TABLE OF CONTENTS

## 1. INTRODUCTION

This project is a fully-functional prototype ASMIS system. It provides all of the functionality to demonstrate basic appointment booking and consultation capturing. The point of the exercise was to implement some amount of the system proposed in Unit 9, and specifically focus on a few aspects of cyber-security.

The main data of interest for hackers / cyber-criminals in such a system are the patient details and patient medical records. It is therefore of utmost importance to ensure that these data are secured as far as possible.

## 2. SECURITY FEATURES

For this demo, and building on the submission of Unit 9, I decided to generally incorporate and demonstrate the following security features:

1. A strict password policy (at least 8 characters long, consisting of at least 1 upper-case, 1 lower-case, 1 special and 1 numeric character) that would help mitigate the threat of spoofing via brute force / dictionary attacks

2. Storing passwords in the form of salted hashes in the database. This helps prevent information disclosure if the database were to be breached; coupled with the strict password policy, passwords with salted encryption provide extra security, making it difficult for hackers to do a hash look up.

3. Extensive logging of all key actions in the system along with exact date and time e.g. user login and logout activities, to appointment booking, patient detail viewing, patient consultation notes viewing, updating details etc. This fosters an environment of responsible use; a malicious user (especially Admin user) would be far more reluctant to engage in abusive activities, knowing that every action is recorded as having been carried out by that user.

4. Admin user permission levels. For this prototype, a basic access control system was provided whereby Admin users are either permission level 2 i.e. Super Admin user, who will have over-sight over other Admins, or a normal Admin user with permission level 1 who has to obtain authorization from a Super Admin for key activities that may be susceptible to abuse. In practice a more extensive access control system would need to be implemented to systematically control access to a variety of activities.

5. Multi-factor authentication to foster an environment of consent and over-sight for key activities. i.e.:

    - When a Physician tries to access a patient's previous consultation notes, an OTP is sent to the relevant patient, and the Physician will only be granted access with this OTP.
    - When an Admin user tries to view or edit a patient's details, an OTP is sent to the relevant patient, and the Admin user will only be granted access with this OTP.

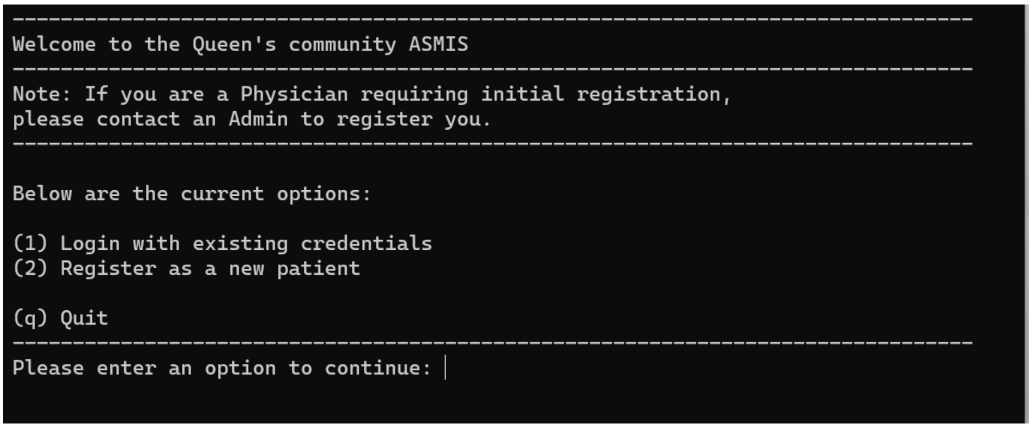
- When a normal Admin user (permission level 1) tries to register a new Physician account or to Cancel an appointment, an OTP is sent to the super Admin user (permission level 2), and the normal Admin user will only be granted access to these actions with this OTP.

6. SQL injection protection by using standard SQL querying libraries and using safe query practices in the form of safe parameter substitution.

The following sections provide results of tests carried out on the system to demonstrate each of the security features above, noting that only a subset of the actual system will be shown below for brevity; the reader is encouraged to try out the application.

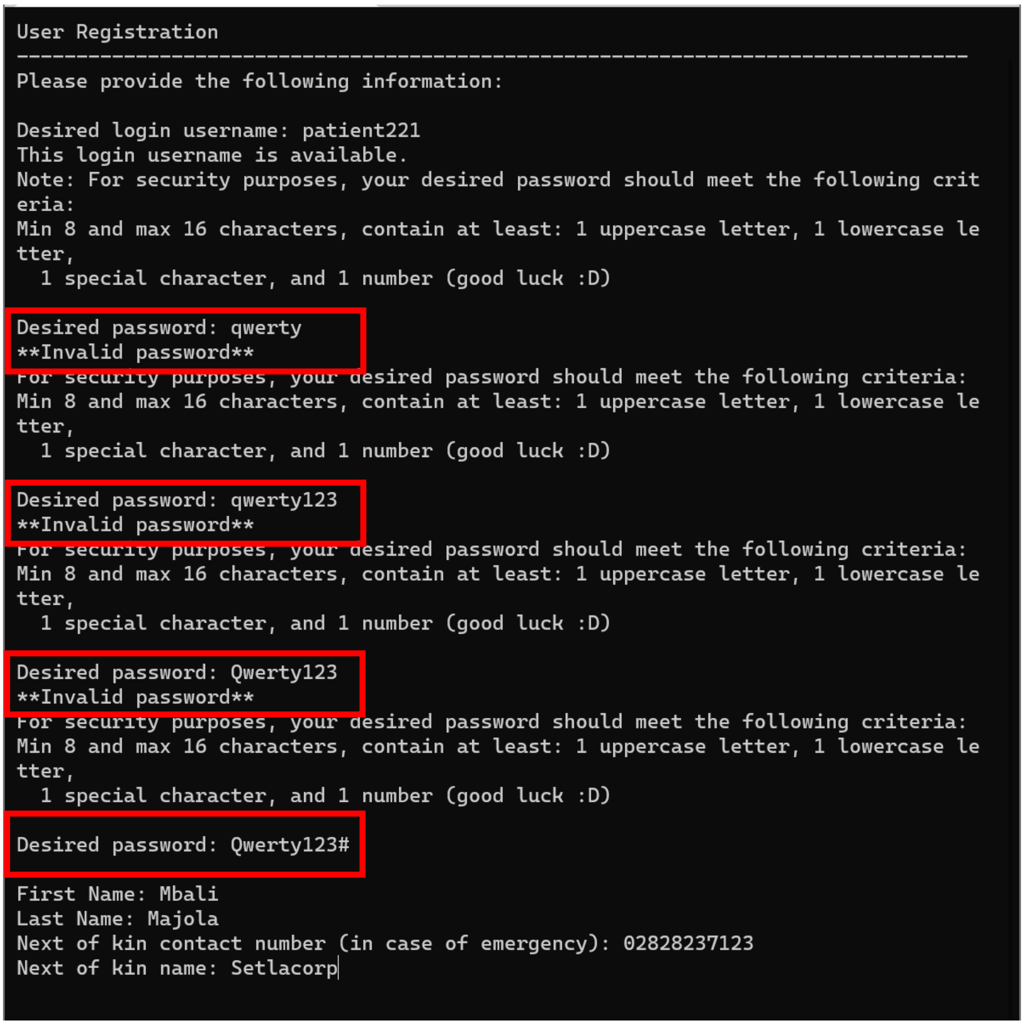
# 3. TESTING AND RESULTS

## 3.1. Strict Password Policy



```
--------------------------------------------------------------------------------
Welcome to the Queen's community ASMIS
--------------------------------------------------------------------------------
Note: If you are a Physician requiring initial registration,
please contact an Admin to register you.
--------------------------------------------------------------------------------

Below are the current options:

(1) Login with existing credentials
(2) Register as a new patient

(q) Quit
--------------------------------------------------------------------------------
Please enter an option to continue: 
```

