AstroCal
System Requirements Specification

COSC 470 2022
Spike
Oct. 14, 2022
Version 1.0.0

TABLE CONTENTS

Revision Sheet

Revision Number	Date	Brief summary of changes
1.0.0	Oct. 14, 2022	Baseline draft document

1 Introduction

This is a System Requirements Specification document for the Astronomical Calendar ("AstroCal") software system, an open-source utility for displaying information about the sun and moon such as the times of sunrise, sunset, and eclipses.

1.1 System Purpose

This is the System Requirements Specification Document (SRSD) of the AstoCal project. It documents the completed requirements for the open-source utility. In this SRSD we provide the collection of individual requirements of the operational AstroCal production system from a user's point of view.

1.2 System Scope

The AstroCal software is a tool for displaying the following information:

- A month calendar with current sun and moon phase showing for a particular day, with the ability to change month, year, and day
- Sunrise, sunset, moonrise, and moonset information for a particular day
- Upcoming moon and sun events such as the next solar eclipse or lunar eclipse, and details related to those events such as when they occur

The system is not intended to be used as a graphing tool, such as for visualizing the time of darkness and light for a particular day.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Key Definitions

Crescent Moon:	When only a small arc-shaped section of the visible portion is illuminated, from the perspective of Earth
Day Length:	The time elapsed between beginning and end of the daytime period Elevation..
Gibbous Moon:	When the Moon is more than half full, but not quite fully illuminated, from the perspective of Earth.
Illumination:	The Moon is fully illuminated at the full moon (visible fraction = 1.00). During the crescent phases the visible fraction ranges from 0.00 to 0.50; in the gibbous phases, it is between 0.50 and 1.00
Julian Day:	The number of days that have passed since the initial epoch defined as noon Universal Time (UT) Monday, 1 January 4713 BC in the Julian calendar
Lunar Eclipse:	A lunar eclipse occurs when the Moon moves into the Earth's shadow, from the perspective of Earth.

Solar Eclipse:	A solar eclipse occurs when the Moon passes between Earth and the Sun, from the perspective of Earth
Waning Moon:	When the moon is in the phase in which its amount of illumination is decreasing, from the perspective of Earth
Waxing Moon:	"When the amount of illumination on the Moon is increasing, from the presentive of Earth

1.3.2 Key Acronyms and Abbreviations

AstroCal	Astronomical Calendar
GUI	Graphical User Interface.
Swisseph	Swiss Ephemeris
SRSD	System Requirements Specification

1.4 Key References

Astrodienst AG	https://www.astro.com/ftp/swisseph/doc/swisseph.pdf
	https://www.astro.com/swisseph/swephprg.htm
swisseph (version 20220905)	https://astrorigin.com/pyswisseph/pydoc/index.html

1.5 System Overview

This SRSD is organized as follows:-

- Section 2 gives an overall idea of product.
- Section 3 describes the Non-Functional requirement related to AstroCal.
- Section 4 describe the interfaces of the system.
- Section 5 contain all details that the system and individual, specific requirements.
- Section 6 is the Requirements Traceability Matrix.

2 General System Description

AstroCal is a platform for macOS and Windows desktop computers. It is set to display a calendar and show information related to celestial objects such as the sun and events related to these objects such as solar eclipses.

2.1 System Context

Below is the Context diagram of the system and its boundaries.

