

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

**MINISTRY OF HIGHER
EDUCATION**

UNIVERSITY OF BUEA



REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

**MINISTRE DE
L'ENSEGNEMENT SUPERIEUR**

UNIVERSITE DE BUEA

UNIVERSITY OF BUEA

FACULTY OF ENGINEERING AND TECHNOLOGY

UI DESIGN AND IMPLEMENTATION

TASK 5 BY GROUP 25

COURSE INSTRUCTOR: Dr NKEMENI VALERY

**COURSE CODE/TITLE: CEF 440/ INTERNET PROGRAMMING(J2EE)
AND MOBILE PROGRAMMING.**

ACADEMIC YEAR 2024/2025

Table of Contents

.....	1
Introduction:	3
App Identity:	3
2.1 Defining the App Identity	3
2.2 App Name	3
2.3. Logo & Iconography.....	4
2.4. Brand Voice & Tone.....	5
2.5. Target Audience.....	6
2.6. Unique Selling Proposition (USP).....	7
2.7. User Interface (UI) & User Experience (UX) Principles	7
Visual Design	8
3.1. Login Page	8
3.2. Home Page	10
3.3. History Page.....	12
3.4. Mechanics Page	14
3.5. Help and Support Page	16
3.6. Dashboard Light Scanning Page.....	17
3.7. Engine Sound Diagnosis Page	19
Front End Implementation	21
4.1. Architecture Overview	21
4.2. Key Features Implemented	22
4.3 UI/UX Design Principles	23
4.4. Technical Implementation Details.....	24
4.5. Future Enhancements	25
Conclusion	25

Introduction:

This report presents a comprehensive analysis of the UI design and implementation process for the Car Fault Diagnosis App, a mobile application developed to empower car owners with tools to diagnose vehicle faults, including dashboard warning lights and engine sound irregularities. The design and implementation focus on delivering an intuitive, user-friendly interface that integrates advanced diagnostic features with offline functionality. This introduction sets the foundation for exploring the app's identity, visual aesthetics, and technical frontend development, ensuring a seamless and reliable user experience tailored to the demands of modern vehicle maintenance.

App Identity:

This section delineates the app identity for the Car Fault Diagnosis App (AutoFix Car), a mobile solution engineered to assist car owners in diagnosing vehicle issues such as dashboard warning lights and engine sound anomalies. A robust app identity is pivotal in establishing a trustworthy and recognizable brand, fostering user confidence, and differentiating the application within the competitive automotive diagnostics market. By defining a cohesive identity, the app aligns with the needs of non-technical users seeking reliable, accessible tools to maintain their vehicles effectively.

2.1 Defining the App Identity

The AutoFix Car App is designed to simplify the process of identifying and addressing vehicle faults, targeting car owners who lack extensive mechanical expertise. Its primary purpose is to provide an intuitive interface for diagnosing issues through features like dashboard light recognition and engine sound analysis, enhanced by offline functionality for convenience. The app's unique selling proposition lies in its integration of artificial intelligence-driven diagnostics with a user-friendly experience, setting it apart from traditional diagnostic methods and empowering users with actionable insights to ensure vehicle health.

2.2 App Name

The app name is the first point of contact for users and should be memorable, relevant, and easy to pronounce and spell.

- **Considerations:**

- ❖ **Clarity:** Should immediately convey the app's function.
- ❖ **Memorability:** Easy to recall.
- ❖ **Availability:** Domain, social media handles, and app store listings.
- ❖ **Conciseness:** Short and impactful.
- ❖ **Scalability:** Should allow for future expansion beyond just fault diagnosis (e.g., maintenance tracking, repair guides).

2.3. Logo & Iconography

The logo and app icon are the visual cornerstone of the brand. They are distinctive, scalable, and instantly recognizable.

- **Design Principles:**

- ❖ **Simplicity:** Easy to understand at a glance, even at small sizes
- ❖ **Relevance:** Reflects the automotive or diagnostic theme.
- ❖ **Memorability:** Unique enough to stand out.
- ❖ **Versatility:** Works well across various platforms (app stores, website).

- **Key Visual Elements:**

- **Iconography:**

- ❖ Stylized car outline.
- ❖ Abstract representation of a diagnostic tool (e.g., a wrench, a gear, a circuit board element, a magnifying glass).
- ❖ A combination of both, perhaps a car with a diagnostic symbol overlaid.
- ❖ A "check" mark or a "green light" symbol to convey problem resolution.

- **Typography:** Clean, modern, legible sans-serif fonts for the app name.
- **Color Palette:**
 - **Primary Colors:**
 - ❖ **Blue:** Conveys trust, reliability, technology, and professionalism.
 - ❖ **Vibrant Green:** Suggests "go," "fix," "solution," and safety.
 - **Accent Colors:**
 - ❖ **Silver/Grey:** For metallic or technical elements, adding a sophisticated touch.
- **Rationale:** The combination of blues/greens instills confidence and efficiency, while bright accents draw attention to critical information or actions.

