

SWINBURNE UNIVERSITY OF TECHNOLOGY

School of Science, Computing and Engineering Technologies

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: SWE30003

Unit Title: Software Architectures and Design

Assignment number and title: 1. Requirements

Due date: 27th Feb 2025

Tutorial Day and Time: 13:00 - 16:00 Thu

Project Group: 5

Tutor: Dr. Do Thi Bich Ngoc

To be completed as this is a group assignment

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature
<u>104182779</u>	<u>Do Duy Hung</u>	<u>Hung</u>
<u>104221559</u>	<u>Doan Trung Nghia</u>	<u>Nghia</u>
<u>104181857</u>	<u>Luong Minh Duc</u>	<u>Duc</u>
<u>104175326</u>	<u>Do Quang Minh</u>	<u>Minh</u>

Marker's comments:

Total Mark: _____

Extension certification:

This assignment has been given an extension and is now due on _____

Signature of Convener: _____

SWINBURN UNIVERSITY OF TECHNOLOGY	1
ASSIGNMENT AND PROJECT COVER SHEET	1
1, Introduction	3
2, Project Background	3
2.1, Domain Vocabulary	3
2.2, Goals	4
2.3, Assumptions	4
2.4, Scope	5
Beside that, if we have a deployment, we will develop some functions like:	6
3, Problem Domain	6
3.1, Pain Points	6
3.2, Domain Entities	6
3.3, Actors	7
3.4, Tasks	7
4, Functional Requirements & Task Descriptions	7
Task 1: Create an account	7
Task 2: Login	8
Task 3: Book a ride	9
Task 4: Track the Ride	10
Task 5: Process Payment	11
Task 6: Rate & Review	11
Task 7: Manage Platform	12
Task 8: Administer & Report System	13
Task 9: Book Additional Services	14
5, Workflow	16
6, Data Model	25
6.1 Domain Model	25
6.2 Entity descriptions	25
7, Quality Attributes of System	26
7.1 Security	26
7.2 Usability	27
Current User Complaints	27
Meeting Usability Requirements	27
7.3 Reliability	27
Driver Assignment Optimization	27
Problem Statement	27
Problem Statement	28
Impact on the System	28
Digitalized Solutions for Optimization	28
7.4 Performance	29
1. Scalable Infrastructure	29
2. High-Performance Ride-Matching Algorithm	29
3. Load Balancing for Optimal Performance	29
4. Predictive Demand Management	29

7.5 Portability	30
8, Other Requirements	30
8.1 Product level requirements	30
8.2 Design level requirements	30
9, Validation of Requirements	31
10, Possible Solutions	31
Employee-Focused System for SmartRide	31
10.1 System for Drivers and Employees	32
a. Driver Account & Profile Management	32
b. Ride Assignment & Management	32
c. Payment Processing & Earnings Management	32
d. Support & Complaint Handling	32
10.2 System for SmartRide Employees	32
a. System Management & Ride Coordination	32
b. Customer & Driver Management	33
c. Reporting & Data Analytics	33
d. Complaint Resolution & Customer Support	33
10.3 AI-Powered Driver & Ride Matching	33

1, Introduction

SmartRide, a transportation service provider in a busy city, faces inefficiencies due to its manual ride-booking and payment processes. Customers often experience long wait times, especially during peak hours, while drivers struggle with unoptimized ride assignments. Manual payments further slow down operations, limiting growth and customer satisfaction.

To solve these issues, SmartRide plans to develop an Online Ride-Sharing Platform (ORSP) to automate ride bookings, provide real-time driver tracking, and support secure online payments. This report examines SmartRide's challenges and presents a scalable solution to enhance efficiency, improve user experience, and support future business expansion.

2, Project Background

The CEO of SmartRide has prioritized redesigning the current system and processes for booking and managing ride-sharing services. The existing system is a mix of manual and digital processes, leading to inefficiencies, long wait times for customers, and reduced service reliability. To address these issues, the CEO has engaged Swinsoft Consulting to analyze and develop specifications for a new Automated Ride-Sharing System that will streamline operations, enhance user experience, and support business growth.

2.1, Domain Vocabulary

- **Ride-sharing:** A transportation service where passengers book a ride through an app to travel from one location to another, usually using private drivers.
- **Driver:** A person who operates a vehicle to transport passengers from their

pickup location to their destination.

- **Passenger:** A customer who books and takes a ride using the ride-sharing platform.
- **Booking system:** A digital platform that allows users to request, schedule, and confirm rides.
- **Real-time tracking:** A feature that allows passengers and drivers to see the exact location of the vehicle at any given moment.
- **GPS navigation:** A system that helps drivers find the best routes to their destinations using satellite-based positioning.
- **Online payment:** A digital method that allows customers to pay for their rides using credit cards, e-wallets, or other electronic payment options.
- **Estimated Time of Arrival (ETA):** A calculated prediction of when the driver will arrive at the passenger's pickup location or destination.
- **User authentication:** A security process that verifies a user's identity before allowing access to the platform.
- **Fare calculation:** A system that determines the cost of a ride based on factors such as distance, time, and demand.
- **Dynamic pricing:** A pricing strategy where ride fares change based on factors like demand, traffic, and availability of drivers. This ensures a balance between supply and demand.
- **Driver assignment algorithm:** A system that automatically matches passengers with the most suitable drivers based on location, estimated arrival time, and availability.

2.2, Goals

The goals of this SmartRide project are:

- **Automate the ride-booking process** – Transition from a manual system to a digital platform, enabling customers to book rides quickly and reducing wait times.
- **Enhance user experience** – Provide real-time tracking, transparent fare calculation, and online payment options to improve convenience and customer satisfaction.
- **Optimize driver assignment** – Implement smart algorithms to efficiently match drivers with passengers, ensuring a balance between supply and demand, especially during peak hours.
- **Support business expansion and growth** – Develop a scalable system that can accommodate increasing demand and integrate advanced features in the future.

2.3, Assumptions

- **User Access:** Customers and drivers have smartphones and internet access
- **Service Area:** Initially limited to one city, with potential future expansion
- **Ride Availability:** Matching based on driver proximity and availability; peak hours may cause delays
- **Payments:** Only digital payments via secure third-party gateways
- **GPS Tracking:** Real-time location updates depend on accurate GPS data and enabled location services
- **Safety & Security:** Drivers are verified, customers and drivers can report issues, rating system ensures service quality
- **System Performance:** Designed for high traffic
- **Pricing:** Dynamic pricing based on distance, time, and demand
- **User Roles:** Customers book rides, drivers accept and complete them, admins manage operations

2.4, Scope

The scope of the SmartRide defines the functionalities and limitations of the system in its initial development phase. The key aspects of the project scope include:

- **Development of the Online Ride-Booking Platform**
 - Build a ride-booking system for mobile and web applications.
 - Allow customers to input pickup and drop-off locations to request rides.
 - Display available drivers and provide estimated time of arrival (ETA).
- **Real-Time Tracking and Navigation System**
 - Integrate GPS for tracking driver locations and route guidance.
 - Provide real-time trip status updates.
- **Driver Assignment Optimization**
 - Implement an algorithm to automatically match drivers with passengers based on location and availability.
 - Ensure efficient driver allocation, especially during peak hours.
- **Online Payment and E-Receipts**
 - Support payments via credit cards, e-wallets, or bank transfers.
 - Send electronic receipts to customers after each ride.
- **User Account Management System**
 - Allow customers and drivers to create and update their profiles.
 - Implement user authentication to ensure security.
- **Reporting and Data Analytics**
 - Provide reports on ride volume, revenue, and booking trends.
 - Help management make data-driven decisions to optimize service.

Beside that, if we have a deployment, we will develop some functions like:

- Integration of loyalty programs or promotional offers.
- Support for transportation methods beyond cars and motorbikes.
- Advanced features such as ride-sharing (carpooling) or dynamic pricing.

This scope ensures that SmartRide focuses on solving key challenges first before expanding to additional features in the future

3, Problem Domain

3.1, Pain Points

Here are some pain points that SmartRide is facing:

- **Long wait times for customers** – During peak hours, customers experience delays in finding available drivers, leading to frustration and ride cancellations.
- **Inefficient driver assignment** – The current system lacks an automated way to efficiently match drivers with passengers, causing delays and uneven distribution of ride requests.
- **Manual and time-consuming payment process** – Handling payments manually slows down operations, increases errors, and causes inconvenience for both drivers and passengers.
- **Limited real-time tracking** – Customers cannot accurately track their ride status, leading to uncertainty and dissatisfaction.
- **Scalability issues** – The current system struggles to handle high demand and lacks flexibility for future expansion.
- **Lack of data-driven decision-making** – Without proper reporting and analytics, SmartRide faces challenges in optimizing operations and predicting demand trends.

