

## Lab 1: Requirements Elicitation & Modeling Bus Ticket Management System

### Documentation & Submission

This section summarizes the key deliverables for **Lab 1: Requirements Elicitation & Modeling**, including the documented Architecturally Significant Requirements (ASRs) and the completed UML Use Case Diagram for the Bus Ticket Management System.

#### 1. Archi Architecturally Significant Requirements (ASR Documentation)

The following table presents the three selected ASRs that exert the strongest influence on the architectural design of the SmartBus system. Each ASR is accompanied by a short justification explaining its architectural impact.

**ASR Table**

ASR	Type	Statement	Architectural Impact
<b>Real-Time Seat Inventory Management</b>	Functional / Quality Attribute	The system must update seat availability instantly across all user sessions to prevent double-booking during peak traffic.	Requires concurrency control (optimistic/pessimistic locking), efficient database transactions, and stateless backend services. Impacts database schema design and communication patterns.
<b>High Scalability for Peak Booking Events</b>	Non-Functional (Scalability)	The platform must support thousands of concurrent users, especially during holidays and special events.	Drives adoption of stateless RESTful API, horizontal scaling, load balancing, connection pooling, and distributed deployment architecture.
<b>Secure Payment Processing Integration</b>	Functional / Security	The system must securely integrate with banking/payment gateways for QR payments and maintain financial transaction safety.	Requires secure API design, encryption, strict validation, error-handling patterns, compliance with financial security standards, and separation of payment logic into isolated components.

## 2. Functional Requirements (FRs)

- **FR1:** Customer can search bus trip by date, route, and time.
- **FR2:** Customer can select seats and add them to a booking.
- **FR3:** Customer can complete bookings with online payment (Banking/QR).
- **FR4:** Customer can view booking history and invoices.
- **FR5:** Customer can manage personal profile (name, phone number, email).

## 3. Non-Functional requirements (NFRs)

- **Performance:** The system must respond to ticket search requests within  $\leq 2$  seconds under 1000 concurrent users.
- **Security:** All system communication must use TLS 1.2+, and passwords stored with bcrypt hashing.
- **Availability:** The system must maintain at least 99.9% uptime to support continuous nationwide ticket booking.

## 4. UML Use Case Diagram Submission

The final **UML Use Case Diagram** for the SmartBus platform visually represents the interactions between system actors and core system functionalities. It clearly distinguishes between:

### Actors

- **Administrator** – Manages companies, vehicles, schedules, routes, stations, tickets, accounts, reports.
- **User** – Books tickets, manages personal info, views history, explores vehicle/company data.

### Use Case Groups

#### Administrator Use Cases:

Manage Company, Manage Vehicle, Manage Route, Manage Station, Manage Seat

Configuration, Manage Schedule, Manage Ticket, Manage Account, View Revenue Report, View Trip Statistics.

Most of these include the Login use case through *«include»*.