2.4. Brand Voice & Tone

The brand voice defines how the app communicates with its users, both within the app and in external communications.

- **Core Characteristics:**
 - ❖ **Authoritative & Knowledgeable:** Users trust the app for accurate diagnoses.
 - ❖ **Helpful & Supportive:** Guides users through complex information.
 - ❖ **Clear & Concise:** Avoids jargon where possible, explains technical terms simply.
 - ❖ **Empowering:** Helps users understand and take control of their vehicle's health.
 - ❖ **Professional:** Maintains a high standard of communication.

- **Tone (Context-Dependent):**

- ❖ **Informative:** When presenting diagnostic codes or explanations.
- ❖ **Reassuring:** When a problem is identified, guiding the user to the next steps.
- ❖ **Urgent (when necessary):** For critical faults that require immediate attention.
- ❖ **Friendly (but not overly casual):** In onboarding or general tips.

2.5. Target Audience

Understanding the target audience is fundamental to shaping the app's identity.

- **Primary Users:**

- ❖ **Car Enthusiasts:** Individuals who prefer to diagnose and fix minor issues themselves.
- ❖ **Cost-Conscious Car Owners:** Those looking to understand problems before visiting a mechanic to avoid unnecessary repairs or costs.
- ❖ **New Car Owners:** Individuals who may be less familiar with car maintenance and want to be proactive.
- ❖ **Small Workshop Mechanics/Technicians:** As a supplementary diagnostic tool.

- **Needs & Expectations:**

- ❖ **Accuracy:** Reliable diagnostic information.
- ❖ **Simplicity:** Easy-to-understand explanations of complex issues.
- ❖ **Actionable Insights:** Clear next steps for repair or professional consultation.

- ❖ **Trustworthiness:** A credible source of information.

- ❖ **Speed & Efficiency:** Quick diagnosis and access to information.

2.6. Unique Selling Proposition (USP)

What makes this car fault diagnosis app stand out from competitors?

- **Potential USPs:**

- ❖ **AI-Powered Predictive Diagnostics:** Goes beyond just reading codes to predict potential future issues.

- ❖ **Integrated Repair Guides & Parts Sourcing:** Not just diagnosis, but also solutions.

- ❖ **Community-Driven Solutions:** Users can share experiences and solutions.

- ❖ **Voice-Activated Diagnostics:** Hands-free operation for convenience.

- ❖ **Comprehensive Vehicle Health Reports:** Detailed, easy-to-understand reports that can be shared with mechanics.

- ❖ **Offline Functionality:** Core features available without an internet connection.

- **Identity Impact:** The chosen USP clearly communicated through the app's messaging, features, and overall design., the visual identity might incorporate elements of data or intelligence since it's AI-powered and it's UI/UX reflect its simplicity.

2.7. User Interface (UI) & User Experience (UX) Principles

While not strictly "identity," the UI/UX is how the identity is experienced.

- ❖ **Clarity & Intuition:** Easy navigation, clear presentation of information.

- ❖ **Minimalism:** Clean design, avoiding clutter, focusing on essential data.

- ❖ **Feedback & Guidance:** Clear indications of progress, success, or errors.
- ❖ **Accessibility:** Legible fonts, sufficient color contrast, logical flow.
- ❖ **Consistency:** Uniform design elements, interactions, and terminology throughout the app.
- ❖ **Visual Hierarchy:** Using size, color, and placement to guide the user's eye to important information.

2.8. Marketing & Messaging

How the app identity translates into external communication.

- ❖ **Key Message:** Empowering car owners with intelligent diagnostics. or Your car's health, simplified.
- ❖ **Marketing Channels:** App store descriptions, social media, automotive forums, online ads.
- ❖ **Visual Consistency:** All marketing materials would consistently use the defined logo, colors, and typography.

Visual Design

Visual design refers to the aesthetic and stylistic aspects of a user interface. It focuses on how an app or website looks, encompassing elements like color, typography, iconography, imagery, layout, and overall visual hierarchy. The goal of visual design is to create an appealing and engaging experience that also guides the user's eye and communicates information effectively.

3.1. Login Page

Purpose: To authenticate existing users and allow new users to register, securing their data and personalizing their experience.

Key Features/Elements:

- **Email/Username Input Field:** For existing users to enter their credentials.
- **Password Input Field:** With an option to show/hide password.
- **Login Button:** To submit credentials.
- **Forgot Password Link:** To initiate a password reset process (typically via email).
- **Sign Up / Register Link:** To navigate to a new user registration form.
- **Social Login Options (Optional):** Buttons for Google, Apple, Facebook sign-in for convenience.
- **Remember Me Checkbox (Optional):** To keep the user logged in on the device.
- **Loading Indicator:** To show progress during authentication.
- **Error Messages:** Clear, concise messages for incorrect credentials, network issues, etc.

