



REPUBLIC OF CAMEROON  
PEACE-WORK-FATHERLAND  
UNIVERSITY OF BUEA



FACULTY OF ENGINEERING AND  
TECHNOLOGY  
DEPARTMENT OF COMPUTER ENGINEERING

## Task 5: Car Fault Diagnosis Mobile Application UI Design and Implementation Report

### Course Information

**Course Title:** Internet and Mobile Programming

**Course Code:** CEF440

**Course Instructor:** Dr. Nkemeni Valery

### Group Members

LEKEUGO DEMELIEU ROCHINEL	FE22A237
CHUYE PRINCELY TATA	FE22A184
DIONE CHANCELINE NZUO SONE	FE22A187
ACHUO ESEIGNI	FE22A135
MISHAEL ZABUD	FE22A248

# Table of Contents

<b>Car Fault Diagnosis Mobile Application.....</b>	<b>3</b>
<b>Introduction.....</b>	<b>3</b>
<b>1. App Identity.....</b>	<b>3</b>
<b>Brand Attributes:.....</b>	<b>4</b>
<b>Logo and Theme:.....</b>	<b>4</b>
<b>2. Visual Design .....</b>	<b>5</b>
<b>2.1. Layout Strategy:.....</b>	<b>5</b>
<b>Color and Font Choices .....</b>	<b>5</b>
a. <b>Color Palette for light mode: .....</b>	<b>5</b>
b.     🌙 <b>Dark Mode.....</b>	<b>6</b>
c. <b>Typography:.....</b>	<b>6</b>
<b>2.2. Explaining the various pages .....</b>	<b>7</b>
A. <b>Login Screen – User Authentication Entry Point.....</b>	<b>7</b>
B. <b>Sign Up → Select User Role (Mechanic or Car Owner) .....</b>	<b>7</b>
C. <b>Home Screen Analysis:.....</b>	<b>9</b>
<b>4. Top-Rated Mechanics Section .....</b>	<b>9</b>
❖ <b>Notification Icon:.....</b>	<b>14</b>
❖ <b>User Profile: .....</b>	<b>15</b>
❖ <b>Mechanic Verification &amp; Dashboard.....</b>	<b>16</b>
<b>Conclusion and Design Assessment .....</b>	<b>18</b>
<b>Summary of First Batch Flows .....</b>	<b>18</b>
<b>3. Frontend Implementation .....</b>	<b>19</b>
<b>3.1 Structure and Componentization .....</b>	<b>19</b>
<b>3.2 Tools and Technologies Used .....</b>	<b>20</b>
<b>3.3 Styling and Theming.....</b>	<b>22</b>
<b>3.4 Performance and Optimization .....</b>	<b>22</b>
<b>3.5 Development Workflow .....</b>	<b>23</b>

# UI Design and Implementation Report

## Car Fault Diagnosis Mobile Application

### Introduction

The **User Interface (UI) Design and Implementation Phase** represents a critical stage in the software development lifecycle where the functional and technical requirements gathered in earlier phases are translated into a practical, interactive, and user-friendly mobile experience. In this task, we focus on designing and building the **visual and operational structure** of the mobile application for car fault diagnosis.

The **objective** of this phase is to ensure that users—including both **car owners** and **mechanics**—can seamlessly interact with the app's features such as scanning dashboard symbols, recording engine sounds, receiving AI-driven diagnoses, giving feedback, and contacting professionals. To achieve this, the task is structured into three subcomponents:

1. **App Identity Design** – Defining the name, branding, and user perception of the application.
2. **UI/UX Visual Design** – Creating a visual layout that is intuitive, responsive, and tailored to the target user base.
3. **Frontend Implementation** – Transforming design prototypes into functional screens using appropriate mobile development technologies.

This phase bridges the gap between backend functionality and end-user experience. It prioritizes **usability, accessibility, clarity, and responsiveness**, ensuring that the final product is not only technically sound but also easy and enjoyable to use. Each design decision is guided by the app's primary goals: rapid fault diagnosis, improved user trust, and simplified access to automotive help through modern mobile technology.

### 1. App Identity

The app identity is the foundational element that defines the personality, purpose, and emotional connection of the mobile application with its users. For this phase, we introduce the official name for our mobile app: **Car First Aid**.

The name “Car First Aid” was chosen to communicate the core value of the app—providing immediate, accessible, and intelligent diagnostic support to car owners, much like a first aid kit

does in medical emergencies. It suggests that the app offers quick, reliable help for car problems, reducing panic and empowering users with actionable knowledge.

The app is primarily targeted toward **non-technical car owners**, offering AI-driven analysis of dashboard lights and engine sounds, and connecting them to certified mechanics. The secondary audience includes **mechanics**, who can create professional profiles, receive ratings, and expand their reach.

### **Brand Attributes:**

#### **Trustworthy:**

The app uses AI-powered diagnosis and connects users to verified, certified mechanics. This builds user confidence in both the system and the professionals recommended.

#### **Helpful:**

It provides clear suggestions, possible causes, and links to video tutorials. Users receive step-by-step guidance, making fault resolution easier.

#### **Simple:**

Designed with a clean layout and easy navigation for all user types. Even non-technical users can perform tasks in just a few intuitive steps.

#### **Modern:**

The app leverages smartphone sensors like the camera and microphone. It uses advanced AI models for real-time image and sound analysis.

### **Logo and Theme:**



The Car First Aid logo features a sleek car silhouette with a road stripe symbolizing motion and diagnostics. The surrounding orange wings represent speed and emergency response, while the semi-circular arcs suggest protection and support. The clean, bold typography reinforces clarity and professionalism.

Overall, the logo effectively conveys the app's purpose: fast, reliable car fault diagnosis and assistance.

## 2. Visual Design

Visual design is at the heart of the user experience (UX). It governs how users interact with the application, from readability and navigation to emotional comfort and aesthetic value.

We designed the visual experience to reflect **clarity, friendliness, and speed**. Using simple layouts, large touch-friendly buttons, and consistent iconography, the interface is both practical and visually appealing.