#### Customer Use case:

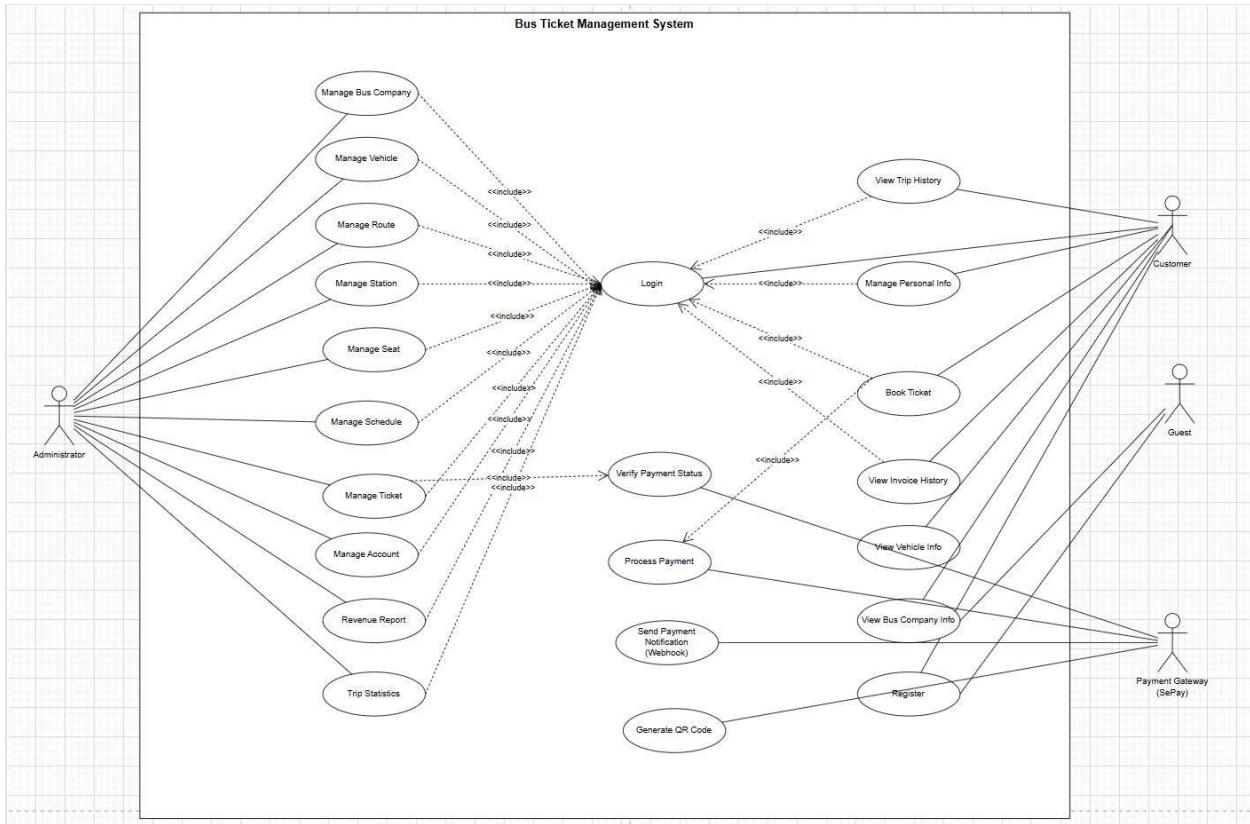
Register, Login, Book Ticket, Manage Personal Info, View Invoice History, View Trip History, View Vehicle Info, View Company Info.

Booking and history-related use cases include **Login** via *«include»*.

### Use Case Relationships

- *«include» Login* for all protected actions (admin + user)

- Optional actions such as coupon application or modifications can be represented with «extend» if included in your diagram.



## 5. Use Case Specification

### 5.1 Login

Use Case Name	Login
Primary Actor	User/Administrator
Description	Authenticate a user so they can access system functionalities based on their assigned role.
Preconditions	User account exists in the system.
Postconditions	User is authenticated and redirected to their dashboard.
Main Flow	<ol style="list-style-type: none"> <li>1. Actor enters username and password.</li> <li>2. System validates input format.</li> <li>3. System checks credentials in the database.</li> <li>4. System logs the user in and displays dashboard.</li> </ol>
Alternative Flow	Forgot Password <ul style="list-style-type: none"> <li>- Actor selects “Forgot Password”.</li> <li>- System sends reset link.</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>- Invalid credentials → System displays error message.</li> <li>- Account locked → System denies access.</li> </ul>
Priority	High.
Business Rules	<ul style="list-style-type: none"> <li>- Password must meet security constraints.</li> <li>- Max 5 failed login attempts before lockout.</li> </ul>

### 5.2 Register

Use Case Name	Register
Primary Actor	User
Description	Allow customers to create an account to book bus tickets.
Preconditions	User does not already have an account.
Postconditions	New user account is created and stored.
Main Flow	<ol style="list-style-type: none"> <li>1. User enters personal info and password.</li> <li>2. System validates data.</li> <li>3. System creates a new account.</li> <li>4. System confirms registration.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>- Email already exists → System suggests login instead.</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>- Invalid phone number, email, or password format.</li> </ul>
Priority	High.
Business Rules	<ul style="list-style-type: none"> <li>- Phone number must be unique.</li> <li>- Password must be at least 8 characters.</li> </ul>

