Multimedia University of Kenya

Faculty of Computing & Information Technology

Public Service Vehicle Safety Management System

By

BRAMWEL JUMA BARASA

CIT-223-028/2021

Supervisor:MS YVETTE OTUKANA

April 24, 2024

Submitted in partial fulfillment of the requirements of Third Year Bachelor of Science in

Computer Science

0.1 Declaration

I hereby declare that this Project is my own work and has, to the best of my knowledge, not been submitted to any other institution of higher learning.

Student: ______________________

Registration Number: ______________________

Signature: ______________________ **Date:** __________________

This project has been submitted as a partial fulfillment of requirements for the Bachelor of Science in Software Engineering of Multimedia University of Kenya with my approval as the University supervisor.

Supervisor: ______________________

Signature: ______________________ **Date:** __________________

0.2 Dedication

My dearest parents,Mr Nalwa Enos and Mrs Wakoli Florence, Throughout my journey, your unwavering love, support, and encouragement have been the guiding stars lighting my path. Every milestone achieved and every success attained is a testament to your boundless sacrifices and endless belief in me.

Mom and Dad, your selflessness knows no bounds. You've instilled in me the values of perseverance, determination, and resilience, shaping me into the person I am today. Your faith in my abilities has been my greatest source of strength, propelling me forward even in the face of adversity.

As I dedicate this project to you, I want to express my deepest gratitude for your ceaseless devotion and for being my pillars of strength. Your love has been the driving force behind every endeavor, and it is with immense pride and joy that I share this achievement with you.

This project is not just a culmination of my efforts but also a reflection of your endless love and support. Thank you for being my inspiration, my motivation, and my greatest cheerleaders. I am forever grateful for the love and guidance you've bestowed upon me.

With all my love and appreciation.

0.3 Acknowledgements

I am grateful to Multimedia University of Kenya for giving me a chance to pursue Bachelor of Science in Computer Science. I wish to thank my classmates and the CIT staff for having contributed in formulation of ideas and for providing a suitable working environment towards the completion of this proposed work. In particular, I would like to thank Ms Otukana Yvette for having contributed much towards the formulation of this project idea and the guidance till end of the project.

0.4 Abstract

- The main aim of this project is to create a system that keeps records of public service vehicle (PSV) realtime statistics such as speed and location ,insurance policy and all the safety features of the vehicle which are used in determining the safety level of a particular PSV.
- The main users of the system are the passengers and the law enforcement. Other key players in the system are drivers, conductors,vehicle owners, saccos and NTSA. Passengers enter the vehicles registration number before boarding to check the safety level of the vehicle and they can make mindful decisions based on the data that the system provides.
- Drivers, conductors and vehicle owners provide details that are linked to the repective vehicle(s)while the NTSA provides inspection reports of the vehicles. The system uses gps devices to get realtime location and speed of vehicles which is recorded and analysed and safety levels of a vehicle are determined.
- The system allows police officers to view the vehicles safety level and inspection report by entering the vehicles registration number and take apropriate actions.
- This system is built using Reactjs java and springboot.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background of study

Public Transport plays a crucial role in the daily life of Kenyan citizens. The country relies majorly on public service vehicles (PSVs) to meet the commuter needs of diverse groups of people. This has led to an increase in passengers which in turn led to the necessity of increased motorization. Public transport in the country is driven by privately-owned vehicles, public service vehicles (PSV) popularly known as matatu, operating in the country following licensing through various licensing bodies. At present, a PSV is any vehicle that is licensed to ferry the public on Kenyan roads. This includes buses, mini-buses, vans, and mini-vans, 3-wheel motorcycles, motorcycles, taxis among others. The most popular category of PSV vehicles is the vans and mini-buses.

Traditionally, Monitoring of safety of the PSVs is done by a serioes of manned rodblocks on the roads and occassional inspection by the NTSA. This system has been faced with loopholes which range from bribes and direct evation of roadblocks. This hs led to reckless driving and unroadworthy vehicles on the roads resulting to increased road accidents which have claimed lives of many passengers

1.2 Problem Statement

Road regulation enforcement is a difficult task considering the vast number of motorists contrasted with the low numbers of law enforcement officerscers. As such, a great deal of cooperation between road users, policymakers, road designers, and law enforcement is needed. The current approach to road traffic and safety places all responsibility on the road user as opposed to an integrated view as witnessed through initiatives like Zusha

Police crackdowns on non-conforming drivers is a solution that causes more inconvenience to passengers than they do to those who disobey traffc laws. As noted PSV drivers frequently allege that they are subjected to abuse and/or harassment by the police for no other reason than to demand a bribe. Static checkpoint systems, the nation's primary monitoring technique, are another flaw in police. The drivers alert other drivers to impending checks by communicating with them or by previously being aware of the monitored zones. In this sense, it is challenging to measure irresponsible driving outside of the monitored zone.