### 2.1. Layout Strategy:

The app uses a **card-based layout**, where each major feature (scan dashboard, upload image, record sound, give feedback) is grouped in blocks with intuitive icons and minimal text.

#### 2.2 Key Screens:

##### 1. Home Screen:

Features quick access to all main actions like scanning the dashboard, uploading an image, and giving feedback.

##### 2. Diagnostic Result Screen:

Displays recognized faults, confidence scores, possible causes, and embedded tutorial videos.

##### 3. History, Diagnostic Results and Feedback Screens:

Shows community History of the previous Diagnostics, Diagnostic Results and feedbacks and allows users to rate their experience Diagnosis and on mechanics

##### 4. Mechanic Profile Screen:

Allows users to view verified mechanic details and contact them directly.

## Color and Font Choices

### a. Color Palette for light mode:

The application uses a friendly and trustworthy visual identity through these colors:

- **Primary Background:** Soft Light Blue (#F0F6FF approx.)
  - Offers a clean and calm space that is easy on the eyes.
- **Primary Text Color:** Deep Navy Blue (#2D2A72 approx.)
  - Used for titles and important text; it provides clear contrast and conveys stability.

- **Accent Color / Primary Button:** Gradient Yellow to Orange (#FEC901 to #FCA311)
  - Highlights the most important actions (e.g., Sign In) with high visibility.
- **Secondary Text & Links:** Muted Purple (#6C63FF to #4C4C6D)
  - Adds vibrancy without overwhelming the design; used for links like “Sign Up”.

### b. 🌙 Dark Mode

The dark mode employs:

- **Black or deep charcoal backgrounds** for minimal eye strain, especially in low-light conditions.
- **Gold (#FFD700)** for key elements such as:
  - ✓ App title
  - ✓ Back arrows
  - ✓ Buttons and highlights
- **White and grey text** for clear legibility across headings, body text, and labels.

This mode promotes visual comfort during nighttime use and provides a tech-forward, sleek aesthetic that aligns well with the app’s identity.

**Both modes** maintain visual consistency and use the same **font, iconography, and layout structure**, preserving familiarity regardless of theme.

### c. Typography:

- **Primary Font:** A geometric sans-serif like **Poppins** was used
  - ✓ Offers excellent readability and a friendly tone.
- **Font Style:** Bold for headings, medium for labels, and regular for body text
  - ✓ This helps create a visual hierarchy and guides the user’s eye.

## 2.2. Explaining the various pages

### Overview:

The pages here showcase essential user-facing elements such as the **login interface** and **sound diagnosis screen**, along with elements of the **bottom navigation system** and **notification handling**. The observations below explore these screens in detail, highlighting layout flow, visual consistency, color usage, and user engagement mechanisms.

### A. Login Screen – User Authentication Entry Point

The login screen serves as the primary **entry point** into the app for registered users. The design prioritizes clarity and ease of access, ensuring a low-friction login experience while reinforcing the application's identity.

#### User Flow

From this screen, the user has two clear paths:

- **Registered users** enter their credentials and tap “Sign In” to proceed to the app’s dashboard or home.
- **New users** tap “Sign Up,” leading to the registration screen for account creation.

### B. Sign Up → Select User Role (Mechanic or Car Owner)

When users tap “**Sign Up**”, the flow diverges depending on the type of user:

