MAARGA - A REAL-TIME BIKERS' TRACKING PLATOFRM.

***Project Reference No.:** 49S_BE_5730*

College : RNS Institute of Technology, Bengaluru
Branch : Department of Artificial Intelligence and Machine Learning
Guide(s) : Ms. Vinutha D
Student(s) : Mr. Samanyu Y N
Ms. Rutu V
Mr. Prithvi V Pulgur
Mr. Gaurav Aditya M D

Keywords:
Geolocation, GPS Tracking, Stop Detection, OpenLayers, OpenStreetMap (OSM).

Introduction:

1. Cycling and biking in groups are increasingly popular activities for fitness, recreation, adventure travel, and long-distance touring. However, maintaining coordination among all riders in a group can be challenging, particularly when traveling across long distances, unfamiliar routes, or areas with varying terrain and traffic conditions. Riders may get separated due to differences in speed, navigation errors, unexpected stops, or emergencies, which can create safety risks and communication difficulties.
2. This project aims to develop a real-time biking group tracking system that improves group coordination, situational awareness, and rider safety through continuous location sharing and intelligent alerts. The system allows riders within a group to track each other's live positions on an interactive map, ensuring that no member gets lost or left behind during the ride.
3. The platform integrates modern web and mobile technologies to enable seamless real-time communication. The frontend is developed using React Native, providing a cross-platform mobile application experience. The backend is built using Node.js and Express.js, with Socket.IO enabling low-latency, bidirectional communication for real-time updates. For mapping and geospatial visualization, OpenLayers is used in combination with OpenStreetMap, allowing accurate display of rider locations, routes, and geofenced boundaries.
4. Key features of the system include real-time location sharing, geofencing alerts when a rider moves outside a predefined distance range, automatic stop detection to identify unusual halts, and emergency notifications that allow riders to quickly alert the group in critical situations. By combining real-time tracking with intelligent safety features, the proposed system enhances group coordination, reduces the risk of riders getting lost, and provides a safer and more connected biking experience.

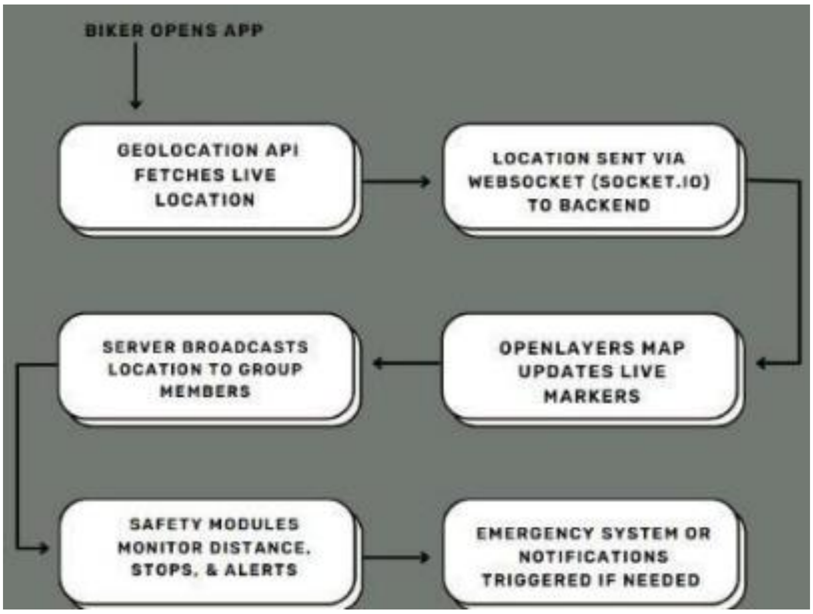
Objectives:

1. The objective of this project is to develop a mobile application that enables biking groups.
2. Track live locations of all group members using the Geolocation API and WebSockets.

3. Set a geofencing radius (500 mts) and notify users when a member moves out of range.

Methodology:

The core functionality of the proposed biker tracking system revolves around real-time geolocation tracking, group-based data broadcasting, dynamic map rendering, safety monitoring, and emergency alerting. The workflow described below outlines the interaction between client and server components, forming the backbone of the system.



Step 1:
Biker Opens the Platform Upon launching the application, the biker is either prompted to join an existing group or create a new one using a unique room code. This forms a private group room where location data will be shared among authorized members only.
Step 2:
Geolocation API Fetches Live Location The application invokes the device's native Geolocation API, which continuously captures GPS coordinates (latitude and longitude). The polling interval is optimized to balance precision and battery usage (e.g., every 5–10 seconds).
Key Data Captured:

1. Latitude & Longitude
2. Timestamp
3. Speed and heading (optional)
4. Accuracy radius This data forms the foundation of all location-based services in the website.

