**Vehicle Park Management System**

**For**

**Open Source Systems Development (SENG 411)**

**Group 14**

**Group Members:**

Raymond Emmanuel Joseph

Rhema Emmanuel-Great Oshiokhua

Robinson Sopiriye Divine

Sanya Israel Eyitope

Taiwo Tomiwa Ephraim

Ukpong Abasimmiofon Aniefiok

Umanah Shepherd Ofonime

Oshioke Labib Oshiomah

Dada Teniola Emmanuel

Chuwuemeka Fortune

20/2297

20/0417

20/2850

20/2527

20/2768

20/0606

20/0855

20/2427

20/1554

20/0430

**Table of Contents**

**1. Introduction**

1.1 Project Overview

1.2 Project Deliverables

**2. Project Organization**

2.1 Software Process Model

2.2 Roles and Responsibilities

2.3 Tools and Techniques

**3. Project Management Plan**

3.1 Tasks

3.1.1 Description

3.1.2 Deliverables and Milestones

3.1.3 Resources Needed

3.1.4 Dependencies and Constraints

3.1.5 Risks and Contingencies

3.2 Conclusion

**4. Additional Materials**

**1. Introduction**

**1.1 Project Overview**

The vehicle park management system is a comprehensive software solution designed to manage vehicle parking facilities efficiently. The open source system was thoroughly enhanced by the members of Group 14 to include into the system the features and functionalities it previously lacked.

The systems initial purpose is to simplify the process of managing vehicles parked under a large organization, though it managed to fulfill its purpose it fell short in certain aspects of the user experience. This documentation outlines the updates and modifications made to the system to enhance its functionality.

The system invites the user to a home page which contains the name of the system without context to its uses. Alongside its name there exists links to the admin and user login pages. These pages allow users to login to their respective accounts depending on the role they play in the system. For users who don’t have an account they are given the chance to sign up for one. For those who might have misplaced their passwords they have been given a page where they can easily reset it by providing their user information.

Once a user logs in they are launched into the dashboard where no important information exists, after which the can proceed to view which vehicles of theirs is in the park or manage their user profile. Once a user is finished they can logout of their accounts from the menu.

The admins on the other hand log into the system from the admin login page, though they are not allowed to sign up they can at least reset their password incase they find themselves forgetting their login details. Once they login they will be greeted with the admin dashboard, unlike the user dashboard this page shows brief information on the ongoing of the system, information like the total number of vehicles in the park, number of registered users, etc. From this page the admin can traverse into a variety of pages and partake in the operations of the system, these pages include:

1. Manage Vehicles

* Add Vehicle Page
* Mange Incoming Vehicles Page
* Manage Outgoing Vehicles Page
* Add Vehicle Category
* Manage Vehicle Category

1. Report Page
2. Registered Users Page
3. Search Page
4. Admin Profile Page

**1.2 Project Deliverables**

Upon completion of the project, our team without a doubt included into the system features and functionalities necessary for user experience and the overall performance of the system that were not initially included in it. These deliverables are:

1. Updated system software with a minimalist user interface designed to provide information to users in a clear and uncluttered manner while still providing an interactive user experience.
2. Advanced search functionalities that allow admins to search through users and vehicles with a name as opposed to the previous method of just searching vehicles with a parking number.
3. Detailed report generation, this feature allows the system to track any and every activity that affects the experience of all users within the system, these activities include, “adding a new vehicle, category, or user”, “editing an existing category”, etc.
4. Minor additional changes include changes to the form values and the information they require so as to provide a more informative system of collecting information from users.

**2. Project Organization**

**2.1 Software Process Model**

During the development of this project, our team members employed the agile software development methodology throughout the life cycle of the system. Agile methodology was chosen due to its flexibility and iterative approach. It divides projects into small, iterative cycles, allowing for continuous feedback and adaptation to changing requirements. This model is useful mostly because its main aim is to facilitate quick project completion, with the time provided for the project and the continually changing requirement from our members, this approach fit the part perfectly. This approach allowed us to adapt to the ever-changing requirements while leaving space for future incremental updates.

**2.2 Roles and Responsibilities**

Each team member played a distinct role in the play of this project. The project roles and responsibilities were as follows:

1. Project Manager:

* Taiwo Tomiwa Ephraim – 20/2768

1. Developer:

* Dada Teniola Emmanuel – 20/1554

1. Designers:

* Robinson Sopiriye Divine – 20/2850
* Raymond Emmanuel Joseph - 20/2297
* Dada Teniola Emmanuel – 20/1554

1. Documentation Specialists:

* Taiwo Tomiwa Ephraim – 20/2768
* Ukpong Abasimmiofon Aniefiok - 20/0606
* Dada Teniola Emmanuel – 20/1554

1. Testers:

* Oshioke Labib Oshiomah - 20/2427
* Sanya Israel Eyitope - 20/2527
* Dada Teniola Emmanuel – 20/1554

1. Researchers:

* Rhema Emmanuel-Great Oshiokhua - 20/0417
* Ukpong Abasimmiofon Aniefiok – 20/0855
* Chukwemeka Fortune – 20/0430
* Dada Teniola Emmanuel – 20/1554

**2.2 Tools and Techniques**

Our members utilized various tools and techniques to achieve its objectives:

**Programming Tools:**

1. **HTML (Hypertext Markup Language):** HTML is the standard language used to create web pages. It defines the structure and content of a web page using elements and tags.
2. **CSS (Cascading Style Sheet):** CSS is used for styling and formatting web pages. It controls the layout, colours, fonts, and other visual aspects of a website.
3. **JavaScript:** JavaScript is a programming language that adds interactivity and dynamic behaviour to web pages. It enabled us to add animations, form-validation and more to make the website more engaging.
4. **PHP:** PHP is a server-side scripting language often used for web development. It processes data on the server and generates dynamic web pages. PHP is known for its role in user authentication, content management systems, custom application development, and seamless integration with frontend technologies, enabling the creation of interactive and data-driven web experiences.
5. **XAMPP:** XAMPP is a free, open-source software package that includes Apache (a web server), MySQL (a database), PHP, and Perl. It's used to set up a local web development environment.

