***Section 9 – Evaluation***

**Programming language – Python 3.8 in visual studio code**

Python was the chosen programming language for my project for several reasons. Firstly, it is the language with which I am most familiar as I have used it from GCSE up to now, providing an easy and familiar environment for development. Additionally, Python boasts a rich set of built-in features and supports modular code organization through functions, facilitating the implementation of various functionalities that came In very useful during the development of my project.

Opting for a language I was already well versed in was crucial for efficiency and effectiveness. Learning a new language while working on this project would have introduced unnecessary difficulties and made the development time a lot larger. Sticking with Python allowed me to leverage my existing knowledge and execute tasks with greater ability, whilst also learning more about the language such as the how functions work across other py files

Python's extensive library further reinforced it was the language for my project. One such library, tkinter, offered a wide variety of widget objects, simplifying the creation of the graphical user interface (GUI). In particular, the ttk TNotebook widget provided sleek tabbed windows, reducing the need for copious amounts of code for window management.

Moreover, Python's syntax allows for easier readability and organised code, encouraging the formation of clear, structured sections of code. This enforced a disciplined approach to coding, resulting in a program that is easy to update and fix

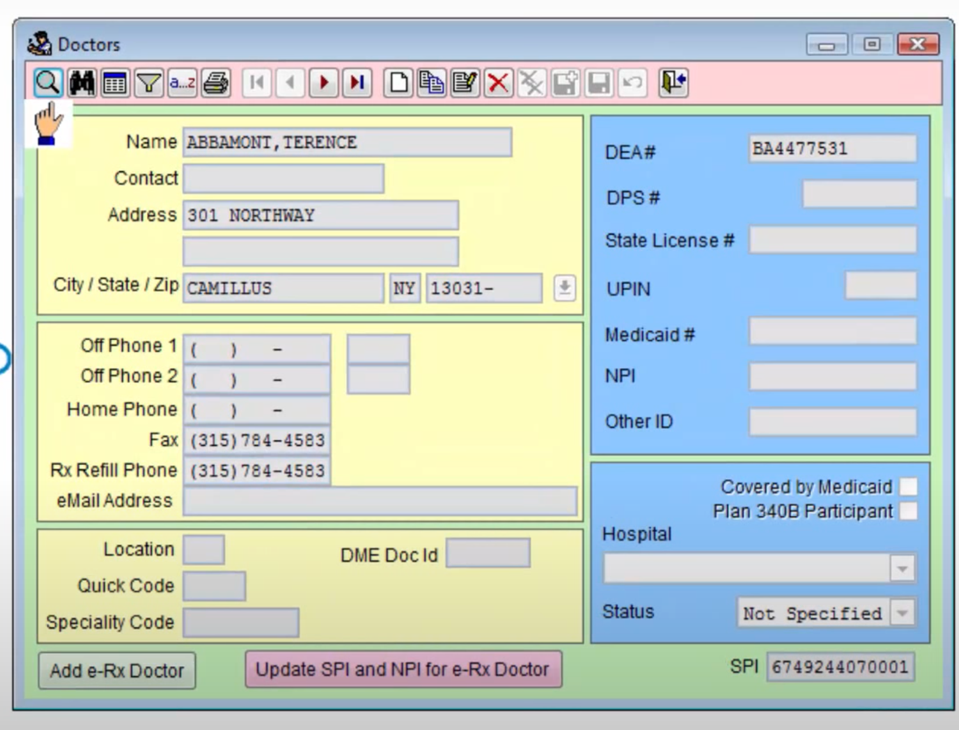
Utilizing Python's regex module allowed for password validation by enabling the identification of special characters such as !. Additionally, the hashlib module enhanced password security by enabling password hashing using encryption. Using these features from the Library made for very efficient development by providing pre-developed functions.

