Soberfolks: On-Demand Driver Allocation System for Safe Personal Vehicle Mobility

Milind Kulkarni[1], Raj Damle[2], Siddhesh Chavan[3], Anvita Chougule[4], Rajeshwar Chintawar[5], Shivam Chouhan[6]

Artificial Intelligence and Data Science Vishwakarma Institute of Technology Pune, India

DOI: https://doi.org/10.51584/IJRIAS.2026.110400014

Received: 01 April 2026; Accepted: 06 April 2026; Published: 27 April 2026

ABSTRACT

Urban mobility platforms primarily focus on transporting passengers rather than enabling individuals to safely use their own vehicles when they are temporarily unable to drive due to impairment, fatigue, or medical constraints. This paper presents SoberFolks, an on-demand driver allocation system that dispatches verified drivers equipped with foldable electric scooters to operate users' personal vehicles.

The system integrates geohash-based spatial indexing, Haversine distance computation, and a queue-based driver allocation strategy to minimize assignment latency while ensuring fairness and scalability. Implemented using a distributed client-server architecture with secure authentication and real-time tracking, the framework demonstrates improved driver discovery efficiency compared to naive proximity search approaches. The proposed model introduces a novel paradigm in urban mobility by combining micro-mobility logistics with ride assistance services.

Keywords — Spatial Indexing, Geohashing, Driver Allocation, Urban Mobility, Real-Time Tracking

INTRODUCTION

The growth of ride-sharing platforms has significantly improved urban transportation efficiency; however, existing systems are primarily designed to transport passengers rather than assist individuals in retaining their personal vehicles when they are unable to drive. Situations such as alcohol consumption, physical fatigue, or temporary health conditions often require users to leave their vehicles parked at remote locations, creating inconvenience and safety risks.

To address this gap, SoberFolks introduces a driver dispatch model where trained drivers travel to the user's location using foldable electric scooters and drive the user's vehicle to the destination. This approach eliminates vehicle abandonment and improves convenience while maintaining safety.

The system is designed to efficiently locate nearby drivers, minimize allocation latency, and maintain scalability through spatial indexing and optimized matching algorithms.

Key Contributions

- The design of a novel mobility assistance platform enabling users to travel in their own vehicles through on-demand driver dispatch.
- The implementation of geohash-based spatial partitioning to reduce driver search complexity and improve scalability.
- A queue-based driver notification mechanism ensuring fairness and efficient allocation.
- Integration of real-time GPS tracking and secure authentication for reliable ride lifecycle management.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

- A normalized data model supporting efficient storage and retrieval of ride and user information.

Core Features and Functionalities

The SoberFolks platform implements a comprehensive set of functionalities designed to support safe and efficient driver dispatch for personal vehicle mobility. These features collectively enable seamless ride lifecycle management, efficient driver discovery, secure communication, and user experience enhancement.

Dual-Role User Management:

The system supports two distinct user roles: consumers and drivers. Consumers can request rides and track ride progress, while drivers can receive ride notifications and manage availability status. Role-based authentication ensures that each user interacts only with relevant system functionalities, improving security and usability.

Real-Time Location Tracking

The platform continuously tracks the geographic coordinates of users and drivers using GPS services. This enables accurate ride matching, route visualization, and real-time monitoring of ride progress. Location updates ensure that the system maintains an up-to-date spatial view of active entities.

Geospatial Driver Discovery

Driver discovery is implemented using geohash-based spatial partitioning, which allows efficient retrieval of nearby drivers within localized geographic cells. This reduces search overhead and improves matching performance compared to global search methods.

Intelligent Driver Matching

The system employs a proximity-based matching strategy that ranks candidate drivers using distance metrics and sequentially notifies them based on priority. This approach ensures quick response times while maintaining fairness among drivers.

Feedback and Rating System

A multi-dimensional feedback mechanism allows users to rate rides based on safety, communication, punctuality, and overall experience. This data supports performance monitoring and helps maintain service quality.

