Traffic and urban mobility analysis platform as Waze

Rubén Darío Fúquene Castiblanco
Thomas Felipe Sarmiento Oñate

Mayo 13 2025

Functional requirements

1. User management
    a. Users must be able to register, log in, and log out securely.
    b. Users must be able to update profile information and preferences.
    c. The system must manage user roles (e.g., regular user, admin).
2. Route Search & Navigation
    a. Users must be able to search for routes between two or more locations.
    b. The platform must display real-time traffic conditions on a map using color-coded routes.
    c. The system must provide users with current traffic conditions, including congestion levels and incidents, based on their location.
    d. The platform must generate alternative routes based on real-time traffic data to reduce travel time or most efficient route.
    e. The system must provide estimated travel times based on historical and live data.
    f. The platform must suggest safe and efficient bike routes, including dedicated bike lanes.
    g. The system must enable logistics managers to track vehicles and suggest optimal delivery routes.
    h. The system must offer route options that minimize carbon emissions, using factors such as distance, average speed, and terrain.
    i. The application must display estimated arrival times for buses and trains, integrating public transportation APIs.

3. Real-Time Data Ingestion & Processing
    a. The platform must ingest traffic data from APIs (Google Maps, OpenStreetMap).
    b. The system must support user-submitted traffic reports (accidents, roadblocks, etc.).
    c. Traffic data must be updated in real-time and stored historically for analysis.
    d. Users must be able to submit real-time traffic reports (e.g., accidents, roadblocks).
4. Analytics & Prediction
    a. The system must generate traffic forecasts using historical and live data.
    b. A web dashboard must allow city planners to access analytics on traffic patterns and congestion trends.
    c. The system must allow real-time heatmaps and trend visualizations.
5. Notifications & Alerts
    a. Users must receive notifications about upcoming traffic jams with recommended alternatives.
    b. Users must be able to customize their alert preferences.
6. Integration with External Services
    a. The platform must integrate with ride-hailing, logistics, and public transport APIs.

Non-functional requirements

1. Performance
    a. The system must process and respond to route queries within 2 seconds under normal load.
    b. The platform must support at least 100,000 concurrent users with 1 million daily API requests.
    c. Real-time traffic data updates must be ingested and reflected every 30 seconds.
2. Scalability
    a. The system must be horizontally scalable to accommodate increasing user and data loads.
    b. It must support dynamic scaling of services (e.g., cloud autoscaling for data ingestion and analytics).

3. Availability
    a. The platform must maintain 99.9% uptime to ensure consistent user access.
    b. Critical services (e.g., traffic updates) must implement redundancy and failover mechanisms.
4. Security
    a. All user data must be transmitted over HTTPS.
    b. Passwords must be hashed and salted before storage.
    c. The system must follow GDPR-like standards for user data privacy and retention.
    d. Admin and API endpoints must have role-based access control.
5. Maintainability
    a. The codebase must follow modular design principles to simplify updates and maintenance.
    b. Logs must be kept for all services with automated error tracking and alerting systems.
6. Usability
    a. The user interface must be responsive and accessible across devices.
    b. New users should be able to complete route searches with no more than 6 clicks/taps.
7. Compatibility
    a. The platform must run smoothly on modern browsers (Chrome, Firefox, Safari, Edge).
    b. The mobile version must be compatible with Android 8+ and iOS 13+.
8. Localization & Accessibility
    a. The interface must support at least English and Spanish.
    b. The app must provide an intuitive interface that allows first-time users to navigate it without prior instruction.
    c. Time, date, and distance formats must adapt to the user's region settings.

User Histories

Title 1	Priority 5	Estimate 5

UserHistory
As a commuter, I want to receive real-time traffic updates, so I can choose the fastest route to work.

AcceptanceCriteria
The user can view two or more routes in the app with similar travel times. The user can understand why we suggest a particular route, such as traffic conditions or road closures. The route should be displayed in less than a second. While the user is connected, the app can notify them in real time when traffic conditions change.