Apart from insecurity, the majority of PSVs have placed very little emphasis on the comfort of their passengers, with behaviors ranging from vagrant insults to cramming people into small spaces without following rules. This hostility and unwillingness of the PSV crew to listen to passenger demands has led to an impediment of campaigns such as the zusha campaign.

1.3 Aim of Study

The aim of this study is to develop an online system that monitors realtime data from PSVs and determing the safety levels of the vehicle with a goal to increase the safety of the most vulnerable i.e the passengers and reduce road traffic accidents.

1.3.1 Research Objectives

1. To design an online system that moniters public service vehicles.
2. To Create an avenue for determining vehicle safety by passengers and the law enforcement
3. Provide a platform that allows passengers to protest against recklessly driven vehicles.
4. To develop a database that stores vehicle's safety details and drivinh history

1.4 Significance of the Study

Passenger safety and overall motorist safety has been a major challenge as evidenced by a large number RTAs .This study aims at providing a system that monitors the PSVs and provides information about specifc PSV to passengers.

Furthermore, this study aims at providing law inforcment with information of recklessly driven vehicles outside of the monitored zones.

The study will be crucial in creating a connection between passengers the PSV crew and the law enforcement.

Ultimately, the signifficance of this study lies in its potential to catalyze positive change improving enforcement strategies, and promoting a culture of responsible road behavior, the study has the potential to save lives, reduce injuries, and contribute to the overall well-being and prosperity of the Kenyan population.

1.5 scope

Every project is done to achieve a set of goals with some conditions keeping in mind that it should be easy to use, feasible, and user-friendly. As the goal of this project is to develop an online PSV monitoring system ,this system will be designed keeping in mind the conditions (easy to use, feasible, and user-friendly) stated above.This is made possible through the use of a GeoLogger GPS data logger which records time-stamped records for latitude and longitude, the vehicle's speed, and the vehicle's heading. The GeoLogger has several features that made it desirable for this study. The device is powered by the vehicle's DC power system via the cigarette lighter or power outlet so that it operates only when the vehicle is turned on. It will help in determining the safety level of the PSVs and providing the information to the passengers and more detailed information to the law enforcement. This project aims to monitor PSVs beyond the monitered zones that are under speed-guns and roadblocks

The main scope and deliverable of the project would be to:

- Understand and prepare detailed requirement and specifications
- Prepare high level and detailed design specifcations for the system
- Prepare Test Plan and test cases
- Devlop the System and coding.
- Perfom unit Test

1.6 Assumptions

- **Acces:**It is assumed that passengers,drivers,conductors and law enforcement own a smartphone that will be used to access the system
- **Data Collection:** : It is assumed every PSV has the capacity to be fitted with gps trackers
- **Connectivity:** : The study assumes that users of the system have internet connection.
- **Enforcement effectiveness:** : It is assumed that improving road monitoring techniques, the system will lead to a reduction in RTAs and improved road safety outcomes.

1.7 Limitations and Countermeasures

The study is subject to several limitations and challenges that may affect the reliability and generalizability of its findings. One such limitation pertains to the reliance on existing data sources, such as government reports and police records, which may suffer from inaccuracies, underreporting, or inconsistencies. To mitigate this limitation, the study will use APIs provided by the NTSA website mantain consistency in data being fed to the system

Another challenge involves the potential for drivers and PSV crew to switch off or disconnect the gps trackers from the vehicles. To mitigate this, the systems employs the use of an OBD logger which provides separate data files for each key-on to key-off vehicle operating period. It also provides the start time for each operating period and the elapsed time between each record in an operating period.This data will be frequently monitered to maximize uptime and minimize downtime

Chapter 2

Literature Review

2.1 Background

Speeding and reckless driving pose significant threats to road safety, leading to a concerning number of accidents, injuries, and fatalities worldwide. In Kenya, like many other countries, public service vehicles (PSVs) are often at the forefront of these road safety challenges. The operation of PSVs in Kenya is a critical component of the country's transportation system, providing essential services to millions of commuters daily. However, the prevalence of speeding and reckless driving among PSV operators has contributed to a worrying trend of road traffic accidents, endangering the lives of passengers, pedestrians, and other road users. This literature review aims to explore the various systems implemented to curb accidents caused by speeding and reckless driving in PSVs operating in Kenya. By examining existing research, policies, and interventions, this review seeks to identify effective strategies and best practices which will be implemented by the system for improving commuter safety in the Kenyan context

2.2 Related Systems

Various sustems have been developed to streamline services in the road transport sector and morever the public transport sector. These systems are Traditional police crackdown through manned roadblocks across the country,use of speedguns ,frequent vehicle inspection by the NTSA and the fleet management systems that are used to manage Vehicles owned by a particular systems.

