****

**REPUBLIC OF CAMEROON**

PEACE-WORK-FATHERLAND

THE UNIVERSITY OF BUEA

P.O Box 63,

Buea, South West Region

Cameroon

Tel: (237) 3332 21 34/3332 26 90

Fax: (237) 3332 22 72

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**Dean: Agbor Dieudonne Agbor PhD**

**Vice-Dean : Nde D. Nguti, PhD**

**Faculty Officer : Mme Ekpombang Patience Taku**

Date:

**CEF440 – TASK 2:**

**REQUIREMENT GATHERING**

**FOR A ROAD SIGN AND ROAD STATE MOBILE NOTIFICATION APPLICATION**

Presented by

**GROUP 4**

|  |  |  |
| --- | --- | --- |
| SN | NAME | MATRICULE N° |
| 1 | DJOUMESSI NGUEFACK IVAN | FE21A174 |
| 2 | NJOYONG GODWILL | FE21A273 |
| 3 | TETUH WIBINAG ENGONWEI | FE21A322 |
| 4 | DJOUMESSI DONGMO RONSARD CARNEGIE | FE21A172 |
| 5 | GATCHUINNE MOMO MURIELLE CYNTHIA | FE21A200 |

**Valery Nkemeni, PhD**

**Course Supervisor**

Table des matières

[Designing and Implementing of a Road Sign and Road State Mobile Notification Application. 3](#_Toc164631793)

[Introduction 3](#_Toc164631794)

[Project objectives 3](#_Toc164631795)

[Requirement Gathering 5](#_Toc164631796)

[Definition 5](#_Toc164631797)

[Steps Involved in Requirement Gathering 5](#_Toc164631798)

[Step 1: Preparation 5](#_Toc164631799)

[Step 2: Gathering Requirements 5](#_Toc164631800)

[Step 3: Refining Requirements 5](#_Toc164631801)

[Step 4: Approval and Baselining 5](#_Toc164631802)

[Step 5: Ongoing Management 5](#_Toc164631803)

[Why is Requirement Gathering Done. 6](#_Toc164631804)

[When is Requirement Gathering Done 6](#_Toc164631805)

[How is Requirement Gathering Done 7](#_Toc164631806)

[Requirement Gathering Tools 7](#_Toc164631807)

[Communication and Collaboration Tools: 8](#_Toc164631808)

[Documentation and Feedback Tools: 8](#_Toc164631809)

[Types of Requirements Gathering 9](#_Toc164631810)

[1. Business Requirements: 9](#_Toc164631811)

[2. Stakeholder (User) Requirements: 9](#_Toc164631812)

[3. Solution Requirements: 9](#_Toc164631813)

[ Functional Requirements: 9](#_Toc164631814)

[ Non-Functional Requirements: 9](#_Toc164631815)

[4. Transition Requirements: 10](#_Toc164631816)

[Functional Requirements Breakdown for a Road Sign and Road State Application 10](#_Toc164631817)

[1. Road Condition Monitoring: 10](#_Toc164631818)

[2. Data Aggregation and Processing: 11](#_Toc164631819)

[3. Real-time Updates: 11](#_Toc164631820)

[4. Incident Detection and Reporting: 11](#_Toc164631821)

[5. Route Planning and Optimization: 12](#_Toc164631822)

[6. Historical Data Analysis: 12](#_Toc164631823)

[7. User Feedback and Collaboration: 12](#_Toc164631824)

[8. Integration with Navigation Systems: 12](#_Toc164631825)

[9. Data Visualization: 13](#_Toc164631826)

[Non- functional Requirements Breakdown for a Road Sign and Road State Application 13](#_Toc164631827)

[Performance 13](#_Toc164631828)

[Usability: 13](#_Toc164631829)

[Security: 14](#_Toc164631830)

[Data Management: 14](#_Toc164631831)

[Additional Considerations: 14](#_Toc164631832)

Design and Implementation of a Road Sign and Road State Mobile Notification Application.

# **Introduction**

Current traffic systems struggle to keep drivers informed and safe. Road signs and updates are often outdated or limited. This project tackles these issues by developing a mobile app for road signs and conditions. The app will use phone technology and live data to give drivers up-to-date information on signs, dangers, construction, and anything else that might affect their safety on the road. It will use GPS and reports from other drivers to deliver the most relevant information.

