

Revised Project Division Among 5 Members

Member 1: Authentication, User Management & Admin Web Panel

Mobile Frontend (React Native):

- Login/Register screens for all user types (Passenger, Driver)
- Social login (Google, Facebook OAuth)
- Profile management screens (view/edit profile)
- Password reset and forgot password screens
- Splash screen and onboarding flow (first-time users)
- Settings screen (preferences, theme, language)
- User verification badges (verified drivers, verified corporate accounts)
- Multi-language support (English, Sinhala, Tamil)
- Biometric authentication (fingerprint/face ID) - if time allows
- Two-factor authentication (2FA) setup screens
- **Corporate user registration flow** (organization details, business verification)

Web Frontend (React - Admin Dashboard):

- Admin authentication and dashboard
- User management (view, edit, suspend, delete users)
- Corporate account verification and management
- Fleet management (add/edit/delete buses)
- Route management interface
- Complaint management dashboard
- System settings and configuration
- Analytics dashboard (user stats, booking trends)
- Revenue and commission reports
- Driver verification and approval system

Backend (Spring Boot):

- User registration and login APIs (all user types)
- JWT token generation and validation
- Password encryption (bcrypt/hashing)
- OTP generation and verification APIs
- Session management APIs
- Role-based authorization middleware (PASSENGER, DRIVER, ADMIN, CORPORATE_USER)
- Social login integration (Google, Facebook OAuth)
- Corporate account verification APIs
- Admin dashboard data APIs
- User management APIs (CRUD operations)
- System configuration APIs

Database (MySQL):

- Users table (id, name, email, phone, password_hash, role, user_type, created_at, verified)
- Sessions table (session_id, user_id, token, device_info, expires_at)
- OTP_verification table (user_id, otp, expires_at, verified)
- User_profiles table (user_id, profile_pic, address, preferences, language)
- Corporate_accounts table (corporate_id, user_id, company_name, business_reg_no, contact_person, verification_status, agreement_details)
- Admin_users table (admin_id, name, email, role, permissions)

Scope Enhancements:

- Multi-device session management
- Audit logs for admin actions
- Corporate account tiered permissions

Member 2: Real-Time Bus Tracking & Map Integration

Mobile Frontend (React Native):

- Interactive map screen in Passenger app (all user categories)
- Live tracking interface with custom bus icons and user location marker
- Route selection and bus details display
- Traffic visualization overlay on map
- Live ETA calculation and arrival countdown timer
- Route and stop filtering (by bus type, route name)
- Bus stop markers with names
- Share live location feature
- Historical route dashboard (past routes traveled)
- **Geofencing for bus stops** (Highway & Non-Highway AC buses)
 - Auto-notification when bus enters/exits stop zone
 - Automatic arrival status updates
- Multi-bus tracking view (track multiple buses simultaneously)

Backend (Spring Boot):

- Real-time location update APIs (WebSocket/Socket.io)
- GPS coordinate processing and validation
- Location data storage with timestamp
- Route calculation APIs (distance, duration)
- Google Maps API integration (traffic data, directions)
- ETA calculation algorithm (based on speed, traffic, historical data)
- Geofencing logic (bus stop proximity detection)
- Location history APIs (for analytics)
- Traffic data aggregation and caching
- Map tile caching for performance

Database (MySQL):

- Buses table (bus_id, registration_no, type, capacity, driver_id, status, current_location)
- Routes table (route_id, route_name, start_point, end_point, stops[], distance_km)
- Bus_locations table (bus_id, latitude, longitude, speed, heading, timestamp, accuracy)
- Bus_stops table (stop_id, stop_name, latitude, longitude, route_id, geofence_radius)
- Traffic_data table (route_id, segment, traffic_level, timestamp, source)
- Location_history table (bus_id, latitude, longitude, timestamp) - for analytics

Scope Enhancements:

- Favorite routes and stops (if time permits)
- Predictive ETA using AI/ML based on historical traffic patterns
- Route optimization suggestions for drivers
- Heatmap of bus density in areas
- Offline map caching for low connectivity areas

Member 3: Booking Systems (Seat & Bus Booking) + Corporate Dashboard

Mobile Frontend (React Native):

Highway Bus & Long distance Bus Users:

- Bus selection with route and timing display
- Visual seat layout with availability status
- Seat selection (single/multiple seats)
- AC/Non-AC filter
- Booking summary screen
- Payment gateway integration UI (PayHere)
- QR code ticket generation and download

- Booking history list
- Upcoming journey details
- Ticket cancellation with refund conditions display
- Promo code input and discount application for regular customers

Trip Booking Users:

- Available buses display with details (condition, capacity, amenities)
- Driver information and ratings
- Start location, destination, date picker
- Number of days selector
- Estimated price calculation display
- Agent contact notification screen
- In-app negotiation status tracker
- Final amount confirmation screen
- Payment gateway integration
- Trip booking history

