Tribhuvan University

Institute of Science and Technology

Himalaya College of Engineering

A Final Year Project Report

On

"Parking Solution using Automatic Number Plate Recognition."

Submitted by:

Abin Timilsina [24042/076]

Kritika Sharma [24053/076]

Ruza Kansakar [24067/076]

Submitted to:

Department of Computer Science and Information Technology

In partial fulfillment of the requirements for the bachelor's in computer science and information technology

Under the supervision of

Mr. Bhupendra Ram Luhar

Acknowledgment

We extend our heartfelt appreciation to everyone who has played a crucial role in the development of this project proposal on "Parking Solution using Automatic Number Plate Recognition”. Numerous individuals have contributed to various capacities, guiding us towards the creation of a robust project proposal. Our gratitude goes to our esteemed institution, "Himalaya College of Engineering," and its management for providing us with this valuable opportunity and unwavering support throughout the project.

We express our profound thanks to our Project Supervisor, **Mr. Bhupendra Ram Luhar** , for his invaluable guidance and insightful suggestions. Additionally, we acknowledge the contributions of the Head of Department, **Er. Himal Chand Thapa** and all those who have offered their assistance, both directly and indirectly, in this endeavor. We apologize if we inadvertently omitted anyone deserving of recognition.

Abstract

This project is concerned on the development of a comprehensive Parking Solution using Automatic Number Plate Recognition (ANPR) technology, integrating cutting-edge methodologies such as the YOLOv8 (You Only Look Once) model for efficient license plate detection and Easy OCR for accurate character recognition. The integration of YOLOv8 ensures real-time and precise identification of vehicles within parking areas, optimizing the overall system's responsiveness. Furthermore, Easy OCR facilitates swift and accurate extraction of alphanumeric characters from license plates, contributing to seamless identification and recording. The project aims to address the prevalent challenges in parking management by enhancing security, reducing manual intervention, and improving overall efficiency. By automating the license plate recognition process, the system is poised to expedite the parking experience for users, providing a streamlined and user-friendly solution. The implementation of such a technology-driven parking solution is anticipated to significantly reduce operational bottlenecks, enhance security measures, and contribute to the optimization of parking space utilization. This project aligns with the broader objective of leveraging advanced technologies to create innovative solutions for contemporary urban challenges, ultimately offering a reliable and efficient parking solution for diverse environments.

Keywords*: ANPR, easyocr, machine learning, vehicle detection, YOLO*

Table of Contents

List of Abbreviations

ANPR	Automatic Number Plate Recognition
OCR	Optical Character Recognition
ROI	Region of Interest
YOLO	You Only Look Once

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Introduction

Parking Solution using Automatic Number Plate Recognition (ANPR) technology has surged in popularity in recent years due to its wide range of benefits for various applications like traffic management, intelligent parking, toll automation, and surveillance. ANPR is a technique designed to read vehicle number plates without human intervention using high speed image capture with supporting illumination, detection of characters within the images provided, verification of the character sequences as being those from a vehicle numberplate, and character recognition to convert the image to text, ending up with a set of metadata that identifies an image containing a vehicle numberplate and the associated decoded text of that plate.

Automatic Number Plate Recognition (ANPR) was first implemented in the 1990s and since then it has come a long way to become more efficient and cost effective. These features have enabled large scale implementation in law enforcement in several countries. However, Automatic Number Plate Recognition (ANPR) still faces some challenges. Vehicle owners have used a variety of techniques to evade Parking Solution using Automatic Number Plate Recognition (ANPR) systems and road-rule enforcement cameras in general. One method increases the reflective properties of the lettering and makes it more unlikely for the system to locate the plate or produce a high enough level of contrast to be able to read it. There are many difficulties that Parking Solution using Automatic Number Plate Recognition (ANPR) systems may face such as poor resolution, poor elimination conditions, blurry inputs, plate occlusion, different font size and variety of plate structures. Image acquisition is quite challenging from vehicles image due to the viewpoint change, when vehicle bodies and license plate have similar color, multi style plate formats and non-uniform outdoor illumination conditions.

