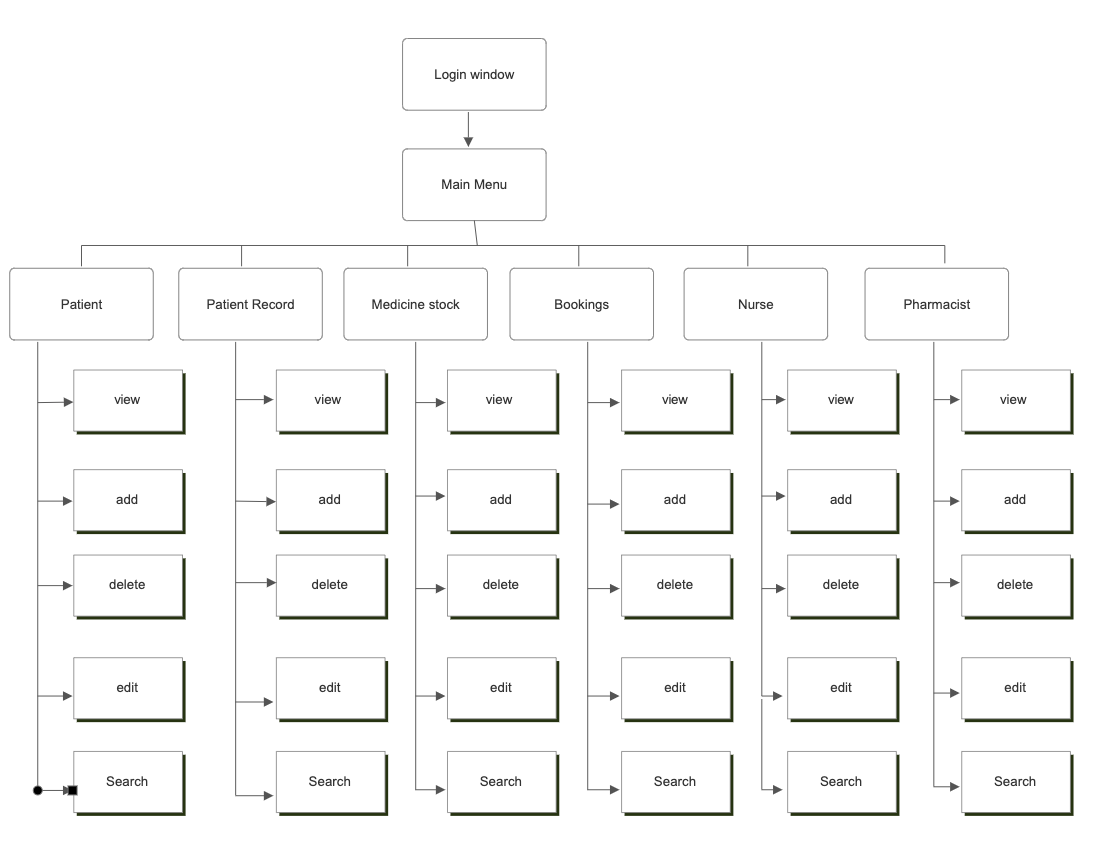
*Section 3 - Design*

**3.1 - Outline of Design**

My main goal at this stage of my project is to plan out a fully comprehensive design for the final version of my system. I will plan out my line of action to the software development stage of the project through various diagrams and by defining in detail many different modules, sub routines, and validation routines that I will use in my code. However, I may differ slightly to this structure as there are time constraints on my work and all sub routines may not be feasible in the given timeframe. I also won’t stick too close to the structure as I will design a prototype and respond to feedback on this given prototype. If I find more efficient ways to improve my code, I may also implement them.

**3.2- Justification of planned sub-routines/System Overview**

Firstly, my project will be divided into smaller sub-programs and functions so that I can code it effectively. Below is my main sub-routines and functions that will be implemented to my system and how they fit the Pharmacy’s needs.

**Login window**

This will be the first thing that the user encounters and they will need a valid Username and Password in order to gain access to the system. The type of user logging in (Nurse, Pharmacist, Assistant) will determine what each user is allowed to view.

**Main menu**

For my main menu I intend to use TNotebook widget to add widgets to my main menu, in order to allow for an easier system to use. From my questionnaire I learnt that the users required a system that was easy to use and had everything in one place and using TNotebook will allow me to have everything they need in one window. TNotebook will also allow for me to add security levels to my system by removing certain widgets for certain users.

**Separate Views**

Different users will be allowed to see different parts of the system based on their level of authority

**View Tables**

Allows for tables to be viewed in the program

**Patient Table**

Patients will be able to view their own records, however only staff of the pharmacy will be allowed to create, edit or delete patient details. Patient details can be searched using the Primary key PatientID.

**Patient Record Table**

Again, Patients will be able to view their own records with staff being able to update or delete patient records. Individual records can be searched using the search term RecordID however a multitude of records can be searched using PatientID

**Medicine Stock Table**

Patients will be able to see the MedicineID for the medicine they have been prescribed however only staff will be able to view, add, delete or edit medicine in the medicine stock table. Individual medicine can be searched using the MedicineID or a patients specific medicine can be searched using PatientID

**Bookings Table**

Patients will be able to view their own bookings only. Staff will be able to view, add, edit, and delete bookings. Bookings can be searched through the BookingID search term.

