

## USE CASE SCENARIOS

### UC0 – Pick Role

1. The User opens the FlightReservationApplication.
2. The System displays the PickRoleScreen with options for Customer, Agent, and SystemAdmin.
3. The User selects a desired Role.
4. The System navigates to LoginView.

### UC1 – Login

1. The Actor (Customer, Agent, or SystemAdmin) launches the FlightReservationApplication.
2. The System displays the LoginView with UsernameField and PasswordField.
3. The Actor enters username and password into the fields.
4. The Actor clicks the LoginButton.
5. The System sends the credentials to the AuthenticationService.
6. The AuthenticationService validates the data against the Database and retrieves the matching UserAccount.

### UC2 – Logout

1. The logged-in Actor clicks the LogoutAction from the NavigationMenu.
2. The System clears cached UserData from the client view.
3. The System displays the LoginView or PickRoleScreen again.

### UC3 – Register Account (Customer)

1. The Customer selects the RegisterAccountOption from the LoginView.
2. The System displays the RegistrationForm with fields for PersonalInfo, ContactInfo, and LoginCredentials.
3. The Customer fills in required information and submits the form.
4. The System validates the input and checks the Database for existing UserAccount with the same email.
5. If validation passes, the System creates a new CustomerAccount record in the Database.
6. The System confirms successful registration and redirects the Customer to the LoginView.

## UC4 – Make Reservation (Customer)

1. The logged-in Customer **selects** the MakeReservationOption from the CustomerDashboard.
2. The System **invokes** the included use case SearchFlights.
3. The Customer **views** the list of AvailableFlights and **selects** a Flight.
4. The System **invokes** the included use case PickSeats for that flight.
5. After seats are chosen, the System **displays** a ReservationSummary showing FlightDetails, SelectedSeats, and TotalPrice.
6. The Customer optionally **invokes** the extending use case ApplyPromoCode.
7. The Customer **confirms** the reservation and the System **invokes** the included use case ProcessPayment.
8. On successful payment, the System **creates** a new Reservation and **invokes** GenerateBookingConfirmation.
9. The System **displays** the BookingConfirmationScreen to the Customer.

## UC5 – View Flight Details (Customer)

1. The Customer **selects** a Flight from a list.
2. The System **retrieves** the FlightDetails from the Database including DepartureTime, ArrivalTime, Route, Aircraft, and SeatAvailability.
3. The System **displays** the details on a FlightDetailsView.
4. The Customer **may** navigate from this screen to the MakeReservation flow.

## UC6 – Update Profile (Customer)

1. The Customer **opens** the ProfileSettingsView from the CustomerDashboard.
2. The System **loads** current ProfileData from the Database.
3. The Customer edits fields such as Name, ContactInfo, and Preferences.
4. The Customer **saves** changes.
5. The System **validates** the new data and **updates** the CustomerAccount record.
6. The System **shows** a success message and **refreshes** ProfileView.

## UC7 – View Profile (Customer)

1. The Customer **selects** the ViewProfileOption from the dashboard.
2. The System **fetches** ProfileData from the Database.
3. The System **displays** the ProfileView including personal and contact details.

## **UC8 – View Reservation (Customer)**

1. The Customer selects the ViewReservationOption.
2. The System queries the Database for all Reservations linked to the CustomerAccount.
3. The System displays a ReservationListView.
4. The Customer selects a specific Reservation to view detailed BookingInformation.

## **UC9 – Manage Customer Data (Agent)**

1. The logged-in Agent opens the ManageCustomerDataView.
2. The Agent view for a Customer list.
3. The System retrieves matching CustomerRecords from the Database.
4. The Agent selects a Customer and views associated ProfileData and ReservationHistory.
5. The Agent can update contact information, add notes, or correct data, then save changes.
6. The System persists updates in the Database.

## **UC10 – Modify Reservation (Agent)**

1. The Agent opens the ModifyReservationView.
2. The Agent searches for a Reservation by ReservationID, CustomerName, or FlightNumber.
3. The System invokes the included use case ViewCustomerReservations to show all bookings for the selected Customer.
4. The Agent selects a Reservation to modify.
5. The System displays current ReservationDetails.
6. The Agent may change Seats, adjust Dates, or update PassengerInfo based on business rules.
7. If the **change** affects price, the System recalculates Fare and **may** invoke ProcessPayment for additional charges or IssueRefund for credits.
8. The System updates the Reservation in the Database and shows an updated Confirmation.

## **UC11 – View Customer Reservations (Included)**

1. The Agent or Customer requests to view reservations for a specific CustomerAccount.
2. The System queries the Database for all related Reservations.
3. The System displays the ReservationListView with key fields like Status, Flight, and TravelDate.

## **UC12 – Cancel Reservation**

1. The Customer or Agent selects a Reservation from the ReservationListView.
2. The Actor chooses the CancelReservationAction.
3. The System shows a CancellationConfirmationDialog with any RefundPolicy.
4. The Actor confirms the cancellation.
5. The System updates the ReservationStatus to "Cancelled" and **free**s associated Seats.
6. The System invokes the included use case IssueRefund if applicable.
7. The System shows a final CancellationSummary.

## **UC13 – Issue Refund (Included)**

1. The System receives a request to create a Refund for a cancelled Reservation or failed Payment.
2. The System calculates the RefundAmount based on FareRules and Fees.
3. The System invokes the PaymentSystem to process the refund to the original PaymentMethod.
4. On success, the System creates a RefundRecord in the Database and **links** it to the Payment and Reservation.
5. The System notifies the Customer via EmailSystem with a RefundConfirmationEmail.

