

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**SVMU: Smart Vehicle Management in UAAR using AI based License Plate Recognition System**

***By***

**Hira Tassadaq 19-ARID-805**

**Muhammad Bilal 19-ARID-825**

***Supervisor*Dr. Muhammad Aqib**

***Bachelor of Science in Computer Science (2019-2023)***

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Hira Tassadaq Muhammad Bilal

--------------------------- ---------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Smart Vehicle Management in UAAR using AI based License Plate Recognition System” was developed by **“Muhammad Bilal (19-Arid-825)” and “Hira Tassadaq (19-Arid-805)”** under the supervision of **“Dr. Muhammad Aqib”** and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science.

---------------------------------------

**Supervisor**

--------------------------- ---------------------------

**Ms Iram Rubab Mr Imran Khurram**

**Examiner**

**Executive Summary**

SVMU (Smart Vehicle Management in UAAR) is an intelligent vehicle management system developed for UAAR. It utilizes AI-based license plate recognition with a user-friendly website and hardware implementation to efficiently manage vehicle access at entry point. The system captures and recognizes license plates of vehicles entering through the gate, employing a camera installed at the entry point to record video and extract license plate images in real-time.

Using advanced image processing techniques in Python, the system intelligently filters the images and converts the vehicle numbers into text format. The extracted text is then compared against the license plate records stored in the database. If a match is found, the system notifies the sensors integrated into the barrier, indicating that the barrier should be raised to allow the vehicle to pass. Sensors placed above the barrier detect the presence of vehicles, ensuring the barrier remains open while a vehicle is underneath. Once the vehicle passes the barrier, the barrier sensor triggers its return to the original position. In the event that the vehicle number is not stored in the database, the system signals the barrier to remain in its original position and emits a beep as an alert. Unregistered vehicles have the option to register with the security guard stationed at the gate. Additionally, manual operation of the barrier is available as a contingency measure in case of system failure.

The SVMU website serves as a comprehensive platform for monitoring and managing vehicle access. It maintains a daily log of passing vehicles, records vehicle inspections and facilitates new vehicle registrations. Furthermore, a real-time hardware implementation showcases the functionality of the proposed system, providing a practical demonstration of its capabilities.

By employing SVMU, UAAR aims to streamline and automate the vehicle access process, enhancing security and improving efficiency at entry points.

In conclusion, SVMU provides a robust and user-friendly solution for intelligent vehicle management. It leverages AI-powered license plate recognition, a well-designed website, and efficient hardware implementation to effectively filter and process license plate information, ensuring seamless access control. The system's intelligent license plate recognition, database integration, and barrier control mechanisms contribute to seamless and effective management of vehicles entering and exiting the premises.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “**Dr. Muhammad Aqib**” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Hira Tassadaq Muhammad Bilal

--------------------------- ---------------------------

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 3

[1.2 Relevance to Course Modules](#_Toc268523779) 4

[1.3 Project Background](#_Toc268523780) 5

[1.4 Literature Review](#_Toc268523780) 6

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 7

**Problem Definition** 9

[2.1 Purpose](#_Toc268523795) 10

[2.2 Product Functions](#_Toc268523796) 11

[2.3 Proposed Architecture](#_Toc268523804) 12

[2.4 Project Deliverables](#_Toc268523806) 13

[2.5 Operating Environment](#_Toc268523807) 14

[2.6 Assumptions and Dependencies](#_Toc268523808) 15

**Requirement Analysis** 19

[3.1 Functional Requirments](#_Toc268523823) 20

[3.2 Non – Functional Requirments](#_Toc268523825) 21

[3.2.1 Usability](#_Toc268523787) 11

[3.2.2 Reliability](#_Toc268523787) 11

[3.2.3 Performance](#_Toc268523787) 11

[3.2.4 Supportability](#_Toc268523787) 11

[3.2.5 Design Constraints](#_Toc268523787) 11

[3.2.6 Licensing Requirements](#_Toc268523787) 11

[3.3 Use case Model](#_Toc268523823) 12

[3.3.1 Use Case Diagarm](#_Toc268523787) 12

[3.3.2 Actors Discription](#_Toc268523787) 16

[3.3.3 Use Case Discription](#_Toc268523787) 16

**The Design** 26

[4.1 UML Structural Diagrams](#_Toc268523830) 26

[4.1.1 Component Diagram](#_Toc268523787) 27

[4.1.2 System Component Diagram](#_Toc268523787) 28

[4.1.3 Package Diagram](#_Toc268523787) 29

[4.1.4 Deployment Diagram](#_Toc268523787) 30

[4.2 UML Behavioral Diagrams](#_Toc268523830) 31

[4.2.1 Activity Diagrams](#_Toc268523787) 31

[4.2.2 State Machine Diagrams](#_Toc268523787) 32

[4.3 UML Interaction Diagrams](#_Toc268523830) 33

[4.3.1 Sequence Diagrams](#_Toc268523787) 33

[4.4 Node Structure](#_Toc268523830) 34

[4.5 Communication Design Protocol](#_Toc268523830) 35

**Implementation** 38

[5.1 Communication Protocol Implementation](#_Toc268523830) 38

[5.2 PC Application Implementation](#_Toc268523830) 39

[5.3 Embedded Application Implementation](#_Toc268523830) 40

[5.4 Wireless Sensor Application Implementation](#_Toc268523830) 42

**Testing and Evaluation** 45

[6.1 Verification](#_Toc268523830) 46

[6.1.1 Functional Testing](#_Toc268523787) 47

[6.1.2 Static Testing](#_Toc268523787) 48

[6.2 Validation](#_Toc268523830) 50

[6.3 Usability Testing](#_Toc268523830) 54

[6.4 Unit Testing](#_Toc268523830) 55

* 1. [Integration Testing](#_Toc268523830) 55

6.6 [System Testing](#_Toc268523830) 56

**Conclusion and future work** 57

[6.1 Conclusion](#_Toc268523830) 58

6.6 [Future work](#_Toc268523830) 58

**References** 59

**List of Figures**

Fig 1.2 Proposed system 4

Fig 1.6.1 Agile Methodology Process Diagram 9

Fig 1.6.2 Agile Values 10

Fig 2.6.1 ANPR China 14

Fig 2.6.2 ANPR India 15

Fig 3.3.1 Use Case Diagram 20

Fig 4.1.1 Three Tired Architecture 26

Fig 4.1.2 System Architecture 27

Fig 4.3.1 Class Diagram 28

Fig 4.3.2 Component Diagram 29

Fig 4.3.3 Package Diagram 30

Fig 4.3.4 Deployment Diagram 31

Fig 4.4.1 Activity Diagram 32

Fig 4.5.1 Sequence Diagram 33

Fig 4.5.1 Sequence Diagram 34

Fig 4.5.1 Sequence Diagram 35

Fig 4.5.2 Node Diagram 34

Fig 4.5.2 Node Diagram 35

Fig 5.3.2 SVMU 38

Fig 5.3.3 Admin login 39

Fig 5.3.3 User login 39

Fig 5.3.3 Admin Register 39

Fig 5.3.3 User Register 39

Fig 5.3.3 Screening 39

Fig 5.3.3 Log 43

Fig 5.3.3 Management 43

Fig 5.3.4 Home Screen 44

**List of Tables**

Table 1.5 Related System Analysis 3

Table 2.3.1 Project Delieverables 7

Table 2.3.2 Development Requirements 8

Table 3.1 Functional Requirements 11

Table 3.2 Non-Functional Requirements 15

Table 3.3.2 Use Case Descriptions 19

# Chapter 1: Introduction

# Brief

The SVMU (Smart Vehicle Management in UAAR) is an advanced vehicle management using AI based License Plate Recognition System. This system incorporates AI-based license plate detection and recognition capabilities, along with a user-friendly website and hardware implementation. The primary objective of the proposed project is to accurately identify and record the license plates of vehicles as they enter through a designated gate. At the entry point, a camera will be strategically positioned to capture the frontal view of each vehicle in real-time. By recording videos, the camera will enable the extraction of images containing the vehicles' license plates. The SVMU system will utilize intelligent image filtering techniques to isolate and extract the license plate numbers from the captured images. These numbers will then be converted into textual format using Python image processing algorithms.

The converted text will be compared against the existing records of license plate numbers stored in the system's database. When a match is found, the SVMU system will activate sensors integrated into the barrier, prompting its removal. This will allow the authorized vehicle to pass through the barrier without obstruction. Sensors installed above the barrier will continuously detect the presence of vehicles beneath it, ensuring that the barrier remains open for as long as a vehicle is present. Once the vehicle successfully passes through the barrier, the barrier sensor will promptly return the barrier to its original position. In cases where the vehicle number is not stored in the database, the SVMU system will signal the barrier to maintain its original position and emit an audible beep as a notification.

For unregistered vehicles, a registration process will be available at the gate, where a security guard can facilitate their inclusion in the system. Additionally, a manual operation mode will be provided as a backup solution in the event of system failure.

The SVMU system will be accompanied by a comprehensive website, designed to maintain a daily log of passing vehicles, vehicle inspections, records of guards on duty including their respective schedules and detailed information, and facilitate the registration of new vehicles. Furthermore, a real-time hardware implementation will be showcased to provide a practical demonstration of the SVMU system's functionalities. To develop the proposed system, the YOLOv5 algorithm will be utilized for efficient license plate detection. The web interface will be built using the MERN stack, ensuring a seamless user experience. OpenCV, a versatile computer vision library, will be employed for effective image processing, enabling real-time analysis of the captured images.



**Figure 1.1: Proposed System**

The SVMU system offers a reliable and user-friendly solution for intelligent vehicle management. Its integration of AI-based license plate detection, a well-designed website, and efficient hardware implementation ensures the smooth operation of access control. By providing features such as vehicle data logging, registration management, and manual operation options, SVMU has the potential to revolutionize vehicle access management, significantly enhancing security and convenience for both users and administrators.

# Relevance to Course Modules

The primary objective of undertaking a Final Year Project (FYP) by students is to showcase the acquisition and application of skills and knowledge gained throughout their academic tenure. This endeavor serves as a comprehensive demonstration of their abilities in integrating and utilizing various subject areas and courses covered within their university curriculum. Students employ diverse tools, technologies, and methodologies to address real-world scenarios during the development of their FYP.

During our degree program, we have been exposed to various courses that have contributed significantly to our knowledge and skillset. Among these courses are website development, database management, and artificial intelligence, which have proven to be particularly relevant and beneficial to our proposed project.

* **Website Development:**

The planned system will have a user interface that will maintain log and will do live screening of the vehicles.

* **Artificial Intelligence:**

The project is an artificial based vehicle license plate detection system that will detect the number plate using YOLO technique. We have studied this course and it turned out helpful for us.

* **Database Administration:**

The project also need a database to store the number plates of the vehicles for that purpose we need to integrate and operate database that learned in database course.

* **Technical and business writing:**

We have to make proposals and reports for final year project the Technical and business writing taught us how to design use case diagrams and reports.

* **Software engineering**

This course educated us about how a project should be developed, the methodology, requirements gathering and the model we should use.

The courses we have studied are ascertained relatively helpful in our final year project because we are quite familiar with the techniques we should know to do this project.

# Project Background

The implementation of a vehicle management system at UAAR has emerged as a crucial requirement for effectively managing vehicle entry and registration processes. Currently, the university entrance lacks a comprehensive system to address these needs, necessitating the development of a tailored solution.

The proposed system aims to automate the authorization process for vehicles by leveraging advanced license plate recognition technology. This automation will streamline the traffic flow within the university premises, resulting in enhanced operational efficiency and a reduced risk of errors. Additionally, the system will maintain comprehensive records of all vehicle-related activities, requiring minimal manual efforts from the security personnel.

One of the significant challenges faced by the university is unauthorized entry of visitors who park their vehicles within the campus without proper registration. This poses security risks and disrupts the smooth functioning of the university. The planned system will address this problem by providing a robust solution that ensures only authorized vehicles gain access to the premises. Unauthorized vehicles will be promptly detected, preventing potential security breaches.

The proposed solution will be specifically designed and deployed at UAAR to mitigate these risks and enhance overall security measures. By automating the vehicle authorization process, the system will alleviate the burden on security guards and improve their efficiency. Furthermore, the system will maintain a detailed daily log, enabling administrators to monitor and track vehicle activities effectively.

The implementation of a vehicle management system at UAAR is vital to address the challenges faced in managing vehicle entry and registration processes. By automating authorization, controlling traffic flow, minimizing errors, and maintaining comprehensive records, the proposed system will enhance security measures, reduce the burden on security personnel, and facilitate efficient management of vehicle-related activities within the university premises.

# Literature Review

Many researches have been done in the past years, and many vehicle license plate detection systems have been implemented few of them are stated below.

Rehman et al. in [1] has worked on the automatic number plate recognition. A convolutional neural work based in the MD-YOLO framework for the multi-directional car license number plate detection has been proposed in this work. This study employs a rapid intersection-over-union assessment and a meticulous rotation angle prediction approach. The experimental outcomes demonstrate that the proposed methodology attained commendable performance in relation to both computational efficiency and precision.

Vehicle identification has been active research for over the last few years. Several research studies have been conducted to ascertain the classification of vehicles, encompassing cars, trucks, scooters, and motorcycles. This particular paper employed the Soble filter as a means to address this matter by detecting the edges of the vehicle, subsequently enabling the identification of the vehicle type. The Contourlet Transform and Support Vector Machine (SVM) were employed to determine the vehicle model. The authors presented numerical results based on a dataset comprising approximately 70 images. However, the technique was not applied to real-time video streams, and vehicle recognition solely relied on monocular images. The researchers utilized canny edge detection to identify the presence of a vehicle and SVM for vehicle recognition. In this study, researchers utilized the Maximum Average Correlation Height (MACH) filter and Log r-theta Mapping techniques to accurately identify vehicle types, regardless of variations in scale and rotation. The MACH filter was employed to effectively detect targets within complex backgrounds and cluttered environments. By leveraging MACH, the researchers achieved orientation invariance, enabling the recognition of targets from any orientation. Additionally, they incorporated log r-theta mapping to establish in-plane rotation and scale invariance, enhancing the overall recognition capabilities of the system. [2]

Employing advanced Neural Network architectures, a cutting-edge system was developed using a publicly available dataset comprising Brazilian license plates. Impressively, during rigorous testing, the system showcased a notable accuracy rate of 63.1% in accurately detecting and recognizing the complete seven characters of license plates. Moreover, the system demonstrated exceptional proficiency in segmenting characters, achieving an impressive accuracy of 98% in character recognition while accurately segmenting 99% of the characters.[3]

# Analysis from Literature Review

The literature review of existing systems revealed certain limitations that our proposed system aims to address. One notable difference between the existing systems and our solution is the lack of comprehensive vehicle log maintenance and a dedicated web interface.

Unlike previous systems, our proposed system will incorporate a web application that efficiently maintains and organizes all relevant data and details. This includes keeping track of vehicles, drivers, and security guards on duty, which is a unique feature not present in other existing systems. The web interface will serve as a user-friendly platform accessible from desktop computers or smartphones, ensuring convenient access to the system's functionalities.

Through the web interface, administrators will have access to a wide range of information, including the number of vehicles that have entered the premises, driver details, vehicle screening records, and the ability to register new entries. This comprehensive data management capability sets our proposed system apart from existing alternatives, providing administrators with a centralized and easily accessible platform to monitor and track vehicle-related activities.