In this project, we created a Parking Solution leveraging Automatic Number Plate Recognition (ANPR) that integrates YOLOv8 for object detection and EasyOCR for Optical Character Recognition (OCR). As a vehicle approaches the entrance, a YOLOv8-equipped camera captures the license plate, and the identified region is processed by EasyOCR for character recognition. The recognized plate information is cross-referenced with a user database for authentication, and if successful, the system allows the driver through. Vehicles with registered plates can automatically enter parking areas while non-registered vehicles will be charged extra at check in and check out. The exit process involves automatic plate recognition, billing based on parking duration, and the opening of the exit gate upon payment confirmation. This Parking Solution using Automatic Number Plate Recognition (ANPR) enhances efficiency, security, and user experience, offering seamless and convenient parking processes. Privacy measures, scalability, and integration with other systems are essential considerations for successful implementation.

1.2. Problem Statement

Traditional parking systems face a great number of challenges that impede their efficiency and user satisfaction. Primarily, reliance on manual ticketing and entry processes results in slow and error-prone operations, leading to long queues, delays, and increased labor costs. Moreover, these systems often lack robust security measures, leaving them vulnerable to unauthorized access and misuse. The absence of efficient authentication measures can contribute to security breaches and instances of unauthorized parking. Manual billing processes are prone to inaccuracies, creating potential disputes and revenue loss for parking facility operators. Additionally, traditional systems may lack user-friendly features, such as mobile app integration or automated payment options, diminishing overall user convenience and satisfaction. Lastly, the environmental impact cannot be overlooked, as inefficient parking systems contribute to increased fuel consumption and emissions due to prolonged wait times, inefficient circulation within parking lots, and congestion around entry and exit points. The integration of modern technologies, in the solution incorporating Parking Solution using Automatic Number Plate Recognition (ANPR), YOLOv8, and EasyOCR, aims to address these challenges comprehensively by automating processes, enhancing security measures, and ultimately improving the overall user experience in parking facilities.

1.3. Objectives

The main objective of Parking Solution using Automatic Number Plate Recognition are:

1. To utilize YOLOv8 detection model for automatic license plate recognition of the uploaded video.

2. To introduce time-based pricing for registered vehicles during check-in and check-out.

1.4. Scope and Limitations

The integration of advanced technologies like Parking Solution using Automatic Number Plate Recognition (ANPR) and YOLOv8 in parking facilities can pose challenges, particularly for smaller establishments with budget constraints due to substantial upfront costs for hardware, software, and system integration. Despite the sophistication of YOLOv8 and EasyOCR, there may be limitations in recognition accuracy, especially in adverse conditions like poor lighting or inclement weather. Additionally, the use of Automatic Number Plate Recognition (ANPR) raises privacy concerns, as it involves capturing and processing license plate information, necessitating strict compliance with privacy regulations and addressing public worries about data security. Furthermore, the system might have limited coverage for non-standard vehicles, such as motorcycles or those with unconventional plate placements, potentially excluding specific user categories.

The entrance process involves capturing vehicle video footage upon entry, utilizing the YOLOv8 model for detetction and EasyOCR for character recognition of license plates. Subsequently, the recognized plate information undergoes authentication by cross-referencing with a user database. Registered vehicles are granted seamless access to parking areas, while non-registered ones are charged based on parking duration during both check-in and check-out. The user database is consistently managed, ensuring up-to-date information for existing registrations and accommodating new entries. At the exit, automatic plate recognition is employed for registered vehicles, with billing determined by parking duration for both registered and non-registered vehicles. The exit gate opens upon payment confirmation, streamlining the overall process.

