SOFTWARE DESIGN DOCUMENT

FOR

AUTOMATIC NUMBER PLATE RECOGNITION AND PARKING OCCUPANCY PREDICTION

PREPARED BY,

SOURAV A K

ADITHYAN S P

MITHUN K

CONTENTS

1. OVERVIEW

1.1 SCOPE

The scope of this project is to provide a fairly accurate prediction of parking occupancy rates in a parking lot in the future. In addition, a web page is designed for the security officials to monitor the vehicles inside the campus. The prediction part of this project can be implemented in public and private parking lots, garages, etc. Clustering of lots can be done to improve the prediction accuracy. Number plate identification can be used in toll booths to automatically charge the toll and reduce traffic. The document is prepared in IEEE 1016-2009 standards.

1.2 PURPOSE

The purpose of this document is to present a detailed description of the Web page that shows parking occupancy prediction rates and details of vehicles in the campus. It will explain the purpose and features of the web page and give a general description of design elements and their interactions, and how the system will be structured. The document aims to provide a detailed overview of our software product, its parameters, and goals.

1.3 INTENDED AUDIENCE

This SDD is intended for the following audiences:

- The team members include project guide, coder, developer, tester, documentation writers
- Professors of the college

The Web page is intended for the following audiences:

- Web page showing details of vehicles inside the campus will be handled by campus security officers.
- Web page showing prediction rates is available to college staff and others who have permission to enter the campus such as students' parents.

1.4 REFERENCES

Websites :

- https://www.javatpoint.com/number-plate-recognition-using-python
- https://www.geeksforgeeks.org/license-plate-recognition-with-opencv-and-tesseract-ocr/
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9765513
- https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

Reference Books :

- Introduction to Machine Learning with Python by Andreas C. Müller, Sarah Guido

2. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

BLOCK DIAGRAM - A diagram showing in schematic form the general

arrangement of the parts or components of a complexsystem or processes.

PC - Personal Computer

SDD - Software Design Description

SRS -Software Requirement Specification STAKEHOLDERS - Any

person or entity that has an interest in the

System.

USER - A person who uses and interacts with the system.

USER INTERFACE - An interface that our system contacts with the user of the

system, It gets all needed information for its running,from the user to our system.

IEEE - Institute of Electrical and Electronics Engineers

HTML -Hypertext markable language

CSS -cascading style sheet

3. CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS

This section includes the concepts and context of SDD in which the documentation is prepared. The purpose of the conceptual model is to better understand the system terminology and the software life cycle that the system resides on. The conceptual model also gives information about stakeholders who will use SDD and how the SDD will be used.

3.1 SOFTWARE DESIGN IN CONTEXT

The web page will be implemented using HTML and CSS for the front end, and Django for the backend. The webpage uses real time video input from an external camera to identify number plates. The trained machine learning model predicts the parking occupancy rate and displays the result in a web page. Another web page shows the details of vehicles inside the campus instantaneously, distinguishing staff vehicles from outside vehicles.

3.2 SOFTWARE DESIGN DESCRIPTIONS WITHIN THE LIFE CYCLE

This software will be created following the IEEE standards. The primary milestones of the system are requirements analysis, design description analysis, implementation, and verification and validation.

3.2.1 INFLUENCES ON SDD PREPARATION

The very first influence on the software design process is the SRS document. In SDD, we considered the product perspective, functional/non-functional requirements, and interface requirements that were included in the SRS. Given specifications and the possible new requests from the stakeholders will specify the design process of this system

3.2.2 INFLUENCES ON SOFTWARE LIFE CYCLE PRODUCTS

Before connecting the software and hardware parts of the system, the user interface should be shown with sample examples, which are processed using the interface, to the stakeholders. As a result of this process, stakeholders can share their ideas and requirements about the project. Finally, software and hardware parts will be connected. Furthermore, SDD will guide us all the way through the system. According to this document or the first phase, some requirements can be added or removed from the software requirements. Consequently, the requirements of the stakeholders can be met more precisely after each sprint of our development process.

3.2.3 DESIGN VERIFICATION AND DESIGN ROLE IN VALIDATION

Verification is the process that will test the prediction rates webpage and vehicle details webpage whether it meets a set of design specifications. In this process, we will look at the SRS and SDD documents for the correctness of specifications. We will control whether all functional and non-functional requirements are correctly implemented according to the requirements of SRS and SDD documents. Furthermore, we will control whether the design viewpoints of the final project are met in the viewpoints part of the SDD document.