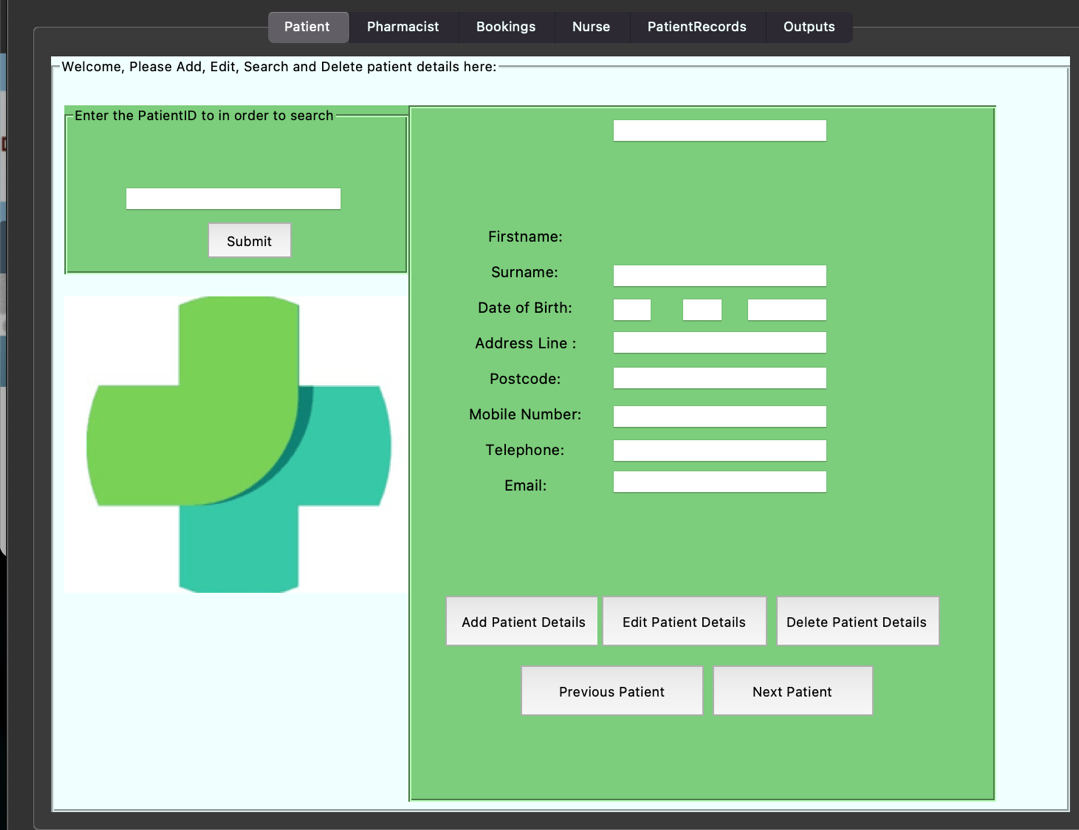
In summary, Python's familiarity, Great library, and readability, made it the perfect choice for my project, enabling efficient development and wide functionality.

**Object Orientated Programming**

I adopted an Object-Oriented Programming approach in developing the code, which centers around the use of classes and objects. In this paradigm, classes serve as templates for data structure and methods, while objects represent instances of these classes. Within my code, I structured it around three distinct classes, each corresponding to a GUI window. Each GUI class contained various lines of code for arranging the placement, dimensions, and visual presentation of Tkinter widgets within the GUI.