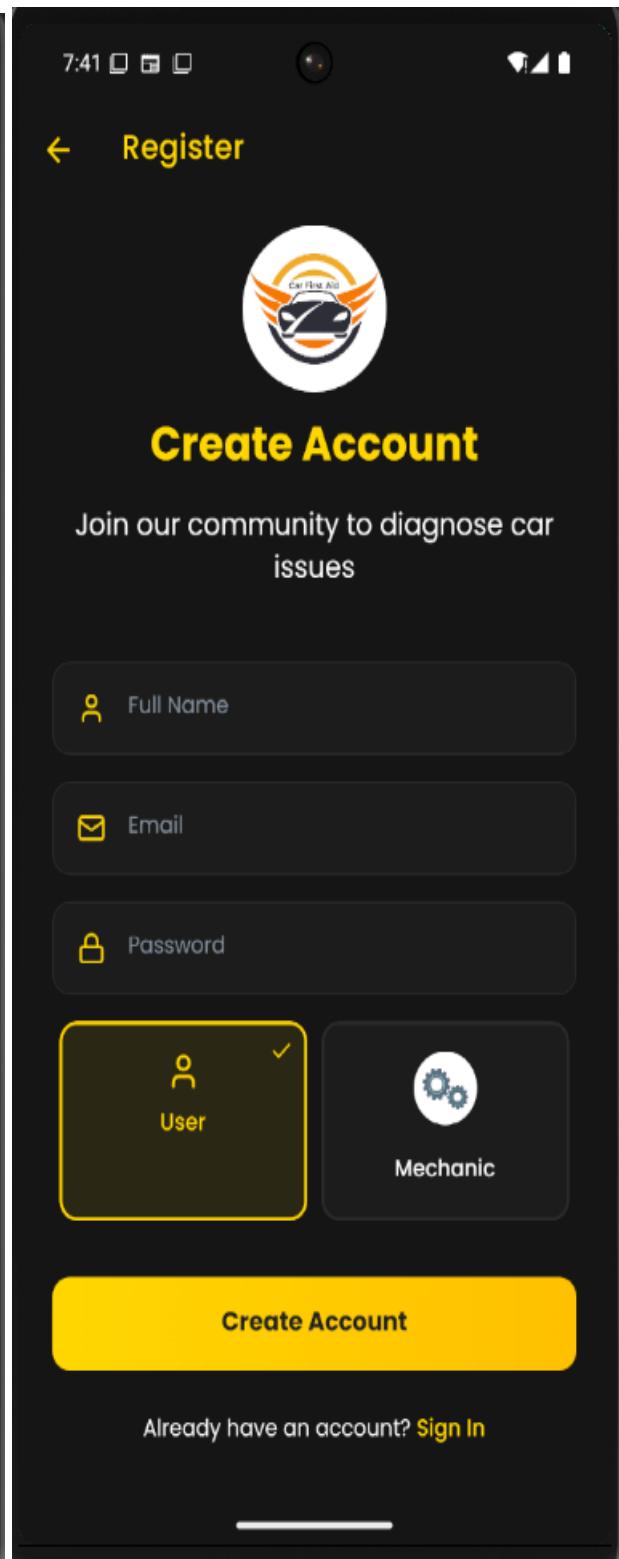
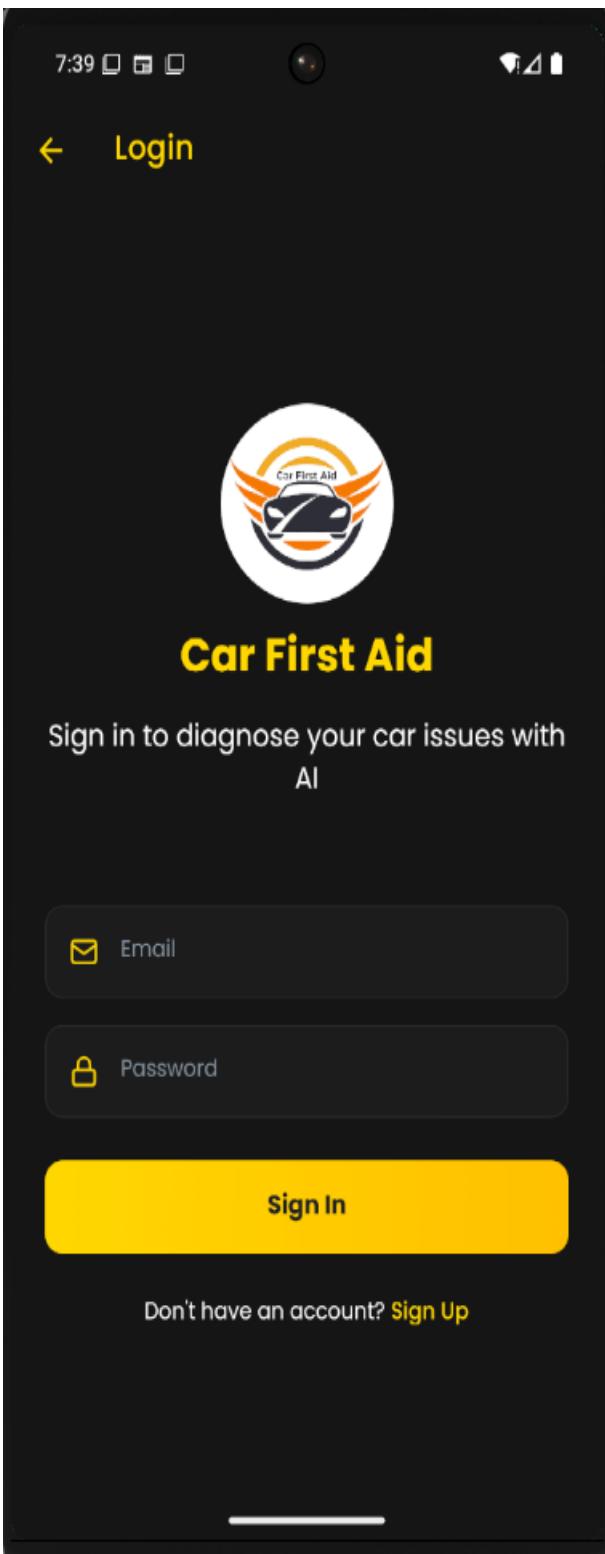
- The app prompts the user to **select a role**:
  - ✓ **Register as Car Owner**
  - ✓ **Register as Mechanic**

This role-selection step is essential because it determines the type of dashboard, functionality, and interface the user will access after registration.

This flow follows conventional UX patterns to reduce confusion and speed up onboarding.

#### Successful Login → Home / Diagnose Dashboard

After a successful login (whether as a car owner or mechanic), users land on their respective **dashboard or home screen**.



## C. Home Screen Analysis:

The user dashboard serves as the central hub for car owners to interact with the **Car Fault Diagnosis System**. Below is a detailed breakdown of the dashboard's design, functionality, and user experience.

### 1. Welcome & Personalization

### 2. Primary Call-to-Action (CTA): "Car Issues?"

- "Get instant AI-powered diagnosis for your vehicle problems"
  - ✓ Clearly communicates the system's value proposition: speed and AI-driven accuracy.
- "Diagnose Now"
  - ✓ A bold, action-oriented CTA that directs users to the core feature—fault diagnosis.

### 3. Diagnostic Methods

The dashboard provides **four intuitive pathways** for diagnosing car issues, catering to diverse user preferences and technical comfort levels:

#### i. Dashboard Light

"Snap or Upload a photo of your dashboard warning."

Leverages the smartphone camera for quick visual analysis of warning icons (e.g., check engine light). Ideal for users who may not know technical terms.

#### ii. Engine Sound

"Record unusual engine sounds."

Uses audio input to detect anomalies (e.g., knocking, squealing). Requires microphone access but simplifies diagnosis for auditory cues.

#### iii. Contact a Mechanic

"Connect with certified mechanics."

Redirects to a mechanic marketplace (shown in IMG\_9269), bridging DIY users with professionals. Builds trust through human expertise.

#### iv. Manual Input

"Describe your car problems."

A text-based fallback for users who prefer typing symptoms (e.g., "car vibrates at high speeds").