Special/Corporate Booking Users:

- Available buses with detailed specifications
- Organization details input form
- Seat capacity, AC/Non-AC filters
- Multiple bus selection for large organizations
- Agreement preview screen
- Pending booking status display
- Agent contact notification
- Contract duration and route management
- Payment processing for long-term bookings
- **Corporate Booking Dashboard (NEW):**

- Active contracts overview
- Daily/Monthly bus allocation calendar
- Employee attendance tracking per bus
- Monthly billing and payment history
- Driver assignment view
- Route modification requests
- Complaint filing specific to corporate bookings
- Usage analytics (buses used, peak times)
- Contract renewal reminders
- Invoice download and payment receipts

Backend (Spring Boot):

- Seat availability checking APIs (real-time)
- Seat booking creation and validation logic
- Bus booking/rental APIs (full bus)
- Trip booking APIs with pricing logic
- Corporate booking APIs with contract management
- Price calculation engine (per km, per day, duration-based)
- PayHere payment gateway integration
- Booking confirmation and ticket generation APIs
- QR code generation for tickets
- Cancellation and refund processing APIs (with conditions)
- Booking modification APIs
- Promo code validation and discount application
- Agent notification trigger APIs
- **Corporate dashboard data APIs:**
 - Contract details and status

- Daily allocation and attendance APIs
- Billing calculation and invoice generation
- Usage analytics and reporting
- Contract renewal management

Database (MySQL):

- Seat_bookings table (booking_id, passenger_id, bus_id, seat_no, route_id, booking_date, journey_date, amount, status, promo_code)
- Bus_bookings table (booking_id, passenger_id, bus_id, start_location, end_location, start_date, end_date, total_amount, status, booking_type)
- Corporate_bookings table (booking_id, corporate_id, contract_start, contract_end, buses[], routes[], monthly_amount, status, agreement_doc)
- Corporate_daily_allocation table (allocation_id, corporate_id, booking_id, bus_id, date, route, employee_count, attendance_status)
- Payments table (payment_id, booking_id, amount, payment_method, transaction_id, status, timestamp, invoice_url)
- Seats table (seat_id, bus_id, seat_number, seat_type, is_available, booking_id)
- Pricing table (route_id, bus_type, base_price, per_km_rate, per_day_rate, corporate_rate)
- Promo_codes table (code, discount_percentage, valid_from, valid_to, usage_limit, usage_count)
- Invoices table (invoice_id, corporate_id, booking_id, month, amount, payment_status, generated_at, paid_at)

Scope Enhancements:

- Recurring bookings for regular commuters
- Digital wallet integration
- Flexible payment plans for corporate clients
- Auto-renewal option for corporate contracts

Member 4: Real-Time Notifications, Communication & Event Management

Mobile Frontend (React Native):

- **Unified notification center (all alerts, messages, updates)**
- Push notification handling (foreground & background)
- Notification preferences screen (customize notification types)
- Journey reminder notifications
- **Emergency SOS button** with auto-location sharing
 - ~~One tap SOS activation~~
 - Emergency contacts notification
 - Admin alert dashboard integration
- **Real-time chat module:**
 - Passenger ↔ Driver chat (~~within active booking~~)
 - Passenger ↔ Admin chat
 - Corporate User ↔ Admin/Agent chat (~~for negotiations~~)
 - Chat conversation list
 - Individual chat screens
 - Typing indicator display
 - Read receipt indicators (single/double tick)
 - Voice message recording and playback
 - Image attachment with camera/gallery integration
 - Media compression preview
 - Message status indicators (sent, delivered, read)
- Geo-fenced alert popup (approaching destination notification)
- SMS and Email notification status display
- Notification history ~~with filtering~~

Backend (Spring Boot):

- **Push notification system:**
 - Firebase Cloud Messaging (FCM) integration
 - Device token management
 - Notification scheduling and dispatch
 - Bulk notification sending (for corporate users)
- **SMS integration:**
 - Twilio API integration
 - OTP delivery via SMS
 - Booking confirmation SMS
 - Emergency alert SMS
- **Email service:**
 - SMTP configuration
 - Transactional emails (booking confirmations, invoices)
 - Corporate contract emails
 - Password reset emails
- **Real-time chat backend:**
 - WebSocket server setup
 - Chat message routing
 - Message persistence
 - Delivery status tracking
 - Read receipt management
 - Typing indicator broadcast
- **Media handling:**
 - Image upload and storage
 - Voice message storage

- Media compression logic (bandwidth-based)
 - **CDN integration for media delivery**
- **Event orchestration:**
 - Scheduled event system (journey reminders, contract expiry alerts)
 - Geo-fence event triggers
 - Booking status change events
 - SOS alert handling and routing
- ~~Notification preference management APIs~~
- ~~Notification analytics APIs (delivery rate, open rate)~~

Database (MySQL):