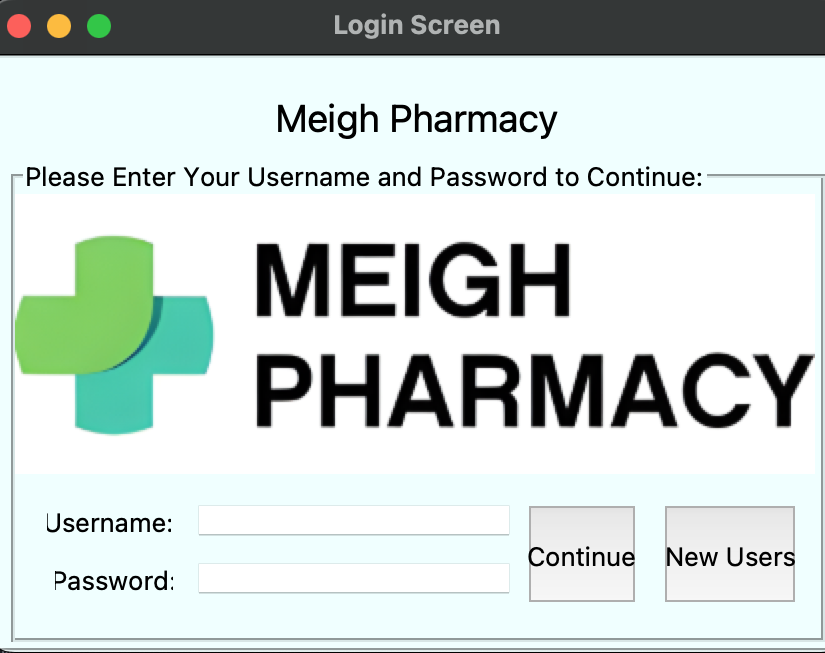
By using Object Orientated Programming, the program embodied these classes, presenting them to the user in the system. This method proved to be highly effective, especially given the nature of GUI programming, which often aligns with an object-oriented approach. Since Tkinter widgets function as objects, turning them into larger objects emerged as a natural and logical decision.

**Comparison with desked based research**



Above is a comparison of my final Main menu GUI compared to the system created by Visual Super Script Software. While my system isn’t as complex and feature packed as Visual Super Scripts’, I am happy with how colorful and eye-catching my own system is and I feel that I have achieved an appealing look that the Visual Super Script defines. I also feel that I have incorporated the main features from Visual Super Script that are vital for the system.



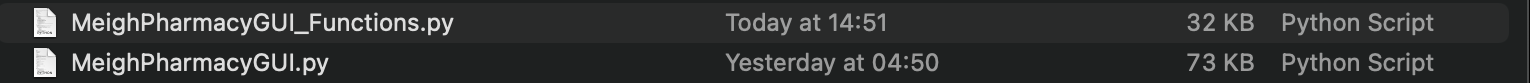


Here is a comparison of my Login window and Pharmaco’s login window. I feel both are quite visually appealing however Pharmaco’s definitely is of a higher standard. My login screen is more accessible as it does contain a new user’s button however one limitation of my own is that it does not contain a forget password button, and I feel this would have been smart to implement. However, I feel both login screens meet their function in that they both log the user in, whilst being visually easy to understand

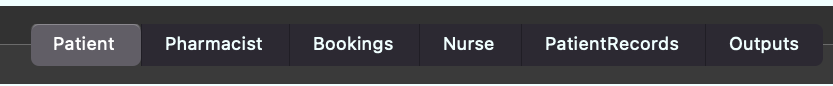
**Positives and Negatives of the System**

Here is some of the positives of my system:

