Software Requirements Specification

Electric Bus Location Tracker App

Prepared for: Asiya Batool
Department of Computer Science
Namal University, Mianwali

Prepared by:

Ali Abbas	NUM-BSCS-2024-06
Laiba Tajj	NUM-BSCS-2024-31
Mehroz Ali Khan	NUM-BSCS-2024-34

28 December 2024

Contents

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a complete description of the Electric Bus Location Tracker App system. The document covers details of all functional and non-functional requirements for real-time tracking and management of electric buses in Mianwali District, Punjab, Pakistan.

1.2 Scope

The Electric Bus Location Tracker App is a mobile-based software system designed to provide real-time tracking and management facilities for the electric bus transportation system in Mianwali.

1.2.1 System Benefits

- Real-time bus tracking
- Enhanced passenger experience
- Supports Drivers registration process
- Improved operational control

1.2.2 System Objectives

- Passengers ease of time management
- Provide Drivers duty schedule and attendence
- Support administrative monitoring and changes

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
API	Application Programming Interface - A set of protocols for building software applications
AWS	Amazon Web Services - Cloud computing platform used for hosting
EAT	Estimated Arrival Time
GPS	Global Positioning System - Satellite-based navigation system
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
JWT	JSON Web Token - Authentication mechanism
MongoDB	NoSQL database used for storing application data
NFR	Non-Functional Requirement
Node.js	JavaScript runtime environment for backend development
PMA	Punjab Mass Transit Authority
REST	Representational State Transfer - API architectural style
RP	Requirement Provider - Stakeholder providing system requirements
SRS	Software Requirements Specification
UI/UX	User Interface / User Experience
WebSocket	Protocol for real-time bidirectional communication

1.4 References

1. IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Standard 29148-2018: Systems and Software Engineering - Life Cycle Processes - Requirements Engineering
3. Google Maps API Documentation: `https://developers.google.com/maps`
4. Geeksforgeeks Documentation: `https://www.geeksforgeeks.org/software-engineering/software-requirement-specification-srs-format`
5. Flutter Documentation: `https://flutter.dev/docs`
6. Node.js Documentation: `https://nodejs.org/docs`
7. MongoDB Documentation: `https://docs.mongodb.com`
8. Socket.io Documentation: `https://socket.io/docs`
9. Claude : `https://claude.ai/chat`

1.5 Overview

This document describes system introduction, overall description, functional requirements, non-functional requirements, external interfaces, and appendices.

2 Overall Description

2.1 Product Perspective

The Electric Bus Location Tracker App is a system integrated with GPS hardware, cloud services, and Google Maps API.

2.2 Product Functions

Passengers: Track buses, view routes, and ETA
Drivers: View duties and share location
Administrators: Monitor buses and manage operations

2.3 User Characteristics

2.3.1 Passengers

- **User Type**: General public with smartphone literacy
- **Age Range**: All ages
- **Language**: English

2.3.2 Drivers

- **User Type**: Person with basic smartphone literacy and are registered by PMA, Mianwali
- **Age Range**: 25-55 years
- **Language**: English

2.3.3 Administrators

- **User Type**: Person with advance smartphone usage expertise and are registered by PMA, Lahore
- **Age Range**: 25-50 years
- **Language**: English

2.4 Constraints

2.4.1 Regulatory and Legal Constraints

- Must comply with Pakistan's data protection and privacy regulations
- Must obtain user permission for location tracking
- Must secure and protect personal user information
- Must comply with Google Maps API terms of service

2.4.2 Hardware Limitations

- GPS accuracy depends on satellite visibility and device quality
- Mobile device battery consumption for continuous GPS tracking
- Network connectivity limitations in remote areas

2.4.3 Technical Constraints

- System must work with all types of Android devices
- Google Maps API usage quotas and rate limits
- Database storage capacity constraints

2.4.4 Development Constraints

- Budget limitations for cloud services and API usage
- Team size limited to three developers
- Limited access to actual buses for testing

2.5 Assumptions and Dependencies

2.5.1 Assumptions

- All drivers will have smartphones with internet connectivity
- Drivers will have working GPS installed in the smartphones
- Passengers will have access to smartphones with GPS functionalities and internet connection
- All the three type of users of the app: Driver, Passenger and Administration have literacy to English language
- We have full support by the government and PMA

