Design Outline

This project is an OOP design using inheritance and encapsulation principles.

All classes are located under the package edu.rmit.cosc1295.carehome

Main Design Structure

Core Classes

1. CareHome: This is the main system class that manages all data.
2. Staff: Abstract parent class for all staff types.
3. Manager, Doctor, Nurse: Subclasses of Staff, each has different functions and permissions.
4. Resident: Stores resident details and prescriptions.
5. Bed: This represents a bed that can be assigned to a resident.
6. Prescription: Holds medicine, dose, and time prescribed by doctors.
7. Shift: Defines a work shift (day + time).
8. CareHomeDatabase: Handles SQLite database connection and queries.

JavaFX GUI

Each role has their own scene loaded from FXML. Controllers handle button actions and data passing between screens.

Manager -> Add Bed, Add Resident, Resident List, View Resident Details, Add staff, Staff List, Assign Shift, Modify Staff Password, View System Logs.

Nurse -> Residents, Move Resident, View Residents Details, Record Administered Medicine.

Doctor -> Residents, View Resident Details, Add Prescription, Record Administered Medicine, Update Prescription, Delete Prescription.

Main role-based menu:

1. DashboardController

Data input pages:

1. AddStaffController
2. AddResidentController
3. AddBedController
4. AssignShiftController
5. AddPrescriptionController
6. AdministerMedicineController
7. ModifyPasswordController
8. MoveResidentController
9. UpdatePrescriptionController

Data display pages:

1. ViewLogsController
2. ResidentListController
3. StaffListController
4. ViewResidentDetailsController

Design Decisions

1. Object-Oriented Hierarchy: Used parent Staff class to reduce duplication among Manager, Doctor, and Nurse.
2. Using CareHome as a single data source that all controllers share.
3. Database Integration: Added CareHomeDatabase with SQLite to persist data between sessions.
4. Error Handling: Using UnauthorizedException, NotWorkingException, BedOccupiedException to prevent invalid operation and improve clarity.
5. Role-Based Access: The DashBoardController dynamically shows buttons based on the user's role type.
6. Junit Testing: Using Junit to ensure data consistency and system reliability.

Data Persistence

1. All data is stored in care_home.db using SQLite.
2. The system supports both database and serialized file storage. A Junit test class named TestSaveData was created to verify the correctness of saveToFile() and loadFromFile() operations. And a Serializable backup file (e.g., SavedData.ser) are also used for fallback.