Republic of Cameroon
******
Peace - Work - Fatherland
*****
Minister of Higher Education
*****
University of Buea

Republique du Cameroun
******
Paix - Travail - Patrie
*****
Ministere de l'enseignement Superieur
*****
Universite de Buea

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

MOBILE APPLICATION DEVELOPMENT

CEF 440

TASK 2 REPORT:

ROAD SIGN AND ROAD STATE NOTIFICATION MOBILE APP

REQUIREMENT ANALYSIS

Presented by:

GROUP 8

N°	NAMES	MATRICULES
1	FOWEDLUNG ATSAFAC AGAFINA	FE21A196
2	KAMDEM KAMGAING GILLES CHRISTIAN	FE21A209
3	NEGUE KWAHAM MAEL GRACE	FE21A252
4	NGUEPI GNETEDEM PATERSON	FE21A264
5	NOUPOUWO DONGMO STEPHANE MERCI	FE21A283

Course Instructor: Dr Valery Nkemeni

April 2024

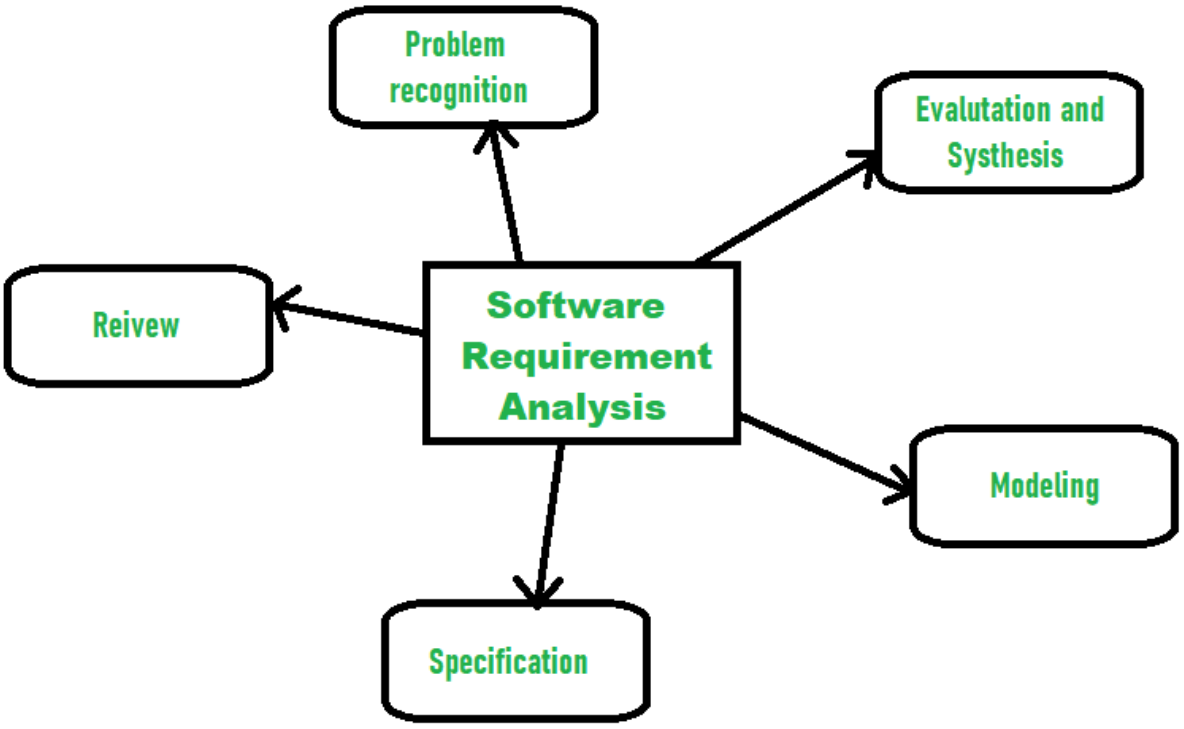
TABLE OF CONTENTS

I- INTRODUCTION

In the domain of software development, requirement analysis assumes a pivotal role akin to laying foundational groundwork prior to construction. It entails a systematic and meticulous examination aimed at comprehensively understanding the requisite functionalities necessary to effectively address a given problem or necessity. Essentially, it encompasses the process of deciphering the intricate amalgamation of user needs and system specifications to articulate a clear and concise set of requirements.

This document serves as an exhaustive guide to requirement analysis, offering an in-depth exploration of its fundamental tenets and methodologies. Throughout this discourse, we shall unravel the intricacies inherent in this phase of software development, elucidating each step with objectivity and precision.

From the initial identification of user requisites to the prioritization and validation of requirements, we shall traverse the diverse stages of requirement analysis, elucidating the methodologies and techniques employed to ensure a comprehensive comprehension of the software's objectives.



II- PROBLEM STATEMENT

Navigating roads efficiently and safely is a universal concern for drivers, yet current methods of accessing critical information about road conditions, weather hazards, and real-time traffic updates often fall short. Users face challenges in receiving timely notifications tailored to their preferences.

In response to these challenges, there is a clear need for a robust road state notification application that provides users with accurate and customizable information in real-time. By addressing the gaps in existing solutions and incorporating features such as road sign notifications, weather hazard alerts, customizable notification preferences, and offline functionality, the project aims to empower drivers with the tools they need to navigate roads safely and efficiently.