**Nurse Table**

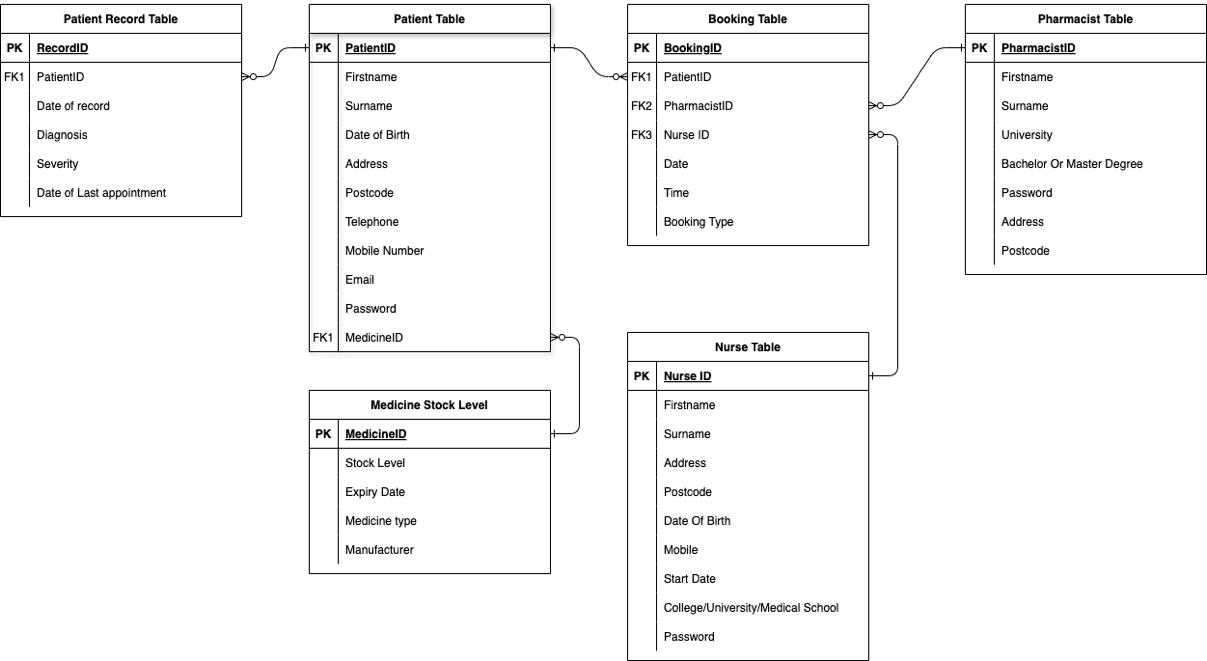
Only staff have access to this tab. Staff will be able to view, add, edit, and delete Nurse details and nurse details can be searched using the NurseID search term.

**Pharmacist Table**

Again, only staff have access to this tab. They can view, add, edit and delete Pharmacist details and this can be searched using Pharmacist

**Search functions**

Records will be able to be searched through using a search term

**Entity Relationship Diagram**

**Relationships**

* Each Patient can have many patient records
* Each Patient can have many bookings
* Each Medicine can be prescribed to many patients
* Each Nurse can have many bookings
* Each Pharmacist can have many bookings

**Data Dictionaries**

Patients Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| PatientID | Primary | Integer | 4 | An ID Specific to each Patient. Can be used as a search term | Presence Check  Type Check | 123 |
| Firstname | - | String | 20 | Patient’s Firstname | Presence Check | John |
| Surname | - | String | 20 | Patient’s Surname | Presence Check | Phelan |
| Date of Birth | - | Date | 10 | Patient’s Date of Birth | Format Check (DD/MM/YYYY) | 23/03/2007 |
| Address | - | String | 30 | Patient’s Address | Presence Check | 45 Jesmond Rd |
| Postcode | - | String | 8 | The postcode of the patient’s Address | Format Check (LL000LL) | BT34 8LX |
| Telephone | - | String | 11 | Patient’s personal Telephone number | Length Check (11 Characters) | 02840858710 |
| Mobile NO. | - | String | 13 | Patient’s personal Mobile number | Length Check (13 Characters) | +447661694494 |
| Email | - | String | 30 | Patient’s personal email | Presence Check | Jphelan@gmail.com |
| Password | - | String | 30 | Patient’s password which can be used to sign into the system | Presence Check | Footy2208 |
| MedicineID | Foreign | Integer | 4 | Unique ID for medicine that this patient is taking | Presence Check  Type Check | 3301 |

Patient Records Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| RecordID | Primary | Integer | 4 | An ID Specific to each Patient’s record. Can be used as a search term | Presence Check  Type Check | 4412 |
| PatientID | Foreign | Integer | 4 | Unique ID from Patients Table | Presence Check  Type Check | 321 |
| Date of record | - | Date | 10 | Date that the record was created | Format Check (DD/MM/YYYY) | 31/05/2023 |
| Diagnosis | - | String | 30 | Diagnosis that the patient has been given by the pharmacist | Presence Check | “Signs of malignant tumor suggest Radiology needed.” |
| Severity | - | String | 30 | An opinion on how severe the patient’s condition is | Presence Check | “Quite Severe” |
| Date of last Appointment | - | Date | 10 | Date of patient’s last visit to the pharmacy | Format Check (DD/MM/YYYY) | 23/08/2023 |

Bookings table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| BookingID | Primary | Integer | 4 | An ID Specific to each booking made. Can be used as a search term | Presence Check  Type Check | 36 |
| PatientID | Foreign | Integer | 4 | Unique ID from Patients Table | Presence Check  Type Check | 231 |
| PharmacistID | Foreign | Integer | 4 | Unique ID from Pharmacist Table | Presence Check  Type Check | 311 |
| NurseID | Foreign | Integer | 4 | Unique ID from Nurse Table | Presence Check  Type Check | 103 |
| Date | - | Date | 10 | Date of the booking | Format check (DD/MM/YYYY) | “16/05/2024 |
| Time | - | Time | 10 | Time of the booking | Format Check (HH:MM)  Type Check | 11:11 |
| Booking Type | - | String | 30 | Type of booking | Presence Check | Vaccination or Checkup. |