Fleet management systems hepls in monitoring fuel consumption, maintenance schedules, and vehicle position and condition all contribute to cost control, equipment longevity, and a decrease in fatalities. It aids businesses in ensuring compliance, enhancing productivity, and lowering accidents. Applications for fleet management provide drivers more control over their cars.Fleet management services are less expensive than post-accident management organizations. For further advantages, we advise utilizing fleet management applications to measure all accident-related effects.However this systems focus majorly on the businss streamline without focusing on the safety of commuters

Police crackdown through manned roadblocks across the country was meant to provide an avenue for the law enforcemnt to track down reckless drivers and unroadworthy vehicles . However this method has been filled with countless flaws from harrasment by the police officers to outright bribery which has led to the ineffeciency of this system. Moreover the matatu crew alert each other of a roadbloack or about a speed gun zone which leads to the difficulty in tracking down reckless driving out of monitered zones.

Introduction of campaigns such as the Zusha campaign has signifficantly hepled in reucing reckless driving in PSVs.This campaign give the commuters the voice to protest against recklessly driven vehicles. It is promoted by the use of mandatory stickers on the PSVs

2.3 Limitations of these systems

This systems have greatly improved the safety of commuters and reduction of road accidents greatly.However, they are greatly flawed as follows:

- Motorist familiarity with monitored zones and roadblock zones which hinders the monitering of reckless driving in unmonitored zones.
- Police bribery and harrasment of matatu crew due to lack of a defined system that clearly shows reckless driven
- Campaigns such as zusha have been rendered ineffecient due to hostility of the matatu crews who fail to loisten to commuter protest when the vehickes are being recklessly driven
- Ineffecient storage of vehicle inspection reports which has led to an increase in the number of unroadworthy public service vehicles

2.4 Proposed solution

The public service vehicle safety management system provides a solution to the shortcomings of the traditional system by:

- The system collects realtime data such as speed, heading, longitude and lattitudes and timestamps them for anaysis.This is made possible by the use of GPS trackers and Geologger.
- The system keeps track of vehicle safety inspections reports, Vehicle crew details , insurance details, vehicle details and a vehicles calculated and analysed speeding history which are used to detrmine the safety level of the vehicle.
- This system allow commuters to view the details , vehicle data and speeding history of the vehicle to determine the safety level of the system and make mindful decisions.
-

Chapter 3

Methodology

3.1 Introduction

The Systems Development Life Cycle (SDLC) is a framework for describing the phases involved in developing and maintaining information systems.

Predictive life cycle in which the scope of the project can clearly be articulated and the schedule and cost can be predicted; and **Adaptive Software Development** (ASD) life cycle in which requirements cannot be clearly expressed, projects are mission driven and component based, using time-based cycles to meet target dates

We are going to use an ASD life cycle model called Agile System Development Methodology for developing PSVSM system.

3.2 Agile System Development Methodology

Most agile methods attempt to minimize risk by developing software in short time boxes called Iterations. Software development being essentially a human activity will always have variations in processes and inputs and the model should be flexible enough to handle the variations [?]. The following are the characteristics of agile model that makes it suitable for PSVSM system development

- Iterative with short cycles enabling fast verifications and corrections.
- Time bound iterative cycles.
- Modularity at development process level.
- People oriented.
- Collaborative and communicative working style.
- Incremental and convergent approach that minimizes risks and facilitates functional additionals

3.3 Overview

The scrum method is incremental, with each increment called a sprint, each sprint is recommended to last for 4 weeks. Before the sprint there is a planning meeting for each sprint, where a customer decides which features should be implemented in the upcoming sprint. During the sprints the teams meets on a daily basis on short meetings called scrum. A sprint review meeting is held at the end of each sprint, and the customer is able to see the existing accomplishments for the preceding sprint. Teams can also hold sprint retrospective meetings to, where they can look at the process and then try to find out what went right and what can be improved. The figure bellow shows the flow of the methods