The main aim of requirement analysis is to fully understand main objective of requirement that includes why it is needed, does it add value to product, will it be beneficial, does it increase quality of the project, does it or will it have any other effect. All these points are fully recognized in problem recognition so that requirements that are essential can be fulfilled to solve business problems.

III- EVALUATION AND SYNTHESIS

Evaluation involves making judgments about the value or worth of something, while synthesis involves creating or forming something new. In the context of software requirement evaluation and synthesis, several important tasks are identified:

- Defining all the necessary functions of the software.
- Identifying and defining all externally observable data objects.
- Evaluating the flow of data to determine its significance or worth.
- Gaining a comprehensive understanding of the overall system behavior, including its functioning as a whole.
- Identifying and uncovering any constraints that have been designed into the system.
- Defining and establishing the characteristics of the system interface to understand how the system interacts with multiple components or with itself.

III.1 Selection criteria

During the requirement analysis phase of project development, rigorous selection criteria are employed to ensure the alignment of identified requirements with established standards. The selection process is guided by the SMART criteria, focusing on principles of specificity, measurability, attainability, realism, and time-bound objectives.

The figures below illustrate the requirements selection process with stakeholder preference statistics;

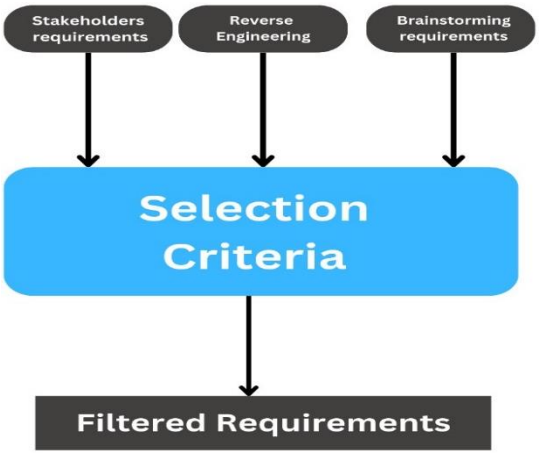


Figure 1: Requirements selection Process

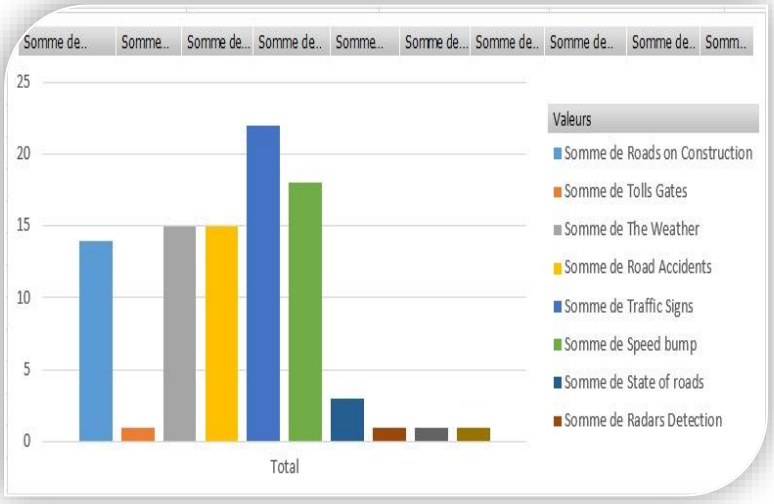


Figure 2: Stakeholders Preferences

1. ***Specificity:*** Requirements undergo meticulous articulation to ensure precision and clarity, leaving no room for ambiguity.

2. ***Measurability*****:** Each requirement is subjected to quantitative evaluation, enabling objective assessment of its fulfillment. This entails defining explicit metrics or criteria against which the completion or success of the requirement can be objectively gauged.

3. ***Attainability:*** Requirements are evaluated for their realistic achievability within the project's scope, considering available resources, expertise, and technological constraints.

4. ***Realism:*** Thorough evaluations are conducted to assess the feasibility and practicality of requirements, ensuring their alignment with project goals and constraints.

5. ***Time-Bound Objectives:*** Each requirement is associated with clearly defined timeframes for implementation or completion.

Additionally, requirements are selected based on meticulous consideration of user preferences, ensuring that identified functionalities resonate with stakeholder expectations and needs. By adhering to the SMART criteria and integrating user-centric considerations, resource utilization is optimized, mitigate risks, and achieve successful project outcomes.

III.2 Filtered Requirements

After going through the gathered requirements in the previous task and applying the selection criteria to them, the following were retained as SMART and mostly preferred by stakeholders;

1. ***Road Sign Notification*****:** This feature is chosen due to its practicality and importance in enhancing road safety, offering users valuable information without the need for manual input or interaction.

2. ***Weather Related Hazards****:* Included for its ability to provide critical updates on weather-related risks, ensuring users are well-informed about potential dangers without requiring them to actively seek out information.

3. ***Real-Time Traffic Updates****:* Selected to offer users valuable insights into traffic conditions, enabling them to make informed decisions about their travel routes based on current data and trends.

4. ***Notification Preferences:*** Implemented to allow users to customize their experience and receive alerts in a manner that suits their preferences, contributing to a more personalized and user-centric application.

5. ***Route Redirection:*** Chosen for its practicality in offering users alternative routes when necessary, minimizing the impact of traffic congestion or road closures on their travel plans.

