A red blue and black logo

Description automatically generated

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

**ASSIGNMENT**

**SEMESTER 202501**

| Programme (Year & Group) | : | DCS1S3 Grp1 |
| --- | --- | --- |
| Tutor Name | : | Ms. Joan Hau |
| Tutorial Group | : | 1 |
| Date Submitted | : | 28 April 2025 |

| No | Name | Student ID | Marks | | | Total Marks |
| --- | --- | --- | --- | --- | --- | --- |
| Section A | Section B | Section C |
| 1 | Tan Wy Hang | 2401976 |  |  |  |  |
| 2 | Tan Ying Nan | 2401920 |  |  |  |  |
| 3 | Chia Zhen Yang | 2402192 |  |  |  |  |

A red blue and black logo

Description automatically generated

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

Diploma in Computer Science (DCS)

Year 1 Semester 3

Group 1

**PLAGIARISM STATEMENT**

Read, complete, and sign this statement to be submitted with the written report.

**We confirm that we have read and shall comply with all the terms and conditions of TAR University of Management and Technology’s plagiarism policy.**

**We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.**

Declaration Statement Acknowledged by

| No | Name | Student ID | Signature | Date |
| --- | --- | --- | --- | --- |
| 1 | Tan Wy Hang | 2401976 |  | 28 Apr 2025 |
| 2 | Tan Ying Nan | 2401920 |  | 28 Apr 2025 |
| 3 | Chia Zhen Yang | 2402192 |  | 28 Apr 2025 |

A red blue and black logo

Description automatically generated

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

Diploma in Computer Science (DCS)

Year 1 Semester 3

Group 1

**PLAGIARISM STATEMENT FORM**

I, Name **Tan Wy Hang** Student ID **2401976**\_\_\_\_\_\_\_\_\_\_\_ Programme **DCS** Tutorial Group **1** confirm that the submitted work are all my own work and is in my own words.

I  **Tan Wy Hang** acknowledge the use of AI generative technology.

| Signature | : |  |
| --- | --- | --- |
| Date | : | 28 Apr 2025 |

A red blue and black logo

Description automatically generated

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

Diploma in Computer Science (DCS)

Year 1 Semester 3

Group 1

**PLAGIARISM STATEMENT FORM**

I, Name **Tan Ying Nan** Student ID **2401920**\_\_\_\_\_\_ Programme **DCS** Tutorial Group **1** confirm that the submitted work are all my own work and is in my own words.

I **Tan Ying Nan** acknowledge the use of AI generative technology.

| Signature | : |  |
| --- | --- | --- |
| Date | : | 28 Apr 2025 |

A red blue and black logo

Description automatically generated

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

Diploma in Computer Science (DCS)

Year 1 Semester 3

Group 1

**PLAGIARISM STATEMENT FORM**

I, Name **Chia Zhen Yang** Student ID **2402192**\_\_\_\_\_\_\_\_\_ Programme **DCS** Tutorial Group  **1** confirm that the submitted work are all my own work and is in my own words.

I **Chia Zhen Yang** acknowledge the use of AI generative technology.

| Signature | : |  |
| --- | --- | --- |
| Date | : | 28 Apr 2025 |

**Plagiarism Statement Form**

I, Name (Block Capitals) \_\_\_\_\_TAN WY HANG\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_2401976\_\_\_\_\_\_\_\_\_\_\_

Programme \_\_\_DCS\_\_\_\_\_\_Tutorial Group \_\_\_\_\_1\_\_\_\_\_\_\_\_confirm that the submitted

work are all my own work and is in my own words.

I (Student Name) acknowledge the use of AI generative technology.

Signature : ****

Date : \_\_28 / 04 / 2025\_\_

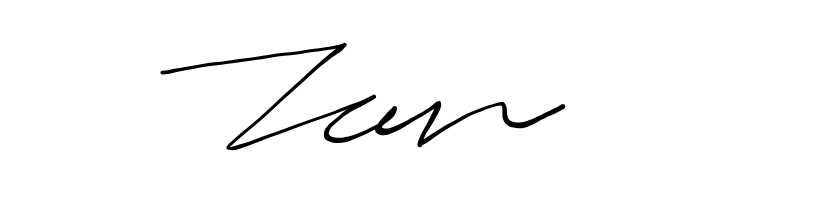
**Plagiarism Statement Form**

I, Name (Block Capitals) \_\_\_\_\_TAN YING NAN\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_2401920\_\_\_\_\_\_\_\_\_\_\_

Programme \_\_\_DCS\_\_\_\_\_\_Tutorial Group \_\_\_\_\_1\_\_\_\_\_\_\_\_confirm that the submitted

work are all my own work and is in my own words.

I (Student Name) acknowledge the use of AI generative technology.

Signature : ****

Date : \_\_28 / 04 / 2025\_\_\_

**Plagiarism Statement Form**

I, Name (Block Capitals) \_\_\_\_\_CHIA ZHEN YANG\_\_\_\_\_\_\_\_\_\_\_\_Student ID\_\_\_\_\_2402192\_\_\_\_\_\_\_\_\_\_\_

Programme \_\_\_DCS\_\_\_\_\_\_Tutorial Group \_\_\_\_\_1\_\_\_\_\_\_\_\_confirm that the submitted

work are all my own work and is in my own words.

I (Student Name) acknowledge the use of AI generative technology.

Signature : 

Date : \_\_28 / 04 / 2025\_\_\_

| No | **Student Name** | **Student ID** | **Final Marks** |
| --- | --- | --- | --- |
| **1** | Tan Wy Hang | 2401976 |  |
| **2** | Tan Ying Nan | 2401920 |  |
| **3** | Chia Zhen Yang | 2402192 |  |
| **4** |  |  |  |

**Assignment Assessment Rubrics**

| **Programme of Study** | D ( CS ) |
| --- | --- |
| **Tutorial Group** | Group ( 1 ) |

| **CLO 2** | Demonstrate an object-oriented program using appropriate programming fundamentals with regards to arrays, methods and exception handling. (P4, PLO3) | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section A: Program Output & Correctness - Individual Marks (25 Marks)** | | | | | | | | | |
| **Task** | **Weight** | **Max. Marks** | **Excellent** | **Good** | **Average** | **Poor** | **Very Poor** | **Scores**  **(Point x Weight)** | **Remarks** |
| **5** | **4** | **3** | **2** | **1** |
| **Completeness of the program** | **2.0** | **10** | Completed 100% of the functional requirements. All operations are implemented correctly. | Completed all of the functional requirements but implemented with minor errors. | Completed most of the functional requirements but implemented with errors. | Completed < 50% of the functional requirement and < 50% of the operations implemented correctly. | Did not complete any of the functional requirement at all | | **No** | **Marks** | | --- | --- | | **1** |  | | **2** |  | | **3** |  | | **4** |  | |  |
| **Exception Handling** | **1.0** | **5** | Output is correct with comprehensive exception handling. | Output is correct with appropriate exception handling. | Output is correct with little exception handling. | Output is somewhat correct with little proper exception handling. | Output is totally wrong with no proper exception handling. | | **No** | **Marks** | | --- | --- | | **1** |  | | **2** |  | | **3** |  | | **4** |  | |  |
| **Design of the Output** | **1.0** | **5** | Excellent formatted output. Program displays more than expected | Good formatted output and program displays as expected. | Appropriate output design and display as expected. | Poorly designed output. | Minimum to no formatted output with messy display. | | **No** | **Marks** | | --- | --- | | **1** |  | | **2** |  | | **3** |  | | **4** |  | |  |
| **Program Code Quality & Standard** | **1.0** | **5** | All identifiers strictly conform to the standard Java naming convention and all are meaningful | Most of the identifiers conform to the standard Java naming convention and are meaningful | Some of the identifiers conform to the standard Java naming convention and some are meaningful | Limited numbers of the identifiers conform to the standard Java naming convention or are barely meaningful. | All of the identifiers conform to the standard Java naming convention or not meaningful | | **No** | **Marks** | | --- | --- | | **1** |  | | **2** |  | | **3** |  | | **4** |  | |  |
| **Total Marks (25 marks):** | | | | | | | **1** |  | |
| **2** |  | |
| **3** |  | |
| **4** |  | |

| **CLO 3** | Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2) | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section B: Object-Oriented Concept - Group Marks (45 Marks)** | | | | | | | | | |
| **Task** | **Weight** | **Max. Marks** | **Excellent** | **Good** | **Average** | **Poor** | **Very Poor** | **Scores**  **(Point x Weight)** | **Remarks** |
| **5** | **4** | **3** | **2** | **1** |
| **Class Implementation** | **1.0** | **5m** | Identified the proper classes, methods and attributes to solve particular problems. | Identified appropriate classes, methods and attributes to solve particular problems with minor errors. | Able to identify only some classes, methods and attributes to solve particular problems. | Inappropriate classes, methods and attributes are used. | Unable to show understanding on the usage of classes, methods and attributes. |  |  |
| **Object Implementation** | **1.0** | **5m** | Able to create a structure of objects collaborating among themselves to carry out tasks properly. | Able to create a structure of object collaborating among themselves to carried out task properly with little minor mistakes | Structure of collaborating object created with some mistakes | Structure of collaborating objects created is barely correct. | Wrong structure of object collaboration like one object doing everything itself. |  |  |
| **Abstraction** | **1.0** | **5m** | Define the class at the proper level of abstraction (abstract classes and methods). | Define the class at the proper level of abstraction with minor mistake | Classes defined at the proper level of abstractions with some mistakes | < 50% of correct abstraction is used | Lack of abstraction (abstract classes and methods). |  |  |
| **Encapsulation** | **2.0** | **10m** | Completely correct implementation of encapsulation (private modifier, setter and getter methods). | Correct implementation of encapsulation with only some minor mistakes | Acceptable amount of correct implementation of encapsulation is applied in program | < 50% of correct implementation for encapsulation applied in program | Completely incorrect implementation of encapsulation (private modifier, setter and getter methods) |  |  |
| **Inheritance** | **2.0** | **10m** | Completely correct use of inheritance (correct use of extends keyword, super & sub class) | Correct implementation of inheritance with only some minor mistakes | Acceptable amount of correct implementation of inheritance is applied in program | < 50% of correct implementation for inheritance applied in program | Completely incorrect use and implementation of inheritance |  |  |
| **Polymorphism** | **2.0** | **10m** | Completely correct use and implementation of polymorphism. Methods toString() & equals() correctly overridden. | Correct implementation of polymorphism with only some minor mistakes | Acceptable amount of correct implementation of polymorphism is applied in program | < 50% of correct implementation for polymorphism applied in program | Completely incorrect use and implementation of polymorphism. |  |  |
| **Total Marks (45 marks):** | | | | | | | |  | |

| **CLO 3** | Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2) | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section C: Object-Oriented Concept - Group Marks (30 Marks)** | | | | | | | | | |
| **Task** | **Weight** | **Max. Marks** | **Excellent** | **Good** | **Average** | **Poor** | **Very Poor** | **Scores**  **(Point x Weight)** | **Remarks** |
| **5** | **4** | **3** | **2** | **1** |
| **Cohesion** | **1.0** | **5m** | The code demonstrates exceptional cohesion with all modules, classes, or functions having a clear and single responsibility. | The code demonstrates good cohesion overall where most modules, classes, or functions have a clear and single responsibility | Some modules, classes, or functions have a clear and single responsibility, but others may be slightly ambiguous | Few modules, classes, or functions have a clear and single responsibility. | Modules, classes represent more than one entity and lack of clear responsibility. Poor class cohesion. |  |  |
| **Coupling** | **1.0** | **5m** | Minimal or no direct dependency between modules, classes, or functions. Coupling is effectively managed, resulting in highly maintainable and modular code | Most modules, classes, or functions have minimal direct dependencies. | Some modules, classes, or functions have moderate direct dependencies. | Many modules, classes, or functions have significant direct dependencies. | Modules, classes, or functions have excessive and tightly coupled dependencies which make module be harder to reuse or test. |  |  |
| **Association, Aggregation and Composition** | **2.0** | **10m** | Associations between classes are accurately identified and properly established. | Associations between classes are generally identified and properly established. | Some associations between classes may be missing or improperly established. | Several associations between classes are missing or improperly established. | Completely incorrect use and implementation of association, aggregation and composition relationship. |  |  |
| **UML Class Diagram** | **2.0** | **10m** | The UML class diagram demonstrates excellent understanding and implementation. All classes, attributes, and methods are accurately represented with appropriate visibility, data types, and associations. | Most classes, attributes, and methods are accurately represented with appropriate visibility, data types, and associations. | Some classes, attributes, and methods may be missing or inaccurately represented, with visibility, data types, or associations not fully defined. | Many classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations not adequately defined. | Classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations lacking clarity or definition. |  |  |
| **Total Marks (30 marks):** | | | | | | | |  | |

**Table of Contents**

[1.0 System Idea (Hospital Management System) 17](#_xqzld7d06o47)

[Introduction 17](#_m3v4od7925rn)

[1.1 Manage Rooms 17](#_nbnefxk0moy2)

[1.2 Manage Patients 17](#_sfpb88x87lxx)

[1.3 Manage Doctors 17](#_qu0xe6az616)

[1.4 Manage Nurses 17](#_el7r5yffl6cj)

[1.5 Manage Appointments 17](#_n5i1qkekxghz)

[1.6 Manage Medical Records 18](#_kiqai8yy71ba)

[1.7 Search/View Data 18](#_os9pva1l60x6)

[1.8 Analytics Dashboard 18](#_herne03g1lhi)

[1.9 Additional Features & Architecture 18](#_y0ck8sbzp9sj)

[Task Allocation 19](#_9xf2t7394n4s)

[2.0 UML Diagram 21](#_wwsduo1yy101)

[3.0 System Workflows 22](#_hi9akockqhw)

[3.1 Main Program 22](#_8n4mb7etv27n)

[3.1.1 Admin Login 22](#_akomjug9nqf)

[3.1.2 Main Menu 24](#_slgeesirr5a2)

[3.2 Room Management 25](#_hii548gfh7y3)

[3.2.1 Add Room 26](#_2kqkuvw7rswg)

[3.2.1.1 Add Room (After chosen an option) 27](#_o6cxq2jh4asc)

[3.2.2 Remove Room 28](#_7e27u7bjeo2d)

[3.2.2.1 Remove Room (After chosen an room to remove) 29](#_oxozgmlolhwp)

[3.2.3 Update Room Status 30](#_rl8a3h6330dm)

[3.2.3.1 Update Room Status (After update a room status) 31](#_wm1dsgalemgr)

[3.2.4 Back to Main Menu 32](#_duzwgf9w172m)

[3.3 Patient Management 33](#_ltxfaokq275v)

[3.3.1 Add Patient 33](#_u7kxnv88i1o4)

[3.3.1.1 Add Patient (After adding a patient) 34](#_r3bv8d8qyxo5)

[3.3.2 Remove Patient 35](#_shediys7cfs6)

[3.3.2.1 Remove Patient (After remove a patient) 36](#_v2kbvl5e7jl)

[3.3.3 Update Patient Details 37](#_j50pbseisysy)

[3.3.3.1 Update Patient Details (After updating a patient detail) 38](#_122d909s4z49)

[3.3.4 View Patients 39](#_s7f2qur7hgwo)

[3.3.4.1 View Patients (After chosen a patient to view) 40](#_a0bji9kf1u0s)

[3.3.3.5 Back to Main Menu 41](#_9jxsnobd59ft)

[3.4 Doctor Management 42](#_7ny4by9lcb1w)

[3.4.1 Add Doctor 42](#_se1sj4h41qx8)

[3.4.1.1 Add Doctor (After adding a doctor) 43](#_asibz1apn4sb)

[3.4.2 Remove Doctor 44](#_c5ndthcuw194)

[3.4.2.1 Remove Doctor (After chosen a doctor to remove) 44](#_ttnqmn6rron2)

[3.4.3 Update Doctor Details 45](#_qaq6n5h4oqzd)

[3.4.3.1 Update Doctor Details (After updating a doctor detail) 46](#_6innt2psrhlz)

[3.4.4 View Doctors 47](#_tdxx8v9fj7nf)

[3.5 Nurse Management 50](#_pdsac7kk62l9)

[3.5.1 Add Nurse 51](#_lppz05mpw99f)

[3.5.1.1 Add Nurse (After adding a nurse) 52](#_tacubncproxd)

[3.5.2 Remove Nurse 53](#_o3m17ormbhj3)

[3.5.2.1 Remove Nurse (After chosen a nurse to remove) 53](#_tgw0ddj162fu)

[3.5.3 Update Nurse Details 54](#_z9qayiojywbm)

[3.5.3.1 Update Nurse Details (After updating a nurse detail) 55](#_o2nder7i3mg6)

[3.5.4 View Nurses 56](#_cbgi6gnko58p)

[3.5.4.1 View Nurse (After chosen a nurse to view) 57](#_7xgxz7j3p36f)

[3.5.5 Back to Main Menu 58](#_c74ppowqcxob)

[3.6 Appointment Management 59](#_3eq3z19zjm23)

[3.6.1 Add Appointment 59](#_kyjwgjkpa37j)

[3.6.1.1 Add Appointment (After adding an appointment) 60](#_fzefgt3dn4tp)