Title 2	Priority 3	Estimate 3

UserHistory
As a commuter, I want optimized route suggestions, so I can complete more deliveries in less time.

AcceptanceCriteria
The user can select one or more points of interest, and the app will show a route that includes those points.
These routes should be displayed in different colors for easy differentiation.
The user can change the destination point anytime they want.

Title 3	Priority 2	Estimate 5

UserHistory
as a commuter, I want real-time bus and train arrival times, so I can plan my trips efficiently.

AcceptanceCriteria
The user can search for routes using bus and train to reach their destination. The user can search for travel routes at different times.

Title 4	Priority 5	Estimate 5

UserHistory
as a commuter, I want an easy-to-use interface, so I can quickly understand how to navigate the app.

AcceptanceCriteria
When the user signs up, the app should guide them on how to use it.
The first search should be completed in less than a minute.

Title 5	Priority 3	Estimate

UserHistory
As an administrator, I want access to traffic data insights, so I can improve urban mobility and reduce congestion.

AcceptanceCriteria
Information should be displayed in less than a minute.
The admin can search for information by day, week, month, or year, if the data is available.

Title 6	Priority 3	Estimate

UserHistory
As an administrator, I want to track and optimize fleet routes, so I can lower fuel costs and delivery times.

AcceptanceCriteria
The admin can search for the best route for a specific hour, and the app should display the optimal route.
This search should be completed in less than a minute.

Title 7	Priority 5	Estimate

UserHistory
As a premium subscriber, I want an ad free experience and advanced analytics, so I can get the best insights for my travels.

Title 7	Priority 5	Estimate

AcceptanceCriteria
Searches by this user should have priority and be displayed in approximately one second or less. The user will not see ads. Their payment should be sufficient to offset the ad revenue. The user can view information about their daily travels if they search for it.

High-Level Architecture Design

The following diagram presents a component diagram illustrating the system's database architecture. It includes the data lake, data mart, and data warehouse components. Data collection is performed through various APIs and a web scraper specifically designed for TransMilenio, which gathers real-time information about buses in Bogotá. Although we could extend the data collection to other sources like the MIO system in Cali, the current scope is limited to TransMilenio for academic purposes.

Data ingestion and processing are managed by an ETL organizer that consolidates data into a structured SQL database. This includes user location data, organized map and route data, and unified traffic information from multiple APIs. The data warehouse stores the complete and structured data, including user information.

Two data marts are defined: - Data Mart for Commuters and Subscribers: Focuses on localized data, such as maps and routes specific to Bogotá. If needed, data from Cundinamarca can be included to provide modular and agile data access. - Data Mart for Administrators: Provides comprehensive, aggregated data that administrators can query to monitor system performance and user activity.

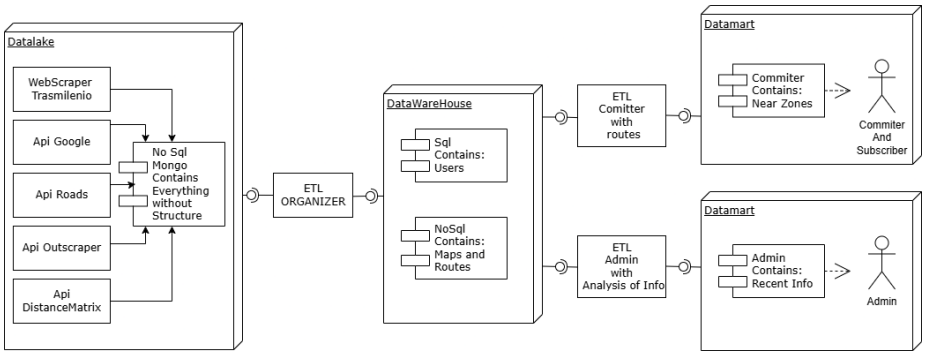

Figure 1: Component Diagram of the System Database Architecture