6. ***Voice Assistance:*** Integrated to provide users with a convenient hands-free option for accessing information, ensuring they can interact with the app safely and efficiently while on the go.

7. ***Location Tracking:*** Adopted to enhance the functionality of the app by leveraging location-based services, enabling features such as geotagged notifications and location-specific recommendations.

8. ***Multilanguage Support:*** Included to accommodate users from diverse linguistic backgrounds, making the app accessible and user-friendly for a broader audience without language barriers.

9. ***Real Time Speed Counter:*** Critical for safety, the speed counter promptly notifies users if they exceed safe driving limits, fostering responsible behavior and preventing accidents.

10. ***Share Location from the App:*** Enhances user connectivity and convenience by enabling seamless sharing of locations with friends, family, or emergency services, ensuring efficient navigation and swift assistance when needed.

11. ***Defined Frequently Accessed Locations:*** Boosts user efficiency by allowing them to predefine frequently visited destinations, eliminating the need for repetitive inputs and streamlining navigation processes, saving time and effort.

12. ***Road State Update Permission:*** Permits users to contribute valuable real-time information about road conditions, accidents, and blocked roads, facilitating a safer and smoother driving experience for all community members through collective awareness and proactive measures.

13. Login and Sign-Up: Included to be able to manage users and provide them with information about their different activities within the app.

Broadly, these selected requirements are considered as necessary for a first deployment of the app but also attainable within the given time frame.

III.3 Identifying External Data Objects

It entails identifying and describing all the data objects involved and that will be used in achieving the project. The data objects considered here are as follows:

- ***Weather API***
  It provides access to weather information such as current conditions, forecasts, temperature, humidity, wind speed, and precipitation, among other data points. This will help in providing the weather-related hazards functionality.

- ***Google Maps API***
  It integrates mapping and location-based features into their applications. It will help providing access to various functionalities such as displaying maps, retrieving location data, real time traffic, wildfires, speed monitoring and calculating directions.

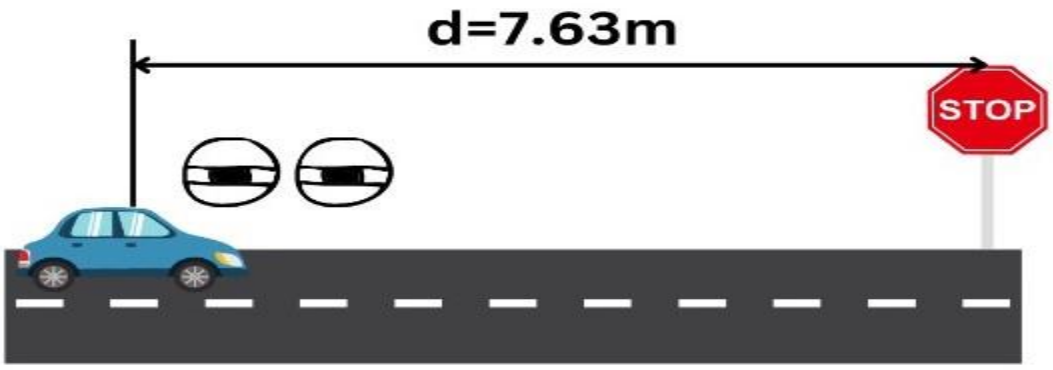
- ***Data Providers***
  These are the entities responsible for providing road state updates to the system, users in this case. They will help in updating and validating relevant data into the system about road state.
    - *Updating* in the sense that, they will be able to send information about any significant change in the road state.
    - *Validating* in the sense that, the reported changes is notified to users in a closed perimeter to confirm the information prior to broadcasting.

- ***Road Sign Position and Description Database***
  This refers to a database containing all deployed road signs positions on the national territory including their descriptions.

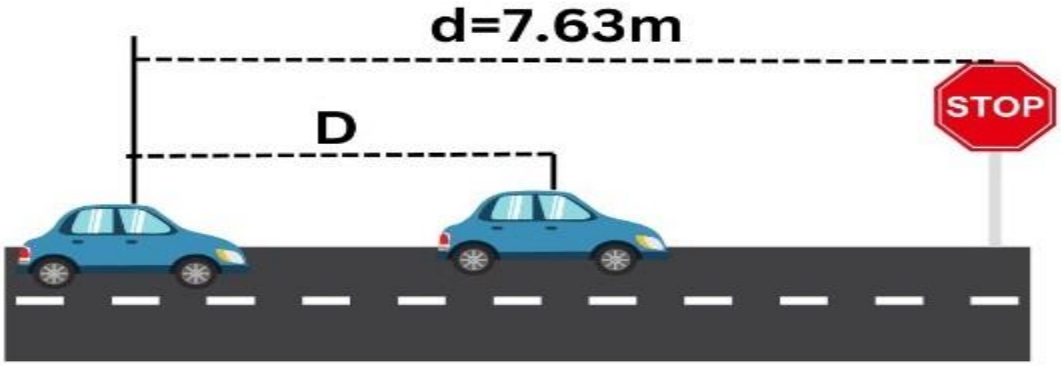
III.4 Interaction of External Data Objects with the System

This involves demonstrating how the different external components interact with each other and the system. This will be demonstrated with the aid of two different scenarios.

Scenario 1



From MIT research, a driver notices a road sign at 7.63meters from it.

