Declaration of Original Work for SC2002/CE2002/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed, and submitted as a collective effort by the group members listed below.

We have honoured the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

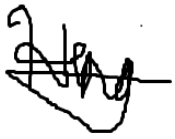
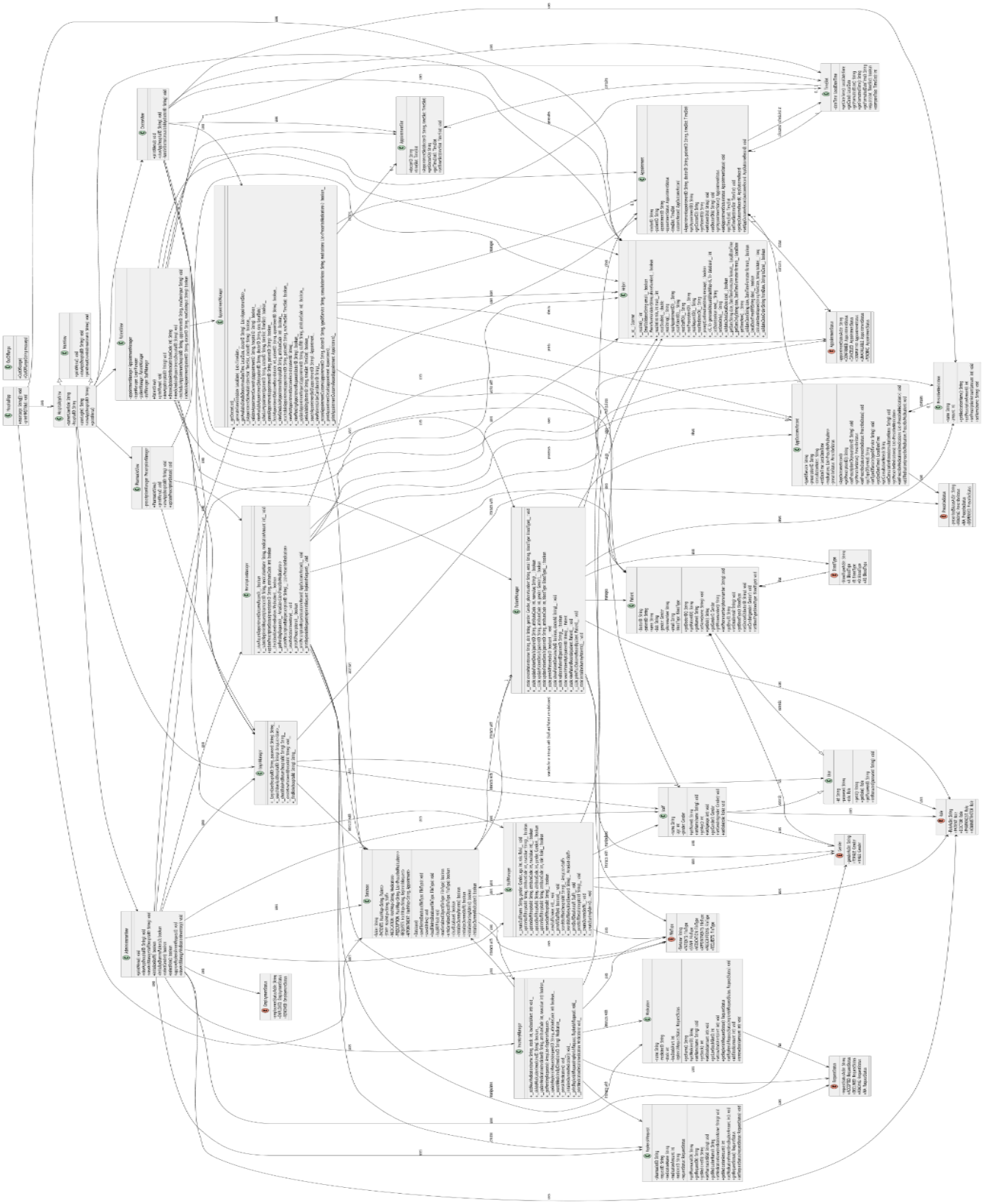
Name	Course	Lab Group	Signature /Date
Liew Jia Wei	SC2002	SCSG (Group 5)	 21/11/2024
Ng Weiu Seann	SC2002	SCSG (Group 5)	 21/11/2024
Kee Lek Heng	SC2002	SCSG (Group 5)	 21/11/2024
Benjamin Kam Jia Zhiang	SC2002	SCSG (Group 5)	 21/11/2024
Balaji Abarnasri	SC2002	SCSG (Group 5)	 21/11/2024