1.5. Development Methodology

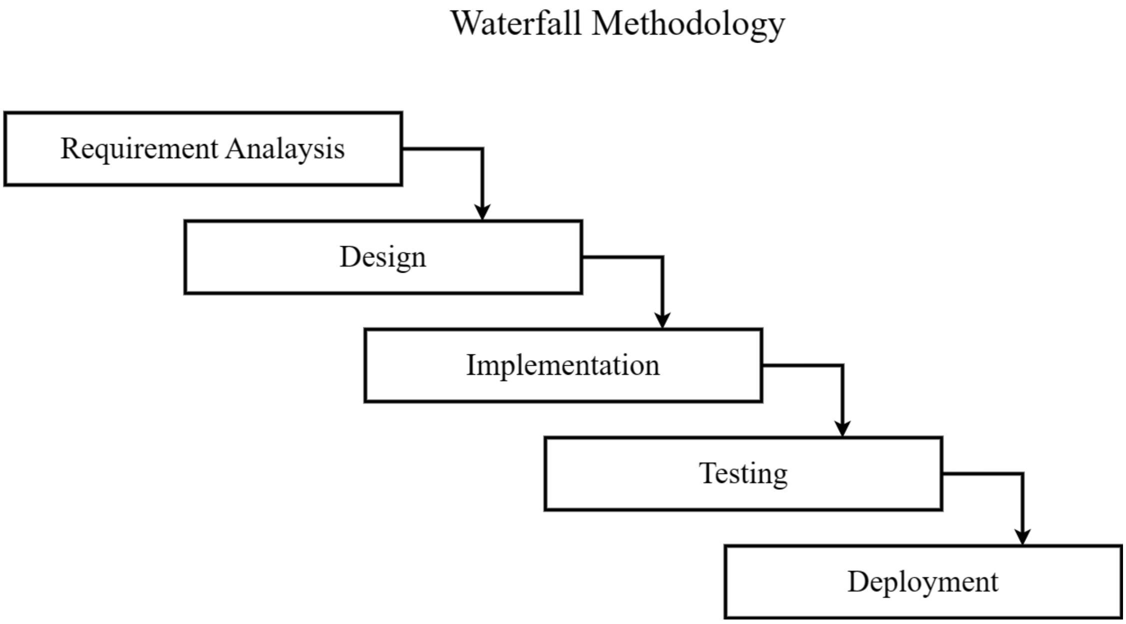


Figure 1.1: Waterfall Model for project development

The Waterfall Model is a well-established software development methodology that progresses through distinct phases in a linear and sequential manner. It begins with Requirements Gathering, where stakeholders' needs and project scope are documented. This phase focuses on understanding end-users' expectations and defining the project's objectives. Next is System Design, where the system architecture and design are planned based on the gathered requirements. Design decisions regarding software components, databases, and interfaces are made during this phase. Implementation follows, during which the actual coding and development of the software take place according to the design specifications. After implementation, encompassing various types such as unit testing and acceptance testing to ensure the software meets the specified requirements and is free of defects. Deployment occurs once testing is successful, and the software is released to end-users after installation, configuration, and deployment activities are completed. Finally, the Maintenance phase involves ongoing updates, enhancements, and bug fixes to address issues and accommodate changing requirements after deployment. This sequential approach provides a structured framework for software development, but it can be inflexible to changes late in the process and requires thorough documentation at each stage.

1.6. Report Organization

Chapter 1 introduces the rising significance of Automatic Number Plate Recognition (ANPR) technology and proposes an advanced Parking Solution integrating YOLOv8 and EasyOCR to address challenges in traditional parking systems, aiming to enhance efficiency, security, and user experience while acknowledging potential limitations and outlining specific objectives and scope.

Chapter 2 reviews existing literature on ANPR systems, highlighting diverse approaches such as deep learning models, machine learning techniques, and surveys. Studies showcase methods like EasyOCR plate recognition with high accuracy, cost-efficient systems utilizing infrared cameras, and comprehensive surveys evaluating ANPR methodologies.

Chapter 3 outlines the system analysis requirements for the created parking solution using ANPR, encompassing functional and non-functional aspects.

Chapter 4 gives information about system design including database design and interface dialogue design and algorithms.

Chapter 5 includes implementation and tools used.

Chapter 6 includes the working of the project along with the task accomplished and tasks yet to be accomplished.

Chapter 2: Background Study and Literature Review

2.1. Background Study

Automatic Number Plate Recognition (ANPR) technology has become increasingly prevalent in recent years due to its wide range of applications in intelligent transportation systems (ITS) [1]. ANPR offers a multitude of benefits, including traffic management, toll automation, security surveillance, and parking management. It functions by capturing vehicle license plates using high-resolution cameras, followed by image processing techniques for character recognition and extraction of the alphanumeric sequence .

ANPR systems have undergone significant advancements in recent times. Deep learning models, particularly Convolutional Neural Networks (CNNs), have emerged as a powerful tool for license plate detection and character recognition, achieving high accuracy rates [2]. Studies like [3] showcase the effectiveness of deep learning models in ANPR, demonstrating superior performance compared to traditional machine learning techniques.