The proposed project will utilize the YOLOv5 algorithm, known for its exceptional accuracy and real-time object detection capabilities. Compared to other state-of-the-art models like Cascade-Mask R-CNN, YOLOv5 achieves higher accuracy while significantly improving inference speed by 509%.

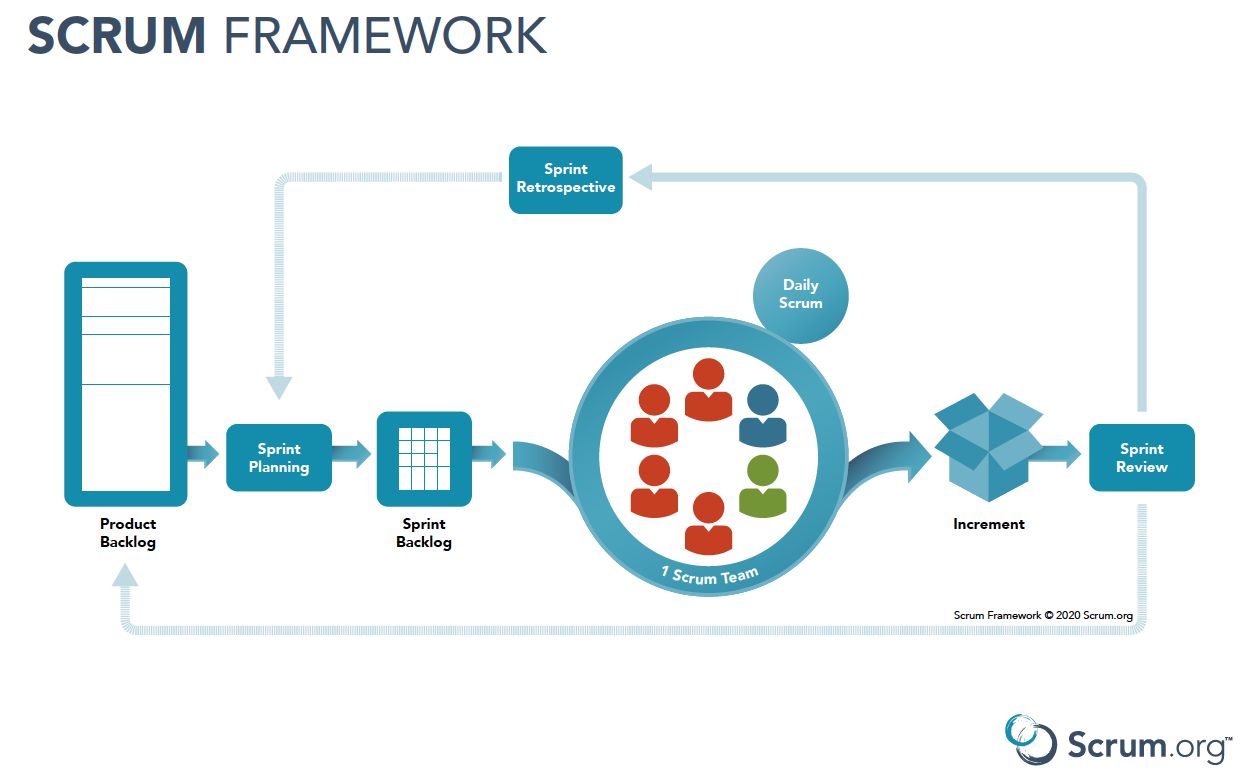
In the YOLOv5 model, image frames are processed through a backbone network, capturing essential features. These features are then combined and processed in the network's neck section before being passed to the head. The YOLO component of the model predicts the locations and classes of objects, enabling the drawing of bounding boxes around them. In addition to object detection, the proposed project incorporates a CNN model for segmenting the cropped number plate. The CNN model generates a sequence of recognized characters, which are further processed and ordered before being saved into the system's records. This integration of YOLOv5 and CNN ensures accurate and efficient license plate recognition within the project's framework.

This proposed system overcomes the limitations identified in the existing systems discussed in the literature review. By introducing a web application with advanced data management capabilities, including the maintenance of detailed logs, comprehensive driver and vehicle information, and the ability to track security guards' duty schedules, our solution offers a superior level of functionality and convenience. The cross-platform nature of the web interface ensures accessibility from a variety of devices, making it a versatile and efficient tool for managing vehicle access and security.

# Methodology and Software Lifecycle for This Project

The **Scrum development methodology** has been chosen for the implementation of this project. Scrum is a recognized framework that promotes efficient and accelerated product development. It follows an iterative and incremental approach, where goals are set to be completed within specific time frames known as sprints.

Under the Scrum methodology, each sprint typically spans two to four weeks. The primary objective of each sprint is to prioritize and develop the most critical features, ensuring that a potentially deliverable product is produced at the end of each iteration. This iterative approach allows for flexibility and adaptability, as changes and adjustments can be readily incorporated throughout the development process.

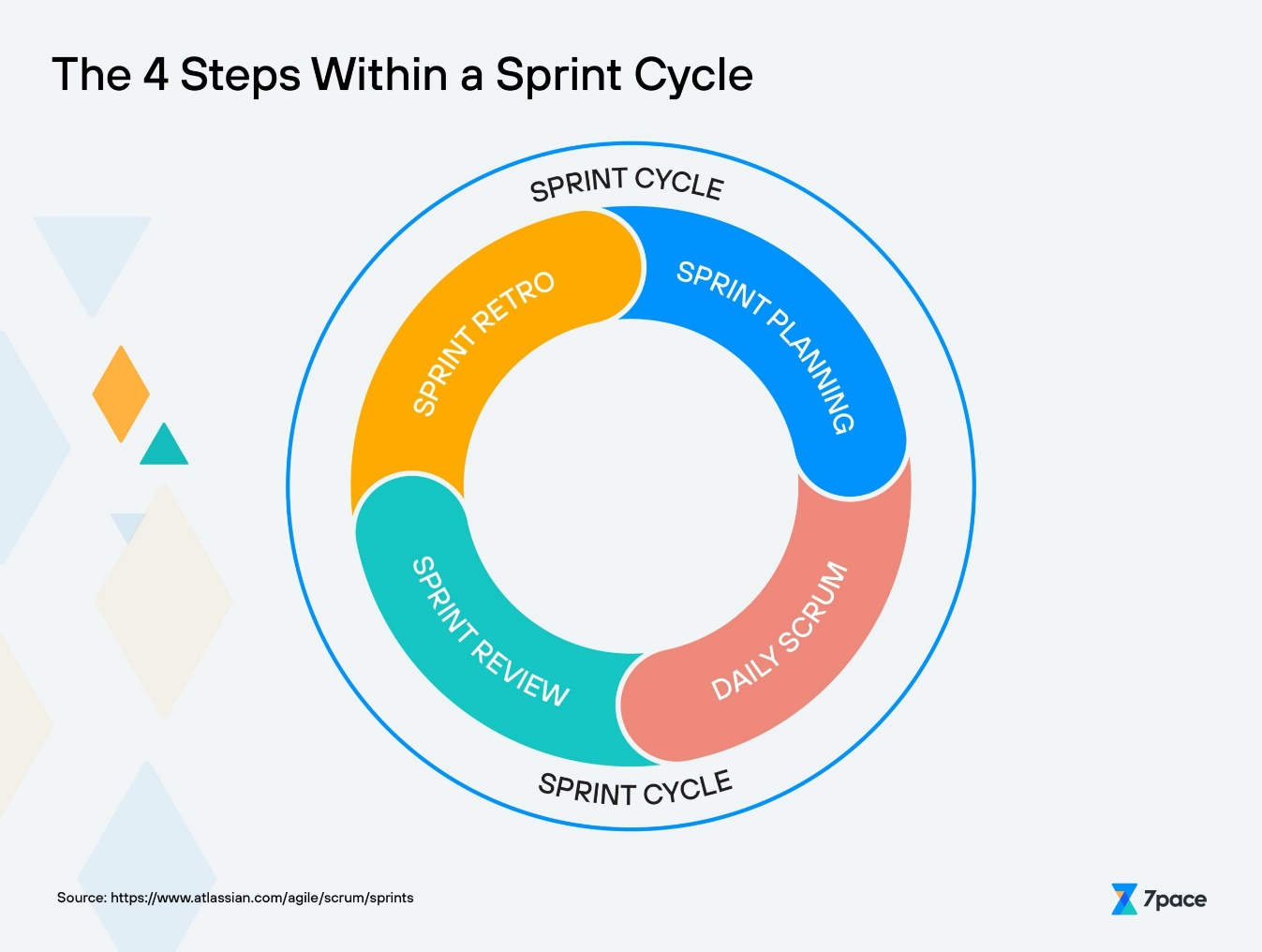


**Table 1.6 Scrum Model**

Scrum, being an agile methodology, aligns well with the project's requirements. Its emphasis on frequent checks, adaptability, and responsiveness enables effective management of unpredictable situations that may arise during the project's lifecycle. By embracing Scrum, the development team can ensure an agile and efficient development process, leading to the successful delivery of the project's objectives.

* + 1. Rationale behind Selected Methodology

The scrum makes the development process done timely because of the sprints, a goal must be completed within the time frame. The purpose to use scrum is to deliver everything on time and the project is divided into phases that results in ready-to-use product. At the end of each sprint a useable product is delivered to the customer by which we can take customer feedback.



**Table 1.6.1 Scrum Cycle**

**Chapter 2: Problem Definition**

# Problem Statement

The University of Arid Agriculture (UAAR) currently faces several challenges related to the management of parking slots and manual entry processes at its entrance gate. These challenges include the absence of a system to prevent unauthorized vehicles from occupying parking slots, as well as the reliance on manual procedures for vehicle checking and identification. At the entrance of UAAR, a substantial queue forms as vehicles undergo manual checking and identification. This process is not only time-consuming but also demanding for the personnel involved. Moreover, the manual nature of vehicle identification increases the likelihood of errors, posing potential security risks for the university. Another issue is the lack of comprehensive records regarding the number of vehicles entering and leaving the premises. The absence of accurate data on vehicle flow limits UAAR's ability to effectively manage and monitor traffic patterns and occupancy rates.

Furthermore, the current manual registration system for visitors proves to be inefficient and time-consuming. The process entails gathering personal information, followed by recording vehicle details and assigning parking slots. This sequential approach not only consumes the visitor's time but also contributes to traffic congestion at the entrance gate. The security guards responsible for managing the registration process also face challenges in effectively executing their duties. UAAR faces significant problems including the presence of unauthorized vehicles occupying parking slots, the time-consuming and demanding nature of manual vehicle checking and identification, the absence of accurate records for vehicle flow, and the inefficiency and traffic congestion caused by the manual visitor registration process. Addressing these issues requires the implementation of an automated and streamlined vehicle management system at UAAR.

# Scope

The scope of SVMU is limited to the premises of UAAR (University of Agricultural and Applied Sciences) and focuses primarily on vehicle number plate identification and access control. The proposed system offers the following functionalities:

* **Vehicle Identification:** The system is designed to accurately identify the vehicle number plates of registered vehicles entering the premises.
* **Access Control:** Upon successful vehicle registration, the system will signal the hardware to open the barrier, allowing authorized vehicles to enter. Simultaneously, the vehicle and driver details will be displayed on the screen for verification.
* **Unauthorized Vehicle Handling:** If a vehicle is not registered in the system, it will be deemed unauthorized. In such cases, the system will signal the hardware to display a message on the web application, notifying the driver that registration is required to gain access to the premises.
* **Log Maintenance:** The system will maintain a comprehensive log of all vehicles entering and exiting the premises. This log will be stored on the website and can be accessed to retrieve historical data.
* **Information Display:** The website associated with SVMU will display detailed information regarding each vehicle and its associated driver. This includes vehicle details, such as make, model, and registration number, as well as driver information, where available.
* **Registration Process:** During the registration process, all relevant information about the vehicle and driver will be gathered and stored in the system for future reference and verification.

It is important to note that the system will solely focus on identifying the vehicle number plate and will not include driver identification. This decision is based on considerations of budget limitations and the need for additional high-quality cameras, as driver identification can pose various challenges. SVMU provides a comprehensive solution for smart vehicle management within the UAAR premises. Its functionalities encompass vehicle number plate identification, access control, registration, log maintenance, and information display.

# Deliverables and Development Requirements

# Deliverable

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Phases** | **Deliverables** | **Estimated Days** |
| **Project**  **Beginning** | Selection of project, Collection of relevant information. | Project Proposal | 20/09/2022 |
| **Requirement Gathering** | Elicitation |  |  |
| Validation |  |  |
| Specification |  |  |
| Verification |  |  |
|  | Result Report | 31/10/2022 |
| **Requirement Analysis** | Identifying Requirements |  |  |
| Defining Boundaries |  |  |
| Establishing Constraints |  |  |
|  | SRS | 28/11/2022 |
| **Design** | Architectural Design |  |  |
| Interface Design |  |  |
| Component Design |  |  |
|  | UML Diagrams | 19/12/2022 |
| **Development** | Coding |  |  |
| Integration |  |  |
|  | Implemented System | 30/04/2023 |
| **Testing** | Unit Testing |  | 15 days |
| Integration Testing |  | 8 days |
| System Testing |  | 6 days |
| Acceptance Testing |  | 3 days |
|  |  | Complete System | 01/06/2023 |

**Table 2.2.2 Deliverables**

**2.2.2. Development Requirements**

This section will include all the system requirements without which we cannot develop the system.

|  |  |  |
| --- | --- | --- |
| **Category** | **Requirements** | **Version** |
| **Hardware** | Personal Computer or Laptop | - |
| Arduino Uno Board-1 | - |
| Servo Motor-1 | - |
| Camera | - |
| Cables | - |
| **Software** | **Tools** |  |
| Visual Studio Code | 1.71 |
| Google Colab | - |
| Arduino IDE | - |
|  | - |
| **Technologies** |  |
| React Js | 0.70 |
| Express Js | - |
| Node Js | 3.10.6 |
| MongoDB | - |
| Python | - |
| Optical Character Recognition OCR | - |
| OpenCV | - |

**Table 2.2.2 Tools and Technologies**

# Operating Environment

The operating environment for SVMU encompasses the necessary hardware and software components required for the system to function effectively within the UAAR (University of Arid Agriculture) premises. The key elements of the operating environment include:

**Hardware:**

* **Cameras:** Cameras capable of capturing clear images of vehicle number plates will be strategically installed at entry and exit points.
* **Barrier Control Hardware:** Hardware components responsible for controlling the entry barrier will be integrated with the system to facilitate automated access control.
* **Display Screens:** Screens or monitors will be set up to display vehicle and driver details for verification purposes.

**Software:**

* **Operating System:** A suitable operating system, such as Windows or Linux, will be deployed on the servers to support the SVMU software.
* **Application Software:** The SVMU application software, developed using appropriate programming languages and frameworks, will provide the necessary functionality for vehicle number plate identification, access control, registration, log maintenance, and information display.
* **Database Management System:** A reliable database management system (DBMS) will be utilized to store and manage the vehicle and driver information, as well as the log of entered and exited vehicles.
* **Web Application:** A user-friendly web application will be developed to provide access to the SVMU system, allowing guards and authorized personnel to perform registration, view vehicle and driver details, and access log records.

It is essential to establish a reliable and secure operating environment for SVMU to ensure the smooth functioning of the system, accurate vehicle identification, and efficient management of vehicle access within the UAAR premises.

