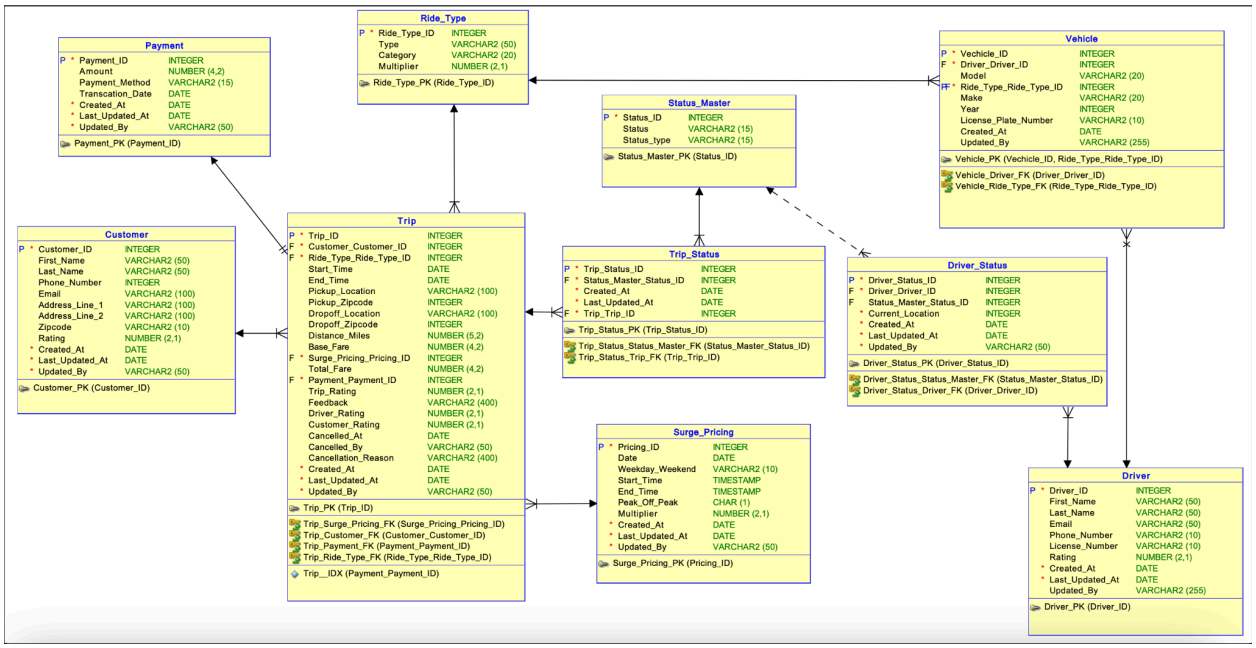
ER Diagram:



The updated schema consolidates previously fragmented tables (like `Trip_Cancellation`) directly into the `Trip` table. This avoids data duplication and adheres to normalization and **ACID principles**, while maintaining clarity and scalability in design.

The ERD follows normalization up to **Third Normal Form (3NF)**, as explained below:

1NF (First Normal Form): Eliminate Repeating Groups

- All attributes are atomic. For instance, customer addresses are split into `Address_Line_1` and `Address_Line_2`, rather than stored as a single composite field.
- Every column contains only one value per row.

Example:
`Phone_Number`, `Email`, and `Zipcode` are separate atomic fields in the `Customer` table.

2NF (Second Normal Form): Eliminate Partial Dependencies

- Each non-key attribute is fully functionally dependent on the **entire** primary key.

Example:
In the `Trip` table, every detail (e.g., `Pickup_Location`, `Dropoff_Zipcode`, `Total_Fare`, etc.) depends solely on the primary key `Trip_ID`, not on partial information like `Customer_ID` or `Ride_Type_ID`.

3NF (Third Normal Form): Eliminate Transitive Dependencies

- All fields are only dependent on the primary key — not on other non-key attributes.

Example:
The status of a trip is moved into a separate `Trip_Status` table and linked via a foreign key to the `Status_Master` table, keeping status-related data centralized and avoiding duplication across the schema.

Key Improvements Based on Feedback

1. **Removed Redundant Mapping Tables**:
    - The earlier one-to-one `Trip_Cancellation` table caused duplication with mappings to both customer and driver.
    - Now, fields like `Cancelled_At`, `Cancelled_By`, and `Cancellation_Reason` are embedded directly into the `Trip` table, reducing unnecessary joins and redundancies.
2. **Clearer Entity Roles**:
    - Entities like `Trip_Status` and `Driver_Status` now act as proper bridges to the `Status_Master`, enforcing consistency across status values (e.g., "Ongoing", "Cancelled", "Completed").
3. **Normalized Time-Based Pricing**:

    - The `Surge_Pricing` table holds date/time/multiplier logic separately, which the `Trip` table references via a foreign key.