Step 3:
Real-Time Location Sent via WebSocket (Socket.IO) Once fetched, the geolocation data is transmitted in real time to the backend server using WebSockets through Socket.IO. This low-latency, event-driven communication mechanism enables quick, two-way data flow between client and server.

Data Packet Format:

{
    "groupId": "xyz123",
    "userId": "abc456",
    "location": {
    "latitude": 12.9716,
    "longitude": 77.5946,
    "timestamp": 1716147800000}
}

Step 4:
Server Broadcasts Location to Group Members The backend, built using Node.js and Express.js, handles incoming location updates and identifies the correct group room based on the group Id. The server then broadcasts the updated location data to all connected clients within the same room using Socket.IO's built-in room-based communication.
This ensures that:
• Each biker receives live updates of all group members.
• No data is leaked outside the group context.

Step 5:
OpenLayers Map Updates Live Markers 7 On the frontend, OpenLayers integrated with OpenStreetMap (OSM) is used to render a dynamic map.
For each incoming location update:

1. A new marker is created if it's a new group member.
2. An existing marker is updated with the latest coordinates.
3. The marker is removed if a user disconnects or leaves the group. Each marker is identified by a unique ID and is optionally labeled with the biker's nickname or avatar.

Step 6:
Safety Monitoring Modules Trigger Alerts The Platform includes intelligent monitoring logic that enhances rider safety by:

1. Stop Detection: If a biker remains stationary for a predefined time (e.g., 2+ minutes), the Platform prompts a confirmation. If the user does not respond within a set timeframe, the system assumes a potential emergency.
2. Proximity Monitoring: Using the Haversine formula, the Platform calculates the distance between group members. If a biker moves outside a 2 km radius, an alert is issued to notify the group.
3. Inactivity Tracking: Tracks whether location updates have stopped completely, which could imply network or device issues.

Step 7:
Emergency Alerts and Notifications In cases of unexpected stops, distance violations, or direct user-triggered events:

1. The SOS System can be activated with one tap.

2. A real-time alert message with GPS location is sent to:
    a. All group members (via Socket.IO and FCM).
    b. Emergency contacts stored in the user's profile.
3. Notifications are pushed via Firebase Cloud Messaging (FCM) to ensure delivery even when the platform is in the background.

Result and Conclusion:

1. In conclusion, The Maarga real-time biker tracking platform was developed with the objective of enabling biking groups to monitor each other's locations continuously and reliably during group rides. The system integrates modern web technologies to deliver a responsive and low-latency tracking experience. Through a structured development process, the project progressed from requirement analysis and architectural design to detailed implementation, testing, and validation. The platform incorporates a React-based frontend, a Node.js and Express backend, and Socket.IO-powered bi-directional communication, supported by an in-memory room management system and map rendering using OpenLayers. Each module was designed to be modular, scalable, and maintainable, ensuring robust performance during real-time tracking.
2. A comprehensive design methodology was followed, beginning with the creation of high-level and low-level architectures, class diagrams, sequence diagrams, and data flow representations. These design artifacts provided clarity on module interactions, data movement, and control flows, guiding the implementation phase. The final system was implemented with a focus on fault tolerance, accurate location synchronization, secure communication, and scalability. Extensive unit testing verified database operations and schema correctness, while system testing validated the end-to-end behavior of the platform under real-world conditions, including multiple simultaneous connections, room creation, location streaming, and user disconnections.
3. Overall, the project successfully demonstrates the feasibility and effectiveness of a browser-based real-time biker tracking solution. The system achieves its core goals of ensuring live visibility of all bikers in a group, maintaining accurate updates, and providing a reliable communication channel between frontend and backend components. The modular structure and extensible design of the platform lay a strong foundation for future enhancements such as geofencing alerts, ride analytics, safety notifications, chat systems, and multi-device synchronization. The successful implementation and testing of Maarga confirm its potential as a practical and scalable platform for enhancing safety and coordination in group biking activities.