### 4. Top-Rated Mechanics Section

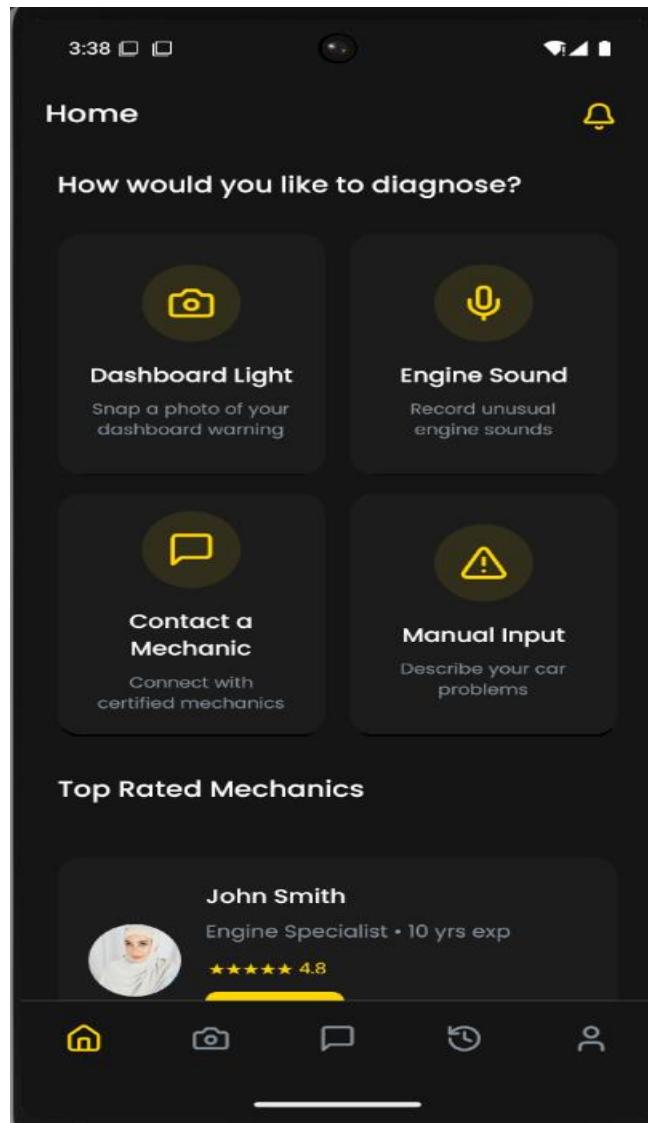
Preemptively showcases vetted mechanics to expedite professional help.

➤ **Design Elements:**

- ✓ **Profile Cards:** Example: "John Smith, Engine Specialist • 10 yrs exp • ★★★★★ 4.8"
  - Highlights credentials, experience, and ratings to aid decision-making.

## 5. Navigation Bar (Bottom)

- ✓ **Home:** Returns to this dashboard.
- ✓ **Diagnose:** To upload an image or record sound
- ✓ **History:** Past scans/reports for tracking recurring issues.
- ✓ **Profile:** To manage their account
- ✓ **Mechanic:** To browse certified mechanics



- ❖ When the user selects "Dashboard Light" from the home screen (as seen in int the image below), the system guides them through a structured diagnostic process, where the user can either take a photo using the phones camera or the user can upload a photo from the mobile device. Below is a representation.

### **Post-Upload Diagnosis:**

After the user uploads a photo, the app displays: **AI-Generated Results**

- **Immediate Fix Suggestions:**

- ✓ Helpful Videos like YouTube tutorial videos to help fix the fault.
- ✓ Then the user can start a new diagnosis.

- ❖ When the user selects "Engine Sound" from the home screen, the system transitions into an intuitive audio-capture interface designed to identify and analyze abnormal vehicular noises. Below is a comprehensive breakdown of this process

Upon clicking the "Engine Sound" option, the interface prompts the user with clear instructions: "Record unusual engine sounds for instant diagnosis."