/home/bram/Documents/psvm-v1/Scrum_Agile_events.png

Figure 3.1: Agile model

3.4 Scrum artifacts

Scrum being an agile method, follows that the formality of the project is as low as possible, it thus facilitated necessary changes to be made to the project. The customer is also able to see the project progress since this improves their motivation and involvement. Artifacts of scrum includes:

- The product backlog
- The sprint backlog
- The sprint burndown chart.
- The impediment lists

3.4.1 The Product Backlog

It can be considered equivalent to the requirement specifications, but it has one big difference in that it does not entail long description of each requirement, it has only single sentence description for each requirement. Being a list of such single sentence requirements, it is thus the customer's priority to keep it prioritized and updated. The customer can add requirements to the list and the team is then responsible for providing estimates of how long the implementation might take.

3.4.2 The Sprint Backlog

It is a list of tasks maintained and compiled by the team based on he the items in the product backlog, initially selected to be part of the sprint. The list is similar to the product backlog, but with a big difference. The items on the product backlog are features requested by the user, the sprint backlog is a list of tasks the developers must do to implement the items that the customer chose from the product backlog. The customer does not need to know about the items on the sprint backlog. A general rule of thumb is that the tasks on the sprint backlog should always be relatively short, that is between one hour and two days.

3.4.3 The Sprint Burn down Chart

This measures the progress of the sprint instead of the project. It is important because it helps the team discover tasks they did not consider but that must be added to the sprint backlog. Since the chart displays the amount of work remaining and not the amount of work completed, the graph can in fact increase from one day to the next.

3.4.4 Impediment list

An impediment is anything holding back development in some way or another. It is the scrum master's responsibility to deal with any such impediments. The list is simply a set of tasks that the scrum master uses to track the impediments that needs to be solved.

3.4.5 The Sprint Planning Meeting

In the first session, the Customer chooses high priority items from the product backlog that should be completed in the upcoming Sprint. The customer explains the items to the team and they give an estimate on how long it will take to complete it. The sprint backlog is filled so that the sum of the item estimates is about the same as the available work time of the team during the upcoming sprint.

3.4.6 The Daily Activities

During the sprint, the developers work on the items in the sprint backlog. Every day the developers synchronize their progress in a daily Scrum meeting that should last no longer than 15 minutes. During the meeting, all the developers will tell the others what they did since the last Scrum, if there are any impediments obstructing their work and what they are planning on doing until the next Scrum. Another important day-to-day activity is updating the sprint backlog and burn down chart.

3.4.7 Sprint Review Meeting

At the end of the sprint, the team meets with the customer and presents the result of the sprint. The users demonstrate the functionality they have completed and gets feedback from the customer. If the demonstrated functionality is what the customer wanted, then this gives the team a feeling of accomplishment as well as the customer a proof that the project is moving in the right direction. If the demonstrated functionality is not quite what the customer was looking for it is now easy to explain how it is different and what should be done next. In some cases, it is enough to make a few changes while in other cases the implemented functionality must be discarded.

3.4.8 Sprint Retrospect Meeting

The intention of this meeting is to help the team improve their development process. The team, the scrum master and the customer (optional) attend the meeting. During the meeting, the team members take turns saying what went well during the last sprint, and what could be improved. After all team members have had their say, they prioritize the possible improvements and discuss them in order. The meeting should not last more than 3 hours.

3.4.9 Project Startup

Ken Schwaber has had much success with his kick starting of Scrum projects as described in the book Agile Project Management with Scrum (Schwaber, 2002). This process goes as follows. The Scrum Master works with the customer and prepares a backlog. Then the Scrum Master, the Customer and the Team uses one day to go over this backlog. During this first day the customer explains the items in the backlog to the team, and the team estimates how much work it would take to implement this. The customer then prioritizes the items in the backlog and divides the backlog items into sprints. The following day is the first day of the first sprint. This first sprint isn't very different from the following sprints, except that the first part of the sprint planning meeting has already been completed.

3.5 Advantages of Agile

i. Customer satisfaction ii. Allows changes to be made. iii. Deliver a working software frequently, ranging from a few weeks to a few months, considering shorter time-scale. iv. Promotes collaboration v. Provides motivation between individual team members. vi. Allows face-to-face Conversation vii. Measure the Progress as per the Working Software. viii. Maintain Constant Pace ix. Monitoring - Pay regular attention to technical excellence and good design to enhance agility. x. Simplicity - Keep things simple and use simple terms to measure the work that is not completed. xi. Review the Work Regularly

3.6 Disadvantages of Agile

i. Limited support for distributed development environments. ii. Limited support for subcontracting. iii. Limited support for developing large, complex software. iv. Limited support for development involving large teams.

