MICHIGAN STATE UNIVERSITY

Project Plan Presentation

Offline-Ready Mobile App for Delivery Optimization

The Capstone Experience

Team Magna MADO

Chetan Chigurupati
Adam Farkas
Mia Granata
Shrey Kohli
Shane Patrarungrong
Muhammad Shaikh

Department of Computer Science and Engineering
Michigan State University

Fall 2024

From Students…
…to Professionals

Project Sponsor Overview

- Magna is a global leader in automotive technology, with over 174,000 employees in 28 countries
- Its diverse portfolio includes automotive parts, powertrain systems, electronics, ADAS (Advanced Driver Assistance Systems), electric vehicle (EV) technologies, and mobility innovations.
- Magna's New mobility wing drives innovation in mobility and sustainability, shaping the future of the automotive and supply chain industry.

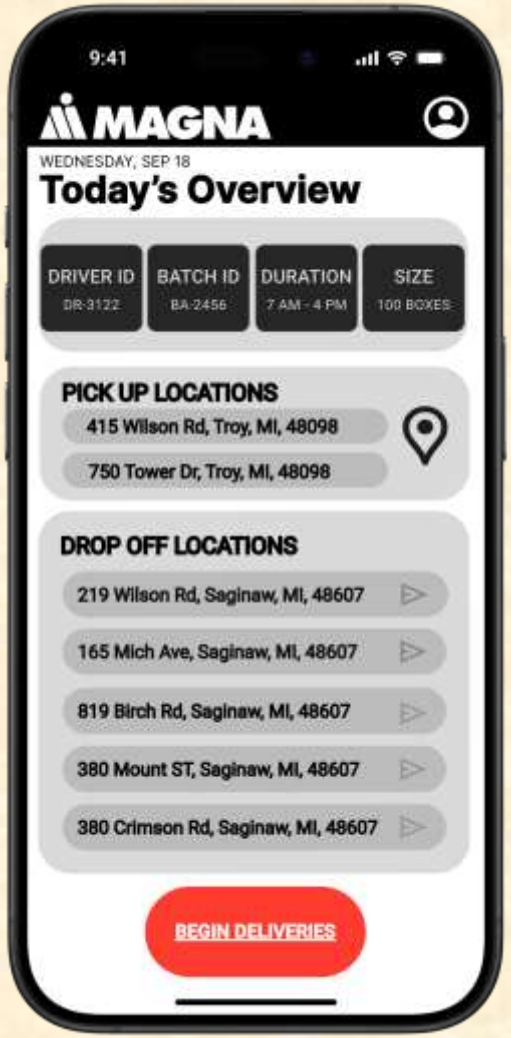
Project Functional Specifications

- The Mobile App for Delivery Optimization enhances delivery efficiency by optimizing routes for drivers and providing offline functionality to ensure continuous service in areas with poor network connectivity.
- For fleet operators to view the status of operations, a web application offers a dashboard that provide Comprehensive view of the current status of deliveries:
  - Displays the real-time location of the drivers.
  - Displays the status of each order/consignment.