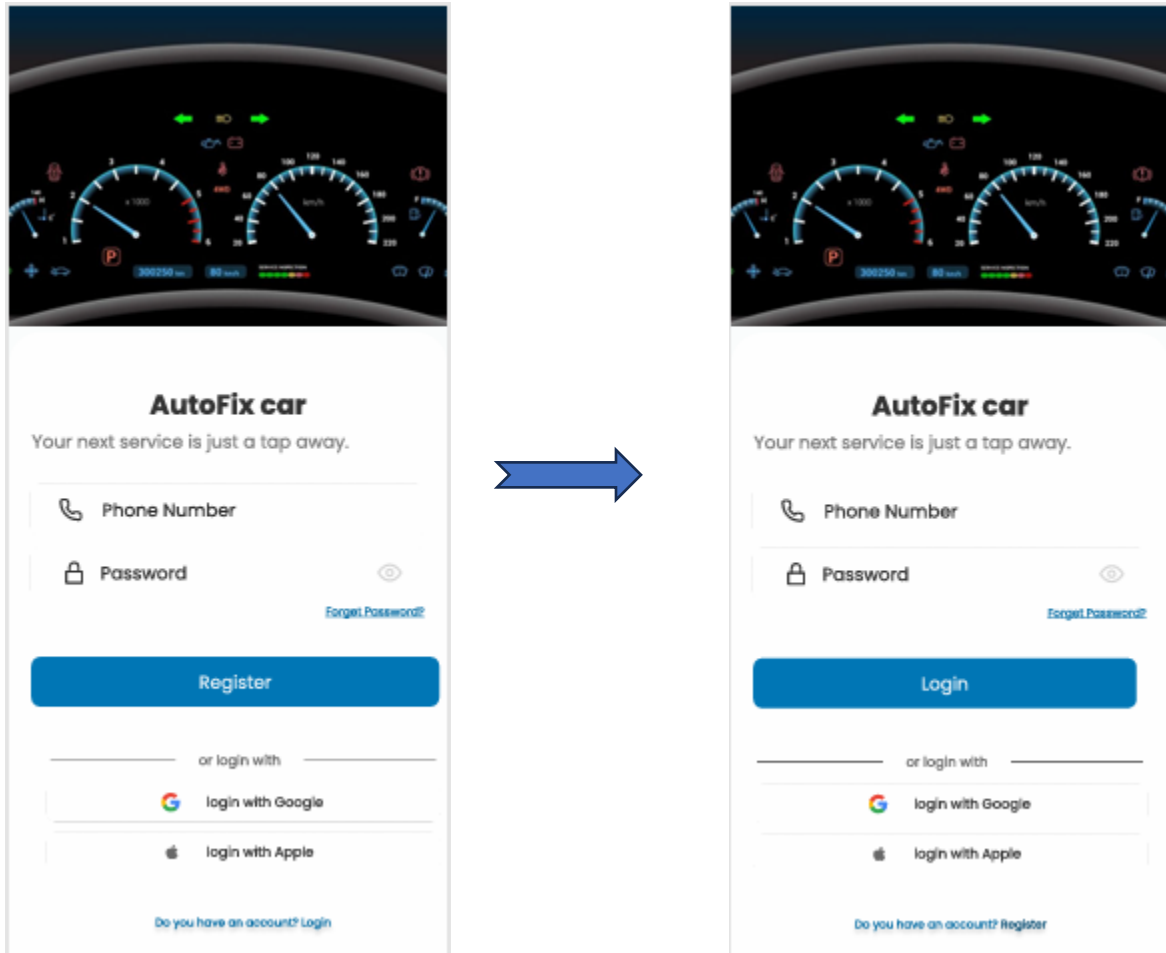
User Flow/Interaction:

1. Existing User:

- User opens app -> presented with Login Page.
- Enters registered email/username and password.
- Taps "Login".
- If successful, redirected to Home Page. If unsuccessful, receives an error message (e.g., "Invalid credentials").
- If "Forgot Password" is tapped, taken to a separate screen/dialog to enter email for reset link.

2. New User:

- User opens app -> presented with Login Page.
- Taps "Sign Up".
- Navigated to a registration form (name, email, password, confirm password).
- Submits details -> account created and logged in, redirected to Home Page (perhaps with a brief onboarding tour).



3.2. Home Page

Purpose: To serve as the central hub, providing a quick overview of the vehicle's health, quick access to primary functions, and recent activity.