[3.6.2 Remove Appointment 62](#_luczst8rx9qr)

[3.6.2.1 Remove Appointment (After chosen an appointment to remove) 63](#_ueqlrmb2kdc2)

[3.6.3 Update Appointment Details 63](#_h5dctglvr2ps)

[3.6.3.1 Update Appointment Details (After updating an appointment detail) 64](#_t1bim48yt3c6)

[3.6.4 View Appointments 64](#_qdw9gk46arbe)

[3.6.5 Back to Main Menu 65](#_y8x671nl75uw)

[3.7 Medical Record Management 66](#_wfa54rlz8406)

[3.7.1 Add Medical Record 66](#_e5gmpdnp1u18)

[3.7.1.1 Add Medical Record (After adding a medical record) 67](#_afwn64jx73np)

[3.7.2 Remove Medical Record 67](#_mn2yloug6od8)

[3.7.2.1 Remove Medical Record (After chosen a medical record to remove) 68](#_nkknx661o19u)

[3.7.3 Update Medical Record Details 69](#_c8bcfe3xxwj6)

[3.7.3.1 Update Medical Record Details (After updating a medical record detail) 70](#_9fvfez2tak77)

[3.7.4 View Medical Record 71](#_189uvn5c6kdm)

[3.7.4.1 View Medical Records (Chosen an appointment to view has records) 72](#_bss9tcqruv4z)

[3.7.4.2 View Medical Records (Chosen an appointment to view has no records) 73](#_tlld3fhvfn5x)

[3.7.5 Back to Main Menu 74](#_oe1g6p3q2xt1)

[3.8 Search and View Management 75](#_od92xua7hopg)

[3.8.1 Search Patients 75](#_qxxmbry5m3he)

[3.8.1.1 Search Patients (Search by name) 76](#_pnogg72pyjs4)

[3.8.1.2 Search Patients (Search by address) 77](#_5wtpdxlhus9)

[3.8.1.3 Search Patients (Search by gender) 78](#_awgvhynxwwyy)

[3.8.1.4 Search Patients (Search by age range) 79](#_bj8hzv8ldg8i)

[3.8.2 Search Doctors 80](#_gp66j2to57ri)

[3.8.2.1 Search Doctors (Search by Name) 81](#_ua0am0udiz6a)

[3.8.2.2 Search Doctors (Search by Department) 82](#_6qhn9bcvyg35)

[3.8.2.3 Search Doctors (Search by Gender) 83](#_864byx7fdgyj)

[3.8.2.4 Search Doctors (Search by Age Range) 84](#_t0tkpdip417u)

[3.8.2.5 Search Doctors (Search by Experience) 85](#_8xwmag3d0yhi)

[3.8.2.6 Search Doctors (Search by Shift) 86](#_ot93qdg1zn2p)

[3.8.3 Search Nurses 87](#_c3jxi1vyl85n)

[Explanation : When the user enters "3" in the main page of the Search and View Data, it will go to the Search Nurses section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria. 87](#_3gyviv7dpznx)

[3.8.3.1 Search Nurses (Search by Name) 87](#_ylhq2iippsjt)

[3.8.3.2 Search Nurses (Search by Department) 88](#_p5ik1xe9tjbw)

[3.8.3.3 Search Nurses (Search by Gender) 89](#_u1u6ysgp5xfz)

[3.8.3.4 Search Nurses (Search by Age Range) 90](#_bb3q66su2cof)

[3.8.3.5 Search Nurses (Search by Experience) 91](#_3h5lqgwgea2d)

[3.8.3.6 Search Nurses (Search by Shift) 92](#_k6ashali5n6x)

[3.8.4 Search Appointments 93](#_6zx83r7hywsg)

[When the user enters "4" in the main page of the Search and View Data, it will go to the Search Appointment section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria. 93](#_ualqgnpzl0di)

[3.8.4.1 Search Appointments (Search by status) 93](#_o96kx1itlrgq)

[3.8.4.2 Search Appointments (Search by date) 94](#_kecghhsiee8c)

[3.8.5 Search Rooms 95](#_voxzslgahtl5)

[When the user enters "5" in the main page of the Search and View Data, it will go to the Search Room section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria. 95](#_rh7kekmlz8b2)

[3.8.5.1 Search Rooms (Search by room type) 96](#_w015tk9re4qr)

[3.8.5.2 Search Rooms (Search by status) 98](#_t87qgt84rim0)

[3.8.6 Back to Main Menu 99](#_4pc6pyeb47kz)

[3.9 Analytic Dashboards 100](#_mvfufwqf0yjx)

[3.10 Change Password 102](#_98xgupq3ofbw)

[3.10.1 Change Password (After changing a new password) 103](#_4xs2xkmtlg2)

[3.11 Exit 104](#_j6ud0u89bm2i)

[4.0 Conclusion 105](#_wigugmy6lio4)

[5.0 References 106](#_9lan0debe3wg)

## 

# **1.0 System Idea (Hospital Management System)**

## **Introduction**

The Hospital Management System (HMS) is a comprehensive, modular Java application engineered to automate and optimize all core hospital operations. Designed for efficiency, flexibility, and robustness, the HMS centralizes patient care, staff management, resource allocation, appointment scheduling, and analytics into a single, scalable platform. The system leverages real-time data, persistent storage, and advanced search and reporting to support high-throughput environments and informed decision-making at every level.

## **1.1 Manage Rooms**

The Room Management module allows administrators to add, remove, and update the status of hospital rooms, supporting all major types: ICU, Ward, Emergency, Operation, and Isolation. Each room is tracked for type and occupancy status. The system provides real-time visibility into room availability and occupancy, with the ability to display all rooms, filter by type, and remove rooms by ID. Room utilization statistics—including occupancy rates for each room type—are calculated and visualized in the analytics dashboard, supporting dynamic resource allocation and operational planning.

## **1.2 Manage Patients**

Patient Management is the operational backbone of the HMS. Users can register new patients, update existing records, and discharge patients. Each patient record includes name, address, phone number, gender, and age, with validation at every step. Advanced search and filtering are available by name, address, gender, and age range. PatientController supports viewing all patients, displaying details for a specific patient, and updating or removing records. Demographic statistics (age and gender distributions) are calculated and visualized, supporting both clinical and administrative planning.

## **1.3 Manage Doctors**

The Doctor Management module enables hiring, updating, and removal of doctor profiles. Each doctor record includes name, address, phone number, gender, age, department, shift, years of experience, and specialization. The system validates all input and supports multi-criteria search by name, department, gender, age, experience, and shift. Doctors can be updated or removed by ID, and all doctors can be displayed in a formatted list. Workforce analytics provide real-time counts and distribution by department and experience, supporting optimal scheduling and resource allocation.

## **1.4 Manage Nurses**

Nurse Management mirrors the doctor module, supporting the hiring, updating, and removal of nurse profiles. Each nurse record includes name, address, phone number, gender, age, department, shift, and years of experience. The system validates all input, supports multi-criteria search, and allows updating or removing nurses by ID. Analytics on nurse distribution by department, shift, and experience are available for efficient coverage and planning.

## **1.5 Manage Appointments**

The Appointment Management module automates the scheduling, updating, and cancellation of appointments. An interactive workflow guides users through selecting patient, doctor, room, date, and time, with validation and conflict detection at each step. Appointments can be viewed, searched, updated (status: Scheduled/Active), or cancelled by ID. The system supports bulk viewing by status or date range (today, week, month). Appointment trends—including weekly distribution, peak days, and most common times—are calculated and visualized in the analytics dashboard. Each patient’s appointment history can also be displayed.

## **1.6 Manage Medical Records**

Medical Record Management ensures secure creation, updating, and removal of patient-linked records. Each record can store diagnoses, prescriptions, and treatments, with audit trails for all changes. Users can add, update, or remove records, and view the full medical history for any patient. The system enforces data completeness and supports longitudinal patient histories for evidence-based care. Medical records are linked to patient IDs and can be searched and displayed as needed.

## **1.7 Search/View Data**

A unified search interface enables rapid querying and cross-referencing of patients, doctors, nurses, appointments, and rooms using multiple criteria. Each entity supports advanced search:

* Patients: by name, age range, gender, address
* Doctors/Nurses: by name, department, gender, age, experience, shift
* Appointments: status, date
* Rooms: by type, status

This module ensures instant access to operational data for decision support and high-throughput environments.

## **1.8 Analytics Dashboard**

The analytics dashboard aggregates and visualizes live statistics, including:

* Total patients, active appointments, available rooms, staff on duty (doctors and nurses)
* Room utilization rates by type (ICU, Ward, Emergency, Operation, Isolation)
* Patient demographics (age and gender distributions)
* Appointment analytics (day-of-week trends, peak hours, most common appointment times)

**Visualizations** include demographic bars and appointment trend graphs. Peak analysis identifies the busiest days and times, supporting proactive resource management and strategic planning.

## 

## **1.9 Additional Features & Architecture**

* **Authentication:** Secure admin login and hidden password input entry for input hiding purposes.
* **Persistence:** All data is saved and loaded from CSV files for repeated access and modifications.
* **Validation:** Comprehensive input validation and error handling throughout.
* **Utilities:** Helper classes for string formatting, color codes, date/time utilities, and menu constants.
* **Extensibility:** Modular design allows for rapid adaptation to new requirements and easy integration of additional features.

# 

# 

# 

# **Task Allocation**

| **Student Name** | **Modules Handled** | **Description** |
| --- | --- | --- |
| **Tan Wy Hang** | 1. **Room model**: definition of room attributes, room status, and room type; management of room data. 2. **Person abstract class:** base class for all person-related entities (patient, doctor, nurse, admin), encapsulating shared attributes and behaviors. 3. **Controller interface**: standardises CRUD method signatures for all controllers. 4. **RoomController**   implementations: concrete classes for managing room and including all CRUD operations related.   1. **Patient model**: definition of patient attributes and integration with appointments and medical records. 2. **PatientController**: manages patient registration, data updates, removal, and listing; links patients to appointments and medical records. 3. **Search/view feature**: unified search and filtering for patients, doctors, nurses, rooms, and appointments; supports multi-criteria queries with the use of array streams and formatted data display. 4. **Analytics dashboard**: data aggregation and visualization of operational statistics such as room occupancy, patient demographics, peak operating periods and appointment trends. | Handled the room model, person abstract class, controller interface, and respective controller classes. Developed the search/view feature and the analytics dashboard. Responsibilities included implementing **CRUD** operations for rooms (e.g., adding a new room, updating room status, deleting a room, and viewing all rooms), designing the abstract person class as a base for patient, doctor, nurse, and admin models, and enforcing a consistent controller structure for CRUD across modules. The search/view feature allows searching for patients by name or filtering rooms by type, while the analytics dashboard displays statistics such as room occupancy rates and patient demographics. Implements **ENUM** as argumentative inputs for readability, scalability and error prevention. |
| **Tan Ying Nan** | 1. **Appointment model**: structure for appointment data, including patient, doctor, room, date, time, and status. 2. **Medical record model**: structure for storing diagnoses, treatments, and prescriptions linked to patients. 3. **AppointmentController:** manages scheduling, updating, canceling, and viewing appointments. 4. **MedicalRecordController**: handles creation, updating, deletion, and retrieval of medical records. 5. **StringUtils model:** Defines the utility functions for repetitive string operations like the string beautifier 6. **DateTimeUtils model**: to simplify handling of date by retrieving valid dates automatically, formatting date into a consistent format, and combines date and time into a datetime object. | Worked on the appointment model, medical record, patient model, and their respective controller classes. Implemented CRUD operations for appointments (e.g., scheduling a new appointment, updating appointment status, canceling an appointment, and listing all appointments), medical records (e.g., adding a new record, editing diagnosis or treatment, deleting a record, and viewing a patient’s medical history), and patients (e.g., registering a patient, editing patient data, removing a patient, and displaying all patients). Ensured seamless integration between patient data, appointments, and medical records for comprehensive patient management. |
| **Chia Zhen Yang** | 1. **Doctor model**: definition of doctor attributes, including department, specialization, shift, and experience. 2. **Nurse model**: structure for nurse data, including department, shift, and experience. 3. **Admin model**: structure for admin user data and authentication. 4. **DoctorController**: manages adding, updating, deleting, and viewing doctor profiles. 5. **NurseController**: handles CRUD operations for nurse records. 6. **AdminController**: manages admin account creation, updates, deletion, and listing; supports authentication and role management for system access. | Responsible for the doctor, admin, and nurse models, each with their respective controller classes. Implemented CRUD operations for doctors (e.g., adding a doctor, updating doctor profile, deleting a doctor, and listing all doctors), nurses (e.g., registering a nurse, editing nurse details, removing a nurse, and viewing nurse records), and admins (e.g., creating admin accounts, updating admin information, deleting admin profiles, and displaying all admins). Ensured that all staff-related data can be managed efficiently and securely within the system. |

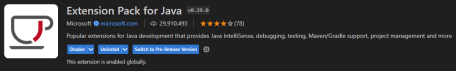
# **2.0 UML Diagram**

# 

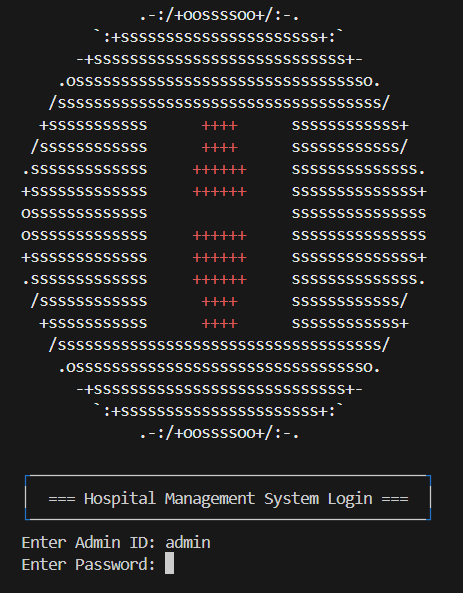
# **3.0 System Workflows**

## 3.1 Main Program

**Note :** Please use **Visual Studio Code (terminal) to run** our system for the best result. Thank you.

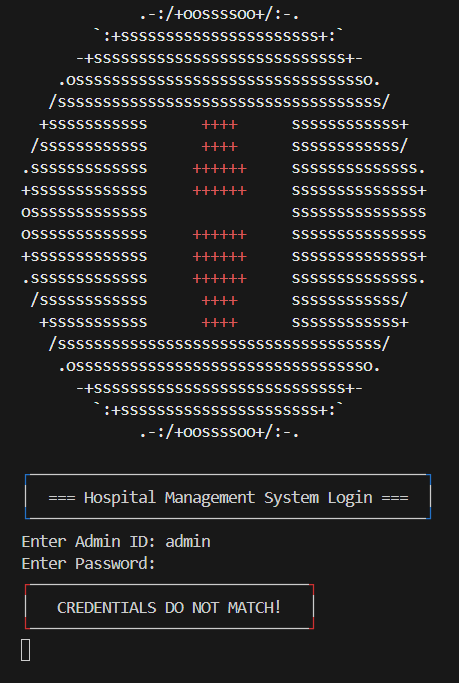


### 3.1.1 Admin Login



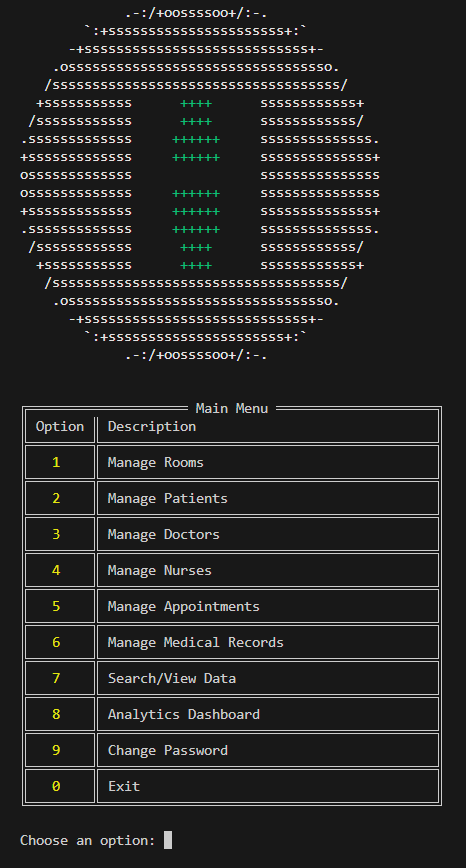
**Explanation :** In this page, admin is required to enter the Admin ID and Password to login into the system. When entering the Admin ID and Password, Admin ID will display as usual but Password will not be shown as the image shown above.

**Note :** The image taken when Admin ID and Password are entered.   
The default Admin ID is **admin** and the default Password is **admin**.



**Explanation :** If the Admin ID or Password doesn't match the preset Admin ID and Password, a message "CREDENTIALS DO NOT MATCH" will be displayed. After pressing Enter again, the system will require admin to enter the credentials again until the correct credentials are matched.

### 3.1.2 Main Menu



**Explanation :** After entering the correct credentials, the system will direct the admin to the **Main Menu**. In this menu, all available options will be clearly displayed, each accompanied by a concise description. The admin is required to select a valid option (ranging from "0" to "9") to proceed with the desired function of the system.