Table of Contents

A. UML Class Diagram

Assumption: There is one admin account initialised in the system before the user runs the program.



B. Github Repository Link

Below is the link to our project repository.

https://github.com/KeeLekHeng/SC2002

C. Additional features and functionalities

In addition to the core functionalities of the HMS, we added several additional features to enhance user experience and the system.

1. Doctor Rating System
   Users can rate doctors after appointments. This functionality allows patients to provide feedback on their experiences with the doctors, helping to ensure continuous improvement in service quality.This feature facilitates the collection of feedback for doctors, enabling the hospital to maintain patient satisfaction.

2. Viewing Upcoming and All Appointments
   Both patients and doctors can view their upcoming scheduled appointments of all pasts and upcoming appointments. This provides an overview of appointments, ensuring that all parties stay informed about their schedules.

3. Phone Number and Email
   A validation function was implemented to ensure that phone numbers input by users start with the country code “65”. This ensures consistency and accuracy in the database. An email format validator was added to ensure that users provide valid email addresses during registration. This validator checks that the email contains the “@” symbol and a valid domain.

4. Password features
   If a user logs in with the default password, the system prompts them to update their password immediately. When users input their passwords during login, the password field is hidden for privacy and security. This prevents unauthorised users from viewing the password during login.

D. Approach & Design Considerations

Our team used the SOLID design principle we learnt in our lectures and the best practices for object-oriented programming. Below are the considerations and approaches we adopted:

SOLID Design Principle

1. Single-Responsibility Principle (SRP)
   We implemented the Model-View-Controller (MVC) architecture to ensure each class has a distinct role and responsibility.

   Model Classes: Represent and manage the data by communicating with the database and provide data to controllers upon request. They do not interact with the View class.

   View Classes: Handles the user interface, presents data gathered by the controller. The view class is exclusively responsible for rendering information and accepting user input. It also only interacts with the controller class.

   Controller Classes: Acts as a mediator between the model and view by managing the application logic by requesting data from model and processing it and returning it to view.

   Example: The PatientView class under view helps to accept inputs. Then, the PatientManager controller handles tasks like retrieving patient records, adding new patients or updating existing patient records. The Patient class has the structure where various objects are stored such as PatientID,phone number and blood type.

2. Open-Closed Principle (OCP)
   The system is designed to be extensible without modifying existing code.

   Example: The MainView abstract class serves as a parent for all specific View classes such as PatientView, DoctorView and PharmacistView. Adding new roles such as a new user type will involve creating subclasses of MainView without changing the existing implementation of MainView. Hence, OCP is followed as the MainView is open for extension but closed for modification.

3. Liskov Substitution Principle (LSP)
   Subclasses can replace their parent classes without altering program behaviour.

Example: For the view subclasses, we do not expect more input from the user and do not provide less than the base case. The DoctorView and PatientView classes override methods like printMenu() and viewApp() to provide specialised implementations. Hence, the subclasses maintain the contract established by the base MainView class, ensuring consistency.

4. Interface Segregation Principle (ISP)
   Although interfaces were minimally used, each class was designed to only focus on the method relevant to its role to avoid bloated interfaces.