Addressing these pain points is key to enhancing efficiency, improving customer experience, and supporting business growth for SmartRide.

3.2, Domain Entities

- **User**
 - Represents all system users, including passengers and drivers.
- **Passenger**
 - A person who books and uses the ride-sharing service.
 - Attributes: ID, name, phone number, payment method, ride history.
- **Driver**
 - A person who provides transportation services to passengers.
 - Attributes: ID, name, vehicle, availability status, customer ratings.
- **Ride**
 - Represents a trip booked between a passenger and a driver.
 - Attributes: Ride ID, pickup location, drop-off location, fare, ride status (pending, ongoing, completed, canceled).
- **Vehicle**
 - Information about the driver's vehicle, including type (car, motorcycle),

license plate, seating capacity.

- **Payment**
 - Represents the payment process for a ride.
 - Attributes: Transaction ID, payment method, total amount, payment status.
- **Location**
 - Represents the pickup and drop-off points of a ride.
 - Attributes: GPS coordinates, specific address.
- **Review & Rating**
 - Feedback provided by passengers for drivers and vice versa.
 - Attributes: Star rating, comments, reviewer ID.
- **Ride Request**
 - A request submitted by a passenger before a driver is assigned.
 - Attributes: Request ID, pickup location, drop-off location, request time, status (waiting, accepted, canceled).
- **Admin**
 - Attributes: Request ID, pickup location, drop-off location, request time, status (waiting, accepted, canceled).

3.3, Actors

- Driver
- Passenger
- Manager

3.4, Tasks

- Create an account
- Log In
- Book a ride
- Track the ride
- Complete & Pay
- Rate & Review
- Manage platform
- Administer & Report System
- Booking Additional Services (in the future)

4, Functional Requirements & Task Descriptions

Task 1: Create an account

Task: Create account	
Purpose:	Allow users and drivers to create an account to access SmartRide services.
Trigger/Precondition:	The user has not registered before.
Frequency:	Done once per user/driver, unless they create a

	new account.
Critical:	If the account is not created successfully, the user cannot book or accept rides.
Subtasks:	Example Solution:
1. Fill in personal details	Users provide their name, phone number, and email.
2. Set up a password	Users create a secure password for login.
3. Verify identity	Users receive a one-time verification code via SMS or email to confirm their identity.
4. Upload documents (for drivers)	Drivers submit their license and vehicle details for verification.
5. Approval process (for drivers)	The system checks and approves driver documents before they can accept rides.
6. Confirmation	Users receive a success message and can log in to the app.
Variants:	
1. User enters incorrect verification code	Prompt the user to re-enter the code or resend a new one.
2. Driver's documents are pending approval	Notify the driver that their account is under review before they can accept rides.
3. User forgets password during registration	Provide an option to reset the password immediately.

Task 2: Login

Task: Login	
Purpose:	Allow users and drivers to access their SmartRide accounts securely.
Trigger/Precondition:	The user or driver already has a registered account.
Frequency:	Every time a user or driver wants to access their account.
Critical:	If login fails, users and drivers cannot book or accept rides.
Subtasks:	Example Solution:

1. Open the login page	The person selects the login option in the app.
2. Provide account details	They enter their phone number or email and their password.
3. Verify identity	If extra security is needed, a verification code is sent for confirmation.
4. Access the account	Once everything is correct, the person reaches their homepage.
Variants:	
1. The password is incorrect	A message appears suggesting a retry or password reset.
2. The person forgot their password	They can reset it using their phone number or email.
3. Extra security is needed	A verification code is sent to confirm the login.

Task 3: Book a ride

Task: Book a ride	
Purpose:	Let passengers request a driver to travel from one place to another.
Trigger/Precondition:	The passenger needs transportation and decides to schedule a ride.
Frequency:	Whenever a passenger wants to travel via ORSP.
Critical:	If the passenger cannot book a ride, they can't use the service, leading to lost trips and revenue.
Subtasks:	Example Solution:
1. Provide trip details	The passenger shares their pickup point and destination.
2. Select ride options	The passenger chooses a vehicle type (car or motorbike) and sees an estimated price.
3. Confirm the request	The passenger agrees to proceed, and SmartRide looks for an available driver nearby.
4. Driver receives the request	A nearby driver is notified and can choose to accept the ride.
5. Driver accepts the request	The driver confirms and begins heading to the pickup location.

6. Receive driver details and ETA	The passenger sees the driver's name, vehicle, and estimated arrival time.
7. Wait for pickup	The passenger gets a notification when the driver is close.
Variants:	
1. No drivers available	The passenger is informed and can try again later.
2. The driver cancels after accepting	The system finds a replacement driver and informs the passenger.
3. The passenger cancels before pickup	A cancellation fee may apply depending on how late they cancel.
4. Price increasing during peak hours	The passenger sees an updated fare and can choose to accept or wait.

Task 4: Track the Ride

Task:Track the Ride	
Purpose:	Passengers can see their driver's location and arrival time, while drivers get directions to the destination.
Trigger/Precondition:	<ul style="list-style-type: none"> - A ride has been booked and accepted. - The driver is on the way.
Frequency:	Happens during every ride.
Critical:	Without tracking, passengers won't know when the driver will arrive, and drivers may struggle to find the destination.
Subtasks:	Example Solution:
1. See driver's location	Passengers can follow the driver on a map.
2. Check arrival time	The system updates estimated arrival time.
3. Get notified when close	Passengers receive an alert when the driver is near.
4. Follow route guidance	The driver gets directions to the pickup and drop-off points.
Variants:	
1. The driver is delayed	The arrival time updates automatically.

2. The driver takes a wrong turn	The system corrects the route.
3. The passenger is not at pickup	The driver can call or message them.

Task 5: Process Payment

Task: Process Payment	
Purpose:	Make sure passengers can pay for their ride quickly and securely.
Trigger/Precondition:	<ul style="list-style-type: none"> - The ride has ended, and it's time for the passenger to pay. - The passenger has a valid payment method set up.
Frequency:	Happens at the end of every completed ride.
Critical:	If the payment doesn't go through, the driver won't receive their earnings, and the passenger may face issues using the service again.
Subtasks:	Example Solution:
1. Show the total fare	The passenger sees the final amount, including any fees or discounts.
2. Choose payment method	The passenger selects their preferred payment method (card, wallet, or other options).
3. Complete the transaction	The system confirms the payment and deducts the amount from the passenger's account.
4. Notify both passenger & driver	A confirmation appears, and both get a receipt for the ride.
Variants:	
1. Payment fails due to insufficient funds	The passenger is asked to use another payment method.
2. Passenger wants to use a different method	They can switch to another saved payment option before confirming the payment.
3. Refund is required	If an issue occurs, the platform processes a refund based on the policy.

Task 6: Rate & Review

Task: Rate & Review	
Purpose:	Allow users and drivers to provide feedback on their ride experience, helping to improve service quality.
Trigger/Precondition:	<ul style="list-style-type: none"> - A ride has been completed. - The user or driver wants to share their experience by giving a rating or review.
Frequency:	After every ride, the option is available for both users and drivers.
Critical:	<ul style="list-style-type: none"> - Helps maintain service quality and accountability. - Poor ratings can impact a driver's ability to receive ride requests or a user's standing on the platform.
Subtasks:	Example Solution:
1. Select the ride to review	The user or driver chooses the recently completed ride.
2. Give a rating	A rating is provided based on the experience (e.g., 1 to 5 stars).
3. Write a review (optional)	Additional comments can be provided about the ride experience.
4. Submit feedback	The rating is recorded and contributes to the overall profile score.
Variants:	
1. The user had a bad experience	A low rating and a comment can be provided for admin review.
2. The driver wants to report a difficult passenger	A review can be left with details of the issue.
3. The user forgets to rate the ride	A reminder may be sent to encourage feedback.

Task 7: Manage Platform

Task: Manage Platform	
Purpose:	Ensure smooth operations by overseeing drivers, monitoring ride activities, and resolving disputes.
Trigger/Precondition:	<ul style="list-style-type: none"> - A need arises to review driver approvals, ride issues, or user complaints.

	- The platform requires supervision to maintain service quality.
Frequency:	Performed regularly, depending on operational needs and emerging issues.
Critical:	Poor management can lead to disruptions, service delays, or unresolved conflicts affecting users and drivers.
Subtasks:	Example Solution:
1. Approve new drivers	Verify driver credentials before allowing them to provide rides.
2. Monitor ride activity	Keep track of ongoing trips, cancellations, and any reported issues.
3. Handle disputes	Address concerns from drivers or passengers and take appropriate action.
4. Manage user accounts	Decide on suspensions or reactivations based on rule violations.
Variants:	
1. A driver submits incomplete documents	Notify the driver and request additional information.
2. A user reports a dispute with a driver	Investigate the issue and take appropriate action.
3. System experiences downtime	Notify users and work on system recovery.