**Figure 1: Initial welcoming page of the ASMIS**



```
User Registration
--------------------------------------------------------------------------------
Please provide the following information:

Desired login username: patient221
This login username is available.
Note: For security purposes, your desired password should meet the following crit
eria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase le
tter,
  1 special character, and 1 number (good luck :D)

Desired password: qwerty
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase le
tter,
  1 special character, and 1 number (good luck :D)

Desired password: qwerty123
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase le
tter,
  1 special character, and 1 number (good luck :D)

Desired password: Qwerty123
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase le
tter,
  1 special character, and 1 number (good luck :D)

Desired password: Qwerty123#

First Name: Mbali
Last Name: Majola
Next of kin contact number (in case of emergency): 02828237123
Next of kin name: Setlacorp
```

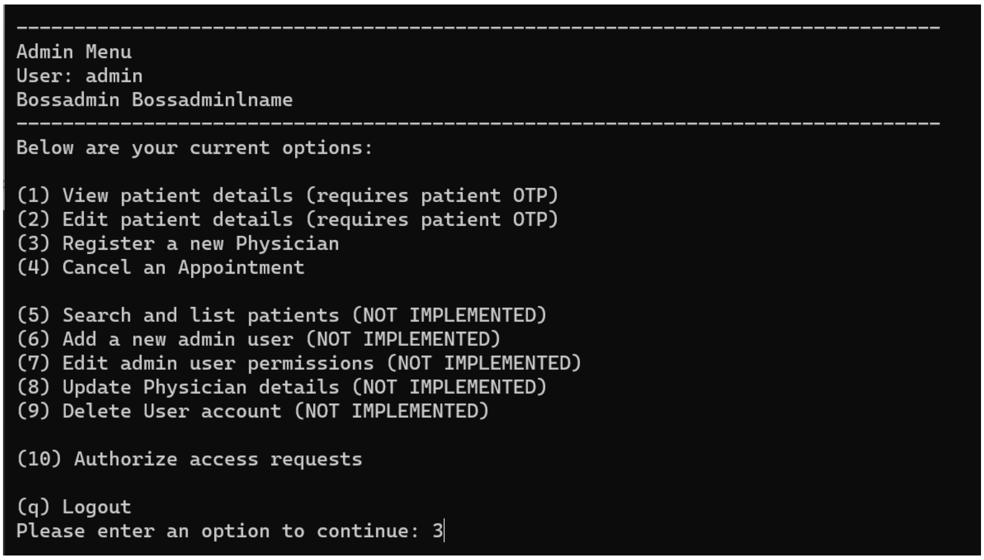
**Figure 2: Patient user registration; user tried to insert multiple uncompliant passwords, which were rejected.**

The user is initially greeted by the welcoming screen shown in Figure 1. The system currently only allows patients to register in this way. Physicians are registered by an Admin user. When the user proceeds to register by selecting option (2), they are met with the user registration screen shown in Figure 2. Looking closely at the figure, it can be seen that:

- The user selected a login username of “patient221” which was accepted because there was no other users with this username on the system currently;
- The user was asked to insert a strong password of at least 8 characters with at least 1 upper-case, 1 lower-case, 1 special, and 1 numeric character.

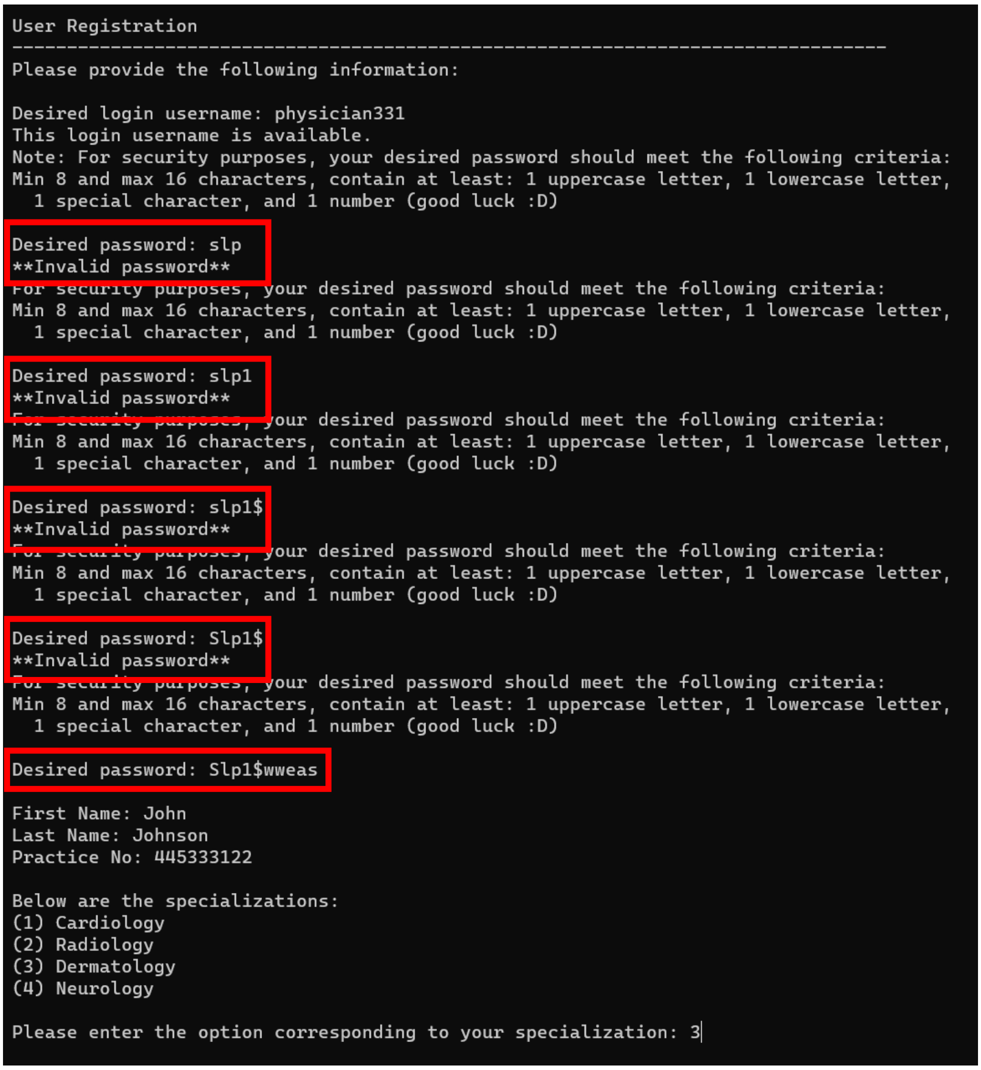
- To test whether or not the system rejects uncompliant passwords, Figure 2 shows a series of uncompliant passwords were entered, and the system rejected them each time
- The system finally accepted the password “Qwerty123#” because it met the password criteria.