# Assumptions and Dependencies

**Assumptions:**

* **Vehicle Registration:** It is assumed that all vehicles accessing the UAAR premises will undergo a registration process. The system relies on accurate and up-to-date vehicle information provided during the registration process.
* **Number Plate Visibility:** The system assumes that the vehicle number plates will be clearly visible and readable by the installed cameras. Adequate lighting conditions and unobstructed views of the number plates are necessary for accurate identification.
* **Authorized Personnel:** The assumption is made that guards or authorized personnel will be available to manage the SVMU system, perform registration, handle exceptions, and oversee the access control process.

**Dependencies:**

* **Hardware Functionality:** The SVMU system depends on the proper functioning of the hardware components, including servers, cameras, barrier control mechanisms, and display screens. Any hardware failures or malfunctions may impact the system's performance.
* **Software Development:** The successful development and deployment of the SVMU system depend on the availability of skilled software developers, adherence to development best practices, and efficient software testing processes.
* **Database Management System:** The system relies on the proper functioning of the selected database management system (DBMS) to store and manage vehicle and driver information, as well as the log records. The DBMS should be reliable, scalable, and capable of handling the expected data load.
* **User Cooperation:** The effectiveness of the SVMU system depends on the cooperation of vehicle owners and drivers in providing accurate and up-to-date information during the registration process. Any inaccuracies or incomplete data provided may affect the system's ability to identify vehicles correctly.
* **Security Measures:** The SVMU system depends on the implementation of appropriate security measures to protect sensitive data and ensure system integrity. The effectiveness of these security measures, including data encryption, user authentication, and access control, is crucial for maintaining system security.

It is important to consider these assumptions and dependencies during the planning, development, and implementation stages of SVMU to ensure its successful operation within the UAAR premises.

# Current System

The implementation of vehicle number plate detection systems has been prevalent in various foreign countries; however, in Pakistan, such systems are not yet in place. The existing systems in other countries do offer certain functionalities, but they also possess several limitations. These systems primarily capture the image of a vehicle within the camera's range and locate the number plate within the image. Subsequently, the captured license plate is transformed into textual format. Subsequently, the obtained number is compared against a database containing records of authorized vehicles. While these existing systems fulfill the basic requirements of number plate detection, they lack a user-friendly interface. In the context of this project, we will discuss two existing systems and the technologies they employ. By analyzing these systems, we aim to identify their strengths and weaknesses, which will help inform the design and development of our proposed system.

**ANPR system China:**

Hui Wu and Bing Li [4] presented a method for locating Chinese number plates, which involves utilizing horizontal and vertical differences to identify the precise rectangle containing the vehicle number. In their study, the Authors converted vehicle images to grayscale and subsequently employed MATLAB to perform automatic binarization. However, no additional information regarding the specific algorithm used for number plate detection was provided in the paper. The authors assert an average recognition rate of 0.8 seconds.



**Figure 2.1: China ANPR**

**ANPR system India:**

Ch. Jaya Lakshmi et al. [5] proposed a novel approach for extracting license plate characters in the specific conditions found in India. Their method relied on texture characteristics and wavelets. Additionally, the authors incorporated morphological operations to improve performance in complex background scenarios. The vertical edges were detected using a Sobel mask. The implementation of the system was carried out using MATLAB. Furthermore, the authors utilized a Sobel edge detection operator in their approach.[6].



**Figure 2.2: India ANPR**

**Chapter 3: Requirement Analysis**

# Use Cases

In the context of the Unified Modeling Language (UML), a use case diagram serves as a visual representation that encapsulates pertinent information regarding the actors involved in our system, as well as their interactions with the system itself. This diagram provides a concise summary of the roles played by users within the system, often referred to as actors, and illustrates their various interactions and relationships with the system's functionalities and services.

### **Use Case Diagram:**



**Figure 3.1: Use Case Diagram**

### **Use Case Description:**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-01 |
| **Use Case Name:** | Vehicle Registration |
| **Actors:** | Security guard, System |
| **Description:** | Visitor will provide the relevant details of vehicle to register in the database. |
| **Trigger:** | When user clicks on Register button. |
| **Preconditions:** | Vehicle details must be provided by the visitor. |
| **Post conditions:** | Vehicle will be registered successfully. |
| **Normal Flow:** | Vehicle details will be provided by the visitor.  Security guard will enter all the details.  Vehicle will be successfully registered. |
| **Alternative Flows:** | None |
| **Exceptions:** | When system failed to register the vehicle details in the database. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | User should be already registered. |
| **Notes and Issues:** | None |

**Table 3.1: Vehicle Register**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-02 |
| **Use Case Name:** | Scan Vehicle Number Plate |
| **Actors:** | Security guard, AI System |
| **Description:** | Security guard will scan the number plate by the camera. |
| **Trigger:** | When user clicks on open camera button |
| **Preconditions:** | Vehicle must be in front of camera with clear and visible number plate. |
| **Post conditions:** | The vehicle number plate is accurately captured by the system. |
| **Normal Flow:** | Vehicle arrives in front of camera.  The security guard will scan the number plate.  The vehicle number is successfully captured by the camera. |
| **Alternative Flows:** | Vehicle arrives in front of camera.  Vehicle will pass without any checking in case of automated system does not work. |
| **Exceptions:** | When camera is failed to scan the vehicle number plate. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | User should be already registered. |
| **Notes and Issues:** | None |

**Table 3.2: Scan Vehicle Number Plate**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-03 |
| **Use Case Name:** | Displays Vehicle Details |
| **Actors:** | Security guard, Admin, System |
| **Description:** | Security guard can see the details of registered vehicles. |
| **Trigger:** | When user clicks on Display records button |
| **Preconditions:** | Vehicle must be registered in the database. |
| **Post conditions:** | All the related details of vehicle. |
| **Normal Flow:** | Security guard will open the website and will click on View vehicle details.  Details will be shown on the screen. |
| **Alternative Flows:** | None |
| **Exceptions:** | Database failed to provide the details. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | Vehicles are registered in the database. |
| **Notes and Issues:** | None |

**Table 3.3: Displays Vehicle Details**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-04 |
| **Use Case Name:** | Edit Vehicle Details |
| **Actors:** | Security guard, Admin, System |
| **Description:** | Security guard and admin can edit the details of registered vehicles. |
| **Trigger:** | When user clicks on edit records button. |
| **Preconditions:** | Vehicle must be registered in the database. |
| **Post conditions:** | Vehicle edited successfully. |
| **Normal Flow:** | User will open the website and will click on Edit vehicle details.  Details will be shown on the screen.  Update record of relevant vehicle and save. |
| **Alternative Flows:** | None |
| **Exceptions:** | Database failed to provide the details. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | Vehicle is registered in the database. |
| **Notes and Issues:** | None |

**Table 3.4: Edit Vehicle Details**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-05 |
| **Use Case Name:** | Delete Vehicle Details |
| **Actors:** | Admin, System |
| **Description:** | Admin can delete the registered vehicles. |
| **Trigger:** | When Admin clicks on Delete records button |
| **Preconditions:** | Vehicle must be registered in the database. |
| **Post conditions:** | Record deleted successfully. |
| **Normal Flow:** | Admin will open the website and will click on Delete vehicle details.  Details will be shown on the screen.  Delete the relevant vehicle and save. |
| **Alternative Flows:** | None |
| **Exceptions:** | Database failed to provide the details. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | Vehicle is registered in the database. |
| **Notes and Issues:** | None |

**Table 3.5: Delete Vehicle Details**

|  |  |
| --- | --- |
| **Use Case ID:** | ID-06 |
| **Use Case Name:** | Automatic Barricade Uplift |
| **Actors:** | Arduino, Servo-motors, System |
| **Description:** | Barrier will automatically be opened in case of successful authentication. |
| **Trigger:** | When user will be successfully authorized by the system. |
| **Preconditions:** | Vehicle must be registered in the database. |
| **Post conditions:** | Barricade will be uplift automatically for sake to pass the vehicle. |
| **Normal Flow:** | Vehicle arrives in front of camera.  Number plate scanning.  Automatic barricade uplift. |
| **Alternative Flows:** | Vehicle arrives in front of camera.  Automatic system failed.  Manual system to uplift the barricade. |
| **Exceptions:** | Barricade failed to uplift automatically. |
| **Special**  **Requirements:** | None |
| **Assumptions:** | System is working fine. |
| **Notes and Issues:** | None |

**Table 3.6: Automatic Barricade Uplift**

# Functional Requirements

There are a number of functional requirements that might be considered when designing a smart vehicle management system using an AI-based number plate recognition system. Some potential functional requirements might include:

* **Accurate and Reliable Number Plate Recognition**:

The AI-based number plate recognition system will be able to accurately and reliably identify vehicles based on their number plates. This may involve using advanced image recognition algorithms and machine learning techniques to accurately extract and interpret number plate data from images or video streams.

* **Real-time Processing:**

The will be able to process and analyze data in real-time, allowing it to respond to events and provide timely feedback or alerts as needed.

* **Database Integration:**

The website will be able to store and retrieve license plate numbers and other relevant information from a database.

* **User Authentication:**

The website will require users to log in with a username and password in order to monitor and access certain features or information.

* **Vehicle Registration:**

The website will be able to register new vehicle record in the database.

* **Search Functionality:**

The website should allow users to search for specific license plate numbers or other relevant information in the database.

* **Reporting Capabilities:**

The website may need to generate reports or other documents based on the information stored in the database.

* **User-Friendly Interface:**

The system will have a user-friendly interface that allows operators and administrators to easily access and manage the system, as well as view and analyze data and receive alerts and notifications.

* **Hardware Compatibility:**

The system should be compatible with the hardware components that are being used, such as the Arduino microcontroller and the barrier. This may involve using software libraries or other resources that are designed to work with these components.

* **Hardware Integration:**

The hardware components will be integrated with the AI-based number plate recognition system in a way that allows them to work together seamlessly. This may involve configuring the hardware components to communicate with the system using appropriate protocols, such as serial communication or Ethernet.

* **Hardware Monitoring and Control:**

The system should be able to monitor and control the hardware components as needed, such as activating the barrier to allow or prevent vehicle access, or triggering alarms or other alerts.

# 3.3. Non-Functional Requirements

* The system will be reliable and able to function effectively without errors or failures, in different challenging conditions.
* The system will be able to perform efficiently and respond quickly to events and requests, to ensure that it can effectively manage and monitor vehicles in real-time.
* The system will permit only authorize vehicles to enter in the university premises.
* Interface and the system itself should be user friendly so that the monitoring person will feel it easy to use.
* The system will be secure and protect against unauthorized access, data breaches, and other security threats.
* The system will be easy to maintain and update, with a clear and well-organized codebase and hardware design that is easy to understand and modify as needed.
* The system will authenticate the vehicle by verifying the credentials to database.

# Chapter 4: Design and Architecture

# System Architecture

The proposed system works on the **“Layered architecture”** there are three layers of the project so it works on three tier or three layered architectures. A three-tier architecture is a widely recognized software application architecture that effectively structures an application into three distinct logical and physical computing layers. These layers include the presentation layer, which pertains to the user interface of the application. The application layer, on the other hand, is responsible for processing the data within the application. Lastly, the data layer encompasses the storage and management of the data associated with the application.

**Front End Application Layer Back End**

** ** 

UserTerminal Server(Local machine)

# 

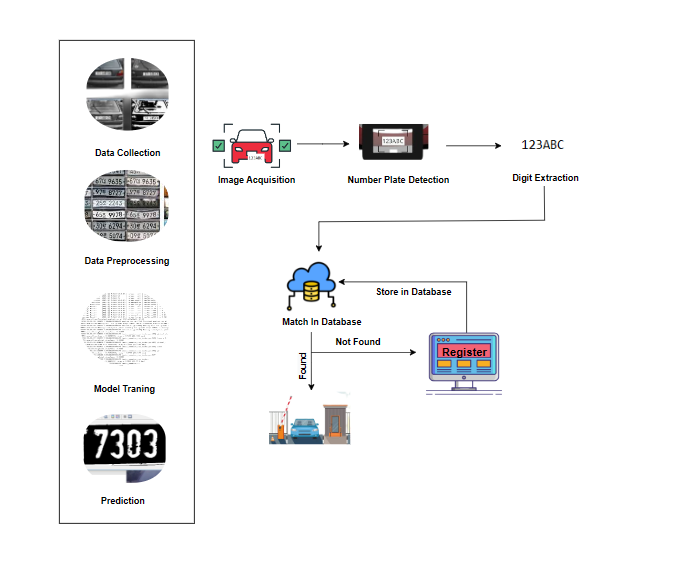
**Figure 4.1: System Architecture**

# System Design

The system design process encompasses the following key aspects:

* **Architecture:** This entails developing a conceptual model that outlines the structure, behavior, and different views of the system. It involves utilizing functional components and combining them to form the system's architecture.
* **Modules:** Modules refer to sub-components that handle specific tasks within the system. These modules are combined to form the functional components within our system.
* **Components:** Components are responsible for providing specific functions or groups of related functions. They are composed of modules. In our case, we employ functional components in React that consist of smaller modules.
* **Interfaces:** Interfaces serve as shared boundaries through which system components exchange information and establish relationships. To facilitate data sharing across components, we utilize props and the context API.
* **Data:** Data management involves handling the flow of information and data within the system. We retrieve data from a backend API, and it traverses across various components within the system.

The primary objective of the System Design process is to furnish comprehensive and detailed data and information regarding the system and its constituent elements. This information enables the consistent implementation of the system in accordance with the architectural entities outlined in the system architecture models and views.

