

THE UNIVERSITY OF BUEA

P.O.BOX 63,

Buea, South West Region Cameroon

CEF 440

Internet Programming and Mobile Programming

Department: Computer Engineering



REPUBLIC OF CAMEROON

Faculty Of Engineering and Technology

TASK 2

Submitted to:

Dr. Nkemeni Valery

By:

Group 14

2023/2024

MEMBERS

LANGEH MOHAMMED YIENEH	FE21A223
JENNA EBOT AGBOR	FE21A205
ABO STEVE AKUM	FE21A125
NDEH TAMINANG	FE21A246
CHINEPOH DIVINE-FAVOUR	FE21A159

Contents

- 1. Introduction 1**
- 2. Stages of the Requirement Gathering Phase 1**
 - 2.1. Assigning roles..... 1
 - 2.2. Define Project Scope..... 2
 - 2.3. Conduct Interviews:..... 2
 - 2.4. Document Requirements 4
 - 2.4.1. Functional Requirements..... 4
 - 2.4.2. Non-Functional Requirements 5
 - 2.5. Verify and Validate Requirements 6
 - 2.6. Prioritize Requirements 6
- 3. Conclusion 7**
- 4. References 8**

1. Introduction

In today's fast-paced world, road traffic has become a significant concern, with increasing numbers of vehicles on the roads leading to congestion, accidents, and delays. Additionally, drivers often face challenges in keeping track of road signs and being aware of real-time road conditions. This calls for innovative solutions that can provide drivers with timely and accurate information to ensure a safer and more efficient journey.

Requirement Gathering

In the world of software development, the success of a project relies heavily on a crucial yet often overlooked phase: Requirement Gathering. This initial stage acts as the foundation for the entire development life cycle, steering the course of the software and ultimately determining its success.

Requirements gathering is a crucial phase in the software development life cycle (SDLC) and project management. It involves collecting, documenting, and managing the requirements that define the features and functionalities of a system or application. The success of a project often depends on the accuracy and completeness of the gathered requirements in software.

2. Stages of the Requirement Gathering Phase

2.1. Assigning roles

The first step is to identify and engage with all relevant stakeholders. Stakeholders can include end-users, clients, project managers, subject matter experts, and anyone else who has a vested interest in the software project. Understanding their perspectives is essential for capturing diverse requirements.

Stakeholders:

- Internal: Abo Steve Akum, Chinepoh Divine-Favour, Jenna Ebot Agbor, Langeh Mohammed Yieneh, and Ndeh Taminang
- External: Supervisor (Dr Nkemeni Vallery), and Target users

Users: Drivers

Project manager: Jenna Ebot Agbor

2.2. Define Project Scope

Clearly defines the scope of the project by outlining its objectives, boundaries, and limitations. This step helps in establishing a common understanding of what the software is expected to achieve and what functionalities it should include.

Scope: “The primary objective of this project is to develop a mobile application that leverages the power of modern technology, including smartphones and real-time data, to address the aforementioned challenges. The application aims to deliver road sign information and road state notifications to drivers in a convenient and user-friendly manner.”

2.3. Conduct Interviews:

Scheduled interviews with key stakeholders and target users to gather information about their needs, preferences, and expectations. Through open-ended questions and discussions, we aimed to uncover both explicit and implicit requirements. These interviews provided valuable insights that contributed to a more holistic understanding of the project.

Questions:

- What do you think would reduce the rate of accidents?
- How frequently do you encounter road signs while driving?
- On a scale of 1 to 5, how important do you consider being informed about upcoming road signs while driving?
- Have you ever missed or misunderstood a road sign while driving?
- How confident are you in your ability to notice and understand road signs while driving?
- Would you find a system that notifies you about upcoming road signs useful?
- What features would you like to see in a system that notifies you about road signs? (e.g., distance to sign, type of sign, voice alerts)
- How would you prefer to receive notifications about upcoming road signs? (e.g., visual alerts on dashboard, auditory alerts, smartphone notifications)
- Would you be willing to pay for a system that provides advanced notification of road signs?

- How likely are you to recommend a system that notifies about road signs to friends or family members?
- How do you think a system like this could improve your driving experience or safety?
- Any additional comments or suggestions about a system that notifies about road signs?

User Response:

Upon talking to the target users of this application that is the road users, here are some the information gathered:

- I encounter road signs regularly, especially during my daily commute and when traveling on highways.
- I would rate it a 4. It's essential to be aware of upcoming road signs to make informed driving decisions and ensure safety.
- Yes, there have been instances where I missed or misunderstood road signs, especially when driving in unfamiliar areas or during poor weather conditions.
- I'm generally confident, but there are times when road signs can be easily overlooked, especially in busy or distracting environments.
- Absolutely, having a system that provides advanced notification of road signs would be incredibly useful and could enhance my driving experience.
- I would like the system to provide information about the type of sign, distance to the sign, and maybe even some context about why the sign is important. Voice alerts would also be helpful, especially when driving in busy traffic.
- I would prefer visual alerts on the dashboard, as they are less distracting than auditory alerts and can be easily glanced at without taking my eyes off the road.
- Yes, I would be willing to pay for such a system if it proves to be effective in improving my driving experience and safety.
- If the system works well and provides value, I would definitely recommend it to friends and family who drive regularly. As a driver, I want to receive real-time updates about road signs so that I can understand their meaning and significance during my travels.
- I want to receive real-time updates on road conditions such as traffic congestion, accidents, weather-related hazards, and road closures so that I can plan my journey accordingly.