5. Dependency Injection Principle (DIP)
   High-level modules are designed to be independent from lower-level modules

   Example: High-level modules like HospitalApp are designed to be independent of lower-level modules like View classes so that the MainView abstract class is used as an intermediary for all View classes, thus ensuring loose coupling and low dependency on the code. If a type of view required changing there would be minimal changes to the overall system.

Object-Oriented Design Principles

1. Inheritance
   Implementation: The MainView acts as a parent class for all user-specific views such as DoctorVIew and PatientVIew. Each of these child classes extends the MainView class and inherits its core functionalities.

2. Polymorphism
   Implementation: All other child View classes extend the MainView parent class and overrides the inherited methods to have different implementations.

3. Abstraction
   Implementation: At the model level, attributes such as patient details, staff roles, and medication information are represented abstractly. This allows for flexibility when modifying or extending the system.

4. Encapsulation
   Implementation: The system uses intermediary methods like validateAppointmmentOwnership() to ensure there is controlled access to sensitive data, such as appointments and user information stored in the database. Instead of

allowing direct access to raw data structures like Database.APPOINTMENT, the method helps to validate user roles before allowing them to access and change any appointments. For example, when a patient attempts to cancel an appointment, the system verifies that the patient owns the appointment by comparing the patient's ID with the record in the database. Thus, this ensures data integrity and prevents unauthorised access by encapsulating the validation and access logic within dedicated methods and protecting the raw data.

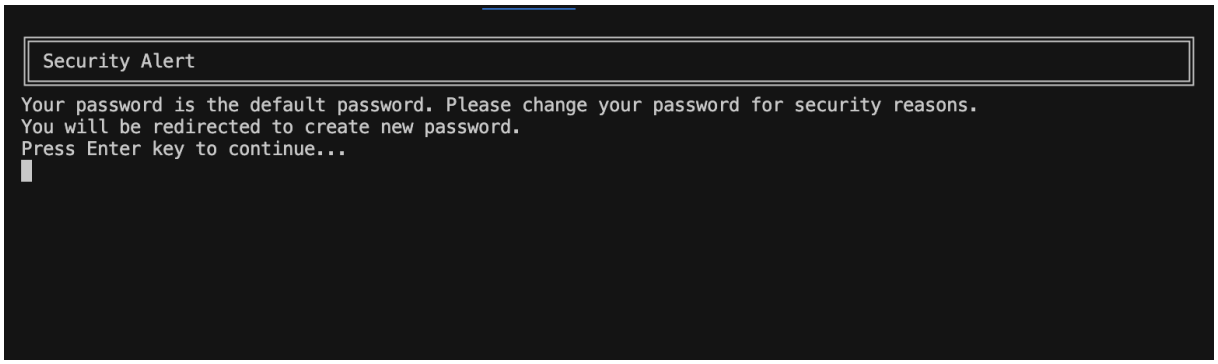
5. Loose Coupling
   Implementation: The system uses identifiers such as user IDs, appointment IDs, and medication names to pass information between classes, rather than tightly coupling one class to another. This design adheres to the Model-View-Controller (MVC) architecture, where Models handle data and business logic, Views handle user interfaces and Controllers mediate between models and views, ensuring separation of concerns.

