

# Design Document

## 1. Overview

Appointment Scheduler v4.20 is a JavaFX-based system that allows users to manage customer records, schedule appointments, and generate reports. It features time zone conversion, user authentication, localization, and logging of user activity. The application interacts with a MySQL database and includes data validation and conflict resolution for scheduling appointments.

## 2. System Architecture

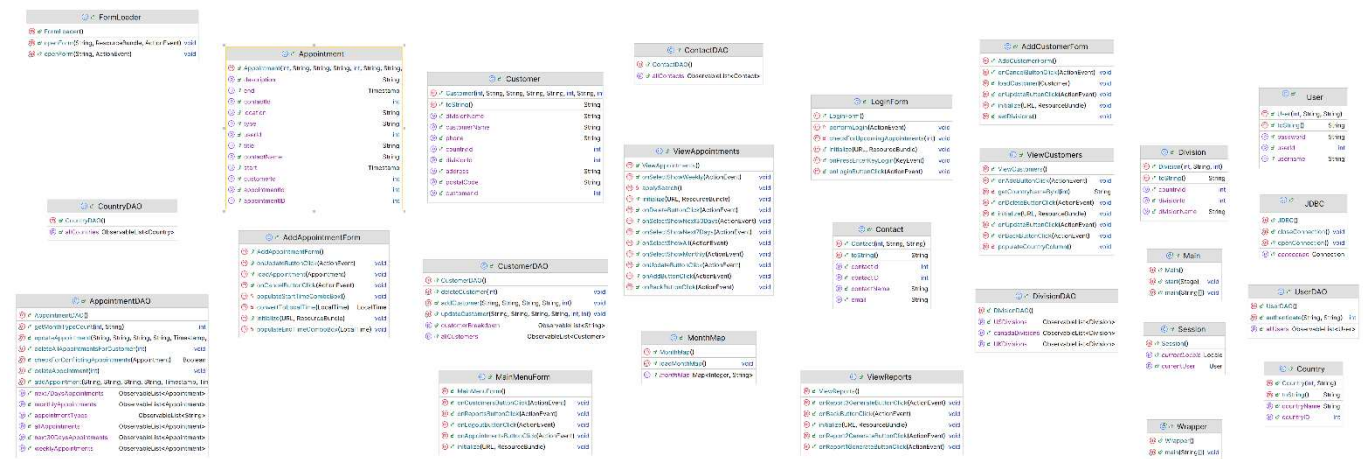
This application follows a **Model-View-Controller (MVC) architecture** with a dedicated **Data Access and Operations Layer**:

- **Model Layer:** Represents business entities such as Customer, Appointment, and User.
- **View Layer:** JavaFX FXML forms (LoginForm.fxml, AddAppointmentForm.fxml, etc.) handle user interaction.
- **Controller Layer:** Manages UI logic, form validation, and user input processing.
- **Data Access and Operations Layer:** Responsible for database interactions (CustomerDAO, AppointmentDAO, etc.), ensuring data integrity and executing SQL operations.