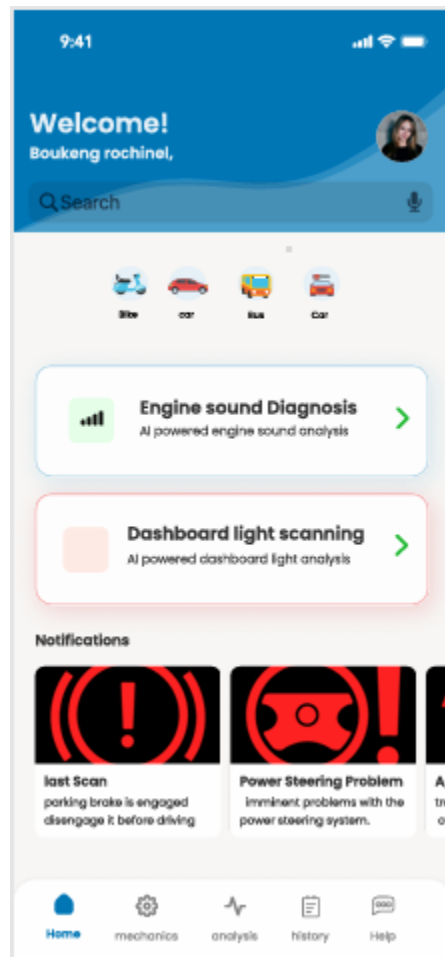
Key Features/Elements:

- **Vehicle Health Summary:** A prominent visual indicator (e.g., a large card with a car icon, green/yellow/red status, and a summary like "No Faults Detected" or "2 Active Faults").
- **"Run Diagnostic Scan" / "Connect OBD" Button:** The most prominent call to action, initiating the primary diagnostic process.
- **Recent Activity/Scan History Preview:** A small section displaying the last few scan results or important notifications.
- **Quick Access Buttons/Cards:** For frequently used features like:

- View Fault Codes
 - Live Data
 - Clear Codes
 - My Vehicle (details)
- **Connected Device Status:** Indicator for the OBD-II adapter connection (e.g., "OBD Connected," "Not Connected").
 - **Bottom Navigation Bar (Common):** Icons for Home, History, Mechanics, and Settings/Profile for easy navigation between core sections.

User Flow/Interaction:

1. **Upon Login:** User lands directly on the Home Page.
2. **Initiate Scan:** User taps Run Diagnostic Scan.
 - If OBD-II device not connected, prompts user to connect.
 - If connected, initiates the scanning process (shows progress bar, then redirects to Fault Codes page or a summary).
3. **Quick Info Access:** User taps a summary (e.g., "2 Active Faults") to jump directly to the detailed Fault Codes page.
4. **Navigation:** User taps icons on the bottom navigation bar to move to other main sections of the app.



3.3. History Page

Purpose: To store and display a comprehensive record of all past diagnostic scans, allowing users to track their vehicle's health over time.

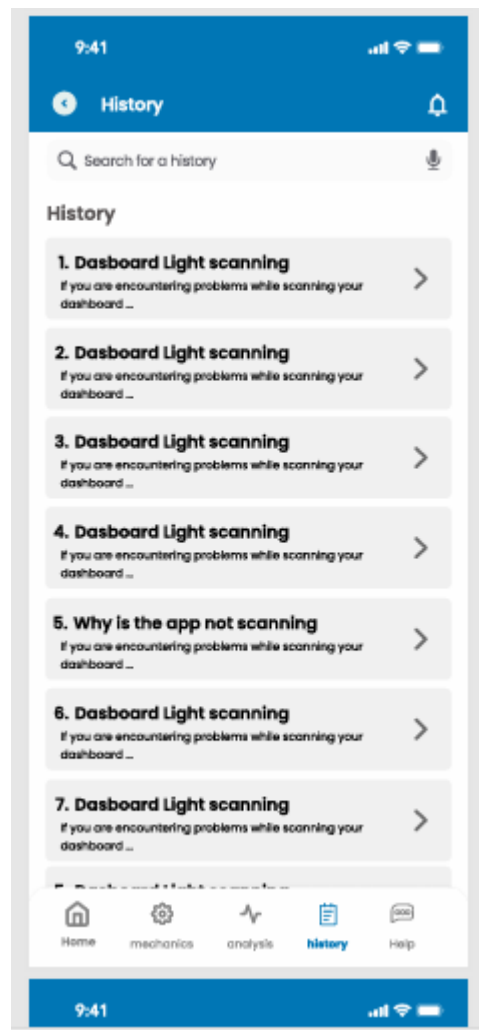
Key Features/Elements:

- **Chronological List of Scans:** Each entry displaying:
 - Date and Time of Scan.
 - Vehicle Used (if multiple vehicles are managed).
 - Summary of Results (e.g., "No Faults," "P0420 Detected," "Cleared 1 Fault").
 - Duration of scan (optional).
- **Detailed Report View:** Tapping an entry expands or navigates to a full report of that specific scan.

- Lists all DTCs (active, pending, historical) at the time of that scan.
- Associated freeze frame data (if available).
- Actions taken (e.g., codes cleared).
- **Filtering/Sorting Options:** By date, vehicle, fault status.
- **Search Functionality:** To find specific historical scans or fault codes.
- **Export/Share Report Button:** To generate a PDF or share the detailed report via email/messaging apps.

User Flow/Interaction:

1. **Access History:** User navigates to the History Page (e.g., via bottom navigation).
2. **Review Scans:** User scrolls through the list of past scans.
3. **View Details:** User taps a specific scan entry to view its full diagnostic report.
4. **Manage Reports:** User can filter the list or use the export/share option for a selected report.