E. Testing

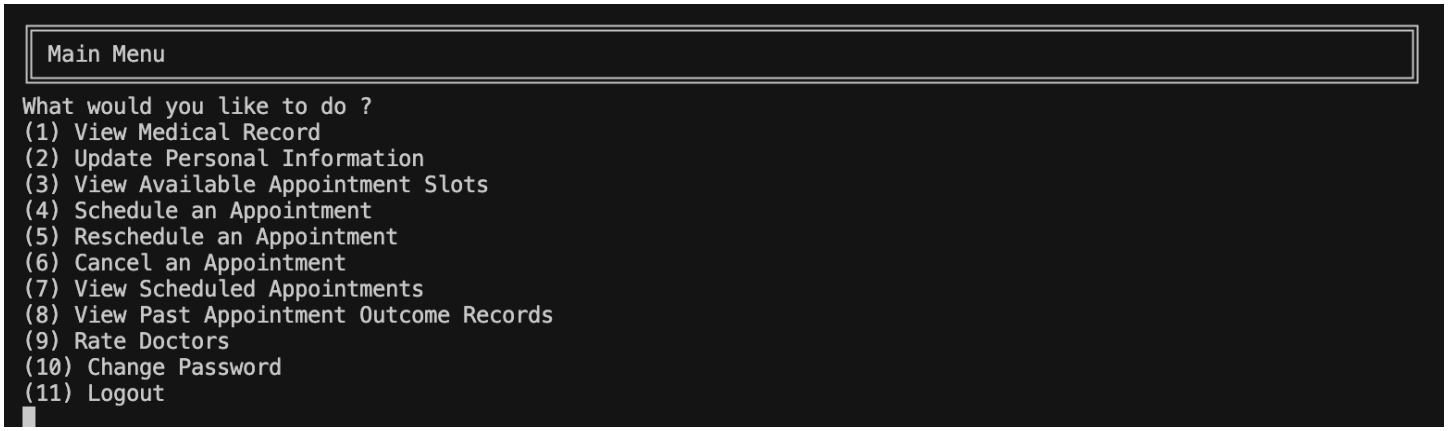
- Home Page

- Create Patient

- Prompt to reset default password



- Patient View



- Patient View Medical Record

```
 Main Menu > View Medical Record

----------------------------------------
Patient ID          : P0001
Name                : Kee Lek Heng
Date of Birth       : 2005-07-28
Gender              : MALE
Blood Type          : A
Phone Number        : 6585445065
Email               : keel0004@e.ntu.edu.sg
Previous Medical Records for Kee Lek Heng: None
----------------------------------------
Press Enter key to continue...
```

- Updating Phone number with validator

```
 Main Menu > Update Personal Information

What would you like to update ?
(1) Phone Number
(2) Email
1
Enter new phone number:
Phone number should begin with 65 and be a valid Singapore number
1234567890
Unsuccessful. Phone number is invalid
Press Enter key to continue...
```

- Updating email with validator

```
 Main Menu > Update Personal Information

What would you like to update ?
(1) Phone Number
(2) Email
2
Enter new email:
wedisfgdjfknsdk.com
Unsuccesful. Email is invalid
Press Enter key to continue...
```

- Successful personal information update (email and phone number)

```
Main Menu > View Medical Record

----------------------------------------
Patient ID          : P0001
Name                : Kee Lek Heng
Date of Birth       : 2005-07-28
Gender              : MALE
Blood Type          : A
Phone Number        : 6584160882
Email               : giveUsA+@gmail.com
Previous Medical Records for Kee Lek Heng: None
----------------------------------------
Press Enter key to continue...
```

- View available appointment slots

```
Main Menu > View Available Appointment Slots

Enter the date you wish to view available slots for (Format: 'yyyy-MM-dd' )
Please enter the date in this format: 'yyyy-MM-dd'
2024-12-12
----------------------------------------
Our hospital has 2 doctors. Doctors List:
Name: Dr. Carol White | DoctorID: D003 | Rating: no ratings
Name: Dr. Alice Smith | DoctorID: D002 | Rating: no ratings
----------------------------------------
Enter doctor ID (DXXX):
```

```
Doctor's name:
Dr. Alice Smith
----------------------------------------
Available Appointment Slots for 2024-12-12:
09:00am
10:00am
11:00am
12:00pm
01:00pm
02:00pm
03:00pm
04:00pm
05:00pm
----------------------------------------
Press Enter key to continue...
```