2.5.2 Dependencies

- **Google Maps API**: System depends on Google Maps for mapping and routing
- **Internet Connectivity**: Requires continuous internet connection for real-time tracking
- **AWS Infrastructure**: System hosting depends on AWS availability
- **GPS Satellites**: Location accuracy depends on GPS satellite availability
- **Third-party Libraries**: Flutter packages, Node.js modules, and other dependencies
- **Mobile OS Updates**: Compatibility may be affected by OS updates

- **Stakeholder Cooperation**: Requires ongoing cooperation from stakeholder and PMA

3 Functional Requirements

3.1 Driver and Administration Combined Functions

3.1.1 User Authentication and Authorization

- **FR-001**: The system shall allow drivers and administrators to login using username, user ID, and password
- **FR-002**: The system shall validate username, user ID, and passwords during login
- **FR-003**: The system shall implement JWT-based authentication with 24-hour token expiry
- **FR-004**: The system shall provide password reset functionality via email/SMS
- **FR-005**: The system shall support two user roles for authentication and authorization: Driver, and Administrator
- **FR-006**: The system shall enforce role-based access control for all features
- **FR-007**: The system shall allow authenticated users to logout and invalidate authentication tokens
- **FR-008**: The system shall lock accounts after 5 failed login attempts for 30 minutes

3.1.2 Profile viewing

- **FR-009**: The system shall allow driver and administration to view their profile information

3.2 Passenger Functions

3.2.1 Show Real-time Bus Tracking

- **FR-010**: The system shall allow passengers to view real-time locations of all active buses
- **FR-011**: The system shall allow passengers to view bus details
- **FR-012**: The system shall allow passengers to pan across the map to view different areas
- **FR-013**: The system shall show passenger's current location on the map

3.2.2 Search Routes and Schedules

- **FR-014**: The system shall allow passengers to view all available bus routes
- **FR-015**: The system shall allow passengers to search for a specific route information by route name
- **FR-016**: The system shall allow passengers to search for routes by bus stop name
- **FR-017**: The system shall display complete route details including start point, end point, estimated time, and all stops
- **FR-018**: The system shall show all bus stops along each route on the map
- **FR-019**: The system shall display schedule of buses for all routes
- **FR-020**: The system shall provide functionality of searching the bus arrival timing of all the stops of all routes
- **FR-021**: The system shall display total distance for the complete route and also from one stop to another
- **FR-022**: The system shall display estimated travel time for the complete route

3.2.3 Calculate Estimated Arrival Time (EAT)

- **FR-023**: The system shall allow passengers to know EAT for a near bus to a specific stop
- **FR-024**: The system shall consider previous bus speed in EAT calculation
- **FR-025**: The system shall update EAT every 30 seconds based on bus movement
- **FR-026**: The system shall show "Arrived at [stop name]" when bus reaches the stop
- **FR-027**: The system shall show "Departed from [stop name]" when bus leaves the stop
- **FR-028**: The system shall display EATs for multiple buses if they serve the same route

3.2.4 Manage Favorite Routes

- **FR-029**: The system shall allow passengers to save routes as favorites
- **FR-030**: The system shall allow passengers to view their saved favorite routes
- **FR-031**: The system shall allow passengers to remove routes from favorites
- **FR-032**: The system shall provide quick access to favorite routes from home screen

3.3 Driver Functions

3.3.1 Duty Management

- **FR-033**: The system shall allow drivers to view their assigned duty for current day
- **FR-034**: The system shall display assigned route and bus number for current duty
- **FR-035**: The system shall allow drivers to view all duties assigned for the current month
- **FR-036**: The system shall allow drivers to search for duties by specific date
- **FR-037**: The system shall display duty schedule including route, bus, and timing
- **FR-038**: The system shall show driver's next upcoming duty
- **FR-039**: The system shall automatically mark its duty complete after arriving at the stop place of route
- **FR-040**: The system shall record duty completion time and date
- **FR-041**: The system shall allow driver to view its duties record means the duties completed and skipped the whole month or on a specific date

