Uber :-,

I can provide you with a general outline of a System Requirements Specification (SRS) for a ride-sharing platform like Uber. Keep in mind that the actual requirements for Uber may have evolved .

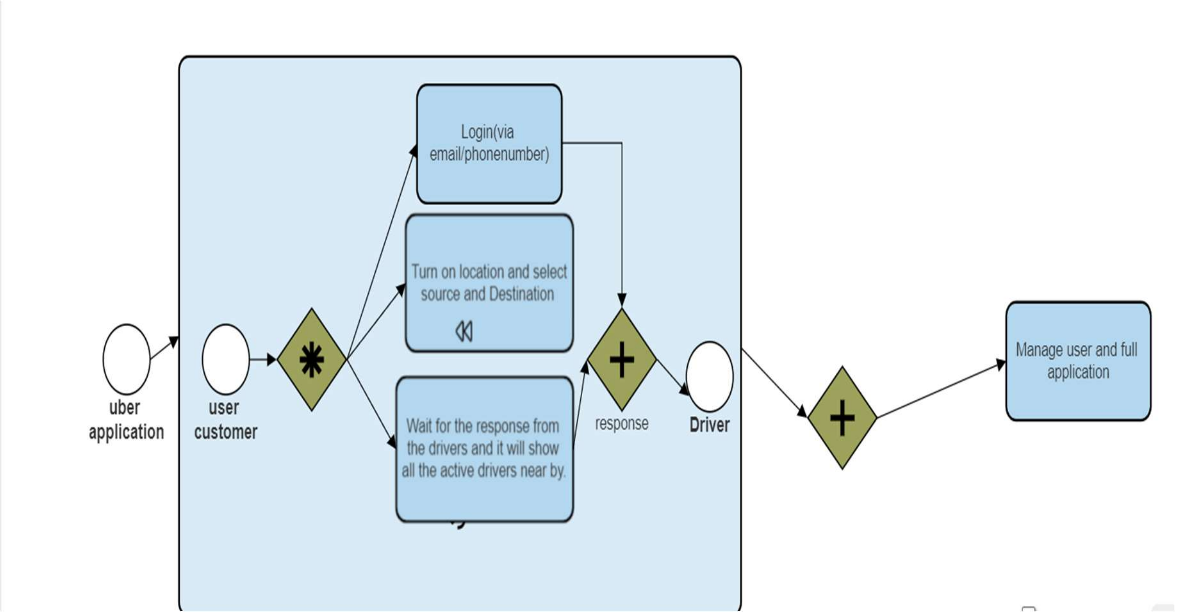
1. Introduction:
    - Purpose: Define the purpose of the system and its intended audience.
    - Scope: Describe the boundaries of the system and the features it will include.
    - Definitions, acronyms, and abbreviations: Provide a list of terms used throughout the document.
2. Overall Description:
    - Product perspective: Describe how the system fits into the larger ecosystem and any related systems.
    - Product functions: Provide an overview of the main functions the system should perform (e.g., rider requests, driver matching, fare calculation, etc.).
    - User characteristics: Describe the target users and their roles (riders, drivers, administrators, etc.).
    - Constraints: Identify any limitations or restrictions that may affect the development or implementation of the system.
    - Assumptions and dependencies: List any assumptions made during the requirements gathering process.
3. Specific Requirements:
    - Functional requirements: Detail the specific functions the system should perform, including use cases and scenarios.
    - Non-functional requirements: Specify qualities the system should have, such as performance, security, usability, scalability, etc.
    - External interfaces: Describe the interactions between the system and external entities like payment gateways, map APIs, etc.
    - Data requirements: Outline the types of data the system will store and manage, including user profiles, ride history, payment details, etc.
4. System Features and Use Cases:
    - Provide a detailed breakdown of the system's features, explaining how users interact with them through use cases.
5. Performance Requirements:
    - Define the expected performance levels, such as response times, maximum concurrent users, and data processing capabilities.
6. Security Requirements:
    - Describe the security measures to protect user data, prevent unauthorized access, and secure communications.
7. Quality and Testing:
    - Outline the testing requirements, including test cases, testing environments, and acceptance criteria.
8. Legal and Compliance:
    - Address any legal and regulatory requirements the system must adhere to, such as data privacy laws and payment regulations.
9. Documentation:

- Describe the documentation required for the system, including user manuals, technical guides, and developer documentation.