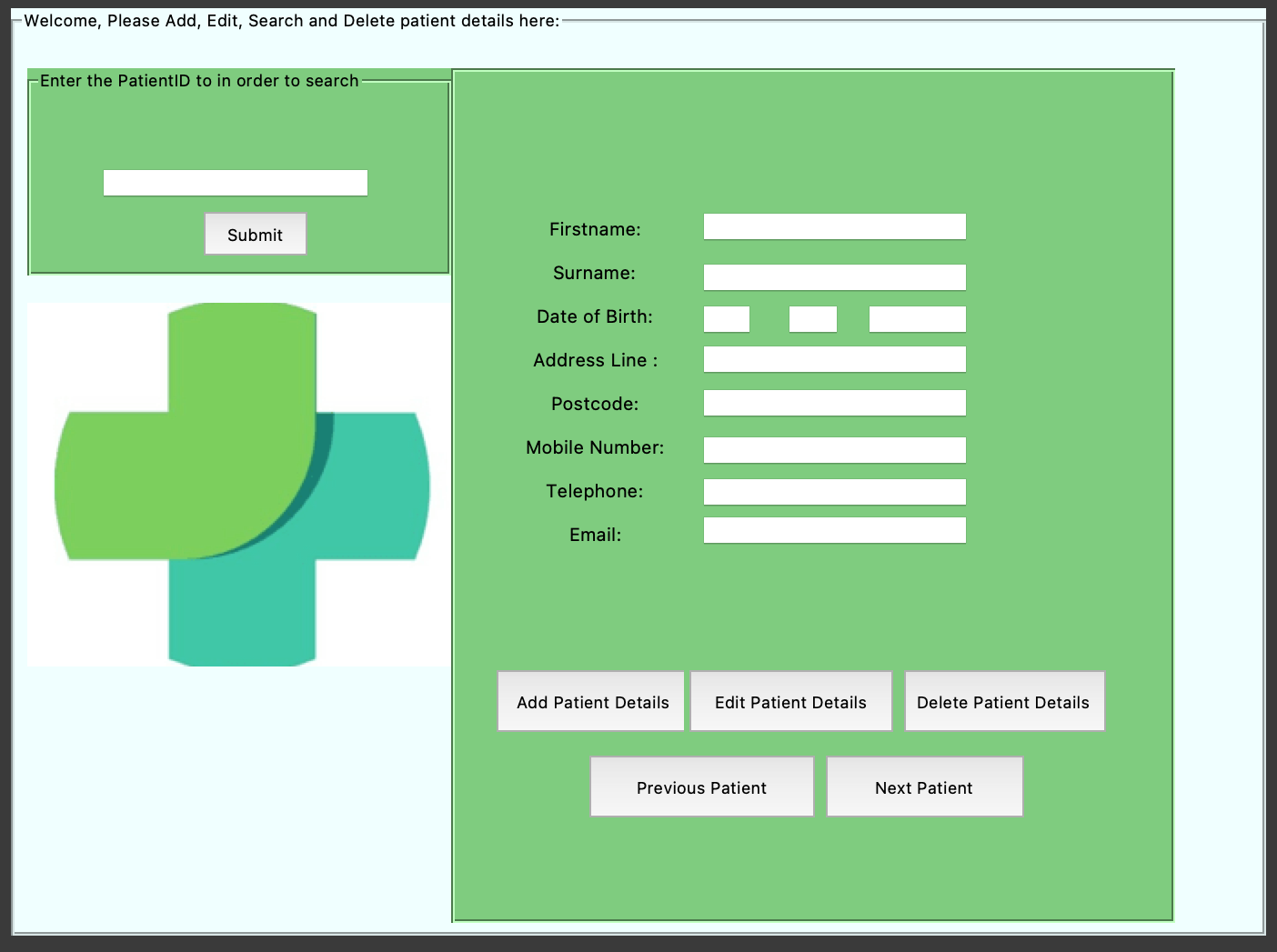
* Having my code being modular has allowed me to become more efficient and also made the code easier to navigate and understand. This has been a great help when solving issues and adding to my system. An example of the modularity is having a MeighPharmacyGUI file and having a MeighPharmacyGUI\_Functions file



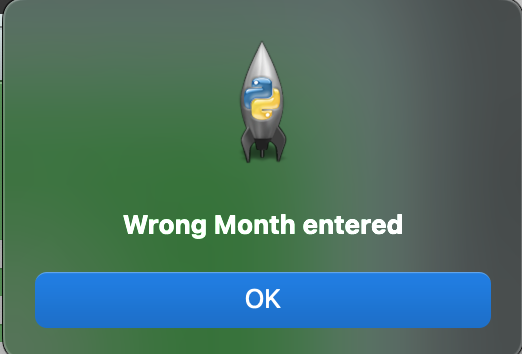
* Window tabs allows for easy access to the system and means less staff training is required

