# **Building a Real-time Bus Tracking Application:**

The goal is to create a user-friendly bus application that helps different actors (entities/roles) with certain privileges or permissions to use real-time data to manage bus operations or manage routes effectively for better user experience in buses . the application is used by different entities like :

* End Users/Commuters – for booking tickets and planning travels according to travel routes and arrival/departure timings and getting live bus status from drivers.
* Drivers/Conductors – for updating live location of buses , updating routes/stops as per passenger demand and checking passenger count , ticket validation etc and updating real-time bus status to commuters.
* Depot managers – for creating/managing fleets of buses and managing bus routes/services in case of breakdown and updating to drivers and conductors .
* Admins – Oversee bus operations and Performance analytics using databases and update bus depots or report to technical team for application errors.
* Auditor – to oversee transactions , compliance with bus travel standards . to oversee financial accounts and bus operations logs .
* Support Staff – Ticketing team for enquiries and incident management.
* Suppliers – Ensure that parts and supplies are being given to the bus for services .

# What is the need for such an application?

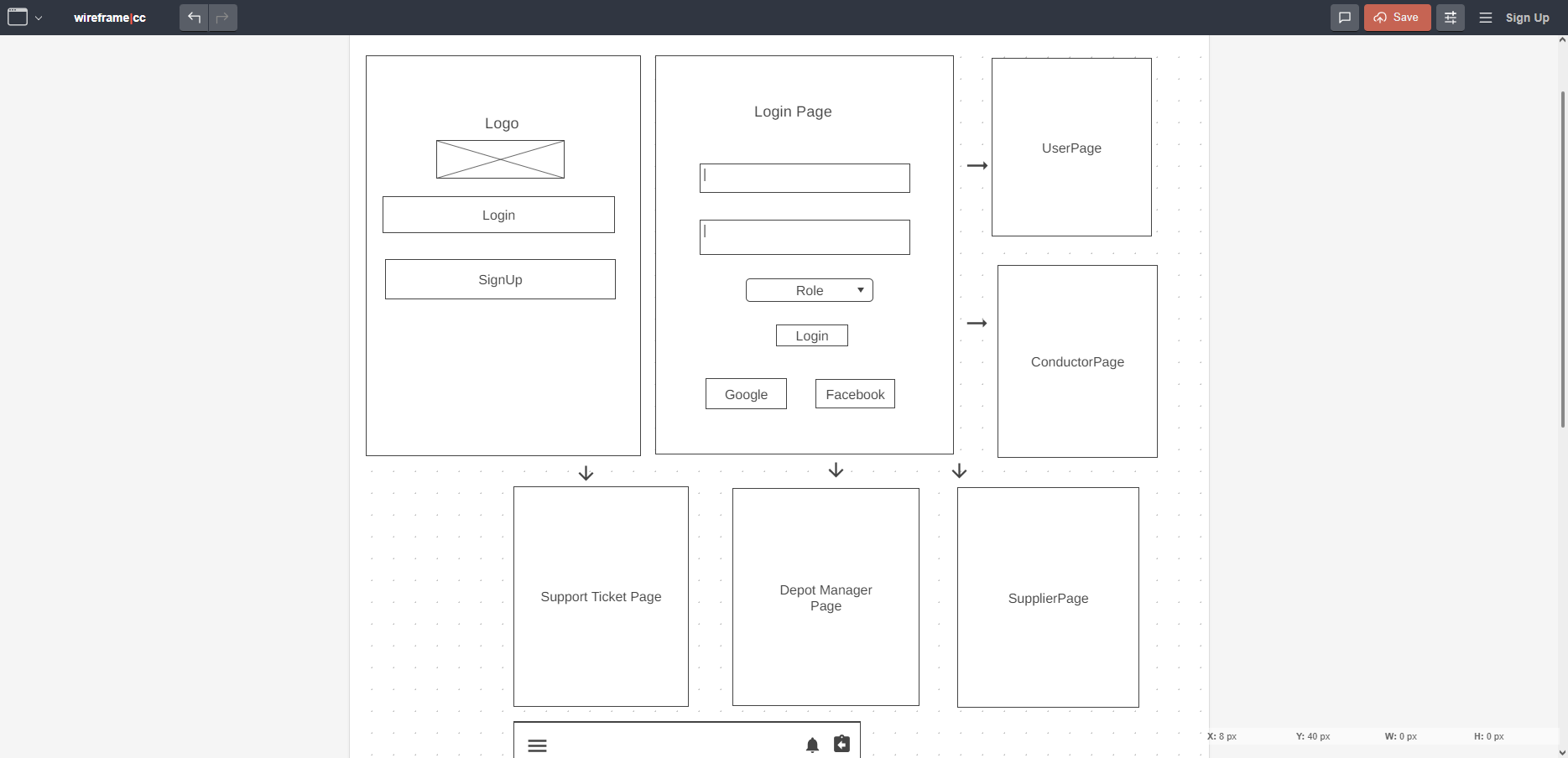
The need for a real time application such as a bus tracking app is to plan/manage bus travels effectively as per arrivals and departure timings , manage/update routes effectively for less operational costs , updating live information to commuters in case of breakdown or service or sharing live location of buses to alert delays and bus stops in different locations . As Admins , we also need to oversee real time bus operations using databases. inefficient route management can lead to operational costs . without real time information , it is unclear when the bus might arrive , leading to frustration among passengers . if the bus breaks down because of frequent usage , without quality check depot managers might find it difficult to manage bus services / travels to different routes. Auditors rely on real-time data to verify operations and compliance. Without it, the auditing process becomes more challenging and less reliable. The absence of real-time tracking and management applications can lead to a host of operational, financial, and safety issues. These side effects can severely impact the efficiency and reliability of bus services, leading to a poor passenger experience, higher costs, and potential safety risks.