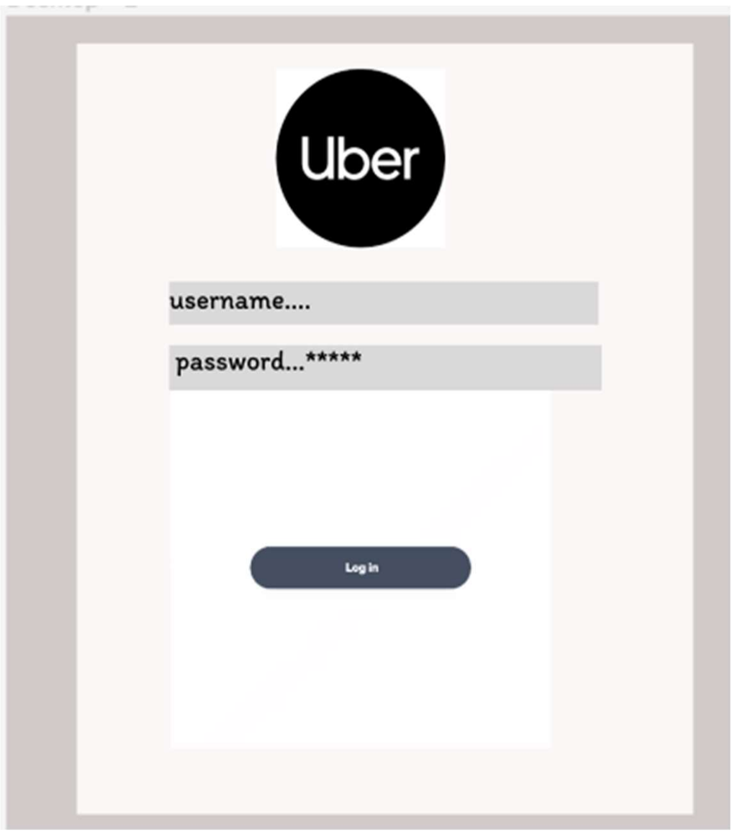
10. Maintenance and Support:
    - Specify the requirements for ongoing system maintenance, updates, and technical support.

It's important to note that creating a comprehensive and accurate SRS requires collaboration between business stakeholders, software architects, developers, and other relevant parties involved in the project. The SRS serves as a foundation for the development process, ensuring a clear understanding of what needs to be built and what criteria the system must meet.