3.3.2 Live Location Sharing

- **FR-042**: The system shall allow driver to start its duty on the starting time of duty
- **FR-043**: The system shall share live location of the driver mobile GPS when he starts its duty
- **FR-044**: The system shall allow drivers to view their own bus location in real-time
- **FR-045**: The system shall allow drivers to view real-time locations of the buses which are on its route
- **FR-046**: The system shall automatically mark its duty done when he reached the destination by bus
- **FR-047**: The system shall stop sharing its location when he reached the destination by bus
- **FR-048**: The system shall store the starting time and ending time of the duty on drivers duty schedule

3.4 Administrator Functions

3.4.1 Bus Monitoring and Management

- **FR-049**: The system shall allow administrators to view real-time locations of all buses
- **FR-050**: The system shall display status of each bus (Active/Inactive)

- **FR-051**: The system shall allow administrators to view detailed information for each bus
- **FR-052**: The system shall allow administrators to add new buses to the system
- **FR-053**: The system shall allow administrators to edit bus information
- **FR-054**: The system shall allow administrators to remove buses from the system

3.4.2 Route and Schedule Management

- **FR-055**: The system shall allow administrators to create new routes
- **FR-056**: The system shall allow administrators to define bus stops for each route
- **FR-057**: The system shall allow administrators to modify existing route information
- **FR-058**: The system shall allow administrators to delete routes not in active use
- **FR-059**: The system shall allow administrators to define schedule timings for each route
- **FR-060**: The system shall allow administrators to modify schedule timings

3.4.3 User Account Management

- **FR-061**: The system shall allow administrators to view all user accounts
- **FR-062**: The system shall allow administrators to create new driver accounts
- **FR-063**: The system shall allow administrators to edit driver account information
- **FR-064**: The system shall allow administrators to activate driver accounts
- **FR-065**: The system shall allow administrators to deactivate driver accounts
- **FR-066**: The system shall prevent deletion of accounts with active duty assignments

3.4.4 Driver Duty Assignment

- **FR-067**: The system shall allow administrators to assign drivers to specific buses
- **FR-068**: The system shall allow administrators to create duty schedules for drivers
- **FR-069**: The system shall allow administrators to modify duty assignments
- **FR-070**: The system shall prevent assigning one driver to multiple duties at same time
- **FR-071**: The system shall prevent assigning one bus to multiple routes at same time

3.4.5 Record Reports

- **FR-072**: The system shall generate daily operational reports automatically
- **FR-073**: The system shall generate weekly operational reports automatically
- **FR-074**: The system shall generate monthly operational reports automatically
- **FR-075**: The system shall store all generated reports in database
- **FR-076**: The system shall send report copies to administrator mobile application
- **FR-077**: The system shall allow administrators to export reports in PDF format

4 Non-Functional Requirements

4.0.1 Performance Requirements

- **NFR-001**: The system shall update bus locations with maximum 5-second latency
- **NFR-002**: The system shall support at least 1000 concurrent users
- **NFR-003**: The system shall load the map interface within 3 seconds on 4G connection
- **NFR-004**: The system shall calculate EAT within 2 seconds of request
- **NFR-005**: The system shall respond to API requests within 1 second under normal load
- **NFR-006**: The system shall handle at least 50 buses tracking simultaneously
- **NFR-007**: The system shall maintain database query response time under 500ms

4.0.2 Security Requirements

- **NFR-008**: The system shall encrypt all data transmission using TLS 1.3
- **NFR-009**: The system shall store passwords using bcrypt hashing with salt rounds of 10
- **NFR-010**: The system shall implement JWT authentication with secure signing algorithms
- **NFR-011**: The system shall enforce HTTPS for all API communications
- **NFR-012**: The system shall validate and sanitize all user inputs to prevent injection attacks
- **NFR-013**: The system shall implement rate limiting of 100 requests per minute per user
- **NFR-014**: The system shall log all authentication attempts for security auditing
- **NFR-015**: The system shall automatically expire inactive sessions after 24 hours
- **NFR-016**: The system shall comply with OWASP Top 10 security standards
- **NFR-017**: The system shall implement role-based access control (RBAC) for all operations