Note that the code re-uses the exact same functions to carry out registration of users (patients or physicians) as well as to update user details, and therefore the password policy enforcement is applied, and works, uniformly across all the different functions. As a demonstration, Figure 3 shows the initial Super Admin user menu, and when the Admin user opts to register a new Physician as shown in Figure 4, a very similar set of initial prompts to those shown in Figure 2 are displayed (to capture generic User details). The re-use of code ensures that a uniform policy is applied.



```
--------------------------------------------------------------------------------
Admin Menu
User: admin
Bossadmin Bossadminlname
--------------------------------------------------------------------------------
Below are your current options:

(1) View patient details (requires patient OTP)
(2) Edit patient details (requires patient OTP)
(3) Register a new Physician
(4) Cancel an Appointment

(5) Search and list patients (NOT IMPLEMENTED)
(6) Add a new admin user (NOT IMPLEMENTED)
(7) Edit admin user permissions (NOT IMPLEMENTED)
(8) Update Physician details (NOT IMPLEMENTED)
(9) Delete User account (NOT IMPLEMENTED)

(10) Authorize access requests

(q) Logout
Please enter an option to continue: 3
```

**Figure 3: Admin User menu screen.**

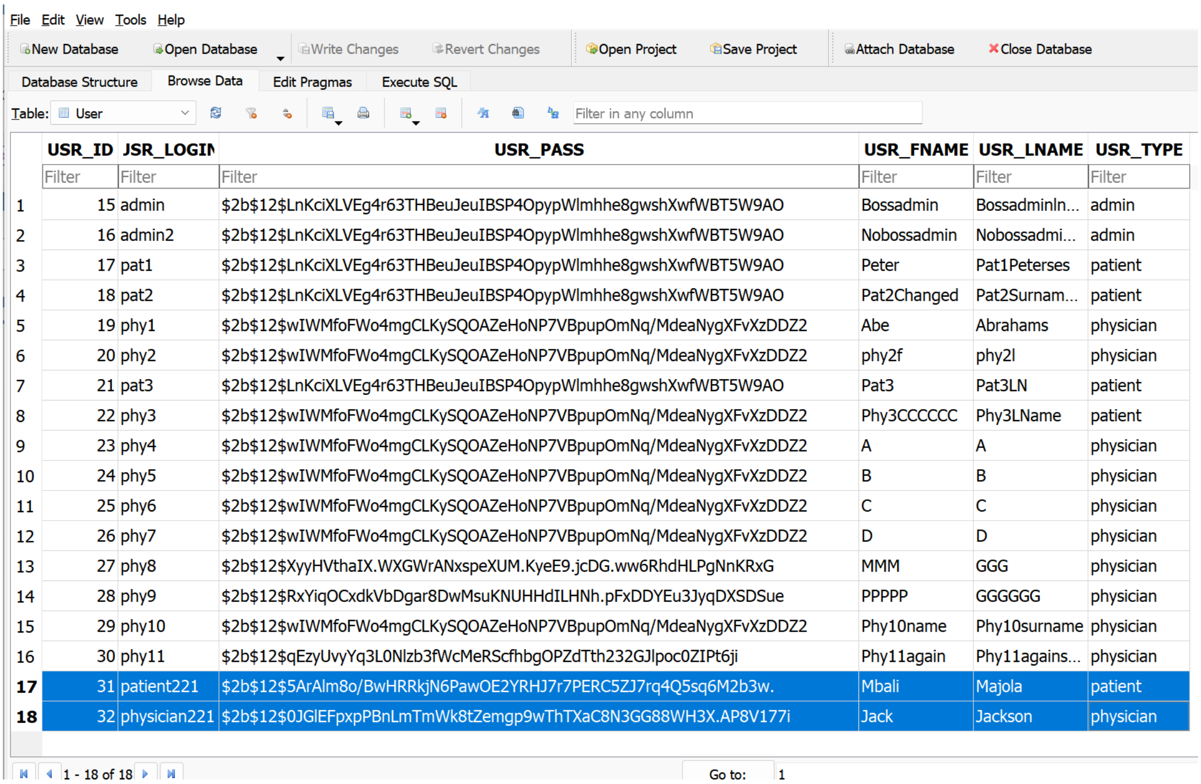


```
User Registration
--------------------------------------------------------------------------------
Please provide the following information:

Desired login username: physician331
This login username is available.
Note: For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase letter,
  1 special character, and 1 number (good luck :D)

Desired password: slp
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase letter,
  1 special character, and 1 number (good luck :D)

Desired password: slp1
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase letter,
  1 special character, and 1 number (good luck :D)

Desired password: slp1$
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase letter,
  1 special character, and 1 number (good luck :D)

Desired password: Slp1$
**Invalid password**
For security purposes, your desired password should meet the following criteria:
Min 8 and max 16 characters, contain at least: 1 uppercase letter, 1 lowercase letter,
  1 special character, and 1 number (good luck :D)

Desired password: Slp1$wweas

First Name: John
Last Name: Johnson
Practice No: 445333122

Below are the specializations:
(1) Cardiology
(2) Radiology
(3) Dermatology
(4) Neurology

Please enter the option corresponding to your specialization: 3
```

**Figure 4: Patient user registration; user tried to insert multiple uncompliant passwords, which were rejected.**

## 3.2. Storing Password As Salted Hashes

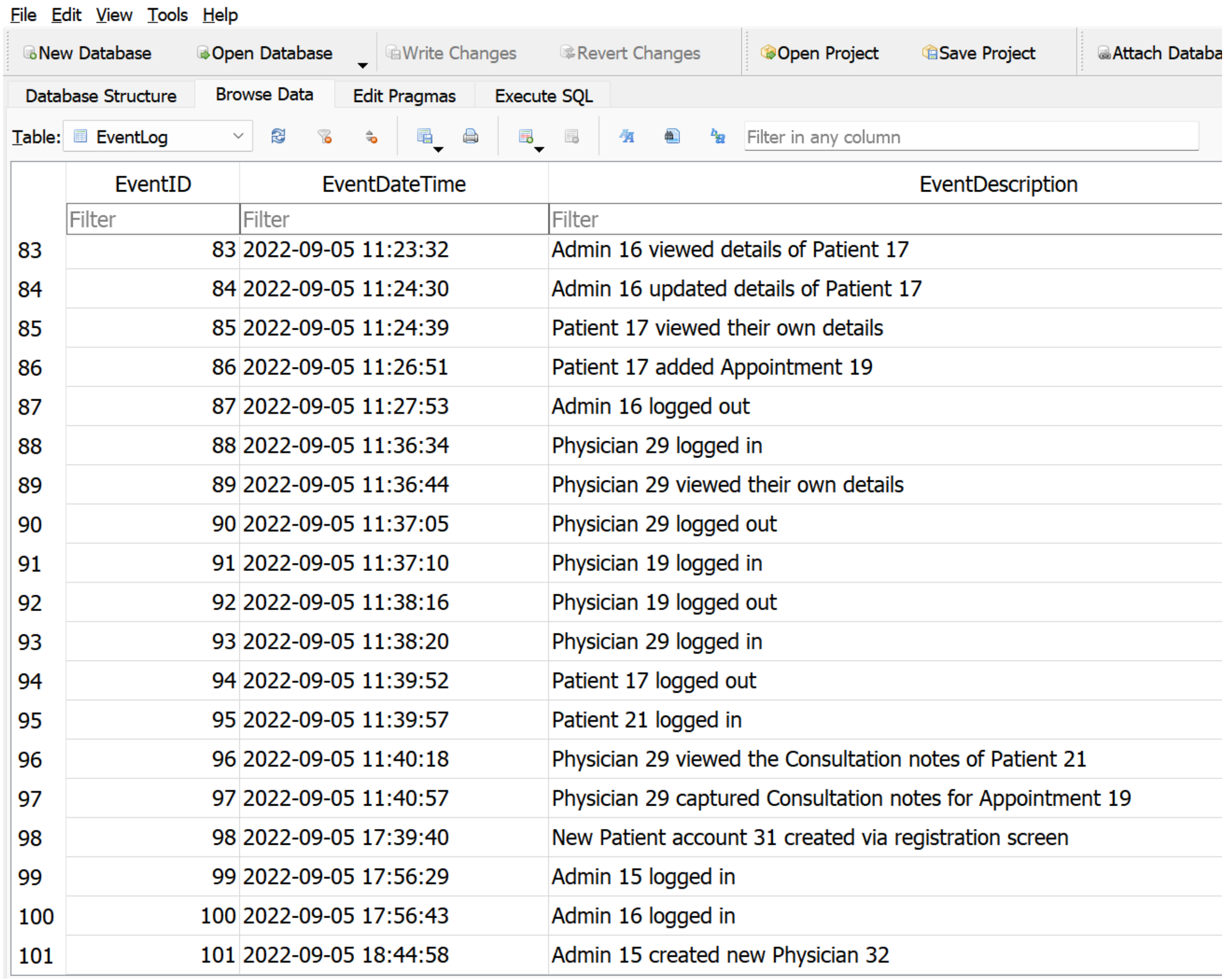
Referring to the procedures of registering a new Patient or Physician described in the previous section, once all the details of the User are captured, the password is encrypted as a salted hash and it is the salted hash that is stored in the database. Figure 5 shows an third-party SQLite viewer showing the User table which stores all generic User information (but not information specific to Patients, Physicians or Admins). It can be seen in the figure that the passwords of the Patient and Physician registered, as well as those of all previously registered Users, have been stored as salted hashes in the “USR_PASS” column of the table.