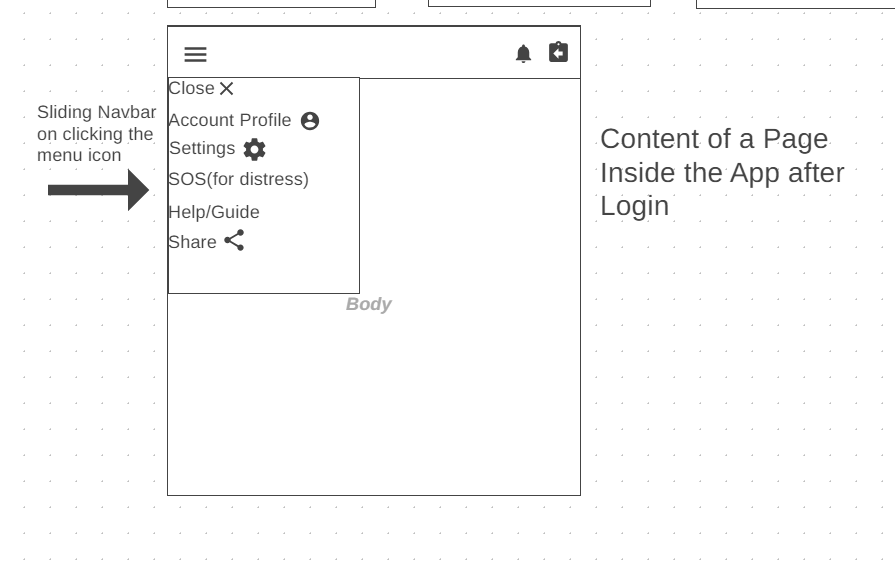
# Approach :

We can make this into a cross platform , PWA App using React Native tools .

* The first step is to create a login/registration page to authenticate ,validate and finally direct each entity to different UI based on their roles .
* If the user is new , he registers and if he registers with a new role , the actor/entity is verified by the admin as per the details given and shall be approved by the admin for the role.
* For End Users , they shall have a UI that can help them to look into different bus routes and bus arrivals/departure timings for planning bus travels effectively and manage their booking tickets . when they book them they shall get the live updates of bus from their pick up location. They will need booking options .
* For Drivers / Conductors , they will have a UI that will enable them to stop/start the travel , to update live location of their bus to commuters , for ticket validation and confirmation .
* For depot managers , UI that Manages fleet schedules, maintenance, and performance analytics
* For Auditors , UI to look into logs of transactions , compliance with bus travel standards and operations
* For Admins , UI to oversee entire bus operations and real time analytics to look into bus performance statistics and reporting system to technical team for technical issues .
* For Support Staffs , Incident management tools and UI to get tickets and enquiries from other entities for resolving issues .
* Dropdown menu , for navigating to different pages/views in UI.
* Logout Button.
* Real-time data can be achieved by integrating with web sockets , NoSQL tools like MongoDB , fetch APIs with google maps .

# **WireFrame:**



****

# **Workflow :**

### **Passenger/End-user** :

 **Login/Registration:**

* Passenger registers and logs into the app.