Task 8: Administer & Report System

Task: Administer & Report System	
Purpose:	Ensure the platform maintains a secure and well-regulated user and driver base.
Trigger/Precondition:	Admin needs to verify, update, or deactivate user accounts
Frequency:	As needed, based on user activity, violations, or registration requests
Critical:	High – essential for security, compliance, and platform trust.
Subtasks:	Example Solution:

1. Manage User & Drivers	Use an automated verification system that cross-checks driver licenses and vehicle registration details with government databases..
2. Monitor ride statistics	Implement automated alerts that notify administrators when ride demand surges beyond driver availability, helping with better resource allocation.
3. Handle Complaints & Disputes	Create a ticket-based support system where users and drivers can submit complaints, track resolution progress, and receive updates.
4. Generate Sales & Revenue Reports	Integrate a financial reporting module that automatically generates revenue summaries, expense breakdowns, and commission reports for drivers.
Variants:	
1. Role-Based Access Control for Admins	Implement different access levels for administrators, where some can only view reports while others can manage users and resolve disputes.
2. AI-Powered Fraud Detection	Use machine learning algorithms to detect suspicious activities, such as fake ride requests or fraudulent payments, and flag them for review.
3. Multi-Region Ride Monitoring	Enable regional admins to monitor ride statistics and handle disputes specific to their geographic location, improving localized service efficiency.

Task 9: Book Additional Services

Task: Book Additional Services	
Purpose:	Provides users with flexible ride options to meet specific needs such as scheduling, accessibility, or cost-saving.
Trigger/Precondition:	The user selects an additional service while booking a ride.
Frequency:	Used occasionally based on user preferences and special needs.

Critical:	Medium – Enhances user experience but is not essential for core ride operations.
Subtasks:	Example Solution:
1. Book a Scheduled Ride	The system allows users to select a future date and time for their ride, sending an automatic reminder before the trip.
2. Request a Special Vehicle	Users can filter available vehicles to select options like wheelchair-accessible cars, ensuring suitable transportation.
3. Book a Shared Ride	The system matches multiple passengers with similar routes, optimizing ride costs and reducing environmental impact.
Variants:	
1. Scheduled Ride with Recurring Option	Users can schedule daily or weekly recurring rides, such as a morning commute to work, reducing the need to book manually each time.
2. Premium Vehicle Selection	Users can choose premium vehicles (e.g., luxury cars or extra luggage space) for a more comfortable ride experience.
3. Group Ride Booking	The system allows users to book a ride for multiple passengers at once, ensuring enough seating capacity in a single vehicle.