| | USR_ID | JSR_LOGIN | USR_PASS | USR_FNAME | USR_LNAME | USR_TYPE |
|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 15 | admin | $2b$12$LnKciXLVEg4r63THBeuJeuIBSP4OpypWlmhhe8gwshXwfWBT5W9AO | Bossadmin | Bossadminln... | admin |
| 2 | 16 | admin2 | $2b$12$LnKciXLVEg4r63THBeuJeuIBSP4OpypWlmhhe8gwshXwfWBT5W9AO | Nobossadmin | Nobossadmi... | admin |
| 3 | 17 | pat1 | $2b$12$LnKciXLVEg4r63THBeuJeuIBSP4OpypWlmhhe8gwshXwfWBT5W9AO | Peter | Pat1Peterses | patient |
| 4 | 18 | pat2 | $2b$12$LnKciXLVEg4r63THBeuJeuIBSP4OpypWlmhhe8gwshXwfWBT5W9AO | Pat2Changed | Pat2Surnam... | patient |
| 5 | 19 | phy1 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | Abe | Abrahams | physician |
| 6 | 20 | phy2 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | phy2f | phy2l | physician |
| 7 | 21 | pat3 | $2b$12$LnKciXLVEg4r63THBeuJeuIBSP4OpypWlmhhe8gwshXwfWBT5W9AO | Pat3 | Pat3LN | patient |
| 8 | 22 | phy3 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | Phy3CCCCCC | Phy3LName | patient |
| 9 | 23 | phy4 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | A | A | physician |
| 10 | 24 | phy5 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | B | B | physician |
| 11 | 25 | phy6 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | C | C | physician |
| 12 | 26 | phy7 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | D | D | physician |
| 13 | 27 | phy8 | $2b$12$XyyHVthaIX.WXGWrANxspeXUM.KyeE9.jcDG.ww6RhdHLPgNnKRxG | MMM | GGG | physician |
| 14 | 28 | phy9 | $2b$12$RxYiqOCxdkVbDgar8DwMsuKNUHHdILHNh.pFxDDYEu3JyqDXSDSue | PPPPP | GGGGGG | physician |
| 15 | 29 | phy10 | $2b$12$wIWMfoFWo4mgCLKySQOAZeHoNP7VBpupOmNq/MdeaNygXFvXzDDZ2 | Phy10name | Phy10surname | physician |
| 16 | 30 | phy11 | $2b$12$qEzyUvyYq3L0Nlzb3fWcMeRScfhbgOPZdTth232GJlpoc0ZIPt6ji | Phy11again | Phy11agains... | physician |
| 17 | 31 | patient221 | $2b$12$5ArAlm8o/BwHRRkjN6PawOE2YRHJ7r7PERC5ZJ7rq4Q5sq6M2b3w. | Mbali | Majola | patient |
| 18 | 32 | physician221 | $2b$12$0JGlEFpxpPBnLmTmWk8tZemgp9wThTXaC8N3GG88WH3X.AP8V177i | Jack | Jackson | physician |

**Figure 5: SQLite Backend showing (highlighted in blue) the Patient and Physician users created/registered in the previous section, and showing that passwords are stored as salted hashes.**

## 3.3. Logging of Key Actions



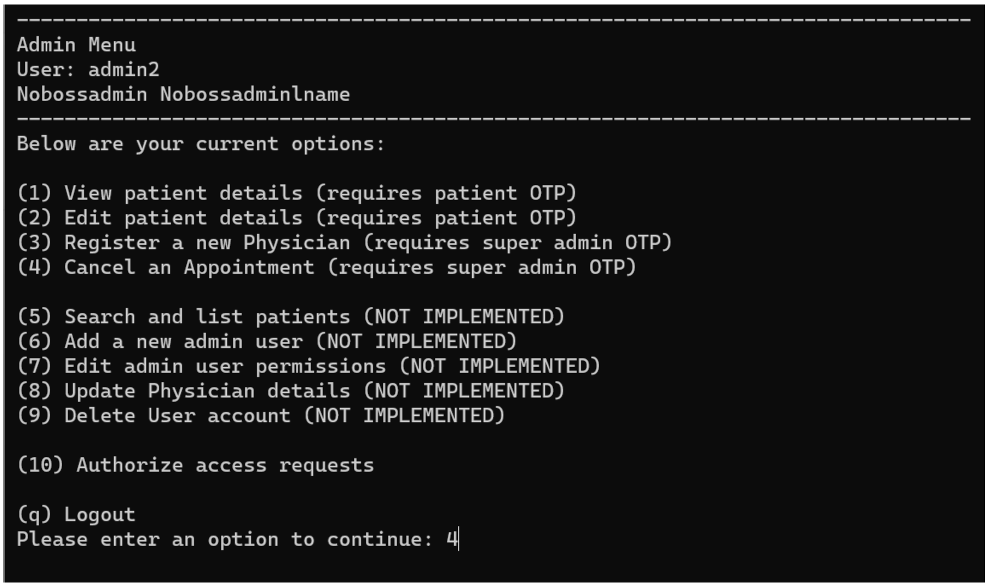
| | EventID | EventDateTime | EventDescription |
|---|---|---|---|
| | Filter | Filter | Filter |
| 83 | 83 | 2022-09-05 11:23:32 | Admin 16 viewed details of Patient 17 |
| 84 | 84 | 2022-09-05 11:24:30 | Admin 16 updated details of Patient 17 |
| 85 | 85 | 2022-09-05 11:24:39 | Patient 17 viewed their own details |
| 86 | 86 | 2022-09-05 11:26:51 | Patient 17 added Appointment 19 |
| 87 | 87 | 2022-09-05 11:27:53 | Admin 16 logged out |
| 88 | 88 | 2022-09-05 11:36:34 | Physician 29 logged in |
| 89 | 89 | 2022-09-05 11:36:44 | Physician 29 viewed their own details |
| 90 | 90 | 2022-09-05 11:37:05 | Physician 29 logged out |
| 91 | 91 | 2022-09-05 11:37:10 | Physician 19 logged in |
| 92 | 92 | 2022-09-05 11:38:16 | Physician 19 logged out |
| 93 | 93 | 2022-09-05 11:38:20 | Physician 29 logged in |
| 94 | 94 | 2022-09-05 11:39:52 | Patient 17 logged out |
| 95 | 95 | 2022-09-05 11:39:57 | Patient 21 logged in |
| 96 | 96 | 2022-09-05 11:40:18 | Physician 29 viewed the Consultation notes of Patient 21 |
| 97 | 97 | 2022-09-05 11:40:57 | Physician 29 captured Consultation notes for Appointment 19 |
| 98 | 98 | 2022-09-05 17:39:40 | New Patient account 31 created via registration screen |
| 99 | 99 | 2022-09-05 17:56:29 | Admin 15 logged in |
| 100 | 100 | 2022-09-05 17:56:43 | Admin 16 logged in |
| 101 | 101 | 2022-09-05 18:44:58 | Admin 15 created new Physician 32 |

**Figure 6: SQLite Backend showing the latest entries in the EventLog table. Each entry has a unique ID (EventID), a timestamp (EventDateTime) and a description of the event that took place (EventDescription).**

Key actions in the system are logged in the EventLog table in the database. In practice, logs should ideally be placed in multiple locations and encrypted to prevent repudiation attacks, and to keep foster an environment of responsible use and over-sight. Figure Figure 6 shows the the EventLog table in the SQLite database. It can be seen that each entry has a unique ID (EventID), a timestamp (EventDateTime) and a description of the event that took place (EventDescription). These EventLog entries are created immediately after each relevant event takes place. Actions logged include: logging in or out, creation of new Patients or Physicians, viewing or update of Patient details, viewing or creation of Patient consultation notes, etc.