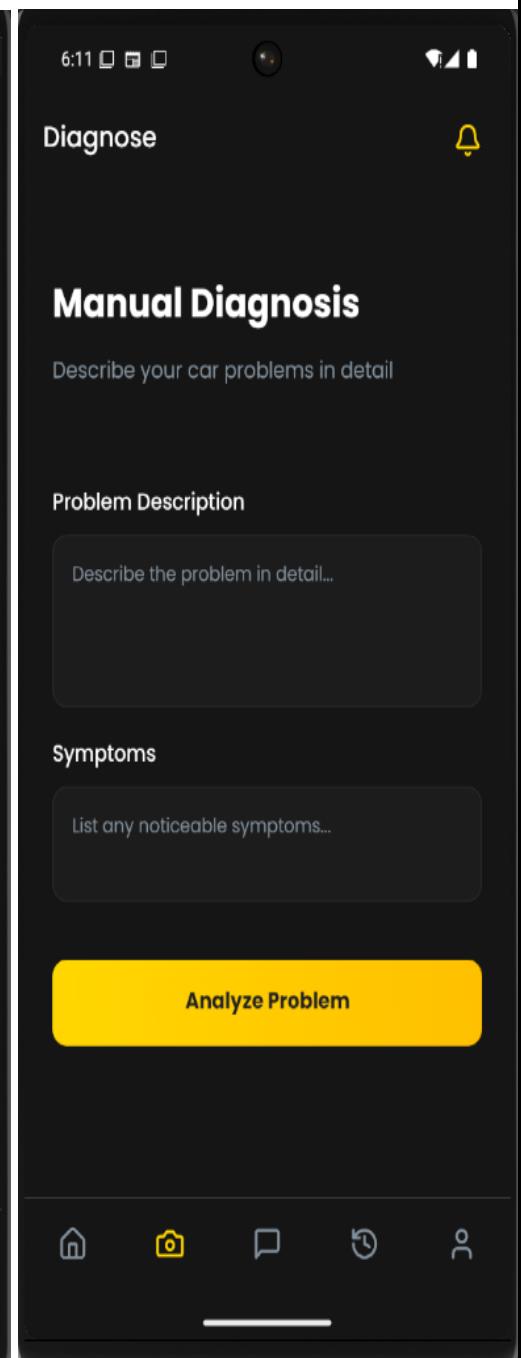
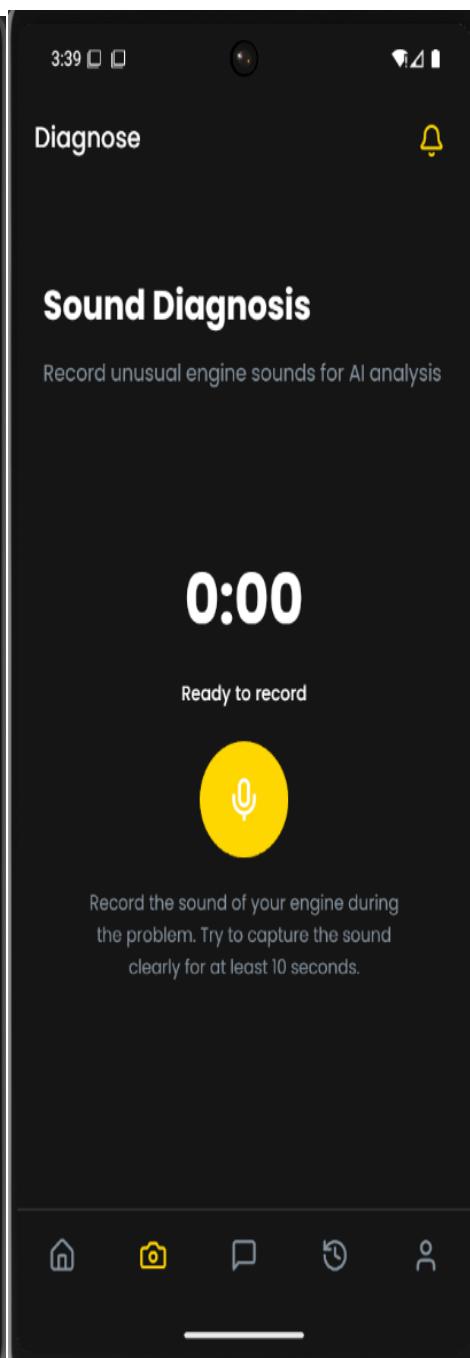
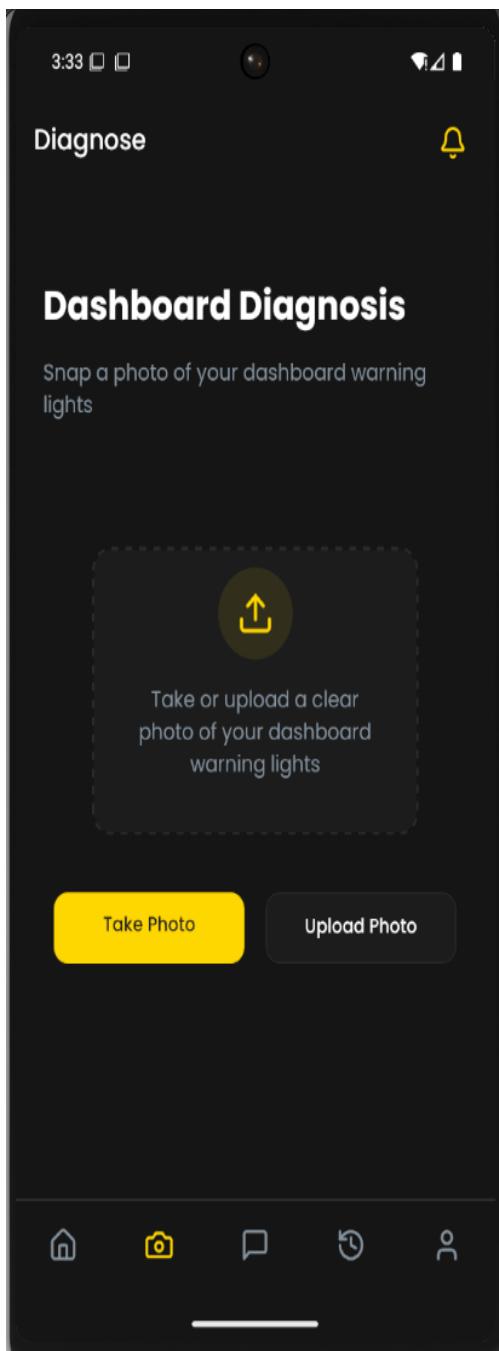
**Guidance:**

"Ensure the engine is running and background noise is minimal."

### **Audio Upload & Processing**

Once the recording is complete, the app processes the audio file through an AI-driven analysis engine:

- ❖ When the user selects “Contact a mechanic”, the mechanic pages opens up which show the different certified mechanics together with their contact information and locations and the user can just message each mechanic from there according to their choice.
- ❖ When the user selects "Manual Input" from the diagnostic menu, they are able to describe their problem in text format.
- All these pages are shown as below explaining all the different screens and navigations



6:30

Diagnose

**Dashboard Diagnosis**

Snap a photo of your dashboard warning lights

Severity: Medium

**Recommendations:**

- Check if the gas cap is loose or damaged
- Use an OBD-II scanner to read specific error codes
- Have a professional mechanic inspect the vehicle

**Helpful Videos:**

- How to Diagnose Check Engine Light
- Common Check Engine Light Causes

Start a New Diagnosis

Home Camera Chat Clock Profile

3:31

Mechanics

**Find a Mechanic**

Search by name, specialty or location

**John Smith** Verified  
Engine Specialist  
10 yrs exp  
★ 4.8

Downtown Auto Shop

Profile Message

**Sarah Johnson** Verified  
Electrical Systems  
8 yrs exp  
★ 4.6

Westside Garage

Profile Message

Home Camera Chat Clock Profile

## ❖ Notification Icon:

When you tap the **notification bell** at the top of your screen, you see a list of alerts as shown below