5, Workflow

Task 1: Create an account

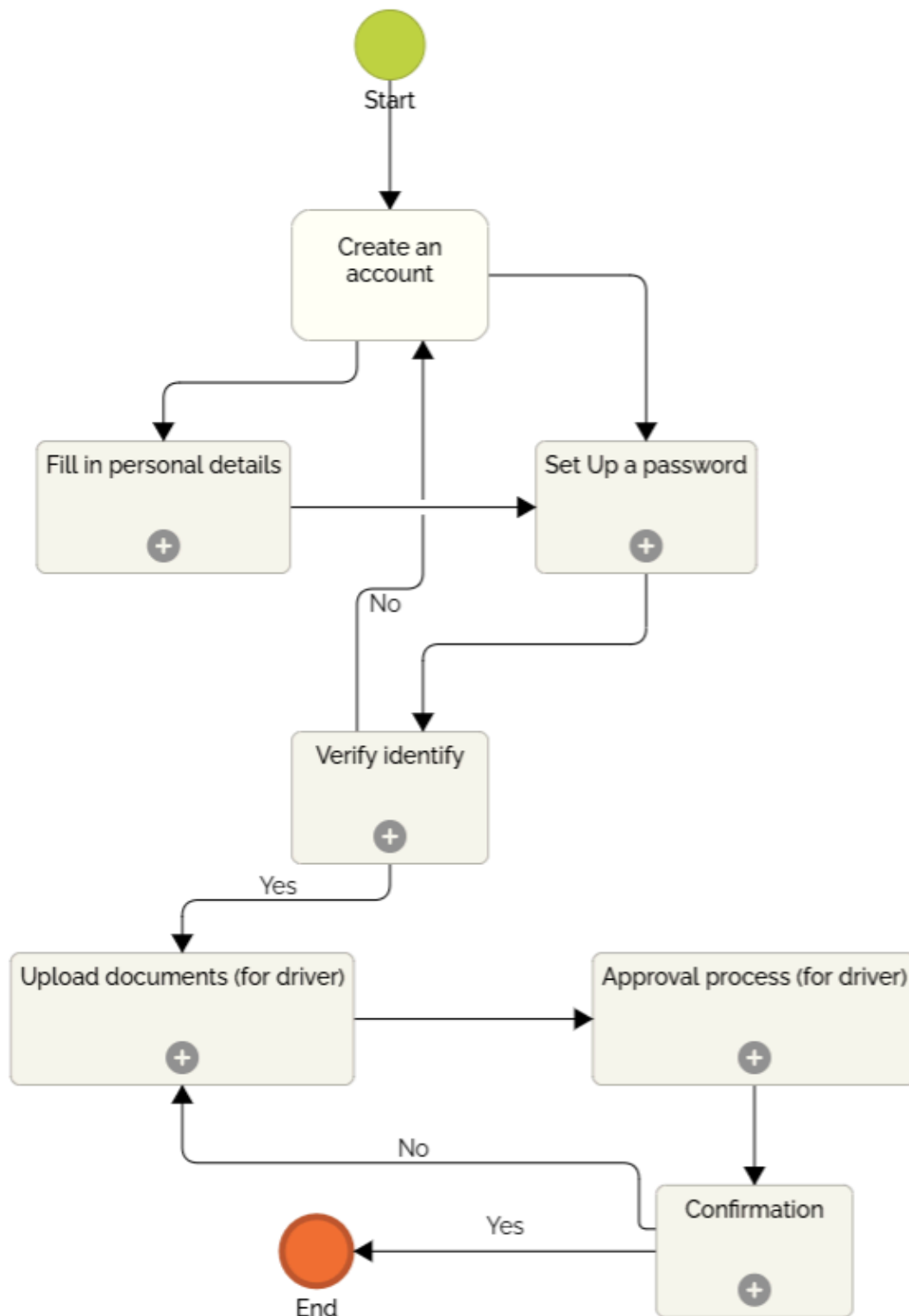


Figure 1 - Account creation model

Task 2: Login

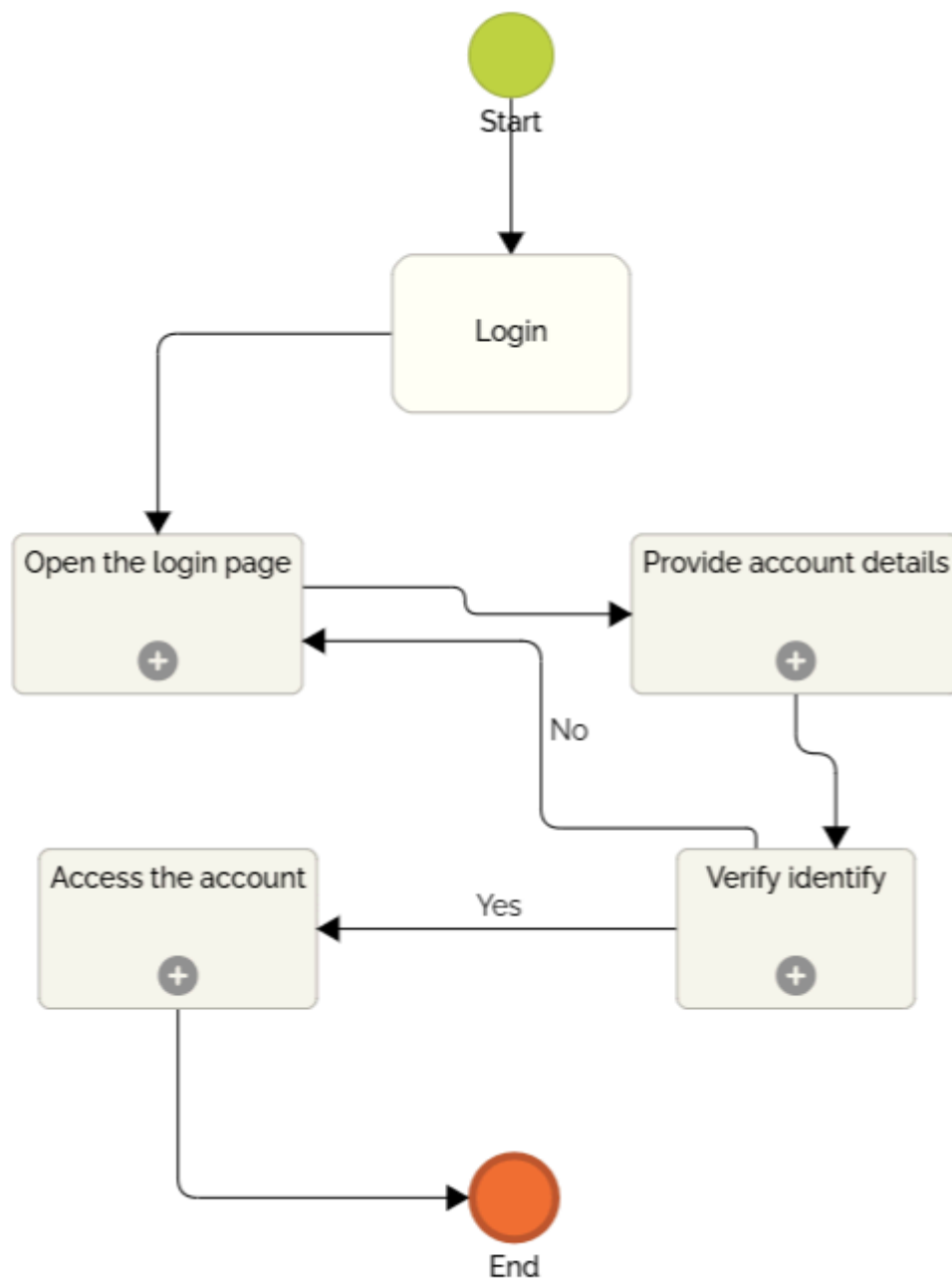


Figure 2 - Login model

Task 3: Book a ride

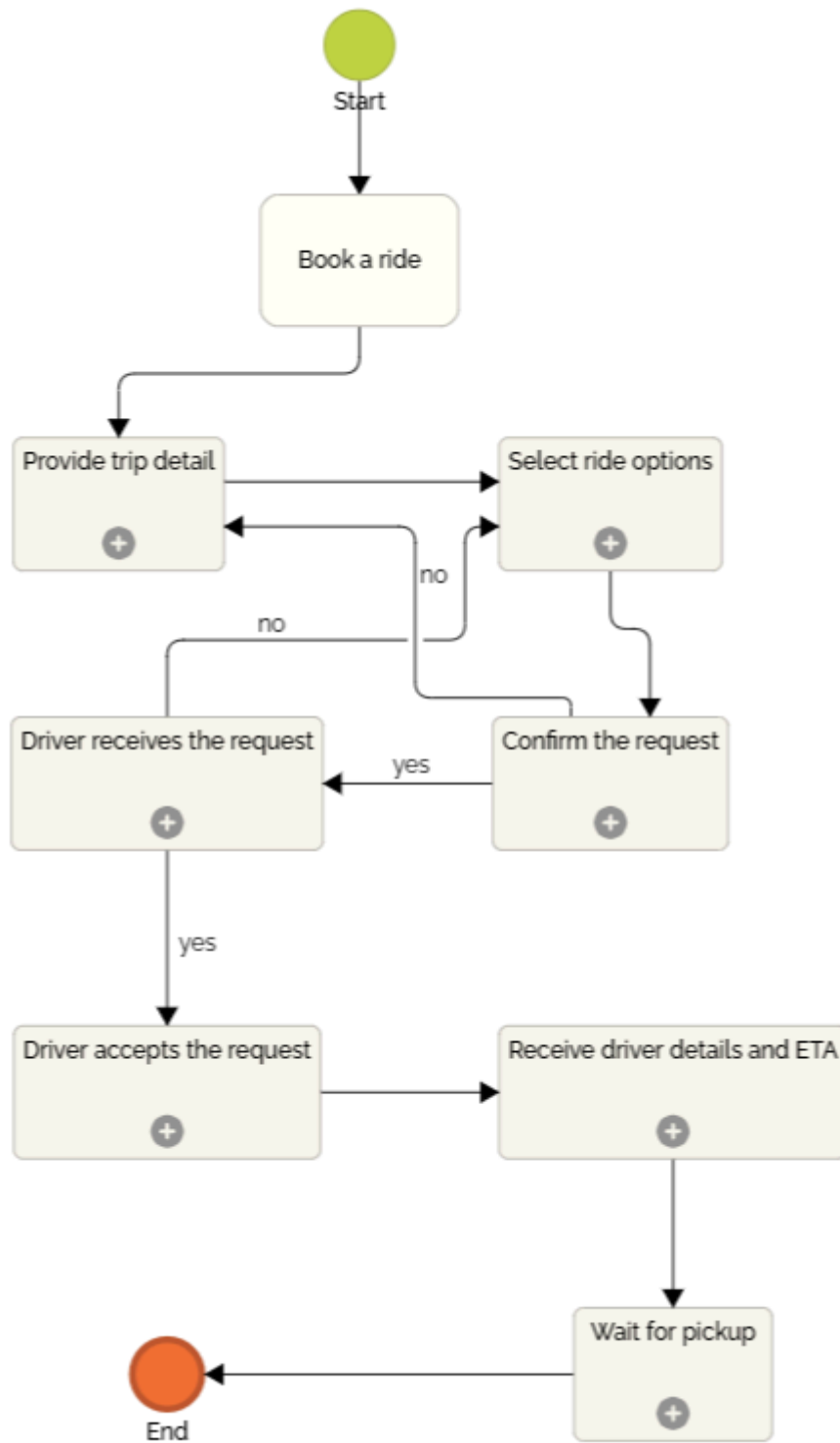


Figure 3 - Book a ride model

Task 4: Track a ride

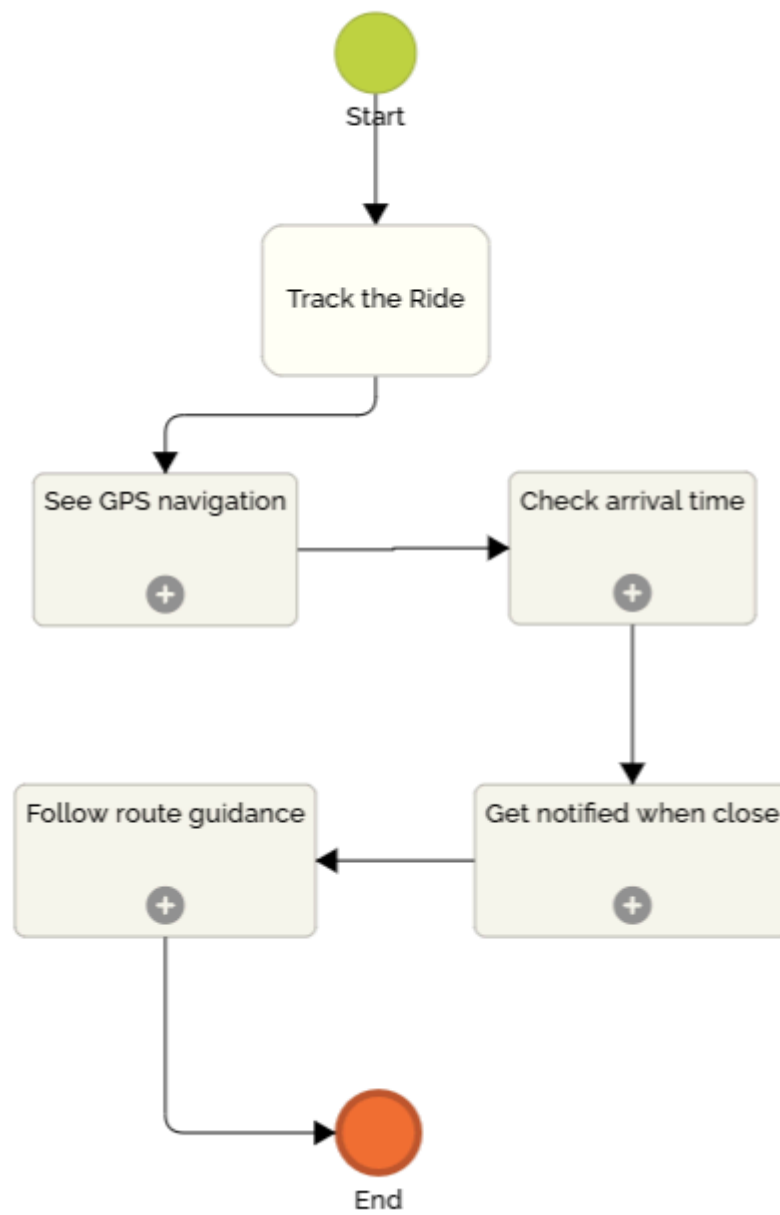


Figure 4 - Track a ride model

Task 5: Process Payment

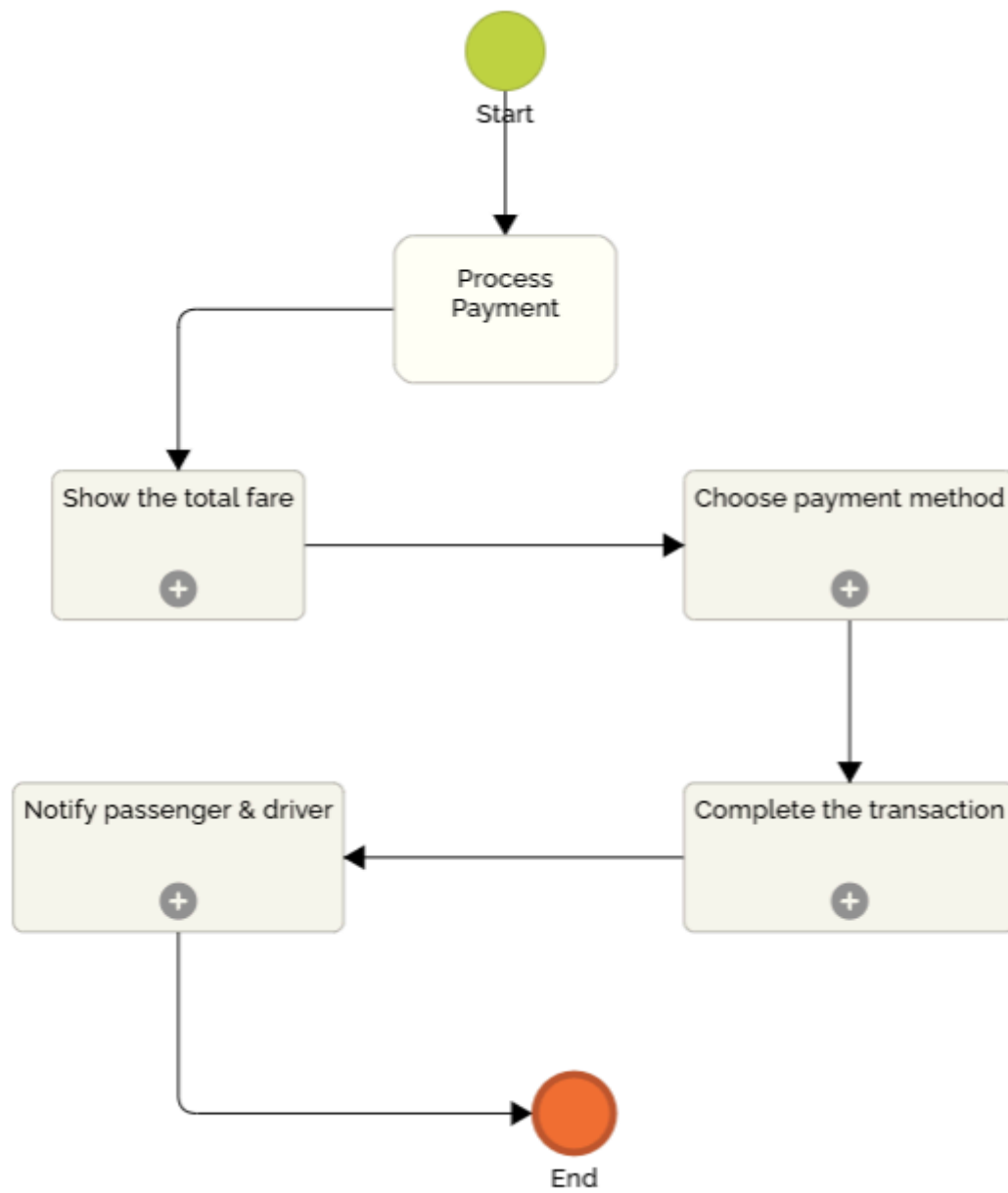


Figure 5 - Process payment model

Task 6: Rate & Review

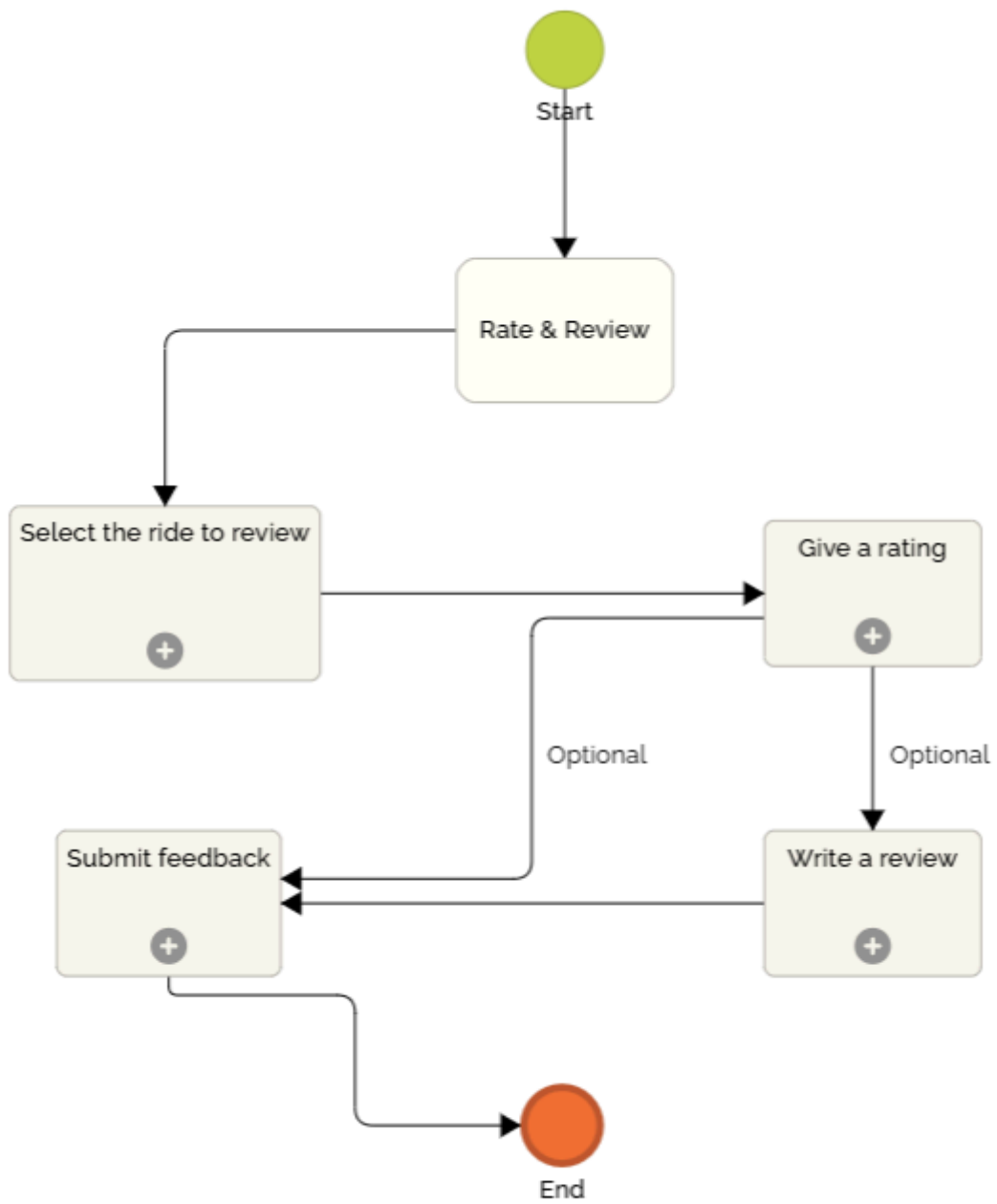


Figure 6 - Rate & Review model

Task 7: Manage Platform

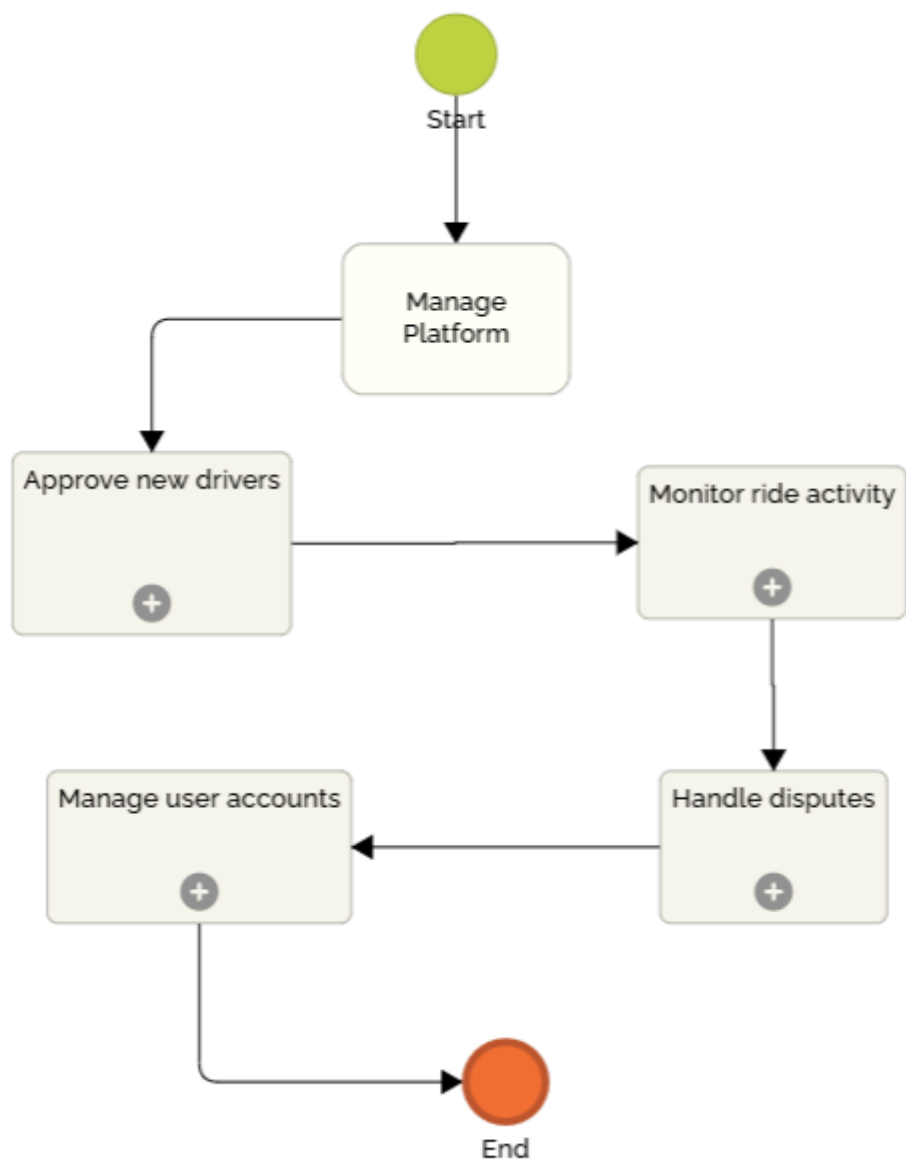


Figure 7 - Manage platform model

Task 8: Administer & Report System

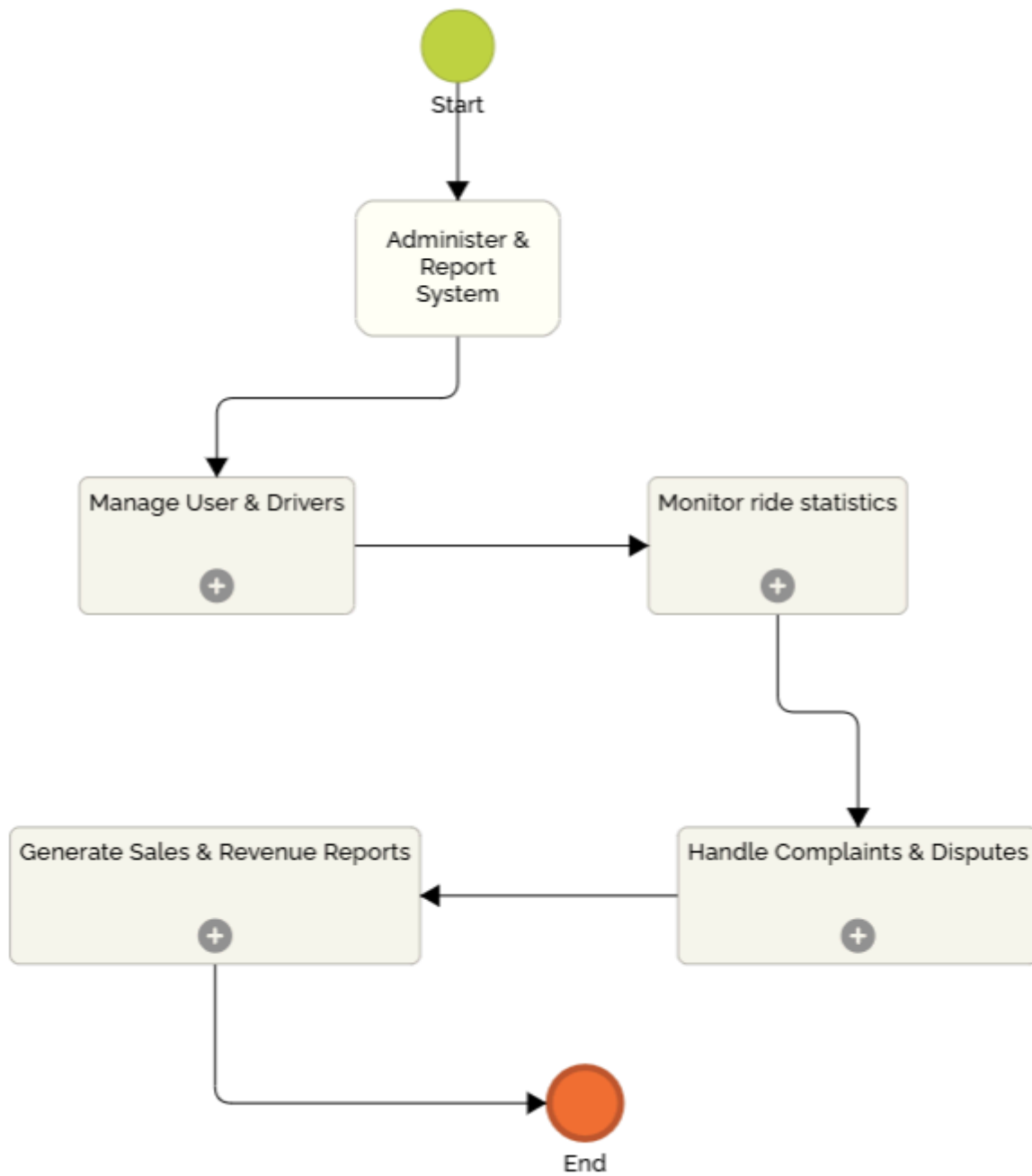


Figure 8 - Administer & Report system platform model

Task 9: Book Additional Services

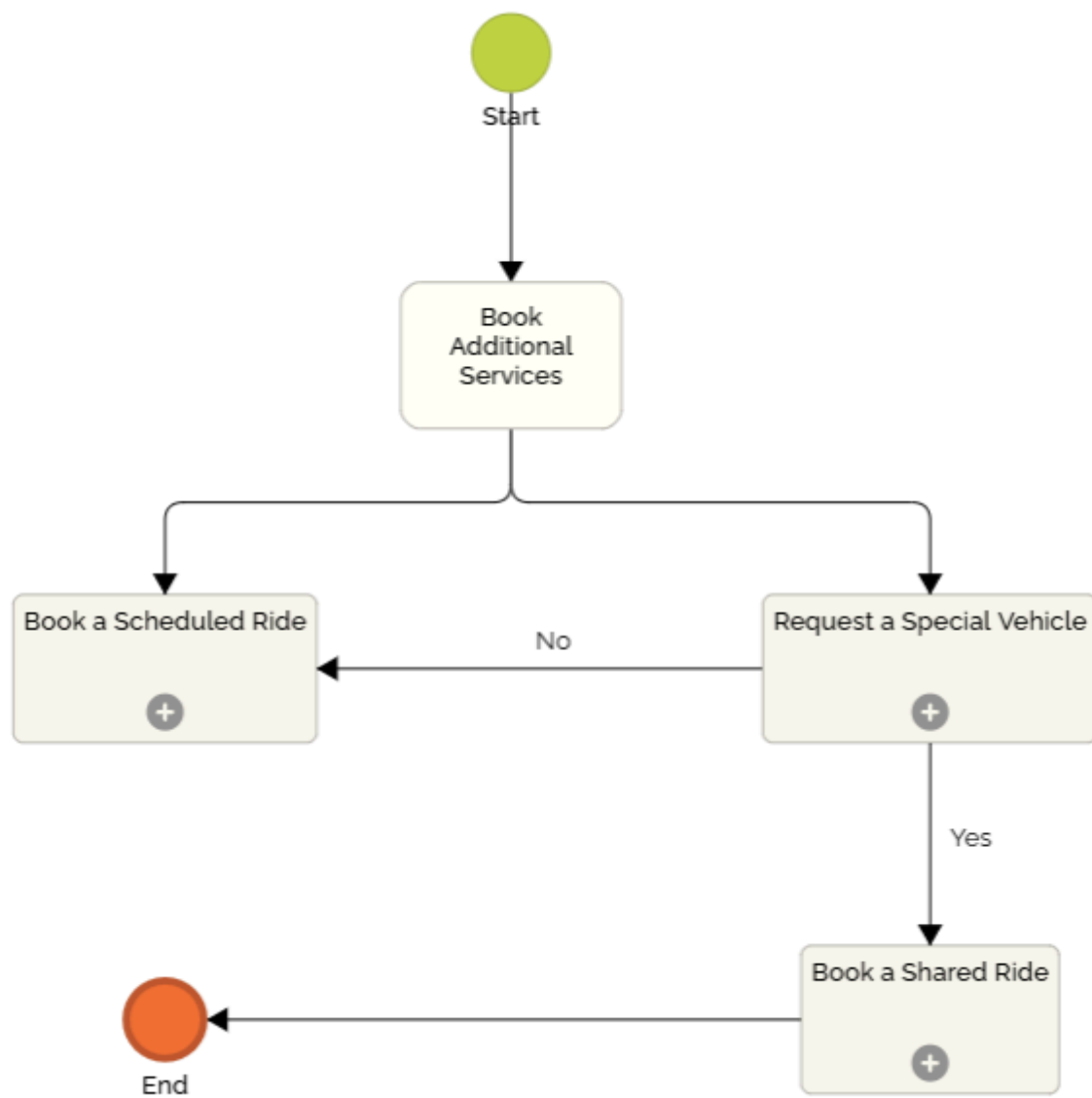


Figure 9 - Book Additional Services model

6, Data Model

6.1 Domain Model

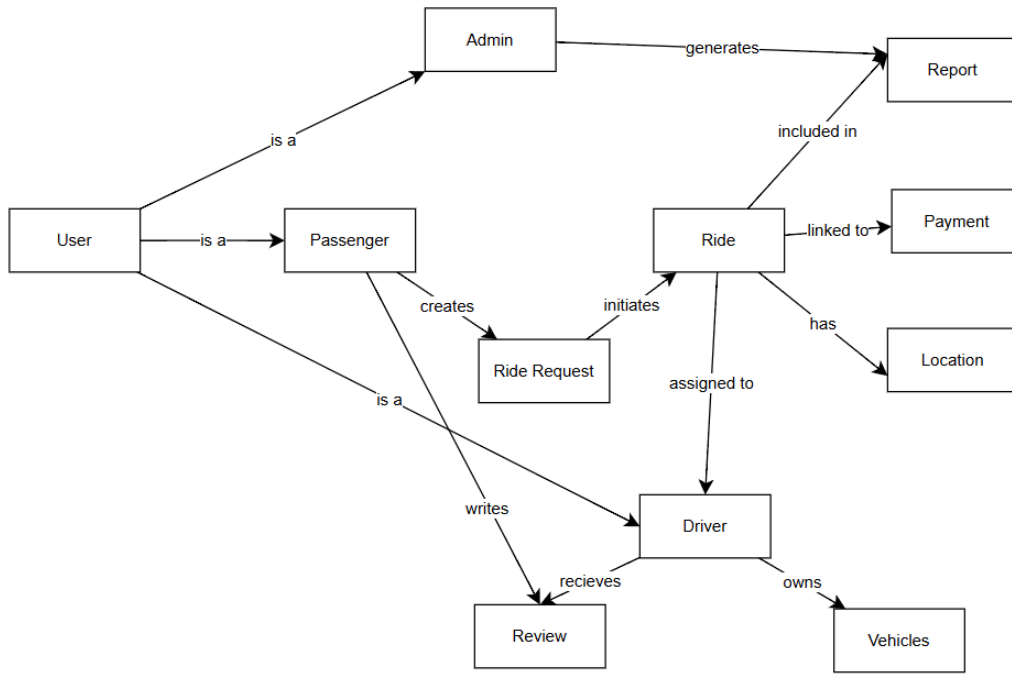


Figure 10 - Domain Model of the SmartRide

6.2 Entity descriptions

- **User** – Represents all platform users, including passengers, drivers, and admins, who can register, log in, and perform their respective roles.
- **Passenger** – A type of user who can book rides, make payments, and leave reviews for drivers.
- **Driver** – A type of user who provides transportation services, accepts ride requests, and completes trips.
- **Admin** – A system user responsible for managing accounts, monitoring reports, and handling customer complaints.
- **Ride** – Represents a trip between a passenger and a driver, containing details such as fare, pickup/drop-off locations, and ride status.
- **Vehicle** – Stores information about the driver's vehicle, including type, license plate, and seating capacity.
- **Payment** – Tracks ride payments, including transaction details, payment method, and status (paid, pending, refunded).