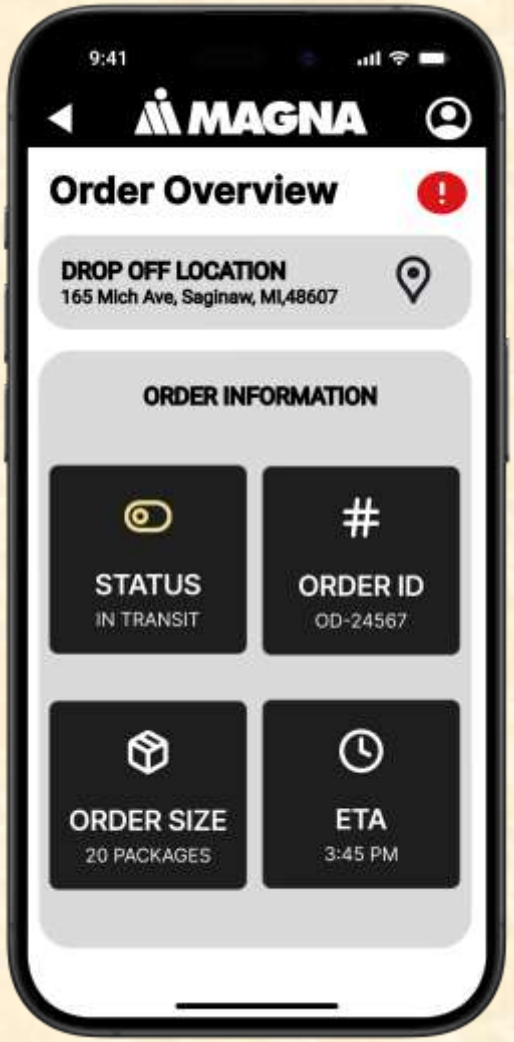
Project Design Specifications

- The mobile app equips delivery drivers with the most optimal route to efficiently complete a batch of orders along with in app Navigation provided by NextBillion AI. It features a user-friendly interface that displays key details, including delivery ETAs and comprehensive order information, ensuring drivers have all necessary data at their fingertips for seamless deliveries.
- The web app designed for fleet operators provides a detailed look into current orders out for delivery and drivers in transit. The web app also allows for fleet operators to make new order batches for drivers and view current locations on the map overview page.

