

# Class Diagram - Alphabetical Ordered Classes

---

## Admin

«entity»

### Attributes

None (inherits from User)

### Methods

- displayDashboard(): void
  - toString(): String
- 

## AdminController

«control»

### Attributes

- flightDAO: FlightDAO
- userDao: UserDao

### Methods

- addFlight(flightNumber: String, departureCity: String, arrivalCity: String, departureTime: LocalDateTime, arrivalTime: LocalDateTime, totalSeats: int, price: double): int
  - updateFlight(flightId: int, flightNumber: String, departureCity: String, arrivalCity: String, departureTime: LocalDateTime, arrivalTime: LocalDateTime, totalSeats: int, price: double): boolean
  - deleteFlight(flightId: int): boolean
  - getAllFlights(): List<Flight>
  - getTotalUsers(): int
  - getTotalFlights(): int
-

# **AdminDashboard**

«boundary»

## **Attributes**

- admin: Admin
- adminController: AdminController
- bookingController: BookingController
- flightSearchController: FlightSearchController
- cardLayout: CardLayout
- cardPanel: JPanel

## **Methods**

- initializeUI(): void
  - createNavButton(text: String): JButton
  - createDashboardPanel(): JPanel
  - createManageFlightsPanel(): JPanel
  - createViewBookingsPanel(): JPanel
  - createReportsPanel(): JPanel
  - showAddFlightDialog(): void
  - showEditFlightDialog(flight: Flight): void
  - addFormField(dialog: JDialog, gbc: GridBagConstraints, row: int, label: String, field: JTextField): void
  - styleButton(button: JButton, bgColor: Color): void
  - logout(): void
- 

# **Agent**

«entity»

## **Attributes**

None (inherits from User)

## **Methods**

- displayDashboard(): void
  - toString(): String
-

# **AgentController**

«control»

## **Attributes**

- userDAO: UserDAO
- bookingDAO: BookingDAO

## **Methods**

- searchCustomerByEmail(email: String): User
  - getCustomerBookings(userId: int): List<Booking>
  - cancelCustomerBooking(bookingId: int): boolean
  - getAllBookings(): List<Booking>
- 

# **AgentDashboard**

«boundary»

## **Attributes**

- agent: Agent
- agentController: AgentController
- cardLayout: CardLayout
- cardPanel: JPanel

## **Methods**

- initializeUI(): void
  - createNavButton(text: String): JButton
  - createDashboardPanel(): JPanel
  - createSearchCustomerPanel(): JPanel
  - createManageCustomersPanel(): JPanel
  - createViewAllBookingsPanel(): JPanel
  - showEditCustomerDialog(customer: User): void
  - addFormField(dialog: JDialog, gbc: GridBagConstraints, row: int, label: String, field: JTextField): void
  - styleButton(button: JButton, bgColor: Color): void
  - logout(): void
-

# **AuthenticationController**

«control» (Singleton)

## **Attributes**

- instance: AuthenticationController (static)
- currentUser: User
- userDAO: UserDAO

## **Methods**

- getInstance(): AuthenticationController (static)
  - login(email: String, password: String): User
  - signup(email: String, password: String, firstName: String, lastName: String, address: String, phone: String, role: String, receivePromotions: boolean): User
  - logout(): void
  - getCurrentUser(): User
  - isLoggedIn(): boolean
  - hashPassword(password: String): String
  - isValidEmail(email: String): boolean
  - isValidPassword(password: String): boolean
- 

# **AuthFrame**

«boundary»

## **Attributes**

- cardLayout: CardLayout
- cardPanel: JPanel
- loginPanel: LoginPanel
- signupPanel: SignupPanel
- role: String

## **Methods**

- initializeUI(): void
  - showLoginPanel(): void
  - showSignupPanel(): void
-

# **Booking**

«entity»

## **Attributes**

- bookingId: int
- userId: int
- flightId: int
- bookingDate: LocalDateTime
- status: String
- seatNumber: String
- customerName: String
- customerEmail: String
- flightNumber: String
- route: String

## **Methods**

- `toString(): String`
- 

# **BookingController**

«control»

## **Attributes**

- bookingDAO: BookingDAO
- flightDAO: FlightDAO

## **Methods**

- `createBooking(userId: int, flightId: int, seatNumber: String): int`
  - `cancelBooking(bookingId: int): boolean`
  - `getUserBookings(userId: int): List<Booking>`
  - `getAllBookings(): List<Booking>`
  - `getBookingById(bookingId: int): Booking`
  - `modifyBooking(bookingId: int, newSeatNumber: String): boolean`
-

# **BookingDAO**

«boundary»

## **Attributes**

None

## **Methods**

- createBooking(userId: int, flightId: int, seatNumber: String): int
  - cancelBooking(bookingId: int): boolean
  - getUserBookings(userId: int): List<Booking>
  - getAllBookings(): List<Booking>
  - getBookingById(bookingId: int): Booking
  - extractBookingFromResultSet(rs: ResultSet): Booking
- 

# **BookingDialog**

«boundary»