Chapter 4

System Analysis

4.1 Introduction

Upon the completion of the Public service Vehicle Safety Managenemt System, there are a number of things that will be expected of it not only by the prospected users, but also for the administrator of the system. These will therefore form the requirements of the Public service Vehicle Safety Managenemt System and will be broadly classified in to the system requirements, functional requirements and the Non-functional requirements.

4.2 System Requirements

4.2.1 Software Requirements Specification

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices.

The minimum software requirement specifications for developing this project are as follows:

- **Operating system:** windows,linux
- **Presentation:**Reactjs, Axios, bootstrap ,apache tomcat, google maps API,browser, java 17
- **Database:** mariadb
- **documentation:** LaTex

4.2.2 Hardware Requirements Specification

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware. The minimum hardware requirement specifications for developing this project are as follows:

- **Processor:** Standard processor with a speed of 1.6 GHz or more
- **RAM:** 256 MB RAM or more

- **Hard Disk:** 20 GB or more
- **Monitor:** Standard color monitor
- **Keyboard:** Standard keyboard
- **Mouse:** Standard mouse

4.3 Functional Requirements

1. Security; to ensure confidentiality, integrity and availability
2. Administrative functions
3. Authentication
4. Authorization levels
5. External interfaces
6. Historical data
7. Legal and regulatory requirement
8. Transaction corrections, adjustments and cancellations

4.4 Non Functional Requirements

4.4.1 Reliability Requirements

The system must perform request accurately for reliability. For example, when the vehicle owner saves the edited details of the vehicle , after he reviews the details later, they must be changed according to the latest details that was updated. Moreover, the client is not allowed to view the details that the administrator has. Besides that, the login form will have validity check to ensure that only the authorized users gain access to the system.

4.4.2 Usability Requirements

This system should be user-friendly and easy to use so that users can perform their tasks easily.

4.4.3 Implementation Requirements

In the implementation of this system, we use React Js which is the frontend.

The backend which hold the bussiness logic of the application is built using Java with springboot framework. Data from the bussiness logic is stored in Mariadb.The frontend and backend are linked using Axios.

4.5 Technologies Used

4.5.1 MariaDB

We started out with the intention of using the Mariadb database system to connect to our tables using our own fast low-level routines. This resulted in a new SQL interface to our database but with almost the same API interface as Mariadb. The following list describes some of the important characteristics of the mariadb Database Software that made us prefer it for PSVSM system to other databases.

features

1. **High Performance**: MariaDB often exhibits better performance than its counterparts due to its advanced optimization techniques and storage engines like Aria and InnoDB.
2. **Open Source**: Being open source, MariaDB is free to use and has a large community contributing to its development and support.
3. **Compatibility**: MariaDB is designed as a drop-in replacement for MySQL, making it easy to switch from MySQL to MariaDB without significant changes to applications or databases.
4. **Advanced Features**: MariaDB offers many advanced features like JSON support, temporal data tables, and dynamic columns, enhancing its functionality for modern database needs.
5. **Security Enhancements**: MariaDB provides enhanced security features such as roles-based access control, encryption at rest and in transit, and plugin authentication for improved data protection.

4.5.2 Springboot

Springboot is the main framework that I used to create the backend. It provides out of the box features that make development of APIs faster. Here are some features that made it a prefered solution for this project:

1. **Rapid Development**: Spring Boot provides a set of pre-configured components and conventions that allow developers to quickly set up and start developing applications without dealing with boilerplate configuration.
2. **Microservices Architecture Support**: Spring Boot is well-suited for building microservices-based architectures, providing features like embedded servers, dependency management, and configuration.
3. **Auto-configuration**: Spring Boot automatically configures many components based on classpath dependencies, reducing the need for manual configuration.
4. **Embedded Servers**: Spring Boot allows embedding servlet containers like Tomcat, Jetty, or Undertow directly within the application, simplifying deployment and reducing overhead.
5. **Production-Ready**: Spring Boot offers production-ready features such as health checks, metrics, and externalized configuration, facilitating the deployment and management of applications in production environments.

howevermuch springboot maybe good for development, it has its own challenges which are listed below

1. **Steep Learning Curve**: Spring Boot's extensive features and configurations can lead to a steep learning curve for beginners.
2. **Complexity**: While Spring Boot simplifies the setup of Spring applications, it can still introduce complexity, especially in larger projects where understanding the underlying configurations becomes necessary.
3. **Overhead**: Spring Boot's auto-configuration feature may lead to unnecessary overhead, especially in smaller projects where simpler frameworks could suffice.
4. **Magic Behind Auto-Configuration**: Although Spring Boot's auto-configuration is convenient, it might be challenging to debug issues or understand the underlying configuration decisions made by the framework.
5. **Limited Flexibility**: While Spring Boot offers a wide range of features, its opinionated approach might limit flexibility for developers who prefer more control over their application's configuration.