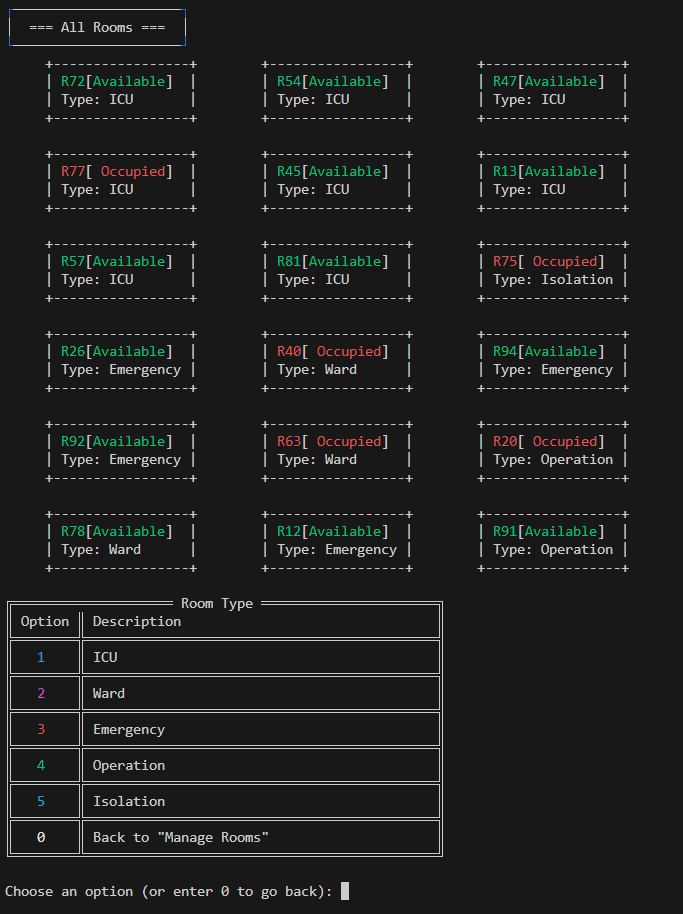
**Note :** The "lock" in the logo will change its colour from red to green after the correct credentials have been entered.

## 3.2 Room Management

****

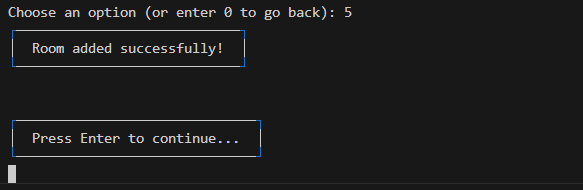
**Explanation :** After entering "1" in the **Main Menu**, the system will lead the admin to the **Room Management** section. In the main page of the **Room Management**, all the rooms will be displayed along with the room status and room type. A menu will also be displayed which shows all the available functions for the room management.

### 3.2.1 Add Room

****

**Explanation :** After entering "1" on the main page of the **Room Management** section, the admin will be directed to the **Add Room** interface. The section displaying all existing rooms will remain visible, just as it appears on the main page. A menu listing all available room types for the new room will also be displayed, allowing the admin to select the appropriate type during the room creation process.

### 3.2.1.1 Add Room (After chosen an option)

****

**Note :** Image taken when chosen option 5 (Isolation).

****

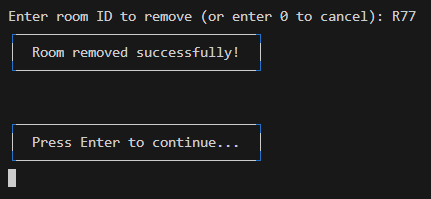
**Explanation :** After the type of room wanted, “Room added successfully!” will be displayed and ask the user to press Enter to continue ( back to the main page of the **Room Management** ). Then there will be an Isolation room added (R22) as the image shown above.

### 3.2.2 Remove Room

****

**Explanation :** After entering "2" on the main page of the **Room Management** section, the admin will be directed to the **Remove Room** interface. The section displaying all existing rooms will remain visible, just as it appears on the main page. Admin is required to enter the room ID that he/she wants to remove.

### 3.2.2.1 Remove Room (After chosen an room to remove)

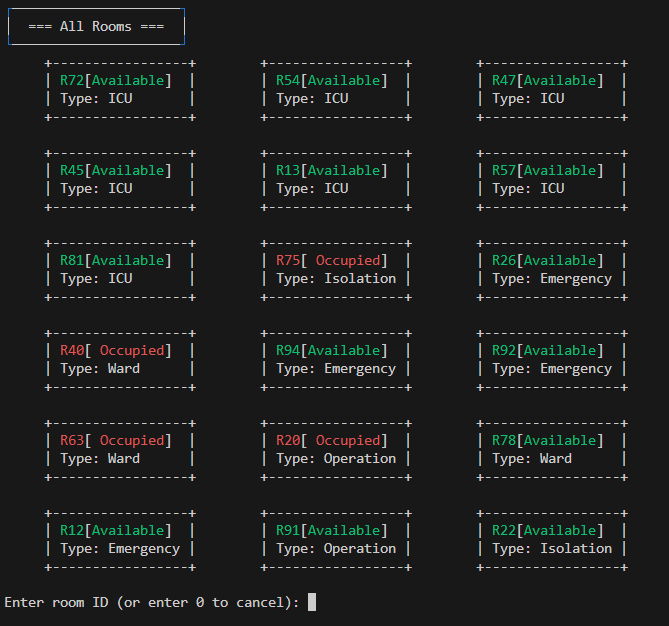
****

**Note :** Image taken when chosen room R77 to remove.

****

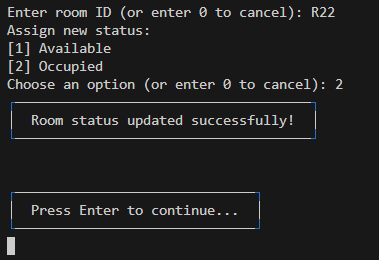
**Explanation :** After entering the room ID, "Room removed successfully!" will be displayed and ask the user to press Enterto continue ( back to the main page of the **Room Management** ). Then the room with the room ID entered (R77) will be removed as in the image shown above.

### 3.2.3 Update Room Status

****

**Explanation :** After entering "3" in the main page of **Room Management**, the user will be directed to the **Update Room Status** section. User is asked to enter the room ID for the room needed to update.

### 3.2.3.1 Update Room Status (After update a room status)

****

**Note :** Image taken when chosen option 2 (Occupied) to update the room.

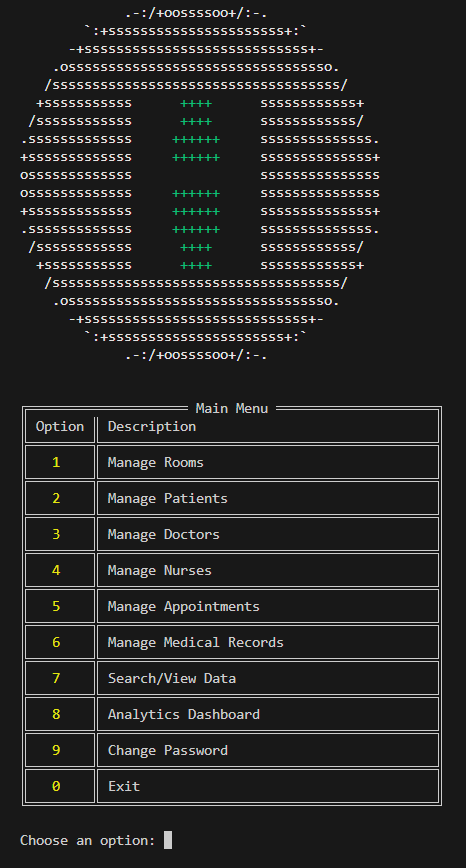


**Explanation :** After entering the room ID, the user is required to assign a new status for the specified room by entering the option. A message, "Room status updated successfully!" will be displayed after all the details have been entered and ask the user to press Enter to continue ( back to the main page of the **Room Management** ). Once confirmed, the status of the specific room (R22) will be changed as in the image shown above.

### 3.2.4 Back to Main Menu

****

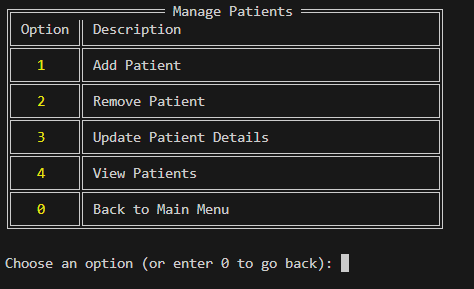
**Note :** Full page image can refer back the image in 3.2

****

**Note :** Full page image can refer back the image in 3.1.2

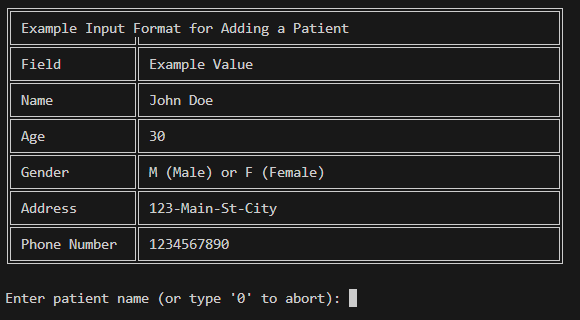
**Explanation :** When entering "0" in the main page of the **Room Management**, the user will go back to the **Main Menu**, where the user can continue with other sections of the system as in the image shown above.

## 3.3 Patient Management

****

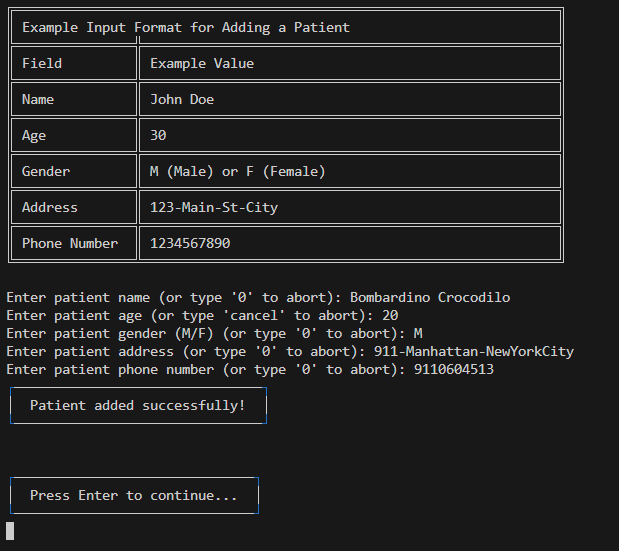
**Explanation :** After entering "2" in the **Main Menu**, the user will enter into the **Patient Management** section. A menu with all available functions of the section will be displayed and ask the user to enter an option.

### 3.3.1 Add Patient

****

**Explanation :** When the user enters "1" in the main page of the **Patient Management**, it will lead the user to the **Add Patient** section. A sample input will be displayed for better understanding and the system will ask the user to enter the details needed to add a new patient as the details show in the sample input.

### 3.3.1.1 Add Patient (After adding a patient)

****

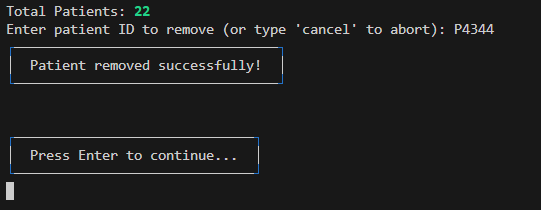
**Explanation :** After all details needed for adding a new patient have been entered,"Patient added successfully" will be displayed and ask the user to press Enterto continue ( back to the main page of the **Patient Management** ).

### 3.3.2 Remove Patient

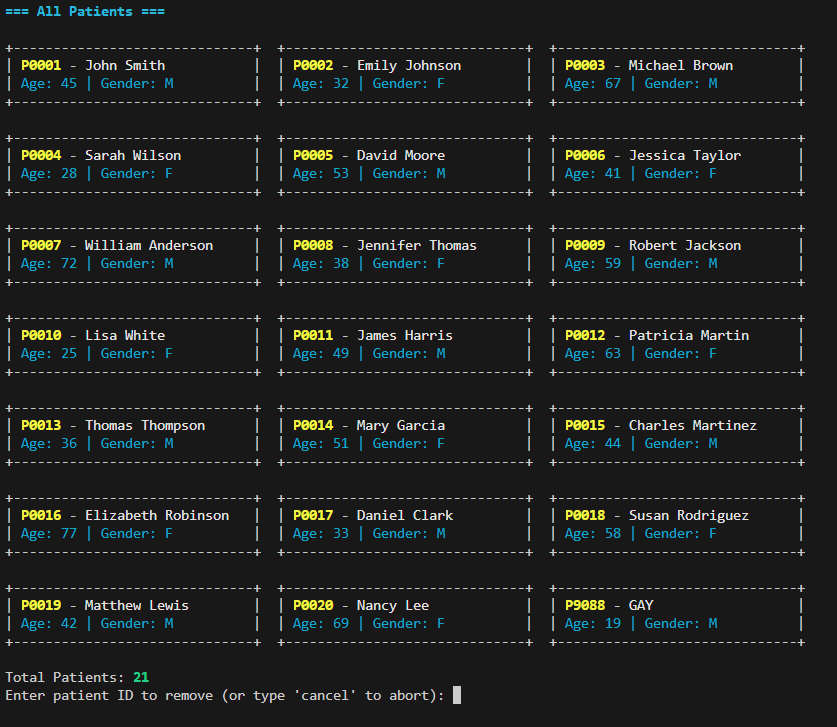
****

**Explanation :** When the user enters "2", it will go to the **Remove Patient** section. All the patients will be displayed along with their ages and genders in the grid format. The number of total patients will also be shown. The user is required to enter the patient ID needed to remove.

### 3.3.2.1 Remove Patient (After remove a patient)

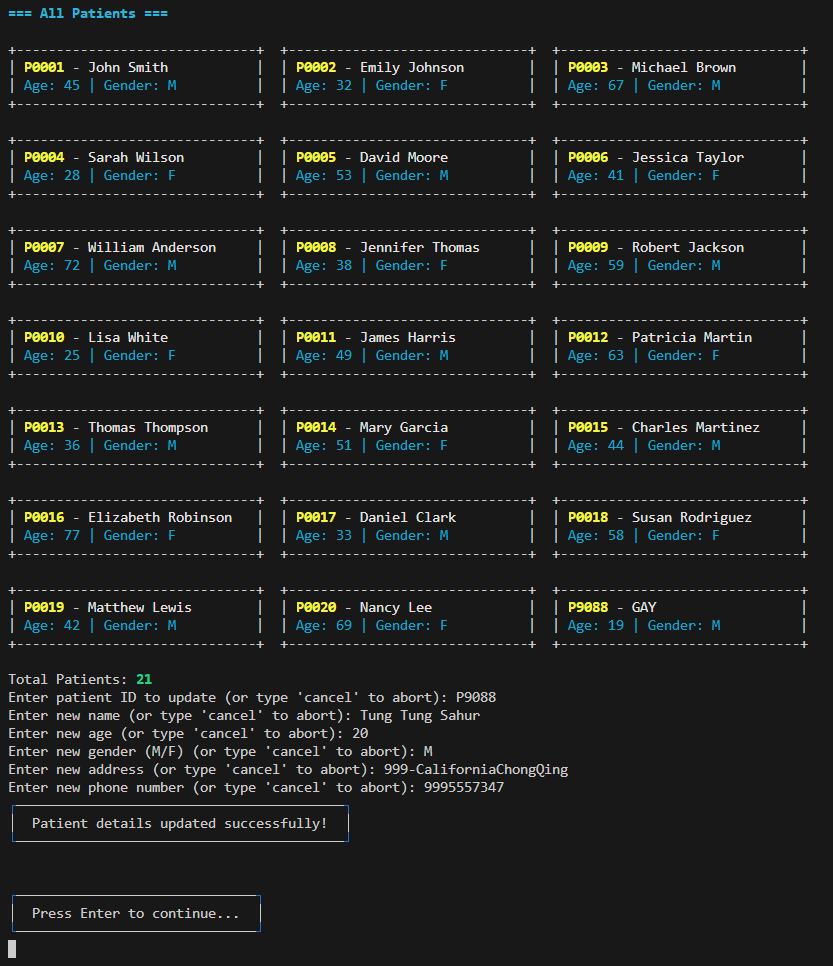
****

**Note :** Full page image can refer back the image in 3.3.2

****

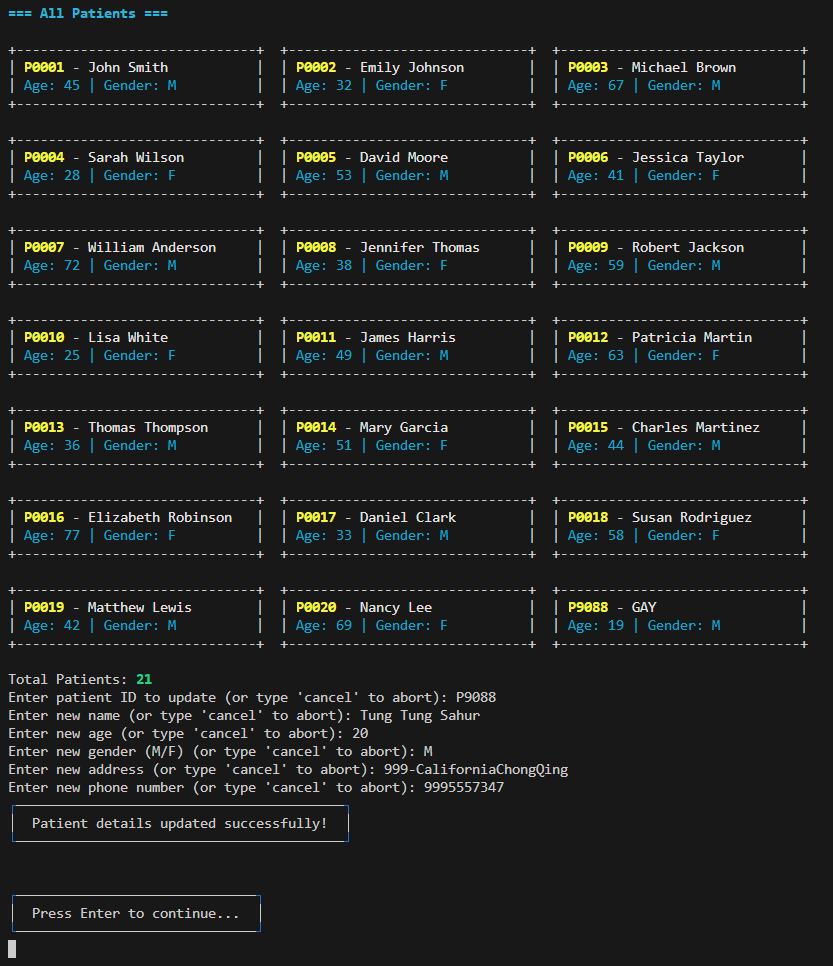
**Explanation :** After entering the patient ID, "Patient removed successfully!" will be displayed and ask the user to press Enterto continue ( back to the main page of the **Patient Management** ).Then the patient with the patient ID entered (P4344) will be removed as in the image shown above.