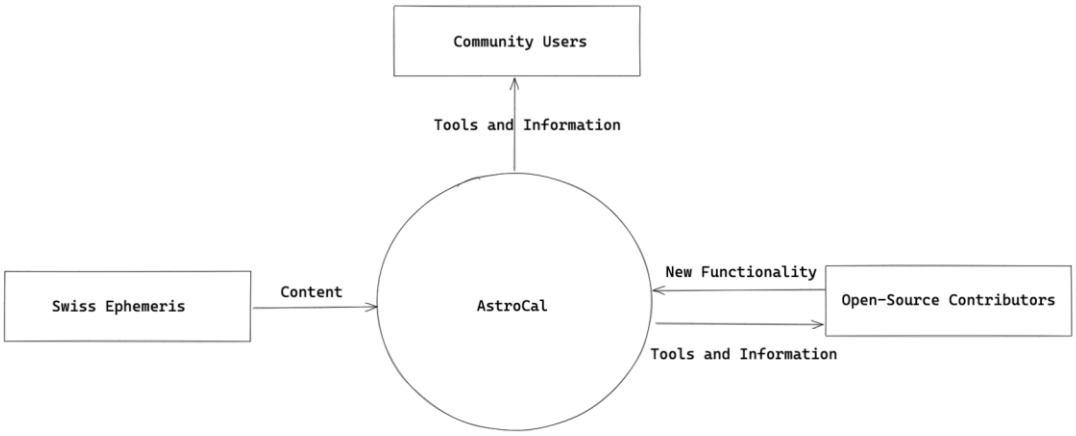


Figure 1: Context Diagram

2.2 System Modes and States

Below is the System Mode and State of the AstroCal system which demonstrates the containers that define control of the functions of the system. The local system waits for user input in the form of a command and replies after analysis with the required information.

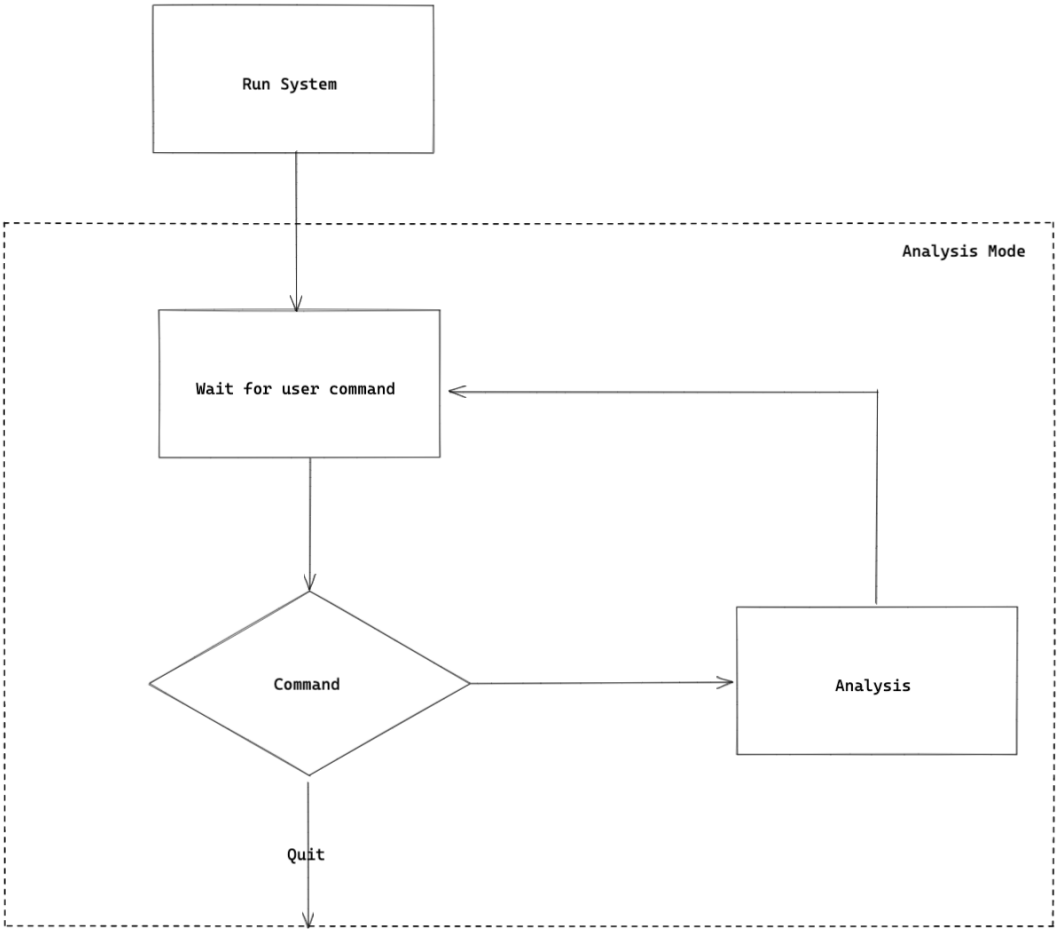


Figure 2: 2.2 System Modes and States

2.3 Configurations

This software should be run on a Mac with macOS version 10.9 or later or a PC with Windows 8 or later. The system was not tested under load, so it is unknown how much available RAM may be required. In addition, the Swisseph library and Kivy framework must be installed on the user's machine in order to run the program correctly.