- Scheduling appointment slot

```
Main Menu > View Available Appointment Slots
Enter the date you wish to view available slots for (Format: 'yyyy-MM-dd' )
Please enter the date in this format: 'yyyy-MM-dd'
2024-12-12
----------------------------------------
Our hospital has 2 doctors. Doctors List:
Name: Dr. Carol White | DoctorID: D003 | Rating: no ratings
Name: Dr. Alice Smith | DoctorID: D002 | Rating: no ratings
----------------------------------------
Enter doctor ID (DXXX):
D002
Doctor's name:
```

```
What date would you like to schedule to?
Please enter the date in this format: 'yyyy-MM-dd'
2024-12-12
Available Slots:
1. 09:00am
2. 10:00am
3. 11:00am
4. 12:00pm
5. 01:00pm
6. 02:00pm
7. 03:00pm
8. 04:00pm
9. 05:00pm
Enter the number corresponding to the Time Slot you wish to schedule your appointment:
1
```

```
8. 04:00pm
9. 05:00pm
Enter the number corresponding to the Time Slot you wish to schedule your appointment:
1
Appointment scheduled for 12/12/24 at 09:00am. Appointment Details:
----------------------------------------
Appointment ID      : A00001
Patient ID          : P0001
Doctor ID           : D002
Doctor's Name       : Dr. Alice Smith
Appointment Status  : PENDING
Time Slot           : 12/12/24 09:00am
Outcome Record      : None
----------------------------------------
Press Enter key to continue...
```

- View Scheduled appointment slot

```
 Main Menu > View Scheduled Appointments

What would you like to view ?
(1) Upcoming Appointments
(2) All Appointments
(3) Back
```

```
Here are your scheduled appointments:
----------------------------------------
Appointment ID      : A00001
Patient ID          : P0001
Doctor ID           : D002
Doctor's Name       : Dr. Alice Smith
Appointment Status  : PENDING
Time Slot           : 12/12/24 09:00am
Outcome Record      : None
----------------------------------------
Press Enter key to continue...
```

- Rating Doctors

```
 Main Menu > Rating Doctors

----------------------------------------
Our hospital has 2 doctors. Doctors List:
Name: Dr. Carol White | DoctorID: D003 | Rating: 5.0
Name: Dr. Alice Smith | DoctorID: D002 | Rating: no ratings
----------------------------------------
Enter the staff ID of the doctor to be rated (DXXX):
D002
Enter a rating out of 5 (5 : excellent | 1 : poor):
5
```

- Doctor View

```
 Main Menu

What would you like to do ?
(1) View Patient Medical Records
(2) Update Patient Medical Records
(3) View Personal Schedule
(4) Set Availability for Appointments
(5) Accept or Decline Appointment Requests
(6) View Upcoming Appointments
(7) Record Appointment Outcome
(8) Change Password
(9) Logout
```

- View Patient Medical Records

```
Patient ID: P0001 | Patient Name: Kee Lek Heng
Enter Patient ID (Format: PXXXX) or type 'back' to go back: P001
Invalid format! Please try again.
Enter Patient ID (Format: PXXXX) or type 'back' to go back: P0001
----------------------------------------
Patient ID          : P0001
Name                : Kee Lek Heng
Date of Birth       : 2005-07-28
Gender              : MALE
Blood Type          : A
Phone Number        : 6584160882
Email               : giveUsA+@gmail.com
Previous Medical Records for Kee Lek Heng: None
----------------------------------------
Press Enter key to continue...
```

- Accept or decline appointment requests

```
 Main Menu > Accept or Decline Appointment Requests

----------------------------------------
Appointment ID      : A00001
Patient ID          : P0001
Doctor ID           : D002
Doctor's Name       : Dr. Alice Smith
Appointment Status  : PENDING
Time Slot           : 12/12/24 09:00am
Outcome Record      : None
----------------------------------------
Enter appointment ID to update request:
A00001
Accept or Decline? (1: Accept, 2: Decline)
1
```