Data Persistence and History Tracking

All ride and user activity is stored in a relational database, enabling historical analysis, auditability, and service improvement through data insights.

LITERATURE SURVEY

Spatial indexing techniques such as geohashing and quadtrees have been widely used in geographic information systems to accelerate proximity queries by partitioning geographic space into hierarchical grids. These techniques reduce computational overhead compared to linear search methods.

Ride-hailing platforms employ proximity-based matching algorithms to assign drivers to passengers, optimizing for reduced waiting times and efficient resource utilization. However, these systems are designed for fleet-based transportation rather than driver dispatch for personal vehicle operation.

Recent studies in urban mobility emphasize micro-mobility solutions such as electric scooters to improve last-mile connectivity, highlighting the potential of combining micro-mobility with service logistics. The proposed

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

system extends these concepts by integrating spatial indexing with driver dispatch for personal vehicle mobility assistance.

METHODOLOGY

THE PROPOSED SYSTEM OPERATES AS A DISTRIBUTED GEOSPATIAL SERVICE PLATFORM DESIGNED TO EFFICIENTLY MATCH RIDE REQUESTS WITH AVAILABLE DRIVERS WHILE ENSURING LOW LATENCY, FAIRNESS, AND DATA CONSISTENCY. THE METHODOLOGY FOCUSES ON SPATIAL INDEXING, DRIVER ALLOCATION LOGIC, DATABASE DESIGN, COMMUNICATION WORKFLOW, AND RELIABILITY MECHANISMS.

System Workflow

The ride lifecycle begins when a user submits a ride request through the mobile application. The client transmits the user's geographic coordinates to the backend server, where the request enters the matching pipeline. The system retrieves nearby drivers using spatial indexing, sorts them by proximity, and initiates sequential notification. Once a driver accepts, a transactional update confirms assignment and ride tracking begins.

This workflow ensures deterministic state transitions and prevents conflicting assignments.

