Dr. D.Y. Patil School of MCA

Charoli (BK), PUNE- 412105

SAVITRIBAI PHULE PUNE UNIVERSITY

MASTER OF COMPUTER APPLICATION

Project Report

on

VEHICLE DETECTION SYSTEM

Under The Guidance Of

“Prof. Asmita Hendre”

BY

“Sanskar Wagavkar & 236”

Class: MCA-I (Sem-II)

Year: 2022-2023

Index

ABSTRACT

The purpose of this project is to develop a vehicle detection system using Python and OpenCV to monitor the vehicles entering and exiting a college campus. The system also has a front-end interface developed using HTML, CSS, and Flask framework, and a back-end database using MySQL to store the relevant information about the vehicles. The vehicle detection system works by processing live video footage of the entrance and exit points of the college campus using OpenCV. The system can detect the presence of vehicles in the video feed and classify them as either cars, trucks, or bikes. The information about the detected vehicles is then stored in the MySQL database along with the vehicle's registration number, date and time of entry, and exit. The front-end interface developed using HTML, CSS, and Flask framework provides an easy-to-use web interface for the college administrators to view the real-time status of the vehicles entering and exiting the campus. The interface also allows the administrators to search for specific vehicles and view their detailed entry and exit logs.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Vehicle detection systems have become an essential part of modern-day traffic management, especially in crowded areas such as college campuses.

A vehicle detection system can help monitor the movement of vehicles entering and exiting a campus, which can help improve security, traffic flow, and overall management of the campus. In this project, we have developed a vehicle detection system using Python and OpenCV to monitor the vehicles entering and exiting a college campus. The system is designed to process live video footage of the entrance and exit points of the college campus using OpenCV.

The system can detect the presence of vehicles in the video feed and classify them as cars, trucks, or bikes. The information about the detected vehicles is then stored in a MySQL database, which can be accessed through a web interface developed using HTML, CSS, and Flask framework. The primary goal of this project is to provide a user-friendly interface for the college administrators to monitor the real-time status of the vehicles entering and exiting the campus.

The interface also allows administrators to search for specific vehicles and view their detailed entry and exit logs. The back-end MySQL database provides a scalable and efficient way to store and manage the large amounts of data generated by the vehicle detection system.

1.2 EXISTING SYSTEM & LIMITATION OF EXISTING SYSTEM

- Currently, most college campuses rely on manual methods such as security personnel to monitor the entry and exit of vehicles. These methods are often ineffective, time-consuming, and prone to errors.

- Manual monitoring can also lead to traffic congestion and security breaches, especially during peak hours. Some campuses may have installed automatic gates or barriers to control the entry and exit of vehicles.

- However, these systems are expensive to install and maintain, and they may not be able to provide real-time monitoring and analysis of the vehicles entering and exiting the campus.

- Manual monitoring of vehicle entry and exit points on college campuses poses several challenges. Firstly, it is a labor-intensive process that requires dedicated security personnel to be stationed at entry and exit points.

- This can be both costly and inefficient, as human error and fatigue can compromise the accuracy and effectiveness of the monitoring process. Furthermore, manual monitoring can lead to traffic congestion during peak hours, creating inconvenience for the campus community and increasing the risk of accidents.

- The lack of real-time data and analysis also limits the ability to identify potential security breaches promptly.

1.3 NEED FOR THE PROPOSED SYSTEM

Need for the System:

- The increasing number of vehicles and the growing concern for campus security have highlighted the need for an efficient and automated vehicle monitoring system in college campuses. The manual monitoring methods are often ineffective and time-consuming, which can lead to traffic congestion and security breaches.

- Therefore, there is a need for a cost-effective and reliable system that can monitor the entry and exit of vehicles in real-time, store the relevant information about the vehicles, and provide a user-friendly interface for campus administrators to access the information.

- The vehicle detection system developed in this project addresses these needs by providing a cost-effective and reliable solution for monitoring the vehicles entering and exiting a college campus.

The system can help improve traffic flow, security, and overall management of the campus, while also providing valuable insights for future planning and decision-making.

CHAPTER 2: PROPOSED SYSTEM

2.1 PROBLEM STATEMENT

The lack of efficient and automated vehicle monitoring solutions poses significant challenges for college campuses. The problems can be summarized as follows:

- Ineffectiveness and Errors: Manual methods of monitoring vehicle entry and exit points on campuses are prone to human errors, leading to inaccurate data and compromised security. Inconsistencies in monitoring procedures can result in unauthorized access, potential security breaches, and increased risks to the campus community.

- Time-Consuming Processes: The reliance on security personnel for manual monitoring slows down the vehicle entry and exit procedures. This often results in long waiting times, traffic congestion, and inconvenience for students, faculty, staff, and visitors. Lengthy queues and delays can negatively impact the overall campus experience and hinder productivity.

- Limited Real-time Monitoring and Analysis: Manual methods lack the ability to provide real-time monitoring and analysis of vehicles entering and exiting the campus. The absence of instantaneous data and insights hinders the campus security team's ability to respond promptly to potential security threats or unauthorized access incidents.

- Traffic Congestion: Manual monitoring contributes to traffic congestion, especially during peak hours. Inefficient traffic management can cause delays, create bottlenecks, and increase the risk of accidents. The lack of dynamic traffic control measures based on real-time data exacerbates these challenges.

2.2 OBJECTIVES OF PROPOSED SYSTEM

The proposed vehicle detection system aims to achieve the following objectives: Automate the vehicle monitoring process.

The system aims to automate the process of monitoring the vehicles entering and exiting a college campus by using computer vision techniques to detect and classify the vehicles in real-time.

- **Improve campus security: -**

  The system aims to improve campus security by providing a more efficient and effective way of monitoring the vehicles entering and exiting the campus. The system can help identify unauthorized vehicles and potential security threats, which can help prevent security breaches.

- **Improve traffic flow: -**

  The system aims to improve traffic flow by providing real-time information on the number of vehicles entering and exiting the campus. Campus administrators can use this information to optimize traffic flow and reduce congestion.

- **Provide detailed vehicle logs: -**

  The system aims to provide detailed logs of the vehicles entering and exiting the campus, including the date, time, and vehicle type. This information can help campus administrators track the movement of vehicles on the campus and identify any suspicious activities.

- **Provide a user-friendly interface: -**

  The system aims to provide a user-friendly web interface for campus administrators to access the vehicle monitoring system. The interface should be intuitive and easy to use, allowing administrators to view real-time data, search for specific vehicles, and generate reports.

2.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT

Functional Requirements:

- Real-time Monitoring: The automated vehicle monitoring system should provide real-time monitoring of vehicles entering and exiting the campus. It should capture and record relevant data such as license plate numbers, vehicle descriptions, and timestamps.

- Security Alerts: The system should be capable of detecting and alerting security personnel about suspicious or unauthorized vehicles. It should have the ability to integrate with existing security infrastructure, such as surveillance cameras and access control systems, to ensure a comprehensive security response.

- Access Control: The system should have the capability to control vehicle access to the campus based on predefined rules and permissions. It should allow authorized vehicles to enter seamlessly while preventing unauthorized vehicles from gaining entry.

- Data Analysis and Reporting: The system should analyze the captured data to identify patterns, anomalies, and potential security threats. It should provide comprehensive reports and analytics to assist in decision-making and proactive security measures.

Non-Functional Requirements:

- Reliability: The system should operate reliably and consistently, ensuring continuous monitoring of vehicle entry and exit points without interruptions or downtime.