## Project objectives

* The project will result in the creation of a user-friendly mobile application designed to provide drivers with intuitive access to road sign information and real-time updates on road conditions. The application will feature a clean and responsive interface optimized for use on smartphones and tablets.
* The application will incorporate a database of road signs on major highways. This functionality will enable users to receive instant information about the meaning and significance of various road signs encountered during their travels.
* The developed application will leverage real-time data sources, including traffic cameras, weather sensors and crowdsourced reports, to provide users with timely updates on road conditions such as traffic congestion, accidents, weather-related hazards, and road closures.
* Users will have the ability to customize their notification preferences based on their specific preferences and travel preferences. They can choose to receive alerts for specific types of road signs, road conditions, or geographical areas, allowing for personalized and relevant information delivery.
* The application will integrate seamlessly with popular navigation systems and mapping platforms, allowing users to access road sign information and road state updates within their preferred navigation app.



*Figure 1.0*

Definition of key terms.

|  |  |
| --- | --- |
| Term | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Requirement Gathering

# Definition

Requirement gathering, also sometimes called requirements elicitation, is the foundation of any successful project. It's the process of uncovering exactly what needs to be built and why. In simpler terms, it's about understanding the problem you're trying to solve and the goals you're aiming to achieve and avoid confusion, wasted resources, and ultimately, project failure.

## Steps Involved in Requirement Gathering

### Step 1: Preparation

* Identify stakeholders (who cares about the project?)
* Plan your approach:
  + How will you gather information (interviews, surveys, etc.)?
  + How will you document everything?
  + How will you finalize requirements with stakeholders?
* Schedule meetings and brief stakeholders.

## Step 2: Gathering Requirements

* Create a central requirements document.
* Include project goals, scope, stakeholder needs, risks, and assumptions.

## Step 3: Refining Requirements

* Review and analyse requirements against project goals.
* Prioritize requirements based on importance.
* Document everything clearly, including assumptions and quality control methods.

## Step 4: Approval and Baselining

* Get stakeholder sign-off on the requirements document.
* Use the document to define project scope and track progress.
* Create a traceability matrix to link requirements to deliverables.

## Step 5: Ongoing Management

* Ensure your team works towards meeting the requirements.
* Use the traceability matrix to manage changes effectively.
* Evaluate new requirements emerging from testing or quality checks.



*Figure 1.2*

# Why is Requirement Gathering Done.

There are many reasons why requirement gathering is carried out. Here are a few of why it is crucial:

* **Defines Direction:**

Clear requirements set the course for the project, ensuring everyone understands the goals and what's being built.

* **Minimizes Risks:**

Early identification of potential issues allows you to mitigate them before they cause problems later.

* **Improves Efficiency:**

Knowing what's required lets teams work efficiently and avoid rework due to unclear needs.

* **Enhances Satisfaction:**

Well-defined requirements lead to projects that deliver what stakeholders truly need, resulting in higher satisfaction.

# When is Requirement Gathering Done

Requirement gathering primarily happens during the **initiation phase** of a project lifecycle. This is the initial stage where the project is conceived, and the groundwork is laid for execution. Here's why it's crucial at this point:

* **Sets the Stage:**

It establishes the project's direction and ensures everyone involved is on the same page from the very beginning.

***Importantly:*** Requirement gathering isn't a one-time event. As the project progresses:

* New information may emerge.
* Needs may change.

Therefore, requirement gathering can be an **iterative process**, revisited throughout the project lifecycle to ensure everything stays on track.

# How is Requirement Gathering Done

There are various techniques to identify and gather requirements. Here are some common approaches:

* **Brainstorming**:

Bringing stakeholders together to discuss the problem and potential solutions, fostering creativity and generating initial ideas.

* **Interviews**:

One-on-one conversations with stakeholders to delve deeper into their specific needs, expectations, and pain points. Useful for smaller projects.

* **Questionnaires**:

 Surveys distributed to a larger group to gather feedback and opinions. They can be anonymous, allowing for more honest responses.

* **Prototypes**:

Creating a working model of the proposed solution for stakeholders to test and provide feedback. Helps identify usability issues early.

* **Document Review**:

Analysing existing project plans, company strategies, and technical documents to gain context and background information.