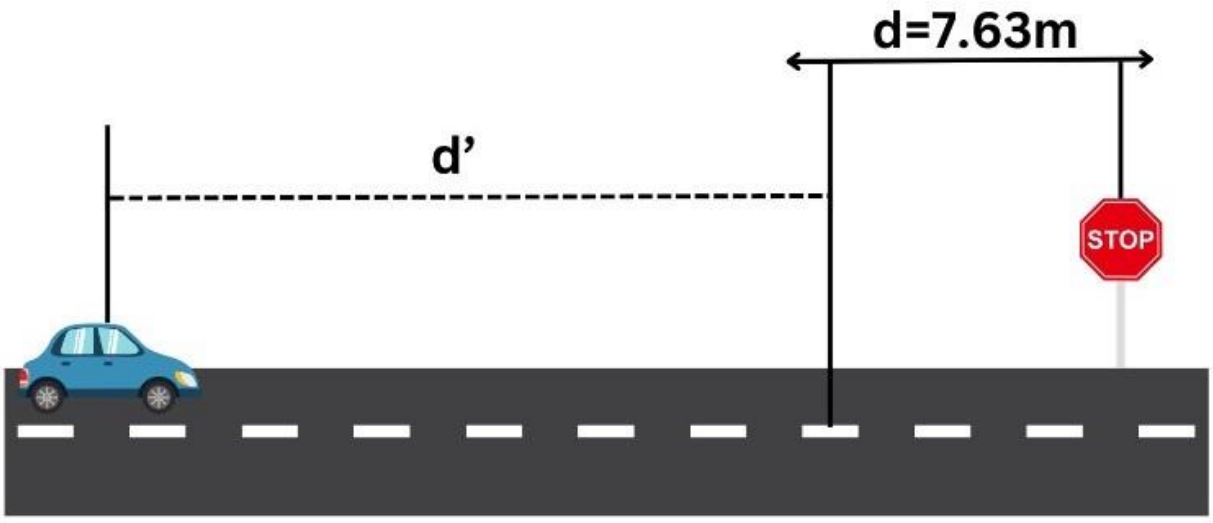


It takes the driver on average from MIT research 0.75s to notice a sign. And as this time goes, the car displacement changes depending on the car's velocity.

$$distance\ moved, D = \frac{dv}{dt} * 0.7$$

How close is the driver to the road sign $New\ distance\ to\ sign = 7.63 - D$

And this is not very effective as the driver was not aware of the abrupt changes he was to undergo.

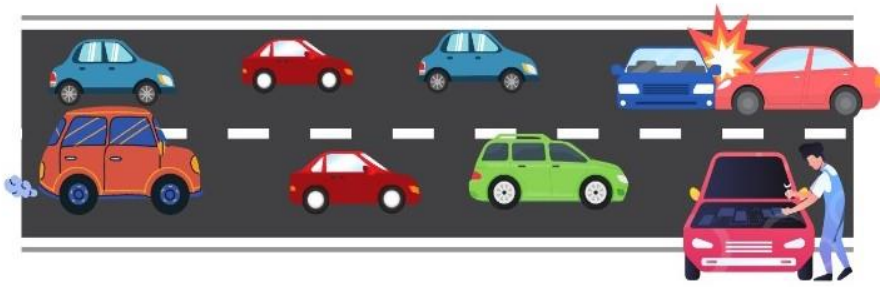


But our system keeps track of the driver's speed and knows the distance to the next road sign. With this, we can prepare the drivers for these activities.

So, depending on his velocity and the distance to the distance of sight to the road sign we can calculate the approximate time we will see the road sign so along d' our app notifies the driver. $\frac{d'}{s} = t$

Scenario 2

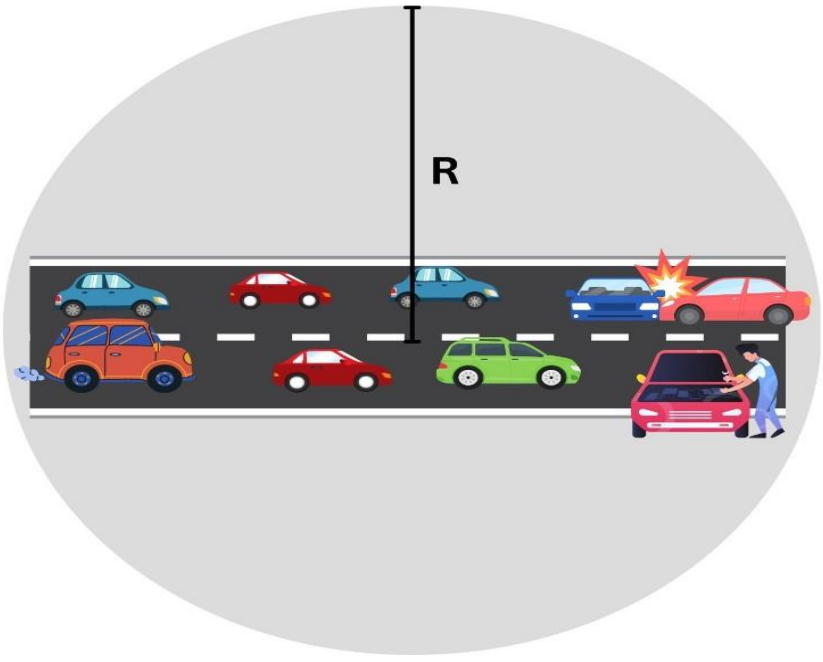
Traffic congestion occurs due to an event on the road. People in traffic congestion will help us to know exactly what is happening there.