- **Location** – Represents the pickup and drop-off points of a ride, including GPS coordinates and addresses.
- **Review & Rating** – Allows passengers and drivers to provide feedback on their experience, including ratings and comments.
- **Ride Request** – A ride booking request created by a passenger before being assigned to a driver.
- **Report** – A system-generated document that provides insights on ride statistics, revenue, and customer issues.

7, Quality Attributes of System

7.1 Security

Handling customer and financial transaction information makes security a top priority for the system. The software must be designed and developed to prevent unauthorized access and fraudulent activities.

To meet this quality requirement, the system must:

- **Protect user information:** The system must ensure the security of personal data for both passengers and drivers, including login credentials, phone numbers, addresses, and ride history. Only authorized users should have access to or be able to modify this information.
- **Manage access rights:** Only users with specific roles should be able to access different parts of the system. Passengers can manage their personal information and book rides, drivers can accept and manage trips, while administrators can monitor system activity and handle violations.
- **Secure payments:** When making transactions, passengers should be able to pay using secure methods such as credit cards, digital wallets, or cash. The system should not store customers' card details but rather process transactions through a secure payment gateway.
- **Authenticate and protect accounts:** To prevent misuse, users must verify their identity before accessing services, especially when performing critical actions such as changing payment methods or withdrawing funds for drivers.