4.0.3 Reliability Requirements

- **NFR-018**: The system shall maintain 99.5% uptime during operational hours (6 AM - 11 PM)
- **NFR-019**: The system shall recover from server failures within 5 minutes using auto-restart
- **NFR-020**: The system shall implement automatic database backups every 24 hours
- **NFR-021**: The system shall retain backup data for minimum 30 days
- **NFR-022**: The system shall gracefully handle GPS signal loss without crashing
- **NFR-023**: The system shall continue functioning offline with cached data
- **NFR-024**: The system shall automatically reconnect after network interruption
- **NFR-025**: The system shall have a Mean Time Between Failures (MTBF) of at least 720 hours
- **NFR-026**: The system shall log all critical errors for debugging and analysis
- **NFR-027**: The system shall provide error messages that are informative and user-friendly

4.0.4 Availability Requirements

- **NFR-028**: The system shall be available 24/7 with planned maintenance window of 2 hours weekly
- **NFR-029**: The system shall notify users 24 hours before scheduled maintenance
- **NFR-030**: The system shall implement load balancing for high availability
- **NFR-031**: The system shall have redundant database servers for failover
- **NFR-032**: The system shall monitor server health continuously

4.0.5 Maintainability Requirements

- **NFR-033**: The system shall follow consistent coding standards and style guides
- **NFR-034**: The system shall include comprehensive inline code documentation
- **NFR-035**: The system shall use modular architecture for easy component updates
- **NFR-036**: The system shall maintain detailed API documentation
- **NFR-037**: The system shall implement automated testing with minimum 80% code coverage
- **NFR-038**: The system shall use version control (Git) for all code changes
- **NFR-039**: The system shall allow feature updates without system downtime
- **NFR-040**: The system shall provide administrator tools for configuration management

4.0.6 Portability Requirements

- **NFR-041**: The mobile application shall run on Android 8.0 (API level 26) and above
- **NFR-042**: The system shall support responsive design for various screen sizes (4" to 7")
- **NFR-043**: The system shall be deployable on any cloud platform (AWS, GCP, Azure)

4.0.7 Usability Requirements

- **NFR-044**: The system shall provide an intuitive interface requiring no training for passengers
- **NFR-045**: The system shall support English language
- **NFR-046**: The system shall use universally recognizable icons and symbols
- **NFR-047**: The system shall provide help text and tooltips for complex features
- **NFR-048**: The system shall follow Material Design guidelines for Android
- **NFR-049**: The system shall allow users to complete common tasks in maximum 3 clicks/taps
- **NFR-050**: The system shall provide visual feedback for all user actions within 200ms
- **NFR-051**: The system shall use consistent color schemes and typography throughout

4.0.8 Scalability Requirements

- **NFR-052**: The system shall scale horizontally to support up to 10,000 concurrent users
- **NFR-053**: The system shall support tracking of up to 500 buses simultaneously
- **NFR-054**: The database shall handle up to 1 million location records per day
- **NFR-055**: The system shall support expansion to multiple cities without architectural changes
- **NFR-056**: The system shall implement caching strategies to reduce database load

4.0.9 Compatibility Requirements

- **NFR-057**: The system shall work with GPS accuracy of ±10 meters
- **NFR-058**: The system shall function on networks with minimum 256 kbps bandwidth
- **NFR-059**: The system shall be compatible with various GPS tracking device manufacturers
- **NFR-060**: The system shall integrate with existing transportation management systems via APIs

4.0.10 Legal and Compliance Requirements

- **NFR-061**: The system shall comply with Pakistan's Personal Data Protection Bill requirements
- **NFR-062**: The system shall obtain explicit user consent for location tracking
- **NFR-063**: The system shall provide users ability to delete their personal data
- **NFR-064**: The system shall maintain data privacy in accordance with Islamic principles
- **NFR-065**: The system shall comply with Google Maps API terms of service
- **NFR-066**: The system shall include proper attribution for third-party services

4.0.11 Environmental Requirements

- **NFR-067**: The mobile application shall optimize battery consumption to maximum 5% per hour
- **NFR-068**: The system shall optimize data usage to consume maximum 10MB per hour for users
- **NFR-069**: The system shall reduce server power consumption through efficient algorithms

