SC2002 Object Oriented Design & Programming

SCSK Group 5

HMS Assignment Report

GitHub Repository: <https://github.com/firepiratex/SC2002>

1. Design Considerations and Design Approach Taken

S

O

L

I

D

2. Detailed UML Class Diagram

3. Testing

* Test cases and results:

1. Login
2. Cannot login:

|  |  |
| --- | --- |
| Invalid user | Valid ID but wrong password |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Patient | Doctor | Pharmacists | Administrator |
|  |  |  |  |

1. Users are asked to change their password when they login for the first time:

|  |  |  |  |
| --- | --- | --- | --- |
| Patient | Doctor | Pharmacists | Administrator |
|  |  |  |  |

1. After successful login, a different main page is displayed for different roles.

|  |  |  |  |
| --- | --- | --- | --- |
| Patient | Doctor | Pharmacists | Administrator |
|  |  |  |  |

1. After login, user can change password

|  |  |  |  |
| --- | --- | --- | --- |
| Patient | Doctor | Pharmacists | Administrator |
|  |  |  |  |

1. User cannot change a password to the default one (password)/old password. (Input error checking)
2. Patient

|  |  |
| --- | --- |
| View Medical Record | Update personal information |
|  |  |

1. View medical record & update personal information

|  |  |  |
| --- | --- | --- |
| Schedule | reschedule | Scheduled appointment |
|  |  |  |

1. Schedule, reschedule, view scheduled appointment
2. View available appointment, cancel, view past appointment outcome records

|  |  |  |
| --- | --- | --- |
| View available appointment | Cancel | View Past appointment records |
|  |  |  |

1. Doctor

|  |  |
| --- | --- |
| View | Update |
|  |  |

1. View & update patient medical records

|  |  |
| --- | --- |
| View | Set |
|  |  |

1. View personal schedule & set availability for appointments

|  |  |
| --- | --- |
| Accept/Decline | View |
|  |  |

1. Accept/Decline appointment requests, View upcoming appointment

|  |
| --- |
| Record |
|  |

1. Record Appointment Outcome
2. Administrator

|  |  |
| --- | --- |
| View | Manage |
|  |  |

1. View and manage hospital staff

|  |
| --- |
| View |
|  |

1. View Appointment details

|  |  |
| --- | --- |
| View | Manage |
|  |  |

1. View and manage medicine inventory

|  |
| --- |
| Approval of replenishment requests |
|  |

1. Approve Replenishment requests

4. Reflection

a. Difficulties Faced

When we were first going through the project, we realised that the requirement did not let us use any database application which restricts us in how we store our data and information. After reviewing the provided files, we realized we could leverage Excel/CSV files as a makeshift database to store our information.

As we worked to implement and optimize our code, we struggled to apply the concepts we learned in lectures, particularly around object-oriented design and the SOLID principles. The UML diagram, which was derived from our code, required multiple iterations as we sought to align our implementation with these principles.

Through this learning process, we recognised that our initial code structure was difficult to read and prone to breaking whenever we attempted to add or implement new features. This experience has highlighted the importance of the concepts that were taught to us in the lectures which ultimately helps us to become better programmers. We learned to appreciate the value of clean, scalable code and the need to adhere to design principles to create robust software.

b. Implementation of Design Principles //Can be put at the design part

S

O

L

I

D

We then had to optimize and modify our code so that we are able implement the design principles while applying OO concepts.

c. Knowledge learnt from this course

From the lectures, we had learnt about the object-oriented (OO) concepts alongside the SOLID design principles. We tried to ensure that any necessary changes to our project could be implemented smoothly with minimal impact. We also learnt to manage the implementation of a relatively large project given a short timeframe by learning the importance of divide and conquer techniques so that we are able to do the tasks efficiently. Additionally, we took the initiative to self-learn and utilize libraries and built-in functions, such as java.time and CSV read/write methods using java.io which enhances our application’s functionality and robustness.

d. Area of improvements for the course

They were project specifications and descriptions which were vague and unclear. This made us confused. We had to come up with assumptions as information given was vague so some parts of the application were up to our interpretation. At first, the given excel files were in xlsm format which was incompatible with the java.io library. We had to email the professor to ensure that we did not go against any restrictions that was preplaced for the project. The suggested sample test cases were not aligned to the features required of the application. The report requirements were also misaligned with the appendix. An example screenshot is attached in Figure XX. The figure on the left does not mention the “Testing” section to be included while the figure on the right mention about adding test cases.

 