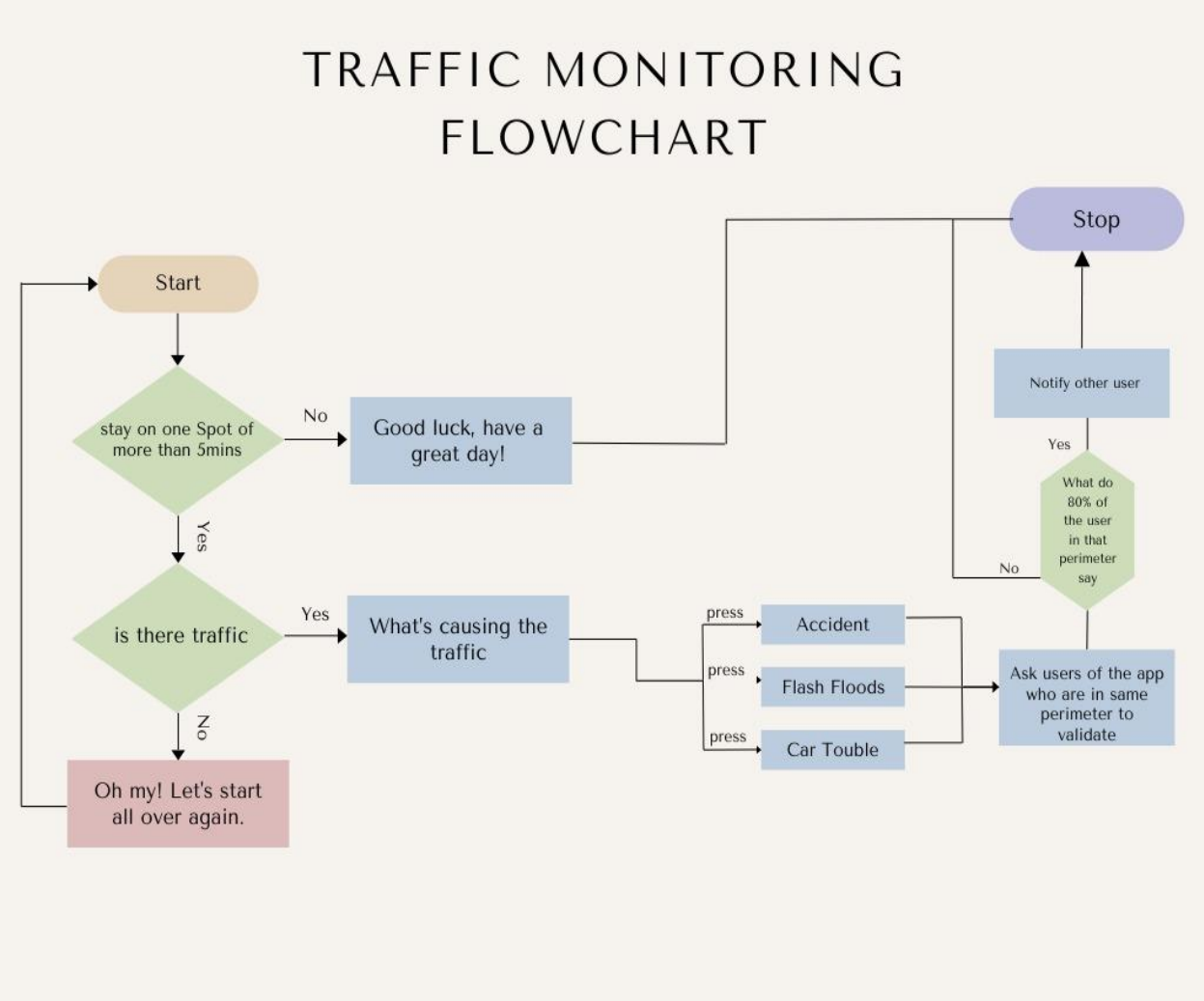
If the user stays on one spot for example 5 minutes while using the app, there are possibilities that he stopped either to buy food or to ease himself or due to something occurring which caused traffic.

Our app will ask him after the 5minutes **why are you on one spot for long, is there traffic, hazards? Please do you mind sharing?**

If he shares, to be sure that the information is valid, we share the question **is it true that there is traffic on this road?** to 10 of our users which are in the same perimeter. If 80% of their answers is **yes**, we notify people using our app at an extended radius of $R + r$ and provide them with route redirection.