- I want to customize my notification preferences so that I receive alerts that are relevant to my specific preferences and travel plans.
- I want the application to integrate with my preferred navigation app so that I can access all the information in one place.

2.4. Document Requirements

This involves systematically documenting the gathered requirements. This documentation can take various forms, such as user stories, use cases, or formal specifications. Clearly articulated are the functional requirements (what the system should do) and non-functional requirements (qualities the system should have, such as performance or security).

2.4.1. Functional Requirements

1. **GPS Location Tracking:** The application should be able to track the user's location in real-time using GPS.
2. **Road Sign Database:** The application should have a comprehensive database of road signs, especially those on major highways.
3. **Real-Time Data Integration:** The application should integrate real-time data from various sources such as weather sensors, and crowdsourced reports.
4. **Customizable Notifications:** Users should be able to customize their notification preferences based on specific types of road signs, road conditions, or geographical areas.
5. **Authentication and User Management:**
 - Allow users to sign in securely, sign up for new accounts, and enable the option for "Remember Me" functionality.
 - Utilize encryption protocols to ensure the security of user credentials and personal information.

2.4.2. Non-Functional Requirements

1. **User-Friendly Interface:** The application should have a clean, responsive, and intuitive interface optimized for use on smartphones and tablets.
2. **Performance:** The application should provide timely updates on road conditions and should not cause significant battery drain.
3. **Reliability:** The application should provide accurate information and notifications based on the user's location and preferences.
4. **Security:**
 - User data, especially GPS location data, should be securely stored and transmitted.
 - Strong authentication and authorization mechanisms are imperative to prevent unauthorized access
 - Regular security audits and adherence to industry standards are essential for vulnerability mitigation.
5. **Scalability:**
 - The architecture should scale horizontally to accommodate growing traffic and data.
 - Load balancing and scalable database infrastructure ensure optimal performance under varying workloads.
6. **Maintainability:**
 - A well-structured, documented codebase facilitates ongoing maintenance and updates.
 - Automated testing ensures changes don't introduce issues. Streamlined deployment and rollback procedures minimize downtime.
7. **Compatibility:** The application must be compatible with popular mobile platforms like iOS and Android.

8. **Regulatory Compliance:** Compliance with privacy regulations like GDPR, CCPA, or HIPAA is essential. Adherence to industry standards such as ISO 27001 and legal requirements for data retention and breach notifications must be maintained.

2.5. Verify and Validate Requirements

Once the requirements are documented, it's crucial to verify and validate them. Verification ensures that the requirements align with the stakeholders' intentions, while validation ensures that the documented requirements will meet the project's goals. This step often involves feedback loops and discussions with stakeholders to refine and clarify requirements.

2.6. Prioritize Requirements

Given our limited time constraints, and resources, and low budget. Here is the suggested prioritization of the requirements:

High Priority (Must have): These are critical for the basic functionality of the application and should be developed first.

1. **Authentication and User Management:** Security is paramount, and the system needs to protect user data from the get-go.
2. **GPS Location Tracking:** Core feature for a location-based application.
3. **User-Friendly Interface:** Essential for user adoption and usability.
4. **Compatibility:** Ensuring the app works on both iOS and Android platforms will maximize its user base.

Medium Priority (Should have): These features are important but not critical for the initial launch. They can be developed once the high-priority requirements are met.

1. **Real-Time Data Integration:** While important, this can initially be simulated or pared down to a few key data sources due to resource constraints.

2. **Road Sign Database:** Start with a basic set of common road signs and expand over time.
3. **Customizable Notifications:** Basic notifications should be in place first, with customization options added as time allows.

Low Priority (Could have): These are desirable features that could enhance the application but are not necessary for the initial launch. They can be added if time and resources allow.

1. **Performance:** While important, performance optimizations are often something that can be improved iteratively after the main features have been developed.
2. **Scalability:** Start with a basic working model, optimize for scalability as user base grows.
3. **Maintainability:** Good development practices should be followed from the start, but major refactoring can be done after the initial launch if necessary.

Future Considerations (Won't have this time): These are aspects that are recognized as valuable but won't be implemented in this development cycle due to constraints.

1. **Regulatory Compliance:** While important, comprehensive compliance may be beyond the scope of the initial 10-week timeline. Basic data protection measures should be implemented, and full compliance should be a goal for future development cycles.

3. Conclusion

The successful implementation of the mobile application is expected to have several positive outcomes and impacts:

- Improved road safety by providing drivers with real-time information about road signs and hazards.
- Enhanced traffic management through timely notifications about traffic congestion, accidents, and road closures.

- Reduction in travel time and fuel consumption by suggesting alternate routes based on real-time traffic conditions.
- Increased driver satisfaction by providing a more informed and stress-free driving experience.
- Potential for future integration with emerging technologies, such as vehicle-to-infrastructure communication and augmented reality.

In conclusion, these requirements and user stories will guide the design and implementation of the Road Sign and Road State Mobile Notification Application. Further requirements may emerge as the project progresses and will be incorporated into the project plan accordingly.

4. References

- <https://www.geeksforgeeks.org/requirements-gathering-introduction-processes-benefits-and-tools/>
- <https://www.geeksforgeeks.org/designing-google-maps-system-design/>
- <https://www.nuclino.com/articles/functional-requirements>
- <https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/>