Validation is the process that the stakeholders and developers decide if the website is consistent with the main goal, which is allowing the user to purchase furniture by checking the compatibility with the respective environment. After the complete implementation of the system, the testing process will be handled with SDD-influenced test plans and cases.

4. DESIGN DESCRIPTION INFORMATION CONTENT

4.1 INTRODUCTION

The software description of both the prediction rates webpage and vehicle details webpage analyzes how the system will be designed and implemented. This section investigates these according to the identification of the SDD, identified design stakeholders and design concerns, related design viewpoints, design views, overlays, rationale, and languages. Furthermore, this section includes design elements which are design entities, attributes, relationships, and constraints.

4.2 SDD IDENTIFICATION

The prediction and vehicle details web dashboard will be released after validation and verification tests. The prototype of the system will be shown in July. Scope, references, and context can be found under the section “Overview”. The glossary can be found under the section “Definitions, Acronyms andAbbreviations”.

4.3 DESIGN STAKEHOLDERS AND THEIR CONCERNS

Stakeholders comprise the team members of this project along with the faculty. Main concerns of the stakeholders are quality of the system because it must have high quality.

4.4 DESIGN VIEWS

Design views help design stakeholders about focusing on design details from a specific perspective and meet relevant requirements. Each identified design concern must be the topic of at least one design view so that SDD is complete. Each design concern identified in the previous subsection is the topic of most of the design views in this document; thus, this SDD is completed. For example, concerns about cost are a topic of composition view. Moreover, concerns about the system's quality are a topic of logical view. In this document, context, composition, logic, dependency, information, patterns of use, interface, interaction, and state dynamics views will be explained in section 4.5 as their corresponding viewpoints.

4.5 DESIGN VIEWPOINTS

This document describes context, composition, logic, dependency, information, patterns of use, interface, structure, interaction, and state dynamics viewpoints.

Context Viewpoint*:* It describes the relationships, dependencies, and interactions between the system and its environments such as users and other interacting stakeholders. Interactions between the system and its actors are very intense, hence concerns from this viewpoint are important and suitable for prediction rate and vehicle details webpages. It includes a use case, context, and block diagram showing the system boundary.

Composition Viewpoint: It describes how the design subject is split up into its components and which roles these components have. It can be used in estimating prediction rates , staffing and scheduling duties of a development team. It includes a deployment and component diagram.

Logical Viewpoint: It describes class structures, interactions between them, and how they are designed and implemented. Also, it supports the development and reuse of existing logical components. It includes a class diagram that defines objects and classes and the relationshipsbetween them.

Dependency Viewpoint: It describes the components of the system and the dependencies between these components. It gives information about shared information and the order of execution of these components.

Information Viewpoint: It describes data items, data types and classes, data stores, and access mechanisms. It gives information about data attributes.

Patterns Use Viewpoint: It describes design patterns and usage of design patterns that meet the design ideas of the project.

Interface Viewpoint: It describes the details of external and internal interfaces. It provides information to the designers, programmers, and testers before proceeding with the detailed design of the system. This also provides designers, programmers, and testers to use the systemas a random user.

Interaction Viewpoint: It describes the sequence of actions and how, why, where, and at what level actions occur in the system. It is preferred to use state dynamics views in detail forthis project.

State Dynamics Viewpoint: It describes the internal behavior of the system. System dynamics include modes, states, transitions, and reactions to events. It gives information step by step about the system operation. It includes a state machine diagram that defines conditions, states, transitions, and relationships between them.

4.6 DESIGN ELEMENTS

Any item which appears in a design view is named as a design element. It may be one or some of these subcases; design entity, design relationship, design attribute, and design constraints. All design elements are defined with subcases under their corresponding viewpoint in section 5 of the software design description

4.7 DESIGN OVERLAYS

Design overlays are usually used to add information to the existing views. This concept will be explained clearly, when necessary, in the design viewpoints in section 5.

4.8 DESIGN RATIONALE

Object - Oriented approach was chosen while designing because in this way the hardware part and software parts will be combined easily. The software part includes parsing and reading notes and transmitting data. To design these lots of packages were used. These packages are connected to each other and can be controlled separately. Furthermore, for the hardware part, a package was used and there is another package to combine software and hardware parts.

4.9 DESIGN LANGUAGES

In this document, unified Modeling Language (UML) will be used as the modeling language for the diagrams. The modeling language is used for emphasizing the static structure and dynamic behavior of the system.