2.4 Major System Capabilities

AstroCal will be capable of processing multiple command line user input for efficient use of the system. The user will be able to choose the command before entering any of the analysis and results of the AstroCal system. The AstoCal will also be capable of processing analysis from a Graphic Interface, where the commands will be replaced with buttons for selection.

2.5 Major System Conditions

When the user runs AstroCal the system it will launch the command line interface for analysis. User input will generate the results. On the graphical interface there will be menus to navigate to the same calculation functionality.

2.6 Major System Constraints

The system will have constraints in place to prevent the user from input command that are out of range of the system or unexpected user input.

2.7 User Characteristics

Profiles of typical users of this system may include, but is by no means limited to:

- Day-to-day users looking to access sunrise or sunset data, for example, to determine when the best time for stargazing might be
- Travelers who may find it useful to know when they should leave in order to get to their destination before nightfall
- Astronomers who might use data about celestial objects in their practice.

A novice level of understanding is needed to interact with this system, but it is very easy to grasp, and most users should not encounter tremendous difficulty in utilizing the system.

3 System Capabilities, Conditions, and Constraints

3.1 System Performance Characteristics

The AstroCal system ensures that all associated performance requirements are met by providing multiple tabs that handle the different functionality. Three tabs are provided a month tab which provide all calendar related features, a specific day tab with sun and moon daily information and an events tab which provides all information on the sun and moon events such as eclipses.

3.1.1 Load

The system is meant to only handle one request at a time. If a user wants to obtain different information than what is currently displayed, they will need to return to the main calendar page or command line. However, a user may open multiple instances of the application to display different data at the same time.

3.1.2 Stress

The system was not tested under stress from other applications, but it can be assumed that, as a small and relatively non-intensive application, the system should run well under most conditions.

3.1.3 Availability

This system is always available to any user who wishes to make use of it. There are no constraints related to the availability of the system other than maybe a miniscule chance it could run slowly on older hardware.

3.2 System Security and Safety

There are no system access restrictions in place; all parts of the system can be accessed freely by all users. Sanitization of input in the command line is implemented to prevent users from entering anything unexpected and potentially breaking the application.

3.3 Information Technology Management

Several functions within the Swisseph library are used for calculations given different parameters, which is then output as data to the user. AstroCal does not store data of a user’s inputs on their computer; all information used for the output is retrieved from Swisseph during program execution.

3.4 System Operations

The AstroCal application can be run through a command line interface or through a desktop GUI application. The graphical user interface for the desktop application

3.4.1 Internationalization

This software is currently only available in the English language. There was no intention on making the system support multiple languages.

3.4.2 System Maintainability

Check GitHub repository for release information. As this is open-source software, users should be able to download and edit individual files in the repository.

3.5 Policy and Regulation

AstroCal was developed using data from Swiss Ephemeris, a Python library created by AstroDienst. The GUI was made using the Kivy framework. These libraries were used under open-source license.

3.6 System Life Cycle Sustainment

This system will not be maintained by its original developers following development completion. Users who download and edit the software will be free to add new functionality or make changes they see will make usage of the system easier. The lifecycle of the system will last until new operating systems stop supporting Python 3.10.7, the version on which AstroCal was built using.

4 System Interfaces

4.1.1 Desktop GUI Application:

Definition: This interface is used by the customer who wants a visualized experience to access the AstroCal features.
The GUI interface will be simple and consistent, using terminology commonly understood by any users of the system. This will eliminate the need for extensive user training.
Figure 3 is the designers' interruption to the interface requirements.