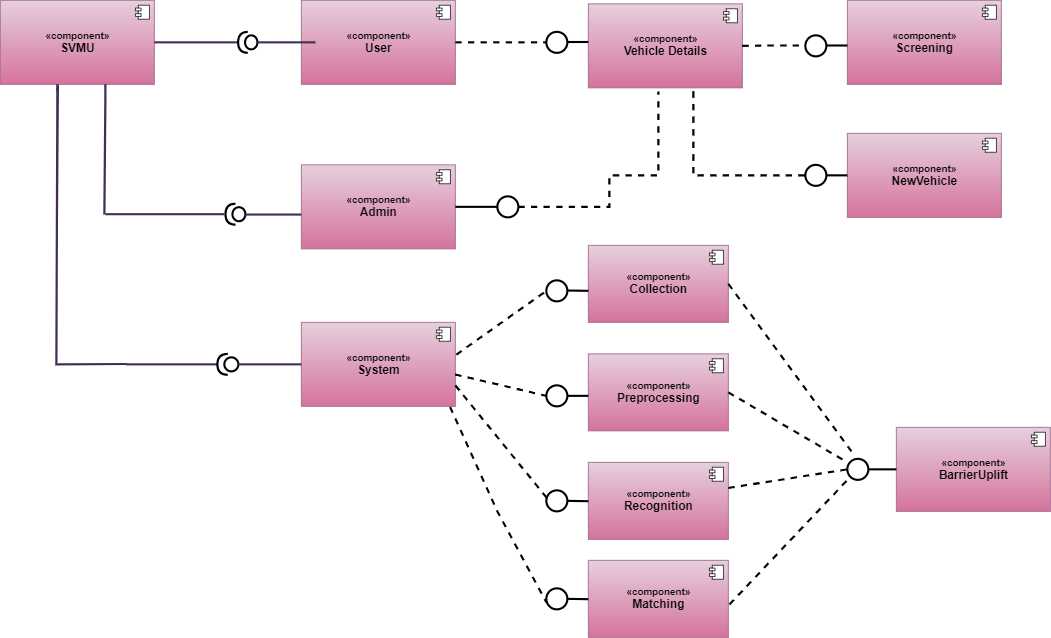


**Figure 4.2 System Design**

# UML structural diagram

# Component Diagram

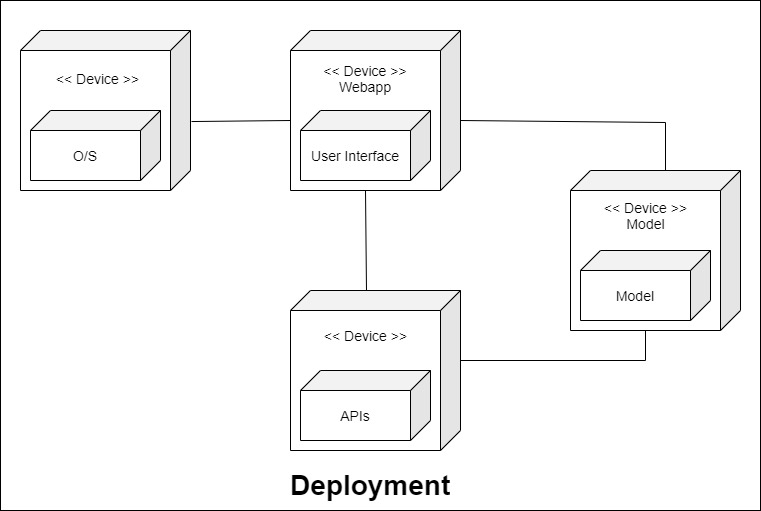
Component diagrams serve as valuable tools for modeling implementation details and ensuring that all necessary functions of a system are adequately addressed in the planned development. These diagrams, which are essentially class diagrams, specifically emphasize the system's components and are employed to represent the static implementation view of the system.



**Table 4.2.1.1: Component Diagram**

# Deployment Diagram

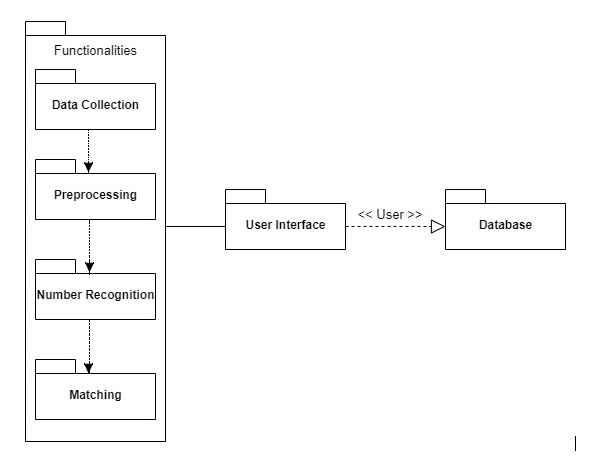
A UML deployment diagram is a graphical representation that illustrates the arrangement of runtime processing nodes and the corresponding components residing on them. These diagrams are classified as structure diagrams and are commonly employed to depict the physical attributes of an object-oriented system. Their primary purpose is to model the static deployment view of a system, specifically focusing on the hardware topology.



**Table 4.2.1.2: Deployment Diagram**

# Package Diagram

A package diagram, categorized as a structural diagram, depicts the organization and arrangement of model elements within middle to large-scale projects. It provides a visual representation of both the structure and dependencies existing between sub-systems or modules. Package diagrams enable the representation of various system views, such as the multi-layered (also known as multi-tiered) application model.

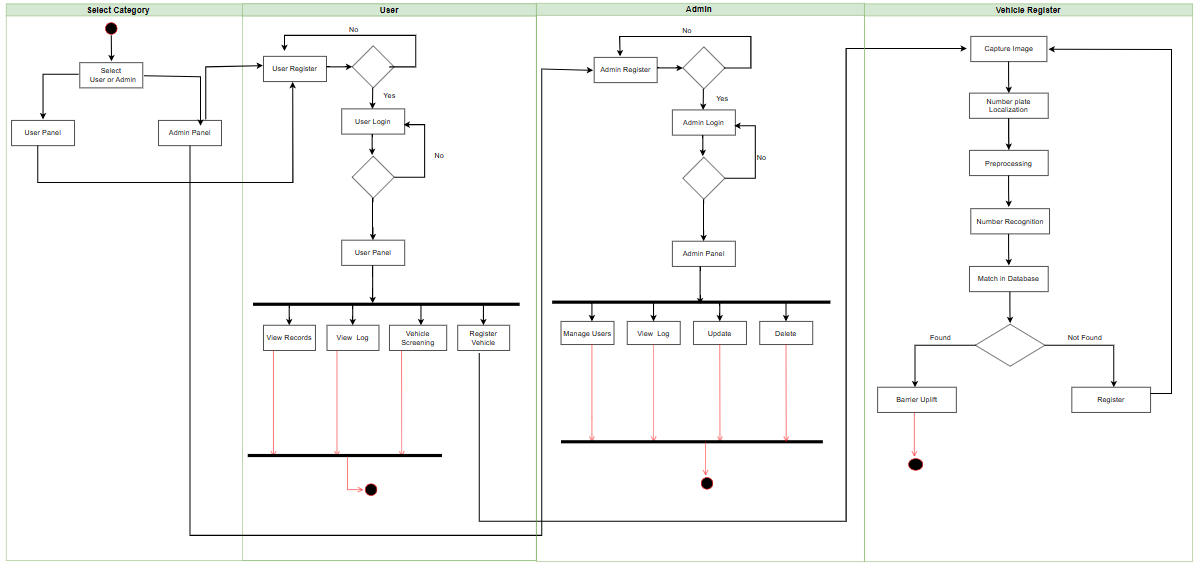


**Table 4.2.1.3: Package Diagram**

# 4.2.2 UML Behavioral Diagram

# 4.2.2.1 System Activity Diagram

An activity diagram, classified as a behavioral diagram within the UML diagram, is a crucial tool for describing the dynamic aspects of a system. It serves as an enhanced version of a flow chart, enabling the modeling of the flow from one activity to another.



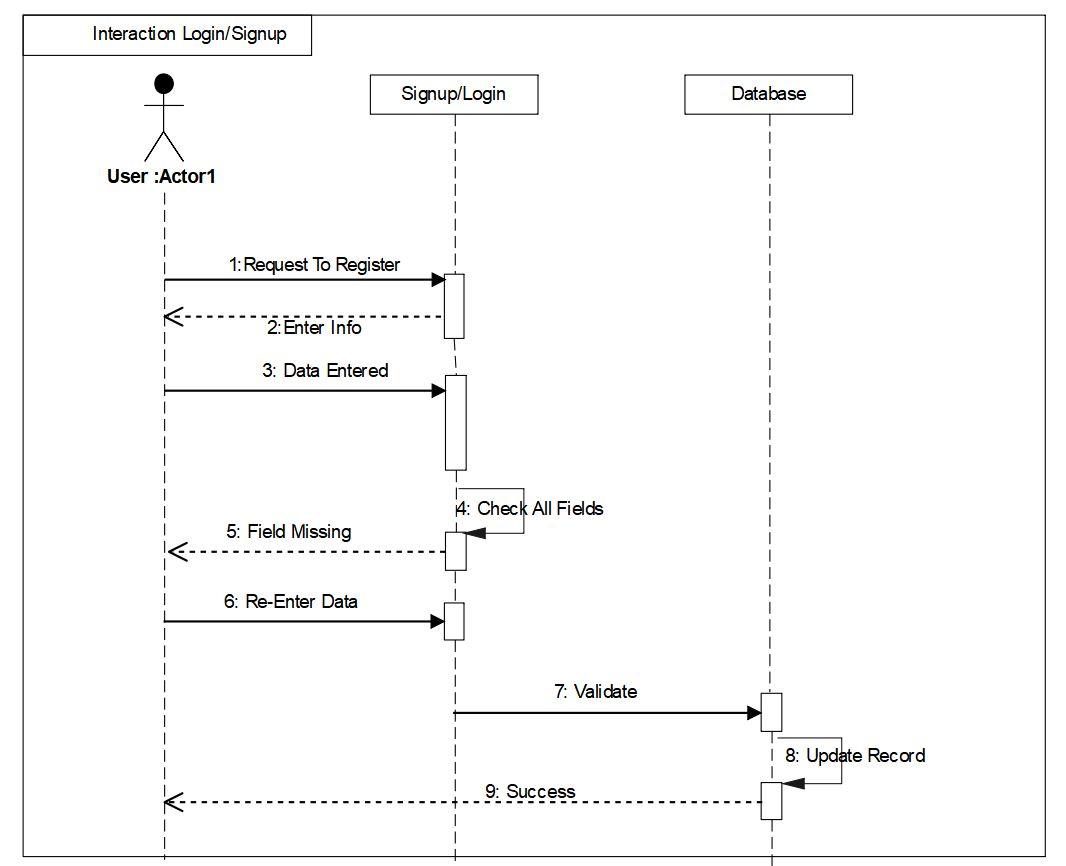
**Table 4.2.3: Activity Diagram**

# 4.2.3 UML Interaction Diagram

# 4.2.2.3 UML Sequence Diagram

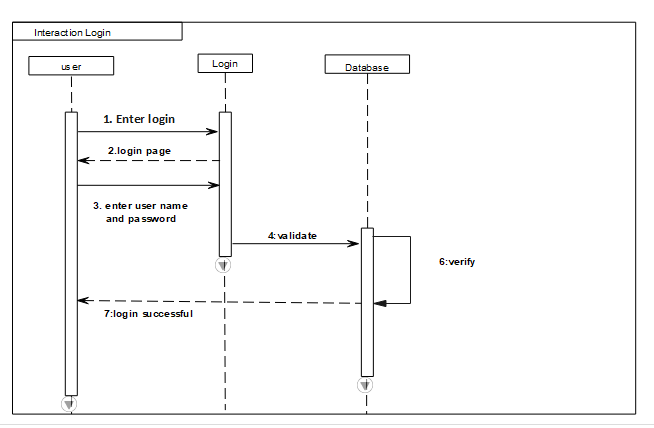
UML Sequence Diagrams are a type of interaction diagram that provide a comprehensive depiction of how operations are executed. These diagrams specifically capture the interaction among objects within the context of a collaboration. By employing a time-focused approach, sequence diagrams visually present the sequence of interactions, utilizing the vertical axis to represent time. They effectively illustrate the timing, order, and details of the messages exchanged between objects.

**Register:**

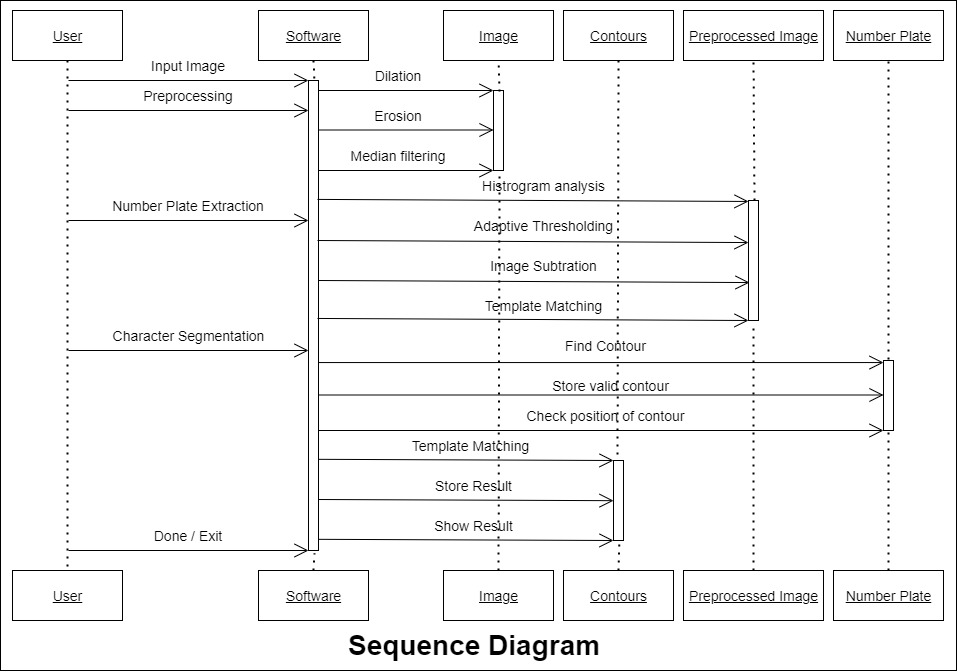


**Table 4.2.2.3: Sequence diagram Register**

**Login:**

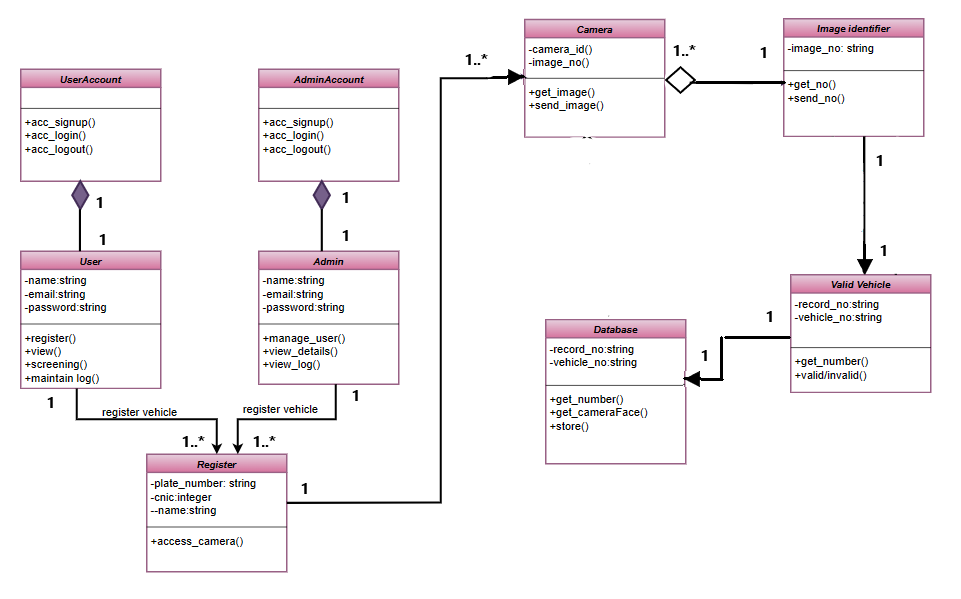


**Table 4.2.2.3: Sequence diagram Login**



**Table 4.2.2.3: Sequence Diagram System**

**Class Diagram:**



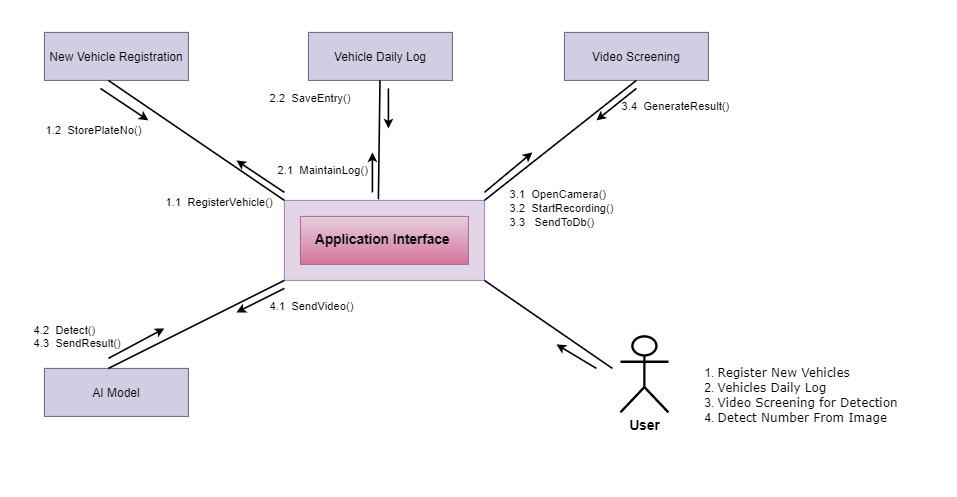
**Table 4.2.3.1: Class Diagram**

* + 1. **Node Structure**

**4.2.4.1 Communication Diagram**

A communication diagram is an interactive UML diagram that showcases how objects communicate and interact with each other. It is an extension of an object diagram, depicting both the objects involved and the messages exchanged between them. It provides a visual representation of the flow of messages among objects, enhancing the understanding of the system's dynamic behavior and the interactions between its components.