- Scalability: The system should be scalable to accommodate the growing number of vehicles and changing campus requirements. It should be able to handle increased traffic volumes without compromising performance or accuracy.

- Usability: The system should have a user-friendly interface, allowing security personnel to easily operate and navigate the monitoring system. Training requirements should be minimal, and the system should provide clear instructions and prompts for users.

- Security and Privacy: The system should adhere to robust security protocols to protect the captured data from unauthorized access or tampering. It should comply with privacy regulations and ensure the confidentiality of personal information associated with vehicles and their owners.

2.4 SCOPE OF SYSTEM

- The scope of the vehicle detection system developed in this project is to provide an efficient and automated solution for monitoring the vehicles entering and exiting a college campus. The system is designed to process live video footage of the entrance and exit points of the college campus using OpenCV and classify the detected vehicles as cars, trucks, or bikes.

- The information about the detected vehicles is then stored in a MySQL database, which can be accessed through a web interface developed using HTML, CSS, and Flask framework. The system's scope also includes providing a user-friendly interface for college administrators to monitor the real-time status of the vehicles entering and exiting the campus.

- The interface also allows administrators to search for specific vehicles and view their detailed entry and exit logs. The back-end MySQL database provides a scalable and efficient way to store and manage the large amounts of data generated by the vehicle detection system. The system's scope also includes generating reports and analytics based on the data stored in the MySQL database, which can provide valuable insights for future planning and decision-making.

- In summary, the scope of the vehicle detection system developed in this project is to provide an efficient, automated, and user-friendly solution for monitoring the vehicles entering and exiting a college campus, which can help improve traffic flow, security, and overall management of the campus.

2.5 MODULE SPECIFICATION

Module 1: Vehicle Registration -

In this module We have to register your vehicle and take the approval from the higher authority

- **USER DETAILS**: - In this user have enter the Personal Details like NAME, GENDER, VEHICLE NUMBER, LICENSE,LOCATION etc.

- **APPLY: -** After Filling the approval form students has to take sign from the admin office and office will give the sticker and permission to the vehicle.

- **PRINT FORM**: - Based on the information provided by the user above we, provide the PDF file to the user.

Module 2: Detect Vehicle by Video -

In this module User can detect the vehicle and get the vehicle information by just facing the camera towards the number plate.

- **SEARCH BY VIDEO**: - In this user have to open the camera and face towards to the number plate.

- **GET INFORMATION: -** After potting the camera towards the number plate number will detect and then it will show all information to the user.

Module 3: Detect Vehicle by Photo -

In this module User can detect the vehicle and get the vehicle information by just upload the picture of the number plate.

- **SEARCH BY PHOTO**: - In this user have to open the camera and click the picture and then upload the photo.

- **GET INFORMATION: -** After potting the Uploading photo, the number plate number will detect and then it will show all information to the user.

Module 4: Penalty -

In this module Authentic User can give the penalty and student can get the SMS on their mobile phone about penalty.

- **GIVE PENALTY**: - In this user has to login again and then enter the Vehicle number.

- **FILL DETIALS**: - In this user has to fill the data like(place, vehicle number, reason for penalty, amount)

- **GET PENALTY: -** After fill all the details user or student will get the SMS on their mobile phone about the penalty and need to report to the office.

2.6 OPERATING ENVIRONMENT

The vehicle detection system developed in this project has specific hardware and software requirements to operate efficiently.

Hardware Requirements:

- A computer system with a multi-core processor (Intel Core i3 or higher)
- Minimum 8 GB RAM
- A camera with a minimum of 360p resolution
- Sufficient storage space (minimum 10GB) for storing the captured video footage
- Database files Stable internet connectivity

Software Requirements:

- Operating System: Windows 10 or Higher
- Python 3 with required libraries, including OpenCV
- MySQL Connector MySQL Server 8.0 or higher
- TKinter for developing the GUI interface

CHAPTER 3: REQUIREMENT DETERMINATION AND ANALYSIS

3.1 FACT FINDING METHODS

- The paper highlights the importance of vehicle monitoring systems in college campuses for preventing unauthorized access, detecting suspicious activities, and improving overall security. The authors propose a system that uses automatic license plate recognition (ALPR) technology and a wireless sensor network (WSN) to track and monitor vehicles entering and exiting the campus.

- The proposed system uses an ALPR camera to capture the license plate number of vehicles, which is then transmitted to a central server using a WSN. The server processes the data and generates alerts if any suspicious activities are detected. The system also includes a web-based user interface that provides real-time monitoring and access to historical data.

- The paper describes the architecture, hardware, and software components of the proposed system and evaluates its performance through simulation experiments. The results show that the system can accurately detect and track vehicles, and generate alerts in real-time.

- Overall, the paper presents a comprehensive and innovative approach to vehicle monitoring and tracking in college campuses. The proposed system has the potential to improve the security of college campuses and provide valuable insights into the traffic flow and patterns of vehicles entering and exiting the campus.

3.2 FEASIBILITY STUDY

1. **Technical Feasibility:**

   i. Hardware Requirements: Assess the technical capability of the hardware required to run the app effectively. Consider factors such as processing power, memory, storage, and camera capabilities.
   ii. Software Requirements: Evaluate the compatibility of the app with the operating systems and versions targeted for deployment. Ensure that all necessary software dependencies are available and can be integrated seamlessly. Image Processing and Number Plate
   iii. Recognition: Investigate the technical feasibility of implementing image processing algorithms and number plate recognition techniques using the chosen framework (Tkinter) and libraries.
   iv. Database Integration: Assess the compatibility and feasibility of integrating a MySQL database with the app for storing and retrieving number plate data.

2. **Economic Feasibility:**

   i. Cost Analysis: Evaluate the cost of developing the app, including hardware, software licenses, development resources, and any additional expenses such as server hosting or cloud services.
   ii. Return on Investment (ROI): Determine the potential financial benefits and returns that can be derived from the app. Consider factors like revenue generation, cost savings, and potential market demand.

3. **Operational Feasibility:**

i. User Acceptance: Analyze the target users' needs, preferences, and expectations to ensure the app meets their requirements. Conduct user surveys or focus groups to gather feedback and assess user acceptance.

ii. User Interface (UI) and User Experience (UX): Evaluate the usability and intuitiveness of the app's interface. Consider factors such as ease of navigation, clarity of instructions, and overall user experience.

iii. Scalability and Performance: Determine if the app can handle the expected workload and scale effectively as the number of users and data volume increases. Evaluate the app's performance under various scenarios and identify any potential bottlenecks.