Screen Mockup: Welcome Page

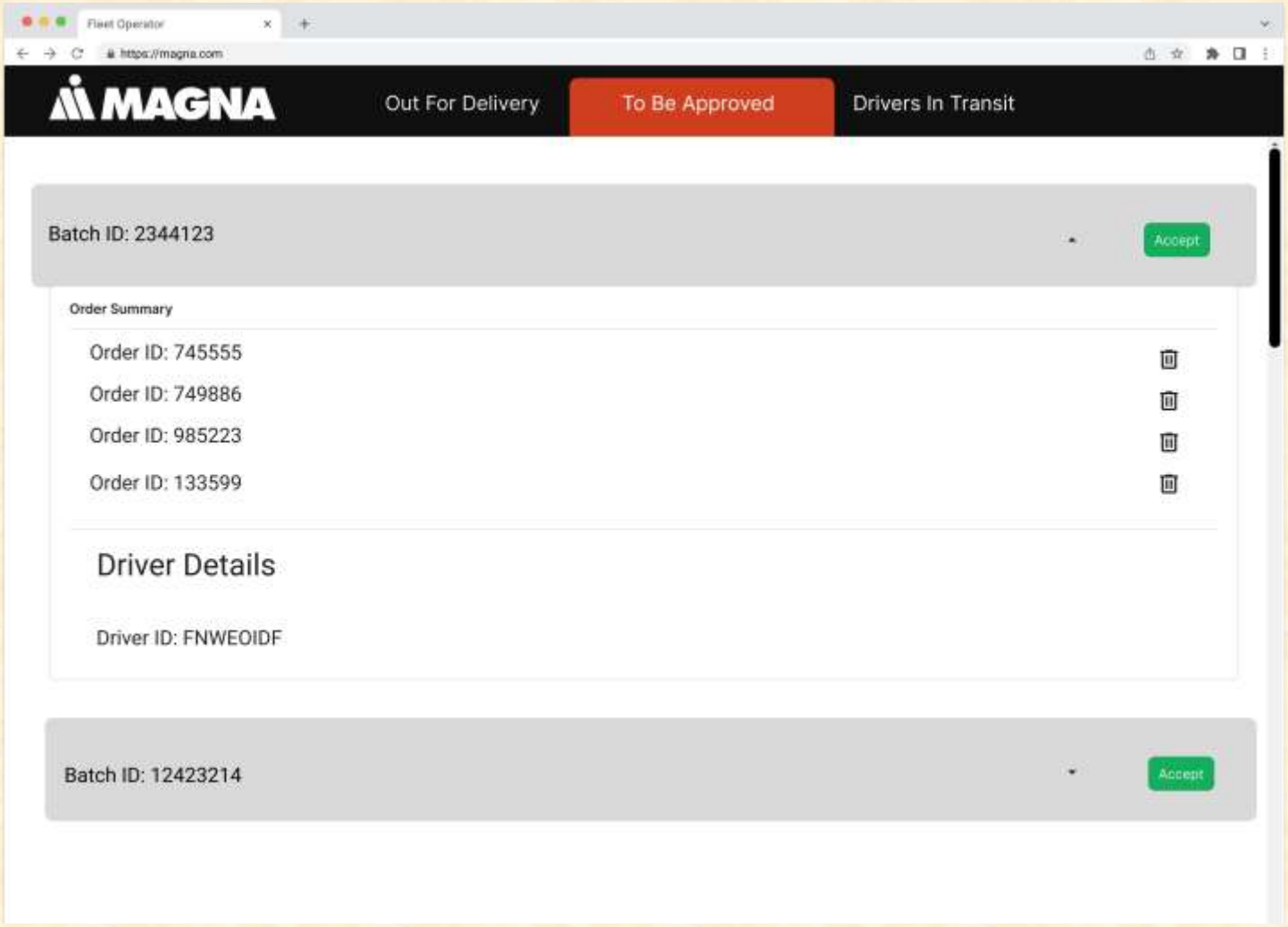


Screen Mockup: Order Overview

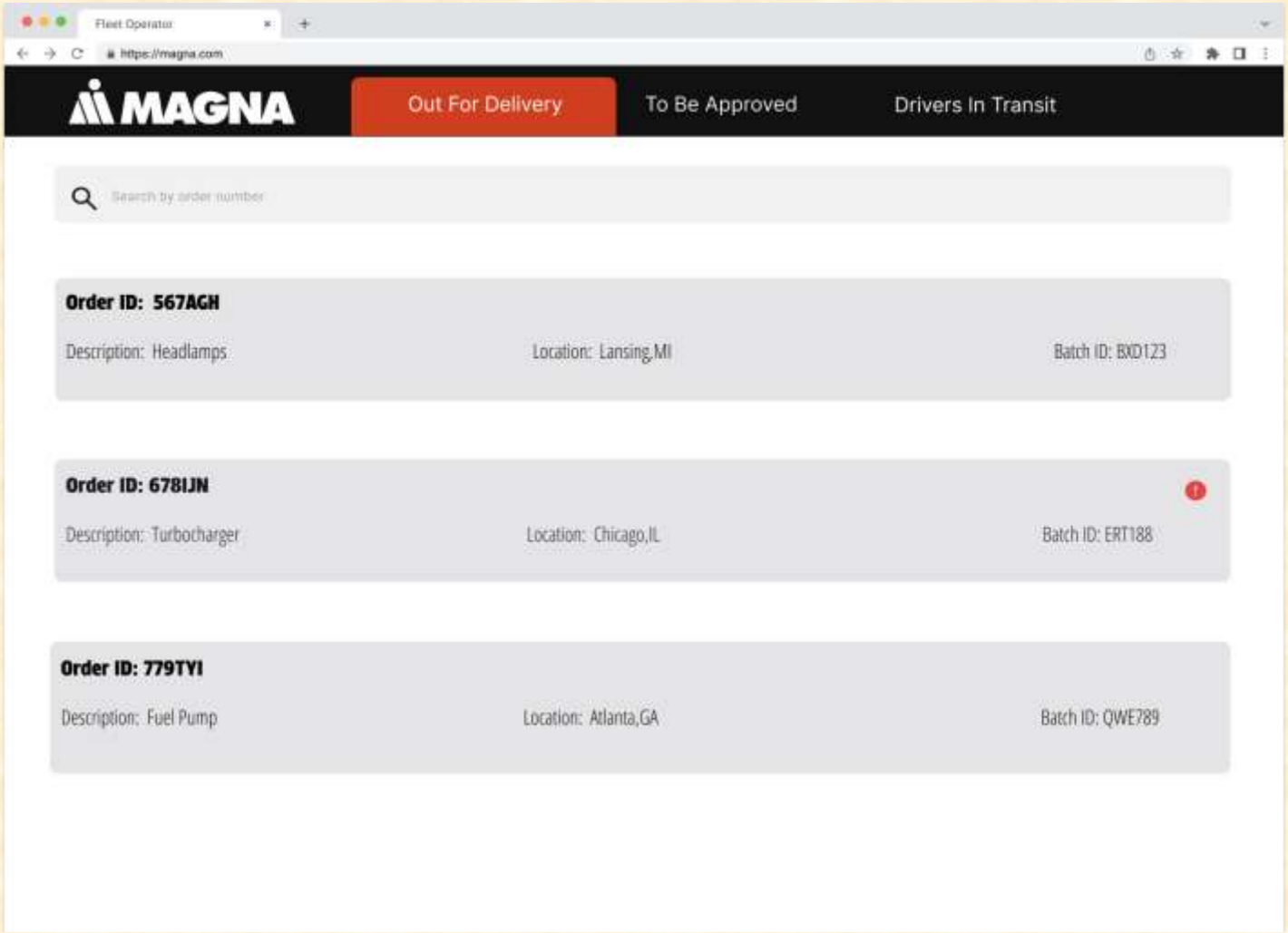


Screen Mockup: Navigation Screen

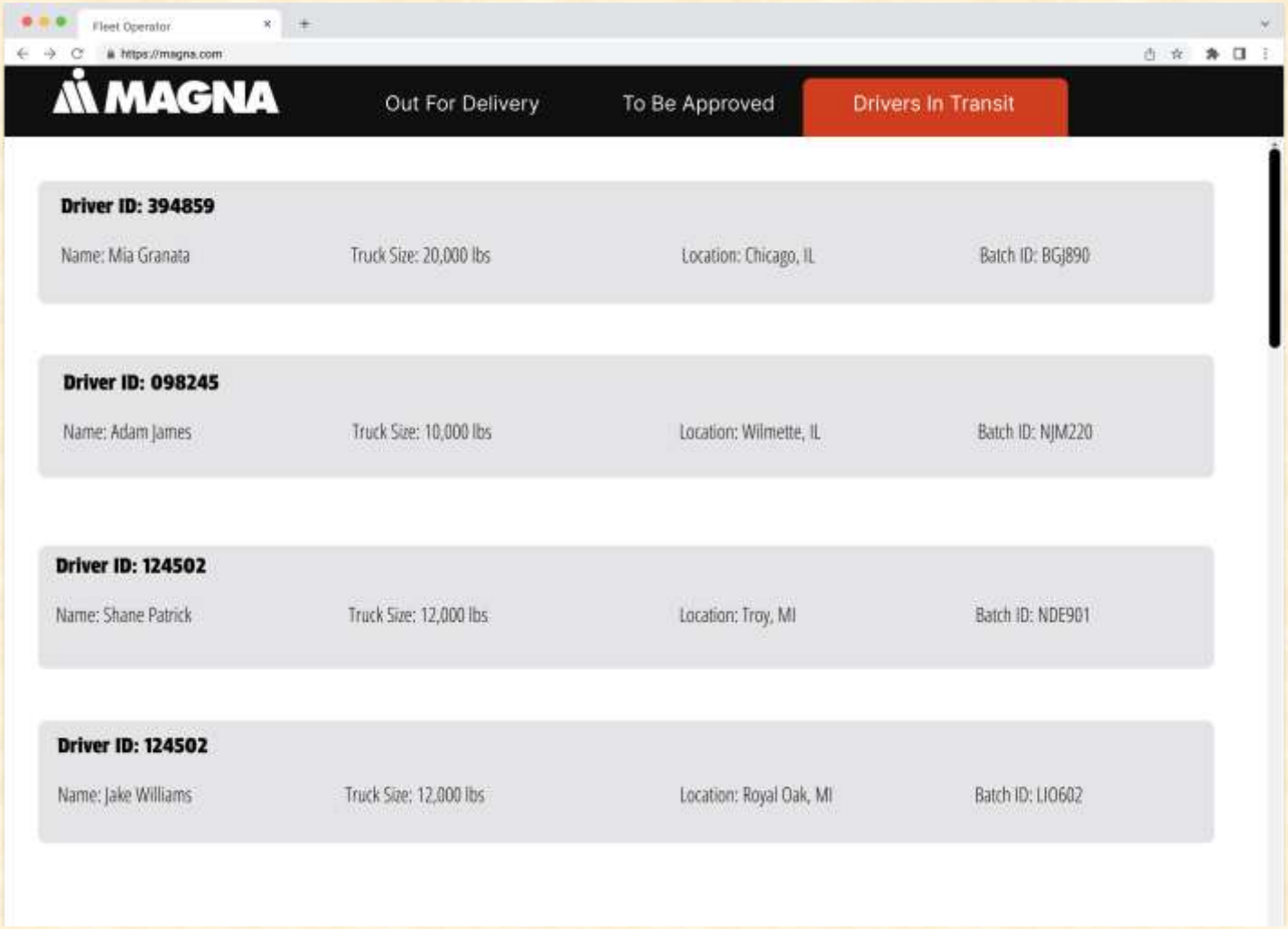
Screen Mockup: To Be Approved Page



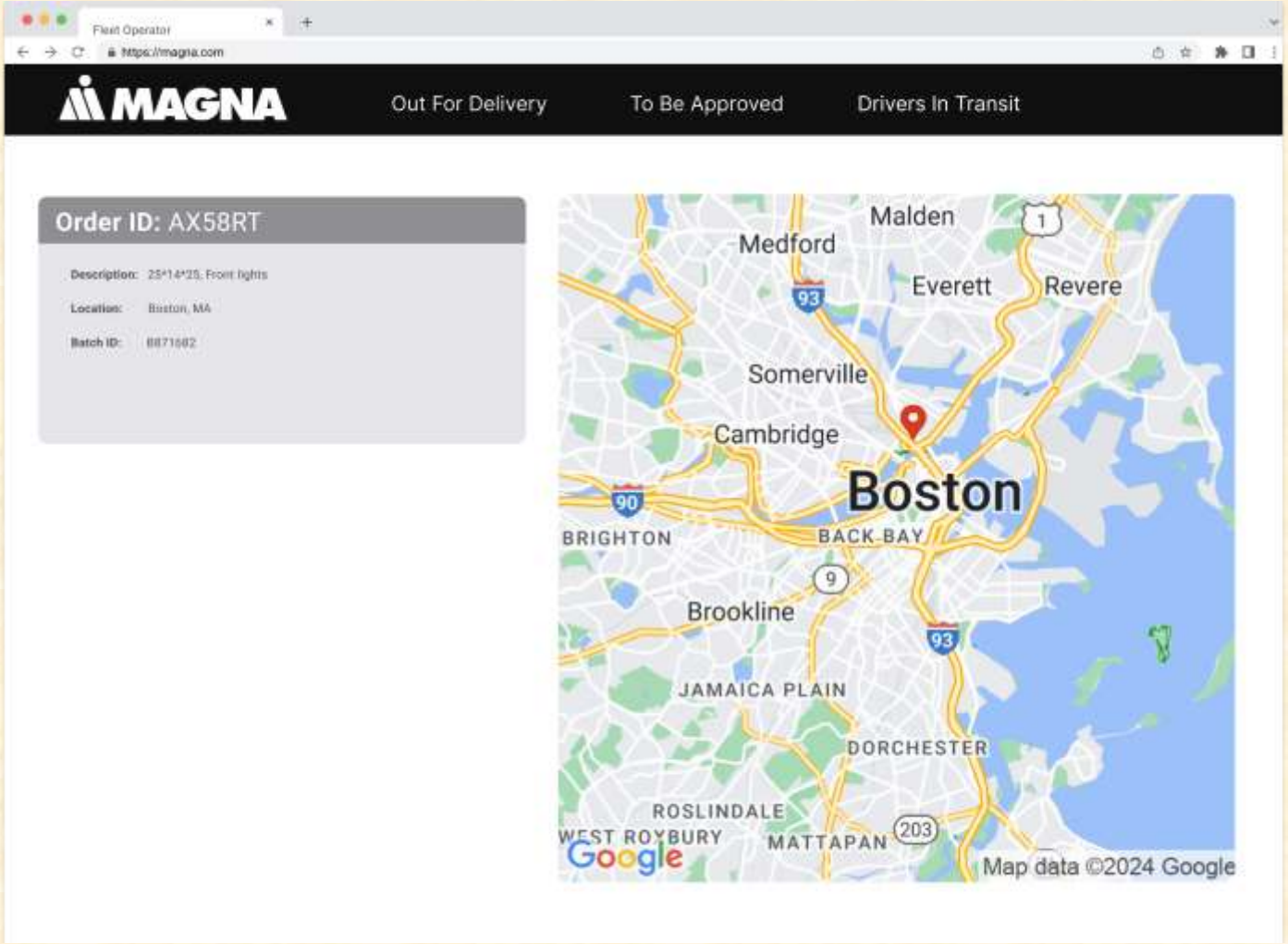
Screen Mockup: Orders View



Screen Mockup: Drivers View



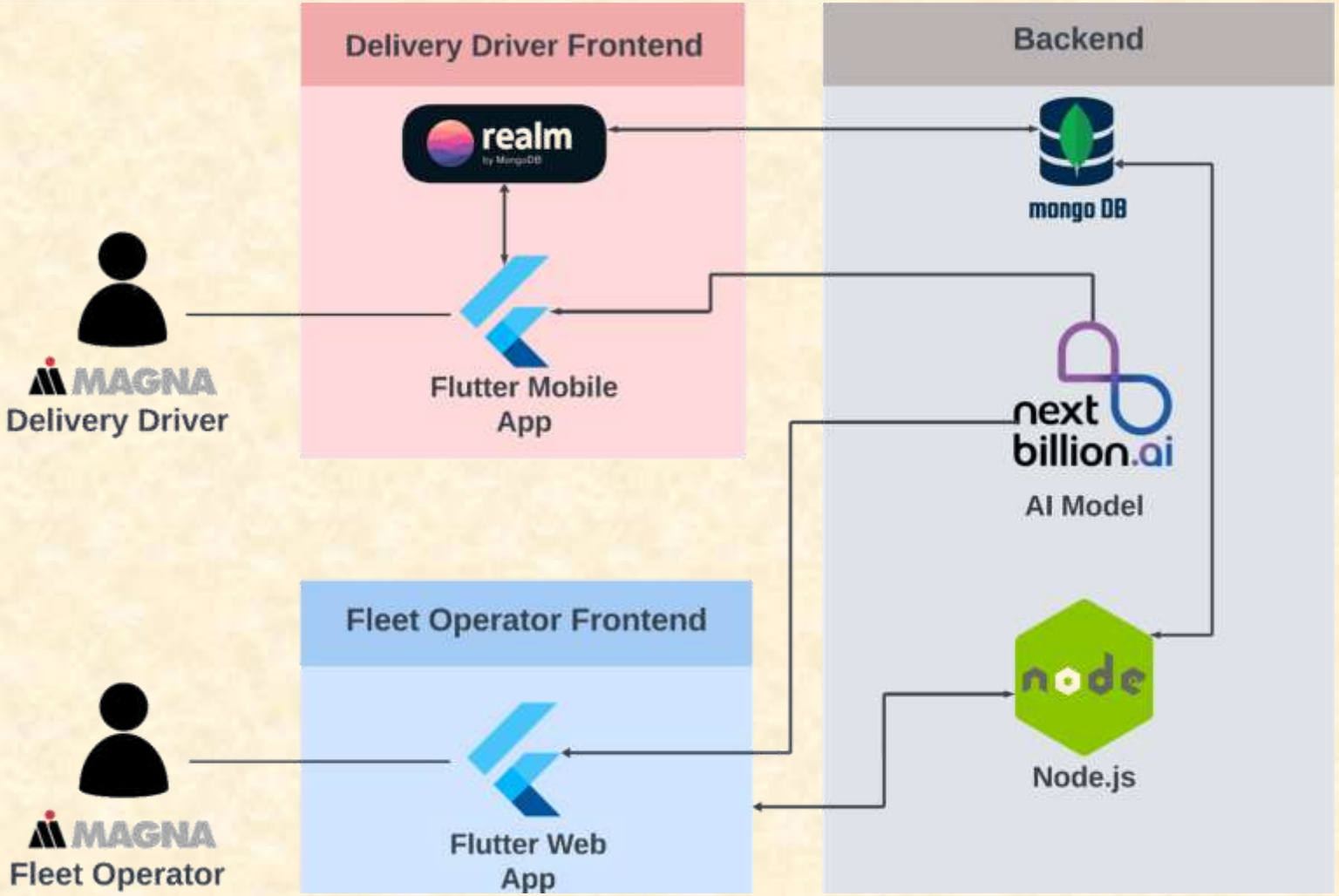
Screen Mockup: Map View



Project Technical Specifications

- A Flutter-based cross-platform mobile application is designed to optimize delivery routes for drivers. It leverages MongoDB Realm for offline functionality, ensuring continuous operation in areas with limited connectivity.