### 5.3 Book Ticket

Use Case Name	Book Ticket
Primary Actor	User
Description	User selects route, schedule, and seat to purchase a ticket.
Preconditions	<ul style="list-style-type: none"> <li>- User is logged in.</li> <li>- Available schedules exist.</li> <li>- Seats must be available.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Ticket is generated.</li> <li>- Seat inventory is updated.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. User enters origin, destination, and date.</li> <li>2. System shows available schedules.</li> <li>3. System selects schedule.</li> <li>4. System displays available seats.</li> <li>5. User selects seats.</li> <li>6. User confirms booking.</li> <li>7. System creates ticket and marks seat as reserved.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>- User Changes Seat → User selects a different seat before .</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>- Seat becomes occupied during selection → System prompts user to choose another seat.</li> <li>- Payment fails → Booking canceled.</li> </ul>
Priority	Critical.
Business Rules	<ul style="list-style-type: none"> <li>- Seat allocation must be real-time (ASR).</li> <li>- Ticket must be confirmed only after successful payment.</li> </ul>

### 5.4 Manage Personal Info

Use Case Name	Manage Personal Info
Primary Actor	User
Description	User updates profile information.
Preconditions	User is logged in.
Postconditions	Updated information is saved.
Main Flow	<ol style="list-style-type: none"> <li>1. User opens Profile page.</li> <li>2. User updates name, phone, or email.</li> <li>3. System validates input.</li> <li>4. System saves changes.</li> </ol>
Alternative Flow	None.
Exceptions	Invalid email or phone format.
Priority	Medium.
Business Rules	Email must be unique system-wide.

## 5.5 View Trip History

Use Case Name	View Trip History
Primary Actor	User
Description	Display all past or upcoming trips booked by the user.
Preconditions	User is logged in.
Postconditions	Trip history is displayed.
Main Flow	<ol style="list-style-type: none"><li>1. User selects “Trip History”.</li><li>2. System retrieves user’s trip list.</li><li>3. System shows trip data.</li></ol>
Alternative Flow	None.
Exceptions	Database retrieval failure.
Priority	Medium.
Business Rules	User can only view their own trip history.

## 5.6 View Invoice History

Use Case Name	View Invoice History
Primary Actor	User
Description	User reviews past payments and invoices.
Preconditions	User is logged in.
Postconditions	Invoice list displayed.
Main Flow	<ol style="list-style-type: none"><li>1. User selects “Invoice History”.</li><li>2. System retrieves payment data.</li><li>3. System displays invoice details.</li></ol>
Alternative Flow	None.
Exceptions	Missing invoice file → System shows error.
Priority	Medium.
Business Rules	Invoice cannot be modified.

## 5.7 View Vehicle Info/View Company Info

<b>Use Case Name</b>	<b>View Vehicle Info/View Company Info</b>
Primary Actor	User
Description	User views details of vehicles or transport companies.
Preconditions	<ul style="list-style-type: none"> <li>- Vehicle/company records exist in the system.</li> <li>- For some sensitive fields, user may need to be logged in (if business policy restricts details to authenticated users).</li> </ul>
Postconditions	Information displayed.
Main Flow	<ol style="list-style-type: none"> <li>1. User selects company or vehicle info.</li> <li>2. System retrieves and displays data.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Partial data available           <ul style="list-style-type: none"> <li>- Requested record exists but some non-critical fields are missing.</li> <li>- System displays available fields and marks missing fields as “Not available.”</li> </ul> </li> <li>2. User requests more details (requires authentication)           <ul style="list-style-type: none"> <li>- User selects “View detailed info.”</li> <li>- If not logged in → System prompts for login (fall back to Login use case).</li> <li>- After successful login → System displays extended info.</li> </ul> </li> </ol>
Exceptions	Data unavailable.
Priority	Low.
Business Rules	Only published information is shown.

## 5.8 Manage Company

<b>Use Case Name</b>	<b>Manage Company</b>
Primary Actor	User
Description	Admin manages transport companies.
Preconditions	Admin is logged in.
Postconditions	Company list updated.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin views company list.</li> <li>2. Admin adds/edits/deletes a company.</li> <li>3. System saves changes.</li> </ol>
Alternative Flow	Editing a company with invalid data.
Exceptions	Duplicate company name.
Priority	High.
Business Rules	Company name must be unique.