3.4. Mechanics Page

Purpose: To connect users with qualified mechanics or repair services, offering features to find, contact, and share diagnostic information with professionals.

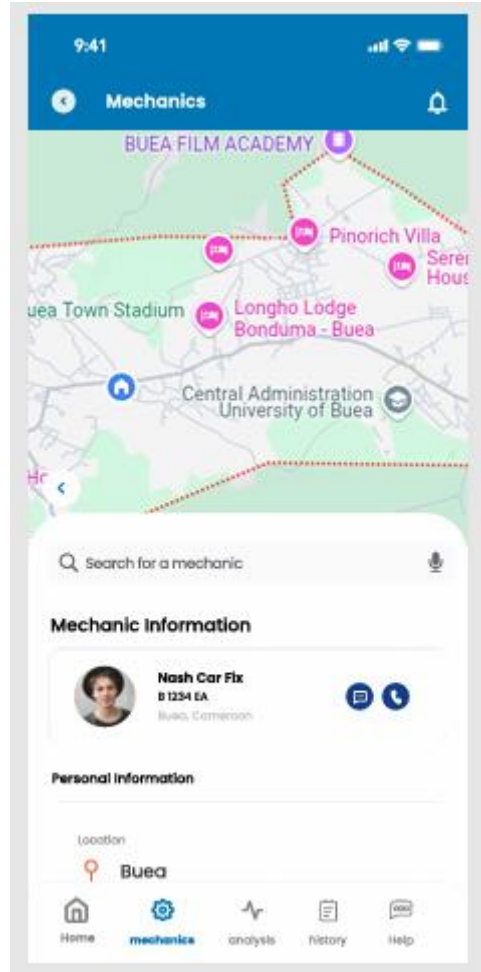
Key Features/Elements:

- **Location-Based Search:**
 - Find Mechanics Near Me button.
 - Map view showing nearby workshops.
 - List view of mechanics with distance.

- **Mechanic Profiles:** Each listing showing:
 - Name and Address.
 - Contact Information (phone, email).
- **Share Report" Functionality:** A prominent button within a mechanic's profile or during report viewing to directly send a selected diagnostic report to the mechanic.

User Flow/Interaction:

1. **Find Mechanic:** User navigates to the Mechanics Page.
2. **Location Access:** App requests location permission if not already granted.
3. **Browse/Search:** User views map or list, filters by specialty, or searches for specific workshops.
4. **View Profile:** User taps a mechanic's listing to see detailed information.
5. **Contact/Share:** User can call the mechanic, visit their website, or use the "Share Report" button to send their vehicle's diagnostic data.



3.5. Help and Support Page

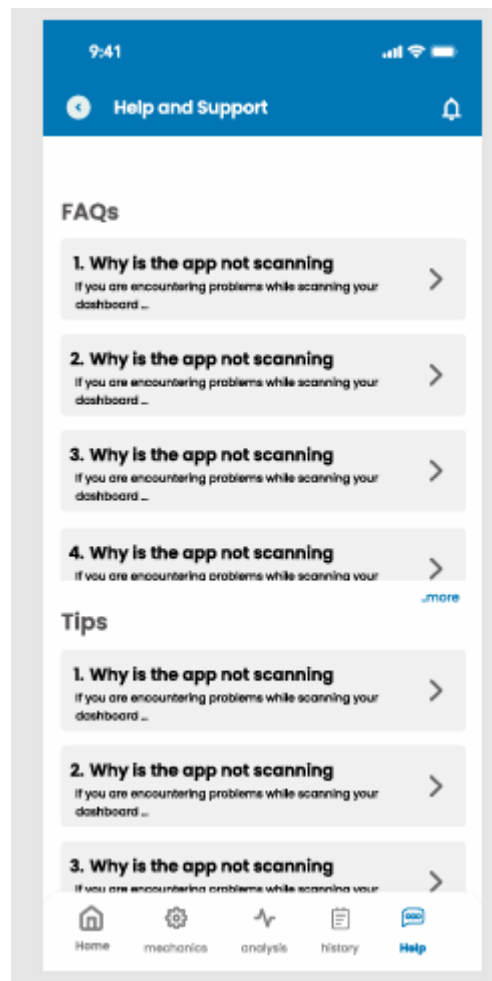
Purpose: To provide users with resources for understanding the app, troubleshooting common issues, and seeking direct assistance.

Key Features/Elements:

- **FAQ Section:** Categorized list of frequently asked questions and their answers (e.g., "How to connect OBD?", "What does P0420 mean?").
- **Troubleshooting Guides:** Step-by-step instructions for common issues (e.g., "My OBD won't connect," "App is crashing").
- **User Manual/Getting Started Guide:** A digital version of comprehensive instructions.
- **Submit Bug Report / Feature Request" Form:** For users to provide specific feedback.

User Flow/Interaction:

1. **Access Help:** User navigates to the Help and Support Page (e.g., via settings or a dedicated icon).
2. **Self-Service:** User browses FAQs, troubleshooting guides, or the glossary to find answers.
3. **Contact Support:** If self-service isn't sufficient, the user selects a contact method to get direct assistance.
4. **Provide Feedback:** User can utilize the bug report/feature request forms.