5 External Interface Requirements

5.0.1 Hardware Interfaces

- **HW-001**: The system shall interface with GPS sensors installed in mobiles of drivers via HTTP/HTTPS protocol
- **HW-002**: The system shall receive location data in standard GPS format (latitude, longitude, altitude, speed)
- **HW-003**: The system shall accept location updates at intervals of 5-10 seconds
- **HW-004**: The system shall support touchscreen input on mobile devices

5.0.2 Software Interfaces

- **SW-001**: The system shall integrate with Google Maps API v3 for map display and routing
- **SW-002**: The system shall use Google Maps Directions API for route planning and EAT calculation
- **SW-003**: The system shall use MongoDB database for storing:
  - Driver accounts
  - Administrator accounts
  - Bus information and assignments
  - Route and schedule data
  - Previous schedules and routes
- **SW-004**: The system shall use Redis cache for storing:
  - Current bus locations (real-time data)
  - Active user sessions
  - Frequently accessed route information
- **SW-005**: The system shall run on Node.js v18+ runtime environment
- **SW-006**: The system shall use Express.js framework for RESTful API development
- **SW-007**: The system shall implement Socket.io for WebSocket communication
- **SW-008**: The system shall be compatible with Flutter SDK 3.0+

5.0.3 Communication Interfaces

- **COM-001**: The system shall communicate via HTTPS protocol for all API requests
- **COM-002**: The system shall use WebSocket (Socket.io) for real-time location updates
- **COM-003**: The system shall transmit data in JSON format
- **COM-004**: The system shall implement JWT-based authentication for API security
- **COM-005**: The system shall support IPv4 and IPv6 protocols
- **COM-006**: The system shall work with minimum bandwidth of 256 kbps
- **COM-007**: The system shall implement connection retry logic with exponential backoff
- **COM-008**: The system shall compress data transmission using gzip compression

6 Appendices

6.1 Appendix A: Use Case Diagram

Description:

The Use Case Diagram illustrates the primary actors and their interactions with the system:

Actors:

- **Passenger**: Who track buses and view information
- **Driver**: Bus operators who start duties, view duties, and share location
- **Administrator**: System managers who monitor and control operations
- **GPS System**: External system providing location data
- **Google Maps API**: External service for mapping and routing
- **PMA Database**: Records every activity and gives access to data when needed

Key Use Cases:

Driver and Admin Combined Functions:

- User Authentication and Authorization (FR-001 to FR-008)
- Profile Management (FR-009)

Passenger Functions:

