Software Design Document

Lunar Pit Patrol

Levi Watlington, Alden Smith, Caden Tedeschi, and Evan Palmisano

Sponsor: Trent Hare

Mentor: Vahid Nikoonejad Fard

09/18/2024 : Version 1.0

Table of Contents

Introduction

With over 570 billion dollars generated in 2023, the global space economy is growing at a noticeable rate creating numerous opportunities for companies. The USGS Astrogeology Department plays a critical role in this industry providing statistical analysis of celestial surfaces. To conduct their research the astrogeologists, use a tool called Craterstats which is a python application for computing statistics related to craters. Currently, the application operates on the command line which requires a variety of arguments for input, and its interface is not intuitive nor easy to use for those who don't have experience with the command line. With our project, we aim to create an interface for Craterstats that is easy to use and intuitive. We plan to do this by using Python libraries for graphing and UI design. Our project hasrequirements to meet this goal. Our first requirement is that the interface (1) must be easy to use by regular people who don't know how to use the command line, (2) show the correct graph and information when used, (3) accept graphs correctly made and allow saving of user-made graphs, and (4) be useable without installing anything. To make our interface easy to use, we have a requirements. These requirements include (1) using buttons and dropdowns with obvious labels that modify a graph, (2) partitioning different setting types so users know how each option will modify a graph, and (3) allowing for easy navigation through the interface. Clicking any button or dropdown menu in our interface will update a command that it will pass to make a graph along with updating the page to view it. Our interface must also (1) have understandable tab labels, (2) allow users to select files with correct graph settings to display and fill settings in, and (3) allow users to use our interface through a website.

Outside of these general requirements, our project must not use any computation-heavy functions. Our project's interface must also always have the correct graph displayed, accurately display graphs, and be responsive. It must also be easy to learn to use and must manage so much that all a user has to do is input a graph's title and both download and upload graph information, and it must be so easy that the only training necessary is with relation to the graphs they want. Currently, our project and its interface meet most (if not all) of these requirements. It only relies on the command-line program for creating graphs, meaning that it does not make graphs on its own. When a user makes any change or clicks any settings button, the graph updated, ensuring its accuracy. Our program also manages most inputs necessary for using Craterstats, which means all a user must do is give a graph a name, and download/upload graph-related data.

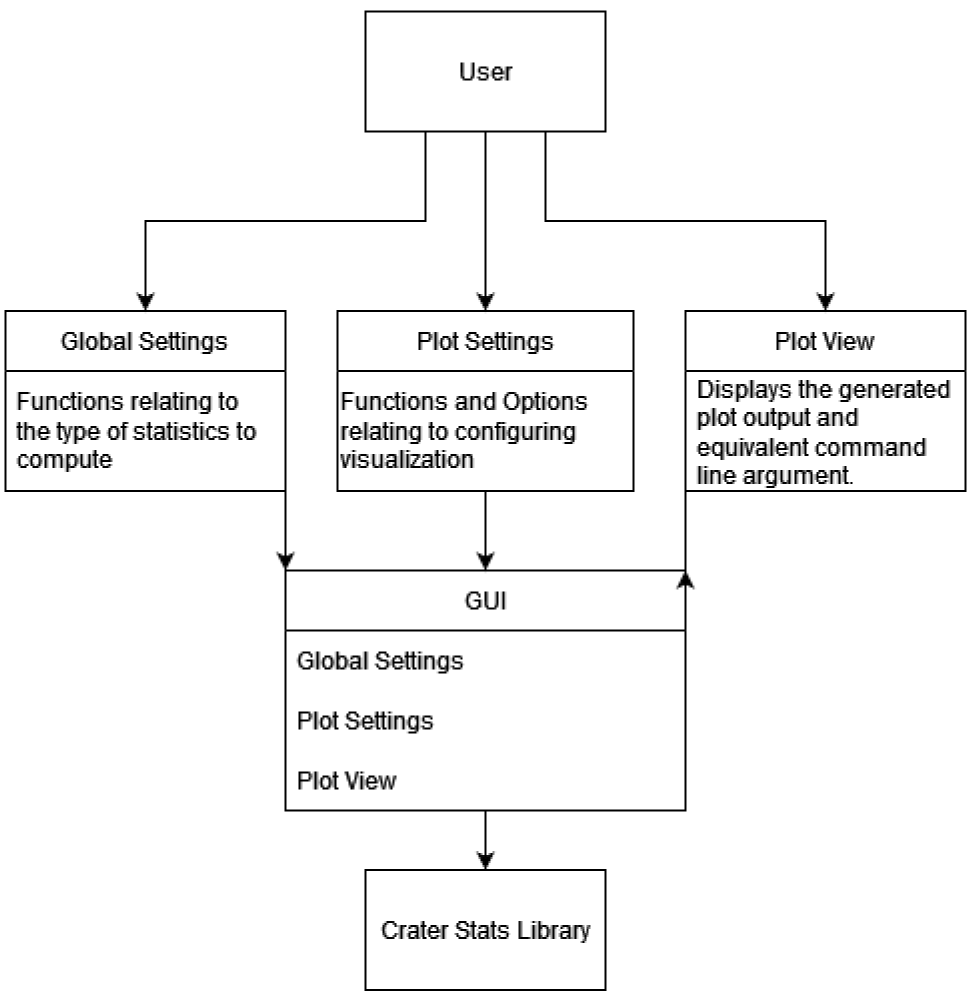
Environmentally, our project has some more requirements. For one, our program must have everything the Craterstats program has. This means our project must have all the packages, libraries, and other files, requiring up to 1.4 gigabytes of storage. Our project must also have easily maintainable frameworks and languages. Our project must also be usable from a website and must be able to upload and download files holding graph settings (both to and from our program).