- View Upcoming Appointments

```
Main Menu > View Upcoming Appointments

Here are your scheduled appointments:
----------------------------------------
Appointment ID      : A00001
Patient ID          : P0001
Doctor ID           : D002
Doctor's Name       : Dr. Alice Smith
Appointment Status  : CONFIRMED
Time Slot           : 12/12/24 09:00am
Outcome Record      : None
----------------------------------------
Press Enter key to continue...
```

- Update patient medical records

```
Main Menu > Update Patient Medical Records

Appointment ID: A00001 | Patient Name: Kee Lek Heng | Appointment Time Slot: 12/12/24 09:00am
Enter Appointment ID (Format: AXXXXX) or type 'back' to go back: A00001
Enter patient's diagnosis:
fever
Enter patient's treatment:
drink water
Enter list of medications to prescribe (type 'done' to finish):
Enter medication name (or 'done' to finish):
ibuprofen
Enter prescription amount (or type 'done' to cancel):
20
Enter medication name (or 'done' to finish):
done
```

- Pharmacist

```
Main Menu

What would you like to do ?
(1) View Appointment Outcome Record
(2) Update Prescription Status
(3) View Medication Inventory
(4) Submit Replenishment Request
(5) Change Password
(6) Logout
```

- Print medical record of the patient without any outcome records

```
----------------------------------------
Patient ID          : P0001
Name                : Kee Lek Heng
Date of Birth       : 2005-07-28
Gender              : MALE
Blood Type          : A
Phone Number        : 6585445065
Email               : keel0004@e.ntu.edu.sg
Previous Medical Records for Kee Lek Heng: None
----------------------------------------
Press Enter key to continue...
```

- Show pending prescription requests

```
--------------------------------------
Record Uploaded Time: 2024-11-21T23:49:02.153100200
Type of Service     : xray
Consultation Notes  : xray 4 more times
Prescription ID     : P00003
Prescription Status : PENDING
Prescribed Medications::
  Medication Name     : ibuprofen
  Amount              : 40
  Medication Name     : paracetamol
  Amount              : 20
--------------------------------------
--------------------------------------
Record Uploaded Time: 2024-11-21T23:48:16.262962900
Type of Service     : Xray
Consultation Notes  : not enough xray
Prescription ID     : P00003
Prescription Status : PENDING
Prescribed Medications::
  Medication Name     : ibuprofen
  Amount              : 20
--------------------------------------
Press Enter key to continue...
```

- Dispense medication to patient (if too low, then low stock level alert is triggered and pharmacist see it)

```
---------------------------------------
Printing all prescriptions that have not been dispensed:
---------------------------------------
Prescription ID     : P00003
Medication Name     : ibuprofen
Amount              : 40
---------------------------------------
Medication Name     : paracetamol
Amount              : 20
---------------------------------------
Prescription ID     : P00003
Medication Name     : ibuprofen
Amount              : 20
---------------------------------------
---------------------------------------
Enter Prescription ID (Format: PXXXXX) or type 'back' to go back: P00003
Select an action:
1. Dispense
2. Back
1
Low Stock Level Detected for ibuprofen! Send Replenish Request Urgently! Current Stock Level: 10
Prescription dispensed successfully.
Press Enter key to continue...
```

- View medication inventory

```
Medications in inventory:
----------------------------------------
MedicationID        : M0003
Medication Name     : amoxicillin
Current Stock       : 75
Low Stock Alert     : 15
----------------------------------------
----------------------------------------
MedicationID        : M0002
Medication Name     : paracetamol
Current Stock       : 80
Low Stock Alert     : 20
----------------------------------------
----------------------------------------
MedicationID        : M0001
Medication Name     : ibuprofen
Current Stock       : 10
Low Stock Alert     : 10
----------------------------------------
Press Enter key to continue...
```

- Submit new replenish request

```
Enter the medication name:
ibuprofen
Enter the quantity:
40
New Replenish Request added into database! Request Details:
----------------------------------------
PharmacistID        : P004
RequestID           : R0001
Medication Name     : ibuprofen
Required Stock      : 40
Current Status      : PENDING
----------------------------------------
Press Enter key to continue...
```