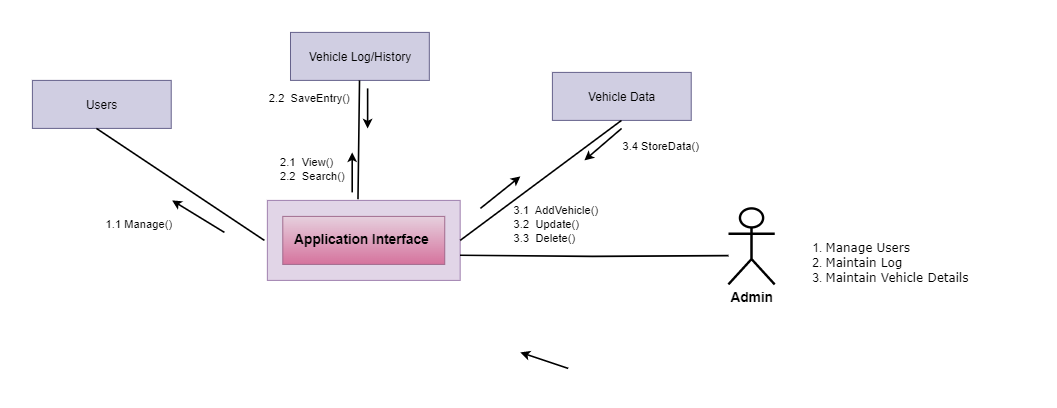
**User:**

****

**Table 4.2.4.3 Node Diagram**

**4.2.4.2 Communication Diagram**

**Admin:**

****

**Table 4.2.4.3 Node Diagram**

**Chapter 5: Implementation**

Implementation refers to the process of detailing the internal structure of a system, encompassing various approaches and programming styles that are adopted for its development. The specific implementation approach and programming style chosen depend on the feasibility and stability considerations of the system. The implementation phase serves as a means to transform the system design into an executable form, where the design specifications are translated into actual code and software components. This phase involves the practical realization of the system, ensuring that it aligns with the intended functionality and requirements outlined during the design phase.

# Component Diagram

Component Diagram for the SVMU system contained the below components: This component represents the client-side code that runs in a web browser. It typically includes HTML, CSS, and JavaScript code that defines the user interface and handles user interactions.

* **Interfaces/Front-End:**

This component represents all the client-side code, user and admin panels that runs on web browser. React.js was used for handling user interactions.

* **User-Device:**

Not typically a component but plays major part in the architecture where actually website would work. It can be laptop, tablets and mobile phones.

* **SVMU Server/ Backend:**

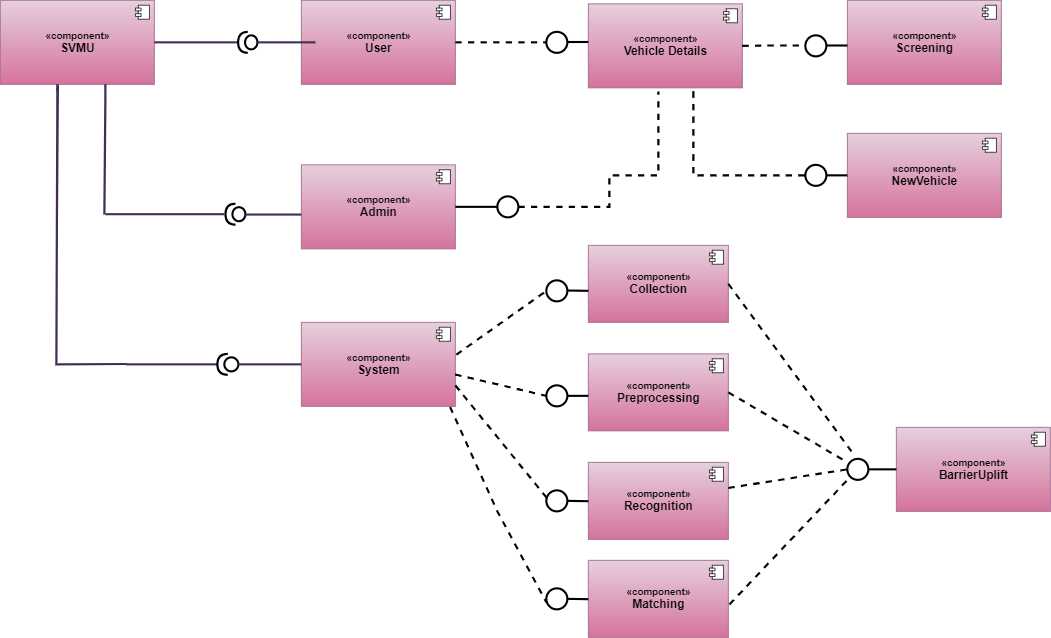
This component represents the server-side code that runs on the web server. It handles business logic, data storage, and communication with external systems. Backend code in SVMU is completed in Node.js

* **Database:**

This component represents the storage system used to store and retrieve data used by the web application. MongoDB was used for the database.

* **Data Processing:**

This component contained AI model that will process the image and detect the number plate.



**Table 5.1 Component Diagram**

# Network and Protocol Choice

The choice of network and protocol for SVMU is Web Sockets. The reasons are discussed in detail as under.

**Web Sockets:**

Web Sockets are employed in MERN stack websites to facilitate the creation of real-time features that demand continuous two-way communication between the client and server. This bi-directional connection can be initiated by either the client or the server, enabling real-time transmission and reception of data, thereby eliminating the need for repetitive requests or page refreshes. In the context of SVMU, Web Sockets enable the website to interact with the model for detection purposes. The utilization of Web Sockets in a MERN stack website depends on the specific requirements of the project and the necessity for real-time functionalities. In cases where real-time communication is not essential, conventional HTTP queries may suffice.

# Choice of Object Middleware

The middleware functions can be used to handle authentication and authorization tasks, such as checking user credentials, creating sessions, and handling access control. The middleware’s were not used because of the following reasons:

**RMI (Remote Method Invocation):**

RMI, which stands for Remote Method Invocation, serves as a middleware technology facilitating communication between Java objects deployed on separate machines. It allows for seamless interaction through Java Remote Method Invocation. However, considering that SVMU relies on MERN technology, which primarily employs JavaScript, RMI is not the most suitable choice for integration within this context.

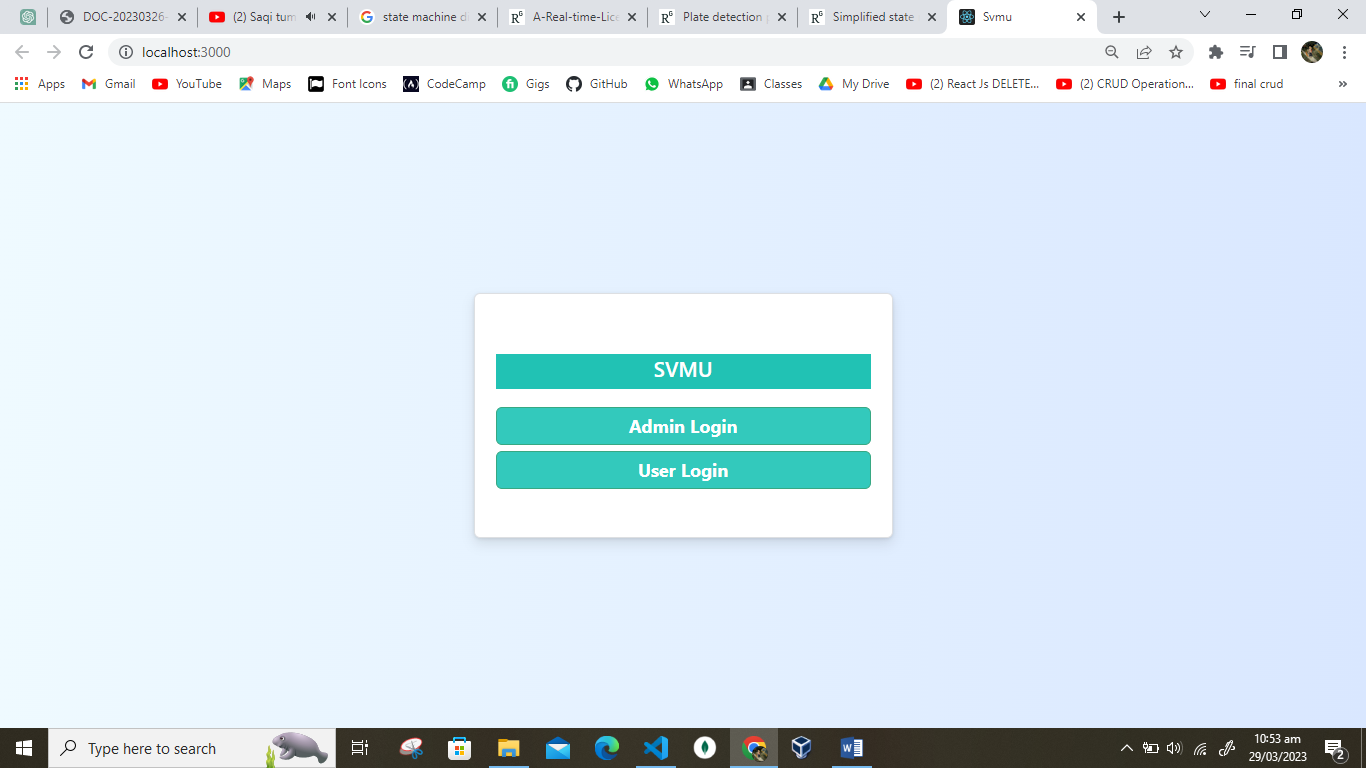
**CORBA (Common Object Request Broker Architecture):**

CORBA, an acronym for Common Object Request Broker Architecture, serves as a middleware technology designed to facilitate communication between objects implemented in various programming languages and operating on different platforms. However, due to the extensive configuration requirements and its declining popularity, CORBA is not as widely utilized as it once was. Its intricate nature, coupled with the absence of browser support, makes it less suitable for the development of SVMU.

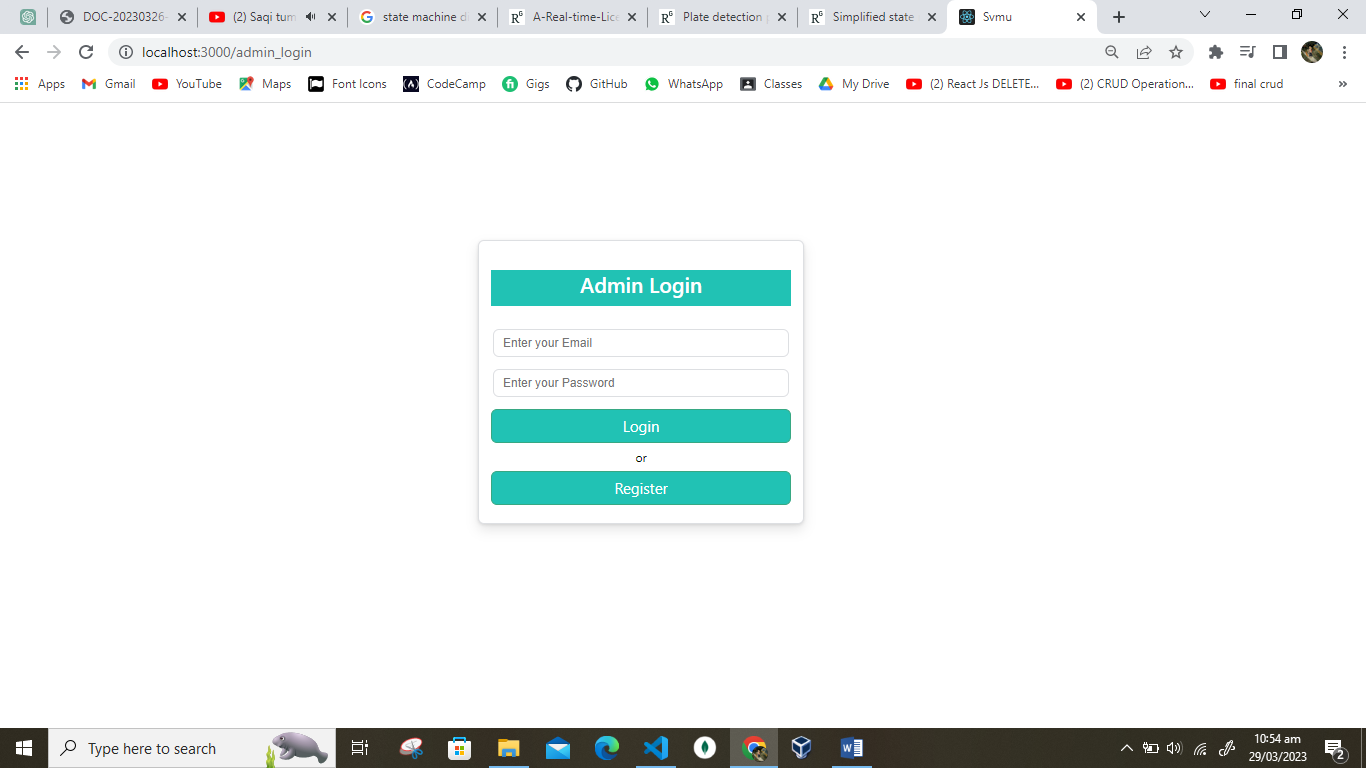
**DCOM (Distributed Component Object Model):**

DCOM, which stands for Distributed Component Object Model, is a technology developed by Microsoft that facilitates communication between objects deployed on separate machines within a Windows environment. However, it is important to note that DCOM is a proprietary technology and may not be fully compatible with non-Windows systems.

