

MoveInSync

Ride-Sharing-Platform

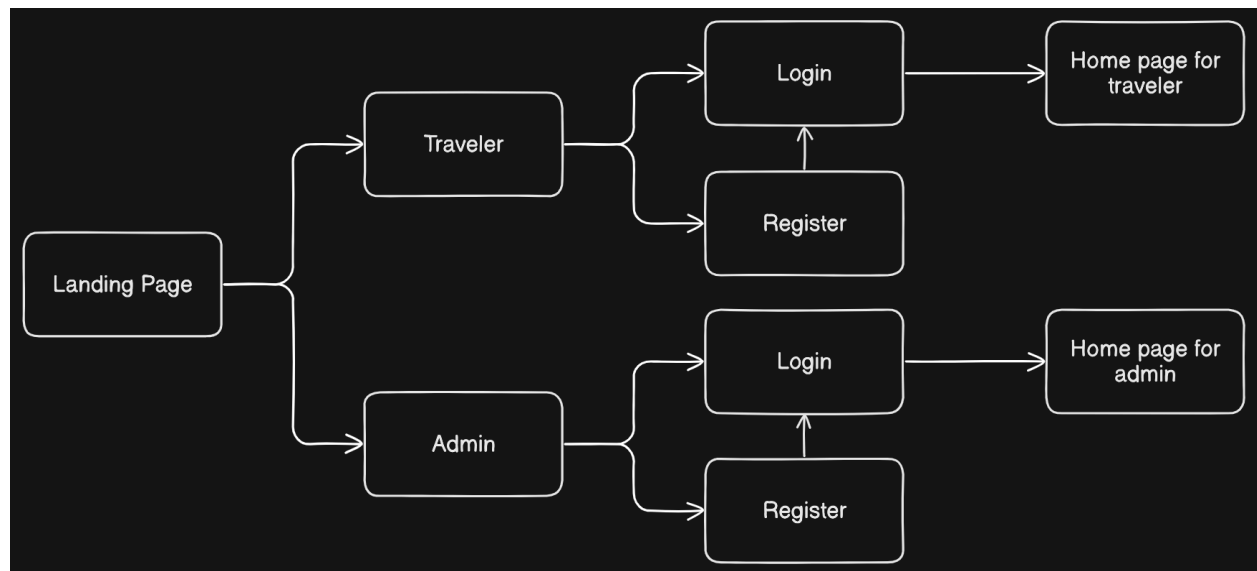
Workflow

User Registration:

- Users register with their username, email, phone number and password.

Admin Registration:

- Admin register with their username, email, phone number and password.

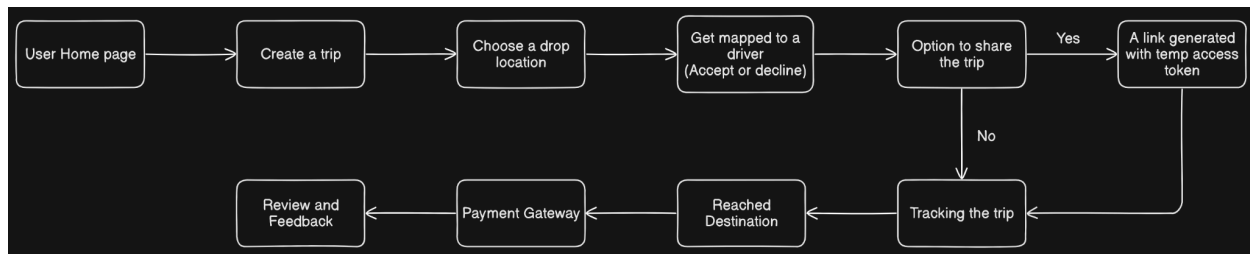


Once Logged In:

Traveler:

- Travelers can initiate a trip by entering ride details.
- A unique shareable link is generated for the trip. (The shareable link expires once the trip is completed)
- Travelers can view a log of all rides they've shared, including details like trip status and sharing history.

Flow for a traveller to book and share the trip:

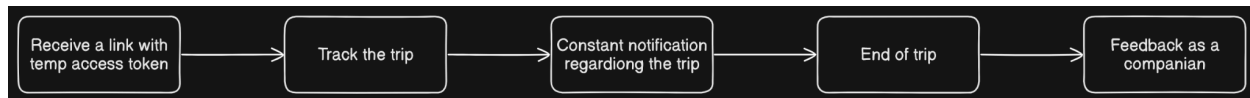


Admin:

- Admin can view all rides shared by users, including details of the trips and status.
- Admin can access overall feedback from travellers and companions about their experiences.