### 3.3.3 Update Patient Details

****

**Explanation :** When the user enters "3", it will go to the **Update Patient Details** section. All the patients will be displayed along with their ages and genders in the grid format. The number of total patients will also be shown. The user is required to enter the patient ID needed to update.

### 3.3.3.1 Update Patient Details (After updating a patient detail)

****

**Note :** Full page image can refer back to the image in 3.3.3

****

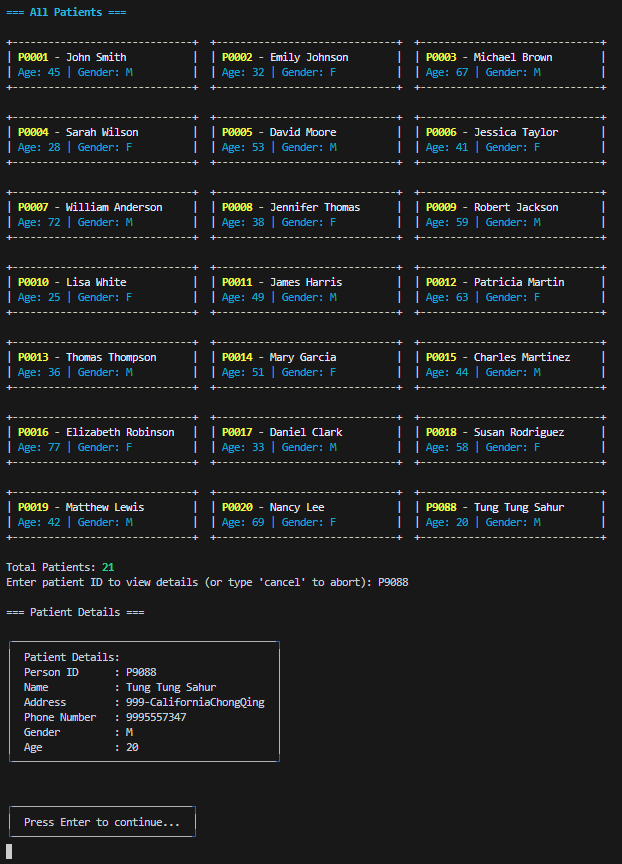
**Explanation :** After entering the patient ID,the user is required to update new details for that patient. A message, "Patient removed successfully!" will be displayed after all details have been entered and ask the user to press Enterto continue ( back to the main page of the **Patient Management** ). Then the patient with the patient ID entered (P9088) will be updated as in the image shown above.

### 3.3.4 View Patients

****

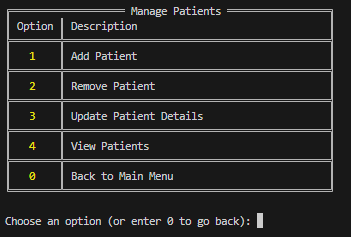
**Explanation :** When the user enters "4", it will go to the **View Patient** section. All the patients will be displayed, then the user is required to enter the patient ID that wanted to view.

### 3.3.4.1 View Patients (After chosen a patient to view)

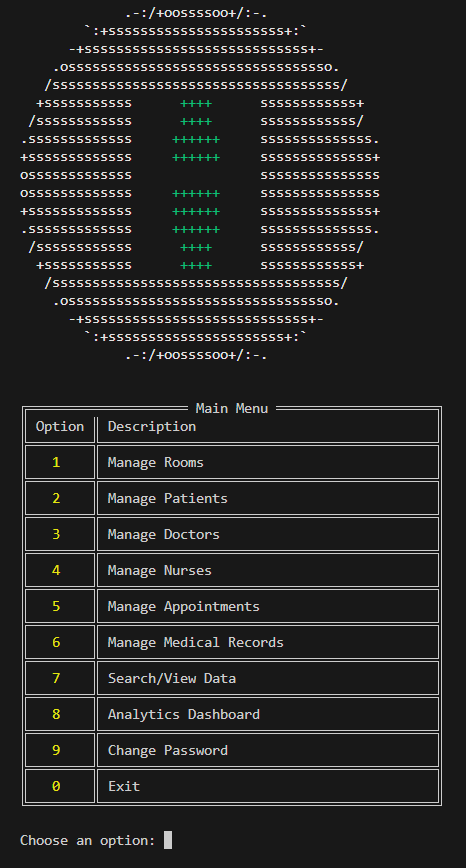
****

**Explanation :** After entering the patient ID, a box of detailed information for the patient with that patient ID will be displayed. Then the user is asked to press Enter to continue ( back to the main page of the **Patient Management** ).

### 3.3.3.5 Back to Main Menu

****

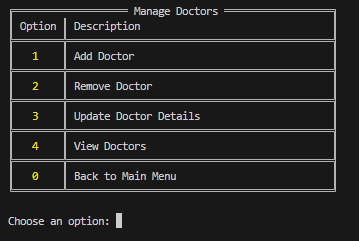
**Note :** Full page image can refer back to the image in 3.3

****

**Note :** Full page image can refer back the image in 3.1.2

**Explanation :** When entering "0" in the main page of the **Patient Management**, the user will go back to the **Main Menu**, where the user can continue with other sections of the system as in the image shown above.

## 3.4 Doctor Management

****

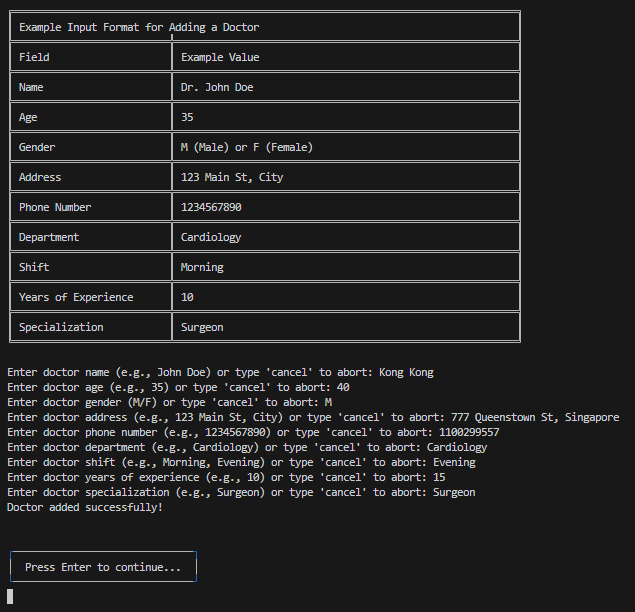
**Explanation :** After entering "3" in the **Main Menu**, the user will go to the **Doctor Management** section. A menu of all available functions for the section will be displayed. User is required to choose an option to perform a specific function of the **Doctor Management**.

### 3.4.1 Add Doctor

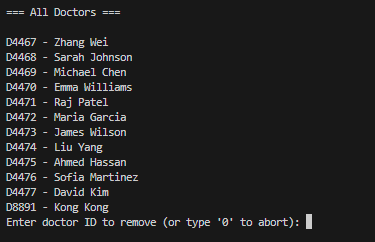
****

**Explanation :** When the user enters "1" in the main page of the **Doctor Management**, it will lead the user to the **Add Doctor** section. A sample input will be displayed for better understanding and the system will ask the user to enter the details needed to add a new doctor, the same as the details shown in the sample input.

### 3.4.1.1 Add Doctor (After adding a doctor)

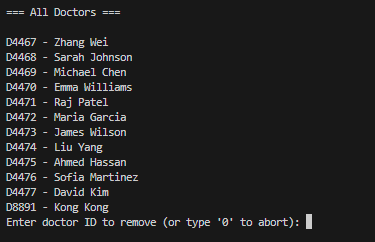
****

**Note :** Full page image can refer back to the image in 3.4.1

****

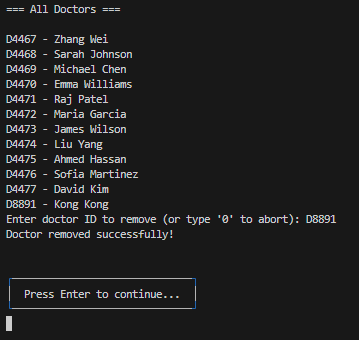
**Explanation :** After all details needed for adding a new doctor have been entered, "Doctor added successfully" will be displayed and ask the user to press **Enter** to continue ( back to the main page of the **Doctor Management** ). A doctor (D8891 - Kong Kong) will be added as in the image shown above.

### 3.4.2 Remove Doctor

****

**Explanation :** When the user enters "2" in the main page of the **Doctor Management**, it will go to the **Remove Doctor** section. All the doctors will be displayed in a list. User is required to enter the doctor ID for the doctor to be removed.

### 3.4.2.1 Remove Doctor (After chosen a doctor to remove)

****

**Explanation :** After entering the doctor ID, "Doctor removed successfully!" will be displayed and ask the user to press **Enter** to continue ( back to the main page of the **Doctor Management** ). Then the doctor with the doctor ID entered (D8891 - Kong Kong) will be removed as in the image shown above.

### 3.4.3 Update Doctor Details

****

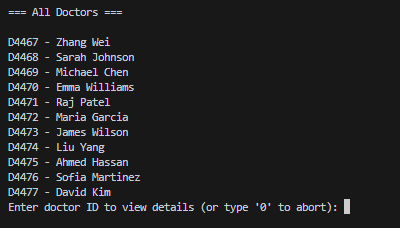
**Explanation :** When the user enters "3" in the main page of the **Doctor Management**, it will lead the user to the **Update Doctor** **Details** section. All the doctors will be displayed in a list and a sample input required for updating doctor details will be displayed for better understanding. Then, the system will ask the user to enter the details needed to add a new doctor, the same as the details shown in the sample input.

### 3.4.3.1 Update Doctor Details (After updating a doctor detail)

****

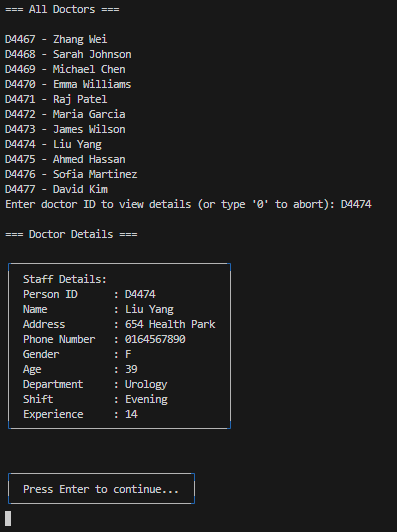
**Explanation :** The doctor details are updated step by step where users can skip the details that do not need to be updated. When the new details are entered, "<detail> updated successfully!" will be displayed while pressing Enter will allow the user to skip the part. After all new details have been updated, the system will ask the user to press Enter to continue ( back to the main page of the **Doctor Management** ).

### 3.4.4 View Doctors

****

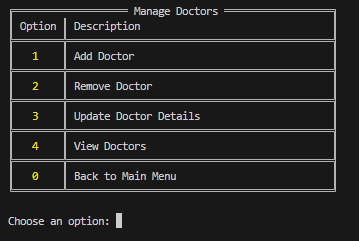
**Explanation :** When the user enters "4" in the main page of the **Doctor Management**, it will go to the **View Doctors** section. All the doctors will be displayed in a list. User is required to enter the doctor ID for the doctor he/she wants to view.

3.4.4.1 View Doctors (After chosen a doctor to view)

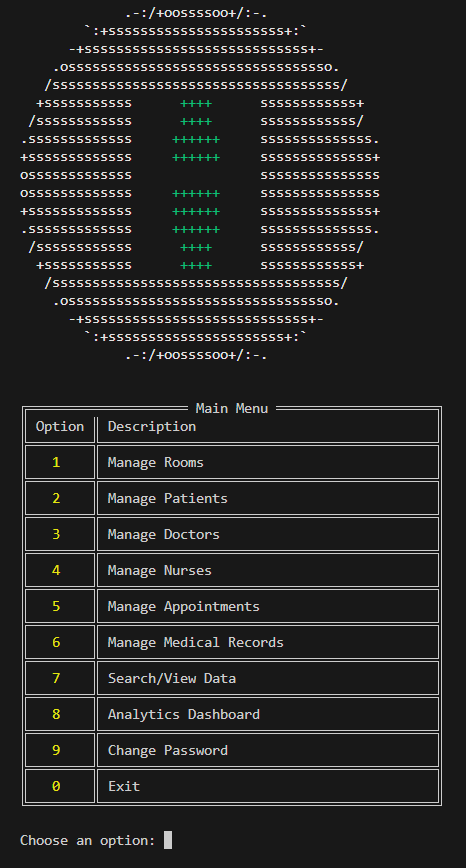
****

**Explanation :** After entering the doctor ID, the doctor details with that ID will be displayed in a box with clean format. The system will ask the user to press Enter to continue ( back to the main page of the **Doctor Management** ).

3.4.5 Back to Main Menu

****

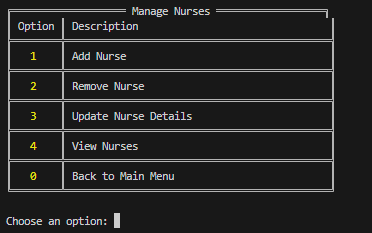
**Note :** Full page image can refer back to the image in 3.4

****

**Note :** Full page image can refer back to the image in 3.1.2

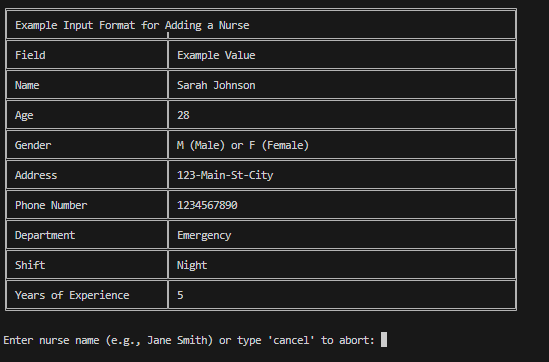
**Explanation :** When entering "0" in the main page of the **Doctor Management**, the user will go back to the **Main Menu**, where the user can continue with other sections of the system as in the image shown above.