Implementation Overview

We are building a sophisticated Python user interface for the Crater Stats Command Line Interface (CLI) application that flows intuitively, responds quickly, and can be used universally across operating systems, including mobile. We use an iterative design approach to our implementation, creating frequent prototypes, assessing each iteration, analyzing our results, and refining our design. The Crater Stats user interface will be made in Python for its universal operating system capabilities. This is a major design decision because we require the application to work across all major operating systems with little to no change for each. Python also has access to several user interface modules, such as Flet. The Python Flet module meets all our needs with the addition of being able to build web applications and mobile applications. Our system has a stretch goal of having a web application based on the same executable. The most important package for our application to work is the Craterstats CLI application. This package holds the functions we need to connect our user interface to generate the statistics shown inside our user interface application. Our implementation strategy promises seamless integration between the Crater Stats CLI and Crater Stats User Interface, providing a robust and efficient tool for users. As we continue to refine our design, we remain focused on delivering a cross-platform solution, including our stretch goal into web and mobile applications.

Architectural Overview

The architecture of a Graphical User Interface (GUI) application plays a pivotal role in determining its usability, scalability, and maintainability. This section outlines the GUI application's design principles and structural components, emphasizing how the various layers and modules interact to deliver a cohesive user experience. The Crater Stats GUI consists of one main driving application. It is written in Python and uses the libraries Flet for the GUI and Craterstats for the statistical computation. This file contains three different pages: Global Settings, Plot Settings, and a Plot View page. The pages are tabbed at the top of the application, just below the toolbar. Just like a browser, users can switch between the tabs as the application runs. Each of these pages contains their respective functions and operations relevant to the computation and generation of crater statistics. The pages consist of a variety of inputs such as dropdown menus, radio buttons, file browsing, and checkboxes.



The flow of the interaction between the user and the application should be a linear sequence of actions. When a user wants to generate a plot, they interact with the Plot Settings page. This page is where the user can configure settings related to the plot generation. Titles, sizing, color, display age, and diameter range are all options that can be optioned. These settings are all relevant to the visualization of crater statistics. The Global Settings page focuses more on the type of computation used for the data analysis. Specifically, the cumulative and differential options build relative chronology. The relative option compares quantities to one another without absolute numerical values. The Hartman option estimates age based on crater density. Chronology relates to dating surfaces based on crater presence. Finally, Rate analyzes the frequency at which craters form on the surface.