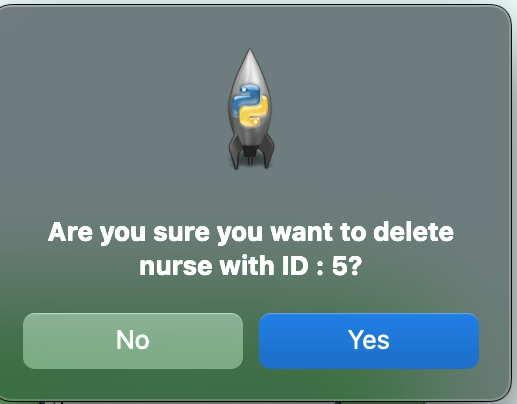


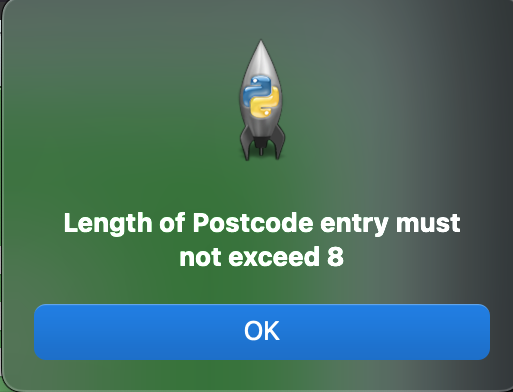
* The use of various entry fields for Adding, Deleting, and Editing data in a proficient and simple way