### What this does:

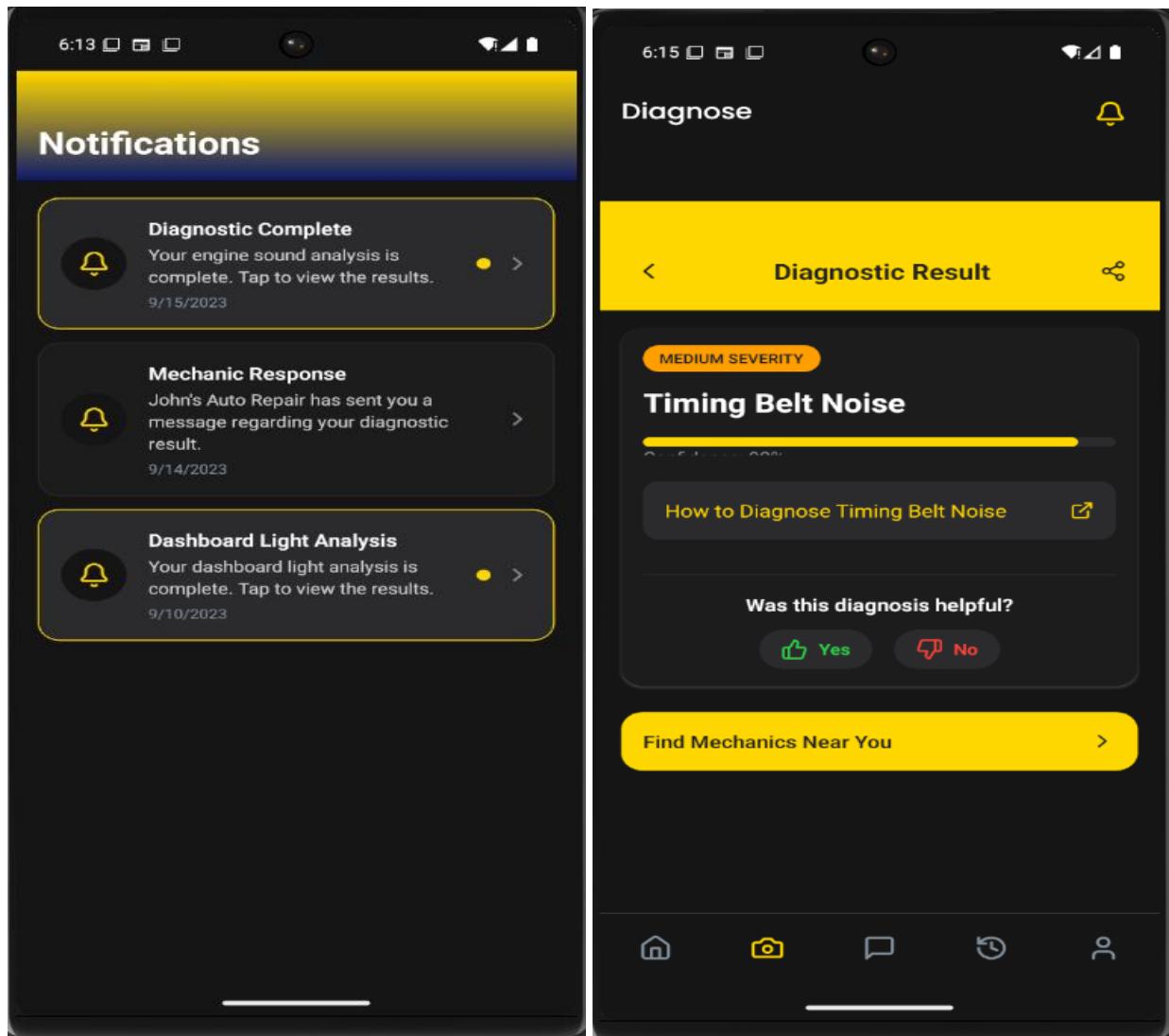
- ✓ Keeps you updated on past diagnostics and mechanic replies.
- ✓ Shows the most recent notifications first.

### Tapping a Notification Leads to the steps below

If you tap the "Diagnostic Complete" notification, it takes you to the **detailed results screen**:

Which shows **Issue Identified** and **Next Steps**:

The user also has the possibility to view Diagnosis History. All, Unresolved and Resolved past diagnosis as seen below:



## ❖ User Profile:

The **Profile** screen is where users manage their account settings, preferences, and app experience.

Under the "Account" section, users can:

- **Edit Profile:** Update name, email, profile picture, or vehicle details.
- **Notifications:** Customize alert preferences (e.g., disable maintenance reminders).
- **Privacy & Security:** Manage password, data sharing, or connected devices