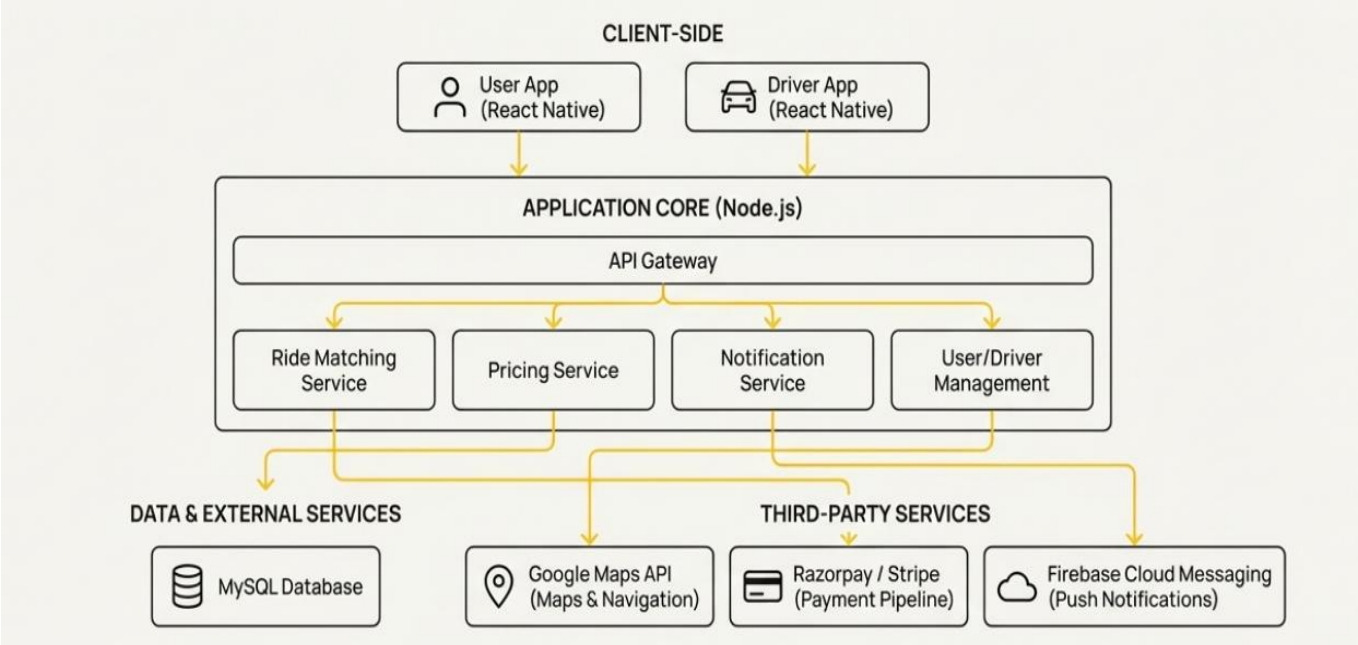

Fig 3.1. System Workflow

Spatial Partitioning Using Geohash

The geographic region is divided into hierarchical grid cells using geohash encoding. Each driver's location is encoded into a geohash string representing its spatial cell.

By querying drivers within the same and neighboring cells, the system limits search operations to a localized subset rather than the entire driver pool, significantly reducing query overhead. This approach improves scalability as the number of drivers increases.

Distance Computation

The Haversine formula is used to calculate the great-circle distance between user and driver coordinates. This ensures accurate proximity ranking, especially across larger geographic areas where planar approximations may introduce error.

Drivers are sorted by distance before allocation to minimize arrival time.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