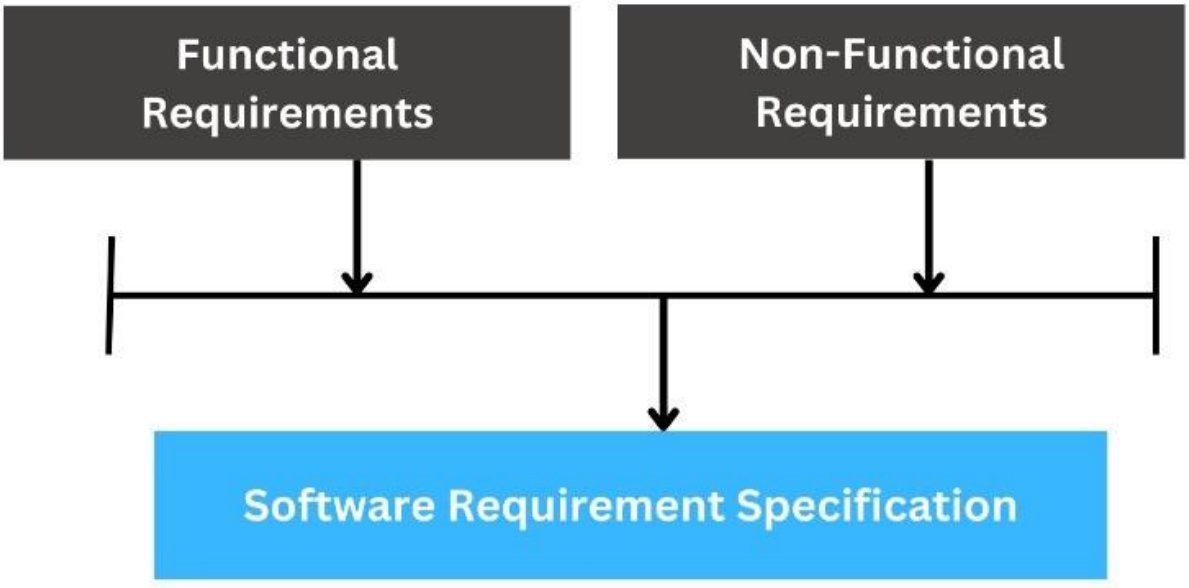


This scenario is summarized in the diagram below;



IV- SOFTWARE REQUIREMENT SPECIFICATIONS

After going through thorough analysis, clear and concise requirements were obtained. This section aims at classifying and fully describing the requirements that shall be implemented for the purpose of this project specifying the acceptance criteria and constraints (in some cases) to each of the requirements. These requirements are hence classified as follows;



IV.1 Functional Requirements

These specify the specific behaviors and functions the software must perform. They outline what the system should do in terms of inputs, processes, outputs, and interactions with users or other systems. The functional requirements to this system are elaborated as follows:

a) User Account Management

- ***Sign Up***:
  - *Description:* Users will be able to create accounts prior to using the app.
  - *Acceptance criteria:* Users being able to successfully create an account and have personalized interface

- ***Login:***
  - *Description:* Users will be able to access their account from any mobile device having the app installed
  - *Acceptance criteria:* Users being able to successfully login to an account they have created

b) Road State Updates

- ***Weather Related Hazards***
  - *Description:* Alerts users about weather conditions affecting road safety.
  - *Acceptance Criteria:* Notifications are sent to users based on real-time weather updates, indicating hazardous conditions.

- ***Real-Time Traffic Updates***
  - *Description:* Provides users with current traffic information.
  - *Acceptance Criteria:* Traffic updates are accurate and refreshed frequently, reflecting real-time conditions.

- ***Road State Update Permission***
  - *Description*: Allows users to report road conditions.
  - *Acceptance Criteria:* Users can submit updates on road conditions, and the information is verified and reflected in the app.

- ***Route Redirection***
  - *Description:* Suggests alternative routes to avoid traffic or hazards.
  - *Acceptance Criteria:* App recalculates routes dynamically based on real-time traffic and road conditions, offering efficient alternatives.

c) Road Sign Notification

- ***Road Sign Notification***
  - *Description:* Notifies users about upcoming road signs and their meanings.
  - *Acceptance Criteria:* Users receive timely notifications before approaching road signs, accompanied by clear explanations.

- ***Real Time Speed Counter***
  - *Description*: Displays the user's current speed in real-time and notifies in case of exceeding speeds.
  - *Acceptance Criteria:* Speed counter accurately reflects the user's speed and updates continuously.

d) User Experience

- ***Notification Preferences***
  - *Description:* Allows users to customize their notification settings.
  - *Acceptance Criteria:* Users can choose which types of notifications they want to receive and adjust their preferences accordingly.

- ***Voice Support***
  - *Description:* Enables hands-free interaction through voice commands.
  - *Acceptance Criteria:* Voice commands are accurately recognized and executed, providing a seamless user experience.

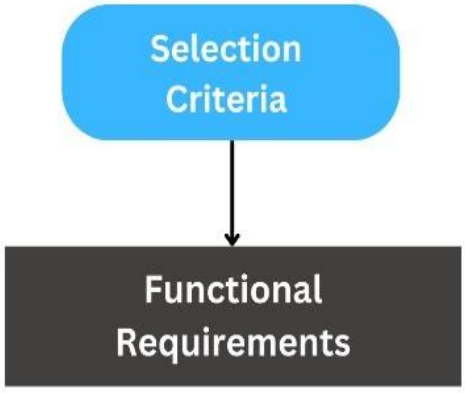
- ***Multilanguage Support***
  - *Description:* Supports English and French languages for national usage.
  - *Acceptance Criteria:* App interface and notifications are available in English and French languages, and users can easily switch between them.

e) Location Management

- ***Location Tracking***
  - *Description:* Tracks the user's location in real-time.
  - *Acceptance Criteria*: Location tracking is accurate and updates continuously, providing precise positioning information.

- ***Share Location from the App***
  - *Description:* Allows users to share their current location with others.
  - *Acceptance Criteria:* Users can share their location directly from the app through various communication channels.

- ***Defined Frequently Accessed Locations***
  - *Description:* Stores frequently visited locations for quick access.
  - *Acceptance Criteria:* Users can save and access commonly used locations easily, reducing the need for manual input.