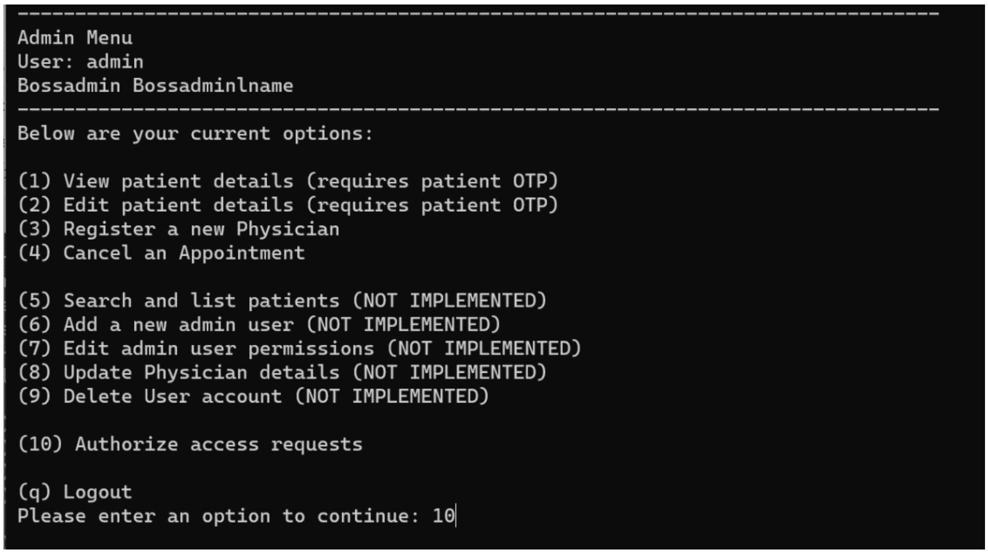
### 3.4. Admin User Permissions

For this prototype, there are two Admin user permission levels: level 2 which specifies a Super Admin user and level 1 constitutes a normal Admin user. In this prototype, the difference between a super and normal Admin user is that a normal Admin may only register Physicians and/or cancel appointments with Multi-factor Authentication (MFA) in the form of a One-Time-Pin  (OTP) authorization (more on this in the next subsection) of a super Admin user, while a super Admin user does not have this limitation.



```
--------------------------------------------------------------------------------
Admin Menu
User: admin2
Nobossadmin Nobossadminlname
--------------------------------------------------------------------------------
Below are your current options:

(1) View patient details (requires patient OTP)
(2) Edit patient details (requires patient OTP)
(3) Register a new Physician (requires super admin OTP)
(4) Cancel an Appointment (requires super admin OTP)

(5) Search and list patients (NOT IMPLEMENTED)
(6) Add a new admin user (NOT IMPLEMENTED)
(7) Edit admin user permissions (NOT IMPLEMENTED)
(8) Update Physician details (NOT IMPLEMENTED)
(9) Delete User account (NOT IMPLEMENTED)

(10) Authorize access requests

(q) Logout
Please enter an option to continue: 4
```

**Figure 7: A normal Admin user menu screen. Options 3 and 4 clearly indicate that these actions will require OTP authorization by a super Admin user.**



```
-------------------------------------------------------------------------------
Admin Menu
User: admin
Bossadmin Bossadminlname
-------------------------------------------------------------------------------
Below are your current options:

(1) View patient details (requires patient OTP)
(2) Edit patient details (requires patient OTP)
(3) Register a new Physician
(4) Cancel an Appointment

(5) Search and list patients (NOT IMPLEMENTED)
(6) Add a new admin user (NOT IMPLEMENTED)
(7) Edit admin user permissions (NOT IMPLEMENTED)
(8) Update Physician details (NOT IMPLEMENTED)
(9) Delete User account (NOT IMPLEMENTED)

(10) Authorize access requests

(q) Logout
Please enter an option to continue: 10
```

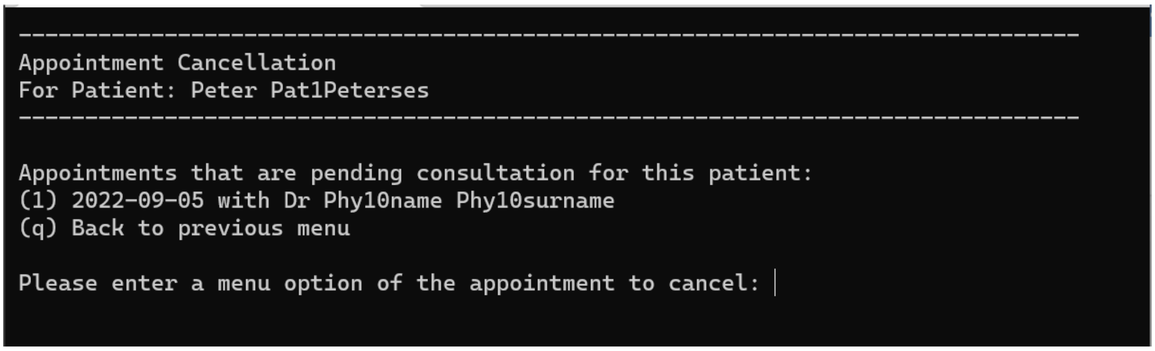
**Figure 8: A super Admin user menu screen. Options 3 and 4 can be executed as is.**

A demonstration of these actions with the MFA OTP will be provided in the next sub-section as they are directly relevant to that sub-section.

## 3.5. Multi-Factor Authentication for Consent and Over-Sight

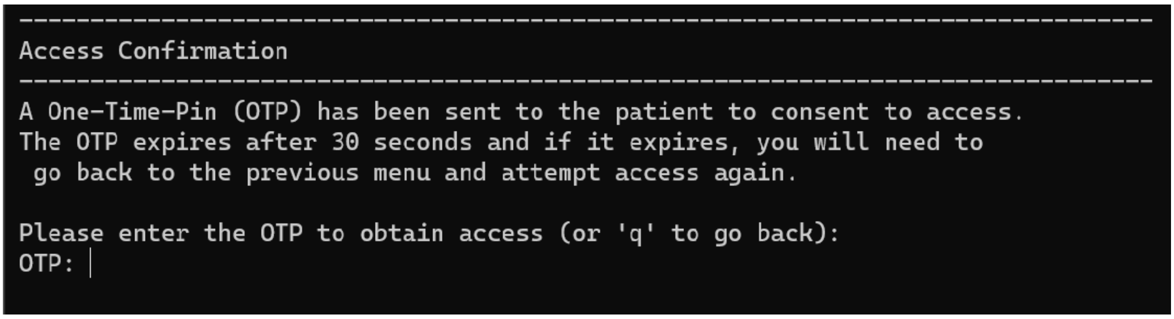
Several actions are protected via MFA OTP authorization. Some of these are demonstrated in this section.

The first that will be tested and demonstrated is the cancellation of an appointment by a normal Admin user, which requires OTP authorization by a super Admin user. In Figures 9 through 13, it can be seen that a normal Admin user tried to cancel an Appointment; before finalizing the cancellation, the system displayed an OTP challenge which rejects incorrect OTP values and expires after 30 seconds; the OTP is displayed to the interface of the super Admin user via menu option 11 on the Admin user menu screen.