5. DESIGN VIEWPOINTS

5.1 INTRODUCTION

This section provides several main design viewpoints of the prediction web page with their corresponding design concerns and appropriate design languages. Respectively, context, composition, logic, dependency, information, patterns, interface, structure, and interaction viewpoints are defined in the following subsections. Short descriptions relating a minimal set of design entities, design relationships, design entity attributes, and design constraints are provided for each viewpoint.

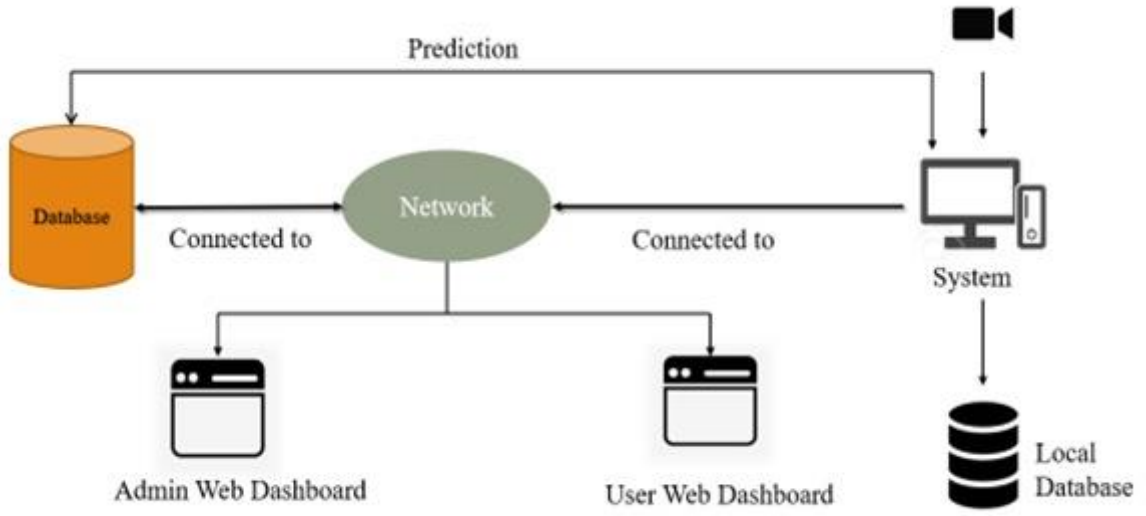
5.2 CONTEXT VIEWPOINT

The context viewpoint of the prediction web page shows the functions provided by a design subject with reference to an explicit context. The services are the functions, which describe the relationships, dependencies, and interactions between the system and its environment like users.

5.2.1 DESIGN CONCERNS

Design concerns consist of services, actors, and system boundaries. Prediction web page and vehicle details web page are formed of the interface and the hardware equipment; there are two types of actors which use the system. The services and the system boundaries are established according to the needs of the users.