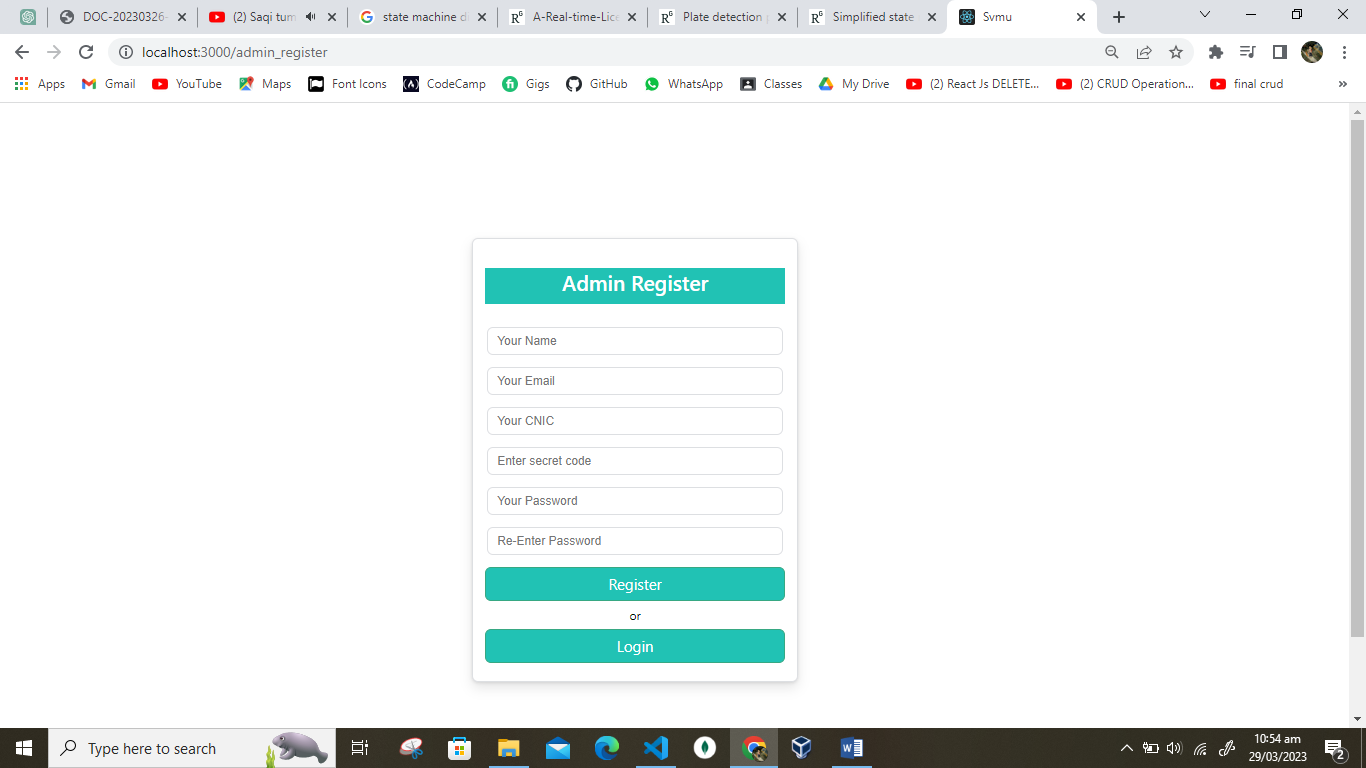
# User Interface



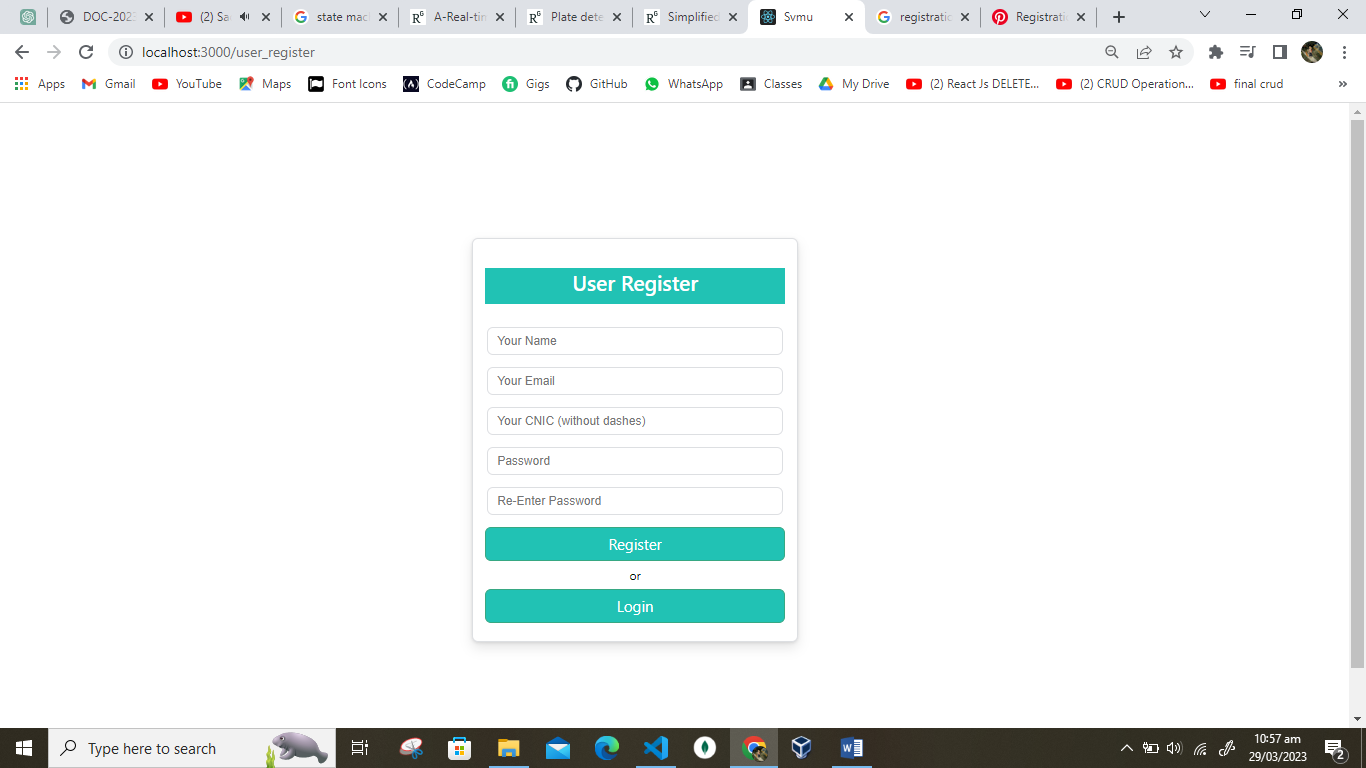
**Figure 5.4.1 SVMU**



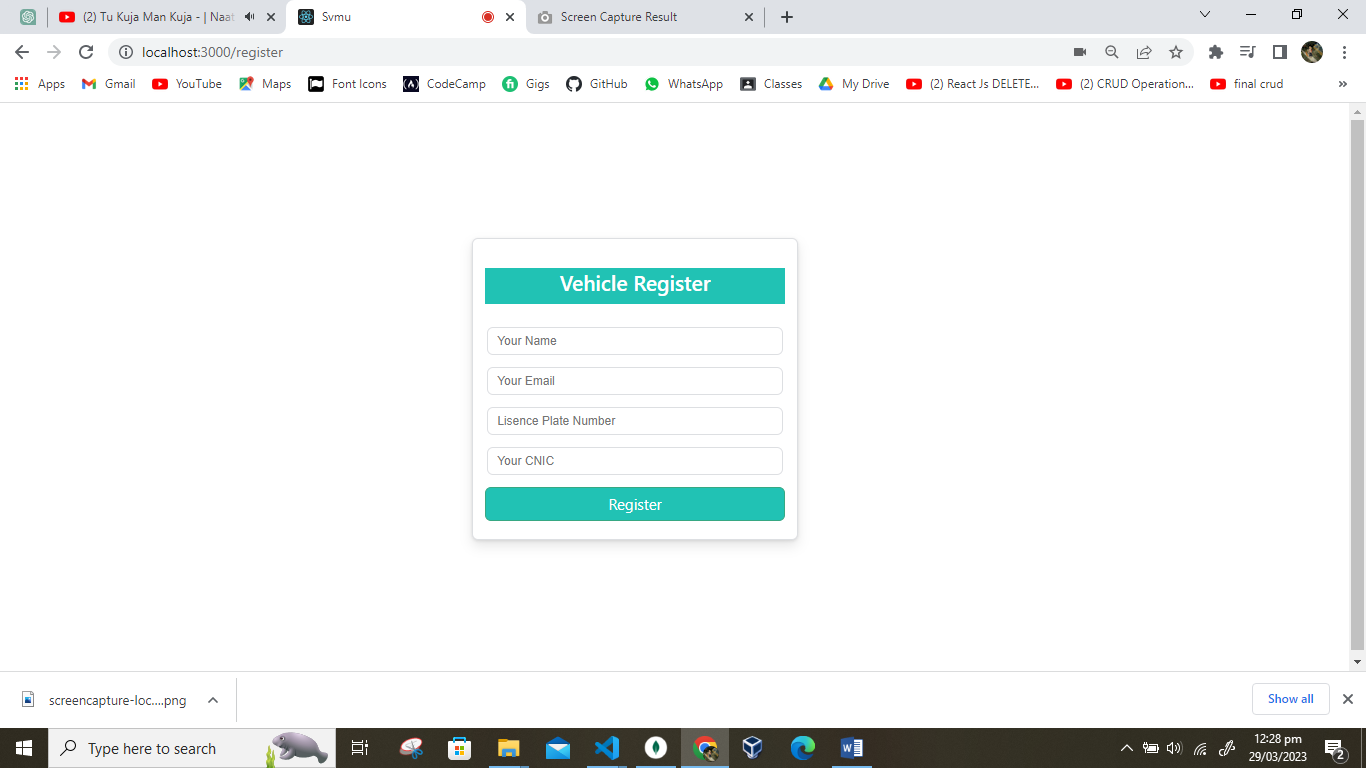
**Figure 5.1.2 Admin Login**



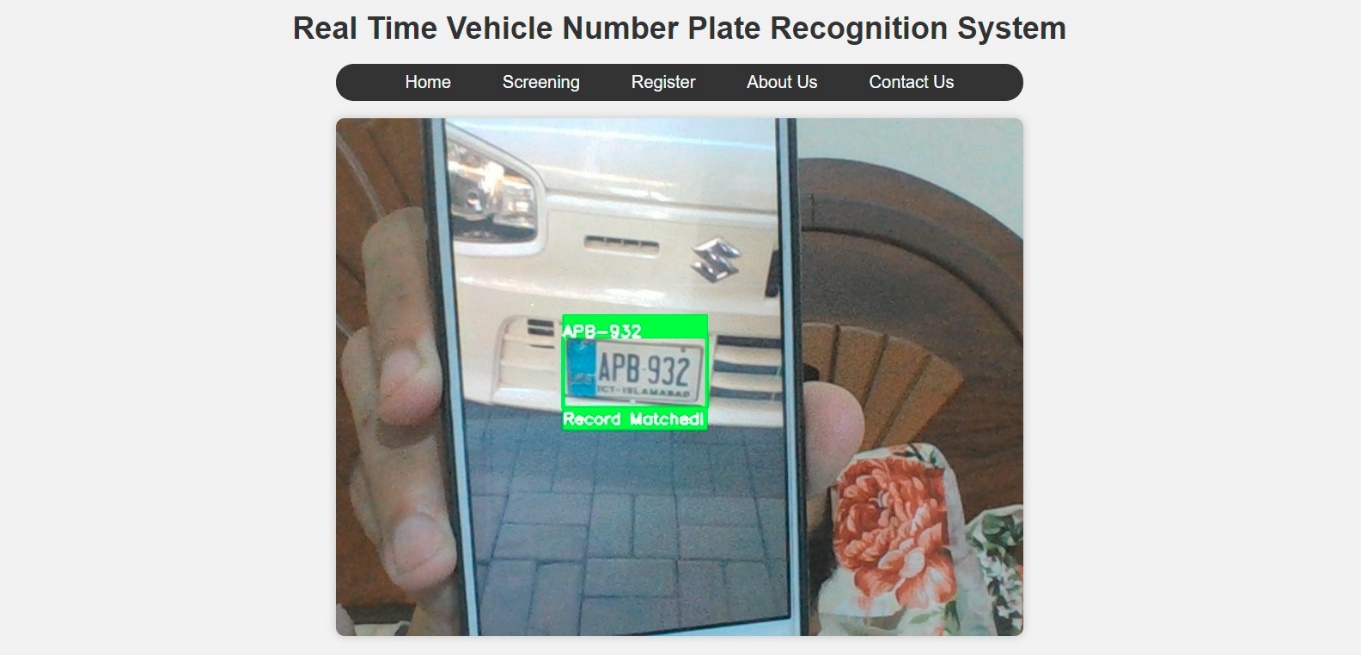
**Figure 5.4.3 Admin Register**



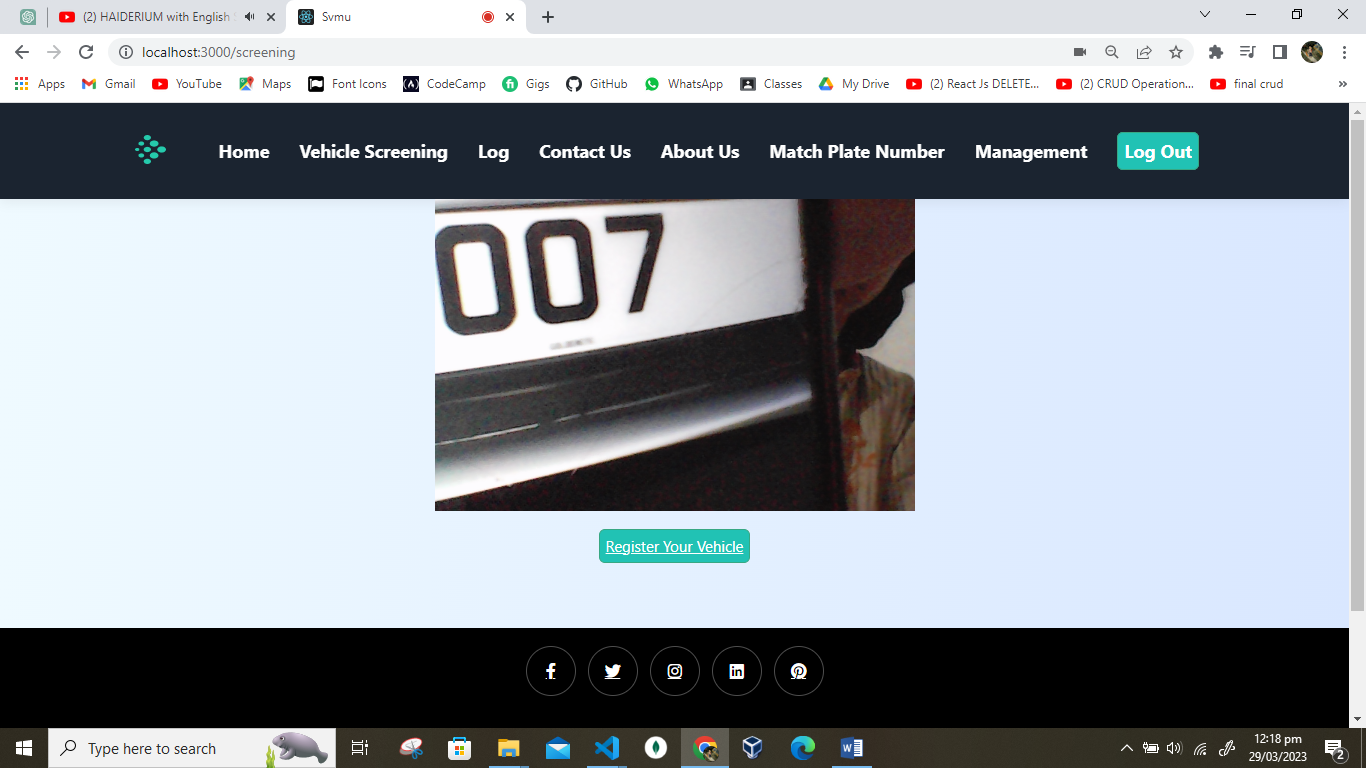
**Figure 5.4.4 User Register**



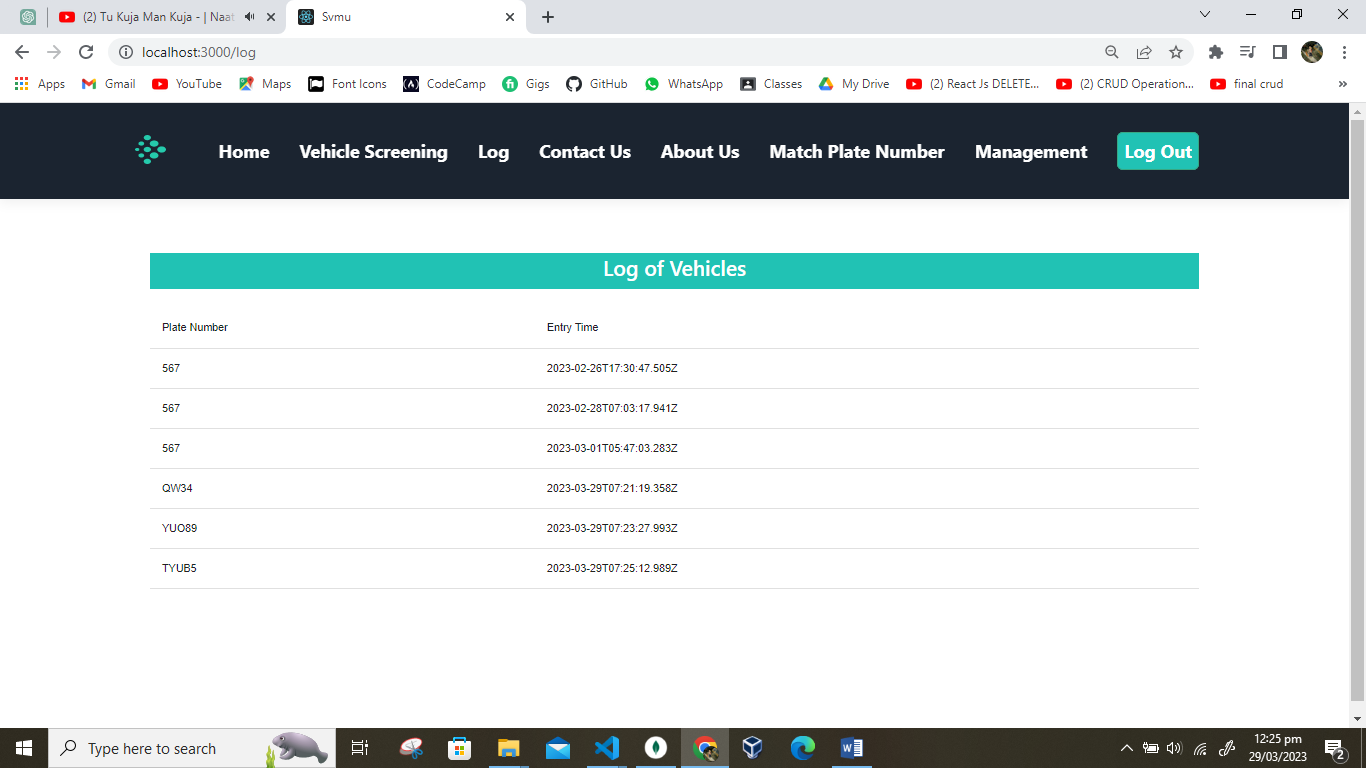
**Figure 5.1.5 Vehicle Register**



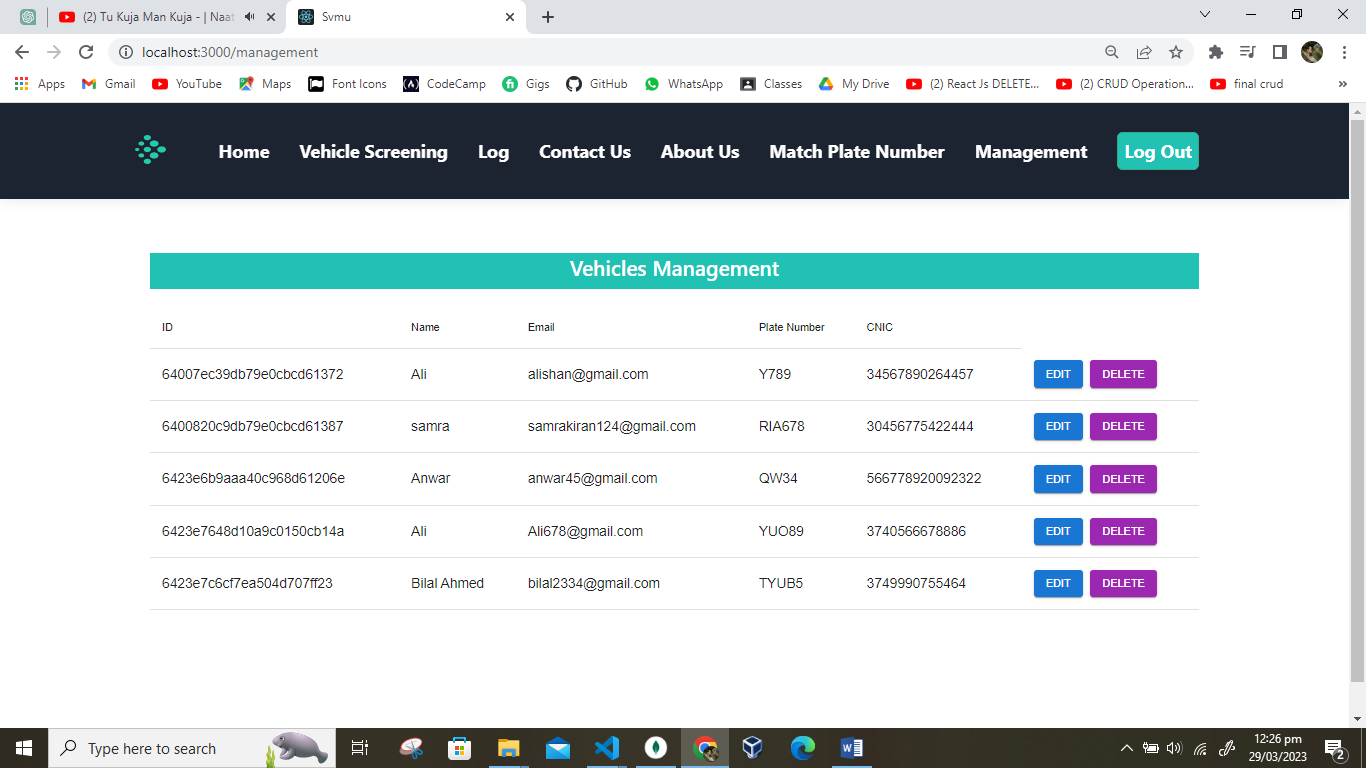
**Figure 5.1.6 SVMU**



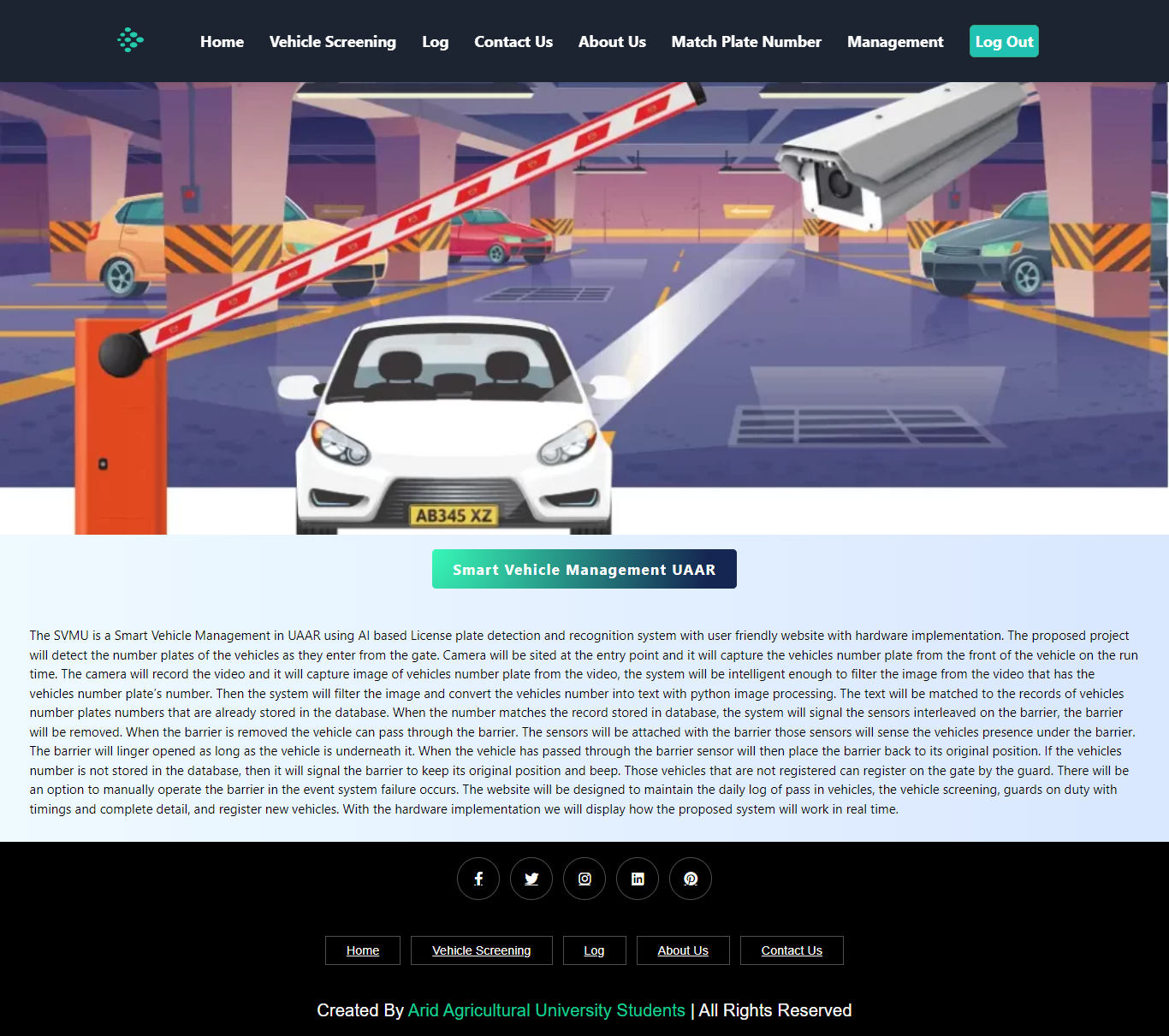
**Figure 5.1.7 Screening**



**Figure 5.1.8 Log**



**Figure 5.1.9 Management**



**Figure 5.1.10 Home**

# Chapter 6: Testing and Evaluation

Following the implementation phase, rigorous testing is crucial to validate the SVMU software and ensure its compliance with the specified requirements. Testing serves as an independent evaluation mechanism, akin to detecting and rectifying errors. It aims to determine whether the software meets the desired functionality and behaves as intended. The testing process involves subjecting the software to controlled conditions to assess its performance and functionality. By verifying that the software operates as specified, testing facilitates the detection of any errors or discrepancies. Additionally, testing serves as a means to validate whether the software aligns with the user's actual requirements. For SVMU, extensive real-time testing has been conducted to validate the accuracy and reliability of the implemented system. The model has undergone multiple rounds of testing and verification to ensure its effectiveness in recognizing and processing license plate information. By employing various test scenarios and datasets, the SVMU system's performance has been thoroughly evaluated to ascertain its adherence to the specified requirements.

The combination of both validation and specification testing is essential in ensuring that SVMU meets the desired requirements and operates correctly. By employing a comprehensive testing and evaluation approach, any potential issues or shortcomings can be identified and addressed, thus enhancing the overall quality and functionality of the SVMU system.

# Verification

Throughout the development of SVMU, verification has been conducted at each phase to ensure the system's adherence to the specified requirements. The completed phases were subjected to comprehensive testing to verify their alignment with the established requirements. The units within the system were thoroughly examined to ensure their satisfactory performance in meeting the requirements.

Verification is an essential activity that occurs at various stages of the software development process. It involves reviewing the documents and deliverables produced in the previous stages to assess whether the system has been built correctly. In the case of SVMU, the verification process began with requirements reviews and continued through design and code reviews, culminating in product testing. By conducting these verification activities, the development team aimed to identify and rectify any faults or deviations from the specified requirements. This iterative approach to verification ensures that the system is continually checked and validated throughout the development life cycle, reducing the likelihood of potential errors and ensuring the system's overall quality and correctness.