Work flow

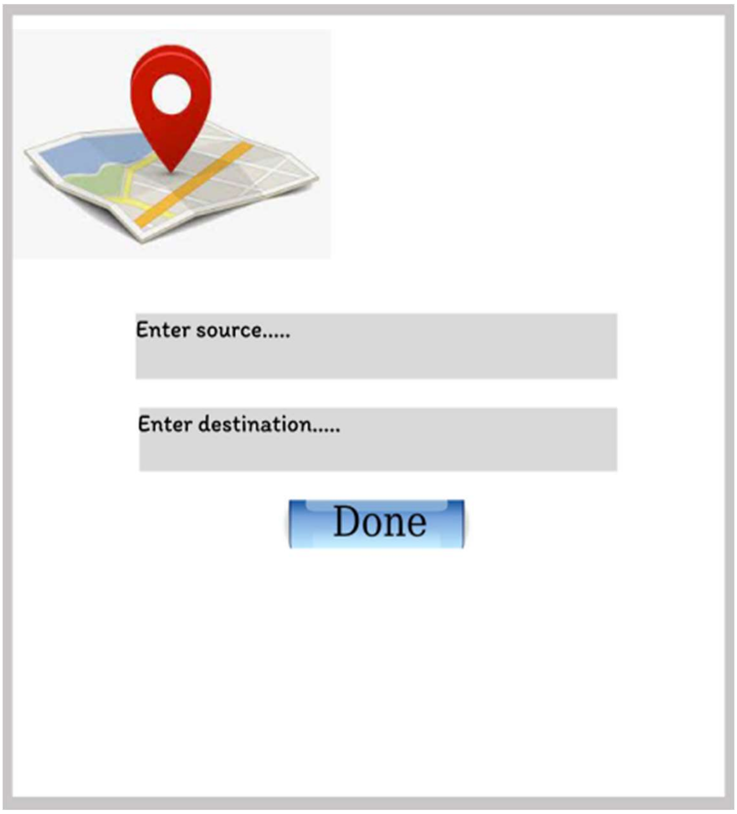

uber application
user customer
Login(via email/phonenumber)
Turn on location and select source and Destination
Wait for the response from the drivers and it will show all the active drivers near by.
response
Driver
Manage user and full application

Wire frame:-



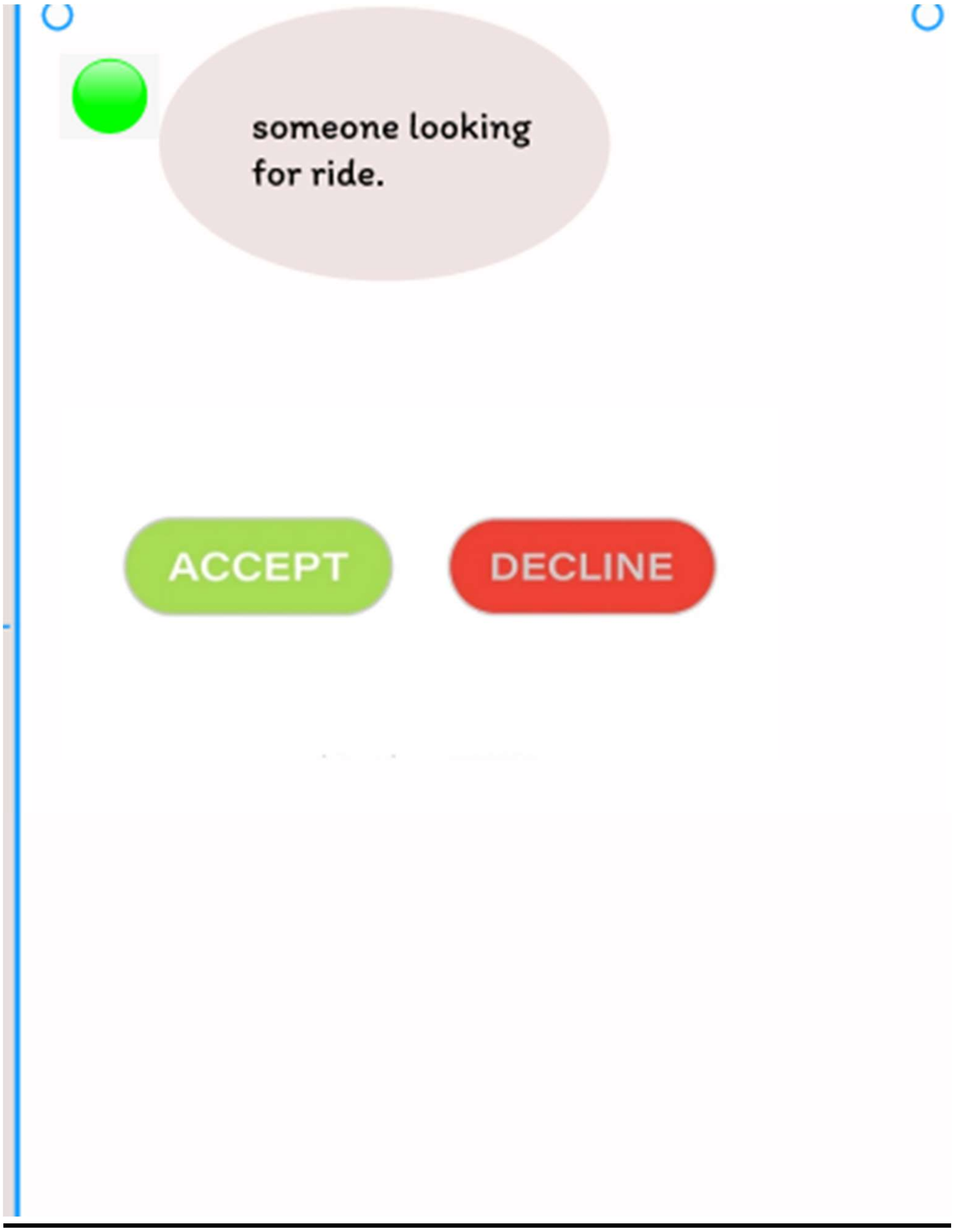
username....