3.6. Dashboard Light Scanning Page

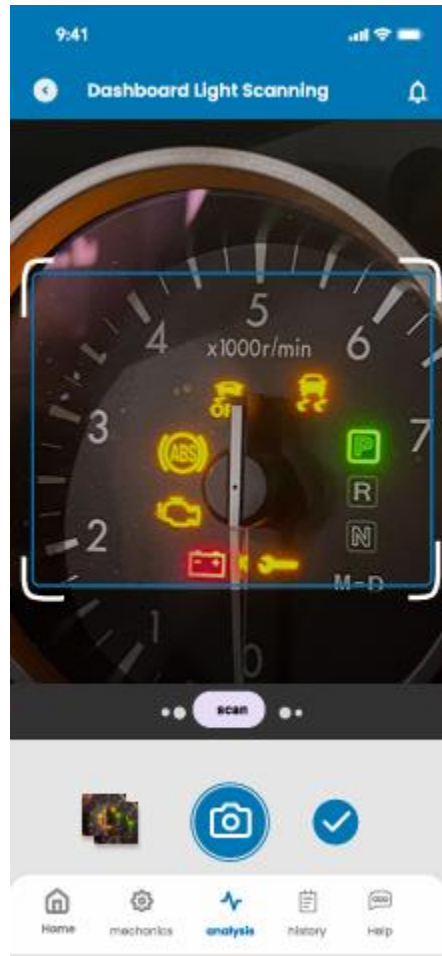
Purpose: To help users identify and understand the meaning of various warning lights displayed on their car's dashboard without needing an OBD-II connection.

Key Features/Elements:

- **Camera Interface:** Activates the device's camera.
- **Image Capture Button:** To take a photo of the dashboard warning light.
- **Image Processing/AI Recognition:** Backend logic to identify the light.
- **Light Gallery/Reference:** A visual catalog of common dashboard warning lights for manual Browse, categorized (e.g., Engine, Safety, Fluid).
- **Explanation of Light:** Once identified (either via scan or manual selection):
- **Learn More / Connect OBD Button:** To get more detailed diagnostic info if the light indicates an OBD-related fault.

User Flow/Interaction:

1. **Access Function:** User navigates to "Dashboard Light Scan".
2. **Capture Image:**
 - User points camera at the dashboard light.
 - Taps "Capture" button.
 - App processes image.
 - Displays identified light and its explanation.
3. **Manual Lookup:**
 - User taps "Browse Dashboard Lights".
 - Scrolls through or searches a gallery of lights.
 - Taps a light to see its detailed explanation
4. **Next Steps:** User follows recommended actions or chooses to connect OBD for deeper diagnosis.



3.7. Engine Sound Diagnosis Page

Purpose: To assist users in identifying potential car problems by analyzing unique engine sounds using the device's microphone.

Key Features/Elements:

