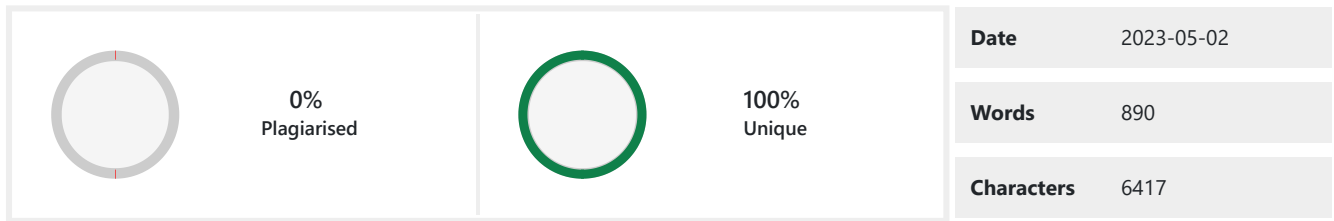


## PLAGIARISM SCAN REPORT



## Content Checked For Plagiarism

### CHAPTER 3: SYSTEM ANALYSIS

#### 1.

##### 1.1. Requirement Analysis

Requirements analysis is the process of obtaining what the client, user, or stakeholders expect from the system or software. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

The functional requirements specify the core features and the expected behaviours of the system and is modelled using Data Flow Diagrams, UML diagrams, flowcharts, function modelling, and so on. The non-functional requirements are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality.

##### 1.1.1. Functional Requirements

The functional requirements specify the core expected behaviour of the system. It is a description of the service that the software must offer and describes a software system or its component. In this system some core functions that have been modelled sing the use-case diagram and use-case descriptions.

The core features of the system are to add and update charging stations by the admin, add and edit charging station ratings by the user, and get recommendation of charging station by user.

##### 1.1.1.1. Use-case diagram

Figure 3.1: Use case of the System

##### 1.1.1.2. Use-case descriptions

Table 3.1: Use case description of rating a charging station

Identifier

UCD1

Name, Description

Providing the rating to a charging station

Actors

User

Pre-condition

The location must be specified and charging station must be selected

Success scenario

Rating is inserted into the database

Success message is displayed to user

Failure scenario

Rating is not inserted into the database and failure message is displayed to admin

Table 3.2: Use case description of adding a charging station

Identifier

UCD2

Name, Description

Adding a charging station

Actors

Admin

Pre-condition

The location must be specified

Success scenario

Charging station is inserted into the database

Success message is displayed to admin

Failure scenario

Charging station is not inserted into the database and failure message is displayed to admin

Table 3.3: Use case description of updating a charging station

Identifier

UCD3

Name, Description

Updating a charging station

Actors

Admin

Pre-condition

The location must be specified and the charging station to update must be selected

Success scenario

Success message is displayed to admin

Charging station is updated in database

Failure scenario

Charging station is not updated in database and message is displayed to admin

Table 3.4: Use case description of getting a charging station recommended

Identifier

UCD4

Name, Description

Get recommendation of charging station

Actors

User

Pre-condition

The location must be specified

Success scenario

Charging stations and their location is displayed to user if the charging station is available

Message is displayed to user if the location has no charging station

Failure scenario

Charging station is not recommended as desired

#### 1.1.2. Non-Functional Requirements

- \* Speed: It is used to determine how fast the system performs certain activities.
- \* Security: Only registered users can login to the system and provide ratings or get recommendations.
- \* Availability: The system will be available at all hours, every day of the year once deployed.
- \* Usability: The system should be easy to access, use, and understand by the users

## 1.2. Feasibility Study

### 1.2.1. Technical Feasibility

The dataset to train and test the recommendation system can be available from sources like NEA and Kaggle. Laravel framework would be used to build the recommendation system and the collaborative filtering technique with cosine similarity would be used for the recommendation algorithm. The next step would be to develop the recommendation system using the Laravel framework and the chosen algorithm. This would involve the implementation of various features such as user authentication, data management, and recommendation generation. The developed system would then be tested to ensure its functionality and performance. This would involve evaluating the system's accuracy and efficiency in generating recommendations, as well as user satisfaction with the system. Once the system has been tested and evaluated, it would be deployed for use by EV owners. This would involve the integration of the system with existing EV charging stations and the implementation of any necessary security measures.

All in all, the technical feasibility of this project is high as the Laravel framework is a robust and widely used platform for web development, and collaborative filtering with cosine similarity is a well-established recommendation algorithm. With the appropriate data and resources, the development, testing, and deployment of this system is achievable.

### 1.2.2. Operational Feasibility

The system will be able to recommend the EV charging station to the user in the specified location. This recommendation will be based on the ratings the user has provided to the EV charging stations that the user has already been to and rated. This solves the problem for the user to manually search the EV charging station and make a guess of whether the one they found may be one of their liking or not and whether they will be able to trust their expensive vehicles on the said charging station or not. Furthermore, having a system of authentication and authorisation will also help prevent fake ratings and make the system more reliable and robust.

### 1.2.3. Schedule Feasibility

Figure 3.2: Work breakdown structure

Figure 3.3: Gantt chart

## 1.3. System Analysis

### 1.3.1. Figure 3.4: Class diagram

Class Diagram

### 1.3.2. Activity Diagram

Figure 3.5: Activity diagram to provide rating or receive recommendation

### 1.3.3. Sequence Diagram

Figure 3.6: Sequence diagram to receive recommendation after adding a station

No plagiarism found

Check By:  Dupli Checker