password...*****

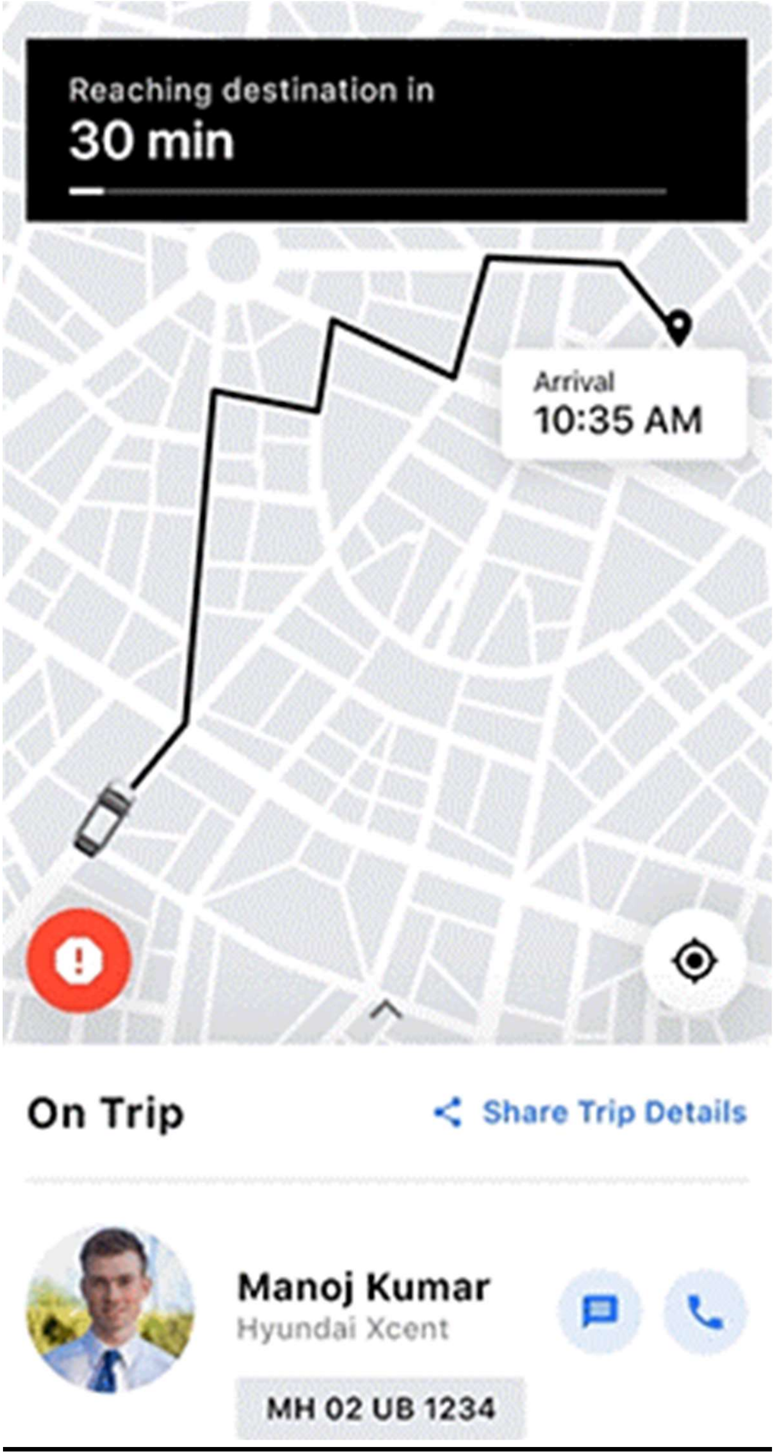


Enter source.....