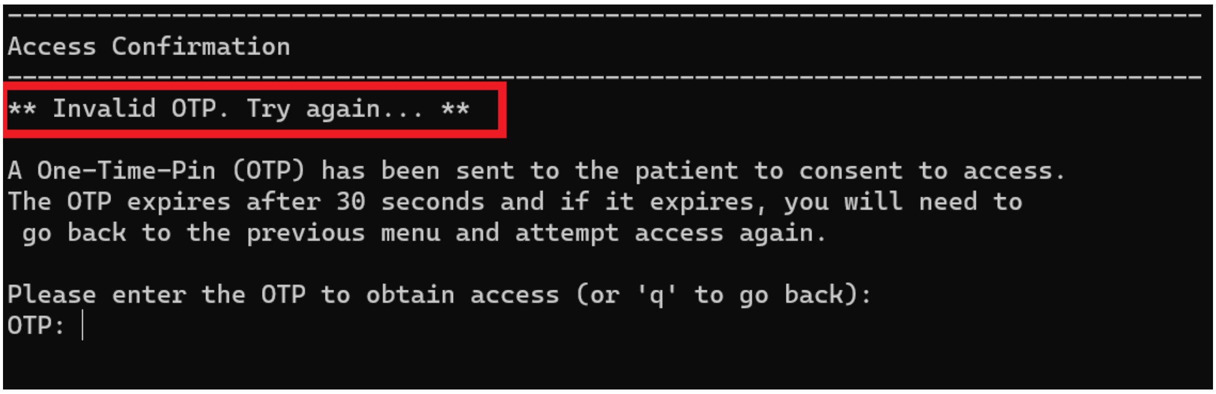
```
--------------------------------------------------------------------------------
Appointment Cancellation
For Patient: Peter Pat1Peterses
--------------------------------------------------------------------------------

Appointments that are pending consultation for this patient:
(1) 2022-09-05 with Dr Phy10name Phy10surname
(q) Back to previous menu

Please enter a menu option of the appointment to cancel:
```

**Figure 9: Appointment cancellation screen accessed by a normal Admin user to cancel an appointment of a specific patient.**



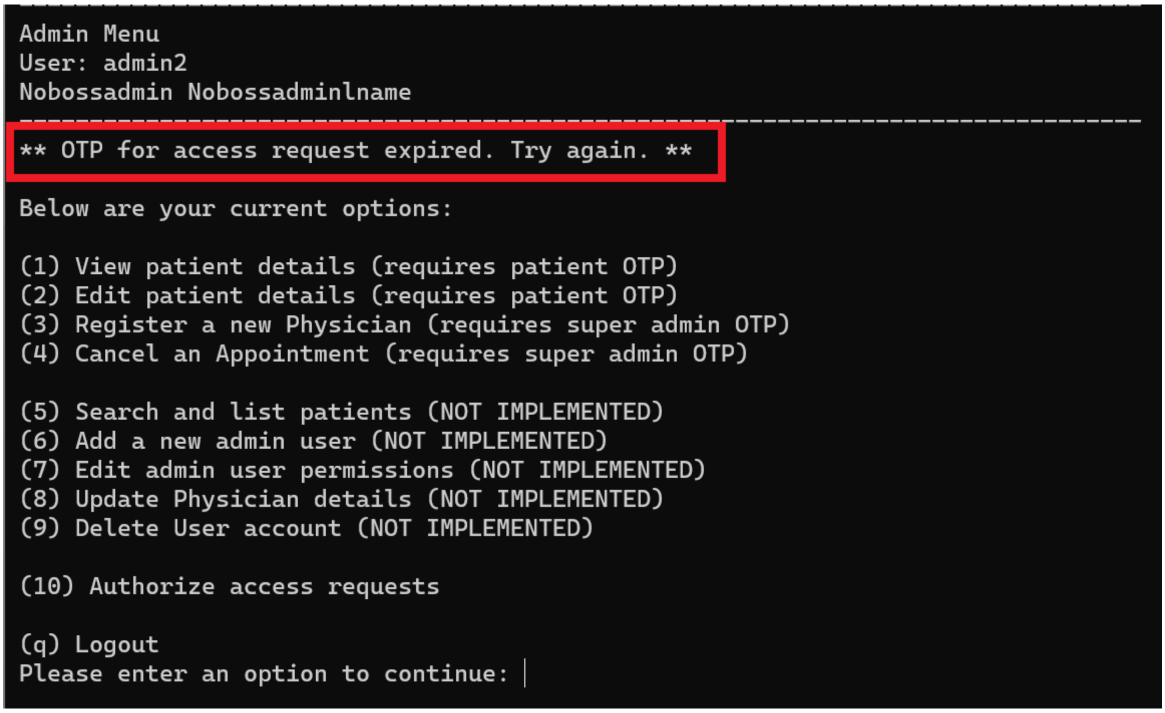
```
--------------------------------------------------------------------------------
Access Confirmation
--------------------------------------------------------------------------------
A One-Time-Pin (OTP) has been sent to the patient to consent to access.
The OTP expires after 30 seconds and if it expires, you will need to
 go back to the previous menu and attempt access again.

Please enter the OTP to obtain access (or 'q' to go back):
OTP:
```

**Figure 10: Access confirmation screen after a normal Admin user tried to cancel an appointment. The normal Admin will have to obtain the OTP issued to the super Admin user before the action is accepted.**



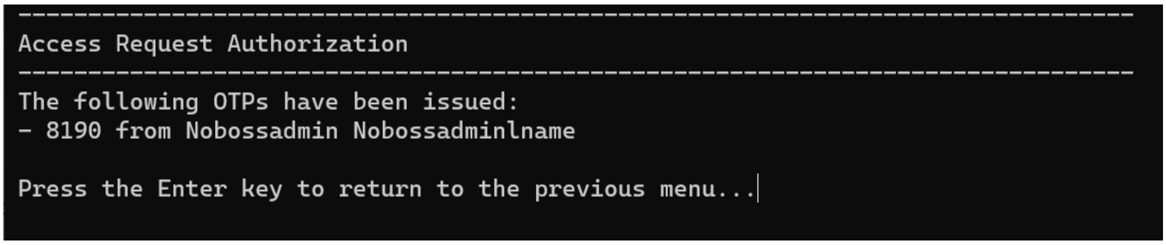
```
--------------------------------------------------------------------------------
Access Confirmation
--------------------------------------------------------------------------------
** Invalid OTP. Try again... **

A One-Time-Pin (OTP) has been sent to the patient to consent to access.
The OTP expires after 30 seconds and if it expires, you will need to
 go back to the previous menu and attempt access again.

Please enter the OTP to obtain access (or 'q' to go back):
OTP:
```

**Figure 11: Invalid OTP number entered and rejected by the system.**



```
--------------------------------------------------------------------------------
Admin Menu
User: admin2
Nobossadmin Nobossadminlname
--------------------------------------------------------------------------------
** OTP for access request expired. Try again. **

Below are your current options:

(1) View patient details (requires patient OTP)
(2) Edit patient details (requires patient OTP)
(3) Register a new Physician (requires super admin OTP)
(4) Cancel an Appointment (requires super admin OTP)

(5) Search and list patients (NOT IMPLEMENTED)
(6) Add a new admin user (NOT IMPLEMENTED)
(7) Edit admin user permissions (NOT IMPLEMENTED)
(8) Update Physician details (NOT IMPLEMENTED)
(9) Delete User account (NOT IMPLEMENTED)

(10) Authorize access requests

(q) Logout
Please enter an option to continue:
```

**Figure 12: The OTP expired and the system automatically rejects the action and send the normal Admin user back to the normal Admin user menu screen.**