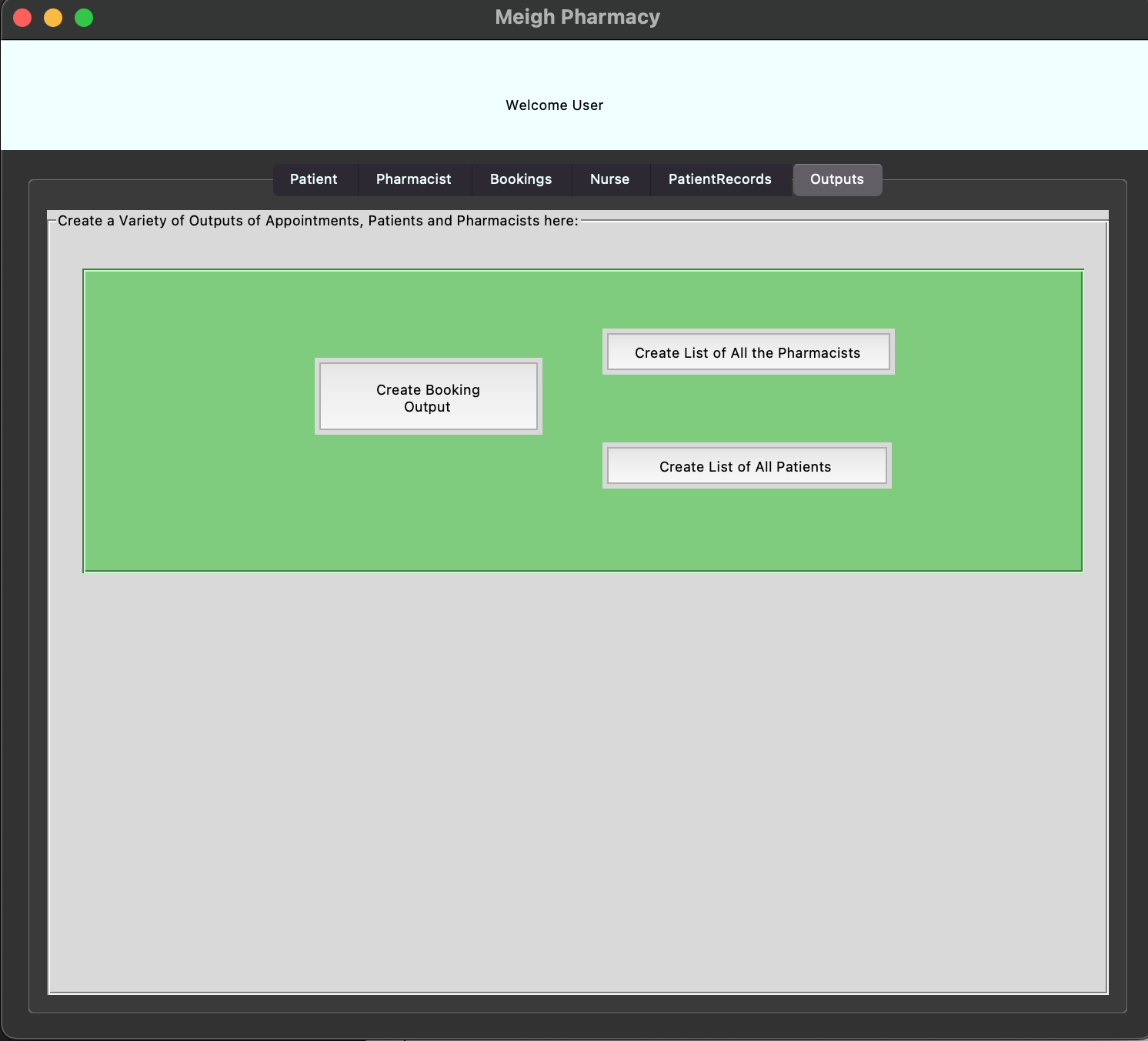


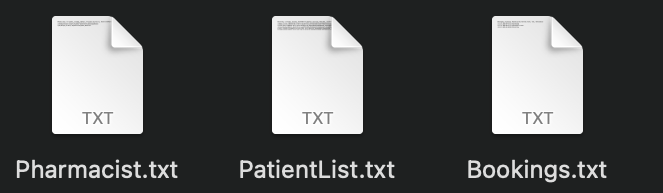
* The system responds to various user actions and educates the user on what they need to do/what they have done.