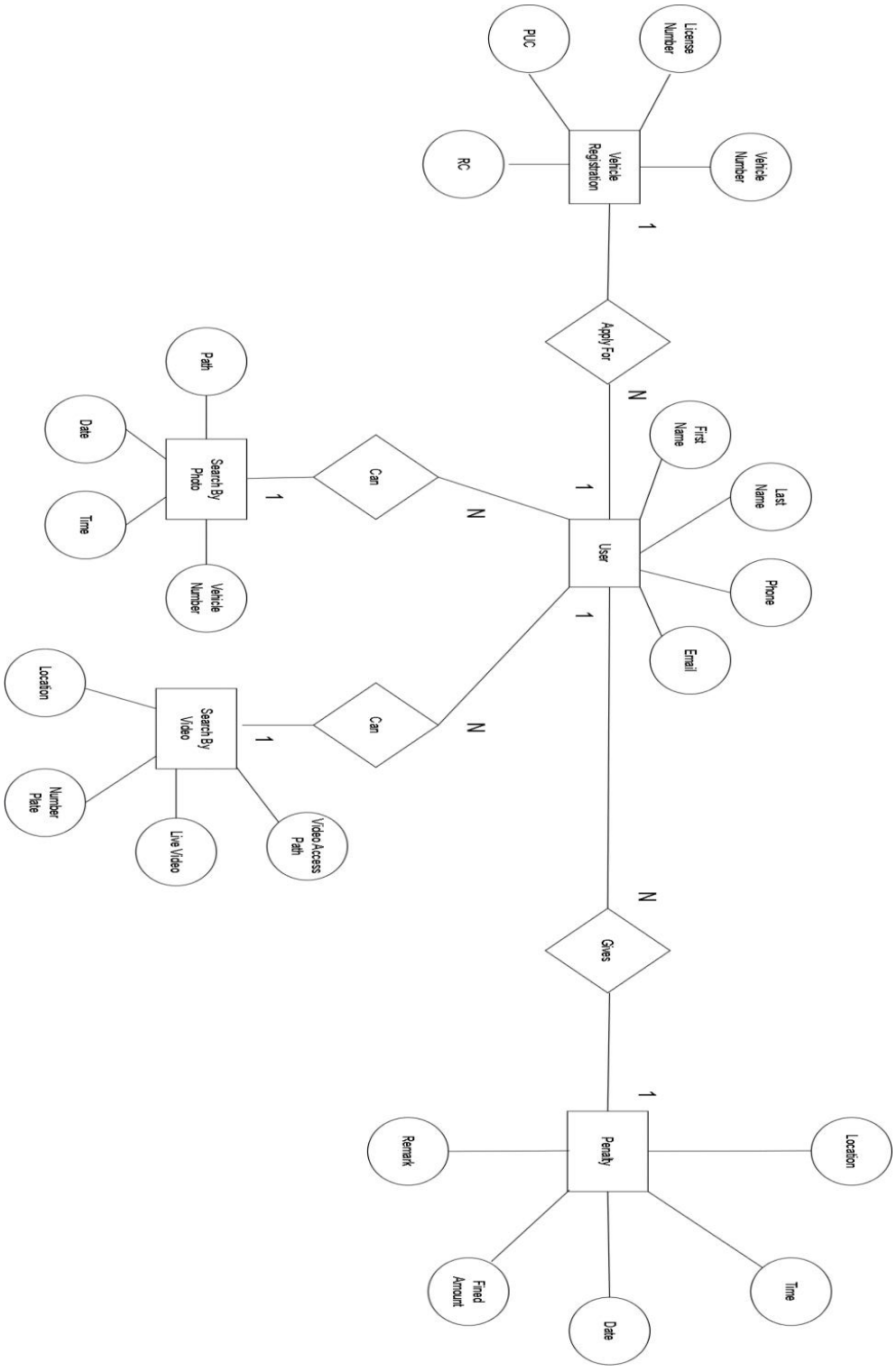
4. **Legal Feasibility:**

a. Privacy and Data Protection: Ensure compliance with relevant privacy laws and regulations when collecting, storing, and processing users' personal information. Implement appropriate security measures to protect the data stored in the MySQL database.

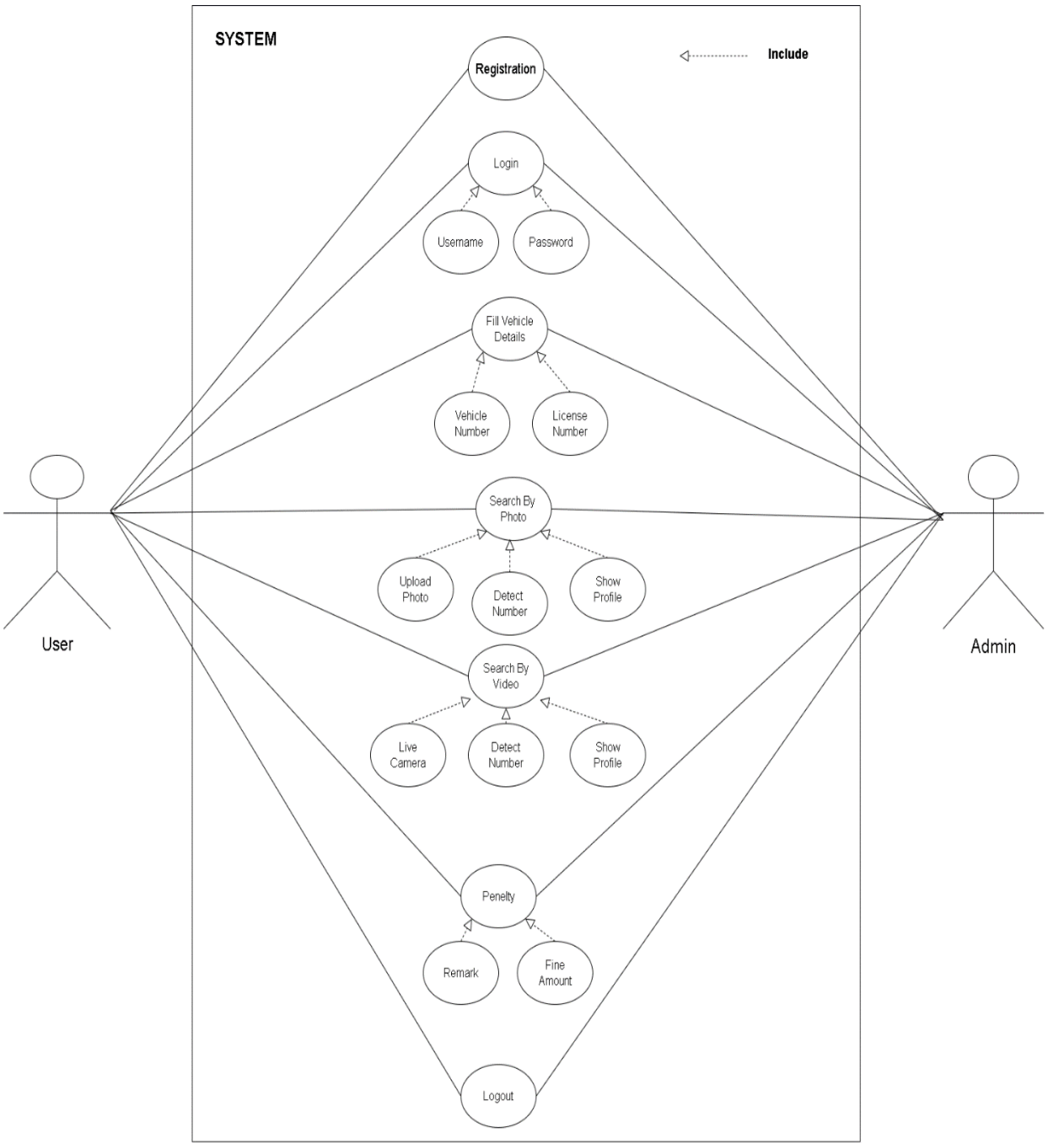
b. Intellectual Property: Assess the legal implications of using image processing algorithms, number plate recognition techniques, or any third-party libraries. Verify that the app's development and deployment do not infringe on any patents or copyrights.