- **Monitor and control activities:** The system should track and record all key activities, such as logins, ride bookings, account changes, and financial transactions. If any suspicious activities are detected, the system should issue alerts and temporarily block the transaction for review.
- **Prevent fraud:** The system should have mechanisms to prevent fraudulent behaviors such as fake ride requests, impersonating drivers, or exploiting promotional programs. Suspicious actions should be detected and promptly addressed.

With these security measures, SmartRide can protect user data, ensure safe transactions, and maintain a reliable service environment for both passengers and drivers.

7.2 Usability

Current User Complaints

Users of the current manual system face several usability challenges, including:

- **Complex and time-consuming ride booking** – Customers struggle with inefficient booking processes, leading to delays and frustration.
- **Lack of real-time updates** – Passengers have no clear visibility of driver availability or estimated arrival times.
- **Difficulties with payment processing** – Manual payments cause delays, errors, and inconvenience for both drivers and customers.
- **Unintuitive user interface** – The system lacks a user-friendly design, making it difficult for new users to navigate.

Meeting Usability Requirements

To enhance usability, the SmartRide platform must:

- **Simplify the ride-booking process** – Reduce the number of steps required to book a ride and provide a seamless experience.
- **Enable real-time tracking and updates** – Display driver locations, estimated arrival times, and trip progress.
- **Streamline payment processing** – Support multiple secure payment options with instant confirmations.
- **Optimize the user interface** – Design a clean, intuitive, and responsive UI for both passengers and drivers.
