## 

## 

**Experiment 4:**

**Using Software Tools and Code Versioning System**

CPE106L (Software Design Laboratory)

**Member 1: Claros, Angelica A.**

**Member 2: Facal, Ma. Catherina N.**

**Member 3: Santos, Angelica Anne D.**

Group No.: **5**

Section: **B2**

## **PreLab**



|  |
| --- |
| **Laboratory Insights** |

**Claros**

In the fourth laboratory, we delved into testing codes and software applications intended for building programs with a user interface (UI), such as Tkinter and Flutter. This laboratory served as an integration of the learnings and insights gained from laboratories 1 to 3, extending our understanding to more complex concepts.

Initially, unit testing was the first topic discussed. Unit testing involves testing individual parts of software to identify and address defects or bugs within modules or programs that hinder developers from achieving their objectives. Based on the handouts, it's important to note that unit testing focuses solely on a unit of code and may not address broader system-level aspects. Furthermore, we explored various software applications for app development, including Tkinter and Flutter. Both applications support Python as a backend language and serve as frameworks for building applications with user interfaces. It was discovered that these tools can enable us to create interactive user experiences, enhancing the overall functionality of projects.

In summary, this laboratory presented challenges but was also crucial in providing us with knowledge, particularly in application development, which will be instrumental in our final project for this course.

**Facal**

The fourth laboratory focused on how to create desktop applications, unit testing, design patterns, and graphical user interface (GUI). Design patterns are useful in solving common software development issues via organizing Python code effectively for easier maintenance and modification. Unit testing makes sure that the code functions as intended, helps in identifying and resolving bugs to meet requirements. Example of GUIs that can be used in Python is Tkinter and Flet. Tkinter provides a set of tools and widgets for building windows, buttons, menus, and other GUI components in Python applications. Flet, on the other hand, appears to be a library or framework specific to the context of your code.

Overall, the fourth laboratory session provided a comprehensive overview of key concepts and techniques essential for developing robust desktop applications in Python. From design patterns to unit testing and GUI development, we gained valuable insights and practical skills that will enhance our ability to create reliable and user-friendly software solutions.

**Santos**

Prelab Number 4 focused on constructing desktop applications, covering both console and GUI formats with Tkinter and Flet. Drawing from concepts learned in Labs 1 to 3, particularly Object-Oriented Design help me with the current domain.

Efficient structuring is key in desktop app development same as the demanding unit testing. Unit testing involves analyzing individual code segments to identify and rectify errors early which saves time in the development process. We delved into the Tkinter framework for GUI development, alongside unit testing frameworks, streamlining the testing process.

This incorporation of practices improved our understanding of unit testing and GUI development. It underscored the significance of structured application design in making desktop applications. The foundational knowledge built in previous labs laid the groundwork for further advancement in software development.