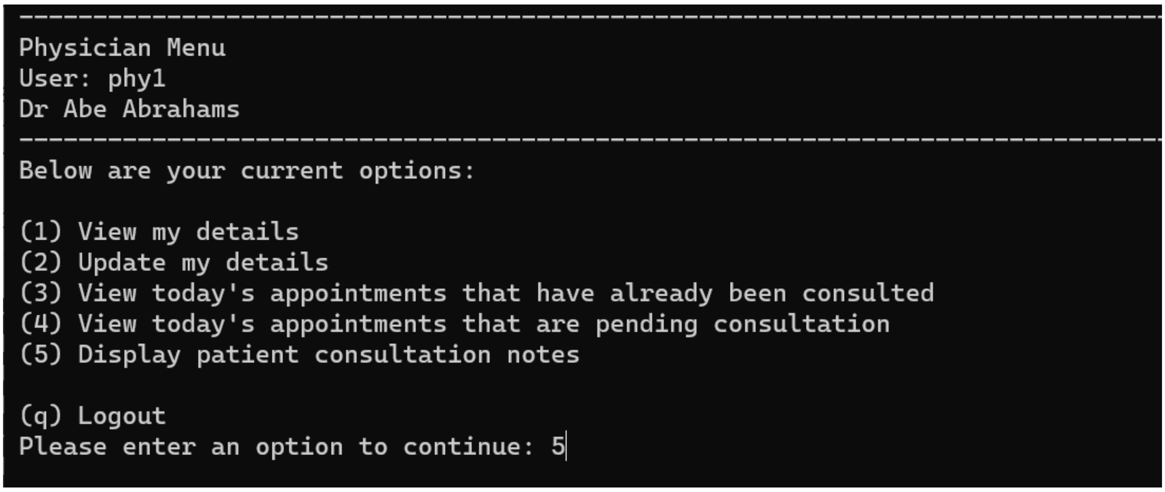


```
--------------------------------------------------------------------------------
Access Request Authorization
--------------------------------------------------------------------------------
The following OTPs have been issued:
- 8190 from Nobossadmin Nobossadminlname

Press the Enter key to return to the previous menu...
```

**Figure 13: Access Request Authorization screen of the super Admin user. The OTP along with the details of the requesting user are displayed on this screen to the super Admin user.**

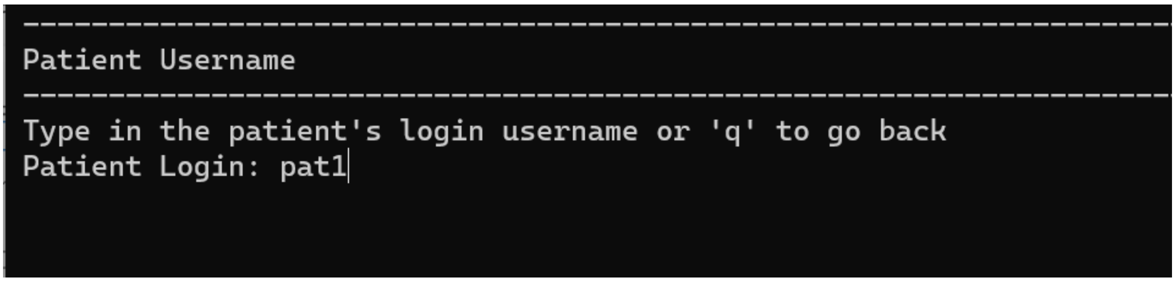
Another use case of an OTP being required is when either an Admin user tries to view of update the details of a Patient, or a Physician user tries to view previous Consultation notes of a Patient. In both cases, an OTP is issued to the Patient, and this OTP is required before any of these actions can be carried out. For brevity, below test results of a Physician attempting to access the Consultation notes of patient are displayed.

In Figures 14 through 19, a Physician user attempts to access the previous consultation notes of a given patient. The system then presents the Physician with an OTP challenge with a timed expiration of 30 seconds, without which the notes are not displayed. Only once a valid OTP is entered is the information displayed to the Physician.



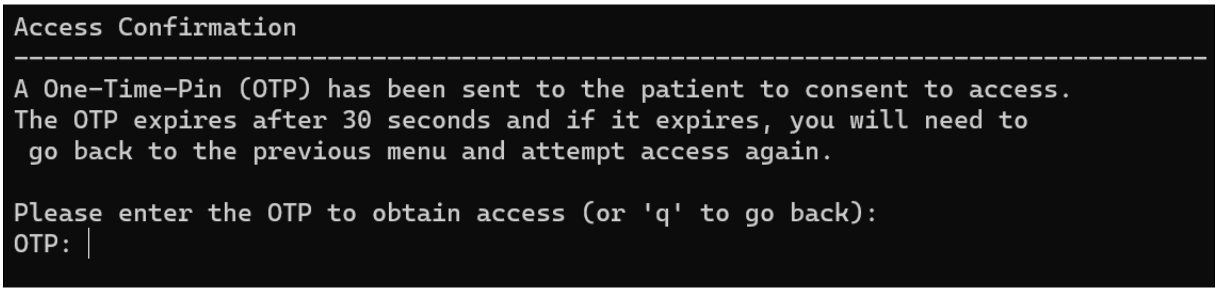
```
-------------------------------------------------------------------------------
Physician Menu
User: phy1
Dr Abe Abrahams
-------------------------------------------------------------------------------
Below are your current options:

(1) View my details
(2) Update my details
(3) View today's appointments that have already been consulted
(4) View today's appointments that are pending consultation
(5) Display patient consultation notes

(q) Logout
Please enter an option to continue: 5
```

**Figure 14: Physician menu screen. Option 5 is one way that a Physician can view previous Consultation notes of a given patient.**



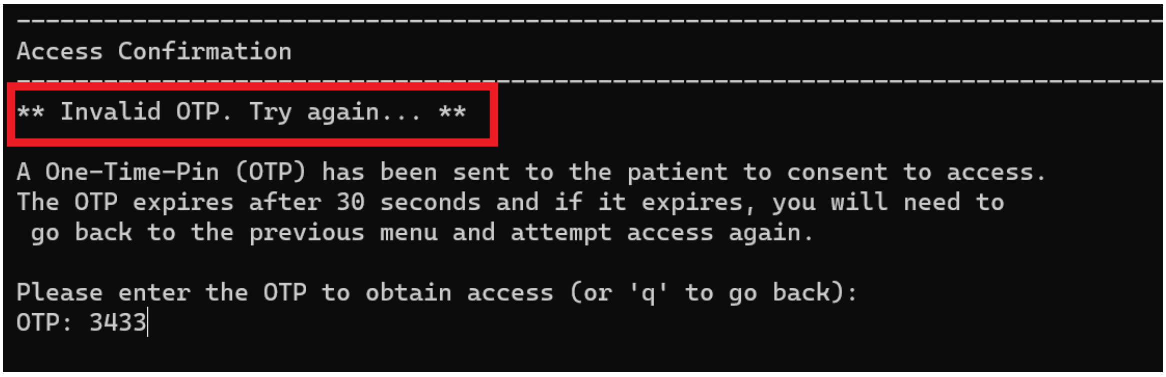
```
-------------------------------------------------------------------------------
Patient Username
-------------------------------------------------------------------------------
Type in the patient's login username or 'q' to go back
Patient Login: pat1
```

**Figure 15: Physician has selected option 5 and is asked to specify the Patient's username whose previous Consultation notes should be displayed.**