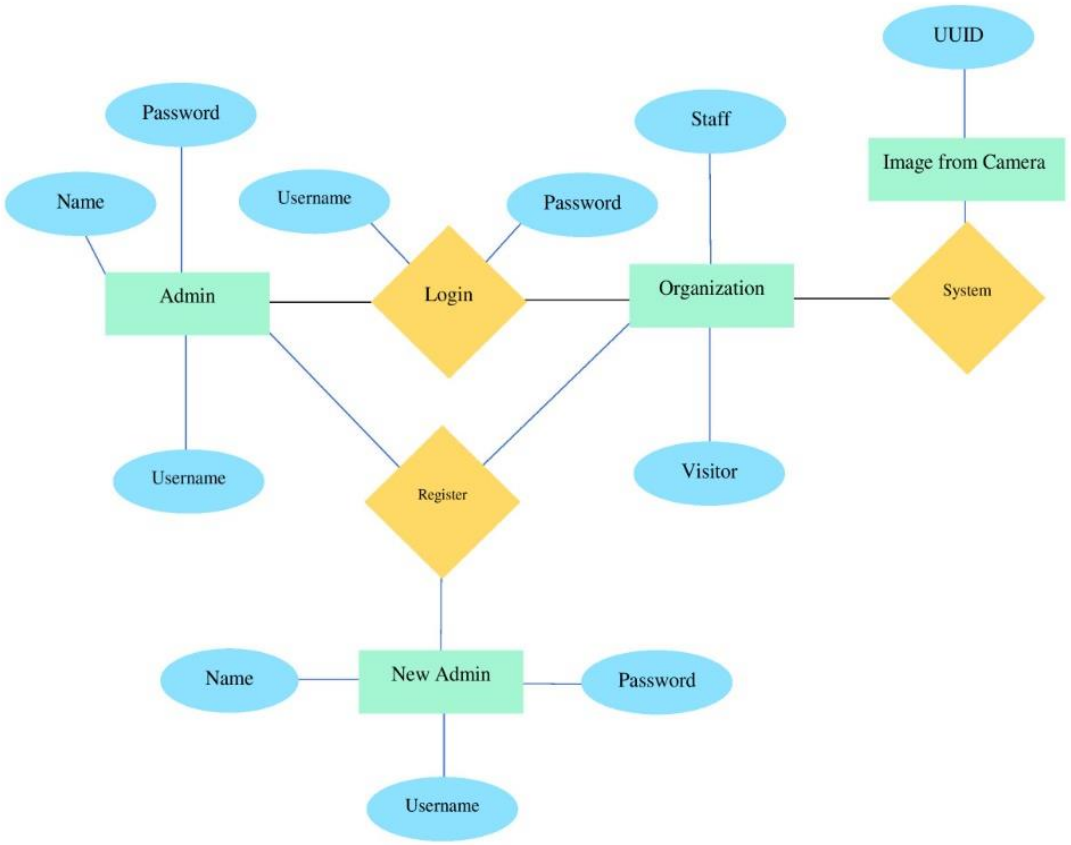
The user needs to log in to the website to check the prediction rate of parking occupancy in the future. The security officials can see the details of vehicles inside the campus by logging into their website. Information flow between the user and the website is provided by the interface.



Block diagram

5.2.2 DESIGN ELEMENTS

The below diagram shows the design relationship which includes provided input and received output between the user and the other major components of the system. The below diagram describes system interaction and functionalities with users.



ER-diagram

5.2.3 EXAMPLE LANGUAGES

The diagram given in the previous subsections is created by the UML. One of them is a block diagram describing the interrelationships of a system. The last one is a use case diagram showing user interactions with the system.

5.3 COMPOSITION VIEWPOINT

The purpose of the composition viewpoint is to define the system as a composition of its subsystems. The project is formed by many submodules.

5.3.1 DESIGN CONCERNS

With the help of the information from a composition viewpoint, system stakeholders and the developer team can plan and control the software product. The design of this project is structured into sub-modules and components such as interface and other packages. The project is managed by planning, monitoring, and controlling these components. All acquired information about the project

provides estimated cost, staffing, and schedule for the development effort

5.3.2 DESIGN ELEMENTS

The project is formed of interface and classes inside packages as design entities. User The interface package consists of the user interface and its actions. It performs connections between other packages via user actions.

5.4 LOGICAL VIEWPOINT

The purpose of the Logical viewpoint is to elaborate on existing and designed types and their implementations as classes and interfaces with their structural static relationships. For each entity, there will be a diagram to overview the entity, and then a table of the name, and return type; the visibility of the entity/class diagram is shown. Also, definitions of each element are provided. After all, elements are explained, the class diagram that shows the relationships between the classes is drawn.

5.4.1 DESIGN CONCERNS

The logical viewpoint is employed to show the development and reuse of abstractions and their implementations. This means object-oriented programming simplifies maintaining and modifying existing code while new objects are created. Since identifying object classes is often a difficult part of object-oriented design, during the implementation phase of the project there can be some changes in object identification.

5.4.2 DESIGN ELEMENTS

The project has three packages named User Interface, Calculator, and Account Manager. All packages have different classes. All package connections and their classes can be seen in the figure.

5.4.3 EXAMPLE LANGUAGES

A class diagram that describes the structure of a system by showing classes of the system has been given in the previous sections using UML modeling language.

5.5 DEPENDENCY VIEWPOINT

This section is not required for our project.

5.6 INFORMATION VIEWPOINT

This section is not required for our project.

5.7 PATTERNS USE VIEWPOINT

This section is not required for our project.