- Approve replenish request

```
----------------------------------------
PharmacistID        : P004
RequestID           : R0001
Medication Name     : ibuprofen
Required Stock      : 40
Current Status      : PENDING
----------------------------------------
Do you want to approve this request?
(1) Approve
(2) Reject
(3) Back
1
All request processed
Press Enter key to continue...
```

- Reset database and add dummy admin

```
Staff Created! Staff Details:
----------------------------------------
Guest ID            : A001
Name                : Admin
Gender              : Male
Password            : password
EmploymentStatus    : EMPLOYED
----------------------------------------
Database Cleared and Starting Admin initialized
Press Enter key to continue...
```

- Administrator View

```
Main Menu

What would you like to do ?
(1) View and Manage Hospital Staff
(2) View Appointments details
(3) View and Manage Medication Inventory
(4) Approve Replenishment Requests
(5) Change Password
(6) Logout
```

- View and Manage Hospital Staff

```
Main Menu > View and Manage Hospital Individuals

What would you like to do ?
(1) View Staff Details
(2) Create Staff
(3) Update Staff Details
(4) Remove Staff
(5) Create Patient
(6) Initialize Dummy Staff
(7) Initialize Dummy Patients
(8) Initialize Dummy Medications
(9) Clear Database
(10) Back
```

- View Staff

```
Main Menu > View and Manage Hospital Staff > View Staff Details

What would you like to do ?
(1) View Staff by ID
(2) View Staff by Name
(3) View Staff by Age
(4) View Staff by Gender
(5) View Staff by Role
(6) View Staff by Employment Status
(7) Back
```

- Update medicine inventory manually

```
Medications in inventory:
---------------------------------------
MedicationID        : M0003
Medication Name     : amoxicillin
Current Stock       : 75
Low Stock Alert     : 15
---------------------------------------
---------------------------------------
MedicationID        : M0002
Medication Name     : paracetamol
Current Stock       : 80
Low Stock Alert     : 20
---------------------------------------
---------------------------------------
MedicationID        : M0001
Medication Name     : ibuprofen
Current Stock       : 50
Low Stock Alert     : 10
---------------------------------------
Enter Medication ID:
Enter Medicine ID (Format: MXXXX) or type 'back' to go back: M0001
What would you like to update:
(1) Add Stock
(2) Remove Stock
(3) Set Stock
(4) Set Low Stock Limit
(5) Back
2
Enter Quantity to Remove:
20
Stock Updated
Press Enter key to continue...
```

View Staff by ID

```
----------------------------------------
Guest ID            : A004
Name                : Admin John Brown
Gender              : Male
Password            : password
EmploymentStatus    : EMPLOYED
----------------------------------------
----------------------------------------
Guest ID            : P005
Name                : Pharm. Frank Lee
Gender              : Male
Password            : password
EmploymentStatus    : EMPLOYED
----------------------------------------
Press Enter key to continue...
```

- Update Staff

```
Enter Staff ID to Update:
A001
----------------------------------------
Guest ID            : A001
Name                : Admin
Gender              : Male
Password            : password
EmploymentStatus    : EMPLOYED
----------------------------------------
What would you like to update:
(1) Name
(2) Age
(3) Gender
(4) Role
(5) Back
```

- Create Patient

```
Main Menu > View and Manage Hospital Staff > Create Patient

Enter full name:
Seann
Enter your date of birth in this format (yyyy-mm-dd):
2003-10-03
Enter your gender (M/F)
M
Enter phone number:
6584160882
Enter email:
weiuseann03@gmail.com
Enter your bloodType (A, B, O, AB)
A
```

```
weiuseann03@gmail.com
Enter your bloodType (A, B, O, AB)
A
Patient Created! Patient Details:
----------------------------------------
Patient ID          : P0003
Name                : Seann
Date of Birth       : 2003-10-03
Gender              : MALE
Blood Type          : A
Phone Number        : 6584160882
Email               : weiuseann03@gmail.com
Previous Medical Records for Seann: None
----------------------------------------
Press Enter key to continue...
```

- Exit App

```
Thank you for using Hospital Management System
Press Enter key to continue...
```

F. Future Enhancements

To enhance security, future enhancements could allow users to reset their password after verifying their identity by creating a new class to handle password resets. By using the database, we can also track when was the last time they updated their password and prompt users to update their password every 3 months.