Medicine Stock Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| MedicineID | Primary | Integer | 4 | An ID Specific to each Medicine. Can be used as a search term | Presence Check  Type Check | 204 |
| Stock Level | - | Integer | 3 | Stock level of specific medicine | Presence Check | 95 |
| Expiry Date | - | Date | 10 | Date that the medicine expires | Format Check (DD/MM/YYYY) | 12/09/2024 |
| Medicine Type | - | String | 30 | Form that the medicine comes in | Presence Check | “Liquid” or “Tablet” |
| Manufacturer | - | String | 30 | Name of manufacturer of the specific medicine | Presence Check | “Pfizer” |

Nurse Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| NurseID | Primary | Integer | 4 | An ID Specific to each Nurse at the pharmacy. Can be used as a search term | Presence Check  Type Check | 290 |
| Firstname | - | String | 20 | Nurse’s Firstname | Presence Check | Linda |
| Surname | - | String | 20 | Nurse’s Surname | Presence Check | Murphy |
| Address | - | String | 30 | Nurse’s personal address | Presence Check | 81 Main Street |
| Postcode | - | String | 8 | The Postcode of the Nurse’s address | Format Check (LL000LL) | BT34 8LQ |
| Date of Birth | - | Date | 10 | Nurse’s date of birth | Format Check (DD/MM/YYYY) | 17/04/1994 |
| Mobile | - | String | 13 | Nurse’s Mobile number for use of contact | Length Check (13 Characters) | +447661793387 |
| Start Date | - | Date | 10 | Nurse’s first day at work for the pharmacy | Format Check (DD/MM/YYYY) | 12/08/2022 |
| College/University/  Medical School | - | String | 30 | Name of the institution that the nurse studied at | Presence Check | “Queens University Belfast” |
| Password | - | String | 30 | Nurse’s password which can be used to login to the system | Presence Check | Lmurphy778 |

Pharmacist Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fieldname | Key Type | Data Type | Field Length | Description | Validation | Example |
| PharmacistID | Primary | Integer | 4 | An ID Specific to each Pharmacist at the pharmacy. Can be used as a search term | Presence Check  Type Check | 401 |
| Firstname | - | String | 20 | Pharmacist’s Firstname | Presence Check | Richard |
| Surname | - | String | 20 | Pharmacist’s Surname | Presence Check | Garvey |
| Address | - | String | 30 | Pharmacist’s personal address | Presence Check | 75 Brook Rd |
| Postcode | - | String | 8 | The Postcode of the Pharmacist’s address | Format Check (LL000LL) | BT25 9LD |
| University | - | String | 30 | Name of university that the Pharmacist studied at | Presence Check | “Trinity College Dublin” |
| Bachelors or Masters | - | String | 20 | Level that the pharmacist has achieved in university | Presence check | Bachelors |
| Password | - | String | 30 | Pharmacist’s Password which can be used to login to the system | Presence Check | Rgarvey220 |

Method of Access:

All data in my proposed solution will be stored in an SQL table called “Mpharmacy.db”. this table will hold all of my programs data, and SQLite commands will be carried out throughout my program in order to connect to the database, and allow the user to view, edit, add, and delete records. Each field will use an appropriate data type and the TNotebook widget will determine what each user can view.

**Normalisation**

**Unnormalised:**

PatientID, Firstname, surname, Date of birth, Address, Postcode, Telephone, mobile, email, password, MedicineID, stock level, expiry date, medicine type, manufacturer, RecordID, date of record, diagnosis, severity, date of last appointment, BookingID, Date, Time, Booking Type, NurseID, Start Date, College/university/medical school, password, PharmacistID, University, Bachelors or Masters.

**1NF**

**Patients** (PatientID, Firstname, Surname, Date of Birth, Address, Postcode, Telephone, Mobile, Email, Password, MedicineID, Medicine Type, manufacturer)

**Medicine Stock** (MedicineID, Stock level, expiry date, medicine type, manufacturer)

**Patient Record** (RecordID, PatientID, Date of record, Diagnosis, Severity, Date of Last Appointment)

**Booking** (BookingID, PatientID, PharmacistID, NurseID, Date, Time, Booking type, Patient Firstname, Patient Surname, Nurse Firstname, Nurse Surname,)

**Nurse** (NurseID, Firstname, Surname, Address, Postcode, Date of Birth, Mobile, Start Date, College/University/Medical School, Password)