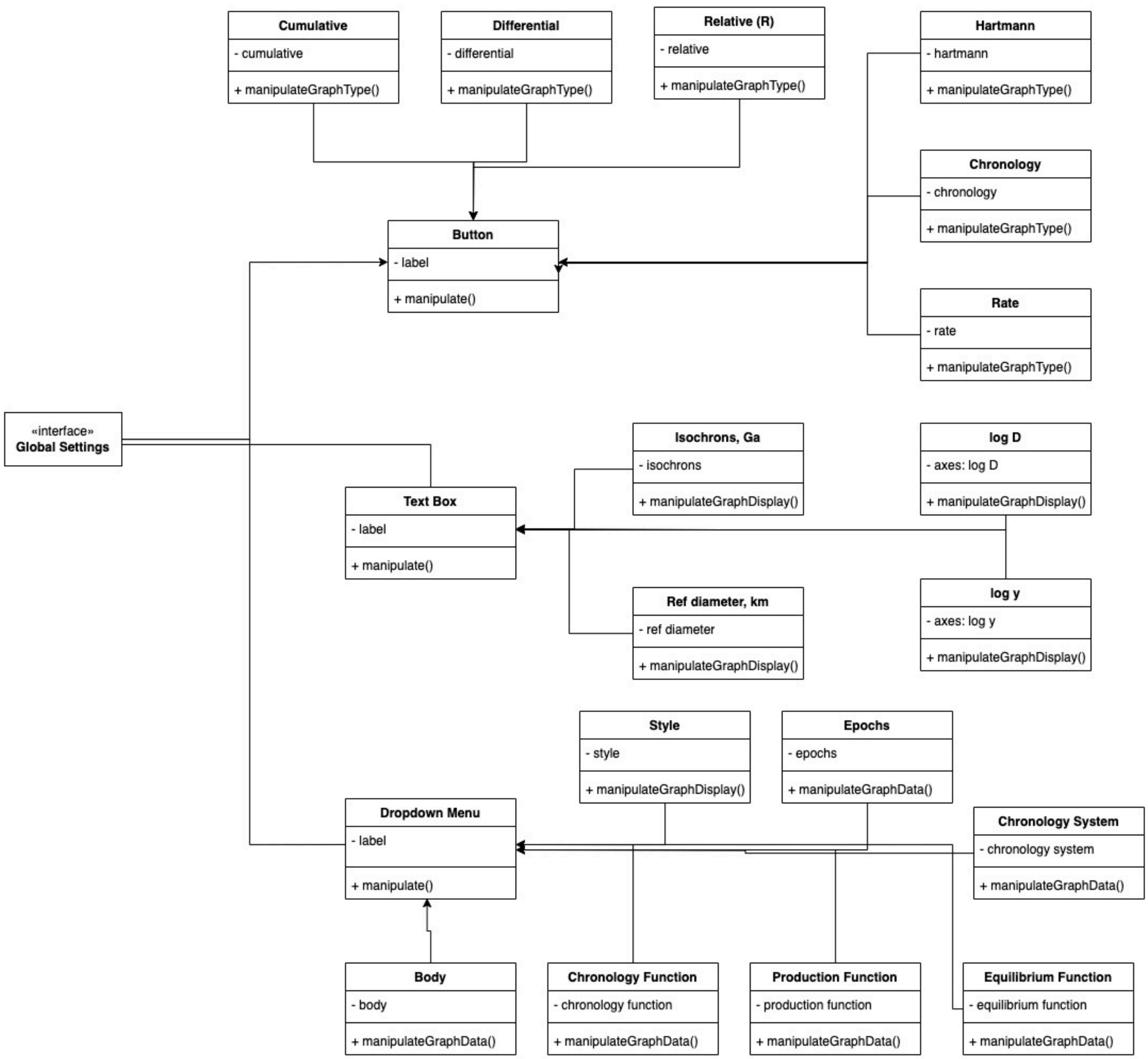
Once the user has configured the settings in the Global Settings and Plot Settings pages, they are taken to the Plot View page, where the selected parameters are applied to generate and display the final visualization. This page acts as the output interface, providing the user with a graphical representation of crater statistics based on the selected options. The Plot View page not only allows users to view the results but also provides interactive capabilities, such as zooming, saving plots, and adjusting display settings. By separating concerns across different pages and leveraging Flet for its responsive and dynamic UI features, the Crater Stats GUI ensures an intuitive workflow for users. This modular approach enhances usability and maintainability, making it easier to introduce future updates or additional features without disrupting the existing system. Furthermore, the integration of Craterstats for computational analysis ensures accurate and reliable statistical outputs, reinforcing the application's core functionality.