## 3.5 Nurse Management

****

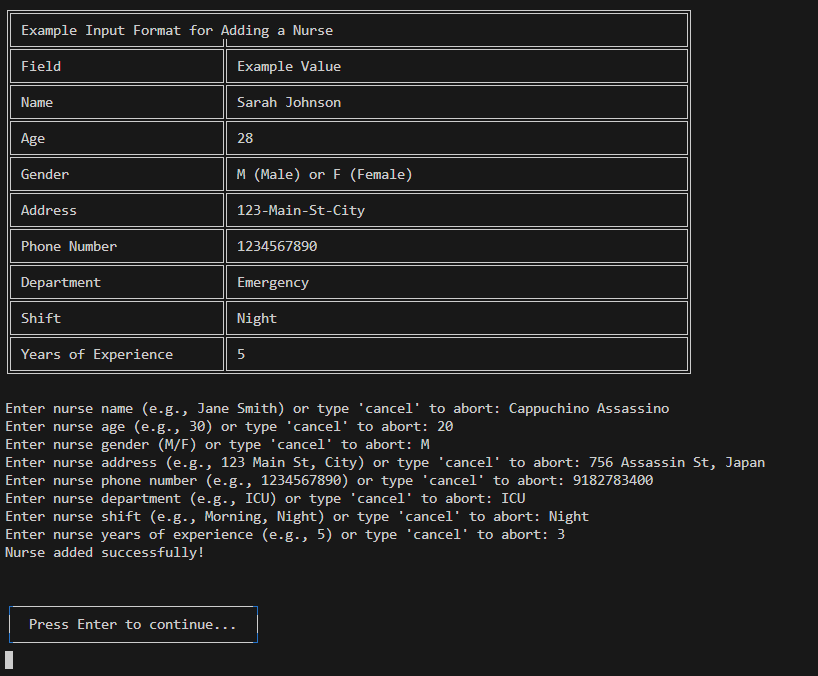
**Explanation :** After entering "4" in the **Main Menu**, the user will go to the **Nurse Management** section. A menu of all available functions for the section will be displayed. User is required to choose an option to perform a specific function of the **Nurse Management**.

### 3.5.1 Add Nurse

****

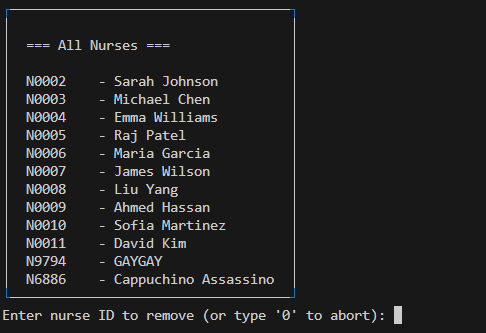
**Explanation :** When the user enters "1" in the main page of the **Nurse Management**, it will lead the user to the **Add Nurse** section. A sample input will be displayed for better understanding and the system will ask the user to enter the details needed to add a new nurse, the same as the details shown in the sample input.

### 3.5.1.1 Add Nurse (After adding a nurse)

****

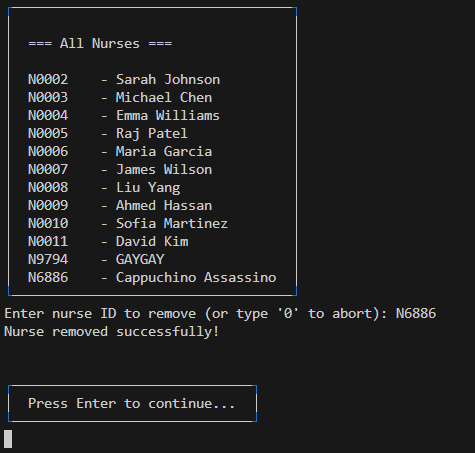
**Explanation :** After all details needed for adding a new nurse have been entered, "Nurse added successfully" will be displayed and ask the user to press Enterto continue ( back to the main page of the **Nurse Management** ).

### 3.5.2 Remove Nurse

****

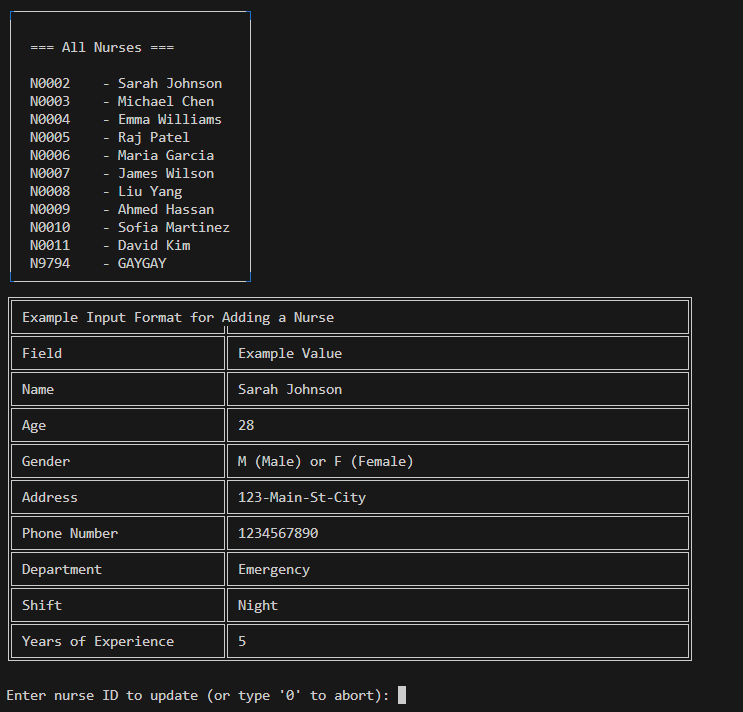
**Explanation :** When the user enters "2" in the main page of the **Nurse Management**, it will go to the **Remove Nurse** section. All the nurses will be displayed in a list. User is required to enter the nurse ID for the nurse who needs to be removed.

### 3.5.2.1 Remove Nurse (After chosen a nurse to remove)

****

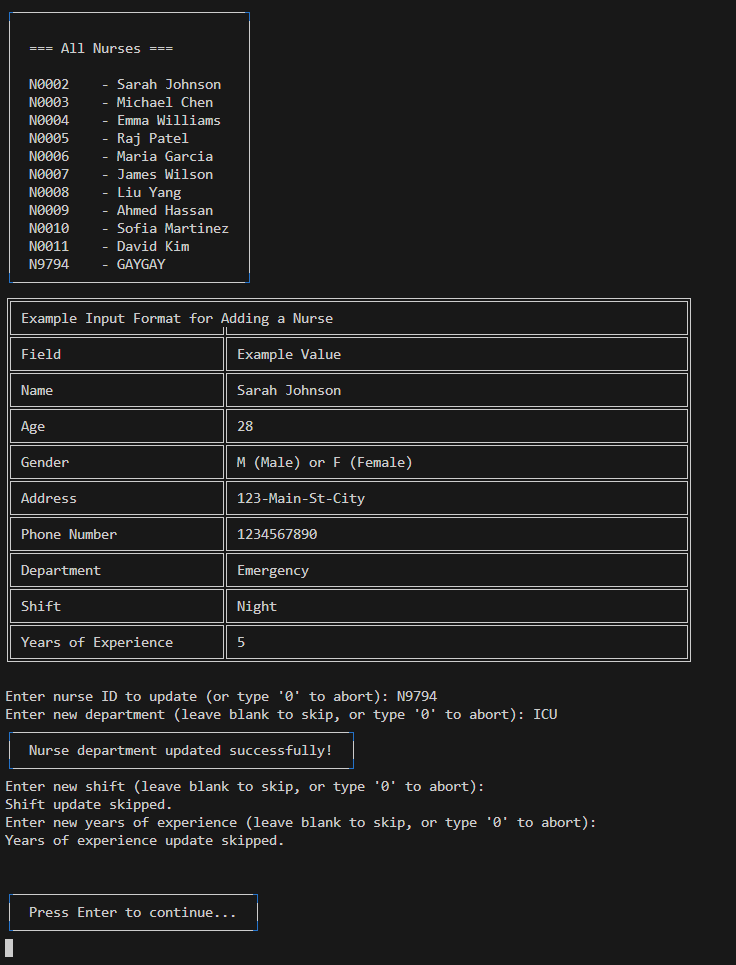
**Explanation :** After entering the nurse ID to be removed, "Nurse removed successfully!" will be displayed and ask the user to press **Enter** to continue ( back to the main page of the **Nurse Management** ).

### 3.5.3 Update Nurse Details

****

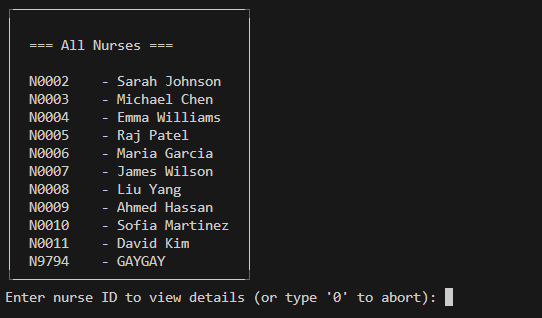
**Explanation :** When the user enters "3" in the main page of the **Nurse Management**, it will lead the user to the **Update Nurse** **Details** section. All the nurses will be displayed in a list and a sample input required for updating nurse details will be displayed for better understanding. Then, the system will ask the user to enter the details needed to add a new nurse, the same as the details shown in the sample input.

### 3.5.3.1 Update Nurse Details (After updating a nurse detail)

****

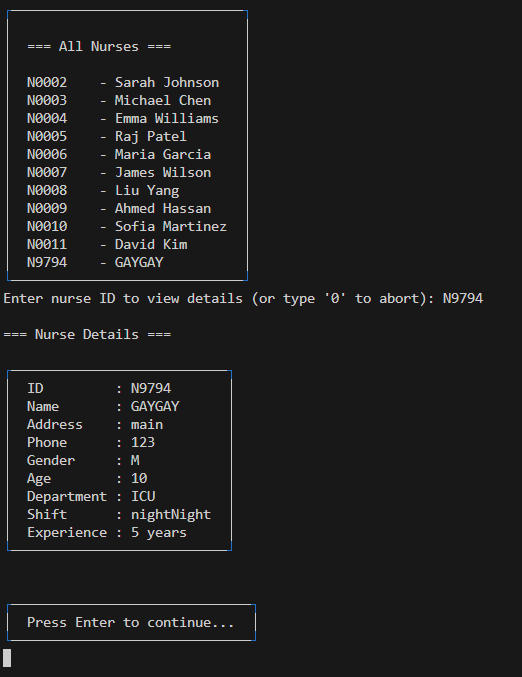
**Explanation :** The nurse details are updated step by step where users can skip the details that do not need to be updated. When the new details are entered, "<detail> updated successfully!" will be displayed while pressing Enter will allow the user to skip the part. After all new details have been updated, the system will ask the user to press Enter to continue ( back to the main page of the **Nurse Management** ).

### 3.5.4 View Nurses

****

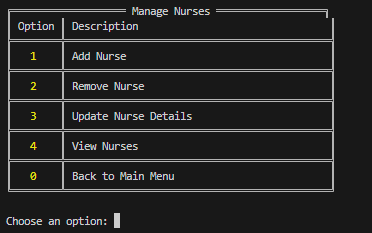
**Explanation :** Entering "4" in the main page of the **Nurse Management** will go to the **View Nurse** section. User is required to enter the nurse ID to view the details.

### 3.5.4.1 View Nurse (After chosen a nurse to view)

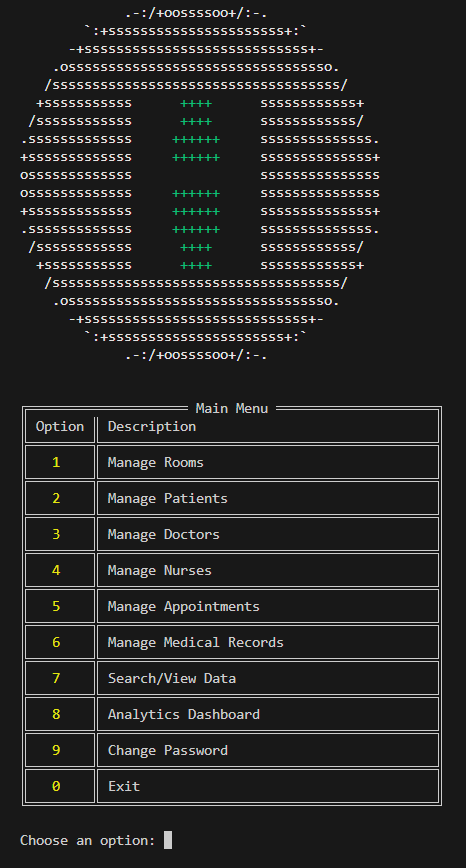
****

**Explanation :** After entering the nurse ID, the nurse details with that ID will be shown in a box. Once finished viewing, the user is asked to press Enter to continue ( back to the main page of the **Nurse Management** ).

### 3.5.5 Back to Main Menu

****

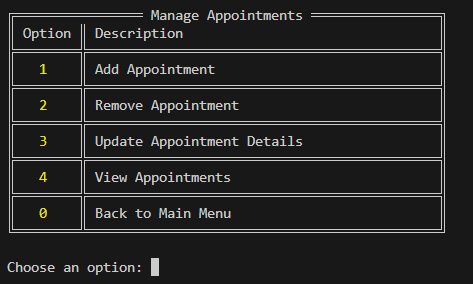
**Note :** Full page image can refer back to the image in 3.5

****

**Note :** Full page image can refer back to the image in 3.1.2

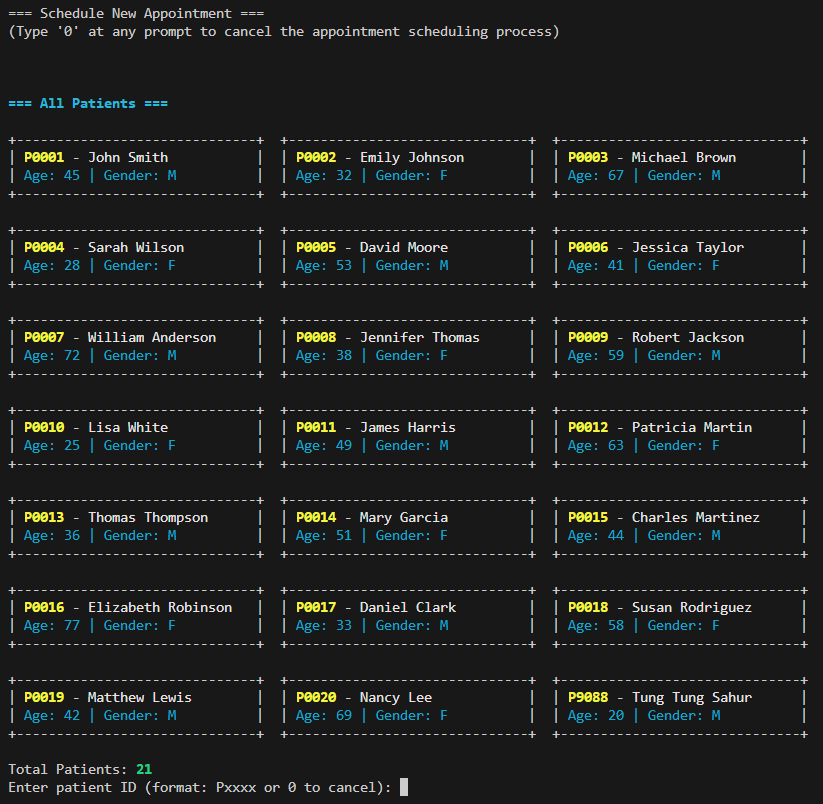
**Explanation :** Entering "0" will lead the user back to the **Main Menu**.

## 3.6 Appointment Management

****

**Explanation :** After entering "5" in the **Main Menu**, the user will go to the **Appointment Management** section. A menu of all available functions for the section will be displayed. User is required to choose an option to perform a specific function of the **Appointment Management**.

### 3.6.1 Add Appointment

****

**Explanation :** After entering "1" in the main page of the **Appointment Management**, the user will go to the **Add Appointment** section. The user is required to enter his / her patient ID and then choose the doctor he / she wants to make an appointment with.

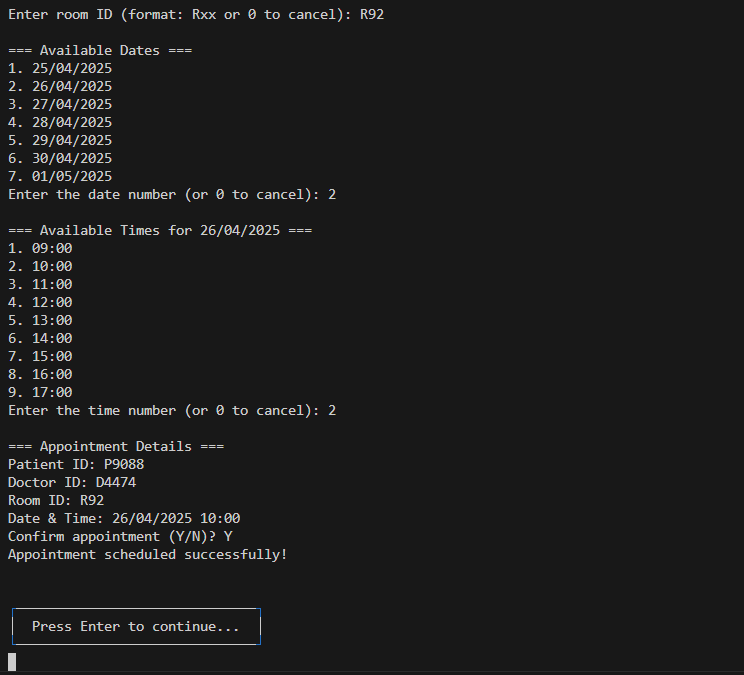
### 3.6.1.1 Add Appointment (After adding an appointment)

****

**Explanation :** After the user enters the patient ID, the user will be guided to the image below to choose the doctor who is in-charge for this appointment.