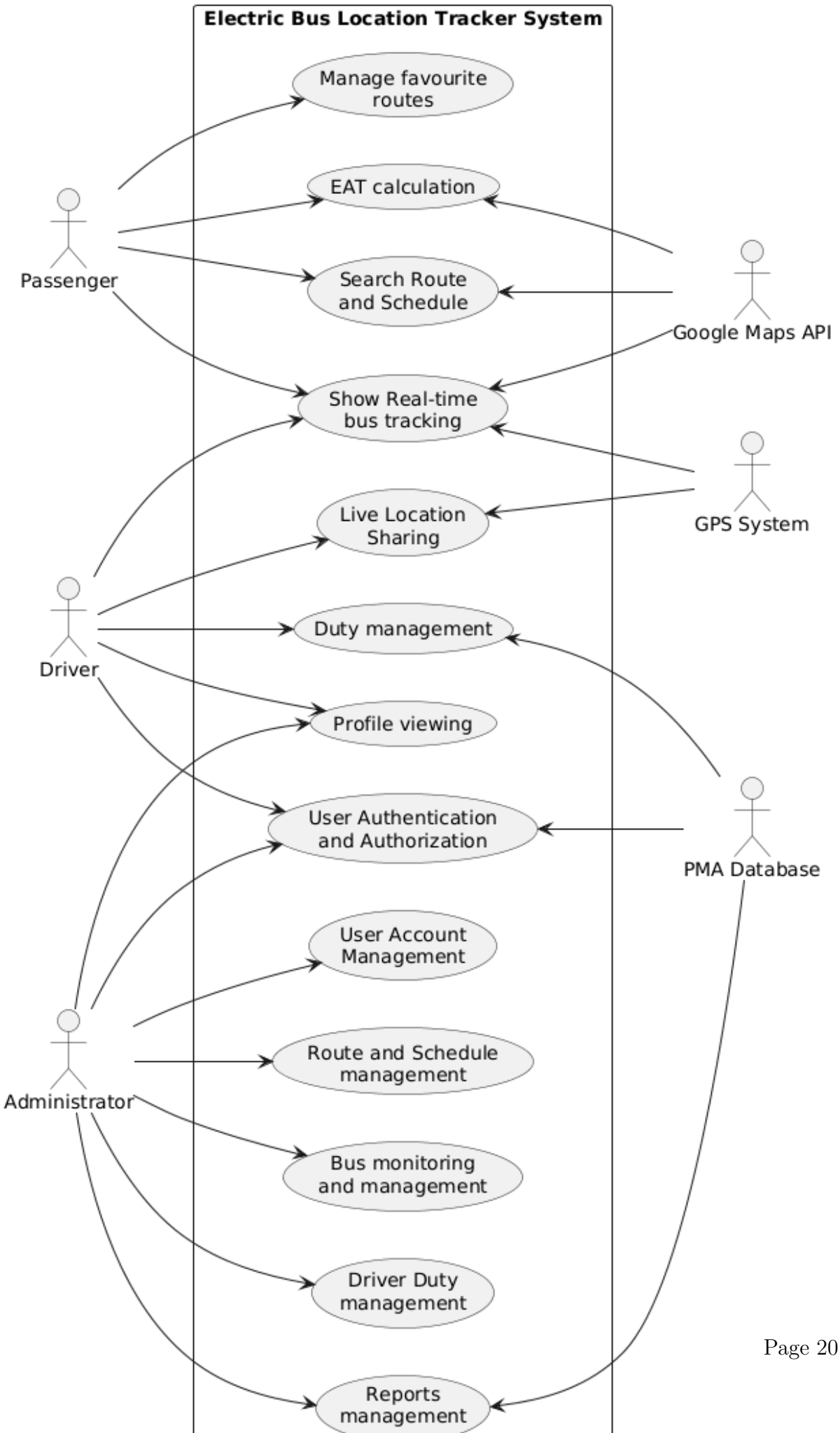
- Real-time Bus Tracking (FR-010 to FR-013)
- Search Routes and Schedules (FR-014 to FR-022)
- Estimated Arrival Time (EAT) (FR-023 to FR-028)
- Manage Favorite Routes (FR-029 to FR-032)

Driver Functions:

- Duty Management (FR-033 to FR-041)
- Live Location Sharing (FR-042 to FR-048)

Administrator Functions:

- Bus Monitoring and Management (FR-049 to FR-054)
- Route and Schedule Management (FR-055 to FR-060)
- User Account Management (FR-061 to FR-066)
- Driver and Duty Assignment (FR-067 to FR-071)
- Reports and Record (FR-072 to FR-077)


Electric Bus Location Tracker System
Manage favourite routes
EAT calculation
Search Route and Schedule
Show Real-time bus tracking
Live Location Sharing
Duty management
Profile viewing
User Authentication and Authorization
User Account Management
Route and Schedule management
Bus monitoring and management
Driver Duty management
Reports management
Passenger
Driver
Administrator
Google Maps API
GPS System
PMA Database