Together, these components form a cohesive architecture, facilitating both novice and experienced users in exploring and analyzing crater data effectively. The design promotes user engagement, allowing users to manipulate settings and observe immediate changes in the generated plots. This interactive feedback loop enhances the user experience of crater statistics, making the Crater Stats GUI a valuable tool for scientists and researchers in making informed decisions based on visualized data.

Module and Interface Descriptions

Global Settings

The project we are working on has three main modules: Global Settings, Plot Settings, and Plot. The Global Settings module controls all the data that will be used in the graph. The interface for the Global Settings has different styles of graphs available to the users., There are five different styles of graphs listed at the top for our users to select based on what they need the graph for. Directly underneath the graph-style buttons is a dropdown menu that allows users to choose the celestial body on which they would like the graph to show information. Below the celestial bodies option is the dropdown that controls the chronology system, the method used to determine the order



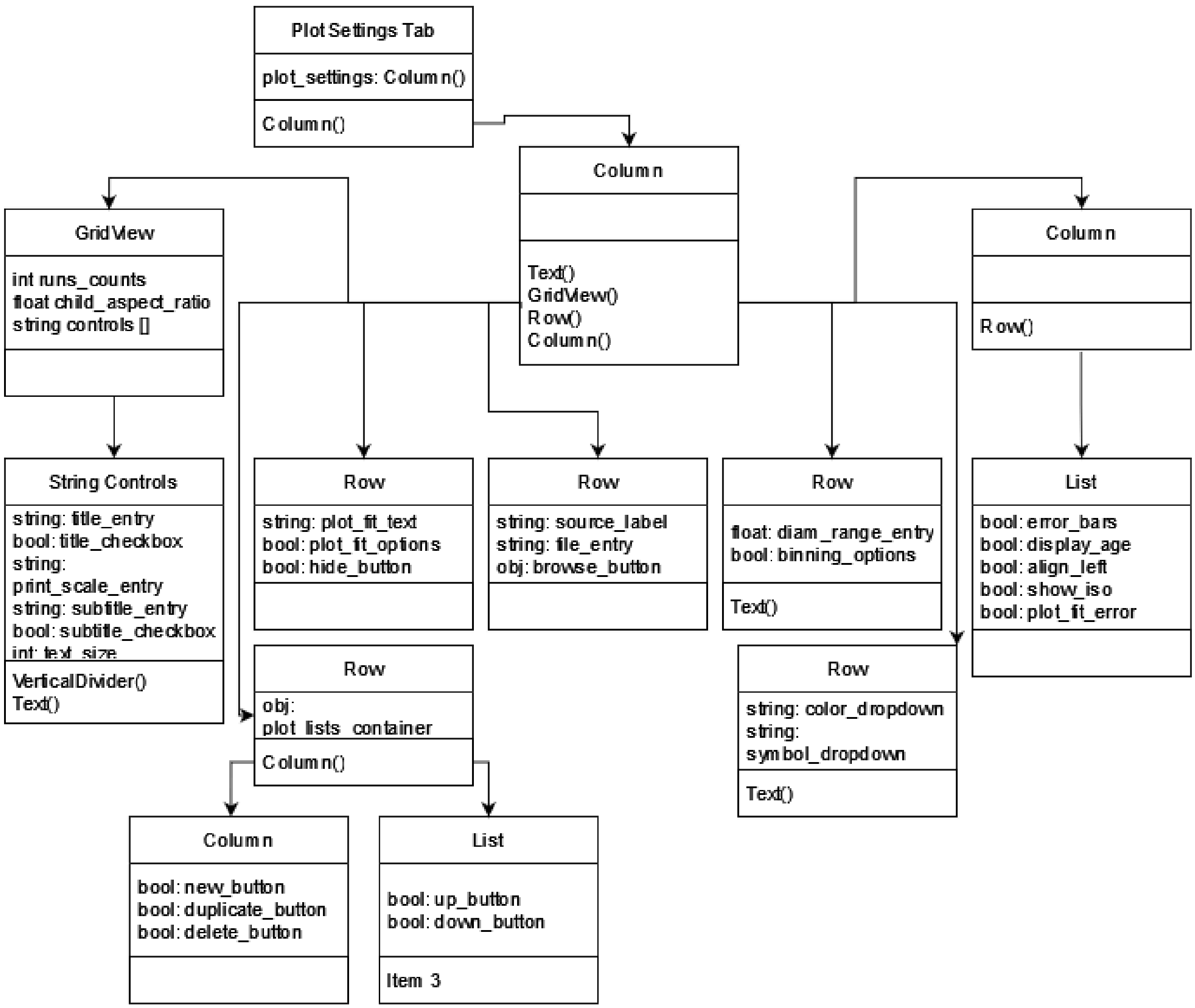
of specific things or events, used by the graph. Under that dropdown, two more dropdown menus that can't be changed by the user are labeled Chronology Function and Production Function. These two options are important for the user to know, so they know all the information the graph needs to be built, but they are dependent on the Chronology System that the graph is using. When the user chooses what Chronology System they want to use, the Chronology Function and Production Function dropdowns both automatically change to match the system. The Epochs dropdown box, right below the Production Function dropdown, is dependent on which celestial body the user selected, but unlike the previous dependent dropdown menus, the user can interact with this one. When a user selects the celestial body they want, the option of whether to display the epochs of that specific celestial body will appear in the dropdown menu. Underneath the Epochs dropdown menu, we have the Equilibrium Functions dropdown menu, which sets the upper limit for the lines of the graph. The series of checkboxes below the Equilibrium Function dropdown menu is used to toggle extra displays for the graph. The Isochron checkbox will display a line connecting events that happened at the same time, and a certain length of time is taken from the textbox right next to it so that the user can input what time they want. The Data checkbox will display some graph information in the top right corner of the graph. The Fit checkbox will display the line of best fit on the graph. The Functions checkbox will display the functions used in the graph. The 3sf checkbox will display the number of significant figures in our graph. The Randomness checkbox will display a graph showing the randomness above the regular graph we display. The μ(mu) notation checkbox will display the numbers in the graph in μ notation. To the right of the μ notation checkbox is a text box for Ref diameter, km where the user can input the diameter they want for their craters in kilometers. The Axes log D and log y text boxes below the checkboxes allow the users to manipulate how the axes are being displayed. The Auto button next to the text boxes just fills in a default value into the boxes. The Style dropdown menu at the bottom of the interface changes the style of the plot.

