Sravya Kuchipudi
Professor Kaisler, Hasanov
CSCI 6917
26 June 2023

Project Report 2: Building a GUI for Unix Commands Execution on MacOS

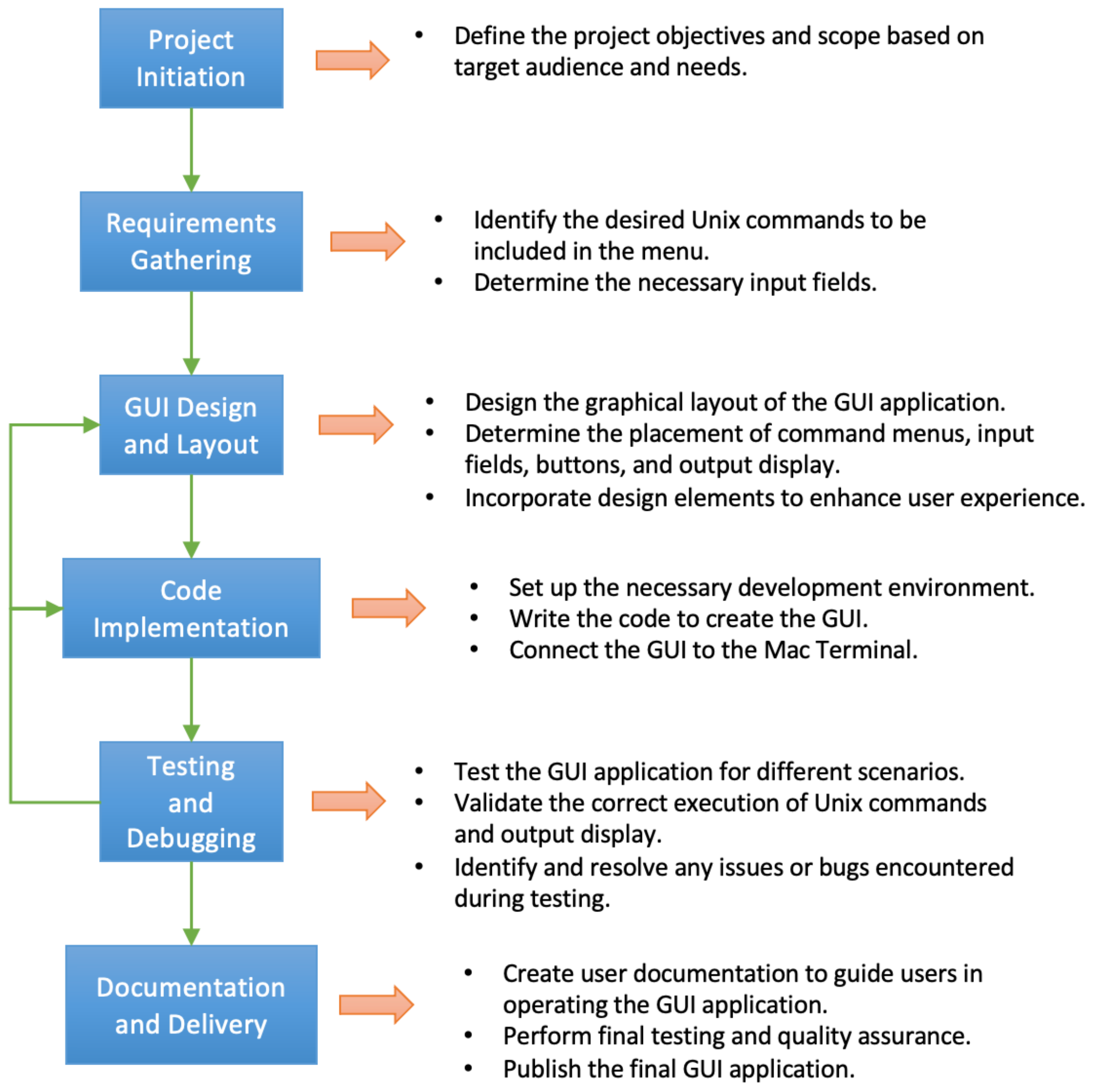
The purpose of this report is to outline the strategy, plan, and data collection approach for the development of a graphical user interface (GUI) that allows users to execute Unix commands on the Mac Terminal using menus and input templates. The project aims to enhance the user experience by providing a user-friendly interface for command execution. This report provides insights into the selected research strategy, the plan, the data collection approach, and the data cleansing techniques employed.