4.5.3 React JS

React.js is a popular JavaScript library for building user interfaces, developed by Facebook. It is renowned for its component-based architecture, which allows developers to build reusable UI components and compose complex user interfaces efficiently. React.js is a powerful library for building interactive and dynamic user interfaces, offering benefits such as a component-based architecture, virtual DOM for efficient rendering, declarative syntax for clarity, and a rich ecosystem with extensive community support.Some of the Advantaeges of React Js are listed below:

1. **Component-based Architecture**: React.js follows a component-based architecture, which allows for the creation of reusable UI components. This is advantageous in a project like yours where there might be various modules and elements that need to be reused across different parts of the system, such as displaying vehicle information, safety statistics, etc.
2. **Virtual DOM**: React.js utilizes a virtual DOM, which efficiently updates only the necessary parts of the actual DOM when data changes. In a real-time system like yours where vehicle data is constantly changing, this can lead to improved performance and responsiveness.
3. **Declarative Syntax**: React.js uses a declarative syntax, making it easier to understand and maintain code. This is beneficial for developers working on the project, as it reduces cognitive load and helps in quickly identifying and fixing issues.
4. **Large Ecosystem and Community Support**: React.js has a large ecosystem of libraries, tools, and resources available, along with a vast community of developers. This provides access to a wide range of pre-built components, middleware, and solutions that can expedite development and enhance the functionality of your system.

Chapter 5

System Design

5.1 Architectural Design

PSVSM system uses three-tier architecture. Three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and well-established software architecture

/home/bram/Documents/psvm-v1/architecture-diagram.drawi

Figure 5.1: PSVSM client serve architecture

5.2 logic design

5.2.1 Behavioural Description

Dataflow

Context Level DFD

In the Context Level, the whole system is shown as a single process. Figure below illustrates PSVSM context level DFD showing Inputs to the overall system and outputs from the overall system shown together with their destinations.

Top-Level DFD

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. Figure below shows PSVSM system Top level

/home/bram/Documents/psvm-v1/psvsms-context-diagram.drawio.png

Figure 5.2: Context Level DFD

DFD illustrating what the system does in more detail.

5.2.2 UseCase documentation

The use case diagram show the relationship between actors and entities in the system. The diagram below shows the general use case of users, drivers, conductors and vehicle owners

vehicle owner usecase diagram

Commuters Usecase diagram

/home/bram/Documents/psvm-v1/Top-level-DFD.drawio.png

Figure 5.3: Top Level DFD

/home/bram/Documents/psvm-v1/UserUsecase.drawio.png

Figure 5.4: General Usecase diagram

/home/bram/Documents/psvm-v1/owners.usecase.png

Figure 5.5: Owners Usecase diagram

/home/bram/Documents/psvm-v1/users-commuter-usecase.drawio.png

Figure 5.6: Commuters Usecase diagram

Inspectors use case diagram

/home/bram/Documents/psvm-v1/users-commuter-usecase.drawio.png

Figure 5.7: Inspector Usecase diagram

5.2.3 Class diagram Documentation

Creating documentation for a class diagram involves explaining the purpose of each class, its attributes, methods, relationships with other classes, and any additional information necessary for understanding the design. Here's a comprehensive overview of the PSVSMs class diagram:

Class diagram

`/home/bram/Documents/psvm-v1/psvms-system-class.drawio.png`

Figure 5.8: System class diagram

5.2.4 Process flow

Activity Diagrams

Activity diagrams show the flow of activities in the sytem.The following diagrams show user activity in the system.