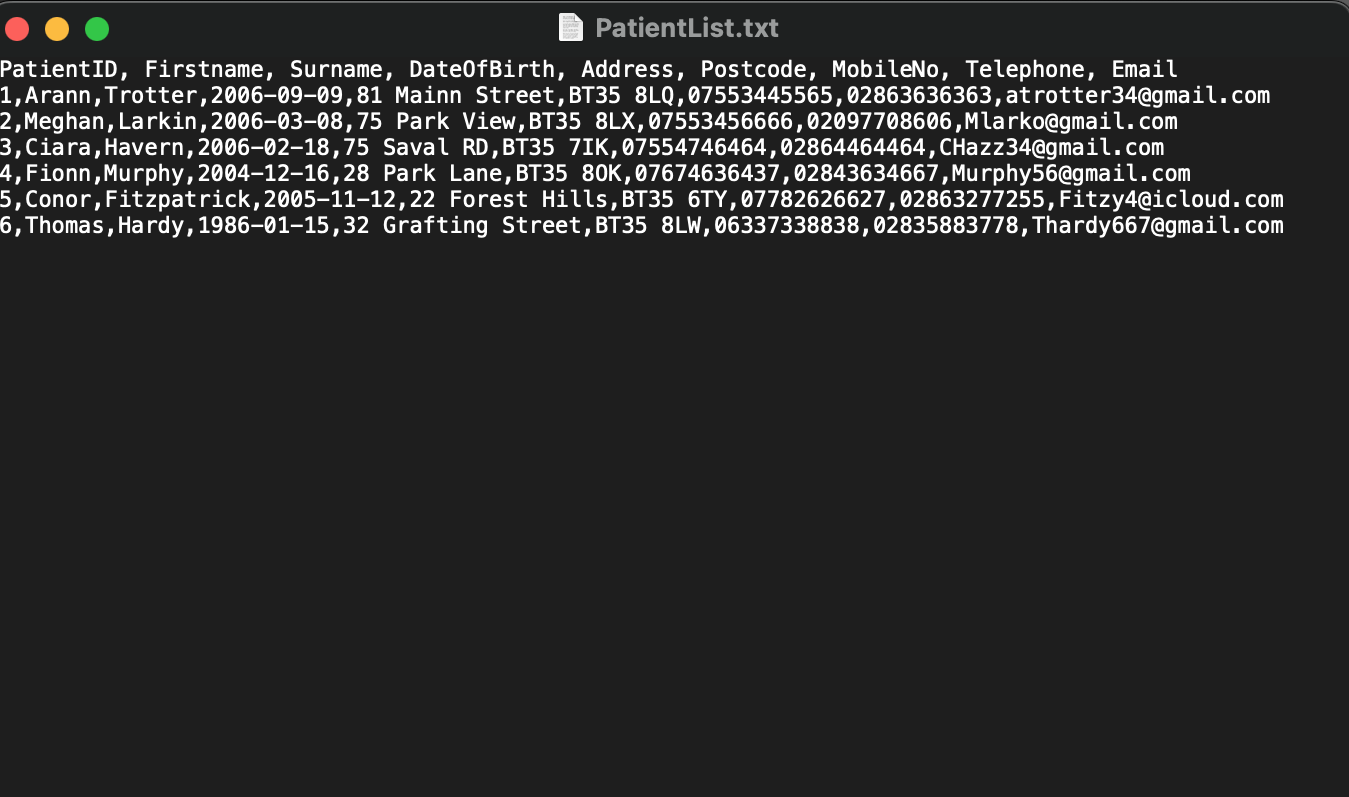


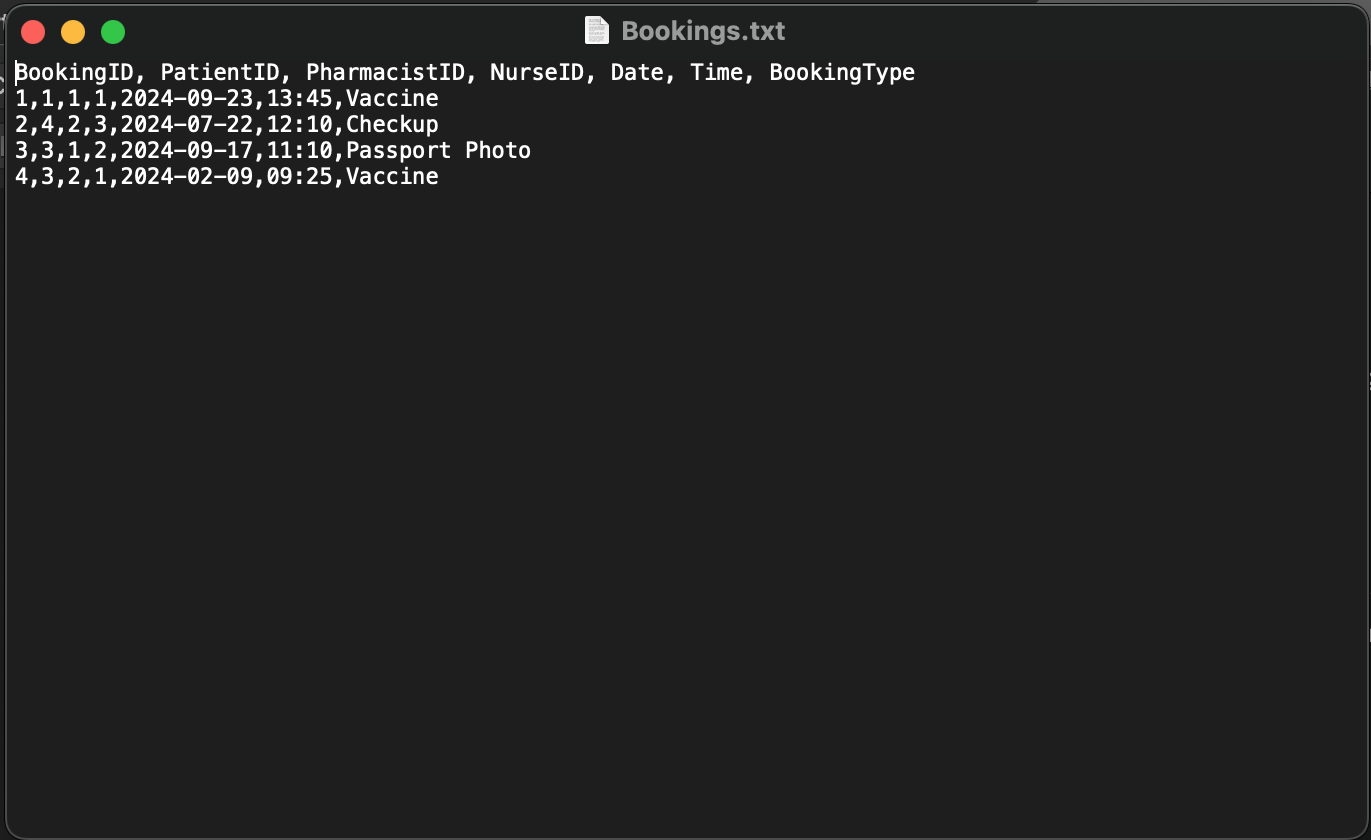




* Being able to create various outputs so that the user can view data in the database