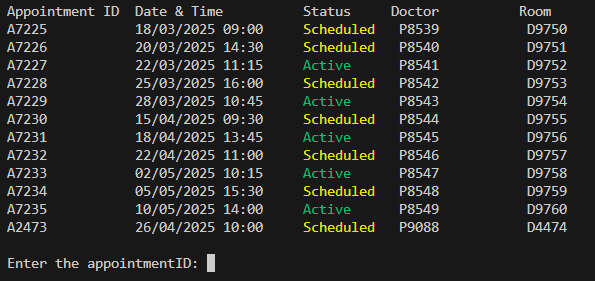
****

**Explanation :** After the user enters the doctor ID, the user will be guided to choose the room where the patient stays. Then the user will be guided to the image below to book the date and time of the appointment this time.

****

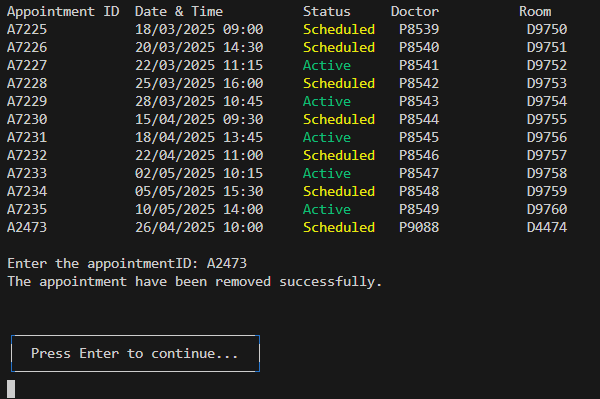
**Explanation :** After finishing entering the patient ID and choosing the doctor, the user is required to choose an available date and time for the appointment which will be displayed as in the image above. After finishing all the steps, the appointment details will be shown with the message "Appointment scheduled successfully!". User is required to press Enter to continue ( back to the main page of the **Appointment Management** ).

### 3.6.2 Remove Appointment

****

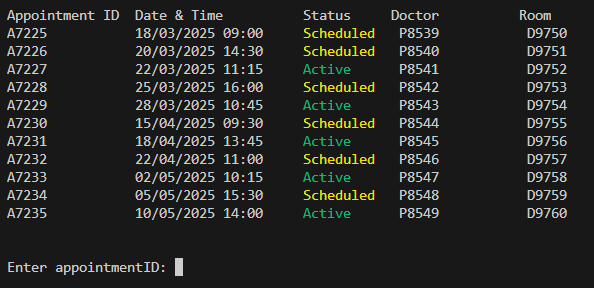
**Explanation :** After entering "2" in the main page of **Appointment Management**, it will go to the **Remove Appointment** section. All the scheduled appointments will be displayed in a table format with all of the details. User is required to enter the appointment ID to be removed.

### 3.6.2.1 Remove Appointment (After chosen an appointment to remove)

****

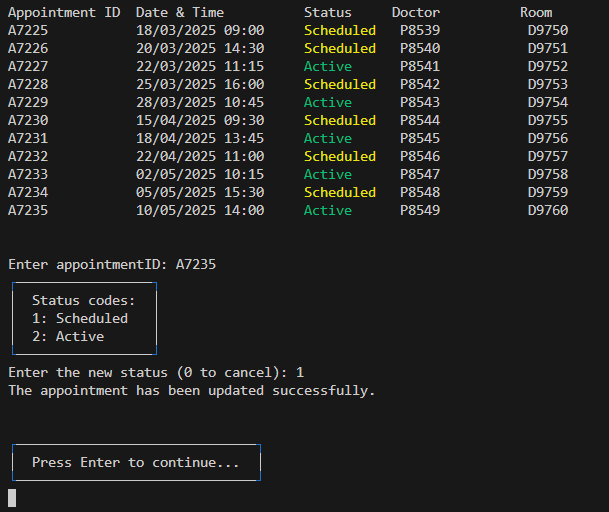
**Explanation :** After entering the appointment ID wanted to be removed, a message "The appointment has been removed successfully." will be displayed and ask the user to press Enter to continue ( back to the main page of the **Appointment Management** ). The appointment with that appointment ID will be removed.

### 3.6.3 Update Appointment Details

****

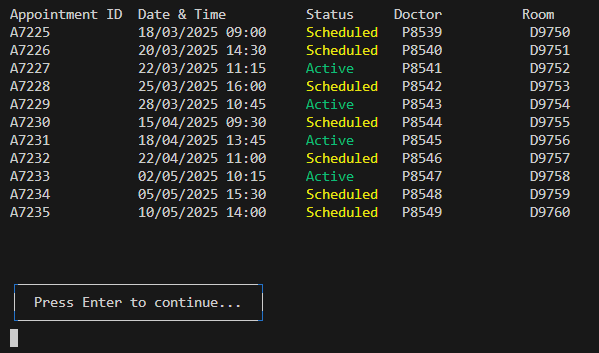
**Explanation :** While entering "3" in the main page of **Appointment Management**, the user will go to the **Update Appointment Details** section. All the appointments will be displayed in a table format. User is required to enter the appointment ID that the user wanted to update its details.

### 3.6.3.1 Update Appointment Details (After updating an appointment detail)

****

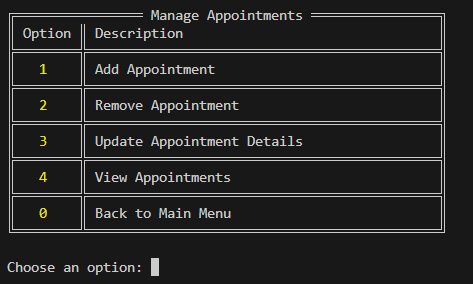
**Explanation :** After entering the appointment ID, the user has to choose the new status for the appointment. Once confirmed, the user is asked to press Enter to continue.

### 3.6.4 View Appointments

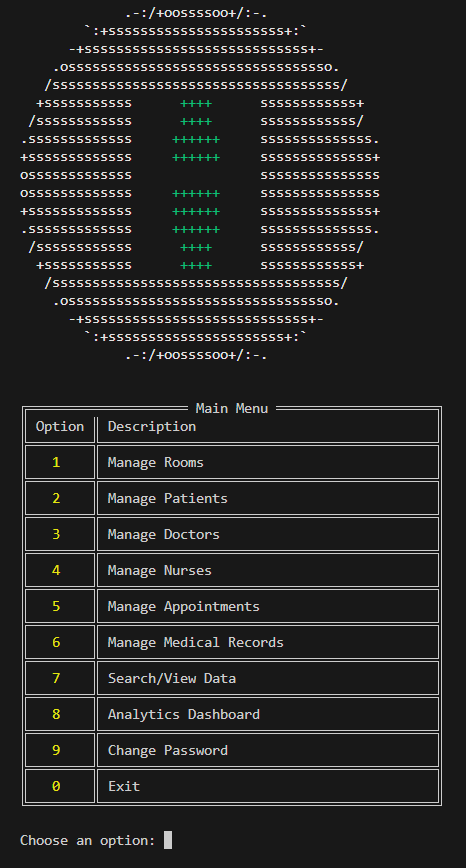
****

**Explanation :** When entering "4" in the main page of **Appointment Management**, it will go to the **View Appointment** section. All the appointments along with each detail will be displayed in a table format.Once finish viewing, the user needs to press Enter to continue ( back to the main page of the **Appointment Management** ).

### 3.6.5 Back to Main Menu

****

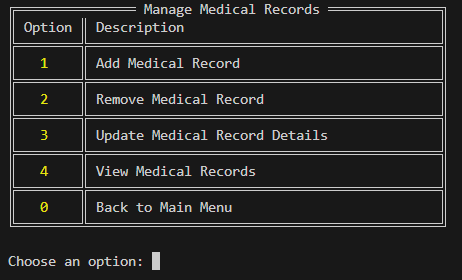
**Note :** Full page image can refer back to the image in 3.6

****

**Note :** Full page image can refer back to the image in 3.1.2

**Explanation :** Entering "0" will lead the user back to the **Main Menu**.

## 3.7 Medical Record Management

****

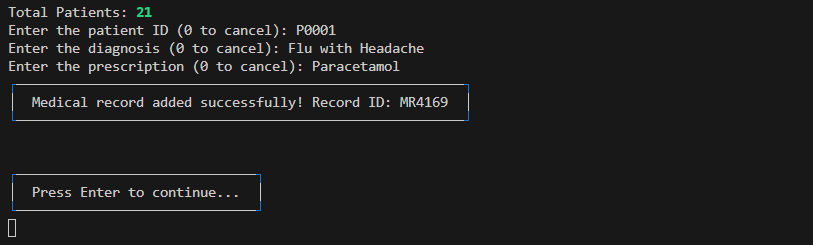
**Explanation :** After entering "6" in the **Main Menu**, the user will go to the **Medical Record Management** section. A menu of all available functions for the section will be displayed. User is required to choose an option to perform a specific function of the **Medical Record Management**.

### 3.7.1 Add Medical Record

****

**Explanation :** Entering "1" in the main page of the **Medical Record Management** will lead the user to the **Add Medical Record** section. All the existing patients will be displayed in a neatly organized grid format along with the number of total patients listed at the bottom. User is prompted to enter the patient ID to create a new medical record.

### 3.7.1.1 Add Medical Record (After adding a medical record)

****

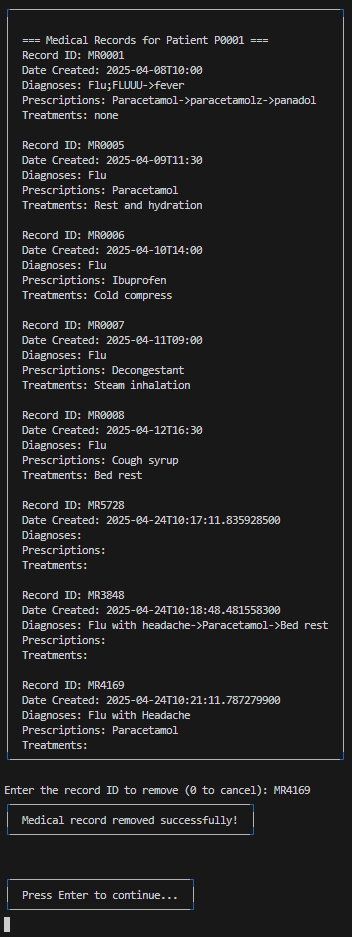
**Explanation :** After choosing the patient, the user needs to enter the medical record details. Once finished, "Medical record added successfully!" will be shown with the record ID. User is prompted to press Enter to continue ( back to the main page of the **Medical Record Management** ).

### 3.7.2 Remove Medical Record

****

**Explanation :** When entering "2" in the main page of the **Medical Record Management**, the user will go to the **Remove Medical Record** section. All the existing patients will be displayed in a neatly organized grid format along with the number of total patients listed at the bottom. User is prompted to enter the patient ID whose medical record needs to be removed.

### 3.7.2.1 Remove Medical Record (After chosen a medical record to remove)

****

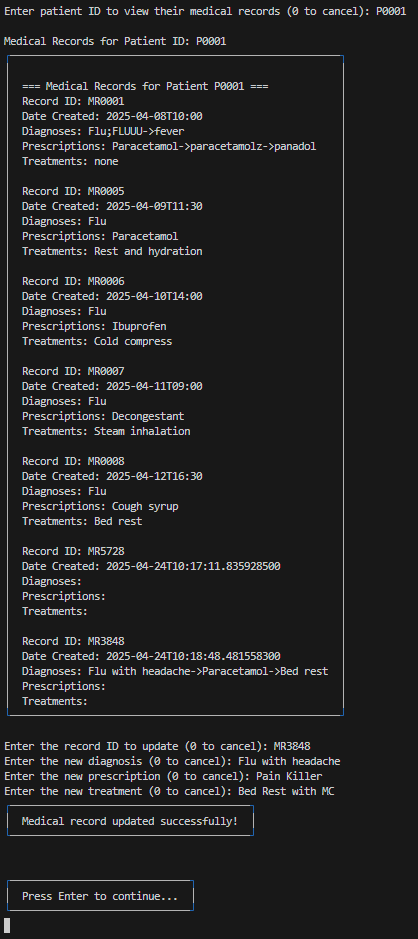
**Explanation :** After entering the patient ID, its medical record will be displayed in a list. User needs to enter the record ID that needs to be removed based on the list. Once confirmed, "Medical record removed successfully!" will be displayed and ask the user to press Enter to continue ( back to the main page of the **Medical Record Management** ).

### 3.7.3 Update Medical Record Details

****

**Explanation :** When entering "3" in the main page of **Medical Record Management**, the user will go to the **Update Medical Record Details** section. All the existing patients will be displayed in a neatly organized grid format along with the number of total patients listed at the bottom. User is prompted to enter the patient ID needed to update medical records.

### 3.7.3.1 Update Medical Record Details (After updating a medical record detail)

****

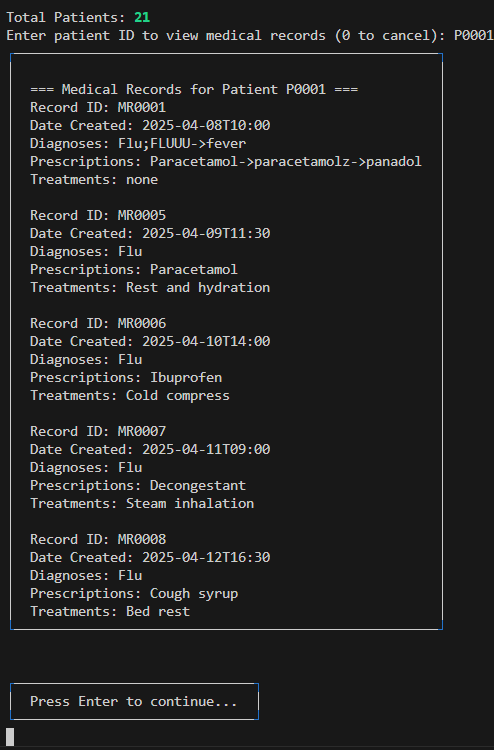
**Explanation :** After entering the patient ID, its medical records will be shown and the user is required to choose the medical record needed to update. Updating a record requires the user to rewrite its details. Once finished updating, "Medical record updated successfully!" will be displayed and ask the user to press Enter to continue ( back to the main page of the **Medical Record Management** ).

### 3.7.4 View Medical Record

****

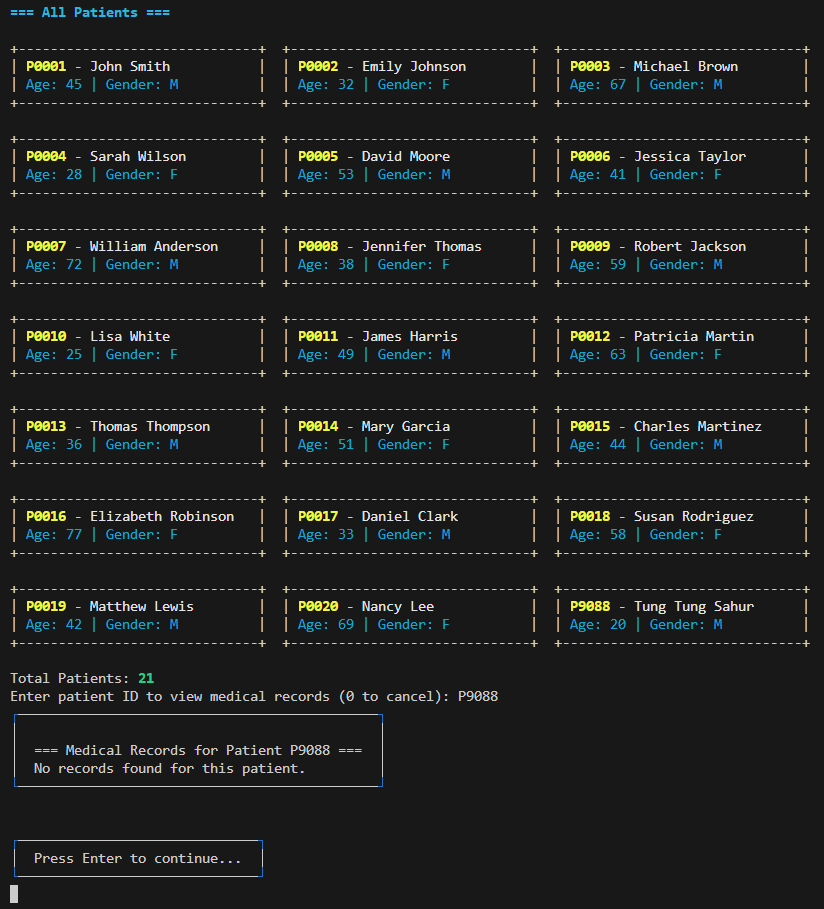
**Explanation :** When entering "4" in the main page of **Medical Record Management**, the user will go to the **View Medical Record** section. All the existing patients will be displayed in a neatly organized grid format along with the number of total patients listed at the bottom. User is required to enter a patient ID to view the medical records.