5.8 INTERFACE VIEWPOINT

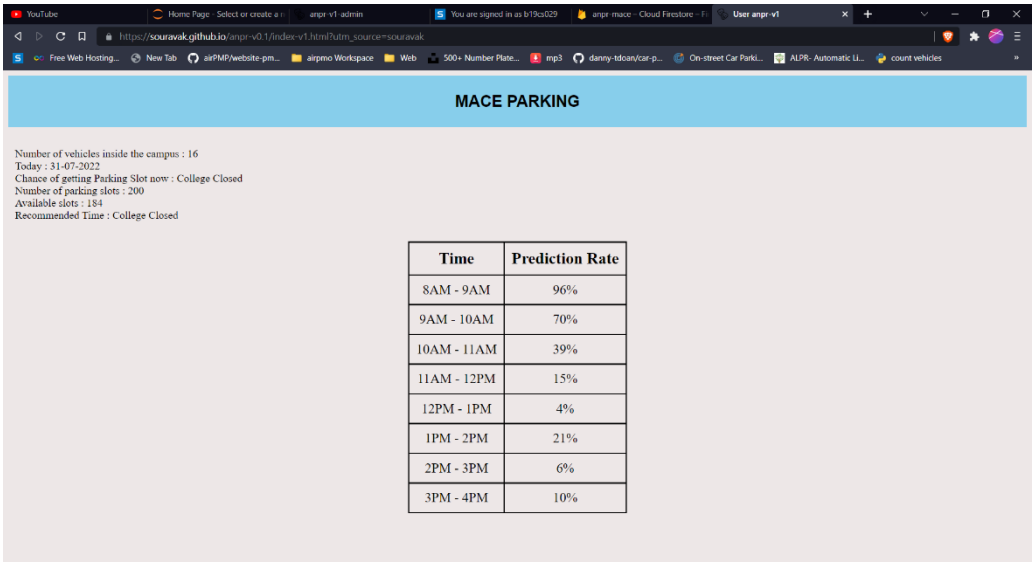
In this part of the document, the details of external and internal interfaces will be defined. There shall be two interfaces in prediction rate website, which are Database Interface and the user interface. The database interface is hidden from users. Clients interact through the user interface and provide necessary information for calculation procedures and those information's stored in the database.

5.8.1 DESIGN CONCERNS

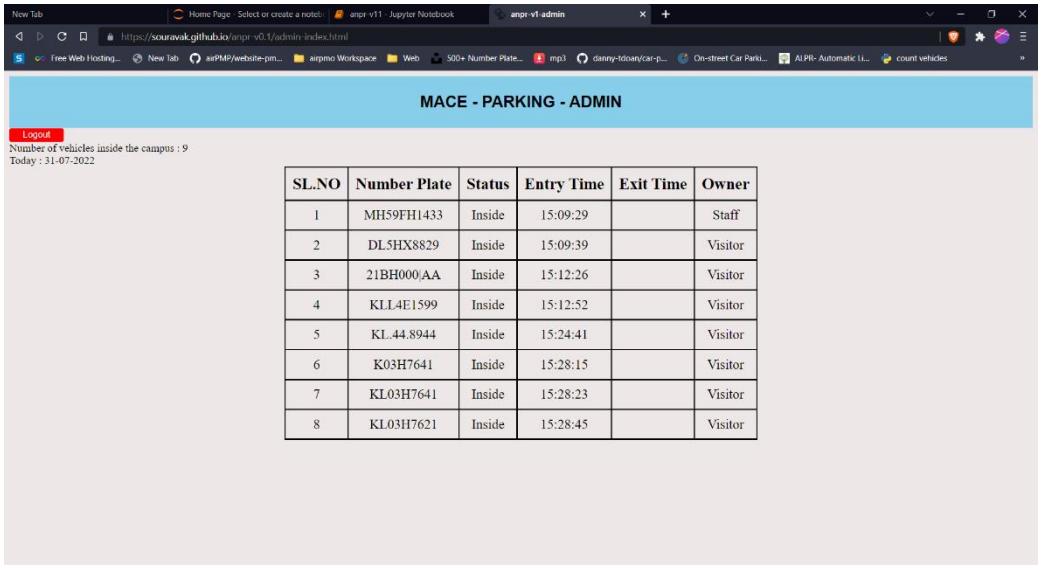
Interface viewpoint provides information to the designers, developers, and testers before proceeding with the detailed design of the system. It also informs them about how cooperating entities will interact. With the ease of each interface description, designers and developers can know the internal and external connections of the system to develop it. This contributes to ease of maintenance.

5.8.2 DESIGN ELEMENTS

In this subsection, the user interface of the home page of the prediction rates website is shown below.



Prediction page



Vehicle details admin page

DATABASE DESIGN

The database design for is given below.

User (Local DB)
Username E-mail Age Password

6. TECHNOLOGICAL STACK

The Technological Stack is given below