Flight Booking System

System Architecture Overview

System Overview

- **Flight Booking Application** - A JavaFX desktop application
- **Purpose:** Allows users to search for flights, manage customers, and create/manage bookings
- **Key Functionality:**
  - Customer management
  - Flight search
  - Seat selection
  - Booking creation and management

Architecture: MVC Pattern

Our system follows the Model-View-Controller (MVC) pattern:

- **Model (Vinnsla Package):** Data objects and database access classes
- **View (FXML Files):** JavaFX UI layouts for each screen
- **Controller (Vidmot Package):** Classes that handle user interaction and business logic

Model Layer

Data Classes:

- Flight - Stores flight information
- Customer - Manages customer data
- Booking - Represents a booking with its status
- Seat - Represents an individual seat on a flight

Database Classes:

- FlightDB - Flight data access
- CustomerDB - Customer data access
- BookingDB - Booking data management
- DatabaseInitializer - Sets up test data

Controller Layer

Domain Logic Controllers:

- FlightController - Flight search and management
- CustomerController - Customer lookup and management
- BookingController - Core booking business logic

UI Controllers:

- FlightBookerApplication - Main application controller
- FlightSearchController - Flight search UI logic
- CreateBookingController - Booking creation UI logic
- ViewBookingController - Booking details and management

View Layer

Main Views:

- main-view.fxml - Main application screen
- flight-search-view.fxml - Flight search interface
- create-booking-view.fxml - Booking creation screen
- view-booking-view.fxml - Booking details view

UI Components:

- List views for flights and bookings
- Form controls for customer and flight data
- Seat selection components

Database Design

SQLite Database with tables:

- Customer - Customer information
- Flight - Flight details and schedules
- Seat - Seat inventory with booking status
- Booking - Reservation records

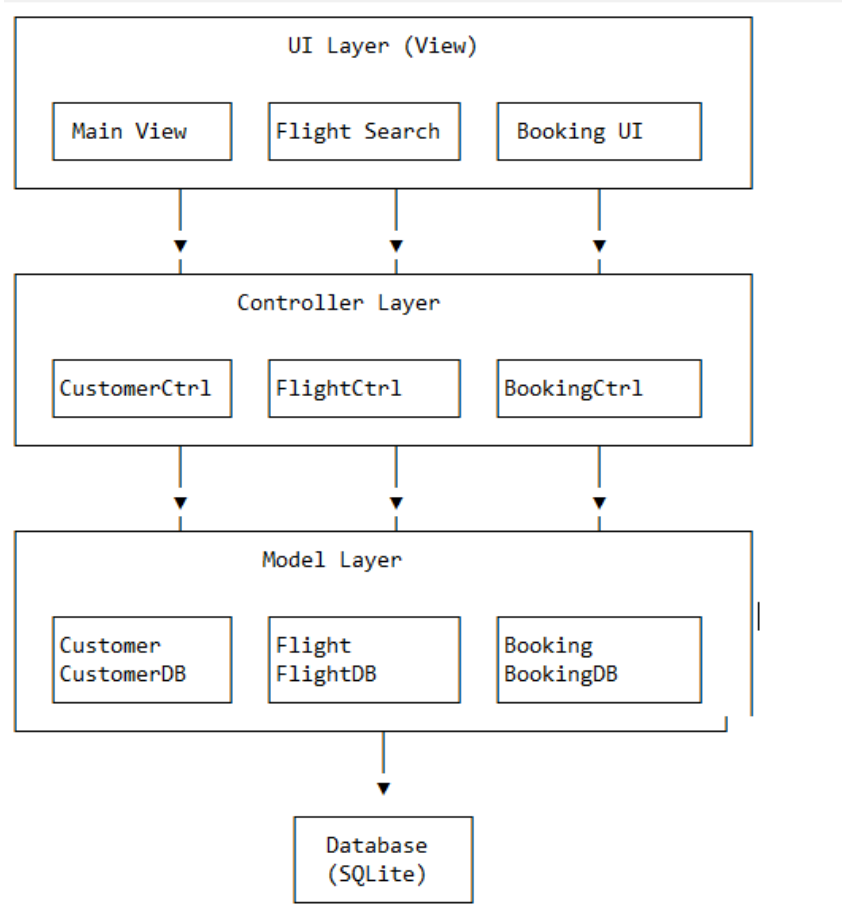
Key Features:

- Transactional operations for booking updates
- Foreign key relationships for data integrity
- Specialized queries for flight search

Data Flow: Creating a Booking

1. User enters his customer id
2. User initiates booking creation
3. User searches for and selects a flight
4. Available seats are loaded from database
5. User selects a seat
6. BookingController creates the booking
7. Seat status is updated in database
8. Booking is displayed in main view

System Architecture Diagram



Project Retrospective

Problems

- Hard to work on the project at the same time
- Some problems with integrating code between us
- Some planing issues / problems taking more time than expected

For the future

- No matter how much you plan you can‘t plan for everything
- Need to be flexible with both the planing and code
- Need to be concise on problem scope