IV.2 Non-Functional Requirements

Non-functional requirements define the qualities or attributes that characterize the operation and performance of a system rather than its specific behaviors. They typically address aspects such as reliability, usability, performance, security, and scalability. The non-functional requirements to this system are as follows:

- **Performance:** Optimizing performance within our system is crucial for delivering a seamless user experience and maximizing operational efficiency. A driver blocked in traffic and wishing to change route should have the other possible routes as fast as possible, also the driver should be able to receive any notifications instantaneously or near-instantaneously without any noticeable delay.

- **Reliability:** Ensuring reliability within our system is fundamental to fostering trust and maintaining uninterrupted operations. The information given to drivers about road states and road signs should be accurate and constantly available, ensuring the user's safety and nurturing user's trust in the app.

- **Security:** In today's interconnected digital landscape, ensuring the security of our system is crucial. Considering a situation where a driver wants to change route due to an accident or any other problem, his location needs to be protected to avoid non-ethical hackers from having access to it and use it to harm the driver. Additionally, when a user plans his trajectory giving his destination.

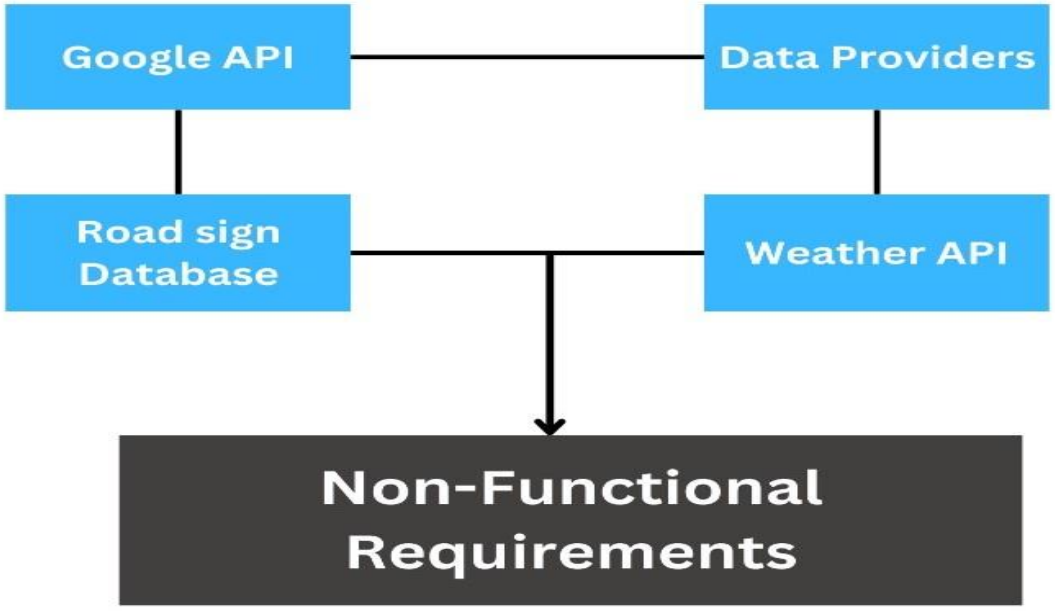
- **Usability:** Usability enhances user satisfaction and facilitates seamless interactions. A driver desiring to share his location with another person should not find it difficult to identify the component responsible for that on the interface. A good user interface is user centric meaning it focuses on facilitating the app's navigation promoting positive user interactions and user adoption

- **Scalability:** Ensuring scalability within our system is essential for accommodating growth and meeting evolving demands. As time progress, the app's data will continue to expand. The app should be able to handle this increase of data without affecting the app's performance or reliability.

- **Availability:** The constant and uninterrupted availability of road states and road signs provide users with a reliable and resilient platform that is accessible whenever they need it, regardless of external factors fostering user satisfaction and trust towards the app.