## 5.9 Manage Vehicles

Use Case Name	Manage Vehicles
Primary Actor	Administrator
Description	Admin manages vehicle information.
Preconditions	<ul style="list-style-type: none"> <li>- Admin is logged in.</li> <li>- Company must exist.</li> </ul>
Postconditions	Vehicle list updated.
Main Flow	<ol style="list-style-type: none"> <li>1. Uploading vehicle with incomplete data.             <ul style="list-style-type: none"> <li>- Admin submits vehicle creation form with missing required fields.</li> <li>- System highlights missing fields and returns to the form for completion.</li> </ul> </li> <li>2. Admin edits vehicle assigned to schedules.             <ul style="list-style-type: none"> <li>- Admin attempts to change vehicle capacity/type that affects existing schedules.</li> <li>- System warns about impacted schedules and requires explicit confirmation.</li> <li>- Admin confirms → System applies change and updates related schedule metadata.</li> </ul> </li> </ol>
Alternative Flow	Editing a company with invalid data.
Exceptions	Invalid license plate format.
Priority	High.
Business Rules	License plate must be unique.

## 5.10 Manage Routes

Use Case Name	Manage Routes
Primary Actor	Administrator
Description	Admin manages route configurations.
Preconditions	<ul style="list-style-type: none"> <li>- Admin is authenticated and authorized to manage routes.</li> <li>- Source and destination stations exist in the system.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- New route is persisted, or existing route is updated/deleted.</li> <li>- Any dependent schedules or trips are validated against route changes.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. Admin opens Route Management.</li> <li>2. Admin selects “Create Route.”</li> <li>3. Admin inputs route name, start station, end station, intermediate stops, distance, estimated duration.</li> <li>4. System validates inputs (no duplicate route name, start ≠ end, stations exist).</li> <li>5. System saves the new route and returns confirmation.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Admin edits route <ul style="list-style-type: none"> <li>- Admin selects an existing route and updates fields.</li> <li>- System validates updates; if valid → saves changes and notifies affected schedules.</li> </ul> </li> <li>2. Admin deletes route <ul style="list-style-type: none"> <li>- Admin requests deletion.</li> <li>- System checks for dependent schedules or active trips.</li> <li>- If no dependencies → deletes route.</li> <li>- If dependencies exist → system prompts to archive route or reject deletion.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Start station equals end station → System rejects creation with error message.</li> <li>- E2: Duplicate route name detected → System rejects with an error.</li> <li>- E3: Database write failure → System displays an error and rolls back transaction.</li> </ul>
Priority	High.
Business Rules	Route must be unique.

## 5.11 Manage Bus Stations

Use Case Name	Manage Bus Stations
Primary Actor	Administrator
Description	Admin manages pick-up/drop-off stations.
Preconditions	<ul style="list-style-type: none"> <li>- Admin is authenticated and authorized.</li> <li>- For updates/deletion: station currently exists in system.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Station record is created/updated/deleted (or marked inactive/archived).</li> <li>- Related route/schedule references are updated/validated if applicable.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. Admin navigates to Station Management.</li> <li>2. Admin selects Add/Edit/Delete station.</li> <li>3. Admin enters station name, address, coordinates, contact info.</li> <li>4. System validates uniqueness of station address and required fields.</li> <li>5. System saves changes and confirms operation.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Add station with duplicate address <ul style="list-style-type: none"> <li>- System detects existing station at same address → prompts Admin to either link to the existing station or abort.</li> </ul> </li> <li>2. Attempt to delete station linked to active route <ul style="list-style-type: none"> <li>- System prevents deletion and offers option to mark station as inactive or to reassign routes</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Station linked to active routes/trips → deletion is blocked.</li> <li>- Geolocation/coordinate validation fails → station not saved.</li> <li>- Network/db failure during save → inform admin and retry option.</li> </ul>
Priority	Medium.
Business Rules	Station address cannot duplicate.