Strategy for Research

This project can be considered more qualitative than quantitative since its focus is on creating a graphical user interface (GUI) to facilitate user interaction with Unix commands, rather than performing extensive data analysis or numerical computations. The emphasis is on improving the user experience and providing a user-friendly interface for executing commands. The qualitative approach allows for in-depth exploration of user needs, preferences, and feedback to inform the design and development process. The research strategy emphasizes understanding user requirements, gathering feedback, and iteratively refining the GUI application. So as discussed in class, the questions trying to be answered in this project are How does it work? How to make it better? since the main aspect is the MacOS Terminal and how the GUI would improve novice user utilization.

Plan

The research project plan is presented in the following flow diagram, outlining the key steps involved in developing the GUI application. The blue block arrows indicate more details about each step while the thin orange arrows indicate the flow of work with loops included where relevant.


Project Initiation
• Define the project objectives and scope based on target audience and needs.
Requirements Gathering
• Identify the desired Unix commands to be included in the menu.
• Determine the necessary input fields.
GUI Design and Layout
• Design the graphical layout of the GUI application.
• Determine the placement of command menus, input fields, buttons, and output display.
• Incorporate design elements to enhance user experience.
Code Implementation
• Set up the necessary development environment.
• Write the code to create the GUI.
• Connect the GUI to the Mac Terminal.
Testing and Debugging
• Test the GUI application for different scenarios.
• Validate the correct execution of Unix commands and output display.
• Identify and resolve any issues or bugs encountered during testing.
Documentation and Delivery
• Create user documentation to guide users in operating the GUI application.
• Perform final testing and quality assurance.
• Publish the final GUI application.

Strategy for the Data Collection

The data collection approach for this project involves a combination of user requirements gathering, referencing external resources, and creating templates based on command data. There are a few key aspects of the data collection approach. Figuring out the user requirements is mainly done using research into what is the most necessary aspects of the Terminal. This will provide insights into the desired commands, input templates, and overall functionality of the GUI application. To begin data collection, there needs to be information gathered from reputable sources, such as Unix command documentation[1], to understand the structure, arguments, and usage of each command. This will help in creating accurate command templates and populating them with the required data. Then "templates" for each command can be created based on the gathered data. These templates will include placeholders for arguments that users will fill in using the GUI menus and input fields. The templates will serve as a guide for the user to enter the relevant information.

[1] Available in GitHub repository in the papers/ folder.

Data Cleansing

Data cleansing is crucial to ensure the accuracy and integrity of user input. There are a few different steps that are necessary to validate and cleanse the user-entered data. There needs to be validation checks done on user-entered data to ensure it meets the requirements specified by the command templates. This may involve checking data types, length restrictions, and allowable values. Users will then be made aware of any errors or discrepancies in their input so they can fix it before trying again. This is why users need to be made aware of any errors or inconsistencies in their input by displaying appropriate error messages. As part of the cleansing process, clear instructions or suggestions will be given for fixing errors. This will help users correct their input and ensure the successful execution of the commands. Based on available time, further work can be done with data cleansing through quality checks. This can be done by generating log entries for every command executed, including information such as the executed command, arguments, and execution results. This will assist in identifying any data quality issues and facilitating debugging if errors occur. So, this could be a more long-term way to regularly reviewing the data entered by users to monitor the issues.

There has been some work done so far in the error checking, since users have the command info panel to refer to when they get an error for a certain input, however, additional work needs to be done to individualize the error response messages to improve user experience. There will be more to discuss in terms of discovery and issues closer to the end of the work due to the way the project is set up, but one of the main things that has been noticed so far has been there are not as many ways to have errors in the Terminal as one would think when the number of commands has been limited. The majority come from improper input due to either user privileges, spelling, the file isn’t in that directory, or the arguments were formatted incorrectly. While there are numerous ways to make these mistakes, when the more harmful commands are removed or at least proper information about their use is given prior to their use, the harm can be mitigated,