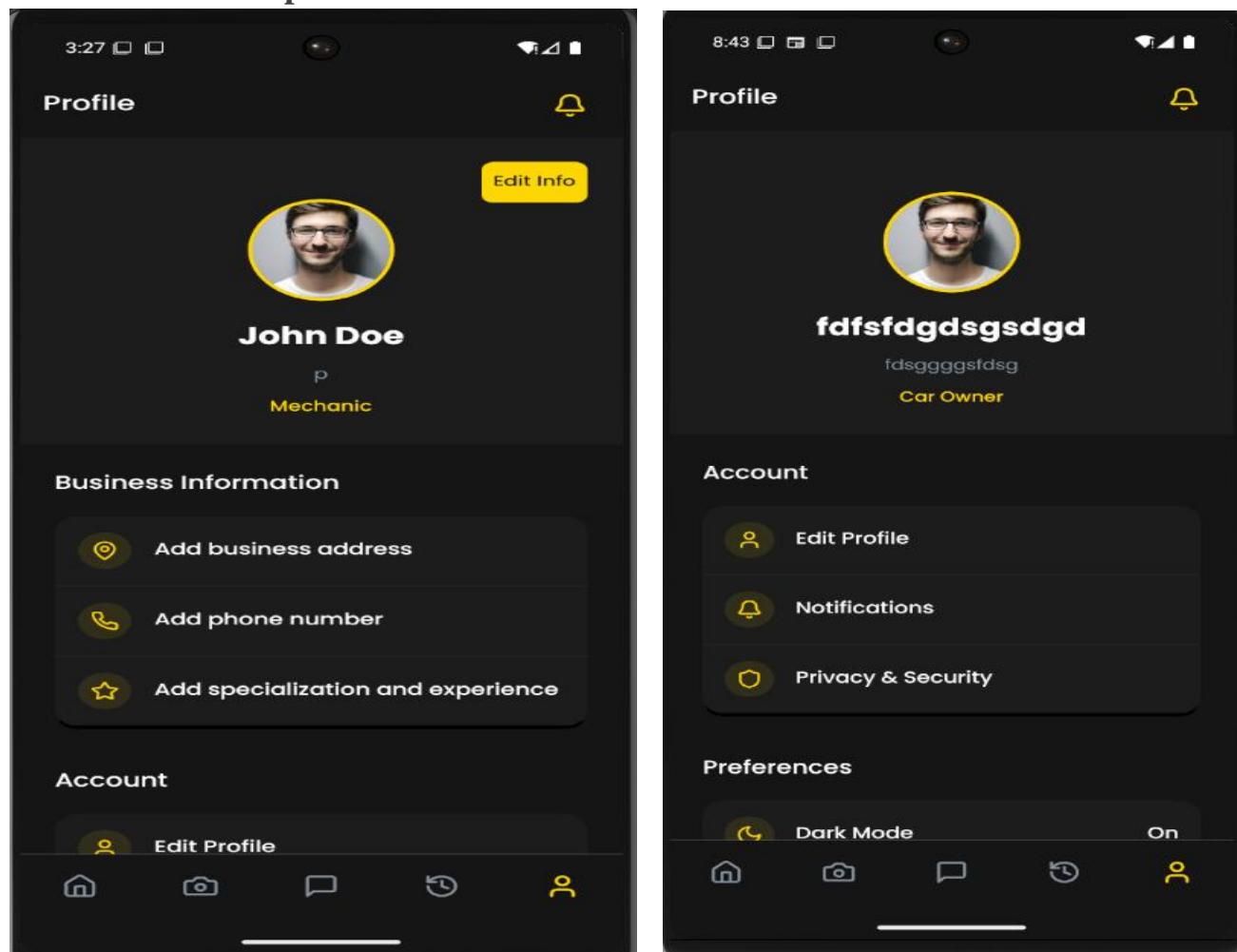
## Preferences

Here, users tailor the app to their liking:

- **Dark Mode Toggle:** Switch between **Light** and **Dark**

Here, the User Profile can be for a normal user(Car owner) or a mechanic who has extra functionalities like providing contact information and other personal information.

**Both of these profiles are shown below**



## ❖ Mechanic Verification & Dashboard

The mechanic's journey in the app begins with **verification**—a gatekeeping process that ensures only qualified professionals can offer services. Once verified, they gain access to a tailored dashboard where they can manage their profile and connect with car owners.

### Mechanic Verification

Before a mechanic can use the app, they must prove their credentials:

#### A. Upload Certification

- **Instruction:** "*Tap to upload a photo of your mechanic certification or license.*"
- **Purpose:** Validates the mechanic's expertise and protects users from unqualified help.

#### B. Verification Requirements

- **Mandatory:**
  - ✓ Valid professional license/certification
  - ✓ Clear image showing their name and credential details.

See image blow:

#

mechanics can also edit their professional details:

- **Business Address:** Lets customers locate their shop.
- **Phone Number:** For direct contact (optional, based on privacy preferences).
- **Specialization:** Tags like "*Engine Repair*" or "*Electrical Systems*" to attract relevant jobs.
- **Years of Experience:** A credibility booster

The mechanic dashboard, it is same as the car owner but the different is the fact that before the dashboard the system asks the mechanic to upload certificates and the mechanic can also edit their information.

8:57

## Mechanic Verification

Please upload your mechanic certification to get verified

Upload Certification

Tap to upload a photo of your mechanic certification or license

Requirements

- Valid professional mechanic certification or license
- Clear, readable image showing your name and certification details
- Proof of current employment at an auto repair shop (optional, but recommended)
- Any specialized training certificates (optional)

Your certification will be reviewed by our team. This process usually takes 1-2 business days.

8:58

## Mechanic Verification

Please upload your mechanic certification to get verified

Upload Certification



Submit for Verification

Requirements

- Valid professional mechanic certification or license
- Clear, readable image showing your name and certification details
- Proof of current employment at an auto repair shop (optional, but recommended)
- Any specialized training certificates (optional)

9:46

## Mechanic Verification

Uploading your certificate...



## **Conclusion and Design Assessment**

The initial batch of UI screens demonstrates a **strong, coherent, and user-centered design**. Navigation is intuitive, text is legible, and the color scheme is accessible and consistent across themes. The user flow—from login to sound diagnosis—is logical, efficient, and requires minimal cognitive effort, making it suitable for non-technical car owners. The use of dark/light mode support and a persistent navigation bar shows an understanding of mobile UX best practices.

As we continue reviewing more screens in the next batches, we will further evaluate how other features—such as image-based diagnosis, feedback submission, mechanic interactions, and user history—integrate into this well-designed flow.

## **Summary of First Batch Flows**

- 1. Launch App → Login / Sign Up**
- 2. Sign Up → Choose Role (Mechanic / Car Owner)**
- 3. Login → Role-Based Dashboard**
- 4. Diagnose → Record Sound → Submit for AI Processing**
- 5. Access notifications and other tabs via bottom nav**

### 3. Frontend Implementation

The frontend implementation phase is a crucial component of the user interface (UI) design and development process. It involves transforming static mockups, wireframes, and design concepts into an interactive, responsive, and functional mobile application that users can engage with in real time. This phase serves as the bridge between visual design and backend functionality, ensuring that the application not only looks appealing but also delivers a smooth and intuitive user experience (UX).

In the development of the *CarAI Diagnosis* mobile app, frontend implementation was carried out with a strong emphasis on **usability, performance, responsiveness, and consistency** across Android and iOS devices. The frontend was developed using **React Native**, a widely adopted cross-platform mobile framework based on JavaScript and supported by the React ecosystem. React Native was selected due to its efficiency in building native-quality mobile apps from a shared codebase, enabling the team to focus on both design and functionality without platform-specific redundancy.