Enter destination.....



someone looking for ride.



Reaching destination in

30 min

Arrival

10:35 AM

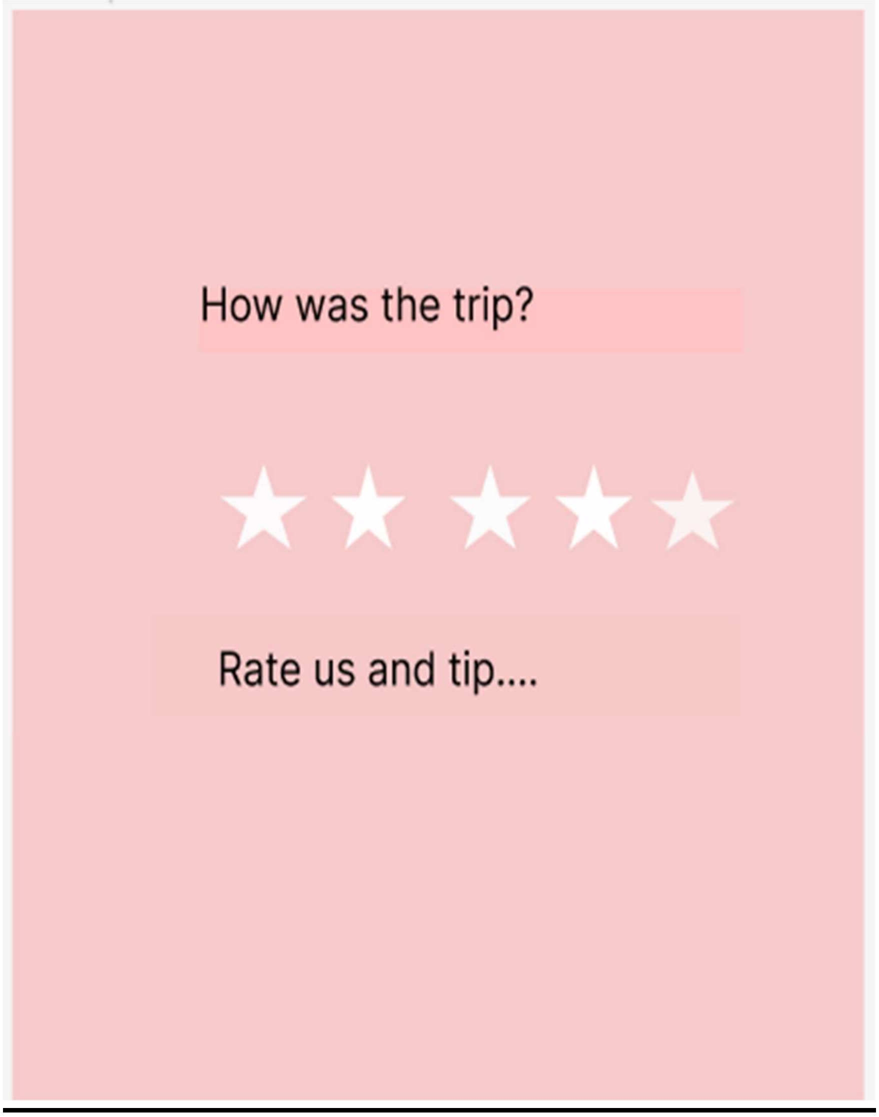
On Trip

Share Trip Details

Manoj Kumar

Hyundai Xcent

MH 02 UB 1234



How was the trip?

★ ★ ★ ★ ★

Rate us and tip....