 **Search and Book a Bus:**

* Passenger searches for available buses based on the desired route.
* The app shows a list of buses, with real-time availability.
* Passenger selects a bus, books a ticket, and receives confirmation.

 **Track Bus in Real-time:**

* Passenger can view the real-time location of the bus after booking.(approx. every 10 secs)
* Notifications are sent for any changes (e.g., delays).

 **Bus Boarding:**

* Passenger checks in with their ticket (e.g., QR code).

So for End-User we have the following UI:

* Login/Register Screen
* Search Bus Screen/Booking Screen
* Real-time Tracking Screen
* Notification Centre

### **Conductor/Driver** :

 **Login:**

* Conductor logs in with their credentials.

 **Passenger Check-ins:**

* Conductor scans passenger tickets as they board the bus.

 **Update Bus Status:**

* Conductor updates the bus status at different points (e.g., departed, arrived).

 **Report Issues:**

* Conductor reports any issues via the app (e.g., delays, incidents).

**Conductor/Driver Components:**

* Login Screen
* Passenger Check-in Screen
* Bus Status Update Screen
* Issue Reporting Screen

## **Depot Manager :**

 **Monitor Buses:**

* Depot Manager logs in to monitor all buses under their depot.

 **Assign Buses to Routes:**

* Assign buses and drivers to specific routes.

 **Manage Schedules:**

* Create and modify schedules for buses.

 **View Reports:**

* Review operational reports and make necessary adjustments.

**Depot Manager Components:**

* Dashboard for monitoring
* Route and Schedule Management Screen
* Reports Screen

## **Auditor :**

 **Login:**

* Auditor logs in to access audit logs.

 **Review Logs:**

* Auditor reviews logs related to bus operations, ticketing, and schedules.

 **Generate Reports:**

* Auditor generates compliance and performance reports.

**Auditor Components:**

* Login Screen
* Audit Logs Screen
* Report Generation Screen

## **Administrator :**

 **Manage Users:**

* Administrator manages user roles and permissions.

 **Oversee Operations:**

* Monitor the overall system’s performance.

 **Security and Compliance:**

* Ensure that the system is secure and compliant with regulations.

**Administrator Components:**

* User Management Screen
* System Monitoring Dashboard
* Security Settings Screen

## **Support Staff :**

 **Login & Authentication:**

* Support staff logs into the app or support dashboard using their credentials.
* They are authenticated and redirected to their dashboard.

 **Dashboard Overview:**

* The dashboard displays a list of current user queries, issues, and feedback.
* Support staff can filter or search for specific issues based on categories like "Real-Time Tracking," "Booking," "Technical Errors," etc.

 **Handling Queries:**

* Support staff clicks on a user query to view details and provide assistance.
* They can respond directly through the app or escalate the issue if it's beyond their capability.

 **Issue Tracking:**

* Ongoing issues are tracked in real-time with status updates (e.g., "In Progress," "Resolved").
* Support staff can update the status or assign the issue to another team member if necessary.

 **Escalation Process:**

* For technical issues, support staff can escalate the problem to the IT or development team.
* They can attach detailed reports or logs to assist in the resolution.

 **Feedback Management:**

* Support staff reviews user feedback and categorizes it for further analysis.
* They can forward feedback to the product or development team to inform future updates.

 **Reporting & Analysis:**

* Periodically, support staff generates reports on common issues, user satisfaction, and feedback trends.
* These reports are shared with the management and development teams to improve the app.

**Support Staff UI consists of :**

* Login Screen
* Bus Fleet Management Screen
* Dashboard
* Support Analytics Screen
* Real Time Bus Tracking Screen
* Incident Reporting & Resolution Screen
* User Management Screen
* Support Ticket System
* Reports and Analytics Screen – to view and generate reports on various aspects of bus operations.
* Settings Screen – for configuring app’s operational settings
* Help & Documentation Screen – provide staff with guidelines & documentation

It seems to have the most no. of UI Screens to handle overall aspects of the app operations .

## **Supplier :**

**Supplier Workflow:**

1. **Login:**
   * Supplier logs in to manage bus maintenance and part deliveries.
2. **Update Maintenance Records:**
   * Supplier updates the app with the status of maintenance tasks or parts deliveries.
3. **Report on Repairs:**
   * Supplier provides updates on ongoing or completed repairs.

**Supplier Components:**

* Maintenance Management Screen
* Part Delivery Status Screen
* Repair Status Update Screen