**Pharmacist** (PharmacistID, Firstname, Surname, University, Bachelors or Masters, Address, Postcode, Password)

**2NF/3NF**

**Patients** (PatientID, Firstname, Surname, Date of Birth, Address, Postcode, Telephone, Mobile, Email, Password,)

**Medicine Stock** (MedicineID, Stock level, expiry date, medicine type, manufacturer)

**Patient Record** (RecordID, PatientID, Date of record, Diagnosis, Severity, Date of Last Appointment)

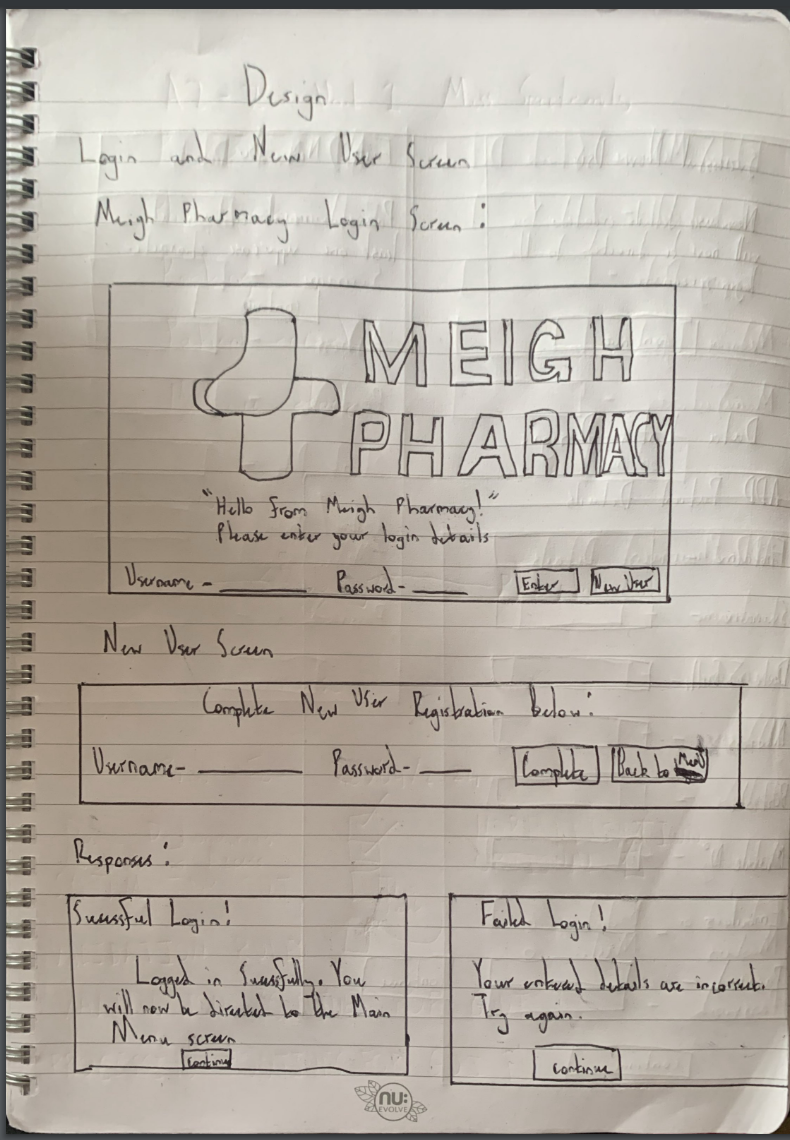
**Booking** (BookingID, PatientID, PharmacistID, NurseID, Date, Time, Booking type,)

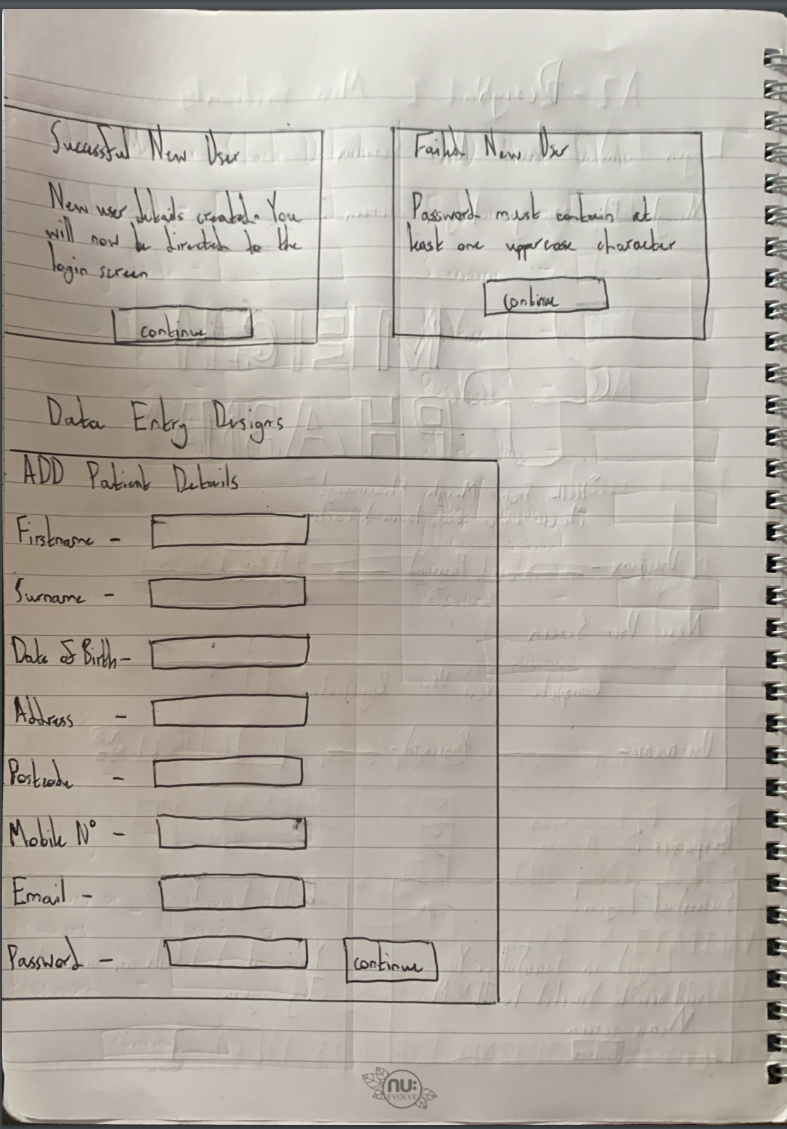
**Nurse** (NurseID, Firstname, Surname, Address, Postcode, Date of Birth, Mobile, Start Date, College/University/Medical School, Password)