Plot Settings

As mentioned in the Architectural Overview, the Plot Settings page is where users can configure settings related to plot generation. The page's main purpose is to provide customizability when generating plots within the application. Customizing plot output for crater statistics is essential for several reasons. Crater statistics provide complex data, such as assessing age, density, or formation frequency. Customizing plots allows users to adjust visual aspects such as titles and axis labels, helping them better understand the trends relevant to their analysis. The number of variables and their quantity can vary in statistical analyses, such as diameter ranges, cumulative vs. differential data, or different chronology models. Customization allows users to focus on the parameters most relevant to their study, making the plot more direct and relevant to their research goals. Additionally, researchers may need to compare craters across different regions, periods, or datasets. By adjusting visual settings, users can customize plots to ensure consistency or highlight differences, making comparisons more effective.

Breaking down the Plot Settings page by features and functions, the page can be split horizontally into two halves. The top half focuses on the plot title and sizing. Both the title and subtitle can be set for the graph. Text boxes are provided for these input options, with checkboxes next to them to indicate if they are desired. On the right side of the page, text boxes for the plot size and text size are available. The plot size is configured by "width x height" in centimeters, while the text is sized using "pt" values. Below the title and subtitle inputs is a small window with buttons to the right. This window allows users to view their selected title placement on the graph. The buttons give options such as New, Duplicate, and Delete, along with Up and Down buttons for positioning the title on the plot.

The bottom half of the page consists of file options and more specific display settings. Just below the plot window, users can select their plot fit options from a dropdown list. They can choose from "crater count," "cumulative fit," "differential fit," "Poisson pdf," and "Poisson buffer pdf." Next to the dropdown list, a boolean checkbox allows the user to choose whether or not the plot will be visible. Below these options, users can select a custom source file by browsing their computer directory. A plot will then be generated from the imported data. A diameter range can also be configured, allowing users to decide the desired range of included craters. Users can set up a bin for logging data, which enables the application to record a log of operations for future reference. Symbols and their colors can be changed using their respective dropdown lists, with 12 different colors and 13 different shapes available for selection. The default options are “black”



and “square.” The row below contains five checkboxes, allowing users to decide if they want to feature certain statistics. These checkboxes enable users to include error bars, display age, align age to the left, show isochrons, and plot fit within the plot generation.