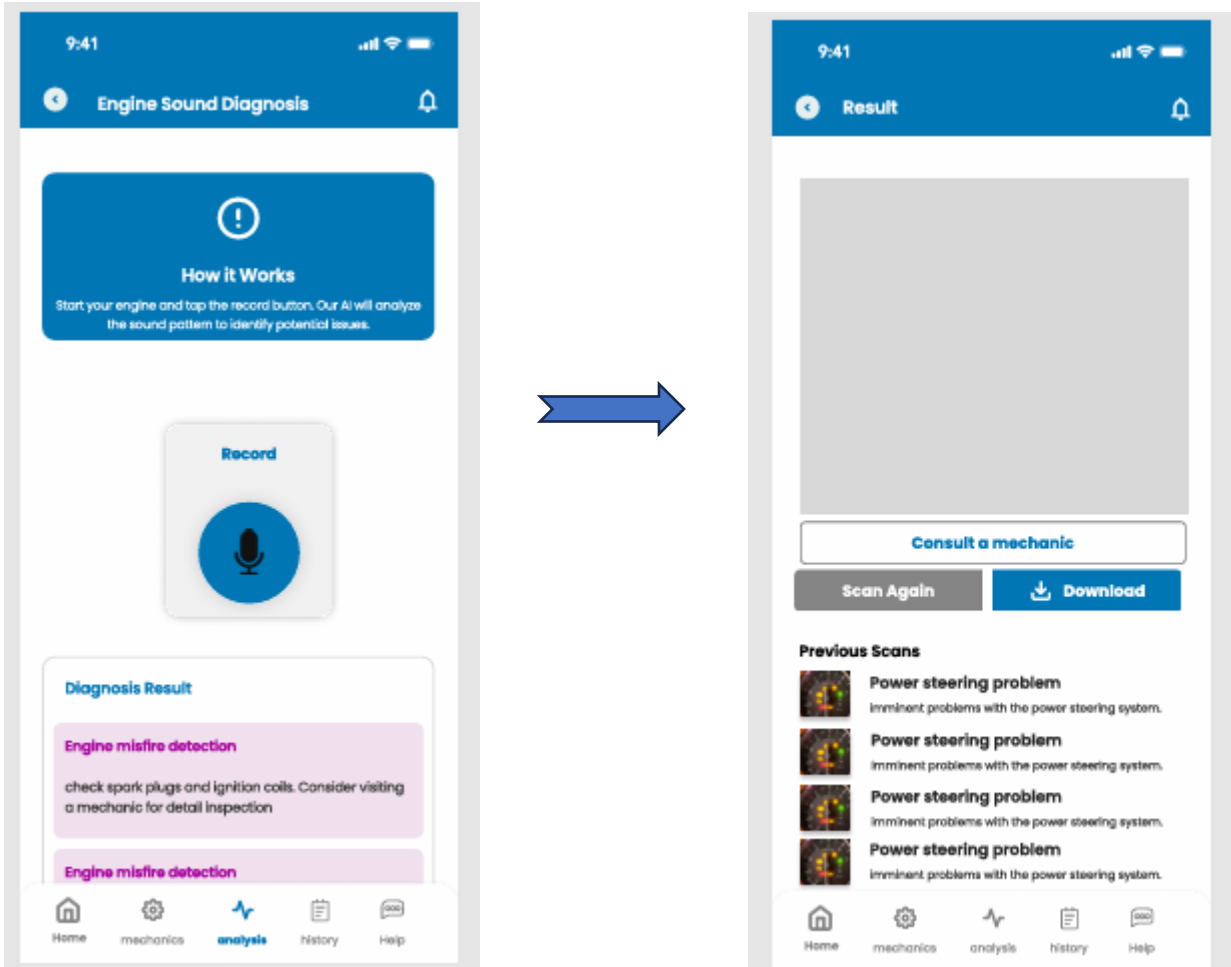
- **Microphone Access:** Prompts user for microphone permission.
- **Record Sound Button:** A clear button to start/stop audio recording.
- **Recording Timer/Indicator:** Shows recording duration.
- **Sound Analysis Progress:** "Analyzing sound..." message after recording.
- **Sound Library/Reference:** A database of common engine sounds associated with faults.

- **Diagnosis Result:**

- Identified sound category .
 - Potential causes.
 - Likely components involved.
 - Severity/Urgency.
 - Recommended Actions (e.g., "Check serpentine belt," "Consult a mechanic").
- **Playback Recorded Sound:** Option to listen to the recorded sound.
 - **Tips for Recording:** Guidance on optimal recording conditions (e.g., engine off, specific location).

User Flow/Interaction:

1. **Access Function:** User navigates to "Engine Sound Diagnosis".
2. **Grant Permission:** User grants microphone access.
3. **Record Sound:**
 - User positions device near the engine.
 - Taps "Record Sound".
 - Records for recommended duration.
 - Taps "Stop Recording".
4. **Analysis & Result:**
 - App processes the recorded sound.
 - Displays the most likely diagnosis, potential causes, and recommended actions.
5. **Review & Action:** User reviews the diagnosis and decides on next steps (e.g., manual inspection, contacting a mechanic).
6. **Manual Browse:** User can browse the sound library to manually match a sound they hear.



Front End Implementation

This part details the front-end implementation of the "AutoFix Car" mobile application, a tool designed to assist users in diagnosing car faults based on engine sound. The application provides a user-friendly interface for recording engine audio, playing it back, and initiating a simulated diagnosis process, presenting results in an intuitive manner. The front-end is developed using Flutter, ensuring a consistent user experience across Android and iOS platforms.

4.1. Architecture Overview

The front-end architecture follows a modular and component-based approach, leveraging Flutter's widget tree structure.

- **Pages (Screens):** Each major view of the application is encapsulated within a StatefulWidget or StatelessWidget (e.g., EngineSoundDiagnosisPage, MechanicsPage, ProfilePage, WelcomePage, LoginPage, RegisterPage).
- **Widgets:** Reusable UI components are abstracted into dedicated widget files (e.g., HowItWorksCard, RecordButtonCard, DiagnosisResultCard, MechanicCard, MapSection, PersonalInfoCard, ProfileHeader). This promotes code reusability, maintainability, and separation of concerns.
- **Models:** Data structures are defined as Dart classes (e.g., Mechanic, MapLocation, UserProfile) to represent the application's data entities.
- **Constants:** Centralized definitions for application-wide constants such as colors (AppColors) and text styles (AppStyles) ensure design consistency and ease of theming.
- **State Management:** Local state management within StatefulWidget is primarily handled using setState(), suitable for managing UI interactions and data within individual screens.