# Requirement Gathering Tools

There are several tools used in this process, some effective tools you can leverage for requirement gathering are as follows:

## Communication and Collaboration Tools:

* **Meeting Software**:

 Platforms like Zoom, Google Meet, or Microsoft Teams facilitate online meetings with stakeholders (drivers, traffic authorities, etc.) to discuss needs and pain points.

* **Project Management Software:**

Tools like Asana, Trello, or Monday.com help organize discussions, store notes, and collaborate on drafts with stakeholders.

## Documentation and Feedback Tools:

* **Document Sharing Platforms:**

Cloud storage services like Google Drive or Dropbox allow sharing documents (surveys, mock-ups) with stakeholders for feedback and version control.

* **Mind Mapping Software:**

Tools like Miro or Mind Meister help visually represent user flows, brainstorm ideas, and organize requirements collaboratively.

**User Research Tools:**

* **Surveys and Forms:**

 Platforms like SurveyMonkey or Google Forms enable gathering data and feedback from a large group of users (drivers) on existing road signs and desired features.

* **Heatmaps and Analytics Tools:**

If you have access to anonymized data on existing traffic applications or in-car navigation systems, heatmaps can reveal areas with frequent confusion or accidents, highlighting potential needs for the app.

***Additionally***:

* **Government and Traffic Authority Websites**:

Reviewing official resources about road signs and traffic regulations can provide insights into existing standards and potential improvements.

* **Industry Reports and News:**

Researching industry reports and news articles can reveal trends, challenges, and upcoming advancements in traffic management technology, informing your app's functionalities.

It is important to note that, the best tool selection depends on your project's specific needs, budget, and stakeholder preferences.

# Types of Requirements Gathering

## 1. Business Requirements:

Considering a state transportation department looking to improve road safety. Their high-level goals might be:

* Reducing traffic accidents
* Keeping drivers informed about road conditions
* Improving traffic flow

These objectives would be documented in a business requirements document to explain the project's purpose to everyone involved (developers, government officials, etc.)

## 2. Stakeholder (User) Requirements:

* **Drivers:**

Need real-time updates on accidents, construction zones, and weather conditions. They might also want features like navigation assistance or hands-free reporting capabilities.

* **State Agencies:**

Might need the app to integrate with existing traffic monitoring systems and allow for quick updates on road closures or hazards.

These needs would bridge the gap between the high-level business goals and the specific functionalities of the app.

## 3. Solution Requirements:

This translates to the app's features and technical specifications:

* Functional Requirements:

The app must display road signs, offer real-time traffic information, and allow users to report incidents.

* Non-Functional Requirements:

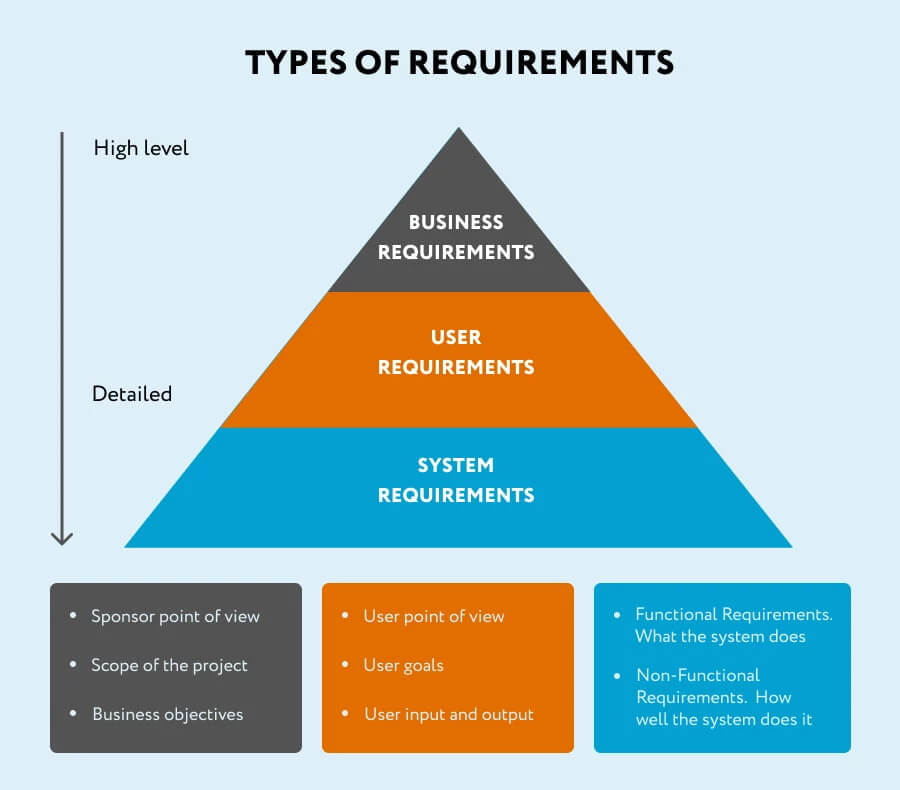
The app should be user-friendly, operate reliably, and be compatible with various devices.