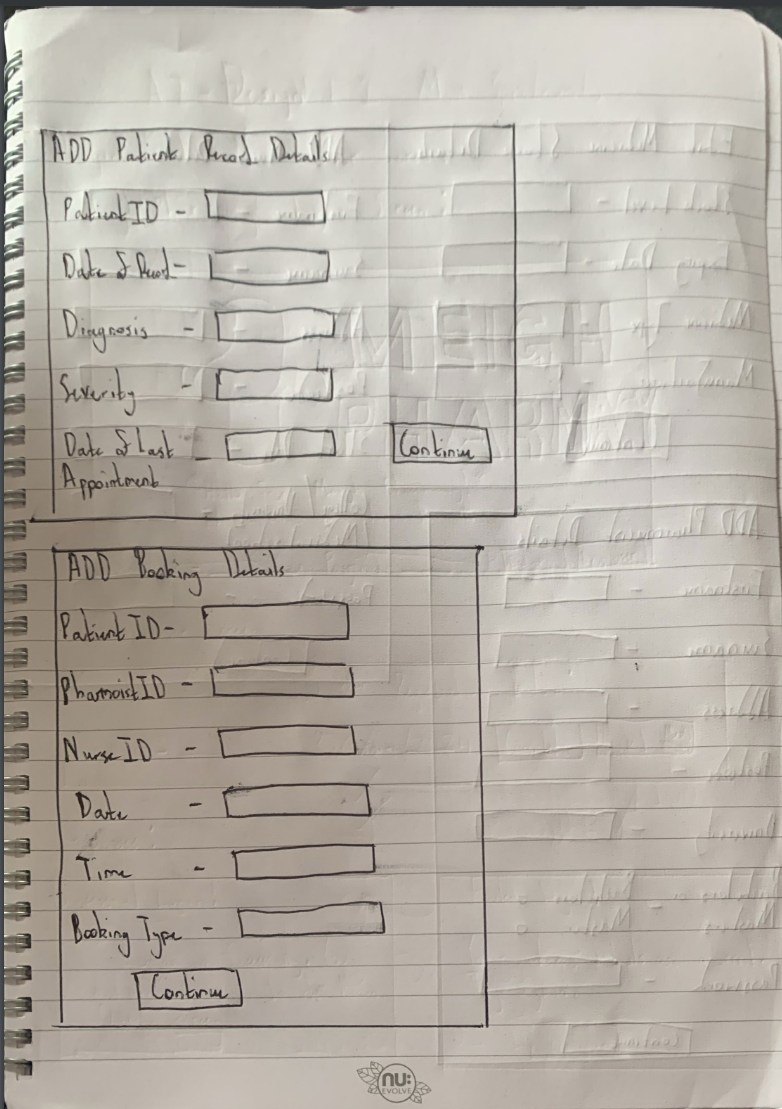
**Pharmacist** (PharmacistID, Firstname, Surname, University, Bachelors or Masters, Address, Postcode, Password)

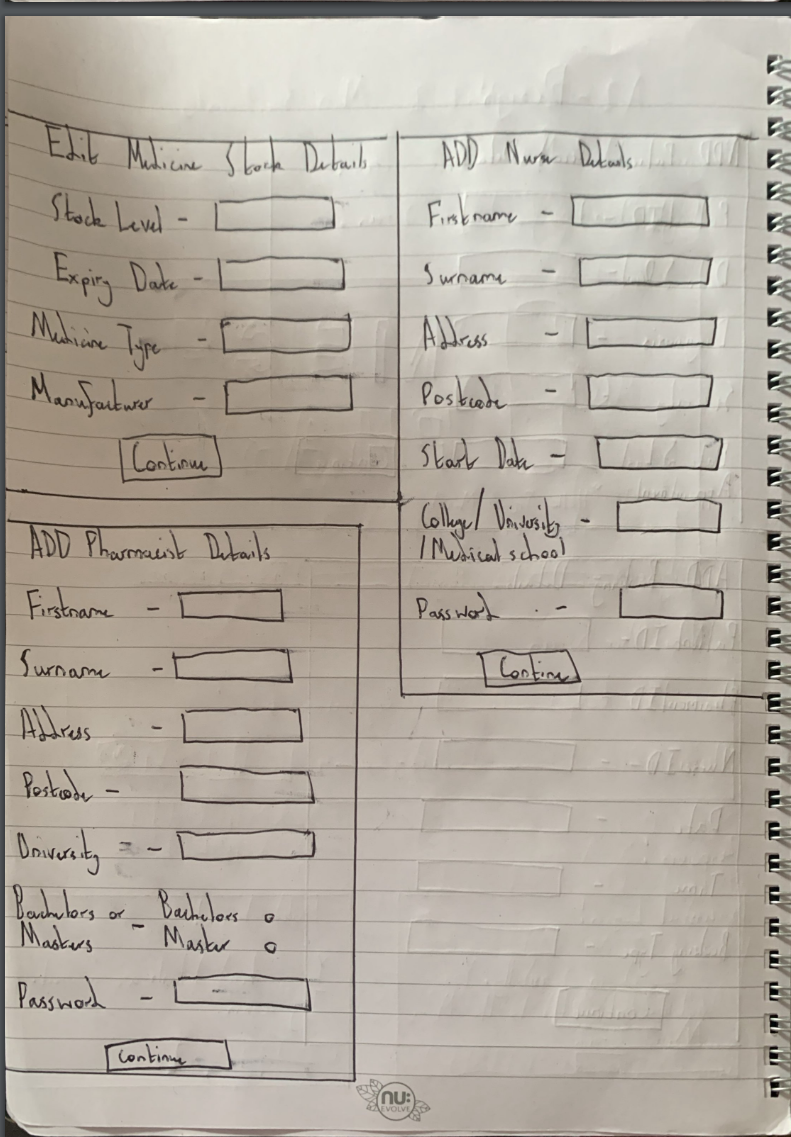
GUI Layout

Below are concepts and first iterations of what some sections of my program should look like. These are hand drawn and show different system inputs and outputs such as the planned login screen.