- **Enhance performance and accessibility** – Ensure fast response times, mobile compatibility, and accessibility features for all users.

By addressing these usability concerns, SmartRide will improve efficiency, user satisfaction, and overall platform adoption.

7.3 Reliability

Driver Assignment Optimization

One of the major challenges SmartRide is facing is optimizing driver assignment to improve ride efficiency and reduce customer wait times.

Problem Statement

One of the major challenges SmartRide is facing is optimizing driver assignment to

improve ride efficiency and reduce customer wait times. To enhance the reliability and scalability of the system, digitalization must be integrated into the driver assignment process with measurable improvements.

Problem Statement

The current system struggles with assigning drivers efficiently, particularly during peak hours when demand is high. The lack of an effective data-driven matching algorithm results in several key issues:

- **Delayed ride matching** – Customers experience long wait times, averaging 8-12 minutes during peak hours due to inefficient driver allocation.
- **Unbalanced supply and demand** – Some areas receive 40% more ride requests than available drivers, while other areas have idle drivers waiting for 15+ minutes without trips.
- **Inefficient driver dispatching** – Drivers travel an average of 3-5 km to pick up passengers, increasing wait times and reducing efficiency.

Impact on the System

These inefficiencies lead to several negative consequences:

- **Poor customer experience** – High wait times result in a 25% ride cancellation rate, lowering customer retention.
- **Reduced driver earnings** – Drivers spend 30% of their time waiting or traveling without passengers, lowering their daily earnings.
- **Operational challenges for SmartRide** – Inefficient ride allocation contributes to 15-20% system downtime during peak usage.