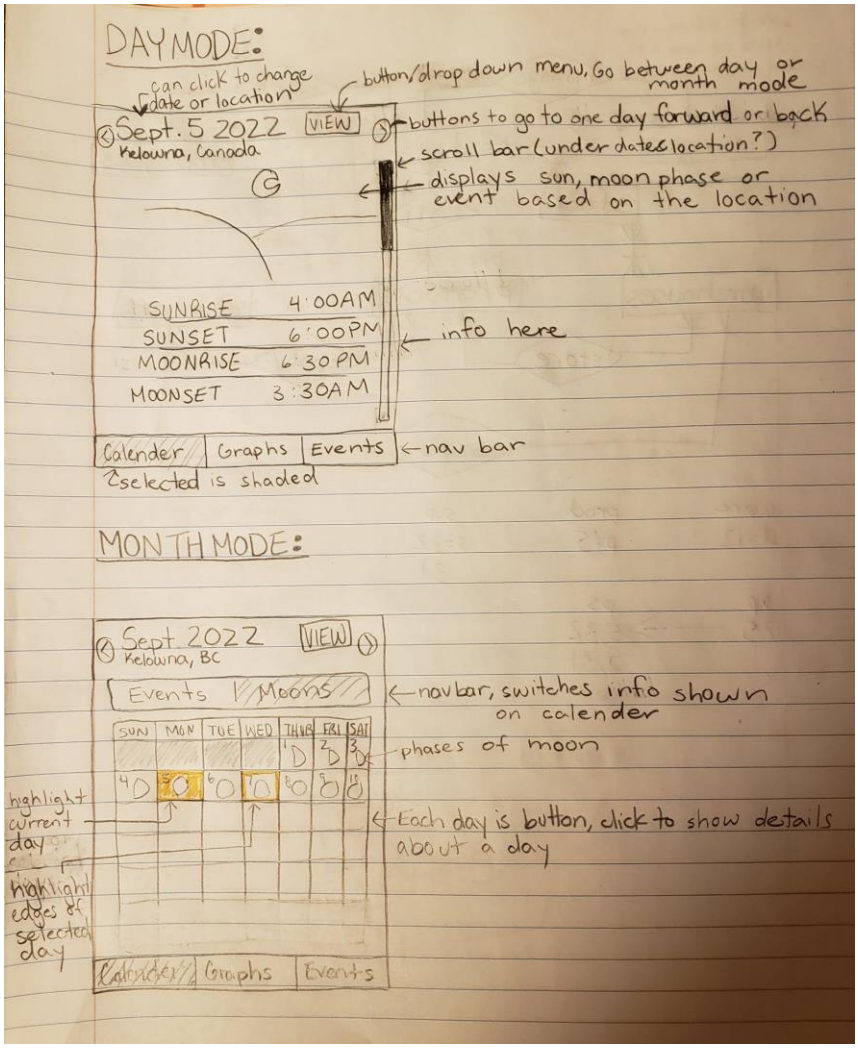


Figure 3 GUI System Interface Requirements

All text (excluding text in the calender that isn't todays date) is white (#FFFFFF)

#081F40
Used for: Header background, Calendar text (Except for text in todays date)

#4575B1
Used for: Menu/Nav bar (Unselected), Days of the week bar (mon-fri), Todays date in calendar

#264772
Used for: Menu/Nav bar (Selected tab)