## **Attributes**

- customer: Customer
- flight: Flight
- bookingController: BookingController
- paymentController: PaymentController
- seatNumberField: JTextField
- paymentMethodCombo: JComboBox<String>
- cardNumberField: JTextField
- cvvField: JTextField
- expiryField: JTextField
- nameOnCardField: JTextField
- bookingSuccessful: boolean

## **Methods**

- initializeUI(): void
- createFlightDetailsPanel(): JPanel
- createSeatSelectionPanel(): JPanel
- createPaymentPanel(): JPanel

- confirmBooking(): void
  - isBookingSuccessful(): boolean
  - createBoldLabel(text: String): JLabel
  - createLabel(text: String): JLabel
  - styleTextField(field: JTextField): void
- 

## BookingPanel

«boundary»

### Attributes

- customer: Customer
- bookingController: BookingController
- paymentController: PaymentController
- bookingsTable: JTable
- tableModel: DefaultTableModel
- viewDetailsButton: JButton
- cancelBookingButton: JButton
- refreshButton: JButton

### Methods

- initializeUI(): void
  - loadBookings(): void
  - viewBookingDetails(): void
  - createBookingDetailsPanel(booking: Booking, payment: Payment): JPanel
  - createBoldLabel(text: String): JLabel
  - createLabel(text: String): JLabel
  - cancelBooking(): void
- 

## Customer

«entity»

### Attributes

None (inherits from User)

## Methods

- `displayDashboard(): void`
  - `receivePromotion(promotionMessage: String): void`
  - `toString(): String`
- 

# CustomerDashboard

«boundary»

## Attributes

- `customer: Customer`
- `cardLayout: CardLayout`
- `cardPanel: JPanel`

## Methods

- `initializeUI(): void`
  - `createNavButton(text: String): JButton`
  - `logout(): void`
- 

# DatabaseConnection

«boundary»

## Attributes

- `instance: DatabaseConnection (static)`
- `URL: String (static final)`
- `connection: Connection`

## Methods

- `getInstance(): DatabaseConnection (static)`
  - `getConnection(): Connection`
  - `initializeDatabase(): void`
  - `close(): void`
-

# Flight

«entity»

## Attributes

- flightId: int
- flightNumber: String
- origin: String
- destination: String
- departureTime: LocalDateTime
- arrivalTime: LocalDateTime
- airline: String
- availableSeats: int
- price: double
- aircraftType: String
- totalSeats: int

## Methods

- hasAvailableSeats(): boolean
  - getDepartureCity(): String
  - getArrivalCity(): String
  - toString(): String
- 

# FlightDAO

«boundary»

## Attributes

None

## Methods

- initializeSampleData(): void
- insertSampleFlights(): void
- addFlight(flightNumber: String, origin: String, destination: String, departureTime: LocalDateTime, arrivalTime: LocalDateTime, availableSeats: int, price: double): int
- updateFlight(flightId: int, flightNumber: String, origin: String, destination: String, departureTime: LocalDateTime, arrivalTime: LocalDateTime, availableSeats: int, price: double): boolean

- deleteFlight(flightId: int): boolean
  - searchFlights(origin: String, destination: String, date: String, airline: String): List<Flight>
  - getFlightById(flightId: int): Flight
  - getAllFlights(): List<Flight>
  - getAllOrigins(): List<String>
  - getAllDestinations(): List<String>
  - getAllAirlines(): List<String>
  - updateAvailableSeats(flightId: int, newSeatCount: int): boolean
  - decrementAvailableSeats(flightId: int): boolean
  - incrementAvailableSeats(flightId: int): boolean
  - getTotalFlightCount(): int
  - extractFlightFromResultSet(rs: ResultSet): Flight
- 

## FlightSearchController

«control»

### Attributes

- flightDAO: FlightDAO

### Methods

- searchFlights(origin: String, destination: String, date: LocalDate): List<Flight>
  - searchFlights(origin: String, destination: String, date: String, airline: String): List<Flight>
  - getAllFlights(): List<Flight>
  - getFlightById(flightId: int): Flight
  - getAllOrigins(): List<String>
  - getAllDestinations(): List<String>
  - getAllAirlines(): List<String>
  - getAllCities(): List<String>
- 

## FlightSearchPanel

«boundary»

### Attributes

- searchController: FlightSearchController

- customer: Customer
- departureCityCombo: JComboBox<String>
- arrivalCityCombo: JComboBox<String>
- airlineCombo: JComboBox<String>
- dateField: JTextField
- flightTable: JTable
- tableModel: DefaultTableModel
- searchButton: JButton
- bookButton: JButton
- clearButton: JButton

## Methods

- initializeUI(): void
  - loadFilters(): void
  - clearFilters(): void
  - loadAllFlights(): void
  - searchFlights(): void
  - displayFlights(flights: List<Flight>): void
  - bookFlight(): void
- 

## Guest

«entity»

### Attributes

None (inherits from User)

### Methods

- displayDashboard(): void
  - canBook(): boolean
  - isGuest(): boolean
  - toString(): String
- 