Digitalized Solutions for Optimization

To address these issues, SmartRide must leverage digitalization to enhance the accuracy and efficiency of driver assignment with quantifiable improvements:

1. AI-Powered Driver Allocation Algorithms

- Utilize machine learning-based algorithms to reduce ride-matching time from 8-12 minutes to under 2 minutes, increasing efficiency by 80%.
- Implement automated decision-making to ensure the fastest and most optimized ride assignment with a 95% success rate in first-time matching.

2. Proximity-Based Driver Prioritization

- Assign the **nearest available driver** to a ride request, reducing the average pick-up distance from 3-5 km to under 2 km, cutting wait times by 50%.
- Optimize routes dynamically based on real-time GPS and traffic data, lowering estimated arrival time (ETA) by 30%.

3. Demand Prediction and Load Balancing

- Use big data analytics to predict high-demand areas with 85% accuracy, allowing SmartRide to pre-position drivers effectively.
- Dynamically adjust driver distribution, ensuring at least a 90% ride fulfillment rate in high-demand areas, reducing ride cancellations by 40%.

4. Automated Driver Guidance System

- Develop a real-time notification system that directs at least 60% of idle drivers to high-demand areas before peak hours.
- Implement an AI-driven incentive system to encourage driver relocation, increasing driver earnings by 15-20% per day.

By integrating digitalization and automation, SmartRide can significantly improve driver assignment accuracy, reduce customer wait times by up to 70%, increase driver earnings, and enhance overall operational efficiency. This transformation will ensure a more reliable, scalable, and data-driven ride-sharing platform, reducing inefficiencies and boosting customer satisfaction.

7.4 Performance

To effectively serve a large number of customers, the SmartRide system must implement key enhancements to ensure smooth operations, minimize delays, and maintain high service availability.

1. Scalable Infrastructure

The system must utilize a cloud-based architecture that can automatically scale resources based on demand. For example, during peak hours, server capacity should automatically increase by 50-100% to handle the surge in ride requests. Cloud providers like AWS, Google Cloud, or Microsoft Azure offer auto-scaling features to ensure system stability under high loads.

2. High-Performance Ride-Matching Algorithm

The driver assignment process must be optimized to handle thousands of ride requests per minute. Implementing a real-time ride-matching algorithm with a response time of under 500 milliseconds will significantly reduce customer wait times. Research shows that reducing response time below 1 second can increase customer retention by 25%.

3. Load Balancing for Optimal Performance

To prevent system overload, load balancing techniques should be used to distribute incoming ride requests across multiple servers. This can reduce downtime by 40% and ensure that the platform remains responsive even with 100,000+ concurrent users.

4. Predictive Demand Management

By leveraging AI-driven demand forecasting, SmartRide can analyze historical ride

data to predict peak demand periods. For instance, if data shows that demand increases by 80% between 5:00 PM - 8:00 PM, the system can proactively allocate more drivers to high-traffic areas, reducing ride cancellations by 30%.

By implementing these strategies, SmartRide can enhance system scalability, reduce ride wait times, and improve overall customer satisfaction, ensuring a seamless experience even during peak demand periods.

7.5 Portability

Allowing the SmartRide platform to be accessible across multiple platforms ensures that users are not restricted by their devices. This flexibility enhances user convenience, enabling passengers to book rides seamlessly and allowing drivers to manage ride requests efficiently from various devices. Additionally, administrative staff can monitor ride statistics and manage user accounts from any platform, ensuring smooth operations.

To meet this quality requirement, the system must:

- Be accessible on mobile platforms (Android & iOS) to allow passengers and drivers to use the service on the go.
- Be available on desktop and laptop computers for customers who prefer booking rides via a web interface and for administrators to manage operations efficiently.

This ensures that SmartRide remains flexible, user-friendly, and scalable across different devices, leading to increased user engagement and operational efficiency.

8, Other Requirements

8.1 Product level requirements

These requirements specify that the ORSP app shall be able to:

- Work all the time, so people can book and take ride whenever they need
- The app should run smoothly, even when a lot of people are using it at once
- Keeps personal details and payment information safe
- Must follow local laws about transportation and data protection
- Should be simple to use
- Supports different payments

8.2 Design level requirements

These ensure the app runs smoothly and can be improved over time:

- Allows new features without breaking the system.
- Quickly saves and retrieves information.
- Stable and avoids crashes.
- Notifies users and fixes issues.
- Consistent on Android and iOS.

- Protects ride and payment history.
- Separate section for handling complaints and operations.

9, Validation of Requirements

To ensure that the system requirements are accurate and complete, the SmartRide team reviewed them with key stakeholders, including passengers, drivers, and administrators. The goal was to confirm that each task meets real-world needs and that the system can handle different situations effectively.

CRUD Check

Task / Entity	Passenger	Driver	Ride	Vehicle	Payment	Location	Review & Rating	Ride Request	Admin
Create an Account	C	C							
Login	R	R							R
Book a Ride	C, R, U		C			C			
Track the Ride	R	R	R			R			
Process Payment	R	R	R		C, U				
Rate & Review	C, R, U	C					C		
Manage Platform								U, D	C, U, D
Administer & Report System			R	R	R	R		R	R
Book Additional Services	C, R		C		C		C		

10, Possible Solutions

Employee-Focused System for SmartRide

This is a traditional approach to managing and operating an online ride-sharing service. The system interactions are primarily handled by employees and drivers, while customers have little or no direct access to internal operations. This system is

designed to support SmartRide's service management, optimize operational workflows, and ensure smooth service delivery.

10.1 System for Drivers and Employees

When drivers join the platform, they are granted access to a dedicated driver application available on mobile devices (smartphones/tablets) or through tablets at SmartRide dispatch centers.

a. Driver Account & Profile Management

- Drivers can log in to manage their personal profiles.
- The system allows drivers to update vehicle documents, driver's licenses, and availability status.
- Driver information is automatically verified against internal databases.

b. Ride Assignment & Management

- Drivers can view a list of ride requests near their location.
- Upon accepting a ride, they receive customer details, pickup/drop-off locations, and estimated fares.
- The app provides GPS navigation to guide drivers through the optimal route.
- Once the ride is completed, the system automatically updates the trip status and the driver's earnings.

c. Payment Processing & Earnings Management

- Drivers can track their earnings daily, weekly, or monthly.
- Payments are automatically processed and transferred to the driver's account at scheduled intervals.
- Drivers can generate earning reports and view detailed transaction history.

d. Support & Complaint Handling

- The system includes a support center, allowing drivers to report issues such as ride cancellations, delayed payments, or passenger disputes.
- SmartRide employees can review and respond to complaints through the internal system.

10.2 System for SmartRide Employees

SmartRide employees use an internal management system to oversee operations and coordinate service delivery. This system is accessible via desktop computers or dedicated management mobile applications.

a. System Management & Ride Coordination

- Employees can monitor the platform's operational status, including active drivers, ongoing rides, and pending ride requests.
- During peak hours, employees can manually assign drivers to high-demand areas to improve efficiency.
- The system features real-time GPS tracking, enabling management to oversee

driver locations and adjust dispatching as needed.

b. Customer & Driver Management

- Employees can access detailed information about customers and drivers, including ride history, account status, and ratings.
- In cases of policy violations or complaints, administrators can suspend accounts or issue warnings.

c. Reporting & Data Analytics

- The system generates statistical reports on ride volume, revenue, completion rates, and customer satisfaction.
- Reports can be customized to meet management needs (daily, weekly, or monthly).
- The system can also forecast ride demand trends based on historical data, helping optimize operations.

d. Complaint Resolution & Customer Support

- Employees manage and resolve complaints from both customers and drivers.
- The system includes a ticket-based support system, allowing for structured issue resolution.
- Employees can respond directly through the system or contact the concerned party for further resolution.

10.3 AI-Powered Driver & Ride Matching

- **Smart Matching Algorithm:** Implement AI & Machine Learning to automatically pair passengers with the most suitable drivers based on distance, traffic conditions, and ride history.
- **Demand Prediction** – Predict high-demand areas at specific times to dispatch drivers efficiently, reducing passenger wait times.
- **Route Optimization** – Use AI-driven GPS navigation to recommend the best routes, saving fuel and time for drivers.

10.4 Cloud-Based Infrastructure

- **Deploy SmartRide on Cloud Services (AWS, GCP, Azure)** – Enables flexible scaling, reduces server overload, and ensures high performance even during peak usage.
- **Automated Scaling** – Utilize auto-scaling features to dynamically adjust resources based on real-time demand, preventing downtime.
- **Cloud-Based Data Storage** – Store ride, user, and transaction data in cloud databases (e.g., Firebase, AWS RDS, Google Cloud Firestore) for reliability and fast data retrieval.