CHAPTER 4: SYSTEM ANALYSIS AND DESIGN

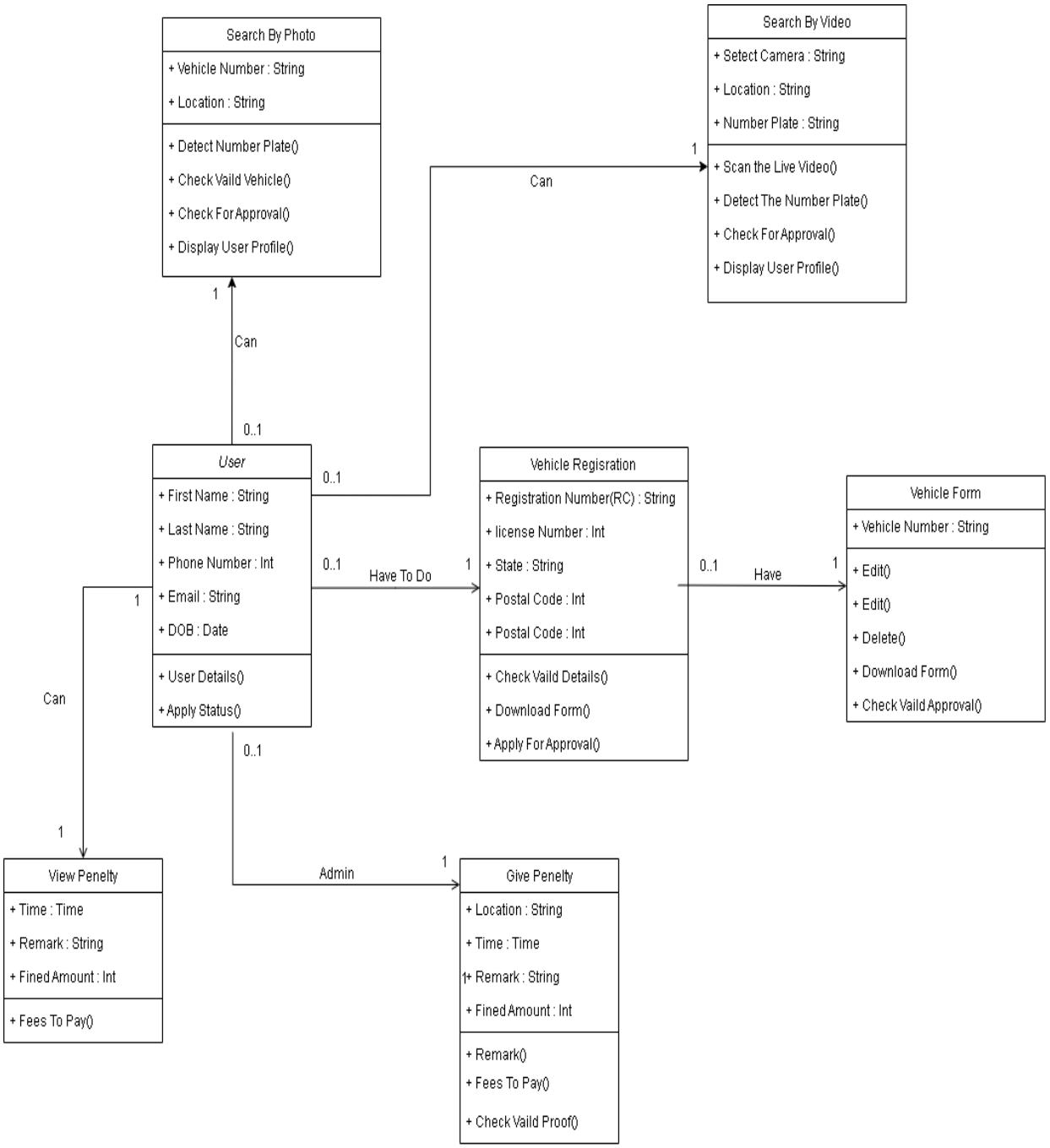
4.1 ENTITY RELATIONSHIP DIAGRAM FOR VEHICLE DETECTION SYSTEM



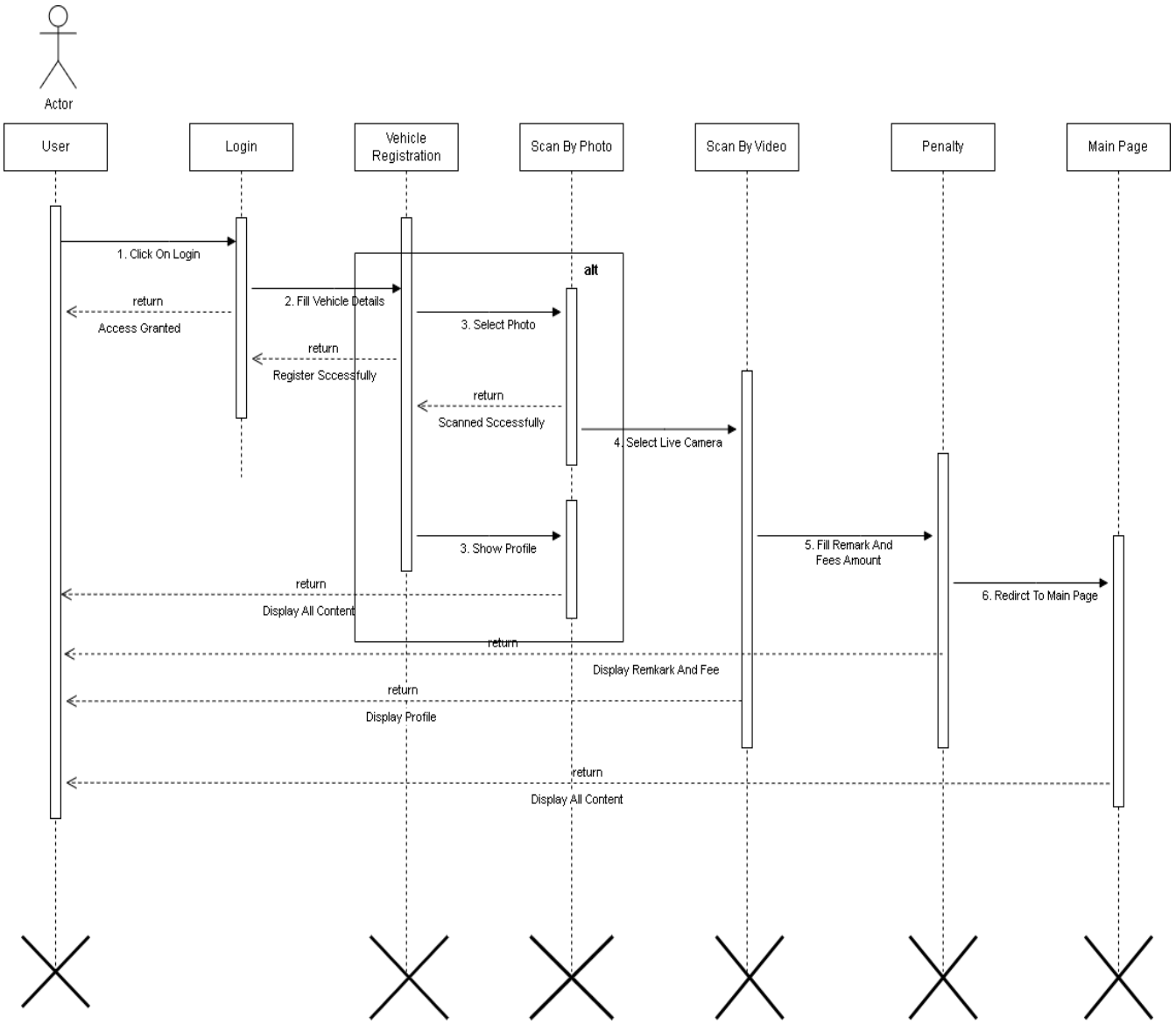
4.2 USE CASE DIAGRAM FOR VEHICLE DETECTION SYSTEM