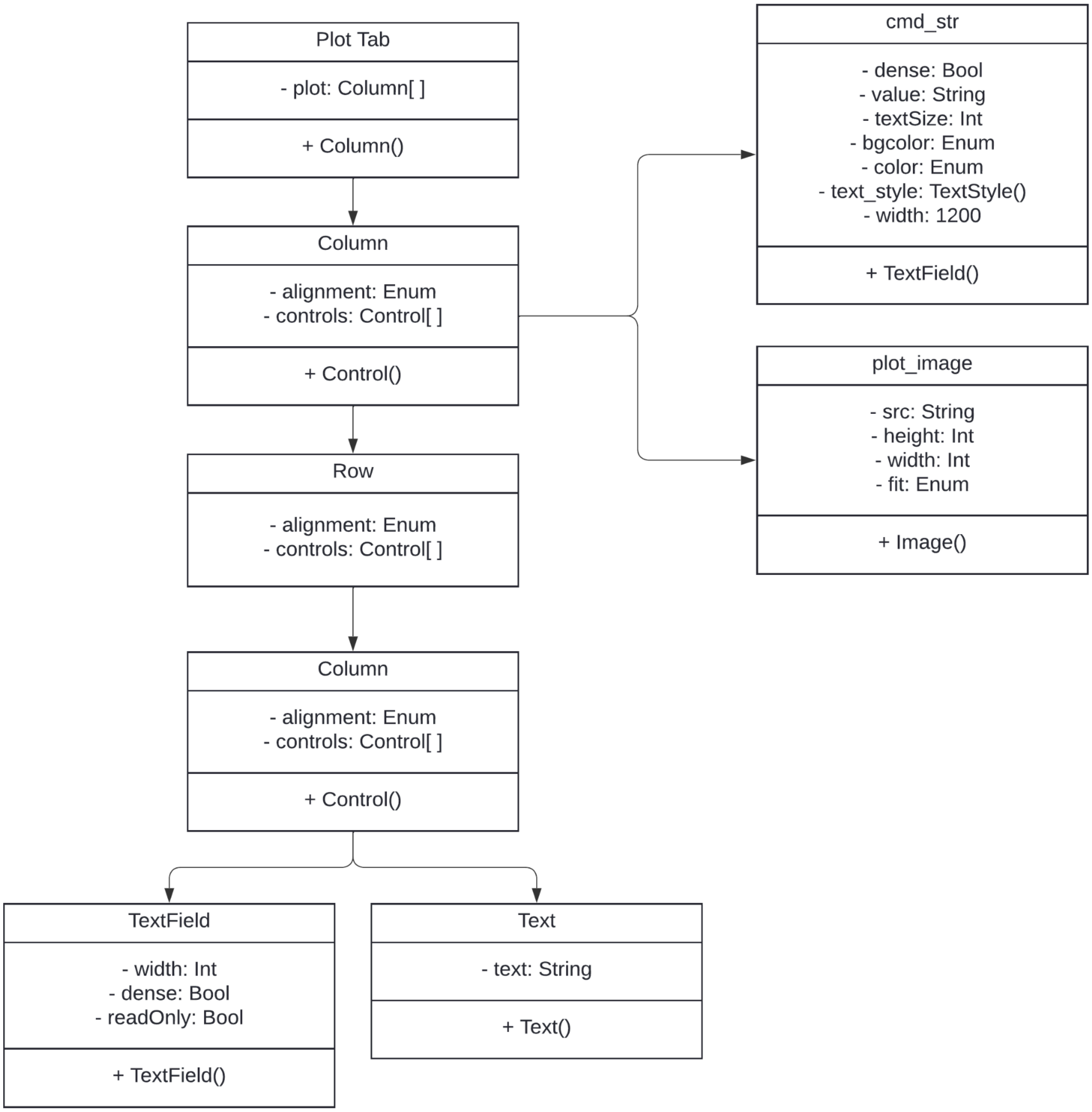
By utilizing the Plot Settings page, users can customize various aspects of their plots to better suit their specific analytical needs. This customizability enables users to fine-tune the visual representation and statistical parameters of their data, ensuring that the plots are informative and aligned with their research objectives.

