**Option 1: Creating a User Interface I**

Miriam V. Armendariz

Colorado State University Global

CSC372: Computer Programming II

Dr. Jeff Yearwood

June 22nd, 2025

**Option 1: Creating a User Interface I**

For this assignment, I chose Option 1: Creating a User Interface I, which includes the implementation of a menu-driven Java Swing GUI application. The app includes a functioning menu bar with four interactive features that blend GUI design, event handling, and file operations.

1. Menu Bar with Four Options:

A screenshot of a computer

Description automatically generated

I built the menu using JMenuBar, JMenu, and JMenuItem, creating a clean, organized top bar menu that responds to user selections in real time.

1. Print Date and Time:

A screenshot of a computer

Description automatically generated

The first menu item inserts the current date and time into a JTextArea. I used LocalDateTime.now() along with DateTimeFormatter to format the output for readability. This demonstrated how to combine GUI input/output with Java’s time API.

1. Write Text to File (log.txt):

A screenshot of a computer

Description automatically generated

The second option writes whatever is in the menu JTextArea into a file called log.txt. I used a BufferedWriter wrapped around a FileWriter for efficient I/O. This added an element of persistent storage and helped me practice safe file handling.

1. Random Green Hue Background Change:

A screenshot of a computer

Description automatically generated

The third feature changes the GUI’s background to a randomly generated shade of green. I did this by generating RGB values with low red, high green, and low blue components to sstay within the green spectrum. Each selection not only updates the background but displays the RGB values used adding transparency to what the program is doing.

1. Exit Application:

The fourth menu item closes the application using System.exit(0), ensuring a clean and immediate shutdown.

**Technical Tools Used:**

1. Java Swing - JFrame, JMenuBar, JMenu, JTextArea, and more to create the GUI.
2. Java IO - BufferedWriter, FileWriter to handle file writing.
3. Java Time API – for real-time date and time.
4. Random Class - to generate custom shades of green.
5. Git – version control throughout the project.

**Version Control Strategy:**

I maintained a Git repo to manage the entire build. Key commits were made at major development stages:

* GUI layout setup
* Date/time functionality
* File writing logic
* Background color generation
* Final polish and debugging

These commits allowed me to compartmentalize tasks and roll back cleanly if anything broke along the way a lesson in disciplined version control that will pay off in future projects.

**Conclusion**

This assignment helped me flex multiple Java muscles: GUI construction, file I/O, color logic, and modular programming. It also showed me how valuable Git is as a development tool not just for tracking changes, but for organizing thought, isolating features, and building like a pro. Every part of this project reinforced a core principle: clean code, clean commits, clean logic.

<https://github.com/Miriamva216/Java-GUI-Menuapp>