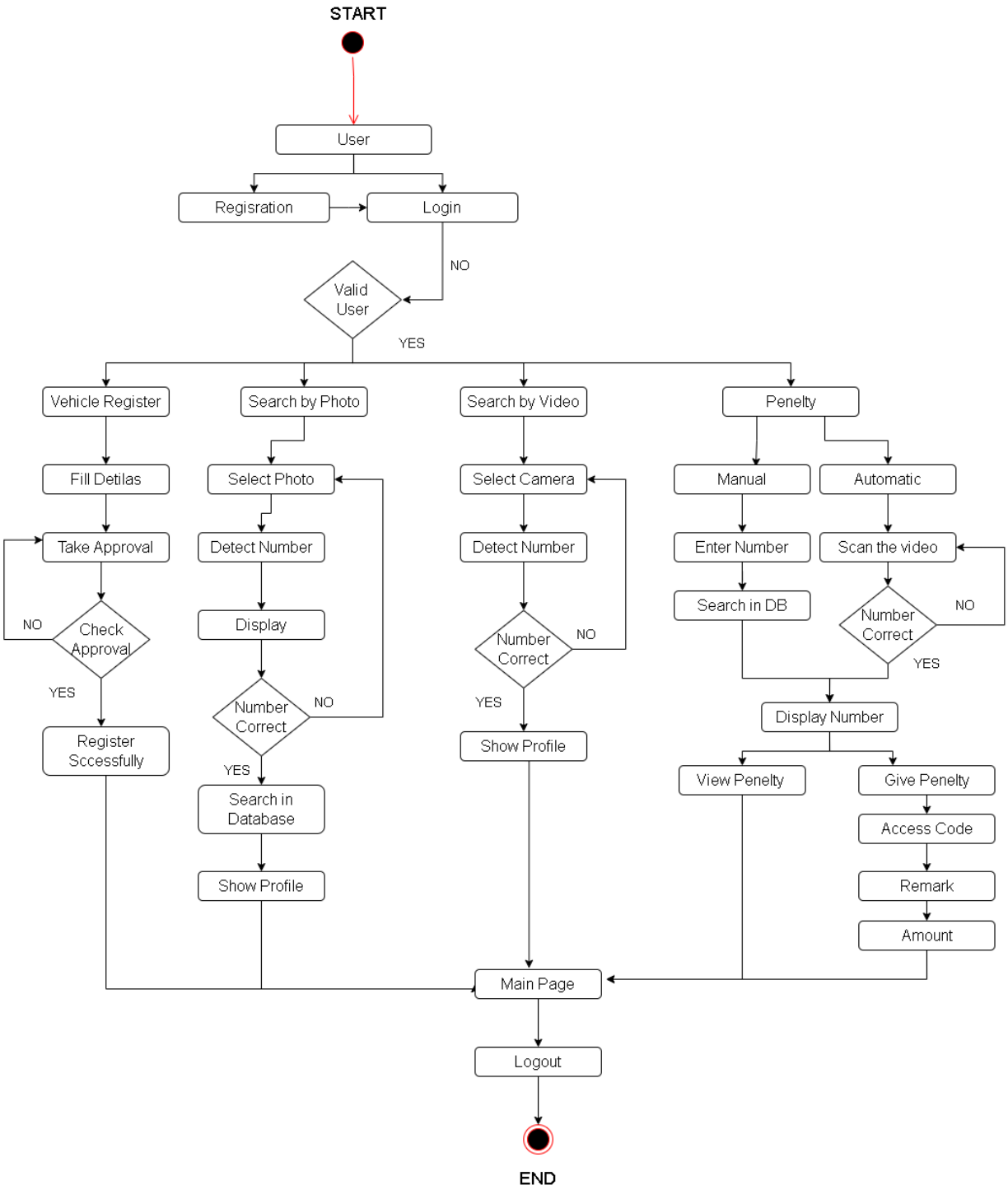
4.3 CLASS DIAGRAM FOR VEHICLE DETECTION SYSTEM



4.4 SEQUENCE DIAGRAM FOR VEHICLE DETECTION SYSTEM



4.5 ACTIVITY DIAGRAM FOR VEHICLE DETECTION SYSTEM



4.6 TABLE SPECIFICATION

Registration Data: -

Fields	Description	Type	Size	Constraints
Name	Name of the User	VARCHAR	20	Not Null
Email	Email id of theUser	VARCHAR	20	Not Null
Phone no	Phone no. of theUser	INT	10	Primary Key
Username	Username of theUser	VARCHAR	20	Not Null
Password	Password for theUser	VARCHAR	20	Not Null
Confirm Password	Confirm Passwordfor the User	VARCHAR	20	Not Null

Vehicle Registration Data: -

Fields	Description	Type	Size	Constraints
First Name	First Name	VARCHAR	20	Not Null
Last Name	Last Name	VARCHAR	20	Not Null
Phone	Phone	INT	10	Primary Key
Email	Email	VARCHAR	20	Not Null
University	University Name	VARCHAR	20	Not Null
Course	Course Name	VARCHAR	20	Not Null
Image Path	Path to where Image stored	VARCHAR	20	Not Null

Detection Data: -

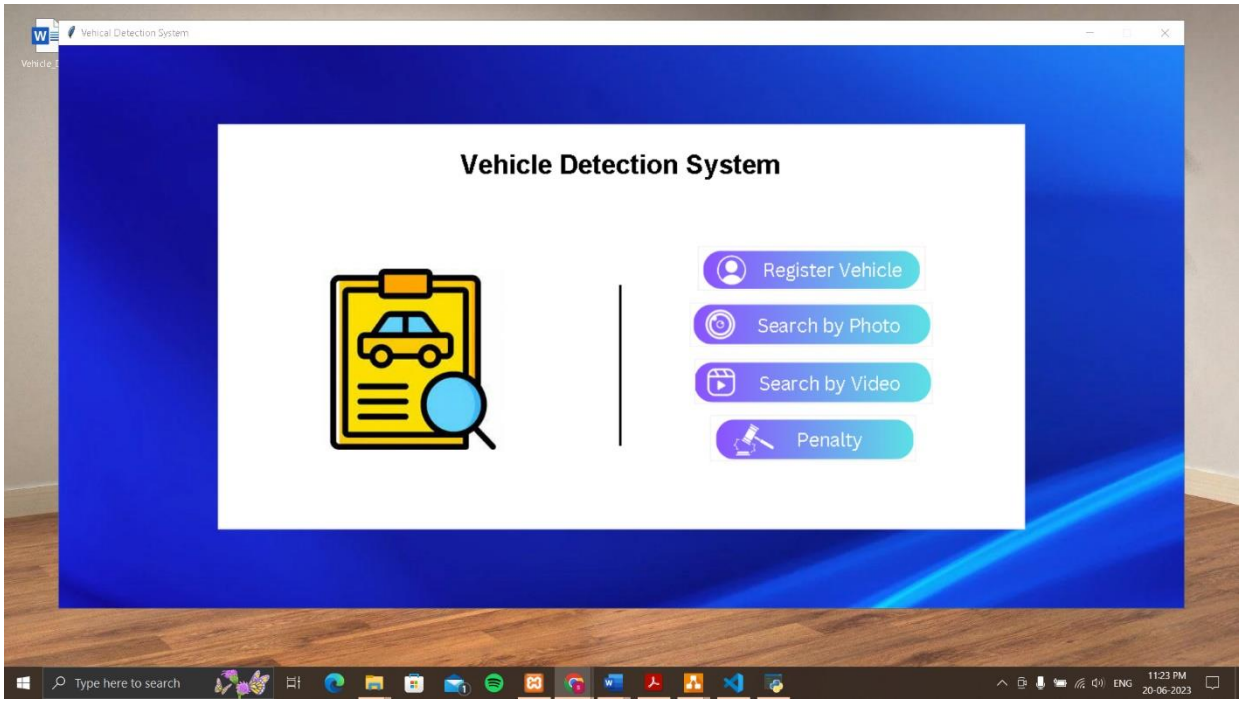
Fields	Description	Type	Size	Constraints
Scan Type	Photo/Video	VARCHAR	20	Not Null
Date	Current Data	DATE	20	Not Null
Time	Current Time	TIME	10	Not Null
Number Detected	Scan Number Detected by Algorithm	VARCHAR	20	Not Null
Vehicle No	Vehicle Number with Name	VARCHAR	20	Primary Name