Plot View

The plot module plays debatably the most important role in our application, allowing for the primary interface for visualizing and interpreting the data generated by the program. It is responsible for displaying two key pieces of information that are essential for users to analyze and understand the results of their crater mapping and analysis. The first and most important element displayed by the plot module is the graph itself. This graph serves as a dynamic representation of the statistical data processed by the application. It is constantly updated in real-time whenever a user makes adjustments to the settings, either through the Global Settings module or the Plot Settings module. These updates ensure that the graph remains an accurate and up-to-date representation of the selected statistics, allowing users to see the impact of their changes.

The graph aggregates data from the Global Settings and Plot Settings, combining various statistical measures into a single, cohesive visual representation. This often results in a complex graph with multiple data points and curves that may overlap, presenting a comprehensive view of the crater density, cumulative crater frequency, and other critical metrics. This detailed visualization is invaluable for researchers and scientists who need to analyze trends and patterns within their data. However, the complexity of the graph can sometimes make it challenging to interpret, especially when there is a large amount of overlapping data. To assist users in navigating this complexity, the plot module includes several features designed to enhance the clarity and usability of the graph.

In addition to the graph, the plot module also displays a secondary object of importance. This secondary object is the command line command required to replicate the current graph in the CLI program. This command is generated automatically by the application and is displayed prominently within the module. It provides users with a precise, reproducible command that can be used to recreate the exact graph



configuration outside of the interface, ensuring that the results can be replicated and verified in other environments or by other users. This feature is handy for users who need to share their findings with colleagues or incorporate the graph into external reports or publications.

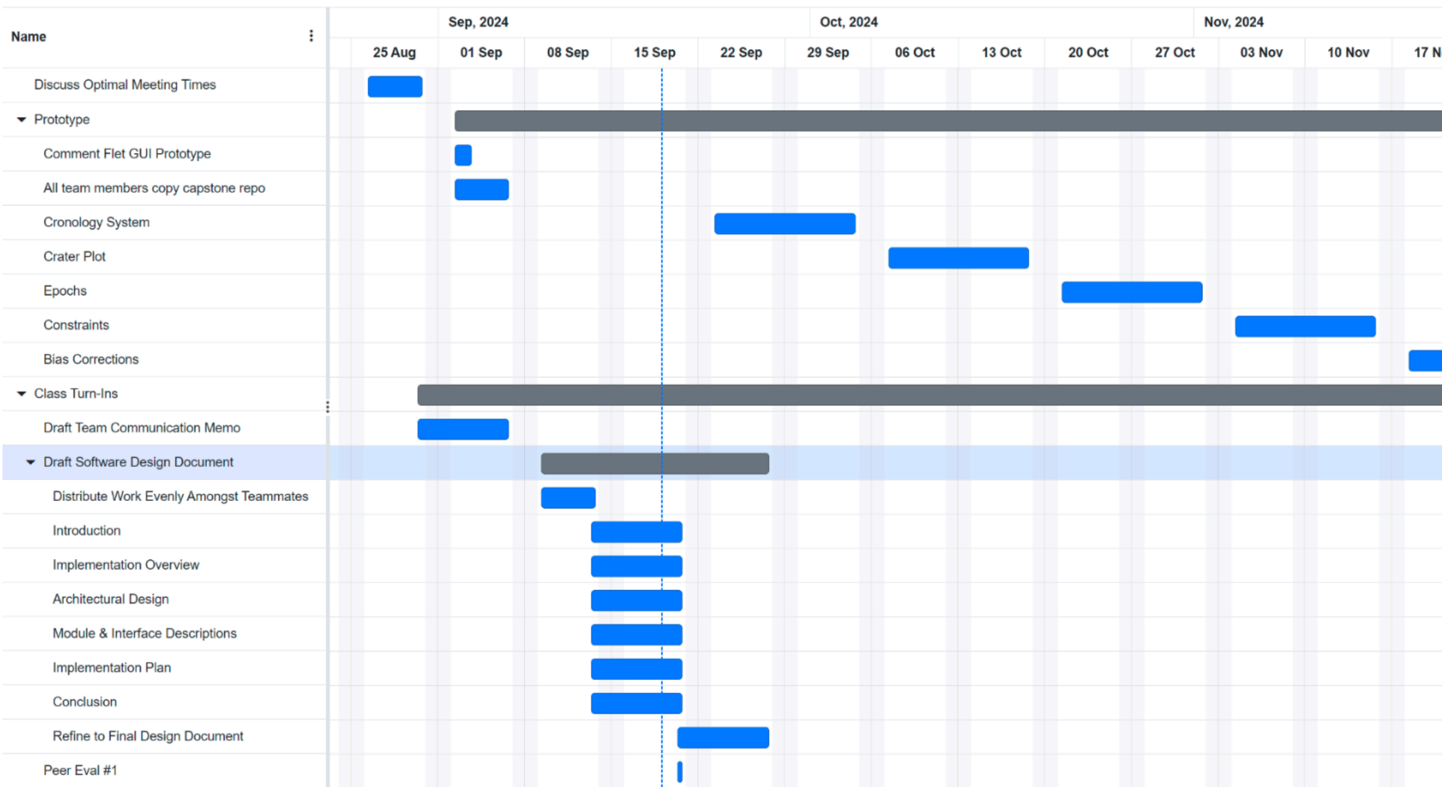
Furthermore, the plot module includes an info bar at the top of the interface that provides real-time, context-specific information as users interact with the graph. As the cursor moves over different points, the info bar dynamically updates to show precise values and metrics, eliminating the need to manually interpret the graph's axes and scales. Key information displayed includes the crater diameter, binning distance, and y-value (crater density in no./km^2). It also provides the age of the crater's cumulative crater frequency (N(1)), and the crater production function constant (a0), which indicates the rate of crater formation over time. This interactive feature makes it easier for users to interpret the graph and gain insights into the underlying data.