6.2 Appendix B: Context Diagram

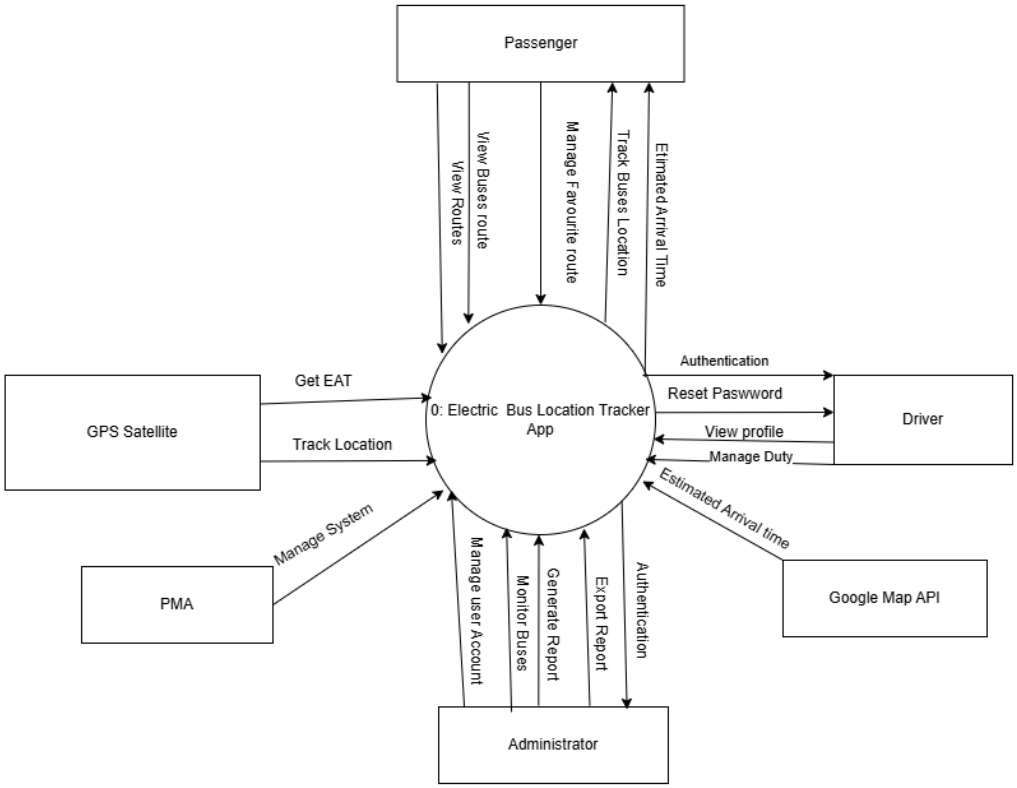


Figure 2: Context Diagram for Electric Bus Location Tracker App

Description:

The Context Diagram shows the Electric Bus Location Tracker System as a central entity with its external interfaces:

External Entities:

- **Passengers**: Receive real-time bus information, routes, schedules, and EAT

- **Drivers**: Send GPS location data, view duties; receive duty assignments and view all previous skipped and completed duties
- **Administrators**: Monitor system, manage routes/schedules/buses, receive reports
- **GPS Satellites**: Provide positioning data for accurate tracking
- **Google Maps API**: Provides mapping services, route calculations, and EAT data
- **PMA (Punjab Mass Transit Authority)**: Registers administrators and gives access to data from their database

6.3 Appendix C: Technology Stack Summary

Component	Technology	Purpose
Mobile Framework	Flutter	Cross-platform app development for Android
Backend Framework	Node.js + Express.js	Server-side application logic
Database	MongoDB	Primary data storage
Cache	Redis	Real-time data caching
Mapping Service	Google Maps API	Maps, routing, ETA calculation
Real-time Communication	Socket.io	Live location updates
Cloud Hosting	AWS (EC2, S3)	Server and file hosting
Authentication	JWT + bcrypt	Secure user authentication
API Architecture	RESTful + WebSocket	Client-server communication
Version Control	GitHub	Code repository management

Table 1: Technology Stack Components

6.4 Appendix D: Glossary

Bus Stop A designated location where buses pick up and drop off passengers

Duty Assigned work shift for a driver including route and bus assignment

EAT (Estimated Arrival Time) Calculated time when a bus will reach a specific stop

GPS Coordinates Latitude and longitude values representing a geographical location

PMA Punjab Mass Transit Authority - Government body managing public transport

Real-time Tracking Continuous monitoring of bus locations with minimal delay

Route Predefined path that a bus follows with designated stops

Schedule Planned timing for bus departures and arrivals

Session Period during which a user remains logged into the system

Stakeholder Individual or organization with interest in the project

WebSocket Protocol enabling real-time bidirectional communication