1st Normal Form (1NF) – No Repeating Groups

- **Customer**: All attributes like name, phone, email, and address fields are atomic—no repeating groups.
- **Trip**: Each trip record contains only one value per attribute (e.g., one start time, one dropoff location).
- **Vehicle**: Each record refers to one vehicle with one model, make, year, etc.
- **Surge_Pricing** and others: Also conform to atomicity.

2nd Normal Form (2NF) – No Partial Dependencies

- Each **non-key attribute** is **fully dependent** on the **primary key** of its respective table.
- Example: In the **Trip** table, attributes like `Pickup_Location`, `Dropoff_Location`, and `Base_Fare` depend entirely on `Trip_ID`, not on only `Customer_ID` or `Ride_Type_ID` separately.

3rd Normal Form (3NF) – No Transitive Dependencies

- No **transitive dependencies** where non-key attributes depend on something **other than the primary key**.
- Example: `Trip_Rating` in the **Trip table** depends directly on `Trip_ID`.

Business Rules:

1. Customer Management Rules:

- Customer registration requires essential details like First Name, Last Name, and Phone Number.
- A customer's rating must be between 1.0 and 5.0.
- A customer must have a unique email and phone number.
- Customer addresses are optional, but if provided, a Zip Code must also be included.

- A customer’s record must track creation and updates with auto-generated timestamps for creation and last update.
- The system should track the last user or system action that updated the record.

2. Driver Management Rules:

- A driver must provide a valid license number before taking trips.
- A driver’s rating must be between 1.0 and 5.0.
- A driver must have a unique email and phone number.
- A driver’s availability status must be updated in real-time.
- The driver's status can be "Active," "Offline," or "On a Trip."
- A driver’s record must track creation and updates with auto-generated timestamps for creation and last update.

3. Trip Management Rules:

- A customer and a driver cannot be associated with two trips at once.
- A trip must be associated with exactly one customer and one driver.
- A trip must have a valid ride type to determine pricing and availability.
- A trip must have a status: "Requested," "Ongoing," "Completed," or "Cancelled."
- A trip must have start and end times and end time > start time
- A trip must have pickup and dropoff locations along with zipcode.
- The total fare for a trip is calculated based on base fare, distance, and surge pricing.
- Both the customer and the driver can provide ratings and feedback for the trip.
- A trip progresses through statuses: "Requested" → "Accepted" → "Ongoing" → "Completed."
- A trip’s total fare is calculated based on base fare, distance, and surge pricing.
  Total Fare = (Base_Fare * Surge_Multiplier * Ride_Type_Multiplier).
  Base fare = (Distance_Miles * standard rate per mile)

4. Trip Cancellation Rules:

- A cancellation must include a valid reason, such as "Rider No-show" or "Driver Delayed."
- The system must record the timestamp when a trip is canceled.

5. Payment Rules:

- Payments cannot be processed for canceled trips.
- A customer can have multiple payment methods, but only one can be used per trip.
- A payment record must track creation and updates with auto-generated timestamps for creation and last update.

6. Vehicle Rules:

- Each vehicle must be associated with exactly one driver.
- Each vehicle must have a unique license plate number.
- A vehicle's details, including Model, Make, and Year, must be correctly stored, with the Year being within a reasonable range (e.g., 2000 - Current Year).

