

Continuous Visualization of CyRide Through an Interactive Map

Team 22: Evan Schlarmann, Endi Odobasic, Braden
Buckalew, Andrew McMahon

Advisor: Selim, Mohamed
Client: Soliman, Mohammed

Table of Contents

- 2 Requirements, Constraints, and Standards.....3**
- 2.1 Requirements and Constraints..... 3
- 2.2. Engineering Standards..... 5

2 Requirements, Constraints, and Standards

2.1 Requirements and Constraints

Functional Requirements:

- Server
 - The application requires a server to host the website. It must allow connections from users and handle all requests for the application.
- Real-Time Bus Tracking with Long/Lat Coordinates
 - Requires connection with LibreNMS to retrieve UE coordinates. This connection must retrieve locations within 0.2 seconds.
 - Requires machine learning to predict a UE's coordinates when the UE is not in range of a base station. The algorithm should provide predictions within 0.2 seconds.
- Prediction Algorithm
 - Machine learning can predict with 95% accuracy where a bus is currently located and when the corresponding UE will be back in range of a base station.
- WebSocket
 - WebSockets will provide efficient data transfers between the frontend and backend components. The WebSocket must transfer data every .2 seconds to update the UI in real time.
- Database
 - The project will utilize a database to store past and current data that can be used in machine learning algorithms. This requires the data to be efficiently stored in a schema that can be queried within 0.1 seconds.

Resource Requirements:

- Real-Time Data Feed Acquisition
 - The server must have enough memory to handle many requests for bus location data and machine learning algorithms so that the application doesn't suffer performance losses.
- Google Maps API
 - Google Maps API costs money per amount of requests. Calls to the API will have to be minimized to ensure the project can stay within budget.

Physical Requirements:

- Computer/Laptop Needed
 - The development will occur on laptops. This ensures that the UI fits the size requirement during development and provides access to developmental environments.

- WiFi/Cell Signal
 - A signal connection is required to retrieve UE data for bus locations. This allows the application to access external interfaces to retrieve necessary data.

Qualitative Aesthetic Requirements:

- User Friendly Design
 - Need an intuitive design that a user can quickly grasp and use completely. Anyone can access the application, so it must be accessible to people with different technical skills.
- Visual Feedback
 - The UI display must provide users with clear feedback when using different functionalities. This feedback will be accurate to its design and clearly provide its data.

Environment Requirements:

- Need of UE Device to Fetch Real-Time Location
 - A UE device is needed to test data retrieval from LibreNMS and prediction algorithms once the UE device is out of range. The UE device will be supplied on the brown bus route for Cyride.
- Computers For Bus Visualization Display
 - Computers are needed to access the application for development and testing.

UI/UX Requirements:

- Minimal + Simple Interface
 - It will be a simple interface to reduce Navigation Complexity for the User. Will have the One UE preselected.
- Error Displays
 - The application will notify users of network or server errors, ensuring a smoother user experience by addressing issues effectively.
- ToolTips
 - The application will include tooltips and a comprehensive FAQ section to assist users in understanding the system, ensuring ease of use and accessibility for all.
- Map Interaction
 - Users can easily interact with the map, including dragging, zooming, and viewing building names, for efficient exploration and navigation.
- On-Screen Displays and Overlays
 - On-screen displays and overlays will provide additional information, such as bus routes and availability, enhancing the user experience with informative features.

2.2. Engineering Standards

- IEEE 29148 - Standard for Systems and Software Engineering Life Cycle Processes:
 - Provides requirements for planning documentation and specifications that will be developed for the project and may be revised during the project's life cycle.
- IEEE 1012 – Standard for System, Software, and Hardware Verification and Validation:
 - Covers processes, methods, and techniques to ensure software products meet specified requirements. This ensures that the project meets all user and client requirements once it is completed.
- IEEE 1061 – Standard for a Software Quality Metrics Methodology:
 - Guides measuring software quality attributes and assessing product quality. This ensures that the product is efficient and meets all computational constraints provided by the client.
- IEEE P3123 – Standard for Artificial Intelligence and Machine Learning (AI/ML) Terminology and Data Formats
 - This provides a standardized way for implementations of machine learning that can be used in the project.
- IEEE/ISO/IEC 23026: Systems and Software Engineering -- Engineering and Management of Websites for Systems, Software, and Services Information
 - This standard provides website life cycle requirements, including design, testing, and management. This provides a guideline to ensure a quality life cycle of the website such that every aspect has been thought out.