System Name:

System steps

* Define the Scope and requirements
* Choose the tech stack
* Plan the architecture
* Plan an MVP (Minimum Viable Product)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Define the Scope and requirements
2. Type of system to create. (Web app/Mobile app)
3. Who are the users. (Passengers, drivers and admins)
4. System features:
   * + Passengers: Book rides, fare estimation, tracking drivers, payments.
     + Drivers: Accepting rides, Navigation and trip history
     + Admins: Managing drivers, tracking vehicles, reports.

Also need real-time tracking (google map API, OpenStreetMap).

1. Choose the tech stack
2. **Frontend**:
3. **Database**: PostgreSQL, MongoDB, MySQLite, or SQLite
4. **Backend**: Node.js, Django, Flask, or Laravel.
5. **APIs**: Google Maps API for routing, Firebase for authentication, Stripe/PayPal for payments.
6. Plan the architecture
7. Design a database scheme (tables for users, bookings, payments).
8. Plan APIs for ride requests, user authentication, payments.

1. Plan an MVP (Minimum Viable Product)
2. Start with a **simple version**: user registration, ride request, driver matching.
3. Build **backend services** first.
4. Develop a **basic frontend** (mobile or web interface).