The verification process serves as a critical step in validating the SVMU system's compliance with the intended functionality, design, and performance. By thoroughly reviewing and testing each phase, any discrepancies or inconsistencies are identified and addressed, contributing to the development of a reliable and robust system.

**6.1.1 Functional Testing:**

Functional testing is a form of black box testing in which each component of the system undergoes testing based on functional specifications and requirements. This testing approach aims to verify that the system's functionalities align with the intended behavior as specified in the functional requirements.

1. After giving right credentials we are able to login in our system.

2. Our system will not open until user login into the system.

3. Our system successfully shows the record in database of registered users.

**TEST CASES**

**Test Case for Login**

|  |  |
| --- | --- |
| **Test Case ID:** | 01 |
| **Test Case Name:** | Login |
| **Description:** | Module must Login the User |
| **Pre-conditions:** | User must be registered |
| **Steps for Execution:** | 1. Enter email 2. Enter Password |
| **Post-conditions:** | Open the system |
| **Expected Results:** | User gets signed successfully. |
| **Actual Results:** | User signed successfully |
| **Status:** | Success. |

**Table 6.1.1 Login**

**Test Case for Register**

|  |  |
| --- | --- |
| **Test Case ID:** | 02 |
| **Test Case Name:** | Register |
| **Description:** | Module must Registered the user |
| **Pre-conditions:** | User have email, name |
| **Steps for Execution:** | 1. Enter email 2. Enter Password 3. Enter name 4. Enter CNIC |
| **Post-conditions:** | Login to the system. |
| **Expected Results:** | User get successful message. |
| **Actual Results:** | User get successful message. |
| **Status:** | Success. |
|  |  |

**Table 6.1.1 Register**

**Test Case for Vehicle Number Plate Recognition**

|  |  |
| --- | --- |
| **Test Case ID:** | 03 |
| **Test Case Name:** | Scan Vehicle Number Plate |
| **Description:** | Security guard will scan the number plate by the camera. |
| **Preconditions:** | Vehicle must be in front of camera with clear and visible number plate. |
| **Post conditions:** | The vehicle number plate is accurately captured by the system. |
| **Steps of execution:** | Vehicle arrives in front of camera.  AI system will scan the number plate.  The vehicle number is successfully captured by the camera. |
| **Expected Results:** | Number Plate will be recognized successfully. |
| **Actual Results:** | Number Plate will be recognized successfully. |
| **Status** | Success |

**Table 6.1.3 Vehicle Plate Number**

# Validation

Throughout the development process of SVMU, validation has been conducted at the completion of each module. The validation process aimed to ensure that the individual modules were integrated effectively to form the required end-product. This merging of modules was crucial to achieving the desired functionality and meeting the specified requirements. In line with best practices, SVMU underwent validation at each stage of the software development process, utilizing the documents generated during the previous stages. The validation process commenced with requirements reviews and continued through design and code reviews, culminating in product testing. This comprehensive testing was conducted to validate that the developed system aligns with the specified requirements and that the testing results yield successful outcomes.