7. Surge Pricing Rules:

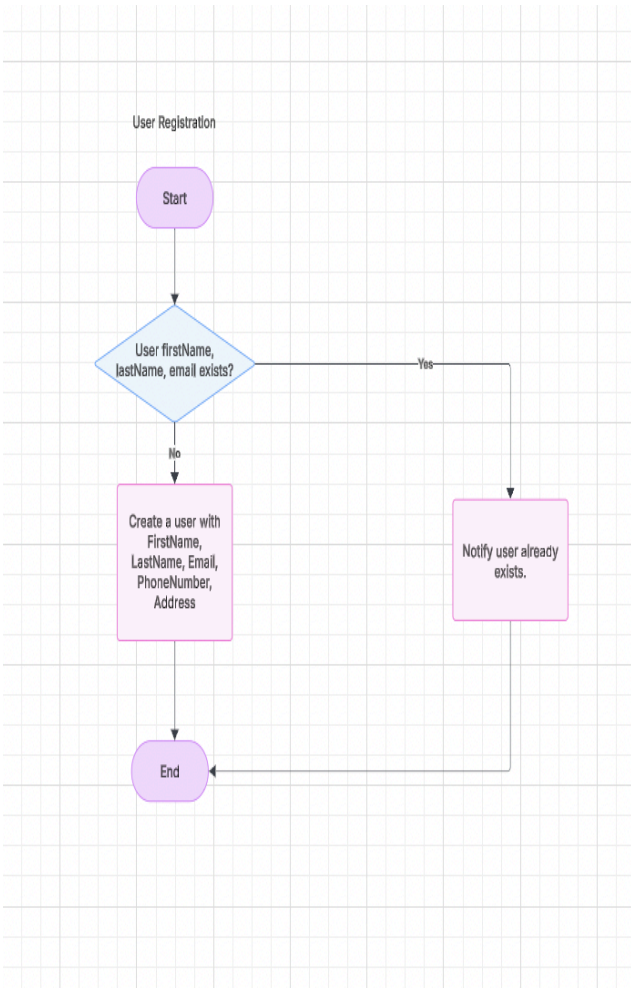
- Surge pricing is determined based on demand, date, and time.
- The surge pricing multiplier must be greater than or equal to 1.0.
- A trip's final fare must include the surge pricing multiplier if applicable.
- Surge pricing rules must be added periodically, with Peak_Off_Peak indicating peak (1) or off-peak (0) hours.

Views:

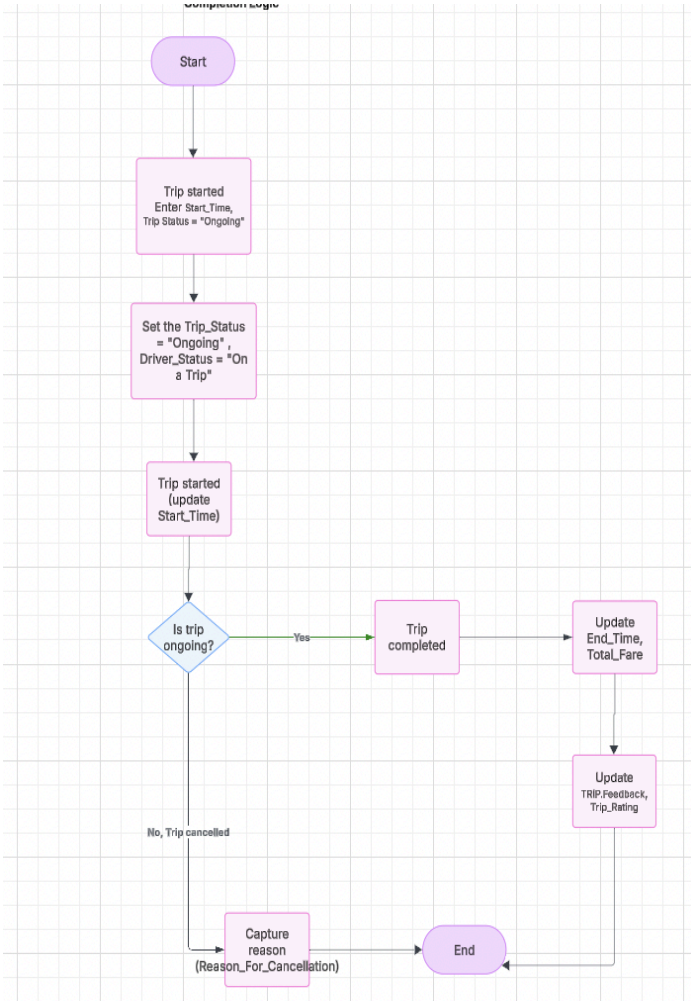
VW_CUSTOMER_PROFILE	Displays basic customer information including name, contact, and address.
VW_CUSTOMER_TRIPS	Shows trip details for each customer including status, fare, ride type, and driver info.
VW_DRIVER_TRIPS	Lists trips handled by each driver with trip timings, earnings, and status.
VW_SUPPORT_CASES	Provides details of cancelled trips for support analysis, including cancellation reason and feedback.
VW_REQUESTED_TRIPS	Lists trips currently in 'Requested' status along with pickup and drop-off details.
VW_AVAILABLE_DRIVERS	Identifies available or soon-to-be-available drivers based on ZIP code proximity and current trip info.
VW_POORLY_RATED_TRIPS	Displays trips rated below 3 to help support or fleet managers investigate issues.
VW_DRIVER_WEEKLY_EARNINGS	Summarizes each driver's weekly trips and total earnings.
VW_DRIVERS_WITHOUT_LICENSE	Lists registered drivers who have not yet provided a license number.