4.2. Key Features Implemented

The current front-end implementation includes the following core functionalities:

- **Engine Sound Diagnosis Flow:**
 - **Instructions (HowItWorksCard):** Provides clear, step-by-step guidance on how to use the diagnosis feature.
 - **Audio Recording (RecordButtonCard):** Allows users to start and stop recording engine sounds using the device's microphone. Visual feedback (button size, color, icon changes) indicates the recording status.
 - **Audio Playback:** Enables users to listen to the recorded audio before proceeding with the diagnosis.
 - **Diagnosis Processing:** Simulates an analysis phase with a loading indicator, providing a visual cue that the system is processing the audio.
 - **Diagnosis Results (DiagnosisResultCard):** Displays simulated diagnosis outcomes, categorizing them (e.g., "Misfire Detected," "Normal Operation") with distinct visual indicators (background and text color).
 - **Record New Sound:** Provides an option to clear previous results and start a new recording session.

- **Mechanics Listing and Map Integration (MechanicsPage):**
 - Displays a list of mechanics with their details (name, address, rating, verification status).
 - Integrates a map section with interactive markers for key locations.
 - Allows users to search for mechanics.
 - Provides direct "Message" and "Call" options for each mechanic.
 - Includes a dropdown for selecting regions in Cameroon.
- **User Profile Management (ProfilePage):**
 - Presents user's personal information (name, ID, location, email, car model, contact).
 - Includes Edit Profile functionality with pre-filled input fields for user data modification.
 - Provides a Change Password option.
 - Features a logout mechanism.
- **Authentication Flow (WelcomePage, LoginPage, RegisterPage):**
 - A welcoming introductory screen.
 - Login and registration forms with basic validation.
 - Navigation between authentication screens and to the main application.
- **Global UI Elements:**

Consistent AppBar across pages with back navigation and notification icons.

Theming applied consistently using AppColors and AppStyles.

4.3 UI/UX Design Principles

The front-end design prioritizes clarity, ease of use, and visual appeal:

- **Color Palette (AppColors):** A defined color scheme (primary blue, accent yellow, various shades of grey for backgrounds and text) ensures a cohesive and branded look. Dynamic colors are used for diagnosis results (green for normal, red for errors) to provide immediate visual feedback.

- **Typography (AppStyles):** Consistent text styles (headlines, body text, button text, small text) are applied throughout the application, improving readability and hierarchical organization of information.
- **Layout and Spacing:** Judicious use of Padding, SizedBox, Row, and Column widgets ensures proper alignment, adequate spacing between elements, and a clean, uncluttered interface.
- **Responsiveness:** The use of Expanded and Flexible widgets, along with SingleChildScrollView, helps ensure the layout adapts gracefully to different screen sizes and orientations.
- **Intuitive Interactions:**
 - The RecordButtonCard provides clear visual cues for recording status.
 - ElevatedButton and OutlinedButton are used for primary and secondary actions, respectively, with consistent styling.
 - DropdownButton for location selection offers a standard and accessible way to choose options.
 - SnackBar notifications provide ephemeral feedback for user actions (e.g., "Recording started," "Profile updated").

4.4. Technical Implementation Details

- **Flutter Framework:** The entire front-end is built using Flutter, leveraging its declarative UI paradigm and hot-reload capabilities for rapid development.
- **External Packages:**
 - record: Used for direct access to the device's microphone for real-time audio recording. It handles the complexities of audio input, format, and saving to a temporary file.
 - just_audio: Provides robust audio playback capabilities, allowing the user to listen to the recorded engine sound.
 - path_provider: Essential for obtaining platform-agnostic paths to temporary directories, where recorded audio files are stored.
 - permission_handler: Crucial for requesting and managing microphone permissions, ensuring the app complies with device security policies.
- **State Management:** StatefulWidgets and their State classes are used to manage dynamic UI elements and data, with setState() triggering UI updates.

- **Error Handling:** Basic error handling is implemented for audio recording and playback operations using try-catch blocks, providing user feedback via Snackbar messages.
- **Image Handling:** Image.network is used for profile and mechanic images, with errorBuilder providing a fallback icon if the image fails to load.

4.5. Future Enhancements

- **Real-time Diagnosis Integration:** Replace simulated diagnosis with actual API calls to a backend service that performs real-time engine sound analysis using AI/ML models.
- **Detailed Diagnosis Reports:** Expand DiagnosisResultCard to link to more detailed information about specific faults, including troubleshooting steps, common causes, and recommended parts.
- **Map Interactivity:** Enhance the MapSection with more interactive features, such as zooming, panning, and displaying mechanic locations dynamically.
- **Push Notifications:** Implement real-time push notifications for diagnosis results, mechanic responses, or service reminders.
- **Offline Capability:** Allow basic functionality (e.g., recording) to work offline, with data syncing when connectivity is restored.
- **Accessibility:** Further improve accessibility features for users with disabilities.

Conclusion

The user interface (UI) design and implementation of the car fault diagnosis app play a pivotal role in ensuring usability, accessibility, and efficiency for both technical and non-technical users. The UI was developed with a focus on intuitive navigation, clear visual hierarchy, and responsive design to accommodate various screen sizes and user contexts. Key diagnostic features were prioritized through logical layout structures and meaningful iconography, enhancing the user experience during fault detection and resolution.