Real-time testing of the SVMU model has been carried out to assess its performance and verify its accuracy. By subjecting the system to various test scenarios and datasets, the validation process aimed to ensure that SVMU operates as intended and satisfies the user's requirements. Validation serves as a crucial step in the software development life cycle, enabling the confirmation that the right product has been built and that it meets the specified requirements. Through thorough testing and validation, SVMU has been evaluated for its functionality, reliability, and performance, leading to the development of a validated and robust system.

# Usability Testing

Usability testing is done to evaluate the usability of the software. It can be done by following ways.

**6.3.1 Goals, Hypothesis and Method**

In order to ensure a successful and effective testing process for the Smart Vehicle Management at UAAR (University of Arid Agriculture Rawalpindi), it is imperative to establish clear goals, formulate hypotheses, and define a comprehensive methodology. The following guidelines should be followed:

**Goals:**

* Clearly define the purpose and objectives of the testing activities for Smart Vehicle Management at UAAR.
* Determine the specific reasons and motivations behind conducting these tests.
* Identify the desired outcomes and define the criteria for a successful test.
* Establish measurable goals that align with UAAR's overall objectives in managing smart vehicles.

**Hypotheses:**

* Formulate well-defined hypotheses regarding potential areas of concern or expected issues in the Smart Vehicle Management system.
* Identify the key aspects of the system that are likely to encounter challenges or require further validation.
* State the reasons behind the anticipated issues and their potential impact on UAAR's smart vehicle management operations.
* These hypotheses will serve as a basis for focusing the testing efforts on critical areas and validating assumptions.

**Methodology:**

* Develop a structured and comprehensive methodology for conducting the tests.
* Define the scope of the testing activities, including the specific aspects of Smart Vehicle Management to be assessed.
* Determine the appropriate testing techniques and tools to be employed, considering the unique requirements and constraints of UAAR's smart vehicle management system.
* Establish a clear timeline, milestones, and deliverables to ensure efficient planning and execution of the testing process.
* Document the methodology in a formal manner, including detailed procedures, guidelines, and protocols.

By adhering to these formal guidelines, UAAR can approach the testing of its Smart Vehicle Management system with a clear understanding of its goals, hypotheses, and methodologies. This will facilitate successful outcomes, ensuring the efficient and reliable management of smart vehicles within the university.

**6.3.2 Recruiting**

In order to conduct effective usability testing for SVMU's product, it is essential to recruit real users who have not been involved in the design or development process. Recruiting suitable users who align with SVMU's target audience is a crucial step in this regard. It is important to emphasize that the individuals selected for testing should closely match the characteristics and demographics of SVMU's intended user base. By recruiting representative users, SVMU can obtain valuable feedback and insights on the product's usability from an unbiased perspective. This approach ensures that the testing process reflects the experiences and expectations of the actual users who will interact with the product in real-world scenarios.

To ensure the recruitment of appropriate users, SVMU should consider the following:

* **Define the Target Audience:** Clearly identify the specific target audience for SVMU's product. This may include factors such as age, profession, technological proficiency, and any other relevant demographic or psychographic characteristics.
* **User Profiling:** Develop a detailed profile of the ideal user for SVMU's product, taking into account their needs, preferences, and goals.
* **Recruitment Channels:** Identify suitable channels and methods for recruiting users who match the defined target audience. This may involve utilizing online platforms, social media, professional networks, or engaging with relevant user communities.
* **Screening and Selection:** Conduct a screening process to ensure potential participants meet the defined criteria. Select participants who closely align with the target audience to ensure representative feedback.
* **Informed Consent and Confidentiality:** Prior to the usability testing, obtain informed consent from the recruited participants, explaining the purpose and nature of the testing process. Assure participants of the confidentiality of their feedback and protect their privacy throughout the testing phase.

By following these guidelines and recruiting users who closely match SVMU's target audience, the usability testing can provide valuable insights into how the product performs, identifies areas for improvement, and ultimately enhances the user experience for the intended users of SVMU's smart vehicle management system.

# Module / Unit Testing

Module / Unit testing is the process of testing each component of the software individually to ensure that it functions correctly. We will use the following methods for module / unit testing: Writing test cases for each individual component of the platform. Running automated tests on each component to ensure that it functions correctly.

# Integration Testing

Integration testing is a crucial phase in software development where previously tested units are combined to form a component, and the interfaces between them are tested. The component refers to an integrated aggregate of multiple units. This type of testing primarily focuses on ensuring the smooth functioning and compatibility of the software release.

Within the context of the SVMU project, integration testing is conducted to combine the model and website components in order to facilitate vehicle number detection. By integrating these two components, the functionality and communication between them are thoroughly tested to ensure seamless operation and accurate vehicle number detection. Integration testing plays a vital role in verifying the interoperability and overall performance of the integrated system, enabling SVMU to deliver a robust and reliable solution for managing smart vehicles.

# System Testing

System testing is an essential phase in software development, focusing on the evaluation of the integrated system as a whole. In this process, the software components are tested collectively to ensure their proper functioning and adherence to the desired specifications. A unit refers to a software component within the SVMU project that cannot be further divided into smaller components. Unit testing, a crucial aspect of software development, involves the thorough examination of these individual units to verify their correct operation. This testing approach assesses the smallest testable parts of the application, known as units, to ensure their independent functionality and compliance with the intended requirements. At SVMU, unit testing can be carried out either through automated means or manual scrutiny. The choice of the testing approach depends on the specific characteristics and complexity of the units being tested. Regardless of the method employed, the primary objective of unit testing within the SVMU project is to ensure the reliability and accuracy of the individual software components before they are integrated into the larger system.

# Acceptance Testing

We will use the following methods for acceptance testing: Testing the platform to ensure that it meets the needs and requirements of our users. Checking that the platform meets the functional and non-functional requirements of the project.

# Stress Testing

Stress testing is a crucial aspect of the testing process for SVMU's smart vehicle management system. It involves assessing the system's performance and stability under high-stress conditions, such as heavy user loads or excessive data processing. In the context of SVMU, stress testing aims to evaluate how the system handles peak usage scenarios and ensures its resilience in demanding situations. This type of testing allows SVMU to identify potential bottlenecks, performance issues, or weaknesses that may arise when the system is subjected to extreme or unexpected conditions.

The stress testing for SVMU may involve the following steps:

**Identify Stress Factors:** Determine the factors that can potentially stress the system, such as a large number of concurrent user requests, high data volumes, or complex computations.

**Define Stress Scenarios:** Design specific scenarios that simulate high-stress conditions, reflecting the anticipated usage patterns or potential peak loads on the system. These scenarios should closely resemble real-world situations that SVMU expects the system to handle.

**Test Execution:** Execute stress tests by subjecting the system to the defined stress scenarios. Monitor and measure key performance indicators, such as response times, throughput, resource utilization, and error rates.

**Performance Measurement and Analysis:** Collect and analyze performance data during stress testing to identify any performance degradation, bottlenecks, or failures. This analysis helps SVMU understand the system's behavior under stress and pinpoint areas that require optimization or further testing.

**Load Balancing and Scalability Assessment:** Evaluate the system's ability to distribute the load effectively and scale resources as the stress increases. This assessment helps SVMU determine if the system can handle increasing demands by dynamically allocating resources or scaling up infrastructure.

**Performance Optimization:** Based on the findings from stress testing, optimize the system by addressing identified performance issues or bottlenecks. This may involve tuning configurations, optimizing code, or enhancing system architecture.

By conducting thorough stress testing, SVMU can ensure that its smart vehicle management system can withstand high-stress scenarios, maintain stable performance, and provide reliable functionality even under demanding conditions. This testing process helps SVMU deliver a robust and scalable solution that can effectively handle the challenges associated with managing a large number of smart vehicles.

# Hardware Configuration for Testing

In the context of SVMU, the hardware configuration for testing involves ensuring the compatibility and proper functioning of the platform across different operating systems and browsers. This is accomplished through the following methods:

1. **Testing on Different Operating Systems:** The SVMU platform will be tested on various operating systems, such as Windows, macOS, Linux, and mobile operating systems (e.g., Android, iOS). This testing ensures that the platform operates correctly and seamlessly on different environments, providing a consistent user experience across multiple operating systems.
2. **Testing on Different Browsers:** The SVMU platform will be tested on various web browsers, including popular options such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. By conducting testing on different browsers, SVMU can verify the compatibility of the platform and ensure its smooth operation and visual consistency across a range of browser environments.

Through comprehensive hardware configuration testing, SVMU can ascertain that its platform performs reliably and consistently across diverse operating systems and browsers. This approach enhances the usability and accessibility of the SVMU system, enabling users to access and utilize its functionalities seamlessly regardless of their preferred operating system or browser choice.

# Evaluation

In the context of SVMU, the evaluation process entails employing the following methods to assess the effectiveness and performance of the platform:

1. **Collecting User Feedback:** SVMU will actively gather feedback from users to gauge their satisfaction with the platform. This can be achieved through surveys, user interviews, or feedback forms. By capturing user opinions, suggestions, and concerns, SVMU can gain valuable insights into the user experience and identify areas for enhancement.
2. **Measuring Performance and Uptime:** It is crucial for SVMU to monitor the platform's performance and uptime to ensure it meets the project's requirements. This involves measuring response times, system availability, and overall reliability. By closely monitoring these metrics, SVMU can identify and address any performance issues or system downtime that may impact the user experience.
3. **Analyzing Usage Data:** SVMU will analyze the platform's usage data to gain insights into user behavior and identify areas for improvement. This includes evaluating user engagement patterns, identifying popular features or functionalities, and understanding user preferences. By analyzing usage data, SVMU can make data-driven decisions to optimize the platform's usability and align it with user needs.

By employing these evaluation methods, SVMU can continuously assess and improve the platform's effectiveness, user satisfaction, and performance. This iterative approach ensures that SVMU delivers a high-quality smart vehicle management solution that meets the evolving requirements and expectations of its users.

# Deployment

In the deployment process for the SVMU website platform project, the following steps are involved:

**Development Server Deployment:**

Initially, the project is deployed on a development server specifically designated for testing purposes. This allows the SVMU team to conduct initial testing, evaluate the functionality, and identify any potential issues or bugs within the platform.

**Staging Environment Deployment:**

After the development server testing is completed, the code is then deployed to a staging environment. The staging environment serves as a replica of the production environment and provides a controlled setting for thorough testing. This step involves conducting comprehensive testing, including functional, performance, and compatibility testing, to ensure the platform functions as intended and meets the required quality standards.

**Testing and Approval:**

During the staging environment deployment, the SVMU team performs rigorous testing, including user acceptance testing (UAT) and final quality assurance checks. This ensures that all the necessary features are functioning correctly and that the platform is ready for deployment in the production environment. Upon successful completion of testing and approval, the code is considered ready for deployment.

**Production Environment Deployment:** Once the code passes all necessary testing and receives approval, it is deployed to the production environment. The production environment is the live environment where the platform will be accessible to end-users. Careful consideration is given to ensure a smooth and seamless transition from the staging environment to the production environment, minimizing any disruptions or downtime during the deployment process.

By following this deployment process, SVMU ensures that the website platform is thoroughly tested and verified at each stage before being made available to users in the production environment. This approach helps maintain the integrity and reliability of the platform while minimizing the risk of any issues impacting the end-users' experience.

# Chapter 7: Conclusion and Future Work

This chapter concludes the project and highlights future work.

# Conclusion

In conclusion, SVMU (Smart Vehicle Management in UAAR using AI-based License Plate Recognition System) offers an efficient and reliable solution for managing vehicle entry and registration at UAAR. The proposed system utilizes AI-powered license plate recognition, a user-friendly web interface, and real-time data logging to enhance security, streamline vehicle authorization, and improve overall efficiency. By automating the identification and registration process, SVMU eliminates the manual and time-consuming tasks associated with vehicle checking, reducing errors and increasing accuracy. The system provides a comprehensive record of vehicle entry, driver information, and guards on duty, ensuring better security and enabling effective management of parking slots. Through the utilization of YOLOv5 algorithm and a web-based interface, SVMU achieves high accuracy in license plate recognition while providing accessibility and convenience to users.

# Future Work

In the future, SVMU can be further enhanced and expanded to incorporate additional features and functionalities. Some potential areas of improvement and future work include:

* **Integration with Access Control Systems:** SVMU can be integrated with access control systems, such as automatic barriers or gates, to enable seamless entry and exit of authorized vehicles without human intervention.
* **Advanced Analytics and Reporting:** Implementing advanced analytics capabilities can provide insights into vehicle traffic patterns, peak hours, and occupancy rates. This data can be used to optimize parking space allocation and improve overall traffic management.
* **Integration with Payment Systems:** SVMU can be integrated with payment systems to enable automated payment for parking fees or permit validation, enhancing the user experience and reducing manual transactions.
* **Mobile Application Development:** Developing a dedicated mobile application can allow users to easily register their vehicles, receive notifications, and access real-time information about parking availability and their vehicle's status.
* **Integration with Security Cameras**: Integration with security cameras can provide visual monitoring and surveillance of the parking area, enhancing overall security measures.

By implementing these future enhancements, SVMU can further streamline the vehicle management process, improve security, and enhance the user experience for both visitors and university staff.

# References

|  |  |
| --- | --- |
| [1] | M. A. A. N. T. A. Saif Ur Rehman, "An Efficient Approach for Vehicle Number Plate Recognition in Pakistan," *The Open Artificial Intelligence Journal,* vol. 6, no. 1, pp. 12-21, 2020. |
| [2] | A. K. D. V. U. Puranic, "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching," *International Journal of Computer Applications,* vol. 134, no. 1, pp. 12-16, 2016. |
| [3] | S. M. J. C. R. Silva, "Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks," *Proceedings - 30th Conference on Graphics, Patterns and Images, SIBGRAPI 2017,* pp. 55-62, 2017. |
| [4] | L. B. Wu Hui, "License plate recognition system," *2011 International Conference on Multimedia Technology, ICMT 2011,* pp. 5425-5427, 2011. |
| [5] | . D. K. T. Student Undergraduate and . U. V. A. Student Undergraduate, "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching," *International Journal of Computer Applications,* vol. 134, no. 1, pp. 12-16, 2016. |
| [6] | "A vehicle license plate detection method using region and edge based methods," *Computers and Electrical Engineering,* vol. 39, no. 3, pp. 834-845, 2013. |