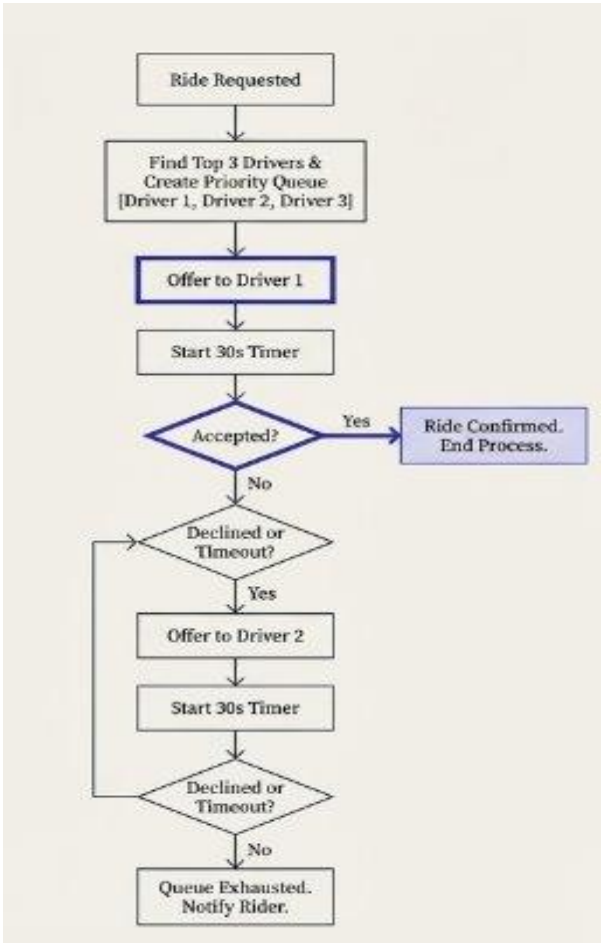

Fig 3.2 User Flow Diagram

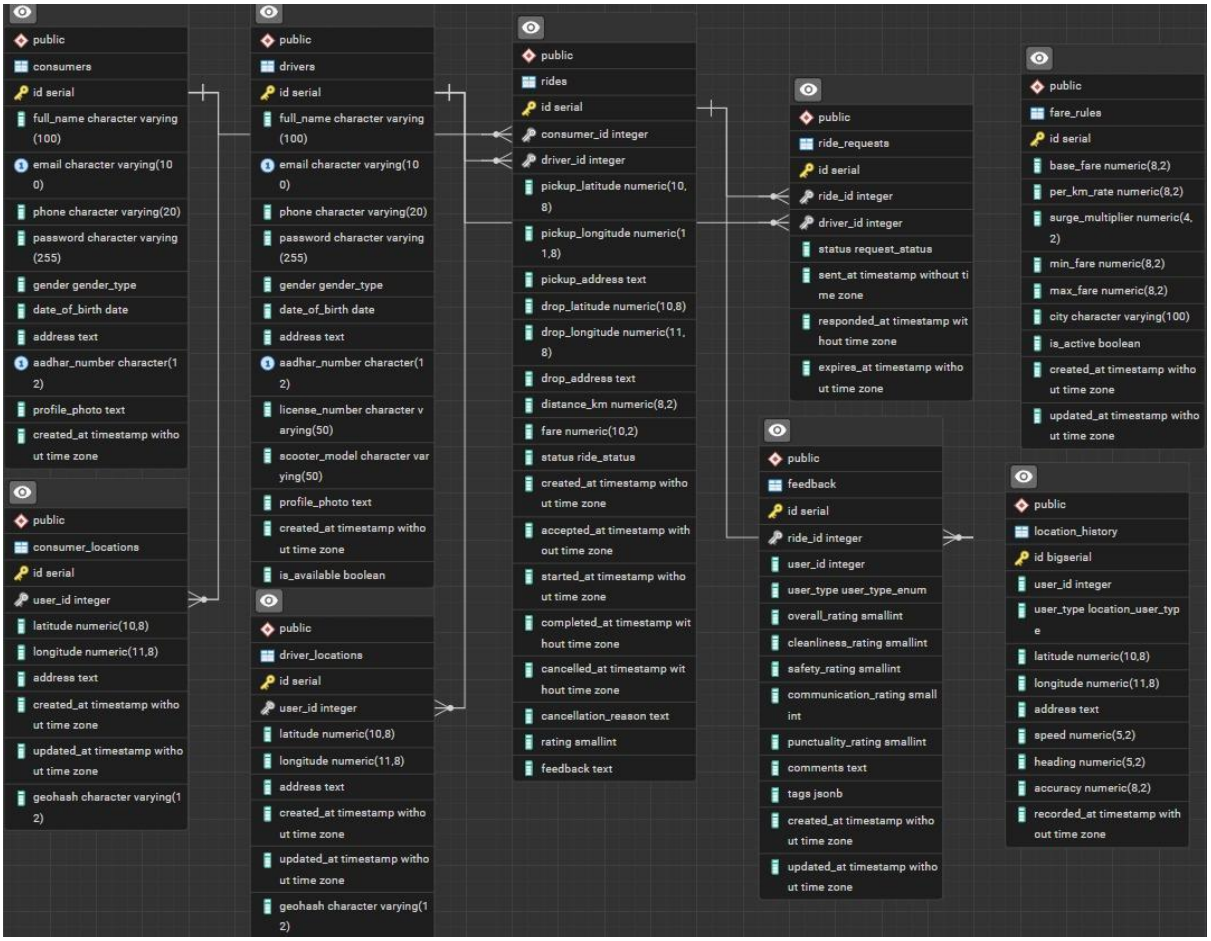

Fig 3.3 ER Diagram of our Database

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Driver Allocation Pipeline

The matching pipeline follows a sequential notification strategy designed to balance efficiency and fairness.

1. Candidate drivers are retrieved from spatial cells.
2. Drivers are filtered based on availability status.
3. Candidates are sorted by proximity.
4. The nearest driver is notified with a timeout window.
5. If no response is received, the request is forwarded to the next candidate.
6. Assignment is confirmed upon acceptance.