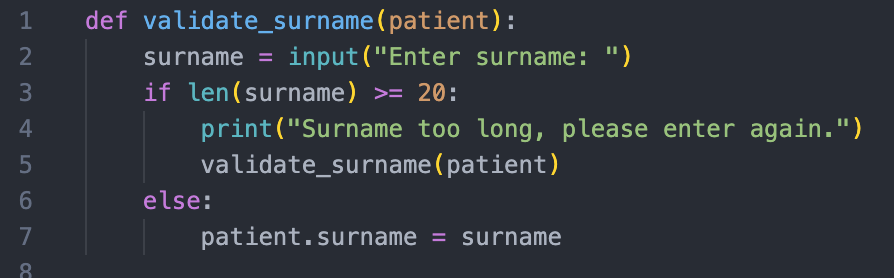






**Validation Routines**

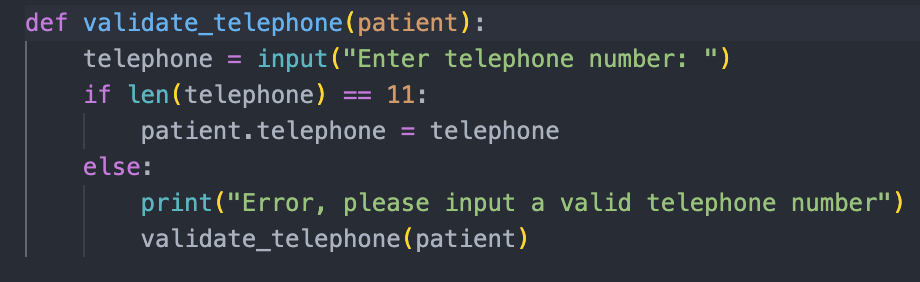
Instead of duplicating my code for each validation routine, I will create a comprehensive validation function. This introduces a number of benefits to the system such as an overall reduced file size and time saving. Every entry widget will have a “InstructionValidation” attribute which allow me to assign placeholders to track each value inside of the widget. These values can then be validated by comparing them against my previously mentioned validation routines.



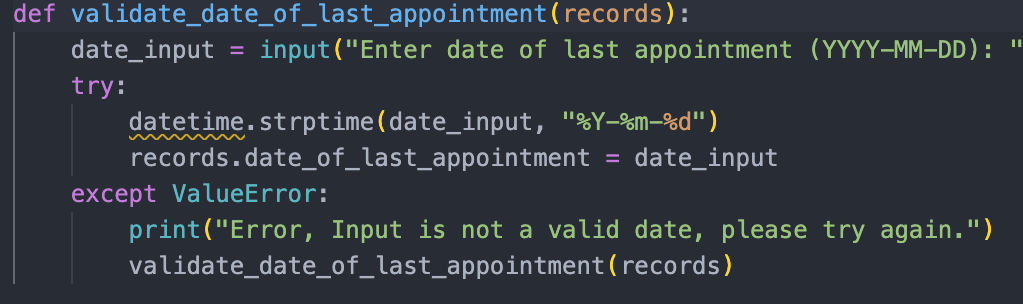
This is a validation routine for the Patients surname. This routine would also be the same for the patients Firstname.



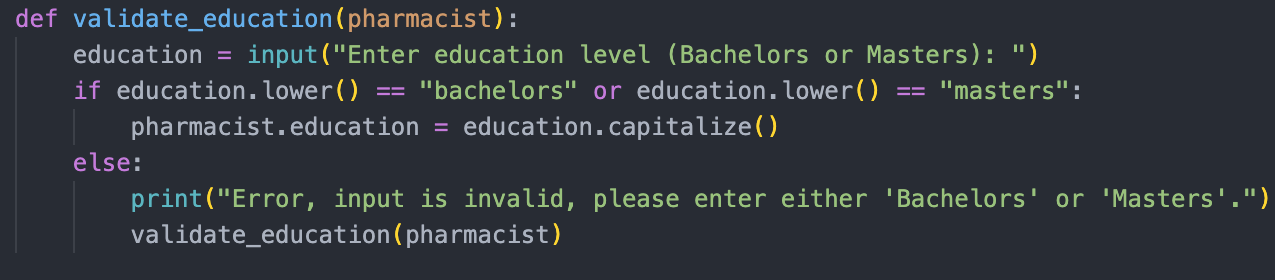
Again, this routine would also be the same for the pharmacist’s Firstname.

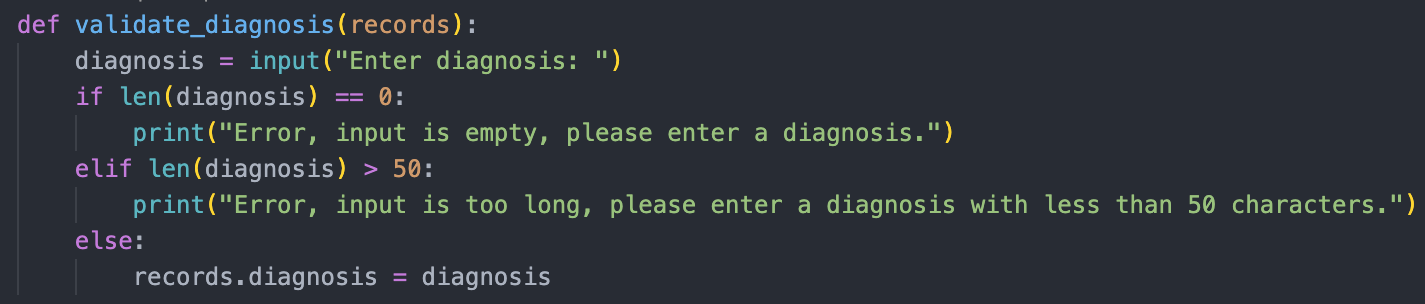


This is a validation routine for the Patients telephone number. This routine would also work for a mobile number.



This Validation routine validates the date of the last appointment that the patient had at the pharmacy.