Vehicle Owners actvity diagram

```
/home/bram/Documents/psvm-v1/Owner-activity-diagram.drawio.png
```

Figure 5.9: vehicle Owner activity diagram

Commuters activity diagram

/home/bram/Documents/psvm-v1/Commuters-activity.drawio.png

Figure 5.10: commuters activity diagram

Inspector activity diagram

`/home/bram/Documents/psvm-v1/Inspector-activity.drawio.png`

Figure 5.11: Inspector activity diagram

crew activity diagram

/home/bram/Documents/psvm-v1/crew-activity.drawio.png

Figure 5.12: crew activity diagram

Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis

/home/bram/Documents/psvm-v1/user-sequence.drawio.png

Figure 5.13: user sequence diagram

5.3 Database Design

Entity-Relationship Diagram (ERD) is a type of database design tool used to visually represent the logical structure of a database. An ERD consists of entities, attributes, and relationships. Entities represent real-world objects or concepts that are stored in the database. Attributes are the properties or characteristics of entities . Relationships describe how entities are related to each other.

Entity Relationship Diagram

`/home/bram/Documents/psvm-v1/psvsm-erd.drawio.png`

Figure 5.14: entity relationship diagram

5.4 User Interface Design

User Interface (UI) design is the process of creating interfaces in software or computerized devices with a focus on looks or style. It encompasses everything users interact with on a screen, including images, sliders, buttons, text, input fields, and all interactive elements. Here's a comprehensive overview of the UI design process

home UI design

`/home/bram/Documents/psvm-v1/UI design.png`

Figure 5.15: User Interface Design

Chapter 6

Implementation and Testing

6.1 Development environment

6.1.1 Technologies Used

- **Java**: Java is used for backend development due to its robustness, platform independence, and extensive libraries and frameworks support.
- **Spring Boot**: Spring Boot is chosen as the backend framework for its rapid development capabilities, dependency management, and support for building microservices.
- **React.js**: React.js is employed for frontend development to create dynamic and interactive user interfaces with its component-based architecture and virtual DOM.
- **Axios**: Axios is used as the HTTP client for making asynchronous HTTP requests from the frontend to the backend API implemented in Spring Boot.
- **MariaDB**: MariaDB is selected as the relational database management system for its compatibility with MySQL, performance optimizations, and open-source nature.

6.1.2 Backend Development

- **IDE**: IntelliJ IDEA is used as the Integrated Development Environment (IDE) for Java development, providing features such as code completion, refactoring, and debugging.
- **Build Tool**: Apache Maven is utilized as the build automation tool for managing dependencies, compiling code, and packaging the application.
- **Spring Boot Starter**: The project is initialized using Spring Initializr to bootstrap a Spring Boot application with necessary dependencies like Spring Web, Spring Data JPA, and MariaDB Driver.
- **Database Setup**: MariaDB is installed and configured locally for development purposes. Database schema and tables are created using SQL scripts or Hibernate ORM entities.

- **Backend Server**: Spring Boot's embedded Tomcat server is used for running the backend application locally during development.

6.1.3 Frontend Development

- **Node.js and npm**: Node.js and npm (Node Package Manager) are installed to manage frontend dependencies and run development scripts.
- **Create React App**: Create React App is used to scaffold a new React.js project with a predefined directory structure, webpack configuration, and development server setup.
- **React Components**: UI components are developed using React.js, following best practices such as functional components, hooks, and component composition.
- **State Management**: React's built-in state management capabilities are utilized for managing component state, and if needed, Redux or Context API can be integrated for global state management.
- **Styling**: CSS modules, SASS, or styled-components are employed for styling React components, providing modularity and scoped styles.
- **Development Server**: The frontend application is served locally using the webpack development server, enabling hot reloading and rapid iteration during development.

6.1.4 Collaboration and Version Control

- **Version Control**: Git is used for version control, allowing collaboration among team members, tracking changes, and managing codebase versions.
- **Git Hosting Platform**: GitHub, GitLab, or Bitbucket are utilized as the hosting platforms for the Git repositories, providing features like pull requests, code reviews, and issue tracking.
- **Agile Methodology**: Agile development methodologies such as Scrum or Kanban are adopted for project management, enabling iterative development, frequent releases, and responsiveness to changing requirements.
- **Communication Tools**: Communication tools like Slack, Microsoft Teams, or Discord are used for real-time communication, collaboration, and coordination among team members.

6.1.5 Testing and Quality Assurance

- **Unit Testing**: JUnit and Mockito are employed for writing unit tests for backend Java code, ensuring code reliability and facilitating automated testing.
- **Integration Testing**: Integration tests are conducted to verify the interaction between different components of the system, including frontend components, backend API endpoints, and database operations.

6.2 Tesing

Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.

6.2.1 Testing Methodology

White box Testing: is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing. We as the developers of SCMMS applied white box testing as our testing methodology.