## GuestDashboard

«boundary»

## **Attributes**

- guest: Guest

## **Methods**

- initializeUI(): void
- 

# **HomePage**

«boundary»

## **Attributes**

None

## **Methods**

- initializeUI(): void
  - createRoleButton(role: String, description: String): JButton
  - openGuestMode(): void
  - openAuthFrame(role: String): void
- 

# **LoginPanel**

«boundary»

## **Attributes**

- emailField: JTextField
- passwordField: JPasswordField
- loginButton: JButton
- switchToSignupButton: JButton
- parentFrame: AuthFrame
- preSelectedRole: String

## **Methods**

- initializeUI(): void

- handleLogin(): void
  - clearFields(): void
- 

## MainFrame

«boundary»

### Attributes

None

### Methods

- main(args: String[]): void (static)
- 

## Payment

«entity»

### Attributes

- paymentId: int
- bookingId: int
- amount: double
- paymentMethod: String
- cardNumber: String
- paymentDate: LocalDateTime

### Methods

- getMaskedCardNumber(): String
  - toString(): String
- 

## PaymentController

«control»

## Attributes

- paymentDAO: PaymentDAO

## Methods

- processPayment(bookingId: int, amount: double, paymentMethod: String, cardNumber: String): int
  - getPaymentByBookingId(bookingId: int): Payment
  - maskCardNumber(cardNumber: String): String
  - validateCardNumber(cardNumber: String): boolean
  - validateCVV(cvv: String): boolean
- 

# PaymentDAO

«boundary»

## Attributes

None

## Methods

- createPayment(bookingId: int, amount: double, paymentMethod: String, cardNumber: String): int
  - getPaymentByBookingId(bookingId: int): Payment
- 

# ProfilePanel

«boundary»

## Attributes

- customer: Customer
- userDAO: UserDAO
- promotionManager: PromotionManager
- emailField: JTextField
- firstNameField: JTextField
- lastNameField: JTextField
- addressField: JTextField

- phoneField: JTextField
- promotionsCheckbox: JCheckBox
- promotionDisplayArea: JTextArea
- promotionPanel: JPanel

## Methods

- initializeUI(): void
  - createPromotionDisplayPanel(): JPanel
  - togglePromotionDisplay(): void
  - createLabel(text: String): JLabel
  - styleTextField(field: JTextField): void
  - updateProfile(): void
  - refreshPanel(): void
- 

## PromotionManager

«control» (Singleton + Observer)

### Attributes

- instance: PromotionManager (static)
- subscribers: List<PromotionObserver>

### Methods

- getInstance(): PromotionManager (static)
  - subscribe(observer: PromotionObserver): void
  - unsubscribe(observer: PromotionObserver): void
  - notifyAllSubscribers(promotionMessage: String): void
  - sendMonthlyPromotion(): void
  - getSubscriberCount(): int
  - isSubscribed(userId: int): boolean
- 

## PromotionObserver

«interface»

### Methods

- receivePromotion(promotionMessage: String): void
- 

## SignupPanel

«boundary»

### Attributes

- emailField: JTextField
- firstNameField: JTextField
- lastNameField: JTextField
- addressField: JTextField
- phoneField: JTextField
- passwordField: JPasswordField
- confirmPasswordField: JPasswordField
- receivePromotionsCheckBox: JCheckBox
- parentFrame: AuthFrame
- preSelectedRole: String

### Methods

- initializeUI(): void
  - addLabelAndField(labelText: String, field: JTextField, gbc: GridBagConstraints): void
  - handleSignup(): void
  - clearFields(): void
- 

## User

«entity» (Abstract)

### Attributes

- userId: int
- email: String
- password: String
- firstName: String
- lastName: String
- address: String
- phone: String

- role: String
- receivePromotions: boolean

## Methods

- displayDashboard(): void (abstract)
  - isGuest(): boolean
  - canBook(): boolean
  - getFullName(): String
  - toString(): String
- 

# UserDAO

«boundary»

## Attributes

None

## Methods

- authenticateUser(email: String, hashedPassword: String): User
  - emailExists(email: String): boolean
  - insertUser(email: String, hashedPassword: String, firstName: String, lastName: String, address: String, phone: String, role: String, receivePromotions: boolean): int
  - getUserId(userId: int): User
  - getUserByEmail(email: String): User
  - updatePromotionPreference(userId: int, receivePromotions: boolean): boolean
  - updateUserProfile(userId: int, firstName: String, lastName: String, address: String, phone: String): boolean
  - getTotalUserCount(): int
  - extractUserFromResultSet(rs: ResultSet): User
- 

# UserFactory

«control» (Factory Pattern)

## Attributes

None

## Methods

- `createUser(userId: int, email: String, password: String, firstName: String, lastName: String, address: String, phone: String, role: String, receivePromotions: boolean): User (static)`
  - `createUser(userId: int, email: String, password: String, firstName: String, lastName: String, address: String, phone: String, role: String, receivePromotionsInt: int): User (static)`
-