#### 3.1 Structure and Componentization

The application follows a modular, component-based architecture that promotes scalability, maintainability, and the reusability of UI elements. The project is organized into folders based on features and responsibilities, ensuring a clean separation of concerns.

Key directories and components include:

- **app/**: Contains screen-level routes and logic.
  - ✓ **(auth)/**: Manages authentication-related screens such as login and registration.
  - ✓ **(mechanic)/**: Displays the list of verified mechanics along with profile information and ratings.

Also deals with the mechanic authentication and verification

- ✓ **(tabs)/**: Contains navigation tabs such as Home, Profile, and Diagnoses and History.
- ✓ **notifications.tsx**: Dedicated screen for listing real-time notifications for the user (alerts, status updates, etc.).
- ✓ **+not-found.tsx**: Custom fallback screen for undefined routes.
- ✓ **\_layout.tsx**: Provides the layout structure and navigation stack.
- ✓ **index.tsx**: Acts as the default entry point for routing.

- **components/**: Reusable UI elements like buttons, input fields, cards, loaders, and modals — each styled consistently for design coherence.
- **services/**: Handles external interactions such as API calls and server communication.
- **context/**: Provides global state management using context providers (e.g., user authentication state).
- **constants/**: Houses static values like color schemes, font styles, and configuration constants.
- **types/**: Includes TypeScript type definitions for better maintainability and error prevention.
- **hooks/**: Contains custom React hooks for shared business logic (e.g., form handling, notification polling).
- **assets/**: Stores images, icons, and other static media.

This structure ensures that the application remains extensible, readable, and maintainable as the project scales.

### **3.2 Tools and Technologies Used**

The development of the *Car First Aid* mobile application leverages a modern and efficient set of tools and libraries tailored to enhance performance, usability, and cross-platform compatibility. The primary focus of the frontend is to deliver an intuitive user experience while ensuring integration with device hardware and backend services. Below is a breakdown of the technologies utilized:

#### **1. React Native**

React Native serves as the core framework for building the mobile application. It allowed us to write a single codebase in JavaScript/TypeScript that compiles to native code for both Android and iOS platforms. Its component-based architecture aligns well with the modular structure of the app, enabling rapid development and reusability of UI elements like buttons, cards, and screens.

#### **2. Expo Framework**

Expo was used to streamline the development workflow by providing tools, libraries, and services for building, deploying, and testing the app. Expo simplifies access to native APIs (like camera, audio, and notifications) without needing to eject from the managed workflow, significantly speeding up the development and debugging process.

### **3. React Navigation**

The application utilizes **React Navigation** to handle in-app navigation with a combination of **stack**, **tab**, and **nested navigators**. It provides a fluid navigation experience across different screens, such as authentication flows, the dashboard, diagnostics results, mechanic listings, and settings. Navigation is intuitive and conforms to mobile UI standards, improving user engagement and usability.

### **4. Expo Camera & Audio**

To enable real-time data acquisition from the user, such as capturing **dashboard images** and recording **engine sounds**, the app integrates:

- ✓ **expo-camera:** Used to access the device's camera for scanning visual car dashboard faults (e.g., engine lights, indicators).
  - ✓ **expo-av (Audio/Video):** Used for capturing audio input from the engine sounds to analyze and identify potential mechanical issues using AI on the backend.
- These features form the core input methods for the diagnosis process and are crucial for achieving the app's primary goal.

### **5. TypeScript**

TypeScript is used across the codebase to enforce type safety, reduce runtime errors, and improve developer productivity. With static typing and interface definitions (found in the types/ directory), the application is easier to maintain and scale.

### **6. Context API and React Hooks**

React's Context API and custom hooks are utilized for global state management and reusability of logic. For example, user session data, theme preferences, and diagnostic results are managed centrally using contexts. Custom hooks simplify complex logic such as form validation and notification fetching.

### **7. Local Asset and Component Management**

The assets/ folder holds the visual elements like logos, icons, and sample data, while components/ encapsulates reusable UI pieces. These structures ensure consistency and make the UI scalable as new features are added.

## **8. Notification Handling (expo-notifications)**

The app includes real-time and scheduled notifications, implemented using expo-notifications, to inform users of:

- Diagnostic results availability
- Mechanic appointment confirmations
- Maintenance reminders

### **Summary**

This technology stack was carefully chosen for its:

- **Cross-platform support**
- **Ease of integration with mobile hardware**
- **Active community and library ecosystem**
- **Support for rapid prototyping and deployment**

Together, these tools enable the Car First Aid app to provide a seamless, interactive, and intelligent car fault diagnostic experience to users directly from their smartphones.

### **3.3 Styling and Theming**

Styling in the application is handled using **styled-components**.

The UI follows a minimalistic and user-friendly design with consistent spacing, rounded edges, iconography, and clear color hierarchies.

Interactive elements such as buttons, input fields, and alerts are styled to be responsive, accessible, and aligned with mobile usability standards. Font sizing, padding, and contrast ratios were carefully chosen to ensure optimal performance on both small and large screens.

### **3.4 Performance and Optimization**

To ensure that the app performs well under varying network conditions and hardware capabilities, performance optimizations were considered during implementation:++

- **Code-splitting and lazy loading** were employed to load screens and media content on demand.

### **3.5 Development Workflow**

The frontend development followed an agile and iterative workflow. Screens were built and tested individually using emulators and physical devices, with continuous integration of feedback from design reviews and peer testing. Tools like **Expo Go**, **Android Studio**, and **VS Code** facilitated rapid prototyping and deployment.

## **Conclusion**

The UI Design and Implementation phase marks a vital step in transforming our system from concept to reality. By creating a strong identity through the name ***Car First Aid***, designing a user-friendly interface, and implementing a robust, modern frontend, we aim to deliver an app that feels intuitive, helpful, and reliable.

With the foundation laid, the next steps will include iterative testing, UI refinements, and integration with AI microservices to complete a smooth and intelligent experience for every user.