Penalty Data: -

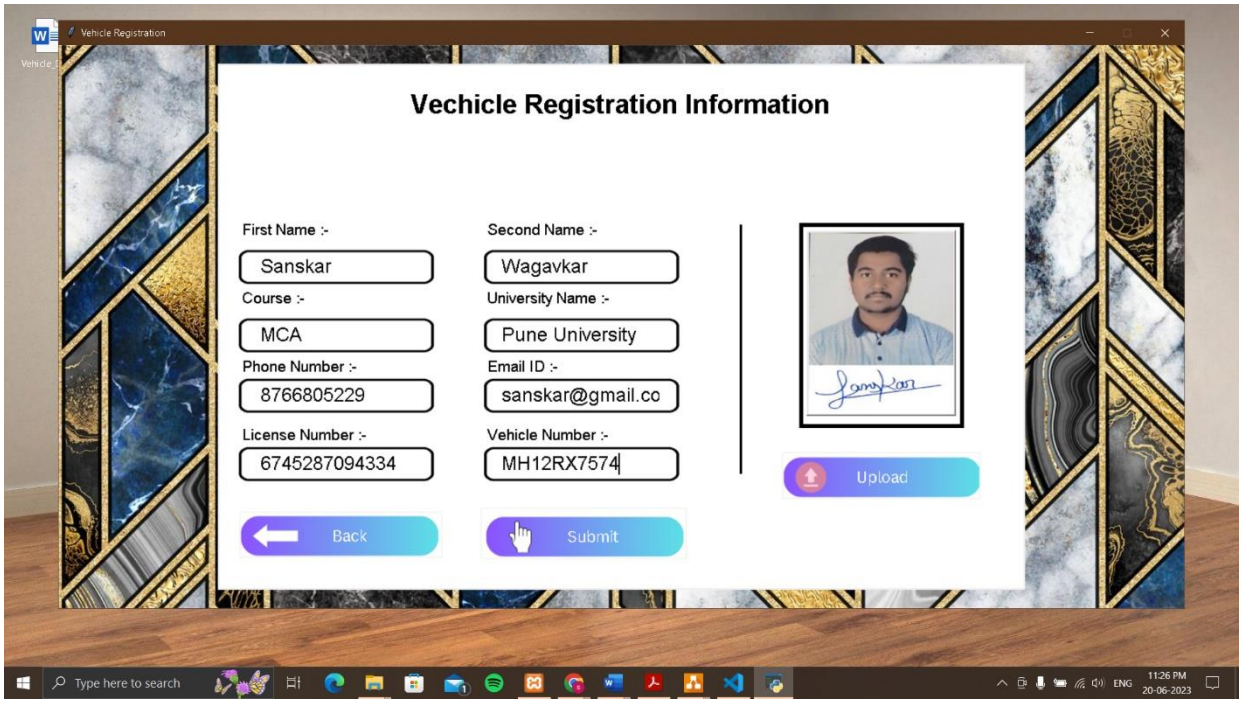
Fields	Description	Type	Size	Constraints
Vehicle Number	Vehicle Number	VARCHAR	20	Primary
Date	Current Data	DATE	20	Not Null
Time	Current Time	TIME	10	Not Null
Remark	Reason For Fined	VARCHAR	100	Not Null
Fined Amount	Fined Amount to be Pay	VARCHAR	20	Not Null