However, there are several limitations of my system:

1. My GUI Lacked a polished look that is standard with commercial systems. May have been able to use custom Tkinter to make the GUI look better. Also, could have added mor colors to the system in order to outline certain features.
2. After creating my solution, I found that my encryption method for the passwords is not the most secure and efficient. Could have also used an admin key for the new user screen to further develop security
3. Could have used more Tkinter widgets such as drop-down windows, in order to further improve the functionality and ease of use of the system. A scroll bar could have also been coded to further aid functionality
4. An absence of access levels is frustrating as it could have been a very useful feature that not only improved security but would have also rivaled commercial software that costs more than my system.

I have some suggestions for potential enhancements that could refine the system further in the future:

1. Enhanced Access Control - Using the "state" attribute of each TNotebook tab enables manipulation of access levels. By enabling or disabling tabs, different staff types can be granted or restricted access accordingly. This method offers effective control over access levels within the system.

2. Diversification of Input Widgets - Expanding beyond just the entry widget can improve the overall appearance and professionalism of the final program. By incorporating a variety of widgets, the need for extensive validation functions, especially for fields such as date and postcode, can be reduced.

3. Integration of External Modules - Using modules not included in the Standard Library, for graph creation or for secure password encryption, can enhance the system's functionality and security. With additional development time, integrating these modules could facilitate the creation of a wider range of outputs and ensure safer storage of passwords.

**My own strengths and Weaknesses**

Throughout the development of my completed system, I identified strengths in several key areas:

Smart Naming - I employed intelligent naming conventions for all aspects of the code, including variables, methods, and classes. This attention to detail enhances readability, which is crucial for collaboration with other developers. Additionally, the use of comments further contributes to the program.

Effective Validation Functions - I developed effective validation Functions, with a particular highlight being the checking of the postcode. Implementing a versatile method for validating entry fields of postcode greater verified the validation process.

Clearly Structured Code - I organized the code in a clear and structured manner. Grouping each section of the tab window's code together ensures easy readability, facilitating maintenance and future Additions.

Willingness to Learn and Improve- I demonstrated a willingness to improve and expand my knowledge in Python programming. During the creation of the code, I had to learn new ways to call on functions across python files in order to fully develop the function of the create output buttons.

However, I also encountered several weaknesses and challenges during the development process:

Limited Experience in Python - Despite dedicating two years to programming in general, I still consider myself relatively inexperienced in Python. While I developed the system to the best of my current abilities, there are areas where I would like to further enhance my skills.

Little Knowledge of SQL - My understanding of SQL is limited, and I believe seeking assistance from someone with more experience in these areas, or an online course could have benefited my overall solution

Poor Time Management - I acknowledge that the time allocated for the project was not managed as effectively as it could have been. The process of configuring each widget in tkinter code consumed vital time, limiting opportunities to refine system processes and improve the overall appearance and function of the final product.

Challenges with Data Manipulation- Due to my unfamiliarity with SQL, the development of code for data manipulation took longer than anticipated. This took away from the time available to focus on programming a medicine stock table and definitely hindered my solution

**Improvements as to how I approach Software Development**

I plan to enhance my coding process by implementing the following improvements:

Conducting In-depth Research - I aim to dive deeper into the features offered by the python language. This greater understanding will aid me as a programmer by allowing me to fully understand what functionality is available to implement.

Seeking Third-party Feedback - I recognize the importance of soliciting feedback from experienced software developers, individuals familiar with similar systems, and my own peers. Their insights will ensure that the functionality I integrate uphold user expectations and provide valuable guidance for system improvement.

Exploring Alternative Programming Languages and Tools – I intend to Recognize the value in learning multiple programming languages, and to experiment with new languages to add to my skills. Additionally, I plan to explore advanced development tools like the Visual studio code IDLE, which offers features such as CodiumAI, Function Testing, and debugging.

Structuring Time Effectively - I plan to adopt a structured approach to time management by creating a development schedule to track the completion dates for each system component. By documenting decisions and requirements for each component, I can plan and execute tasks more efficiently and more quickly.