## 5.12 Manage Seats

Use Case Name	Manage Seats
Primary Actor	Administrator
Description	Configure seat map for each bus type.
Preconditions	<ul style="list-style-type: none"> <li>- Admin authenticated and authorized.</li> <li>- Vehicle record exists to which seat layout will be assigned.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Seat layout/configuration is saved and associated with the vehicle type.</li> <li>- Seat availability metadata initialized for future schedules.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. Admin opens seat configuration for a specific vehicle type.</li> <li>2. Admin designs/selects seat map (numbering, type—regular/premium, aisle/window).</li> <li>3. System validates seat count against vehicle capacity and business rules.</li> <li>4. System saves configuration and updates vehicle record.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Upload seat map file <ul style="list-style-type: none"> <li>- Admin uploads a seat-layout file (CSV/JSON).</li> <li>- System parses file and validates mapping; displays a preview for confirmation.</li> <li>- Admin confirms → system saves configuration.</li> </ul> </li> <li>2. Change seat numbering <ul style="list-style-type: none"> <li>- Admin modifies seat numbering which affects existing bookings → system warns and requires confirmation.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Seat count does not match vehicle capacity → system rejects change.</li> <li>- Seat map file parsing error → system reports parsing error and points to required format.</li> </ul>
Priority	Medium.
Business Rules	Seat count must match vehicle type.

### 5.13 Manage Schedules

Use Case Name	Manage Schedules
Primary Actor	Administrator
Description	Admin controls departure times and schedule info.
Preconditions	<ul style="list-style-type: none"> <li>- Admin authenticated and authorized.</li> <li>- Vehicle and route exist for schedule assignment.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Schedule entry is created/updated/deleted.</li> <li>- Seat inventory for each scheduled trip is initialized or updated.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. Admin selects “Create Schedule.”</li> <li>2. Admin chooses route, vehicle, departure time, arrival time, fare, and other attributes.</li> <li>3. System checks for vehicle/time conflicts, required rest times, and regulatory constraints.</li> <li>4. System saves the schedule and initializes seat inventory for scheduled trip.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Admin duplicates a recurring schedule <ul style="list-style-type: none"> <li>- Admin selects an existing schedule and chooses “duplicate/recurrence.”</li> <li>- System offers options (daily/weekly) and creates new schedule instances after validation.</li> </ul> </li> <li>2. Admin updates schedule time <ul style="list-style-type: none"> <li>- Admin modifies departure/arrival time.</li> <li>- System checks conflicts and notifies admin about affected bookings or required passenger notifications.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Vehicle already assigned to another schedule at overlapping times → system rejects with conflict details.</li> <li>- Route not available or route mismatch → reject.</li> <li>- Failure initializing seat inventory → roll back schedule creation.</li> </ul>
Priority	High.
Business Rules	A vehicle cannot overlap schedules.

## 5.14 Manage Tickets

Use Case Name	Manage Tickets
Primary Actor	Administrator
Description	Admin views or modifies ticket information in special cases.
Preconditions	<ul style="list-style-type: none"> <li>- Admin authenticated and authorized.</li> <li>- Ticket record exists (for review/modification).</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Ticket status updated (e.g., refunded, canceled, reissued) and audit log recorded.</li> <li>- Customer and seat inventory updated according to the ticket modification.</li> </ul>
Main Flow	<ul style="list-style-type: none"> <li>- Admin opens ticket management and searches for a ticket (by ID, customer, trip).</li> <li>- Admin views ticket details.</li> <li>- Admin selects action (refund, reissue, cancel, correct passenger info).</li> <li>- System validates action against ticket status and business rules.</li> <li>- System updates ticket and notifies customer if required.</li> </ul>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Refund processing <ul style="list-style-type: none"> <li>- Admin initiates refund.</li> <li>- System validates refund eligibility and calls Secure Payment Processing.</li> <li>- On successful refund, system updates ticket to “refunded” and adjusts revenue reports.</li> </ul> </li> <li>2. Forced cancellation for operational reasons <ul style="list-style-type: none"> <li>- Admin cancels due to bus breakdown/route change.</li> <li>- System processes refunds or alternative rebooking options.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Ticket already used/boarded → modification blocked.</li> <li>- Payment provider refund failure → system retries or flags for manual handling.</li> <li>- Attempt to delete a paid/confirmed ticket → system rejects.</li> </ul>
Priority	Medium
Business Rules	Admin cannot delete a paid ticket.