These detailed requirements ensure everyone is on the same page about what the app should do and how it should perform.

## 4. Transition Requirements:

These are temporary steps needed to launch the app smoothly:

* Developing user training materials
* Partnering with cellular providers to ensure widespread accessibility
* Integrating the app with existing traffic management systems

These steps would help bridge the gap between the current situation (static road signs) and the future state where drivers have access to real-time information.

*Figure 1.3*

# Functional Requirements Breakdown for a Road Sign and Road State Application

This section dives deeper into the functional requirements mentioned previously, providing additional details and considerations:

## ****1. Road Condition Monitoring:****

* **Data Sources:**
  + Sensors embedded in roads can detect traffic flow, weather conditions (e.g., rain, snow, fog), and even road surface temperature.
  + Weather reports and forecasts provide insights into potential hazards like storms or icy conditions.
  + User-submitted reports can alert others about accidents, potholes, or debris on the road.
* **Data Collection Methods:**
  + The application should have mechanisms to securely connect with various data sources through APIs or dedicated communication protocols.
  + User reports can be submitted through a user interface within the app, potentially with options to include photos or videos for verification.

## ****2. Data Aggregation and Processing:****

* **Data Validation and Cleaning:**
  + The app should implement algorithms to verify the accuracy and consistency of data received from different sources. This might involve removing outliers, correcting inconsistencies, and potentially flagging reports requiring further investigation.
* **Data Fusion:**
  + The application needs to combine data from various sources to create a comprehensive picture of road conditions. This might involve weighting data based on reliability or applying machine learning techniques to identify patterns and trends.

## ****3. Real-time Updates:****

* **Update Frequency:**
  + The app should determine the optimal frequency for updating road conditions based on factors like traffic volume and the nature of the reported incident. Critical issues like accidents might require immediate updates, while less urgent conditions can be updated at regular intervals.
* **Delivery Methods:**
  + The app should offer users options for receiving real-time updates. This could include visual alerts on a map, text messages, voice notifications, or integration with existing navigation applications.

## ****4. Incident Detection and Reporting:****

* **Automated Detection:**
  + The app can leverage data analytics to identify potential incidents based on sudden changes in traffic flow, sensor readings, or a surge in user reports from a specific location.
* **User-reported Incidents:**
  + The app should provide a user-friendly interface for reporting accidents, hazards, or other road issues. This might include options to specify the location, type of incident, and even upload supporting evidence like photos or videos.
* **Integration with Emergency Services:**
  + The app should have mechanisms to seamlessly report confirmed incidents to the appropriate authorities, potentially including functionalities for users to directly contact emergency services.

## ****5. Route Planning and Optimization:****

* **Dynamic Routing:**
  + The app should consider real-time road conditions when suggesting routes to users. This might involve factoring in traffic congestion, road closures, accidents, and even weather conditions to provide the fastest or safest route.
* **Alternative Route Suggestions:**
  + The app should offer users options for alternative routes if the primary route is significantly impacted by an incident. This allows users to make informed decisions based on their priorities and risk tolerance.
* **User Preferences:**
  + The app might allow users to set preferences for route optimization. For example, some users might prioritize the fastest route even if it has some traffic congestion, while others might prefer a slower route with fewer obstacles.

## ****6. Historical Data Analysis:****

* **Data Storage:**
  + The app should store historical road state information securely for future analysis. This data can be anonymized to protect user privacy while still providing valuable insights.
* **Data Analysis Tools:**
  + The app can leverage data analytics tools to identify trends and patterns in road conditions over time. This might reveal areas prone to accidents, recurring traffic congestion during specific times, or the impact of weather events on road safety.

