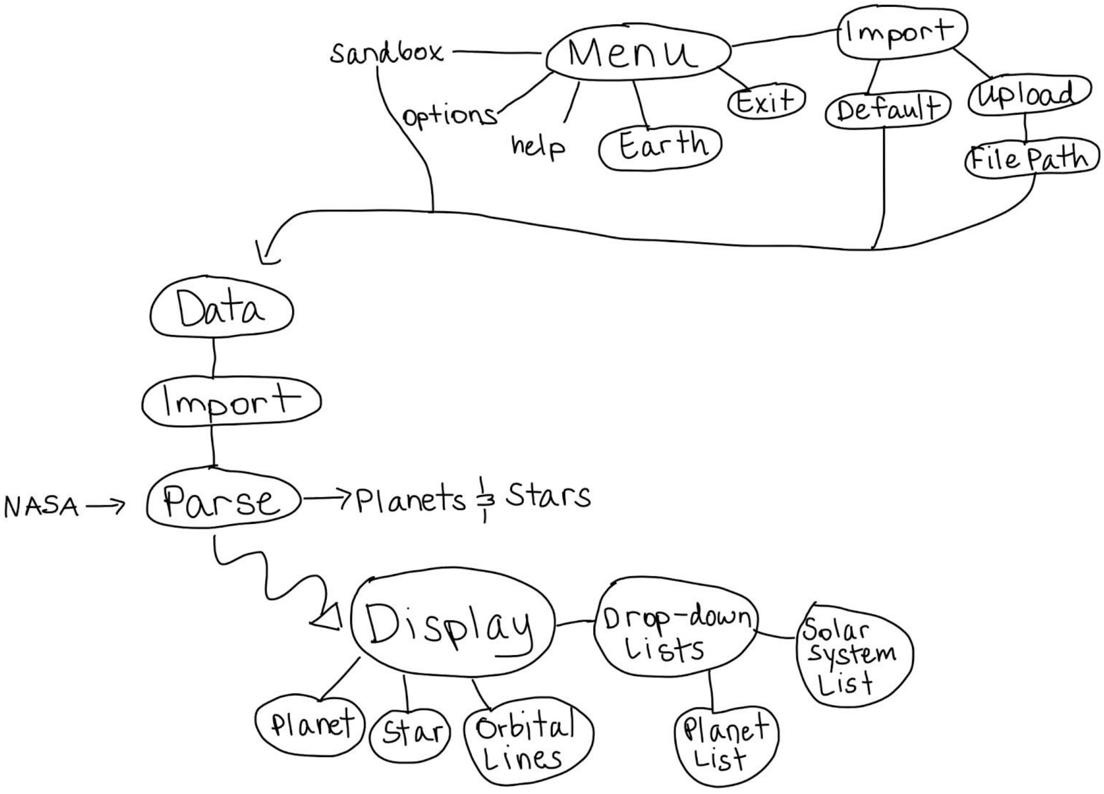
6. Other Requirements

No other requirements have be indentified.

Appendix A: Glossary

No terms are necessary for this document.

Appendix B: Analysis Models



A simplified diagram of how the system function section interact with each other.

Appendix C: To Be Determined List

1: The aspect ratio of the window will be set to a size that will function well on most computers. This needs to be explored, hence why it has not been established yet.
2: The number of stars and planets the system can handle, while it could be a processing power issue, is more accurately a physics problem (what values would be reasonable from a physics standpoint). This is why this has not be established yet.