Overall, the plot module is significantly important for the application, which offers a platform for data visualization and analysis. The module's comprehensive feature set ensures that users have all the necessary tools at their disposal to conduct thorough and accurate crater studies, making it an essential component.

Implementation Plan

We have an in-depth implementation timeline for our project that covers a schedule that focuses on the implementation of each component. A Gantt chart has been created to cover our timeline for the estimate of completion in a graphical display. The chart shows a snippet of our planned objectives. It has been zoomed in to display the implementation portion.

Our application uses a Python package named Flet. This is a redesign choice that happened over the Summer of 2024 and is reflected in our Gantt chart on the first of September. The next five major modules all come from the same package but are each individually important. These modules represent important functions needed to generate the statistics that our plot outputs. The first one is the Chronology System. In this system, we will start the hookup at the delivery of this draft of the software design document. This will represent the date ranges that users want from the statistics. The next system is the creator plot system. This will be the result graph that is output. This is an extraordinary accomplishment because it allows us to test more accurately that the inputs from our user interface are outputting the correct graph. This will accelerate the implementation of the next few major systems. The third system on the chart is the Epocs input. This will graph major events in any crater of a celestial body. This work will start immediately after the end of the crater plot system. The fourth system to implement is the constraints. We want the ability to limit the output of all input. This includes Date, Size, Depth, Epocs, contents, and hardness of the ground. This work will start immediately after the Epocs portion is implemented. The final system to be implemented and the completion of our software is Bias correction. This work will start immediately after the constraints are completed and will finish by mid-November. This should leave the team enough time to correct any mistakes or make small final adjustments before the final delivery of the software.



Conclusion

In conclusion, we have successfully planned a user-friendly GUI application that meets all key requirements: providing real-time graph displays, minimizing complexities when operating, and providing visualization of computations. These features ensure that anyone researching craters can engage with the application, regardless of their computer proficiency. For seasoned scientists, the application offers a convenient way to generate graphs related to the astro-geological features of various celestial bodies While some astrogeologists may be comfortable navigating command-line interfaces, not all possess this expertise. Our GUI simplifies the graphing process, enabling a broader range of astrogeologists to create and utilize both original and pre-made graphs with just a few clicks. This accessibility not only enhances their workflow but also provides valuable visual aids to support their research and analyses.