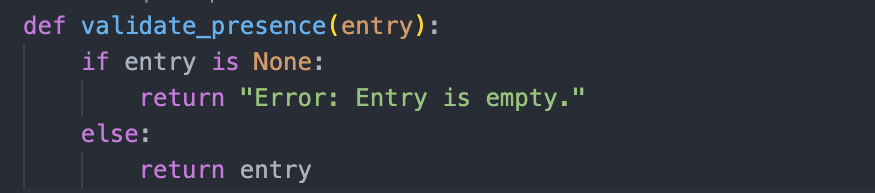


This Validation routine validates whether the Pharmacist has a bachelors or master’s degree achieved at University.

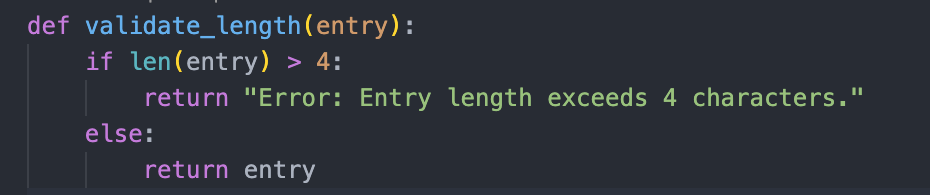
This validation routine Validates that the diagnosis is entered and is the correct length.