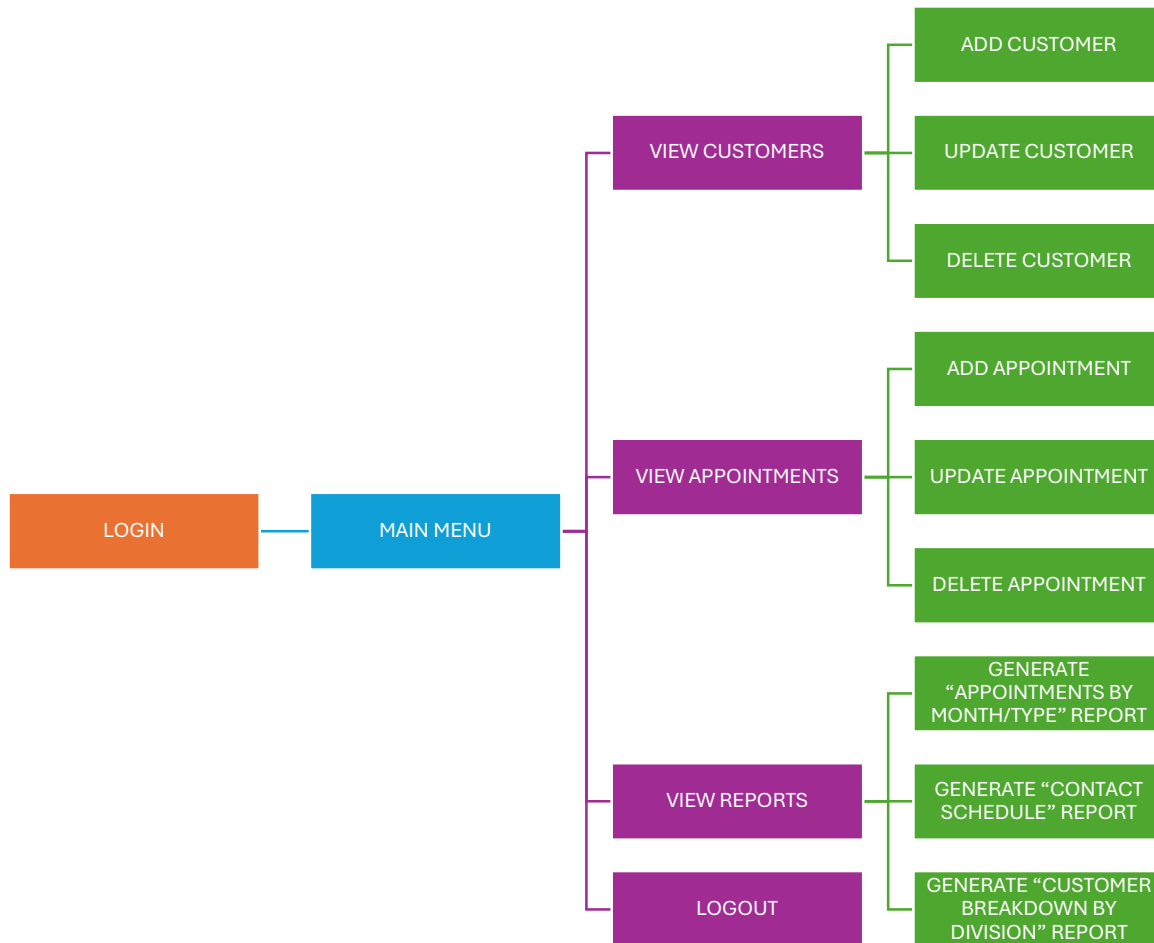
### 3. Class Diagram

The class diagram illustrates the core structure of the application, showcasing UI controllers, DAO classes, and utility components. It details each class's attributes and methods but does not depict relationships. Key elements include:

- **UI Controllers:** Manage user interactions (e.g., LoginForm, MainMenuForm, ViewCustomers).
- **DAO Classes:** Handle database operations for User, Customer, Appointment, and related entities.
- **Application Entry Points:** Main and Wrapper handle startup execution.



## 4. Design Diagram



The system begins with the Login screen, where user credentials are validated. After a successful login, users navigate to the Main Menu, which serves as the hub for accessing different functionalities.

From the Main Menu, users can:

- Navigate to the View Customers section, where they can:

- Add new customers using the Add Customer form.
- Update existing customer information through the Update Customer form.
- Delete customers via a dedicated option.

- Access the View Appointments section, enabling them to:

- Add appointments through the Add Appointment form.
- Modify details using the Update Appointment functionality.
- Remove unwanted appointments.

- Generate various reports in the View Reports section, such as:

- Appointments grouped by month and type.
- Schedules filtered by contact.
- Customer breakdowns by division.

Logout to terminate the session and return to the Login screen.

## 5. System Behavior

### User Authentication

- Users log in via LoginForm, which verifies credentials through UserDAO.authenticate().
- Login attempts (successful and failed) are logged in login\_activity.txt.
- Upon login, the system alerts users if they have an appointment within the next 15 minutes.

### Customer Management

- ViewCustomers lists all customers, allowing **add, update, and delete** operations.
- AddCustomerForm dynamically updates division selections based on the selected country.

### Appointment Scheduling

- Users can schedule, update, or cancel appointments.
- The system enforces **no overlapping appointments for the same customer**.
- Appointments are stored in **UTC** and displayed in the user's local time.

### Reports

The ViewReports controller generates three reports:

- Total number of appointments grouped by month and type.
- Schedules for a selected contact, listing all their appointments.
- Total number of customers per division.

### Security & Error Handling

- Passwords are not stored locally—authentication is handled via database verification.
- Try-catch and throws clauses blocks wrap all database queries to prevent application crashes.
- Localization is fully supported, dynamically loading UI text based on system locale.
- Data validation ensures that all fields contain valid input before submission.