- A Flutter web application allows fleet operators to track the real-time status of consignments.
- The back-end is built with Node.js, enabling interaction between the web application and the MongoDB Atlas database.
- MongoDB Atlas serves as the central database, utilized by both the mobile and web applications.
- Route optimization and map integration are powered by NextBillion.AI, ensuring accurate navigation for drivers.

Project System Architecture



Project System Components

- Hardware Platforms
  - None
- Software Platforms / Technologies
  - Flutter
  - MongoDB
  - NextBiilion.AI
  - Node.js

Project Risks

- **NextBillion.AI**
  - Unsure if NextBillion.AI can display a map when not connected to the network.
  - If NextBillion.AI does not support offline navigation capabilities, then we plan to have Google maps free API as a back up if the mobile application loses connectivity.
- **Offline Capabilities**
  - We are unfamiliar with NextBillion.AI's geocode data processing, this has created uncertainty around the best approach for storing geocode data offline and syncing it with the database when back online.
  - We store location data from the mobile device every few milliseconds in an offline database (Realm). Upon connection, the system will automatically sync the data to the online database, ensuring seamless updates without relying on NextBillion.AI's specific geocode processing.
- **Syncing Common Databases Between Two Different Apps**
  - This project involves both a mobile application and a standalone web application, both of which connect to the same database. Our concern is maintaining the integrity of our database with simultaneous connections.
  - We will implement a Node.js-based queue that holds database write commands from both the mobile and web applications. This queue will process requests sequentially, ensuring consistent communication with the database and maintaining data integrity across both platforms.
- **Database Being Stored Locally**
  - With MongoDB realm storing data locally on the device this is a security risk and could be exploited.
  - To ensure secure storage, data encryption will be implemented across all platforms. Additionally, we will utilize Keychain for iOS and Keystore for Android to securely store sensitive information.

Questions?