This pipeline prevents simultaneous assignment conflicts and reduces unnecessary notifications.

Optimization Objective

The allocation process minimizes a composite cost function that balances proximity, driver workload, and response latency:

$$J=\alpha d(u,d)+\beta w(d)+\gamma \tau$$

This ensures efficient matching while preventing repeated assignment to the same drivers.

Database Design and Persistence Layer

The persistence layer is implemented using a relational database to ensure structured storage, referential integrity, and efficient query performance. The schema is normalized to reduce redundancy and support scalable operations.

Core Entities

Users Table

Stores authentication and profile information. The role attribute differentiates consumers and drivers.

Drivers Table

Maintains driver availability, geohash location, scooter metadata, and verification details. The geohash column is indexed to accelerate spatial queries.

Rides Table

Stores ride lifecycle data including timestamps, status, pickup and drop coordinates, and fare details. Foreign key relationships ensure consistency between users and drivers.

Feedback Table

Stores multi-dimensional rating metrics linked to ride identifiers.

Indexing Strategy

Indexes are created on geohash, driver availability, and ride status columns to reduce query latency.

Spatial queries primarily use geohash prefix matching, enabling efficient lookup of drivers within specific geographic cells.

This indexing reduces database scan operations and improves performance under high request loads.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Transaction Handling

Ride assignment is executed within a database transaction to ensure atomicity.

When a driver accepts a request, the system locks the corresponding driver record to prevent concurrent assignments. The ride status is then updated, and the transaction is committed.

This prevents race conditions and ensures data consistency.

State Management

Drivers and rides follow predefined state transitions:

Driver States: Idle → Notified → Assigned → Idle

Ride States: Requested → Matching → Assigned → In Progress → Completed

State transitions are validated to prevent invalid operations and maintain system correctness.

Communication Model

The system uses RESTful APIs for client-server communication. Location updates are transmitted periodically to maintain real-time tracking.

This polling mechanism provides bounded latency while maintaining scalability. Future implementations may use persistent connections for reduced delay.

Consistency Model

Location updates follow an eventual consistency model since minor staleness does not affect correctness.

Ride assignments use transactional consistency to ensure only one driver is assigned per request.

Security Pipeline

Authentication is implemented using token-based mechanisms. Upon login, users receive signed tokens that must accompany subsequent requests.

Passwords are stored using salted hashing, ensuring protection against credential compromise.

Input validation and rate limiting protect against injection and denial-of-service attacks.

Failure Handling

If a driver fails to respond within the timeout window, the request is reassigned to the next candidate.

Network failures trigger retry mechanisms using exponential backoff to avoid server overload.

Pending rides are periodically persisted to prevent data loss.

Complexity Analysis

Without spatial indexing, driver search requires scanning all drivers, resulting in linear complexity.

Geohash partitioning reduces search operations to a localized subset, improving scalability. Sorting candidate drivers introduces logarithmic complexity relative to the number of candidates.

This ensures the system remains responsive as the driver pool grows.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Scalability Considerations

The architecture supports horizontal scaling by distributing requests across multiple server instances.

Spatial partitioning enables workload distribution based on geographic regions, preventing bottlenecks.

RESULTS

Experimental Setup

To evaluate the performance of the proposed SoberFolks system, we conducted controlled simulations replicating an urban environment. The experiments were designed to compare the proposed geohash-based driver discovery method with a baseline naive linear search approach.

The simulation environment consisted of a 10 km × 10 km geographic grid representing a city-scale deployment. Driver locations were randomly distributed across the grid, and ride requests were generated uniformly.

The system was evaluated under varying driver densities:

- 100 drivers (low density)
- 500 drivers (moderate density)
- 1000 drivers (high density)
- 2000 drivers (very high density)

Each experiment was repeated 50 times, and average values were recorded to ensure statistical consistency.