**Validation routines for a GUI System**

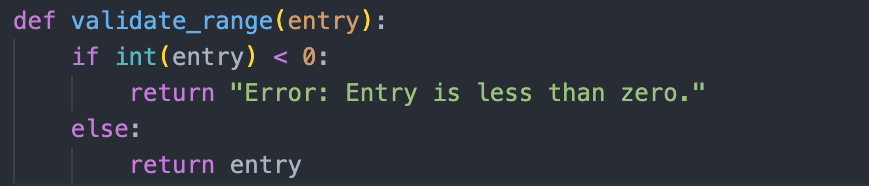
**Presence**

****

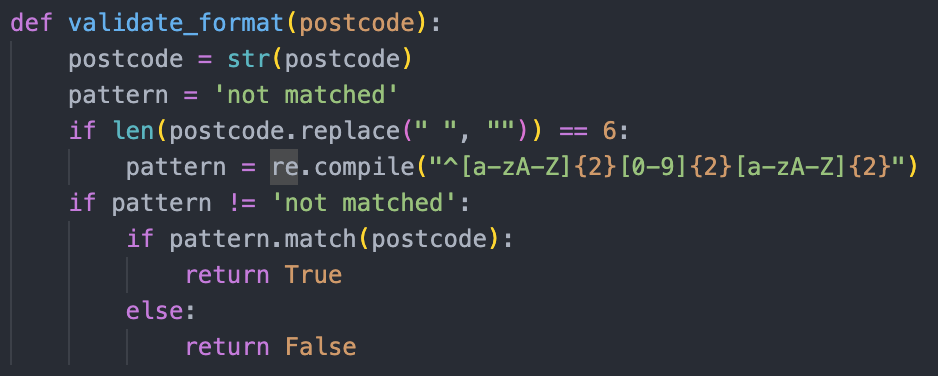
**Length**

****

**Range**

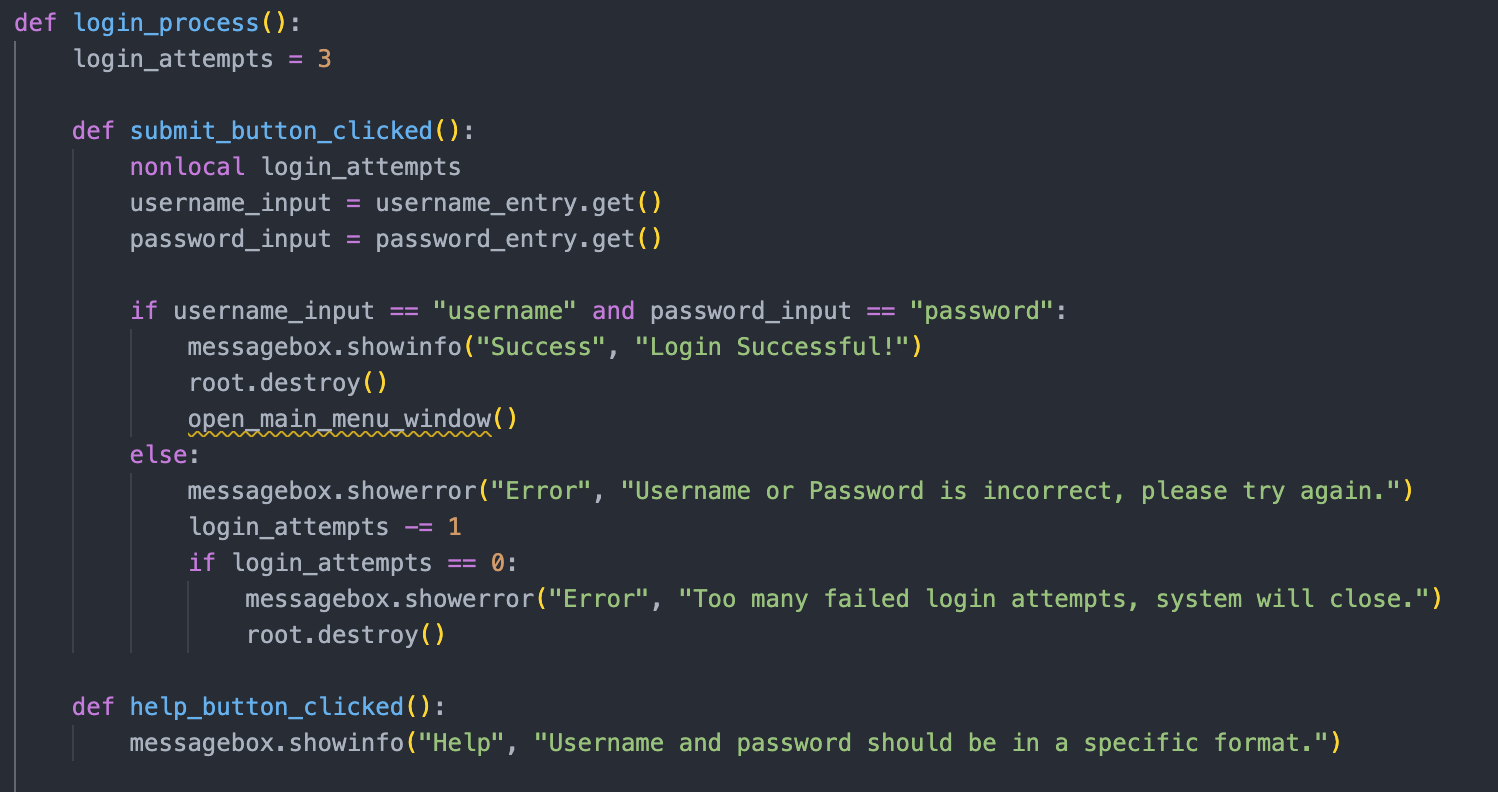
****

**Format**

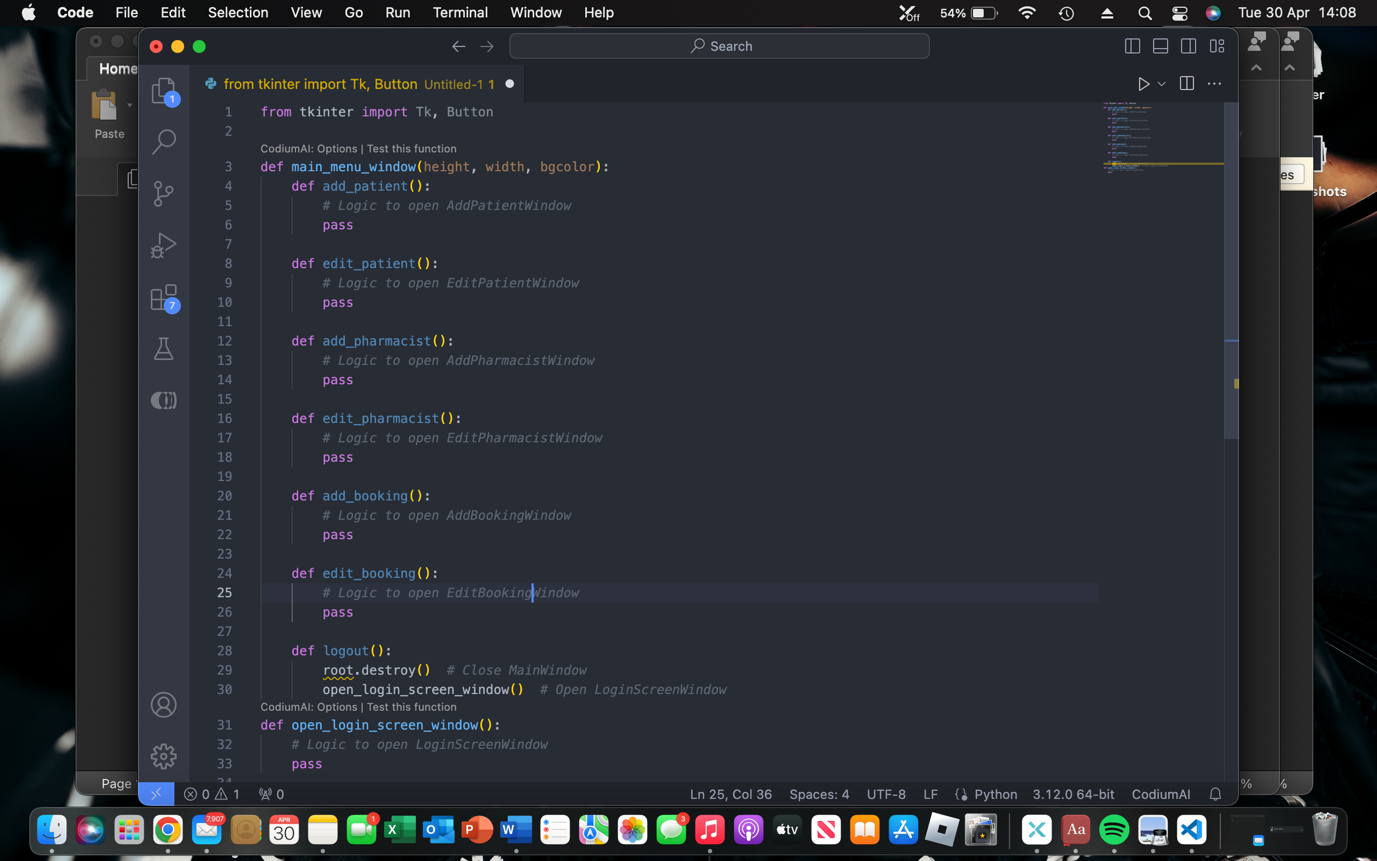
****

**Processing Routine**

For my proposed solution my program will use the Event-Driven programming approach, where the flow is influenced by user actions such as input. My routines will respond to user interactions such as clicking buttons in order to fit a good GUI. Unlike a CLI, users in my solution will interact by selecting options or functions. This makes the user experience better and will allow for the pharmacy to require less training for the new system.



This is a processing routine for the login process of the system.



This processing routine is for the main menu window of the system.

**Note –** These routines are not final versions and may be subject to change due to the complexity of the programmed solution.