DFD (Data Flow Diagrams)

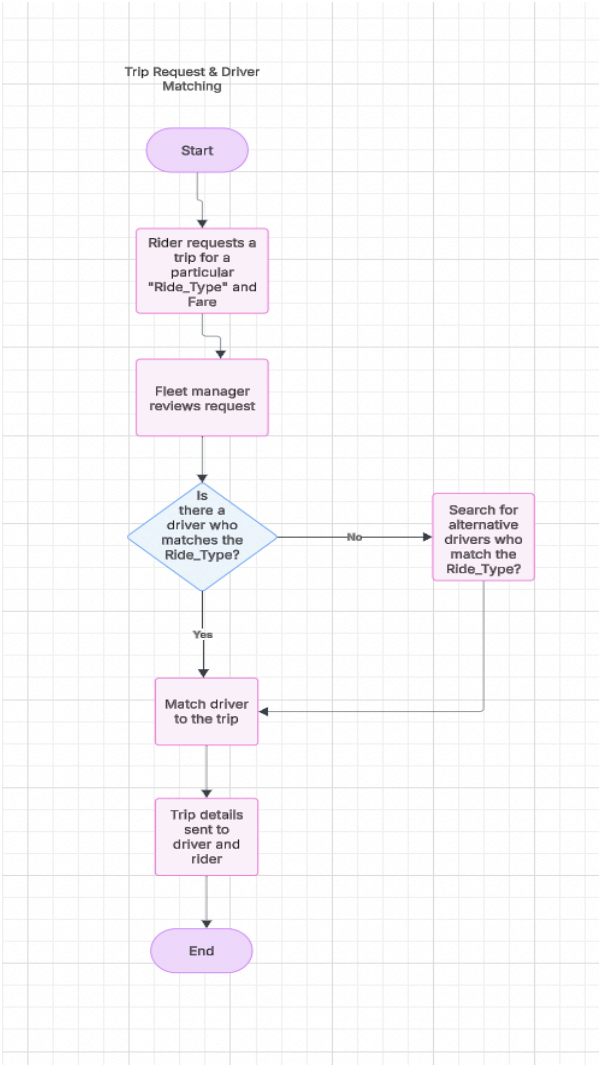
User registration flow



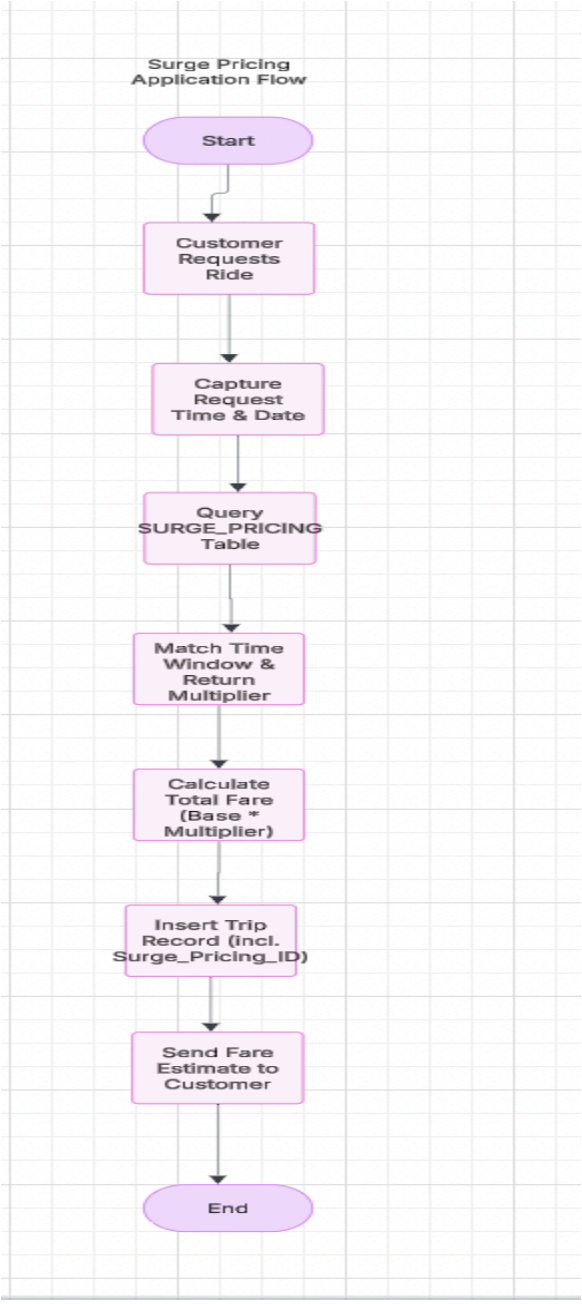
Trip Start & Completion



Trip Request & Driver Matching



Surge Pricing Application Flow



Payment Workflow