Companion:

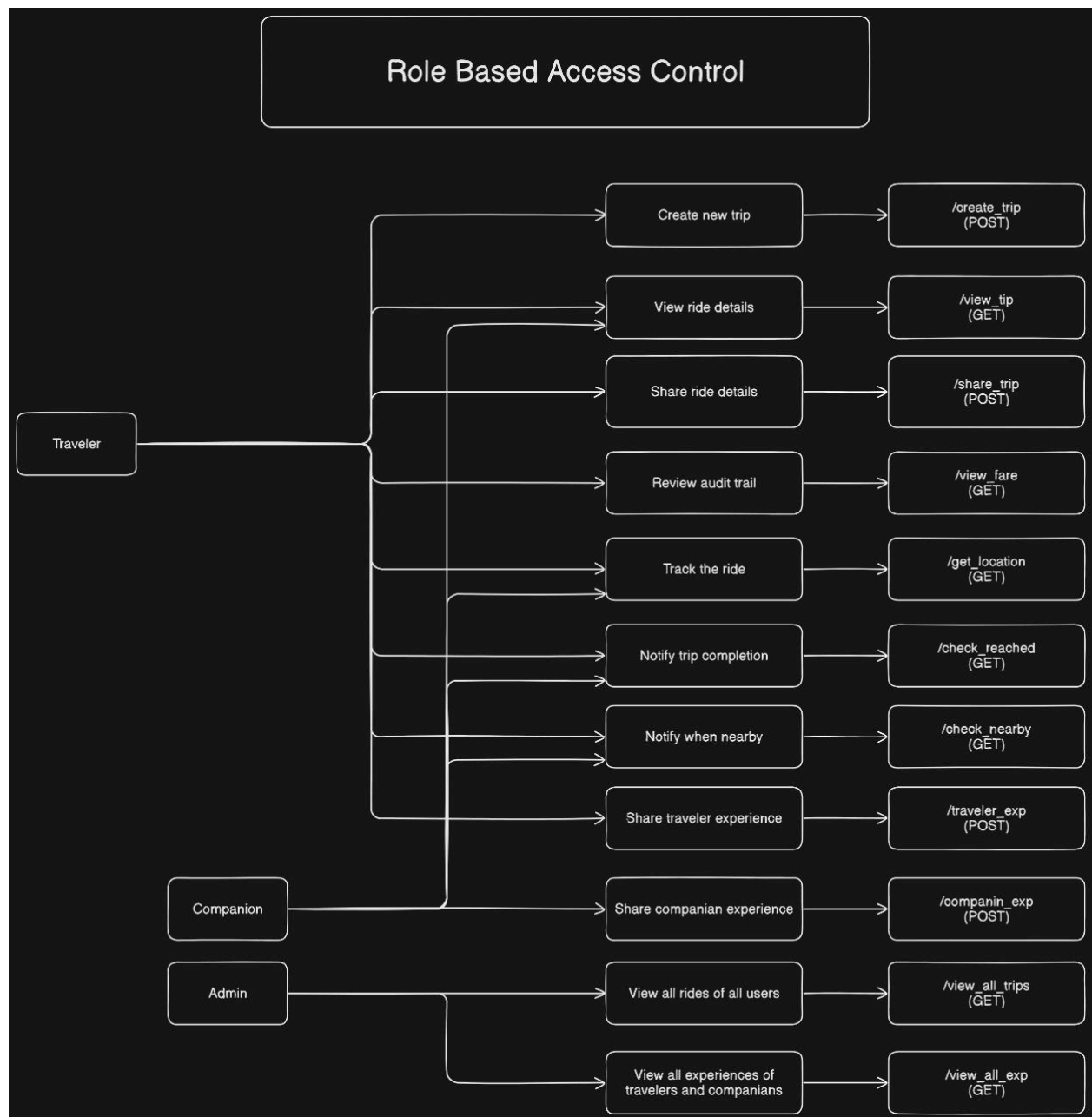
- A companion can track the ride's location in real-time using the shared link.
- Companion receives a notification when the trip is complete.
- Companion receives a notification when the cab enters the geofence of the traveller's drop-off location.
- The access token expires once the trip ends.



Key Functionality:

1. **Trip Creation:** Ability to initiate a trip and enter ride details (e.g., TripId, driver information, destination).
2. **Share Trip Details:** Generate and share a unique link with companions via WhatsApp or SMS.
3. **Link Expiry:** Automatically expire the shareable link after the trip is completed.
4. **Payment Gateway:** Integrates with a payment gateway for online payments.
5. **Ride History:** traveller can view their ride history
6. **Authentication:** Includes various authentication mechanisms such as email verification, emergency contacts, and two-step authentication.
7. **Emergency Services:** Features an SOS button for emergencies

Role-Based Access Control:



Authentication:

- **Password Hashing:** Store passwords securely using strong hashing algorithms (e.g., bcrypt, Argon2) to protect against unauthorized access.
- Multi-Factor Authentication
- OAuth and JWT
- **Secure Cookies:** Use secure and HTTP-only cookies to store session tokens.
- **Token Expiry:** Implement token expiration and refresh mechanisms to reduce the risk of token misuse.

Handling System Failure Cases:

- **Regular Backups:** Schedule regular backups of data and configuration to prevent data loss.
- **Redundancy:** Implement redundancy in critical components (e.g., load balancers) to ensure continued operation during failures.
- **Error Logging:** Implement comprehensive logging to track errors and facilitate troubleshooting.

System Monitoring:

- **Real-Time Metrics:** Create dashboards to visualize system metrics such as response times, error rates, and resource usage.
- **Alerts:** Set up alerts for critical issues to notify administrators of potential problems.

Caching:

- Use content delivery networks (CDNs) to cache static assets, such as images and scripts, for faster content delivery.
- **Database Caching:** Implement caching for database queries to reduce read times.
- **Least Recently Used (LRU):** Use LRU or other eviction policies to manage cache size and resource utilization.