- Notifications table (notification_id, user_id, type, title, message, read_status, created_at, delivery_status)
- Messages table (message_id, sender_id, receiver_id, booking_id, message_text, message_type, media_url, timestamp, read_status, delivered_status)
- Chat_rooms table (room_id, passenger_id, driver_id, booking_id, created_at, last_message_at, status)
- Notification_preferences table (user_id, push_enabled, sms_enabled, email_enabled, notification_types[], quiet_hours)
- Emergency_alerts table (alert_id, user_id, location, timestamp, status, admin_response, resolved_at)
- Scheduled_events table (event_id, user_id, event_type, scheduled_at, executed, data)
- Media_uploads table (media_id, user_id, file_url, file_type, file_size, compressed_url, uploaded_at)

Scope Enhancements:

- ~~Read receipts and typing indicators (already included)~~
- Notification analytics dashboard
- ~~Scheduled notification templates~~

- Multi-language notification support
- Voice call integration (if time permits)

Member 5: Driver Dashboard, Feedback, Rating & Complaint Management

Mobile Frontend (React Native - Driver App):

- **Driver Dashboard:**
 - Today's earnings summary
 - Active trips and bookings count
 - Upcoming bookings timeline
 - Total trips completed (lifetime)
 - Average rating display
 - Monthly earnings graph
 - ~~Notifications badge~~
- Driver profile management
 - Personal details (name, photo, license info)
 - Vehicle assignment
 - Experience and certification
- **Booking management:**
 - ~~Incoming booking notifications~~
 - ~~Booking acceptance/rejection screen~~
 - Seat/bus availability toggle controls
 - Current booking details with passenger info
 - Booked seat layout visualization
 - ~~Passenger contact information~~
- Trip history with filtering (date, route)
- ~~Earnings breakdown (trip-wise, commission deducted)~~
- **Ratings and reviews:**

- View passenger ratings and reviews
- ~~Average rating trends~~
- Feedback details
- ~~Navigation assistance to pickup/drop points~~

Mobile Frontend (React Native - Passenger App):

- **Feedback and rating screen:**
 - Driver rating (1-5 stars)
 - Bus condition rating
 - Journey experience rating
 - Text review input
 - ~~Photo upload for complaints (if applicable)~~
- **Complaint filing:**
 - ~~Complaint type selection (driver behavior, bus condition, booking issue, payment issue)~~
 - Description text area
 - ~~Booking reference auto-fill~~
 - ~~Image/video attachment~~
 - Submit and tracking
- **Complaint tracking:**
 - View submitted complaints
 - Status updates (pending, under review, ~~resolved~~)
 - Admin response viewing

Backend (Spring Boot):

- Driver dashboard data aggregation APIs
- ~~Booking acceptance/rejection APIs~~
- Seat/bus availability toggle APIs

- Trip history APIs with filters
- Earnings calculation APIs (trip-wise, monthly)
- ~~Commission calculation logic~~
- **Feedback and rating APIs:**
 - Rating submission
 - Review posting
 - Average rating calculation
 - ~~Rating analytics per driver~~
- **Complaint management APIs:**
 - Complaint submission
 - ~~Complaint categorization (automated or manual)~~
 - Status update APIs
 - Admin response APIs
 - ~~Complaint resolution workflow~~
- ~~Driver performance analytics APIs~~
- Revenue reporting APIs (for admin)
- ~~Driver leaderboard APIs (top-rated drivers)~~

Database (MySQL):

- Driver_profiles table (driver_id, user_id, license_no, license_expiry, experience_years, rating, total_trips, total_earnings, status, verified, bank_account)
- Ratings table (rating_id, booking_id, passenger_id, driver_id, bus_id, driver_rating, bus_rating, journey_rating, review_text, images[], timestamp)
- Complaints table (complaint_id, user_id, booking_id, complaint_type, category, description, images[], status, priority, admin_id, admin_response, created_at, resolved_at)
- Driver_earnings table (driver_id, booking_id, trip_date, gross_amount, commission_percentage, commission_amount, net_earnings, payment_status, paid_at)

- Reports table (report_id, report_type, generated_by, filters, data, created_at)
- Driver_analytics table (driver_id, date, trips_completed, earnings, avg_rating, acceptance_rate)

Scope Enhancements:

- Automated complaint categorization (ML-based)
- Bus maintenance tracking (schedule, history)
- Driver recruitment handling via app
- Performance-based incentive tracking
- Driver training module integration

Technology Stack (Unchanged):

- **Frontend:** React Native (Mobile Apps) + React.js (Admin Web Dashboard)
- **Backend:** Java Spring Boot
- **Database:** MySQL
- **Real-time:** WebSocket/Socket.io
- **Maps:** Google Maps API
- **Payment:** PayHere
- **Notifications:** Firebase Cloud Messaging
- **SMS:** Twilio
- **Storage:** Cloud storage for media files