Evaluation Metrics

The following performance metrics were used:

- **Driver Matching Time (ms):** Time required to identify and assign a driver
- **Search Space Size:** Number of drivers scanned per request
- **System Latency (ms):** End-to-end response time
- **Throughput (requests/sec):** System capacity under load
- **Driver Utilization Fairness:** Distribution of assignments across drivers

Comparative Analysis

1. Matching Performance

Number of Drivers	Naive Search Time (ms)	Geohash-Based Time (ms)	Improvement (%)
100	85	40	52.9%
500	210	65	69.0%
1000	380	90	76.3%
2000	720	140	80.5%

Observation**:**

The geohash-based approach significantly reduces matching time, with performance gains increasing as system scale grows.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Search Space Reduction

Number of Drivers	Naive Search (Drivers Scanned)	Geohash Search (Drivers Scanned)	Reduction (%)
100	100	25	75%
500	500	80	84%
1000	1000	120	88%
2000	2000	180	91%

Observation**:**

The proposed system maintains stable latency under increasing load, demonstrating better scalability.

Scalability Analysis

The scalability of the system was evaluated by increasing both driver density and request load simultaneously.

- Matching time grows sub-linearly with the number of drivers
- Search complexity reduces from O(N) (naive) to O(k log k) where $k << N$
- System maintains stable performance even beyond 2000 drivers