For convenience, we can implement automated SMS reminders for patients and doctors for upcoming appointments as the database contains their contact numbers so we would just need to implement a manager class to send out the SMS.

G. Reflection

The assignment has been an interesting experience for every member of our group. Each of us learned a lot about object-oriented programming, but there were some difficulties that we faced along the way.

Difficulties encountered:

Initially we overloaded several classes and functions with too many responsibilities. This violates the Single Responsibility Principle(SRP), making the code harder to manage, understand and debug. This issue arose as we initially started the project without thoroughly understanding the key concepts of the SOLID design approach from the lecture.

Secondly, the time management skills of our group could be further improved. We found it hard to track the project progress as we did not have a very clear vision of the whole system structure. It was quite challenging given the limited time to complete the project, implement additional features, and prepare for the presentation. Near the submission deadline, we were worried about our progress as we struggled to balance the final touches on the system and the preparation required for a successful presentation.

How we solved it:

After identifying the problem, we restructured the code to separate functionalities into distinct classes and methods.We adopted SRP as we ensured each class had a single responsibility. This made the codebase significantly more organised and easier to manage.

Revisiting the lecture notes and following the examples provided a strong foundation for improving the structure.

We also held regular group meetings to discuss the current tasks and set deadlines for each member to complete their assigned work. During these meetings, we updated each other on our progress to ensure that everyone was clear on the project's status. If any issues arose, we communicated openly and ensured that every team member was involved in the discussion, allowing us to address problems collaboratively and efficiently.

Lessons learnt:

Through this experience, we have learned the importance of adhering to design principles. By ensuring that each class or function has a clear, single responsibility, we made our code more maintainable and easier to debug. This not only improved the quality of our code but also made it more readable and organised. We also recognized that well-structured code is key to scalability, allowing us to implement new features more efficiently and with fewer complications. This process taught us the value of planning and refining our design early in the project to avoid unnecessary complexity later on.

Secondly, we learned the importance of effective time management and planning. Without a clear vision of the system structure and specific deadlines, it became challenging to track progress and allocate time effectively. The pressure we faced near the submission deadline taught us the value of breaking down tasks and setting achievable milestones earlier in the project. This will help prevent last-minute stress and ensure a smoother workflow throughout the project.

Conclusion:

In conclusion, creating the Hospital Management System using Java OOP principles has been a highly valuable learning experience for our team. Through this project, we have not only applied key concepts of object-oriented design but also developed a deeper understanding of best practices in software development. Each team member has grown in both technical skills and problem-solving abilities, improving their understanding of system architecture and collaboration. As we reflect on our journey, we are eager to take on more

complex projects in the future, where we can continue to build on this foundation, explore new technologies, and further refine our teamwork and design processes.

H. Member's work contribution and distribution breakdown

Kee Lek Heng – Developed the controller classes and debugged the program to ensure it passed the required test cases. Contributed to the main report.

Ng Weiu Seann – Developed the view classes and debugged the program to ensure it passed the required test cases. Contributed to the main report.

Benjamin Kam Jia Zhiang – Developed the controller classes and debugged the program to ensure it passed the required test cases. Contributed to the main report.

Liew Jia Wei – Developed the model classes, created the UML diagram, completed Javadoc for the program files. Contributed to the main report.

Balaji Abarnasri – Developed the database. Contributed to the main report.