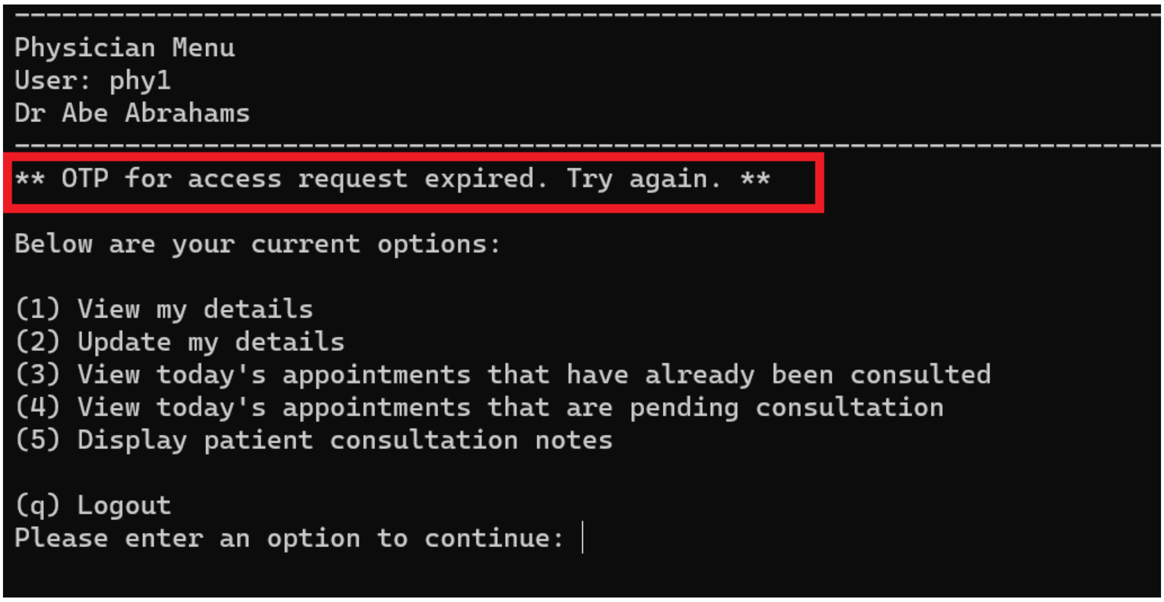
```
Access Confirmation
--------------------------------------------------------------------------------
A One-Time-Pin (OTP) has been sent to the patient to consent to access.
The OTP expires after 30 seconds and if it expires, you will need to
 go back to the previous menu and attempt access again.

Please enter the OTP to obtain access (or 'q' to go back):
OTP:
```

**Figure 16: Physician is presented with an OTP challenge before the Consultation notes are displayed.**



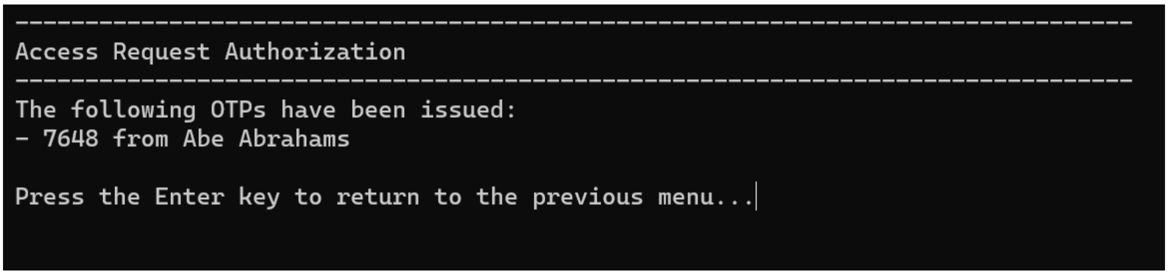
```
--------------------------------------------------------------------------------
Access Confirmation
--------------------------------------------------------------------------------
** Invalid OTP. Try again... **

A One-Time-Pin (OTP) has been sent to the patient to consent to access.
The OTP expires after 30 seconds and if it expires, you will need to
 go back to the previous menu and attempt access again.

Please enter the OTP to obtain access (or 'q' to go back):
OTP: 3433
```

**Figure 17: An invalid OTP was entered and the system rejected the action.**



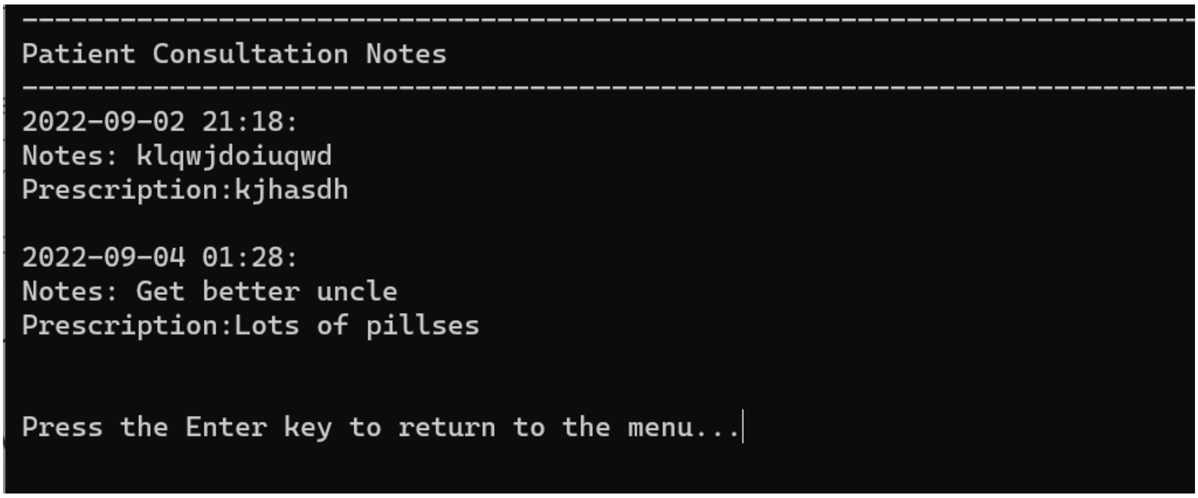
```
------------------------------------------------------------------------
Physician Menu
User: phy1
Dr Abe Abrahams
------------------------------------------------------------------------
** OTP for access request expired. Try again. **

Below are your current options:

(1) View my details
(2) Update my details
(3) View today's appointments that have already been consulted
(4) View today's appointments that are pending consultation
(5) Display patient consultation notes

(q) Logout
Please enter an option to continue:
```

**Figure 18: The OTP expired and the system automatically rejects the action and send the Physician back to the Physician user menu screen.**



```
--------------------------------------------------------------------------------
Access Request Authorization
--------------------------------------------------------------------------------
The following OTPs have been issued:
- 7648 from Abe Abrahams

Press the Enter key to return to the previous menu...
```

**Figure 19: The Patient's Access Authorization Request screen showing OTPs along with which user requested access.**

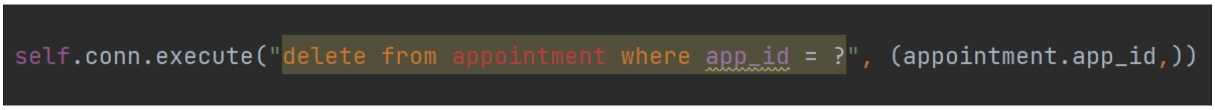


```
--------------------------------------------------------------------------------
Patient Consultation Notes
--------------------------------------------------------------------------------
2022-09-02 21:18:
Notes: klqwjdoiuqwd
Prescription:kjhasdh

2022-09-04 01:28:
Notes: Get better uncle
Prescription:Lots of pillses


Press the Enter key to return to the menu...
```

**Figure 20: The Patient's Consultation notes are displayed to the Physician once a valid OTP is entered.**

## 3.6. SQL Injection Protection Using Safe Parameter Substitution

Safe parameter substitution was also carried out by making use of the standard sqlite3 module in Python. An example of this safe parameter substitution can be seen in the Figure below.



```
self.conn.execute("delete from appointment where app_id = ?", (appointment.app_id,))
```

**Figure 21: Example of safe parameter substitution. The Appointment's unique ID (app_id) is substituted into the SQL query using the execute function.**