4.7 USER INTERFACE DESIGN

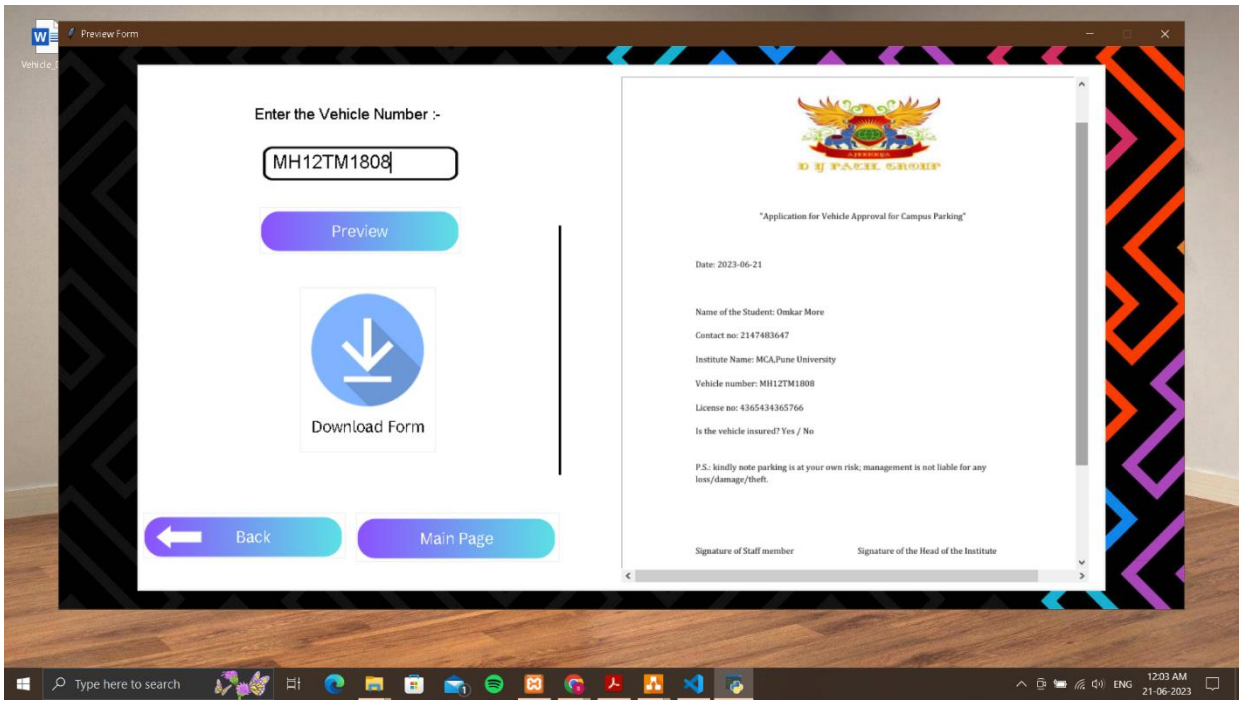
Main Page: -



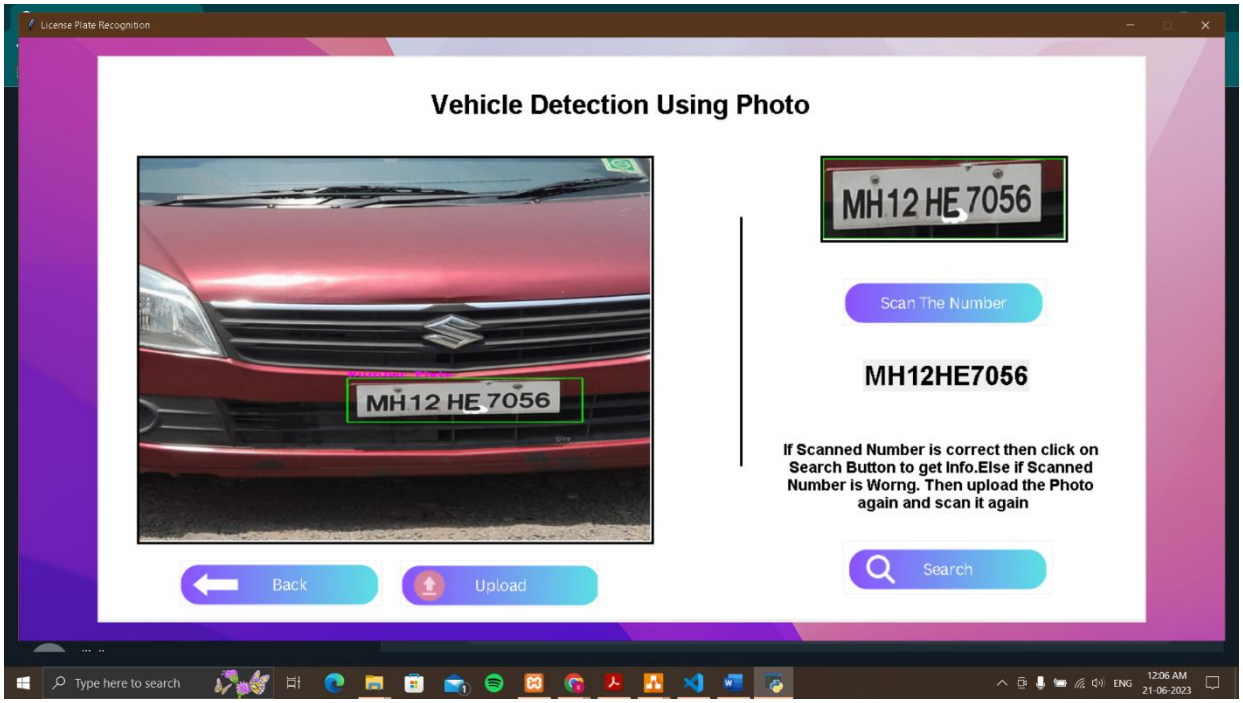
Vehicle Registration Page: -



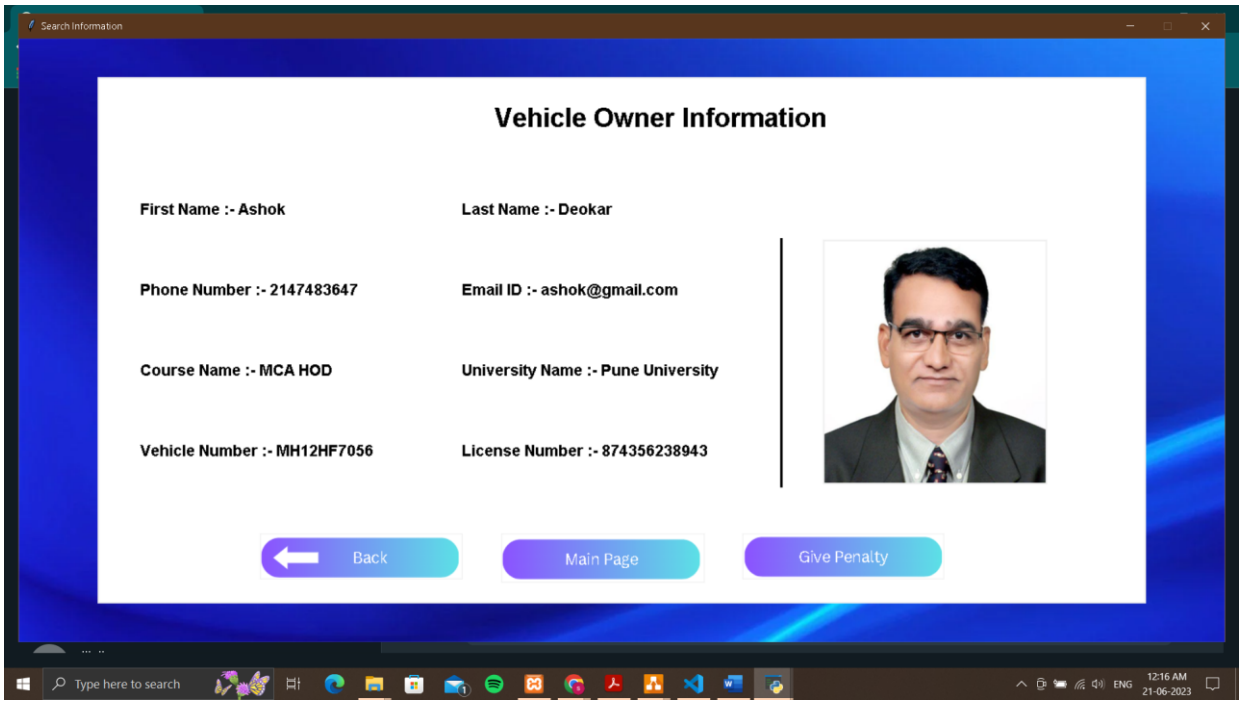
Approval Form: -



Detection By Photo: -



User Information: -



CHAPTER 5: DRAWBACKS AND LIMITATIONS

- Initial Setup Costs: Implementing an automated vehicle monitoring system can require significant upfront investment. The costs associated with acquiring hardware, software, and integrating the system with existing infrastructure can pose financial challenges for campuses with limited budgets.

- Infrastructure Requirements: These systems may require the installation of additional infrastructure components such as cameras, sensors, and communication networks. Ensuring proper coverage and connectivity across the campus can be complex and time-consuming, especially in larger or geographically dispersed campuses.

- Maintenance and Upkeep: Automated systems require regular maintenance, updates, and technical support to ensure their continued functionality. This can involve ongoing costs for system monitoring, software upgrades, hardware repairs, and personnel training. Failure to adequately maintain the system can result in decreased performance or system malfunctions.

- Technological Limitations: Automated vehicle monitoring systems heavily rely on technologies such as computer vision, machine learning, and artificial intelligence. While these technologies have advanced significantly, they may still have limitations in accurately capturing and interpreting data under certain conditions, such as poor lighting, adverse weather, or obscured license plates.

- Privacy Concerns: The collection and storage of vehicle data raise privacy concerns, as the system captures license plate numbers and vehicle information. It is crucial to have appropriate protocols and safeguards in place to ensure the secure handling and storage of this sensitive information. Campuses must comply with relevant privacy laws and regulations to protect the privacy rights of individuals.

CHAPTER 6: PROPOSED ENHANSMENT

- **Advanced Analytics and AI:**

  Enhance the system's analytics capabilities by incorporating advanced algorithms and artificial intelligence (AI) techniques. This would enable the system to perform more sophisticated data analysis, identify patterns, and generate actionable insights for improved campus security and traffic management.