**Version Control:**

1. **Git:** Git is a version control system used to track changes and maintain multiple versions of the codebase.

**Database Management System:**

1. **MySQL:** MySQL is an open-source relational database management system commonly used for storing and managing data in web applications.

**Integrated Development Environment (IDE)**

1. **Visual Studio Code:** Visual Studio Code (VS Code) is a popular, lightweight, and open-source code editor developed by Microsoft. It's widely used for web development and supports various programming languages and extensions.

**3. Project Management Plan**

**3.1 Tasks**

**3.1.1 Description**

The already existing system we were provided performed its operations properly with little or no error during its operations, in spite of this the system still lacked certain functionalities when it comes to user experience.  
These short comings were the tackled by our development team to provide an overall better user experience:

1. **User Interface Updates:** the previously existing system was old, following the design conventions of the early 20s. These conventions practice a somewhat cluttered interface revealing to the user a large amount of menu options leaving them stranded when accessing the features of a new system, these conventions were also fund of using highly contrasting colors alongside small texts with large amount of blank space. Though these conventions aimed to keep things simple their time has passed as the world has moved to more visually appealing designs, from minimalist to brutalist, 2023 has far exceeded the traditional conventions of design.

Out updated system battled these conventions by implementing a minimalist user interface only showing the user on first site the necessary information needed for them to easily operate it, with the presence of color harmony and visually appealing designs and animations, our system presents a more modern approach to the UI design.

1. **Advanced Search Implementation:** another short coming of the original system was the search algorithm, for what we hope to be a good reason the system only searches through vehicles in the database and it does this by requesting from the user a maximum of 10-digit number called the “parking number”. As simple as that seems, in a system of over 1000 users not only will the admin not be able to keep track of parking numbers of all users but the users themselves won’t remember as they would be forced to memorize a parking number each time they park their vehicle.

Our system solved this by providing an advanced search functionality that now requests from the user a “name”. This name will be used to query the database for users, vehicles, users whose vehicles are still in and those whose vehicles are out. This new functionality now provides a variety of information on the users and vehicles in the system thereby reducing the burden of memorization from the users.

1. **Detailed Report Generation:** the previous system has a page for report generation, this page takes from the user a “from date” and “to date”, this helps the system know from where to when the report is supposed to be from. This feature though it works is incomplete as it only generates reports on vehicles that have been added during the inserted period of time, meaning the system only checks from vehicles whose time of entry is between the “to and from date”.

Our updated system pushes this feature to more intelligent heights by keeping track of almost every query to the database performed during the course of using system. The actions tracked by the system are with the exception of:

* Password Change
* User Login or Logout
* Changing of a vehicle from In to Out

The system tracks all other information and orders them by “date of event” in descending order.

1. **Minor Adjustments:** alongside the above there were minor adjustments we made to the system that barely impact the overall user experience these changes are mentioned below.

When adding a vehicle there is a field to input contact number of the owner of the vehicle, in order to be able to track the record, we edited the system to stop taking just any contact number, now the admin selects an existing users name from a list of options to show that the vehicle was parked under a user registered in the system.

Changes were also made to the mode of deleting users and vehicles from the system, this feature as minor as it is, now allows the deletion of records from an page its displayed as opposed to the original way where only incoming and outgoing vehicles could be deleted.

Layout changes to the menu by grouping similar pages under one name making the pages less cluttered.

**3.1.2 Deliverables and Milestones**

1. **User Interface Updates**
   * Deliverable: A minimalist, uncluttered, interactive and modern user interface.
   * Milestone: Completion of UI design
2. **Advanced Search Implementation**
   * Deliverable: Functional advanced search feature
   * Milestone: Successful implementation of advanced search
3. **Detailed Report Generation**
   * Deliverable: Enhanced reporting system
   * Milestone: Completion of detailed report functionality
4. **Minor Adjustments**
   * Deliverable: Bug-free and updated software
   * Milestone: Resolution of all identified issues

**3.1.3 Resources Needed**

* **Hardware:**
* Development PCs
* Testing Environment
* **Software:**
* Visual Studio Code
* XAMPP
* Figma
* Git
* **Human Resources:**
* Full-Stack Web Developer
* UI/UX Designers
* Document Specialist
* Testers
* Researchers

**3.1.4 Dependencies and Constraints**

* User Interface Updates must precede Advanced Search Implementation.
* Time constraints due to client requirements limiting the further implementation of additional software features and functionalities.

**3.1.3 Risks and Contingencies**

* **Risk**: Unforeseen technical challenges during search implementation.
  + **Contingency**: Frequent progress meetings to address challenges promptly.
* **Risk**: Code from previous system affecting the updated system.
  + **Contingency**: Careful evaluation of existing code before making changes to it.

**3.2 Conclusion**

The project management plan provided a structured framework that meticulously defined the key components of the Vehicle Park Management System. It laid out in detail the tasks, deliverables, resources, dependencies, and potential risks associated with this complex undertaking. This comprehensive document served as the project's North Star, offering clear guidance and a roadmap for success. By adhering to this well-structured project management plan, the project team efficiently steered the VMS project to its successful conclusion.

**4. Additional Materials**

Additional Materials related to the project include:

* Updated UI design mockups.
* Source code for the updated system.
* Project presentation.