References:

1. [1] D. Ashbrook and T. Starner, "Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users," Personal and Ubiquitous Computing, vol. 7, pp. 275–286, 2003. doi: 10.1007/s00779-003-0240-0.
2. [2] G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets, "Threshold Settings for TRIP/STOP Detection in GPS Traces," 2016. [Online]. Available: https://www.researchgate.net/publication/297650658_Threshold_settings_for_TRIP_STOP_detection_in_GPS_traces

3. [3] Y. Shevchenko and U.-D. Reips, "Geofencing in Location-Based Behavioral Research: Methodology, Challenges, and Implementation," Behavior Research Methods, 2023. doi: 10.3758/s13428-023-02213-2.
4. [4] O. Ramadani, "Real-Time Location Tracking with JavaScript and Socket.IO," 2024. [Online]. Available: https://medium.com/@okiramadani53/real-time-location-tracking-with-javascript-and-socket-io5c00ea6ef7d5
5. [5] A. Skoogh, J. Michaloski, and N. Bengtsson, "Towards Continuously Updated Simulation Models: Combining Automated Raw Data Collection and Automated Data Processing," 2010.Available:https://www.researchgate.net/publication/224208989_Towards_continuously_updated_simulation_models_Combining_automated_raw_data_collection_and_automated_data_processing
6. [6] I. Fette and A. Melnikov, "The WebSocket Protocol," IETF RFC 6455, 2011. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6455
7. [7] J. A. Stankovic, "Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems," IEEE Computer, vol. 21, no. 10, pp. 10–19, 1988. doi: 10.1109/2.7050.
8. [8] P. A. Zandbergen, "Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning," Transactions in GIS, vol. 13, no. s1, pp. 5–25, 2009. doi: 10.1111/j.1467-9671.2008.01181.x.
9. [9] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, 2008. doi: 10.1109/MPRV.2008.80.
10. [10] D. Flanagan, JavaScript: The Definitive Guide, 7th ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
11. [11] D. Crane and P. McPherson, Rich Internet Applications with AJAX, WebSockets, and Comet. Upper Saddle River, NJ, USA: Prentice Hall, 2008.
12. [12] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Upper Saddle River, NJ, USA: Addison-Wesley, 2012.
13. [13] L. Richardson and S. Ruby, RESTful Web Services. Sebastopol, CA, USA: O'Reilly Media, 2007.
14. [14] D. Norman, The Design of Everyday Things (Revised and Expanded Edition). Cambridge, MA, USA: MIT Press, 2013.
15. [15] G. J. Myers, C. Sandler, and T. Badgett, The Art of Software Testing, 3rd ed. Hoboken, NJ, USA: Wiley, 2011.
16. [16] React Documentation, "React – A JavaScript Library for Building User Interfaces," 2025. [Online]. Available: https://react.dev/
17. [17] Node.js Foundation, "Node.js Documentation," 2025. [Online]. Available: https://nodejs.org/en/docs/
18. [18] Express.js Team, "Express.js – Fast, Unopinionated, Minimalist Web Framework," 2025. [Online]. Available: https://expressjs.com/
19. [19] Socket.IO Contributors, "Socket.IO Documentation – Real-Time Bidirectional Communication," 2025. [Online]. Available: https://socket.io/docs/
20. [20] MongoDB Inc., "MongoDB Atlas Documentation," 2025. [Online]. Available: https://www.mongodb.com/docs/atlas/

21. [21] OpenLayers Community, “OpenLayers Documentation,” 2025. [Online]. Available: https://openlayers.org/en/latest/doc/
22. [22] OpenStreetMap Foundation, “OpenStreetMap,” 2025. [Online]. Available: https://www.openstreetmap.org/

Future Scope:

1. The real-time biking group tracking system can be further enhanced with several advanced features to improve safety, scalability, and user experience. One major improvement could be the integration of AI-based predictive analytics to identify risky riding patterns, such as sudden speed drops, unusual route deviations, or frequent stops, and proactively alert the group. The system can also incorporate machine learning models to predict potential hazards based on historical ride data, weather conditions, and traffic patterns.
2. Integration with IoT-enabled smart wearables or bike-mounted sensors could allow automatic accident detection by monitoring parameters like sudden impact, abnormal tilt angle, or heart rate irregularities. In such cases, the system could automatically trigger emergency alerts and share the rider’s live location with predefined contacts.
3. Future versions could include offline tracking support using local data caching and delayed synchronization to handle areas with poor internet connectivity. The platform could also be expanded to support route optimization, suggesting safer or less congested paths based on real-time traffic and terrain information.
4. Adding voice-enabled commands would allow riders to interact with the app hands-free, improving usability and safety while riding. The system could also include group performance analytics, providing insights such as average speed, distance covered, elevation changes, and ride history visualization.
5. To enhance scalability, the application could be extended to support large biking events, marathons, or delivery fleet tracking. Integration with third-party fitness platforms like Google Fit or Strava could provide comprehensive activity tracking. Security features such as end-to-end encryption, role-based access control, and temporary location sharing links can further improve data privacy.
6. Additionally, incorporating augmented reality (AR) navigation, voice alerts, and smart notifications could significantly improve user engagement and situational awareness. With continuous advancements in mobile computing, geospatial technology, and AI, the system has strong potential to evolve into a comprehensive smart mobility safety platform.