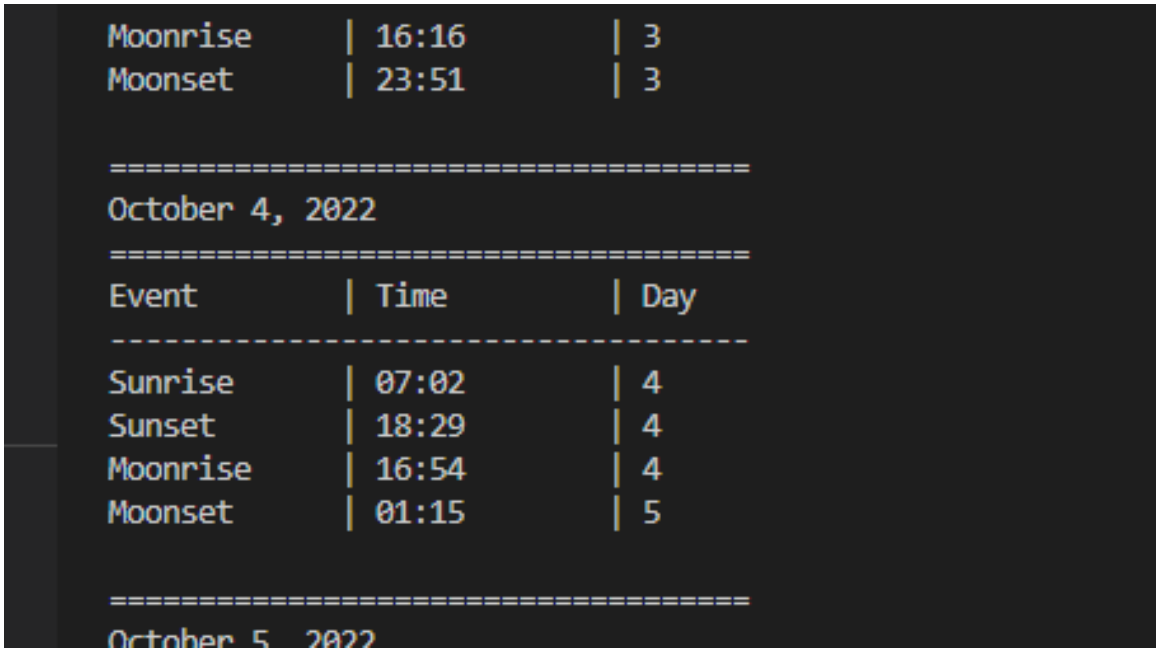
#D6E1EE
Used for: Days of the month that are part of the current month

#B5C2D2
Used for: Days of the month that aren't part of the current month

#102646
Used for: Page background (Lighter middle part)

4.1.2 Command Line Interface

Definition: A text-based user interface used to run programs from the terminal or command prompt of must operating system.
The system will have a simple interface, consistent with industry standard for command line programs. For those experienced this will be a familiar experience. For infrequent users of command line programs may not desire this experience and as such, the GUI will be available.



5 Specific Requirements

Product Requirements that were originally collected from client:
- Needs to be in python
- Needs to be macOS first and Windows (low priority) application
- One app to run on both
- Needs to run on iOS
- Android if time permits
- Need to use nasa swiss ephemeris
- Graphic interface
- Needs to have table/calendar to view moon and sun status
- 4 different graphs displaying the location of moon/sun/other planets in solar system
- Display moon-day, sun-day, rise and set of moon and sun
- Display Solstice
- Lunar eclipse, solar eclipse
- Need to be localized information
- Need to be able to switch location
- Need to be able to filter/search for specific days and moon phases
- Future and Past functionality
- Print/Export specific day information - no longer than one page if pdf format
- Adding extra information into weekday cells by a customer and store parameters into *.csv or formatted text file
- Calendar shall allow using temperature and altitude corrections for sunrise and sunset calculations

6 Requirements Traceability Matrix

Appendix A// Requirements Traceability Matrix

Requirement Name	Priority	Risk	Validation Method(s) *
Needs to be in python	H	L	D
Needs to be macOS first and Windows application	H	L	D
Need to run on mobile	L	M	I

Need to use nasa swiss ephemeris	H	L	D
Graphic interface	M	M	T
Needs to have table/calendar to view sun status	H	L	D
Needs to have table/calendar to view moon status	M	M	T
4 different graphs displaying the location of moon/sun/other planets in solar system	L	H	I
Need to be localized information	L	M	I
Need to be able to switch location	M	M	T
Need to be able to filter/search for specific days and moon phases	L	H	I
Future and Past functionality	H	L	T
Print/Export specific day information	L	H	I
Allow using temperature and altitude corrections	L	M	T

*** Validation Method(s)**

D - Demonstration
T - Test
A - Analysis
I - Inspection

Field Descriptions

1. Requirement Name A short description of the requirement to be satisfied

2.	**Priority**	High (H), Medium (M), Low (L); to be negotiated with customer and remains **fixed** throughout System development lifecycle
3.	**Risk**	High (H), Medium (M), Low (L); determined by technical staff and will change throughout System development lifecycle