## **UC14 – View Reports (System Admin)**

1. The SystemAdmin selects the ViewReportsOption from the admin dashboard.
2. The System displays a ReportsMenu with options such as SalesReport, LoadFactorReport, and CustomerActivityReport.
3. The SystemAdmin chooses a report type and **sets** filters like DateRange or Route.
4. The System queries the Database and **generates** the requested Report.
5. The System displays the report on a ReportView and optionally lets the SystemAdmin export it as PDF or CSV.

## **UC15 – Manage Airports (System Admin)**

1. The SystemAdmin opens the ManageAirportsView.
2. The System lists existing Airports from the Database.
3. The SystemAdmin may add a new Airport, **edit** an existing Airport, or **deactivate** an Airport.

4. When saving, the **System** **validates** codes, names, and city fields and **updates** the **AirportTable**.

## UC16 – Manage Airlines (System Admin)

1. The **SystemAdmin** **opens** the **ManageAirlinesView**.
2. The **System** **displays** all **Airlines**.
3. The **SystemAdmin** **can create, edit, or remove** **Airline** entries (subject to rules).
4. The **System** **persists** changes in the **Database** and **updates** any dependent **Flights** or **Routes** as needed.

## UC17 – Manage Routes (System Admin)

1. The **SystemAdmin** **opens** the **ManageRoutesView**.
2. The **System** **loads** existing **Routes** connecting **Airports**.
3. The **SystemAdmin** **can define** new **Route** pairs or **edit** existing ones.
4. The **System** **validates** that **OriginAirport** and **DestinationAirport** are different and active.
5. The **System** **saves** all route changes to the **Database**.

## UC18 – Manage Aircraft (System Admin)

1. The **SystemAdmin** **opens** the **ManageAircraftView**.
2. The **System** **displays** a list of **Aircraft** with **Model**, **Capacity**, and **SeatLayout**.
3. The **SystemAdmin** **may add** a new **Aircraft**, **update** details, or **retire** an **Aircraft**.
4. The **System** **validates** capacity values and **updates** the **AircraftTable**.

## UC19 – Manage Flight (System Admin)

1. The **SystemAdmin** **opens** the **ManageFlightView**.
2. The **System** **lists** all scheduled **Flights** with **Route**, **Aircraft**, and **Schedule**.
3. The **SystemAdmin** **can create** a new **Flight** by selecting a **Route** and **Aircraft** and **setting** departure and arrival times.
4. The **SystemAdmin** **may** also **edit** times or statuses of existing **Flights**.
5. The **System** **saves** changes and **updates** related **SeatInventory** and **Availability** in the **Database**.

## **UC20 – Send Monthly Promos (System Admin)**

1. The SystemAdmin **opens** the SendMonthlyPromosView.
2. The System **retrieves** a target list of Customers based on MarketingCriteria (e.g., routes, inactivity).
3. The SystemAdmin **composes** a PromoEmailTemplate and associated PromoCode.
4. The System **invokes** the EmailSystem to **send** personalized PromoEmails to all selected Customers.
5. The System **logs** the campaign in the Database for reporting.

## **UC21 – Search Flights (Included)**

1. The Customer or Agent **opens** the FlightSearchForm.
2. The Actor **enters** OriginAirport, DestinationAirport, TravelDate, and optional filters.
3. The Actor **submits** the search.
4. The System **queries** the Database for Flights matching criteria.
5. The System **displays** a FlightSearchResultsView with available options.

## **UC22 – Pick Seat(s) (Included)**

1. After choosing a Flight, the System **displays** a graphical SeatMap showing AvailableSeats and BookedSeats.
2. The Customer or Agent **selects** one or more Seats based on traveler count and preferences.
3. The System **temporarily reserves** the selected Seats and **updates** the ReservationDraft.
4. The Actor **confirms** seat selection and the System **returns** to the main reservation flow.

## **UC23 – Apply Promo Code (Extends Make Reservation)**

1. On the ReservationSummaryView, the Customer **enters** a PromoCode into the promo field.
2. The System **validates** the code against the PromoDatabase for eligibility and expiry.
3. If valid, the System **recalculates** the TotalPrice and **updates** the ReservationSummary with discounted value.
4. If invalid, the System **displays** an error message and keeps the original price.

## **UC24 – Process Payment (Included)**

1. The Customer or Agent **opens** the PaymentForm for a ReservationDraft or modification.

2. The Actor **enters** PaymentDetails such as CardNumber, ExpiryDate, CVV, or **selects** a stored PaymentMethod.
3. The System **validates** the format of the details and **calculates** the final AmountDue.
4. The System **sends** a payment request to the external PaymentSystem.
5. The PaymentSystem **authorizes** or **declines** the transaction and **returns** a response.
6. On success, the System **creates** a PaymentRecord and **links** it to the Reservation.
7. The System then **invokes** GenerateBookingConfirmation.
8. On failure, the System **displays** an error and **allows** the Actor to **retry** or **cancel**.

## UC25 – Generate Booking Confirmation (Included)

1. After successful Payment, the System **creates** a finalized Reservation with status "Confirmed".
2. The System **generates** a BookingReference and ItineraryDocument.
3. The System **displays** the BookingConfirmationScreen to the Customer or Agent.
4. The System **invokes** the EmailSystem to **send** a BookingConfirmationEmail including the itinerary and booking reference.