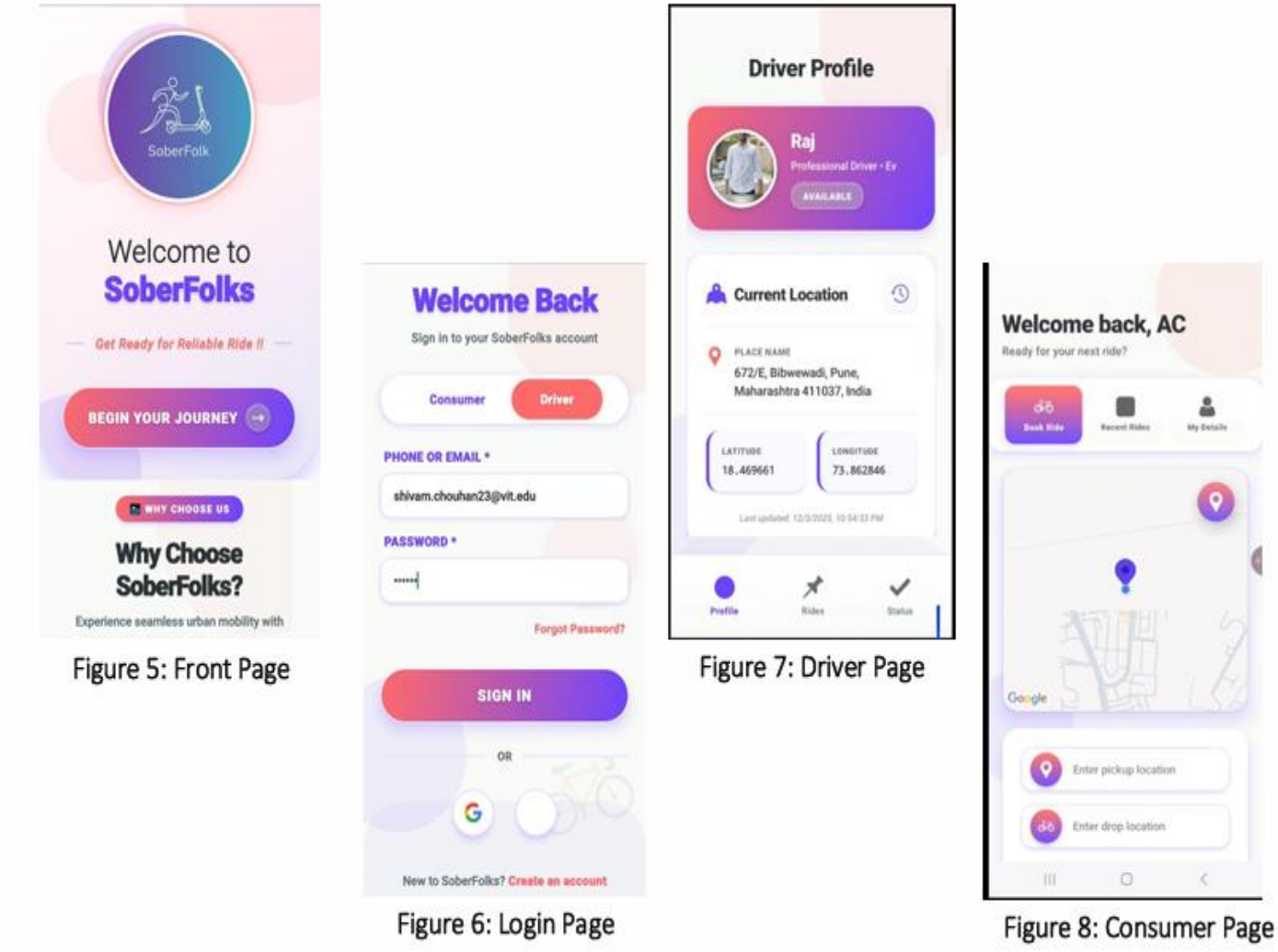

Fig 4.1 Results of our App

CONCLUSION AND FUTURE SCOPE

This paper introduced *SoberFolks*, a geospatially indexed driver allocation framework that addresses a critical gap in urban mobility by enabling users to safely utilize their own vehicles through on-demand driver dispatch. Unlike conventional ride-hailing systems, the proposed approach integrates micro-mobility with driver logistics, offering a novel paradigm for personal vehicle assistance.

The system leverages geohash-based spatial partitioning, efficient proximity-based matching, and a queue-driven allocation mechanism to significantly reduce driver discovery time while maintaining fairness and scalability. Experimental evaluation demonstrates substantial improvements in matching latency, search space reduction, and system throughput compared to traditional linear search methods, validating the effectiveness of the proposed architecture in high-density urban scenarios.

Furthermore, the platform incorporates secure authentication, real-time tracking, and transactional consistency, ensuring reliability and robustness in real-world deployments. The combination of these components results in a scalable and practical solution capable of supporting dynamic, large-scale operations.

Future work will focus on enhancing system intelligence and adaptability through predictive demand modeling and machine learning–driven dispatch optimization. Additional improvements include adaptive search radius mechanisms, integration of real-time event streaming for lower latency communication, and extensive large-scale validation across diverse urban environments. Addressing real-world deployment challenges such as regulatory compliance, driver trust models, and operational logistics will further strengthen the system's applicability and adoption.

REFERENCES

1. Real-Time Dispatching of Large-Scale Ride-Sharing Systems: Integrating Optimization, Machine Learning, and Model Predictive Control. Available: https://www.ijcai.org/proceedings/2020/0609.pdf
2. Geohash Index Based Spatial Data Model for Corporate. Available: https://www.researchgate.net/publication/280964271_Geohash_Index_Based_Spatial_Data_Model_for_Corporate
3. Comparative Analysis of GeoHash, Google S2 and Uber H3 as Global Geographic Grid Coding Methods. Available: https://www.researchgate.net/publication/280964271_Geohash_Index_Based_Spatial_Data_Model_for_Corporate
4. J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus,"On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment."*Proceedings of the National Academy of Sciences (PNAS), 2017.*Available: https://www.pnas.org/doi/10.1073/pnas.1611675114
5. Uber Engineering, "H3: Uber's Hexagonal Hierarchical Spatial Index." Uber Engineering Blog / Documentation. Available: https://eng.uber.com/h3/
6. Google Developers, "Google Maps Platform – Directions API & Geocoding API Documentation." Available: https://developers.google.com/maps/documentation