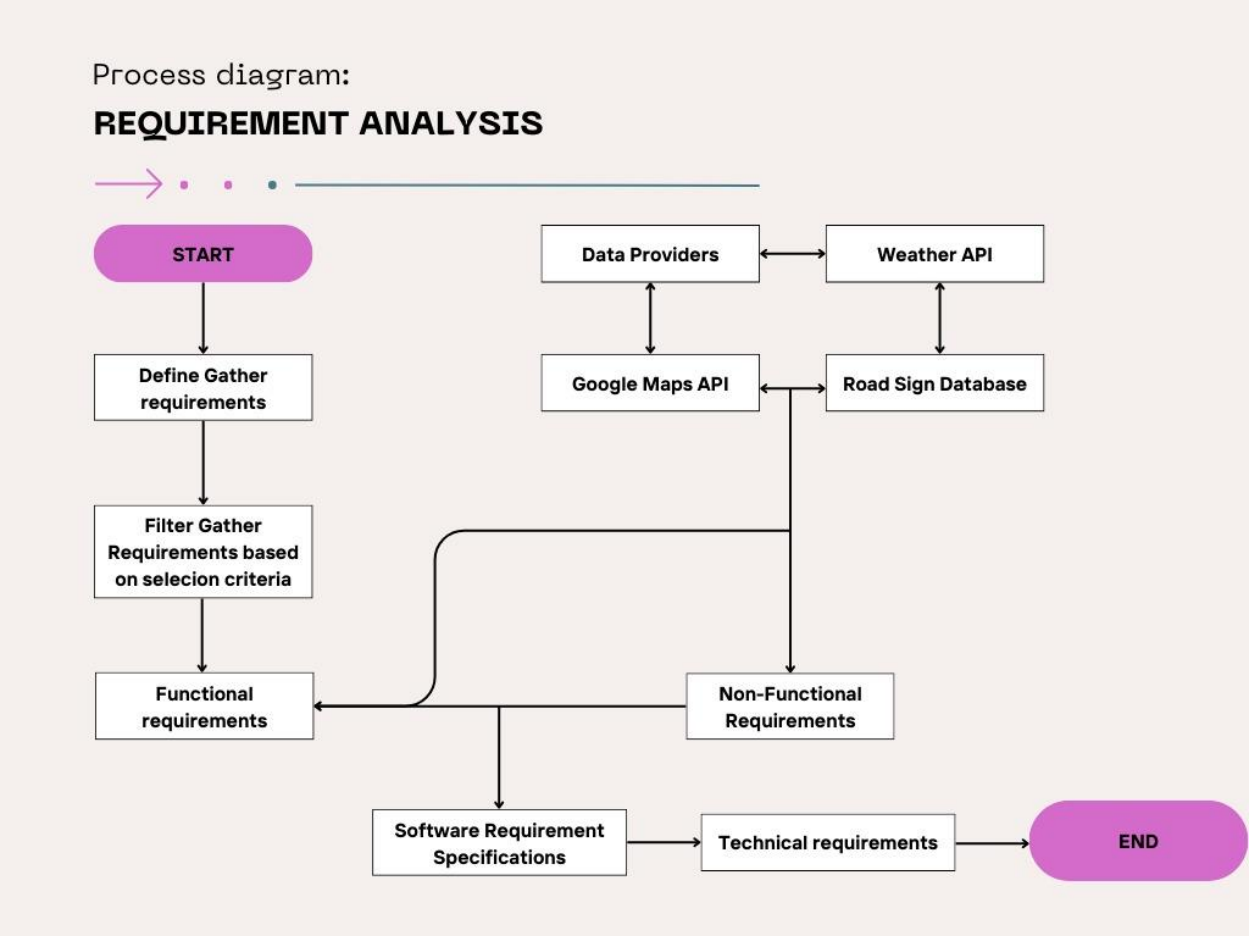
- **Maintainability:** By prioritizing maintainability, we optimize the process of updating, diagnosing issues and administering our system, fostering resilience and operational efficiency. The app should provide necessary updates, easy to adapt to and not too heavy (that is the amount of internet connection needed to update the app should be small).

- **Compatibility:** Different operating systems exists and more and being developed. Developing an application which integrate with all these operating systems and all different screen sizes that exist maximizes user research and usability.



V- CONCEPTUAL DESIGN

This section demonstrates the whole requirement analysis process. After gathering the requirements, they are filtered using a selection criterion. The filtered requirements are classified as functional requirements, the non-functional requirements are gotten from the interaction of external actors. The software requirement specification is the combination of the functional and non-functional requirements and finally the technical requirements are gotten from the software specification document.



VI- TECHNICAL REQUIREMENTS

While requirement gathering is crucial, they remain inert without selecting a technology to implement them. An app can be divided into three parts as **frontend, backend** and **database**.

VI.1 Frontend

This part takes care of the visual part of the application (the user interface). **React native** which is a mobile app development framework will be used here based on the following reasons;

- Provides cross-platform compatibility by enabling applications to run both on Android and IOS operating system.
- Easily integrate APIs in the system such as weather API.
- Gives access to the phone's native functionalities such as accelerometer and GPS.
- Provides access to tools for creating user-centric interfaces.

Additionally, react native has a very good performance, has an easy to learn programming language (**JavaScript),** is cost effective, rapid to develop and has a large community, increasing the reasons for our choice.

VI.2 Backend

This part concerns the logic of the application. It operates behind the scenes, processing and managing data in the application. **Nodejs** together with **Expressjs** and **Firebase** will be used here for the following reasons;

- Firebase for login and signup
- Expressjs provide a clean architecture to enhance the scalability, performance, reliability, availability and maintainability of the application.

Node.js is an open-source, server-side JavaScript runtime engine. Express.js simplifies the process of building applications and APIs allowing developers to create powerful and scalable server-side applications with ease. Firebase will be used for authentication, by giving access to already developed solutions, facilitating the user's access to the system.

VI.3 Database

This part is responsible for data storage in a structured and organized manner. The database management system **MongoDB** as well as **Fire-store** will be used here. MongoDB is a NoSQL database known for its flexibility and scalability, particularly well suited for applications with rapidly changing data. A NoSQL database is a type of database management system that provides a mechanism for storing and retrieving data in a format other than traditional tabular relations used in relational databases. **Firebase** can also be used here since it provides real-time database and a cloud fire store, which can be used as a database management system (DBMS) for storing and syncing data in real-time.

CONCLUSION

Following the comprehensive analysis of the system requirements, the foundation has been laid for informed decision making and efficient system design. With a clear understanding of user needs and functionality specifications, the subsequent phases of development are ready to proceed with clarity and purpose, ensuring the delivery of a customized and effective solution.

REFERENCES

[1] https://www.geeksforgeeks.org/activities-involved-in-software-requirement-analysis/
(Visited on the 05/06/2024)

[2] https://www.mit.edu/

(Visited on the 05/13/2024)