## ****7. User Feedback and Collaboration:****

* **Feedback Mechanisms:**
  + The app should allow users to provide feedback on the accuracy and usefulness of the road state information they receive. This feedback can be used to improve the data collection, processing, and presentation methods.
* **Community Reporting:**
  + The app can facilitate collaboration among users by allowing them to report incidents and road conditions in real-time. This crowdsourced information can help to create a more comprehensive and up-to-date picture of the road network.

## ****8. Integration with Navigation Systems:****

* **API Integration:**

The app should provide a well-documented API to enable seamless integration with existing navigation systems or GPS devices. This allows users to access road state information directly within their preferred navigation app for a smooth and unified user experience.

* **Data Sharing Protocols:**

The app should establish secure data sharing protocols to ensure the reliable and consistent exchange of road state information with navigation systems.

## ****9. Data Visualization:****

* **Clear and Concise Presentation:**

The app should present road state information in a clear, concise, and easy-to-understand manner. This might involve using color-coded maps, intuitive icons, and clear legends to represent different types of road conditions and their severity.

* **Customization Options:**

The app might allow users to customize the data visualization based on their preferences. This could include filtering information based on

# Non- functional Requirements Breakdown for a Road Sign and Road State Application

This section dives deeper into the non-functional requirements mentioned previously, providing additional details and considerations:

Performance:

* **Responsiveness:**
  + The app should load quickly and respond to user actions instantly to avoid frustration while driving. This is critical for safety reasons.
* **Low Latency Notifications:**
  + Real-time updates on road conditions need to be delivered with minimal delay to ensure users have the latest information for safe navigation.
* **Battery Efficiency:**
  + Since the app might run in the background for extended periods, it should be optimized to minimize battery consumption on mobile devices.

## Usability:

* **Intuitive Interface:**
  + The user interface (UI) should be clean, uncluttered, and easy to navigate, even for users with limited experience with smartphones. Icons and buttons should be clear and easily recognizable.
* **Minimal Distraction:**

Visual information and notification alerts should be designed to be clear and informative at a glance, minimizing the time users need to look away from the road.

* **Voice Control Options:**
  + Consider incorporating voice commands for hands-free interaction with the app, allowing users to access information or report incidents without taking their eyes off the road.

## Security:

* **User Privacy:**
  + The app should collect minimal user data, ideally just location for notification purposes. This data should be anonymized and used only for the stated purpose of improving road safety.
* **Data Security:**
  + The app should implement robust security measures to protect user data from unauthorized access, hacking attempts, or malware. Secure data storage and transmission protocols are essential.

## Data Management:

* **Road Sign Information Source:**
  + Decide whether road sign data will be pre-loaded on the app or updated dynamically. Pre-loaded data might be suitable for static signs, while real-time updates are crucial for temporary signs or variable message boards.
* **Real-time Data Acquisition:**
  + Explore options for obtaining real-time road state data. This could involve integrating with existing traffic management systems, leveraging data from connected vehicles, or incorporating user reports.
* **Data Update Mechanism:**
  + Establish a system for updating the app's database with new signs, changes in road conditions, or any other relevant information. This might involve automatic updates or user-triggered refresh options.

## Additional Considerations:

* **Target Platforms:**

Specify whether the app will be available for Android, iOS, or both. Consider the dominant mobile operating systems in your target market.

* **Accessibility Features:**

Integrate features that cater to users with visual or auditory impairments. This could include options for larger fonts, voice alerts, or text-to-speech functionality.

* **Localization**:

If the app is intended for a global audience, consider localization for different languages and variations in road signs across different regions.

* **Offline Functionality:**

Explore the possibility of allowing users to access basic road sign information or pre-downloaded data even when offline, providing some level of functionality in areas with limited internet connectivity.

It is also important to note that, this list serves as a starting point. Further discussions with stakeholders, including potential users (drivers), can help refine and prioritize these requirements to ensure the app meets their needs effectively.

Systems Requirements Specification (SRS) Document.

# ****Introduction****

This document outlines the System Requirements Specification (SRS) for a mobile application designed to provide drivers and stakeholders with access to road sign information and real-time road state updates through notification.

