Sravya Kuchipudi
Professor Kaisler, Hasanov
CSCI 6917
8 August 2023

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

The focus of the last week was doing further testing of the GUI so any errors or unnoticed issues could be addressed as well as creating the presentation slides. Crafting an intuitive graphical interface to execute Unix commands required comprehensive testing across various aspects, each playing a pivotal role in delivering a seamless user experience, so it was necessary to ensure functionality, reliability, and user-friendliness. The slides helped bring the project together and summarize the work done so that it could be presented in a concise manner.

The first step of the process was to do code review which is critical to identifying and rectifying bugs, logical errors, and inefficiencies in the code. One thing that was realized during this process was that even though the code was easy to read it was a bit difficult to find issues just from reviewing it. While there were no noticeable logic errors that were observed, this step was still important because it makes the developer take a closer look at what they've written to see if there is anything that is odd.

Functionality testing was crucial to validate that all commands listed in the drop-down menu were operational and produce the desired results. Each command needed to execute correctly with both valid and invalid arguments, ensuring the application responds appropriately to different inputs. This process also included confirming that the drop-down menu displays all available commands correctly and that the 'Arguments' entry field accepted user input and displays the relevant tooltip. Additionally, the 'Run' button's functionality needed to display the executed command's output accurately in the output window. Proper error messages needed to be displayed when necessary, such as when an invalid command is selected, or arguments are missing. The errors that could have occurred in this part were mostly addressed as each new command was added since it was important to make sure what was being introduced would work properly. The only real issue that needed to be fixed for functionality was an added note about only using .txt files when trying any of the viewing options. There is a general issue when trying to view any file that isn't a .txt file since they are not processed correctly and show odd text. So, to counter this, a warning was added to instructions and Project README.file to help counter act the problem.

Compatibility testing played a vital role in guaranteeing a consistent user experience across various systems. Despite the application's development on a single computer, testing on a different system was essential to ensure its compatibility. This was done using a virtual machine that was MacOS based on Virtual Box due to the ease of access. Through the virtual machine, the testing revealed that the GUI application responded uniformly, regardless of the underlying machine.

The presentation slides were created based on the work that was done and the various points that should be highlighted. The slides initially outlineD the project objective, emphasizing the GUI's role in simplifying command execution and offering a solution to the traditional command prompt. They also addressed the Heilmeier Questions to underscore the project's impact. The next slides were used to the technical approach, showcasing the GUI's architecture diagram, key research steps, and its innovative aspects, like menu-based command selection and error handling mechanisms. The following results slide were created to display the GUI and the various outputs it can have. The conclusion slide summarized the project's achievements while the last slide covered future work possibilities, such as tracking features and other usability improvements.

In conclusion, the Unix Commands GUI application underwent comprehensive testing to ensure its functionality, reliability, and user-friendliness. Each aspect of testing played a significant role in

creating an efficient working tool for executing Unix commands. By adhering to a systematic and thorough testing approach, any minor issues were identified and addressed. Overall, the slides presented a coherent narrative of the GUI project's development and its potential future impact. Moving forward, regular maintenance and updates based on user feedback will further contribute to the application's success and usability, cultivating a positive and satisfying user experience. As the developer, the project can definitely be further refined so the application meets the needs and expectations of its users, ultimately enhancing their interaction with Unix commands.