### 3.7.4.1 View Medical Records (Chosen an appointment to view has records)

****

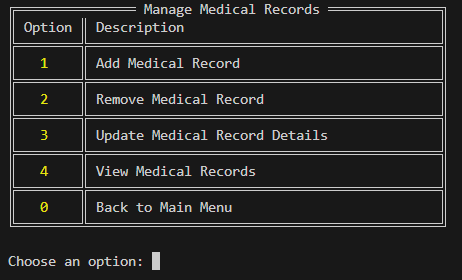
**Explanation :** After the user enters the patient ID, the patient’s medical records will be listed. Once finished viewing, the user is required to press Enter to continue (back to main page ofthe **Medical Record Management)**

### 3.7.4.2 View Medical Records (Chosen an appointment to view has no records)

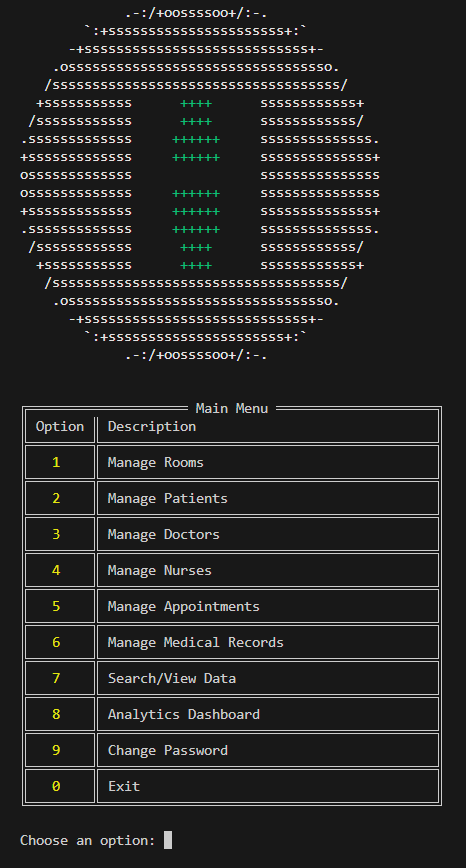
****

**Explanation :** If no medical record exists for the patient with the patient ID entered by the user, "No records found for this patient" will be shown in the box.

### 3.7.5 Back to Main Menu

****

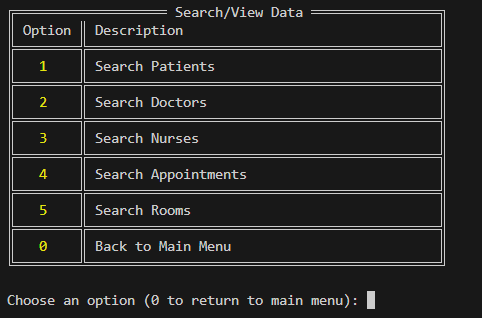
**Note :** Full page image can refer back to the image in 3.7

****

**Note :** Full page image can refer back to the image in 3.1.2

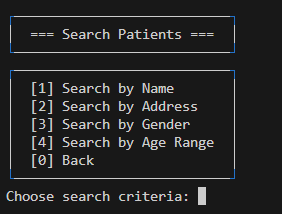
**Explanation :** Entering "0" will lead the user back to the **Main Menu**.

## 3.8 Search and View Management

****

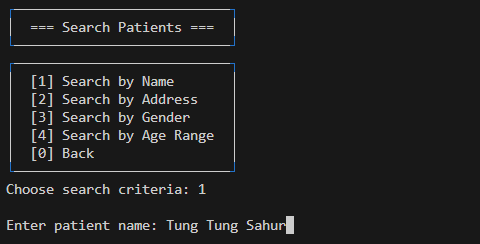
**Explanation :** After entering "7" in the **Main Menu**, the user will go to the **Search and View Data** section. A menu of all available functions for the section will be displayed. User is required to choose an option to perform a specific function of the **Medical Record Management**.

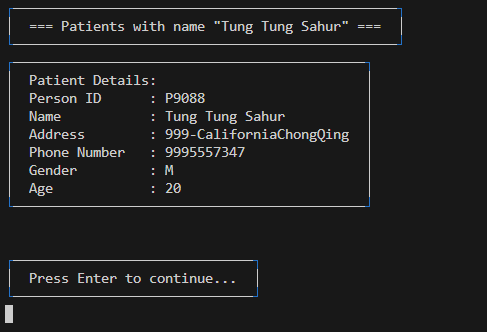
### 3.8.1 Search Patients

****

**Explanation :** When the user enters "1" in the main page of the **Search and View Data**, it will go to the **Search Patients** section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria.

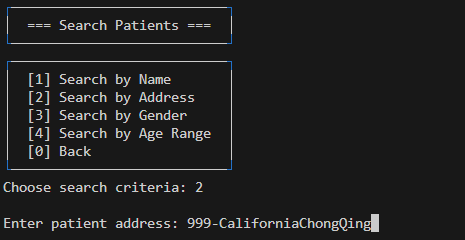
### 3.8.1.1 Search Patients (Search by name)

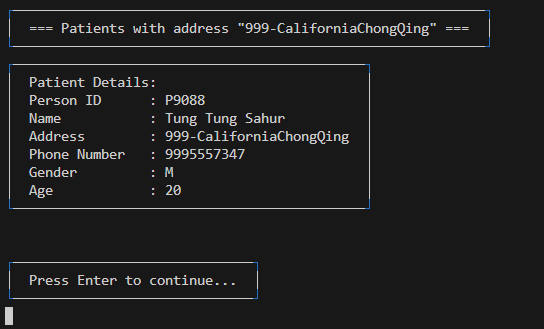
****

****

**Explanation :** If the user enters "1", the system will ask the user to enter the **patient's name** needed to search. After entering, all the patient' details of that patient will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Patients**) .

### 3.8.1.2 Search Patients (Search by address)

****

****

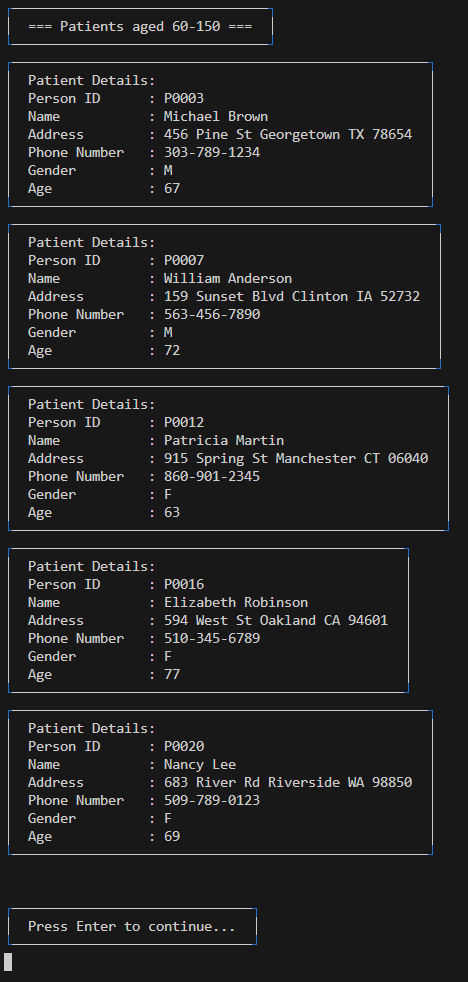
**Explanation :** If the user enters "2", the system will ask the user to enter the **patient's address** needed to search. After entering, all the patient' details of each patient with that address will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Patients**) .

### 3.8.1.3 Search Patients (Search by gender)

|  |
| --- |

**Explanation :** If the user enters "3", the system will ask the user to enter the **patient's gender** needed to search. After entering, all the patient' details of each patient with that gender will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to the main page of **Search Patients**) .

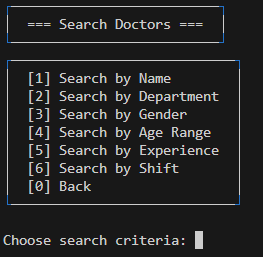
### 3.8.1.4 Search Patients (Search by age range)

****

**Note :**

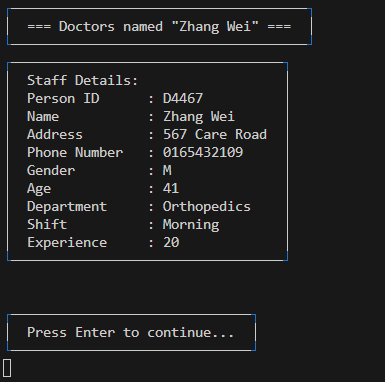
**Explanation :** If the user enters "4", the system will ask the user to enter the **patient's age range** needed to search. After entering, all the patient' details of each patient within that age range will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Patients**) .

### 3.8.2 Search Doctors

****

**Explanation :** When the user enters "2" in the main page of the **Search and View Data**, it will go to the **Search Doctor** section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria.

### 3.8.2.1 Search Doctors (Search by Name)

****

**Explanation :** If the user enters "1", the system will ask the user to enter the **doctor's name** needed to search. After entering, all the doctor' details of each doctor with that name will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 3.8.2.2 Search Doctors (Search by Department)

|  |
| --- |

**Explanation :** If the user enters "2", the system will ask the user to enter the **doctor's department** needed to search. After entering, all the doctor' details of each doctor within that department will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 3.8.2.3 Search Doctors (Search by Gender)

|  |
| --- |
|  |

**Explanation :** If the user enters "3", the system will ask the user to select the **doctor's gender** needed to search. After entering, all the doctor' details of each doctor with that gender will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 3.8.2.4 Search Doctors (Search by Age Range)

|  |
| --- |

**Explanation :** If the user enters "4", the system will ask the user to select the **doctor's age range** needed to search. After entering, all the doctor' details of each doctor within that age range will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 

### 3.8.2.5 Search Doctors (Search by Experience)

|  |
| --- |
|  |

**Explanation :** If the user enters "5", the system will ask the user to select the **doctor's experience level** needed to search. After entering, all the doctor' details of each doctor within that experience-level will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 3.8.2.6 Search Doctors (Search by Shift)

|  |
| --- |

**Explanation :** If the user enters "6", the system will ask the user to select the **doctor's working shift** needed to search. After entering, all the doctor' details of each doctor within that shift will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Doctors**) .

### 3.8.3 Search Nurses

|  |
| --- |

### **Explanation :** When the user enters "3" in the main page of the **Search and View Data**, it will go to the **Search Nurses** section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria.

### 3.8.3.1 Search Nurses (Search by Name)

|  |
| --- |

**Explanation :** If the user enters "1", the system will ask the user to enter the **nurse's name** needed to search. After entering, all the nurses' details of each nurse with that name will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 3.8.3.2 Search Nurses (Search by Department)

|  |
| --- |

**Explanation :** If the user enters "2", the system will ask the user to select the **nurse's department** needed to search. After entering, all the nurses' details of each nurse within that department will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 3.8.3.3 Search Nurses (Search by Gender)

|  |
| --- |
|  |

**Explanation :** If the user enters "3", the system will ask the user to select the **nurse's gender** needed to search. After entering, all the nurses' details of each nurse within that gender will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 

### 3.8.3.4 Search Nurses (Search by Age Range)

|  |
| --- |

**Explanation :** If the user enters "4", the system will ask the user to select the **nurse's age range** needed to search. After entering, all the nurses' details of each nurse within that age range will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 3.8.3.5 Search Nurses (Search by Experience)

|  |
| --- |
|  |

**Explanation :** If the user enters "5", the system will ask the user to select the **nurse's experience level** needed to search. After entering, all the nurses' details of each nurse within that experience level will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 3.8.3.6 Search Nurses (Search by Shift)

|  |
| --- |
|  |

**Explanation :** If the user enters "6", the system will ask the user to select the **nurse's working shift** needed to search. After entering, all the nurses' details of each nurse within that shift will be displayed in a box. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Nurses**) .

### 3.8.4 Search Appointments

|  |
| --- |
|  |

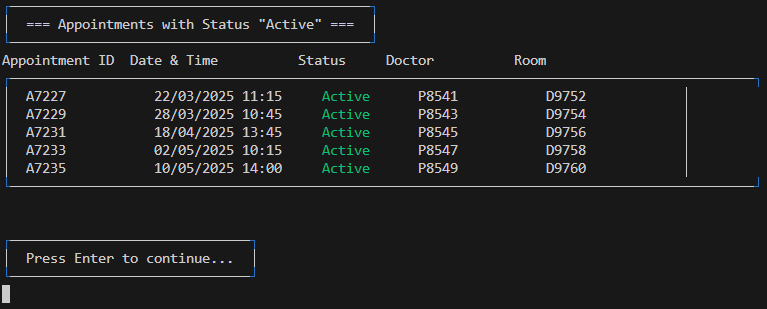
**Explanation :**

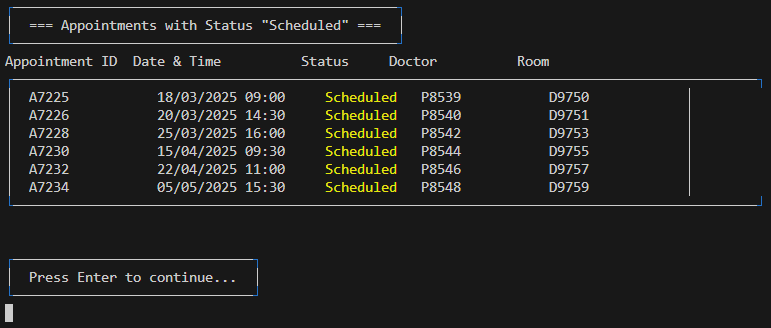
### When the user enters "4" in the main page of the **Search and View Data**, it will go to the **Search Appointment** section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria.

If the user enters "1", the system will ask the user to select the **appointment's status** needed to search.

If the user enters "2", the system will ask the user to select the **appointment's date range** needed to search.

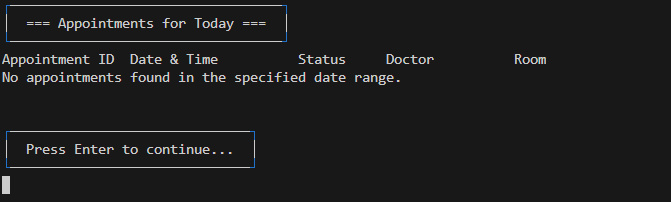
### 3.8.4.1 Search Appointments (Search by status)

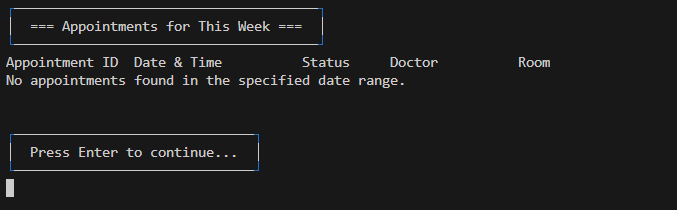
****

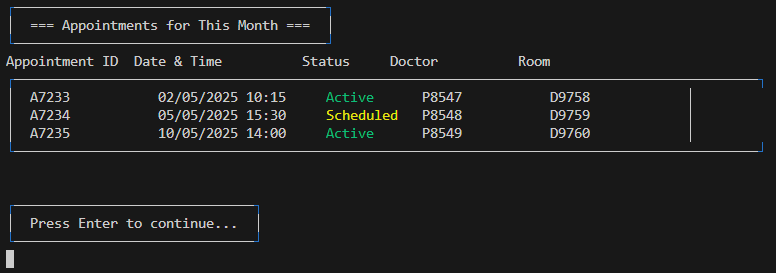
****

**Explanation :** After entering, all the appointments and its details with that status will be displayed in a table. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Appointment**) .

### 3.8.4.2 Search Appointments (Search by date)

****

****

****

**Explanation :** After entering, all the appointments and its details within that date range will be displayed in a table. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Appointment**) .

### 3.8.5 Search Rooms

|  |
| --- |
|  |

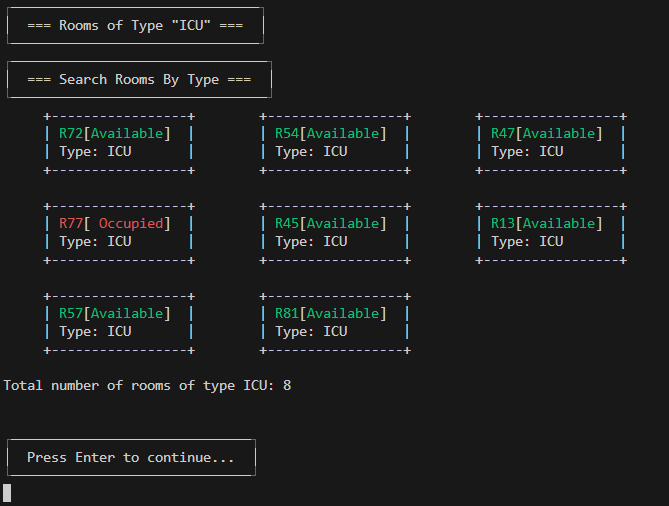
**Explanation :**

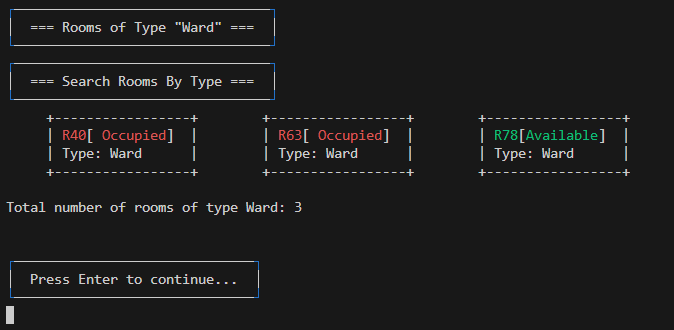
### When the user enters "5" in the main page of the **Search and View Data**, it will go to the **Search Room** section. A menu of several search criteria which indicate data used for the search function for this section will be displayed. User is required to choose one search criteria.