## **Purpose**

The application aims to improve driver awareness and safety by offering:

* A comprehensive database of road sign definitions and meanings.
* Real-time updates on traffic conditions, accidents, and weather hazards.
* Personalized notification preferences based on user needs.
* Seamless integration with popular navigation apps.

**1.2. Document Conventions**

* Bold text highlights important requirements or user stories.

**1.3. References**

* Google maps
* GPS
* Ministry of Transport
* Team Investigation

## ****2. Overall Description****

**2.1. Product Perspective**

The target users are drivers and stakeholders who want to:

* Enhance their understanding of road signs.
* Stay informed about current road conditions during their journeys.
* Receive timely alerts about potential hazards or delays.

**2.2. Product Functions**

The application will provide the following functionalities:

* **Road Sign Information:**
  + Searchable database of road signs commonly found on major highways.
  + Clear definitions and explanations for each road sign's meaning and significance.
  + Visual representations (icons or images) of each road sign for easy identification.
* **Real-time Updates:**
  + Integration with real-time data sources (traffic cameras, weather sensors, crowdsourced reports).
  + Display of traffic congestion information on a map, with severity levels and estimated delays.
  + Alerts about accidents, road closures, and weather hazards relevant to the user's location or planned route.
  + User-defined filtering options for alerts based on severity, type, and location.
* **Navigation Integration:**
  + Seamless integration with popular navigation apps and mapping platforms.
  + Sharing of road sign information and real-time road state data with navigation apps.
  + Ability to access this information directly within the user's preferred navigation app interface.
* **User Customization:**
  + Setting notification preferences for road signs, road conditions, and specific geographical areas.
  + Choosing the type of alerts received (visual, audio, text message).
  + Filtering information displayed on the app's main interface based on user preferences.
* **Search Functionality:**
  + Search function for finding specific road signs within the database by name, symbol, or keyword description.

## ****2.3. General Constraints****

* This application would be compatible with major smartphone and tablet operating systems (e.g., Android, iOS).
* The user interface should be intuitive and user-friendly for drivers of all ages and technical skill levels.
* The application should prioritize user privacy and collect minimal user data. Any data collected should be anonymized and used solely for improving the app's functionality.
* The application should be optimized for battery efficiency to minimize drain on mobile devices.

## **3. Specific Requirements**

**3.1. Functional Requirements**

**3.1.1. Road Sign Information**

* For each road sign, the application shall display a clear and concise definition of its meaning and significance.
* The application shall include visual representations (icons or images) for each road sign to aid in identification.
* The application may offer functionalities like voice descriptions of road signs for users with visual impairments.

**3.1.2. Real-time Updates**

* The application shall integrate with real-time data sources, including traffic cameras, weather sensors, and crowdsourced reports.
* The application shall display traffic congestion information on a map, with clear indications of severity levels and estimated delays.
* The application shall provide alerts about accidents, road closures, and weather hazards (e.g., flooding, fog, icy roads) relevant to the user's location or planned route.
* The application shall allow users to define notification preferences for alerts based on severity, type (accident, closure, etc.), and geographical area.

**3.1.3. Navigation Integration**

* The application shall integrate seamlessly with popular navigation apps and mapping platforms (specify target apps if known).
* The application shall share road sign information and real-time road state data with navigation apps through a documented API.
* The application shall allow users to access road sign information and real.

## 4. External Interface Requirement

**4.1 User Interfaces**

* Intuitive and User-friendly Design: The UI should be clean, uncluttered, and easy to navigate for drivers of all ages and technical skill levels. This means using clear menus, recognizable icons, and large, easy-to-read fonts.
* Minimal Distraction While Driving: The UI should be designed to minimize distraction while driving. Information should be presented concisely and visually, requiring minimal time for users to glance at the screen.
* Customization Options: Offer user customization options for the UI. This could include:
  + Map View: Allow users to choose the level of detail displayed on the map and potentially switch between map views (e.g., standard, traffic-focused).
  + Notification Preferences: Users should be able to set notification preferences for different types of road signs, road conditions (accidents, closures), and geographical areas. They should also be able to choose the format of notifications (visual, audio, text message).
  + Information Filtering: Users should be able to filter information displayed on the main interface based on their preferences. This could involve filtering by road sign category, severity of road conditions, or specific geographical areas.