## 5.15 Manage Accounts

Use Case Name	Manage Accounts
Primary Actor	Administrator
Description	Admin manages user/admin accounts.
Preconditions	Admin authenticated with appropriate role (super-admin for admin account creation or high-privilege actions).
Postconditions	User or staff account created/updated/deactivated and audit trail recorded.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin opens Accounts module.</li> <li>2. Admin creates a new user/staff/admin account or updates an existing account's role/status.</li> <li>3. System validates role permissions and uniqueness constraints (username/email).</li> <li>4. System saves the account and sends an activation notification/email.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Reset password <ul style="list-style-type: none"> <li>- Admin triggers password reset for a user; system sends reset link or temporary password.</li> </ul> </li> <li>2. Deactivate account <ul style="list-style-type: none"> <li>- Admin deactivates account; system prevents further logins and records the action.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Attempt to create another admin by an admin without super-admin rights → system rejects.</li> <li>- Deleting an account that owns active schedules or transactions → system requires reassignment or blocks deletion.</li> </ul>
Priority	Medium
Business Rules	Only super-admin can create another admin.

## 5.16 Revenue Report

Use Case Name	Manage Accounts
Primary Actor	Administrator
Description	View revenue generated across time periods.
Preconditions	<ul style="list-style-type: none"> <li>- Admin authenticated and authorized to view financials.</li> <li>- Financial transactions (confirmed tickets/payments) are present in the database.</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>- Revenue report generated and displayed/exported; system logs report generation for compliance.</li> </ul>
Main Flow	<ul style="list-style-type: none"> <li>- Admin selects reporting period (date range, filters).</li> <li>- System aggregates confirmed transaction data, fees, refunds, and computes totals.</li> <li>- System presents revenue charts/tables and offers export (CSV/PDF).</li> </ul>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Export only summary <ul style="list-style-type: none"> <li>- Admin selects “Export Summary” → system generates lightweight report without raw transaction details.</li> </ul> </li> <li>2. Filter by company/route <ul style="list-style-type: none"> <li>- Admin applies filters → system recomputes aggregates and displays filtered report.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Insufficient permissions → system denies access.</li> <li>- Data warehouse or aggregation job failed → system returns an error and suggests retry or contact ops.</li> <li>- Partial data due to pending refunds → system notes “data incomplete” with timestamp.</li> </ul>
Priority	Medium
Business Rules	Only admin role can view revenue data.

## 5.17 Trip Statistics

Use Case Name	Manage Accounts
Primary Actor	Administrator
Description	Admin views statistics such as number of booked seats, trip performance.
Preconditions	<ul style="list-style-type: none"> <li>- Admin authenticated and authorized.</li> <li>- Confirmed booking and trip data is available in analytics DB.</li> </ul>
Postconditions	Statistical dashboards updated and snapshots stored if requested.
Main Flow	<ul style="list-style-type: none"> <li>- Admin selects metrics/time frame (e.g., bookings per route, load factor, cancellations).</li> <li>- System queries analytics, computes KPIs (occupancy rate, average fare, cancellations).</li> <li>- System displays charts, tables, and key insights; admin can export or schedule reports.</li> </ul>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Drill-down into a single trip <ul style="list-style-type: none"> <li>- Admin clicks a data point → system displays detailed trip-level metrics and passenger list (subject to privacy rules).</li> </ul> </li> <li>2. Schedule automated report <ul style="list-style-type: none"> <li>- Admin configures scheduled report → system stores schedule and triggers periodic generation.</li> </ul> </li> </ol>
Exceptions	<ul style="list-style-type: none"> <li>- Analytics pipeline lag (data not up-to-date) → system displays last available timestamp and warns data freshness.</li> <li>- Query timeouts for large date ranges → system suggests narrower range or async export.</li> </ul>
Priority	Medium
Business Rules	Stats must be based on confirmed tickets only.