

Sravya Kuchipudi  
Professor Kaisler, Hasanov  
CSCI 6917  
03 July 2023

### Project Report 3: Building a GUI for Unix Commands Execution on MacOS

The purpose of this report is to outline the measurement strategies, statistical analysis techniques, and data visualization approaches employed in the development of a graphical user interface (GUI) for executing Unix commands. The project focuses on qualitative research methods, including parts dedicated to error handling and user interface aspects. This report will describe the measurement strategies utilized for identifying and remedying input or usage errors, present the used statistical analysis techniques, and provide visualizations to illustrate the GUI's output and user interface elements.

#### Measurement Strategies

In this project, measurement strategies are used to record errors encountered by users and identify opportunities for improvement, such as input validation and detailed error notification. Errors encountered during command execution within the GUI application are recorded to gather insights into the types and frequency of errors. The error recording process is simply collecting errors as they occur since using a specific logging mechanism would likely generate unnecessary work and an excess of data that would need to be managed. If any errors seem to occur more frequently than others than they are noted. This allows for any necessary targeted handling or code modification to address the issue. To enhance error handling within the GUI application, the first catch is using input validation. With proper input validation mechanisms in place, the system can ensure that user-entered data meets the required format, type, and constraints. This involves checking user inputs against predefined validation rules, such as data type validation, range validation, and format validation. By validating inputs before command execution, potential errors can be minimized.

Also providing clear and detailed error notifications to users when errors occur is essential to alert them of any issues. Instead of generic error messages, frequently reoccurring errors get specific information outputted about the error type and possible reasons for its occurrence. By including detailed error notifications, it helps novice users understand the cause of the error and guide them towards rectifying it effectively. This means that the included guidance or suggestions within error messages need to be to assist users in resolving errors without being too specific to miss other potential issues. For this, in addition to the tips, there might be links to relevant documentation or resources if the error is not able to be fixed by the provided help. By offering guidance, users are empowered to overcome errors more efficiently, leading to a smoother and more satisfying user experience.

#### Statistical Analysis

In this project, a traditional statistical analysis approach may not be suitable due to the nature of the project and the limited number of users. The only user of the GUI is primarily testing the GUI application to fix code issues, therefore, conducting a statistical analysis based on user data would likely yield skewed results and may not provide meaningful insights. The project's current phase involves testing and development, with a single user actively using the GUI application, so the small sample size and lack of diverse user data make it challenging to test statistical analysis techniques that rely on larger sample sizes for reliable results. The focus of this project is on identifying and resolving errors, improving user experience, and refining the GUI application. These aspects primarily involve qualitative evaluation, and iterative development. Traditional statistical analysis methods would not adequately capture the nuances of qualitative aspects and may not provide valuable insights for the project's objectives.

## Visualization of the Data

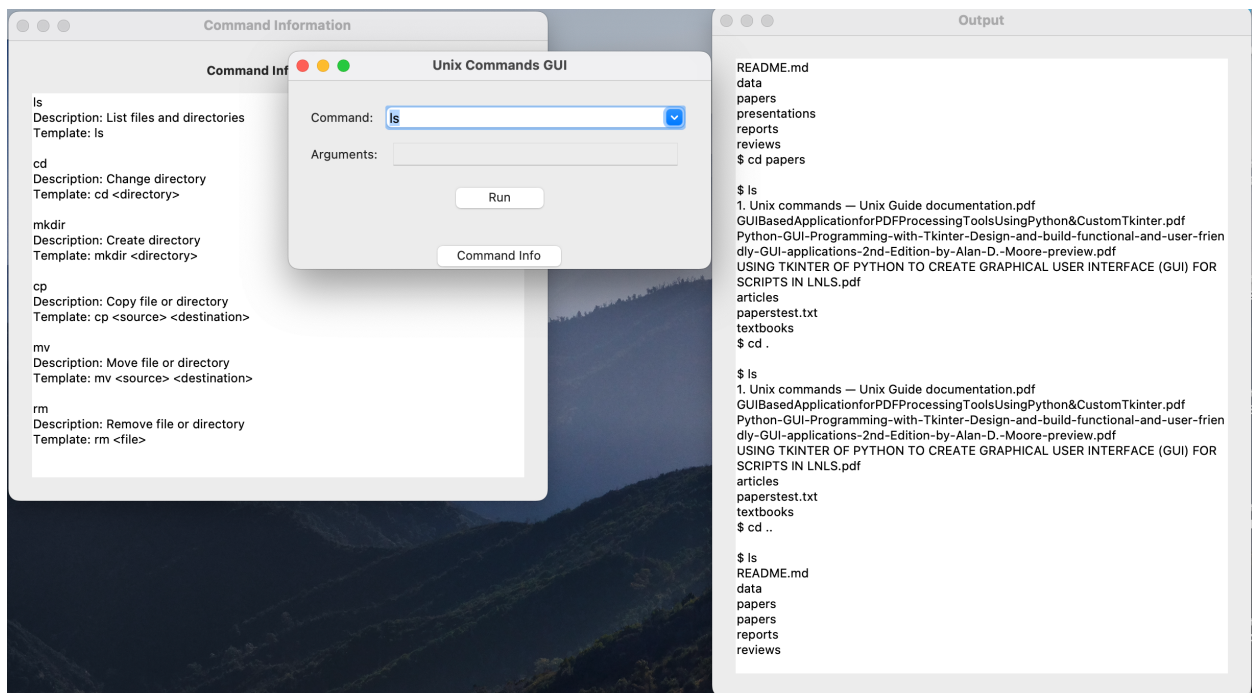
The project so far has created a graphical user interface (GUI) that helps users interact with Unix commands in a more intuitive way. The GUI consists of several windows that effectively visualize the information and provide a user-friendly experience.

The main window, titled as "Unix Commands GUI," serves as the central interface for the user. It includes a command label, a command combo box, an arguments label, an arguments entry field, and a run button. The command combo box allows the user to select a command from a dropdown list, while the arguments entry field allows them to enter the necessary arguments. This setup helps users easily understand the available commands and input their desired arguments.

Upon clicking the run button, the output window appears with the title "Output." This window displays the results of the executed command in a text widget. By separating the output from the main interface, users can clearly see the command they executed and view the corresponding output without clutter. The output text widget is initially disabled to prevent accidental modifications, maintaining the integrity of the displayed information.

To provide users with additional information about the available commands, the code includes a Command Info button. When clicked, this button opens a separate command information window with the title "Command Information." The window contains details about each command, including descriptions and templates, which allows users to quickly grasp the purpose and usage of each command.

From the tkinter package, the Toplevel function was used to create these additional windows, enabling an organized and focused presentation of information. By dividing the functionality into separate windows, the GUI design enhances user comprehension and interaction. Users can easily navigate between windows, input commands and arguments, view command information, and review command output—all within an intuitive and visually appealing environment.



This research report discussed the measurement strategies implemented to identify and fix input or usage errors within the GUI application. Overall, this project has demonstrated the importance of qualitative research methods, such as error handling strategies and user feedback, in improving the GUI application's usability and performance. While statistical analysis may not be applicable in this context, the project's focus on qualitative evaluation and iterative development has laid a solid foundation for creating a user-friendly GUI. Additionally, the report emphasized the importance of data visualization in presenting the GUI's output and user interface elements. By leveraging these strategies, the project aims to improve error handling, enhance the user experience, and provide a more intuitive graphical interface for executing Unix commands.