**4.2 Software Interfaces**

**1. Programming Language:**

* The choice of programming language depends on the target platforms (Android, iOS, or both) and desired functionalities. Here are some options:
  + Native Development:
    - Android: Java is our primary choice for native Android app development.
    - iOS: Swift is the preferred language for native iOS app development.
  + Cross-platform Development: Frameworks like React Native development using languages like JavaScript for apps that can run on both Android and iOS.

**2. Integrated Development Environment (IDE):**

* The choice of IDE is flexible and depends on developer preference. Popular options include:
  + Android Studio: The official IDE from Google for Android app development (supports Java and Kotlin).
  + Visual Studio Code: A versatile code editor with extensive plugin support, making it suitable for various languages and frameworks, including React Native and Flutter.

**3. Design Tools:**

* UI/UX design plays a crucial role in this app. Popular design tools to consider include:
  + Figma: A web-based design tool for creating user interfaces and prototypes.
  + Star UML:  A popular design tool for creating user interfaces, especially for mobile apps

**4. External APIs and Data Sources:**

* The app relies on external APIs to connect with real-time data sources and potentially integrate with navigation apps. These APIs will have their own documentation and specifications that need to be followed during development.

**5. Version Control System:**

* Using a version control system like Git is crucial for managing code changes, collaboration among developers, and easy rollback if needed. Popular options include GitLab and GitHub.

**4.3 Hardware Interfaces**

**1. User Devices**

* Smartphones and Tablets: These are the primary hardware platforms where the app runs. The app interacts with the device's functionalities like:
  + Operating System (OS): The app needs to be compatible with the target OS, most likely Android or iOS.
  + GPS: The app uses the device's GPS hardware to determine the user's location, allowing it to display relevant road sign information and real-time road state data.
  + Internet Connectivity: The app relies on internet connectivity to access real-time data sources (traffic cameras, weather sensors) and potentially download updates.
  + Display: The app utilizes the device's display to present road sign information, real-time road state data, and notifications to the user.
  + Speaker/Audio Output: The app might use the device's speaker to deliver audio notifications about road conditions or alerts.

**2. External Data Sources (Indirect Hardware Involvement):**

* Traffic Cameras: While the app doesn't directly control traffic cameras, it interacts with their data through APIs. Traffic cameras are the hardware that captures real-time traffic flow information.
* Weather Sensors: Similar to traffic cameras, the app retrieves weather data (temperature, precipitation, etc.) through APIs provided by weather monitoring networks. Weather sensors are the hardware that collects this environmental data.

## 5 Quality Attributes

**5.1 Non-Functional Requirement**

* **Performance**
* Responsiveness:
  + The app should be lightning-fast, loading quickly and responding to user actions instantly to avoid frustration and ensure safety while driving.
* Low Latency Notifications:
  + Real-time updates on road conditions need to be delivered with minimal delay to empower users with the most recent information for safe navigation.
* Battery Efficiency:
  + As the app might run in the background for extended periods, it should be optimized to minimize battery drain on mobile devices.
* **Usability:**
* Intuitive Interface:
  + The user interface (UI) should be clean, uncluttered, and a breeze to navigate, even for users who aren't tech-savvy. Icons and buttons should be clear and instantly recognizable.
* Minimal Distraction:
  + Visual information and notification alerts should be designed for quick comprehension at a glance, minimizing the time drivers need to take their eyes off the road.
* Voice Control Options (Optional):
  + Consider incorporating voice commands for hands-free interaction, allowing drivers to access information or report incidents without compromising their focus on the road.

Functional Decomposition and Work Breakdown.

A functional decomposition is the process of breaking down a complex problem, system or structure into a simpler and more understandable parts. In software engineering, functional decomposition helps creates a detailed visual representation of a system’s functionality known as the work breakdown structure.

\*

Functional Decomposition

# Conclusion

//We have successfully outlined the requirements for a mobile application designed to enhance

# References

1. <https://www.brightwork.com/blog/project-requirements>
2. <https://www.geeksforgeeks.org/10-most-popular-requirement-gathering-tools-for-2024/>
3. <https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/what-is-requirements-gathering>
4. <https://www.altexsoft.com/blog/functional-requirements/>
5. <https://www.mindk.com/blog/requirements-gathering-process/>
6. <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>