6.2.2 STLC (Software Testing Life Cycle)

Test Planning

- While planning for testing of SCMMS system, we identified objectives of the test, areas that needed to be tested, areas that were not to be tested, and Scheduling Resource Planning.

Test Development

- In the test development, we prepared a checklist (Test case Development) and the description of the test cases (Test Procedure preparation).

Test Execution

- Test execution phase of SCMMS involved implementation of test cases and observing the result of the test.

Result Analysis

- Result analysis of testing SCMMS system contained Expected value: behavior of the system and Actual value: actual behavior of the system.

Bug Tracing

- We collected all the failed cases and prepared related documents.

Reporting

- The results of the test.

6.2.3 Types of Testing

Smoke Testing

- We performed an initial test on SCMMS system for the availability of all the functionalities.

Sanity Testing

- We then conducted an initial test on the system to check for the proper behavior of an application.

Regression Testing

- During the project, we carried out several tests on functionalities that we tested before whenever we made a change to the existing functionality.

Adhoc Testing

- We performed an Adhoc test (positive and negative) to perform GUI testing.

6.3 TCD (Test Case Document)

6.3.1 Test Scope

This involved the areas of the system to be tested including homepage, fund seeker dashboard and projects dashboard.

6.3.2 Test Scenario

Our test scenario for the system included when the fund seeker fills in the application form, upload documents and hit the quit button.

6.3.3 Test Procedure

Positive testing, Negative testing using the corresponding GUI test cases, helped us in performing adhoc testing.

6.4 Test Cases

6.4.1 Positive Test Cases

The positive flow of the functionality was considered, valid inputs were used for testing and expected value was positive to verify whether the requirements are justified. Table 1 below shows SCMMS positive test case.

Test Case Number	Description	Expected Value	Actual Value	Result
1	check for all features in the screen	must contain all the features	pass	Pass
2	Enter valid Username and Password	It should accept	Pass	Pass

6.4.2 Negative Test Case

Must have negative perception and invalid inputs must be used for test. Table below describes PSVSMs negative test case

Test Case Number	Description	Expected Value	Actual Value	Result
1	Try to modify vehicle	should not allow	pass	Pass
2	Enter invalid Username and Password	It should not accept	Pass	Pass

6.5 System Implementation

It is the process of ensuring that the information system is operational, and then allowing users to take over its operation for use and evaluation. Implementation of PSVSMS includes activities such as:

1. Training of end users how the whole system operates and the functionalities of the system
2. Completion of user documentation
3. System changeover
4. Evaluation of the system on regular intervals

Chapter 7

Results and Conclusion

7.1 Achievements

- Real-time Data Integration: Successfully integrating GPS devices to capture real-time location and speed data of public service vehicles (PSVs) is a significant achievement. This allows for accurate tracking and monitoring of vehicles.
- Comprehensive Record Keeping: Building a system that records detailed information such as insurance policies, safety features, and inspection reports contributes to better safety management of PSVs. It ensures that all necessary data is accessible and organized.
- User-Friendly Interface: Developing a user interface that is accessible to both passengers and law enforcement officers is crucial. Achieving a user-friendly design that allows passengers to easily access safety information by entering the vehicle registration number enhances user experience and promotes safety awareness.
- Integration with External Parties: Successfully integrating data provided by various stakeholders such as drivers, conductors, vehicle owners, SACCOs, and the National Transport and Safety Authority (NTSA) demonstrates effective collaboration and data management within the transportation ecosystem.

7.2 Lessons Learned

- Data Security: Ensuring the security and privacy of sensitive data, such as vehicle details and inspection reports, is paramount. Implementing robust security measures, including encryption and access controls, helps prevent unauthorized access and data breaches.
- Scalability: As the system grows and more vehicles are added, scalability becomes a key consideration. Designing the system architecture with scalability in mind allows for smooth expansion without compromising performance.

- User Training and Support: Providing adequate training and support for users, especially law enforcement officers who rely on the system for regulatory enforcement, is essential. Clear documentation and ongoing support help users effectively utilize the system and address any issues that may arise.
- Feedback Mechanism: Establishing a feedback mechanism for passengers and other stakeholders can provide valuable insights for continuous improvement. Soliciting feedback on the usability and effectiveness of the system allows for iterative enhancements based on user needs and preferences.
- Technology Stack Optimization: Continuously evaluating and optimizing the technology stack, including frameworks like ReactJS, Java, MariaDB, and Spring Boot, helps ensure optimal performance and efficiency of the system. Keeping abreast of advancements in technology allows for incorporating new tools and techniques to enhance system capabilities.

Chapter 8

Appendix