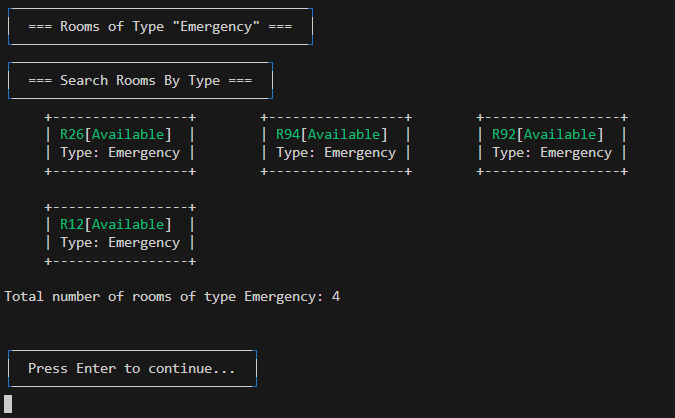
If the user enters "1", the system will ask the user to select the **room type** needed to search.

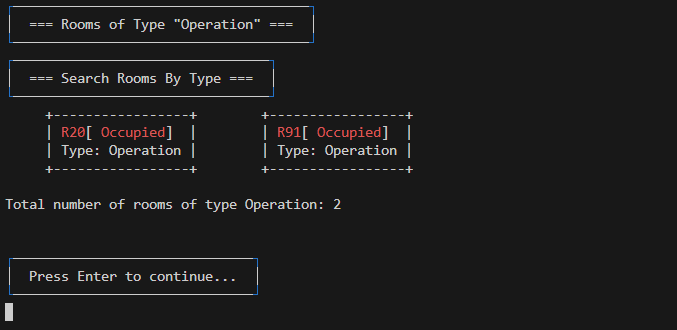
If the user enters "2", the system will ask the user to select the **room status** needed to search.

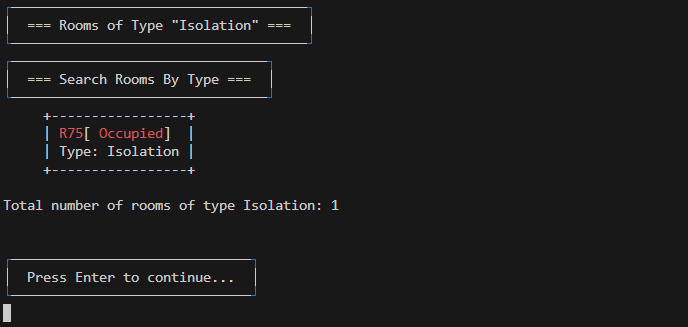
### 3.8.5.1 Search Rooms (Search by room type)

****

****

****

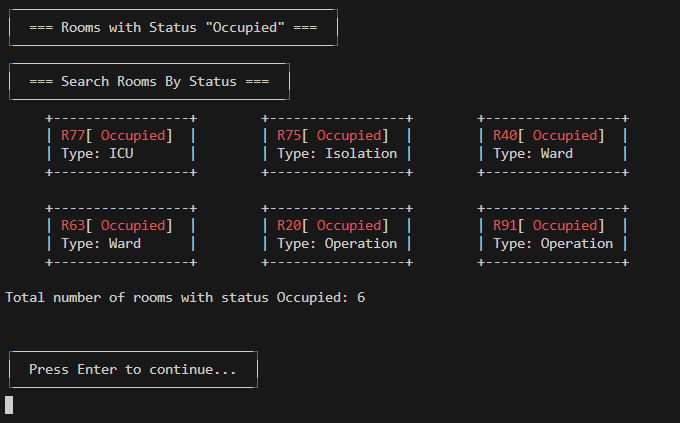
****

****

**Explanation :** After entering, each room with that room type will be displayed in a box with its room type. A total number of rooms for that room type will also be displayed. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Rooms**) .

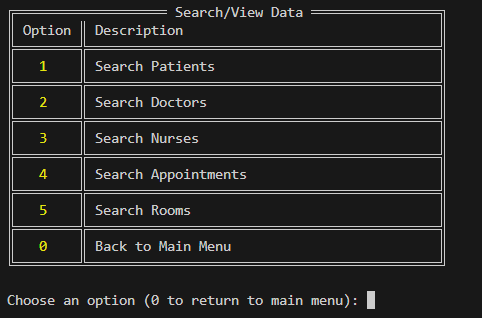
### 3.8.5.2 Search Rooms (Search by status)

****

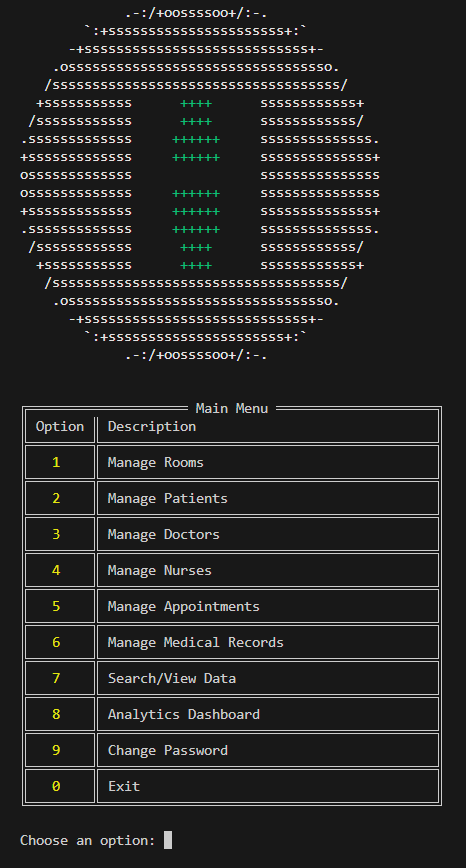
****

**Explanation :** After entering, each room with that status will be displayed in a box with its room type. A total number of rooms for that status will also be displayed. Once finished viewing, the user is prompted to press Enter to continue (back to **Search Rooms**) .

### 3.8.6 Back to Main Menu

****

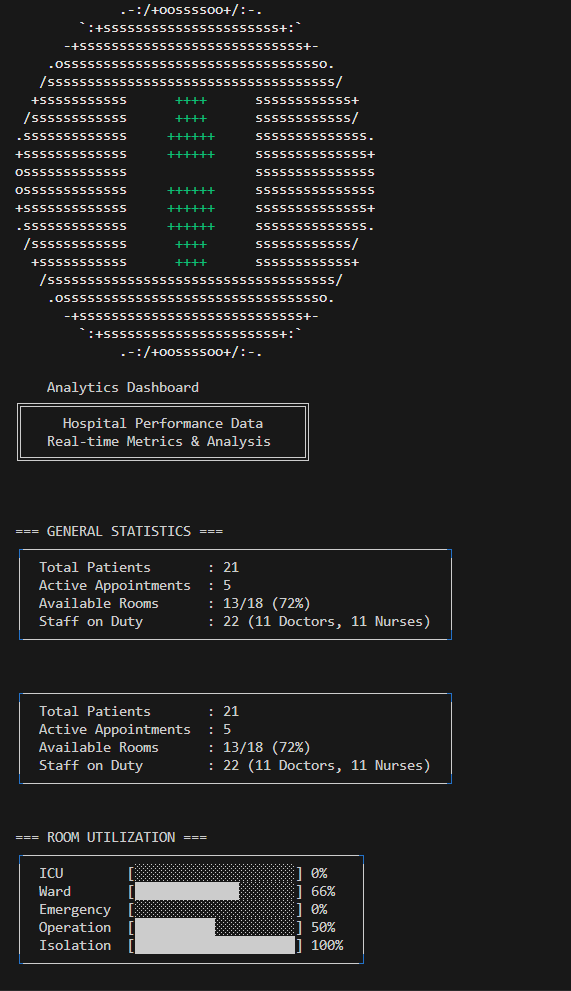
**Note :** Full page image can refer back to the image in 3.8

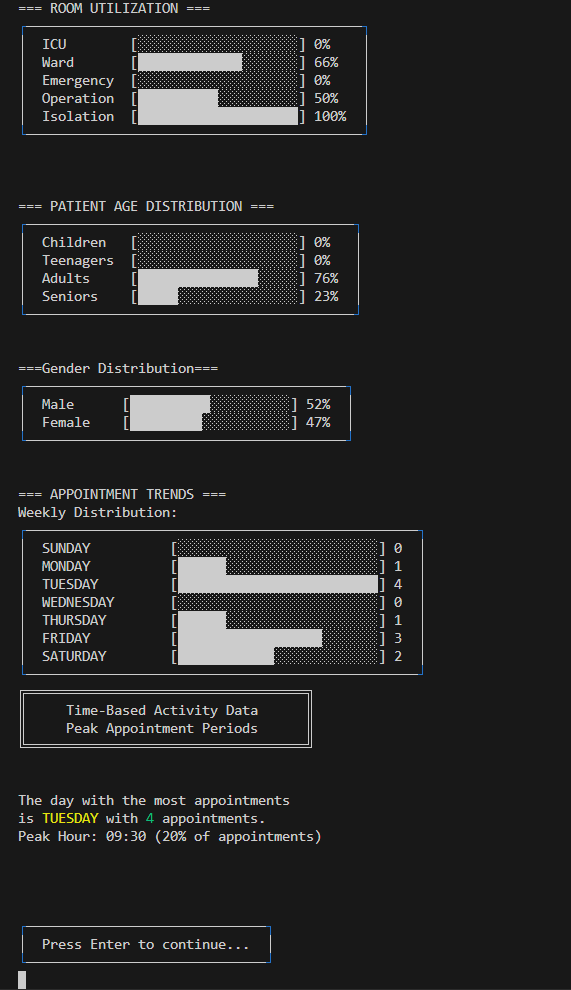
****

**Note :** Full page image can refer back to the image in 3.1.2

**Explanation :** Entering "0" in the **Search and View Data** will lead the user back to the **Main Menu**.

## 3.9 Analytic Dashboards

****

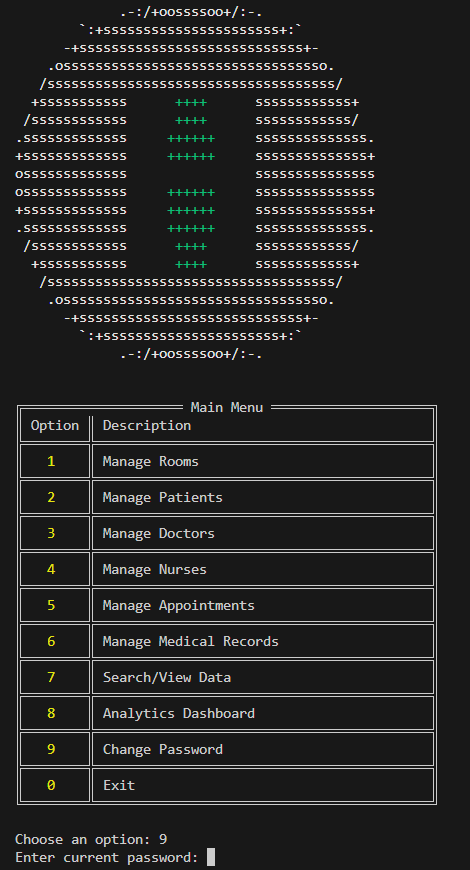
****

**Explanation :**

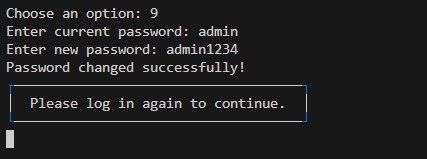
After entering "8" in the **Main Menu**, the user will go to the **Analytic Dashboard** section. A **general statistics** for each part of the hospital management system will be displayed in a box. **Room utilization**, **patient age distribution**, **gender distribution** and the **appointment trends** will be displayed by using the horizontal bar chart format. There is also a message showing the peak appointment periods.

Once finished viewing, the user is required to press Enter to continue (back to the **Main Menu)** .

## 3.10 Change Password

****

### 3.10.1 Change Password (After changing a new password)

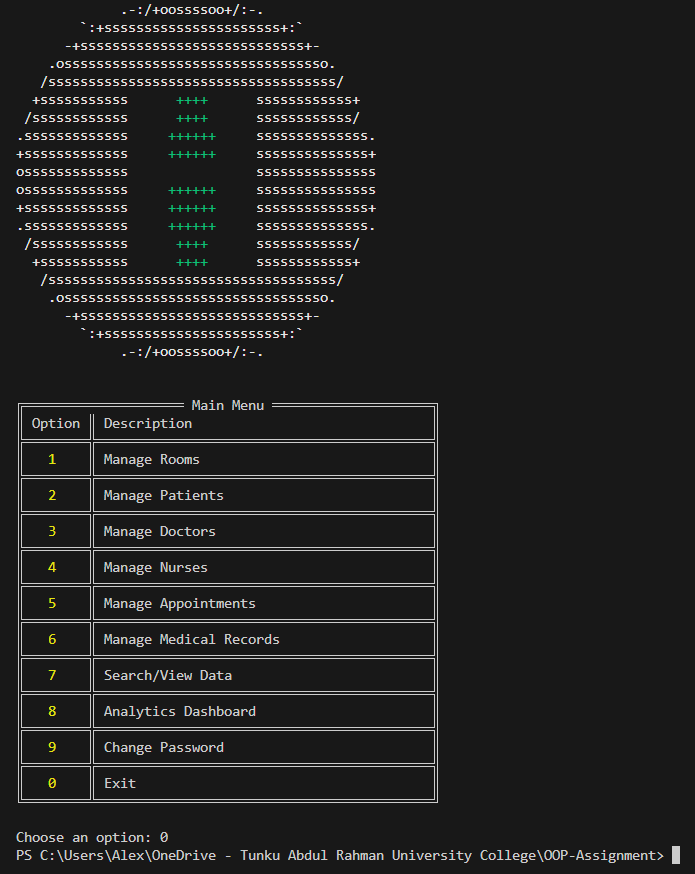
****

**Explanation :**

After entering "9" in the **Main Menu**, the user will go to the **Change Password** section.The user is required to enter the current password for confirmation before changing the password. Once confirmed, the user is required to enter the new password.

A message "Password changed successfully!" will be displayed once the change is completed and the user is prompted to press Enter to log in again to continue ( back to the login page ).

## 3.11 Exit

****

**Explanation :** After entering "0" in the **Main Menu**, the system will close.

# 4.0 Conclusion

The Hospital Management System (HMS) stands as a fully integrated, end-to-end solution for modern hospital administration. By unifying patient management, staff coordination, resource allocation, appointment scheduling, and medical-record keeping within a single, modular platform, the HMS ensures that every operational aspect is both streamlined and data-driven. Its robust validation, persistent storage, and comprehensive audit trails guarantee data integrity and regulatory compliance, while advanced analytics and real-time dashboards empower administrators to make proactive, evidence-based decisions.

The system’s flexibility allows for seamless adaptation to changing healthcare demands, whether scaling to accommodate more patients, integrating new personnels, and responding to constant updates. Role-based access controls and secure authentication safeguard sensitive information, ensuring that users interact only with the data and functions relevant to their responsibilities. The HMS’s extensible architecture supports future enhancements, such as integration with external health information systems or the addition of new modules for laboratory, pharmacy, or billing management.

Ultimately, the HMS not only reduces administrative overhead and operational bottlenecks but also enhances the quality and safety of patient care. By providing instant access to critical information, automating routine workflows, and delivering actionable insights through comprehensive analytics, the system enables hospitals to operate at peak efficiency. This positions healthcare institutions to respond rapidly to emergencies, optimize resource utilization, and continuously improve service delivery—ensuring that both clinical and administrative teams are equipped to meet the evolving challenges of the healthcare landscape.

# 

# 

# 

# 

# 

# 5.0 References

1. *Password masking in the Java programming language.* (2004, July 16). <https://www.cse.chalmers.se/edu/year/2016/course/TDA602/Eraserlab/pwdmasking.html#:~:text=To%20mask%20the%20password%20field,new%20TextField(8)%3B%20password>.
2. *Java Interfaces (With Examples).* (n.d.). <https://www.programiz.com/java-programming/interfaces>
3. Polymorphism in Java - javatpoint. (n.d.). [www.javatpoint.com](http://www.javatpoint.com). <https://www.javatpoint.com/runtime-polymorphism-in-java>
4. Inheritance (The JavaTM Tutorials > Learning the Java Language > Interfaces and Inheritance). (n.d.). <https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html>
5. UML Class Diagram Tutorial. (n.d.). <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>