- **Integration with Mobile Apps:**

  Develop a mobile application that allows authorized users, such as students, faculty, and staff, to register their vehicles and receive virtual permits. The app can provide real-time notifications about parking availability, traffic updates, and personalized alerts, enhancing the overall user experience and reducing administrative burdens. Integration with Smart Parking

- **Systems:**

  Integrate the automated vehicle monitoring system with smart parking systems that utilize sensors and real-time data to efficiently manage parking spaces. This integration would enable seamless tracking of vehicle movement from entry to parking spaces, optimizing parking allocation and reducing congestion. Integration with Public Transportation Systems: Collaborate with local public transportation authorities to integrate the automated system with public transit systems. This would provide real-time information on bus schedules, routes, and availability, facilitating multimodal transportation options and reducing reliance on private vehicles.

- **Biometric Authentication:**

  Explore the use of biometric authentication methods, such as fingerprint or facial recognition, to enhance the system's security and prevent unauthorized access. This would further strengthen the authentication process, ensuring that only authorized individuals gain access to the campus premises.

CHAPTER 7: CONCLUSION

Although some campuses have implemented automatic gates or barriers, these systems often come with limitations, including high installation and maintenance costs and inadequate real-time monitoring and analysis capabilities. However, with the proposed enhancements, campuses can overcome these limitations and reap the benefits of automated systems.

The proposed enhancements, such as advanced analytics and AI, integration with mobile apps and smart parking systems, biometric authentication, and integration with public transportation and emergency response systems, will transform the way campuses monitor and manage vehicular traffic. These enhancements will provide real-time monitoring, enhance security, optimize traffic flow, improve user experience, and align with sustainability goals.

CHAPTER 8: BIBILOGRAPHY

1. https://www.analyticsvidhya.com/blog/2021/12/vehicle-detection-and-counting-system-using-opencv/

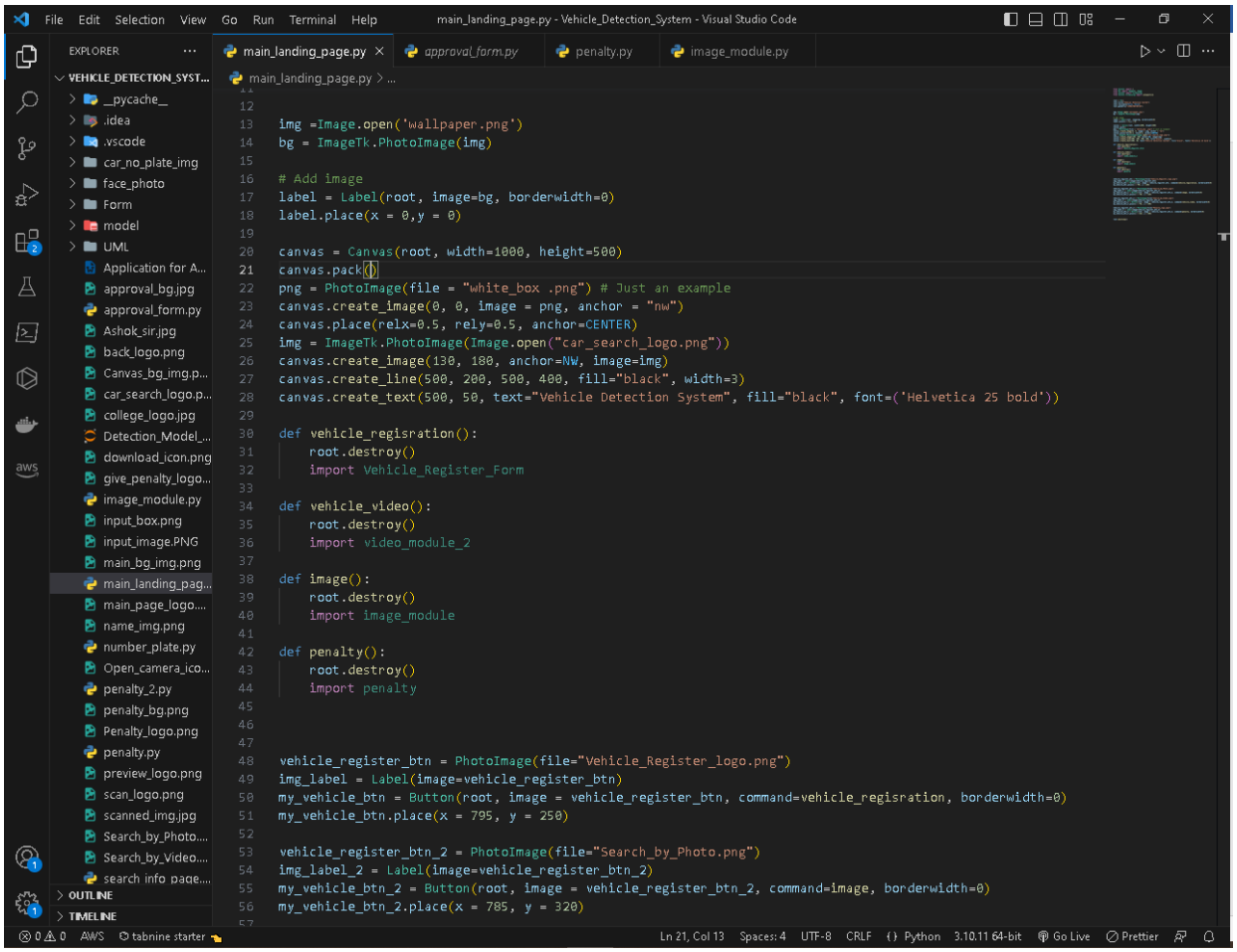
2. https://www.axiomtek.com/Default.aspx?MenuId=Solutions&FunctionId=SolutionView&ItemId=36&Title=Vehicle+Detection+System

3. https://etrr.springeropen.com/articles/10.1186/s12544-019-0390-4

4. https://www.sciencedirect.com/topics/computer-science/vehicle-detection

CHAPTER 9: SAMPLE CODE

Main Page Coding: -




```python
img =Image.open('wallpaper.png')
bg = ImageTk.PhotoImage(img)

# Add image
label = Label(root, image=bg, borderwidth=0)
label.place(x = 0,y = 0)

canvas = Canvas(root, width=1000, height=500)
canvas.pack()
png = PhotoImage(file = "white_box .png") # Just an example
canvas.create_image(0, 0, image = png, anchor = "nw")
canvas.place(relx=0.5, rely=0.5, anchor=CENTER)
img = ImageTk.PhotoImage(Image.open("car_search_logo.png"))
canvas.create_image(130, 180, anchor=NW, image=img)
canvas.create_line(500, 200, 500, 400, fill="black", width=3)
canvas.create_text(500, 50, text="Vehicle Detection System", fill="black", font=('Helvetica 25 bold'))

def vehicle_regisration():
    root.destroy()
    import Vehicle_Register_Form

def vehicle_video():
    root.destroy()
    import video_module_2

def image():
    root.destroy()
    import image_module

def penalty():
    root.destroy()
    import penalty



vehicle_register_btn = PhotoImage(file="Vehicle_Register_logo.png")
img_label = Label(image=vehicle_register_btn)
my_vehicle_btn = Button(root, image = vehicle_register_btn, command=vehicle_regisration, borderwidth=0)
my_vehicle_btn.place(x = 795, y = 250)

vehicle_register_btn_2 = PhotoImage(file="Search_by_Photo.png")
img_label_2 = Label(image=vehicle_register_btn_2)
my_vehicle_btn_2 = Button(root, image = vehicle_register_btn_2, command=image, borderwidth=0)
my_vehicle_btn_2.place(x = 785, y = 320)
```