However, ANPR systems also face certain challenges. Factors like poor lighting conditions, adverse weather, and variations in license plate formats can hinder recognition accuracy [4]. Additionally, privacy concerns regarding data collection and usage necessitate strict adherence to data security regulations [5].

2.2. Literature Review

Several studies and work have been carried out earlier. In [6-7], a deep learning model is created to recognize the number plate using the Turkish dataset made by them. They used TensorFlow framework with the Keras deep learning library. They collected 34, 58 images of which the smearing algorithm is applied using a MATLAB program. 75% of the images were used for training, 25% for testing and 5% for validation. Since the images were taken from the real time background, they carried out several image processing techniques like median blur smoothening, Adaptive Gaussian thresholding and morphological transformations. After these preparations, the CNN model is trained using the images. The image features extracted from CNN are applied to LSTM network followed by the decryption algorithm. By this method, they achieved an overall accuracy of 96.36% for plates, 99.43% for numbers, 99.05% for letters and 99.31% for all the characters.

In [8], an automatic number plate recognition system using a machine learning approach is developed. They got the input from an Infrared camera followed by contrast enhancement and noise reduction as preprocessing steps. Then they localized the number plate in the image by finding out the Region of Interest (ROI). After that, contour tracing is applied to get the salient features of the image. Then Canny's edge detection is done to find out the edges of the characters in the number plate. Finally, segmentation is applied to separate the characters. The individual characters are recognized using pattern matching by Artificial Neural Networks (ANN). The whole system was developed using MATLAB software. The author describes this system as a cost efficient and accurate system.

A survey was carried out [9] on various methodologies used in implementing Automatic Number Plate Recognition (ANPR). The authors took nearly 78 reference papers and evaluated their accuracy results. The basic steps in ANPR include vehicle Image capture, number plate detection, Character segmentation and Character recognition. For number plate detection the factors such as plate size, plate location, plate background and screw must be considered. The maximum accuracy for plate detection was achieved by Canny's edge detection as per the survey. Character segmentation can be implemented using image binarization, CCA (Connected Component Analysis), vertical and horizontal projection which produces better results. It is followed by character recognition which is usually done by Artificial Neural Networks, template matching or Optical Character Recognition (OCR) techniques. The maximum accuracy for Character recognition was achieved by tesseract OCR with 98.7% results.

In [10], a review based on Automatic Number Plate Recognition (ANPR) is carried out. The proposed system consists of a camera module, sensor, control unit, GSM, and an embedded server. It tries to block unauthorized vehicles by comparing the vehicle database saved already. The images captured from the camera are converted to gray scale and enhanced by adjusting the histogram. The edges are detected using Sobel's edge detection method. Then morphological image processing is done. After that, the segmentation is done on the edge detected image. Finally, the characters are recognized using a machine learning approach.

In [11], a plate recognition system using a deep learning approach is developed. They developed an OCR system with a customized dataset. The dataset was made artificially by taking some images from the internet and adding noises and backgrounds to those images. For background, SUN database and Standford database are used. For number plate detection YOLO (You Only Look Once), an object detection framework is used. For character recognition, Convolutional Neural Network (CNN) is used. The output layer of the CNN consists of 7.62 neurons for 7 characters. 10-fold-cross validation is applied at the output to find the accuracy. The overall accuracy for the CNN plate detector is 98.5% and for the CNN character recognition is 96.8%. This OCR based system gives an overall system efficiency of 94%.

In [12], a number plate recognition system is created using Convolutional Neural Network. The images taken from a camera is preprocessed by converting the RGB image to gray scale, noise removal and binarization. Then the license plate is extracted by using Connected Component method depending upon the properties such as major axis length, minor axis length, area and bounding box etc., The characters in the extracted license plate is segmented using horizontal and vertical scanning. Finally, the characters are recognized using Convolutional neural network (CNN). The dataset used to train the CNN consists of 1000 images for each 36 characters. Out of 36,000 images, 30,000 samples are used as training data and 6000 for testing data. They used a descent algorithm to minimize cross-entropy with a learning rate of 0.5. The overall accuracy obtained was 97%.

Chapter 3: System Analysis

3.1. Requirement analysis

3.1.1. Functional requirements

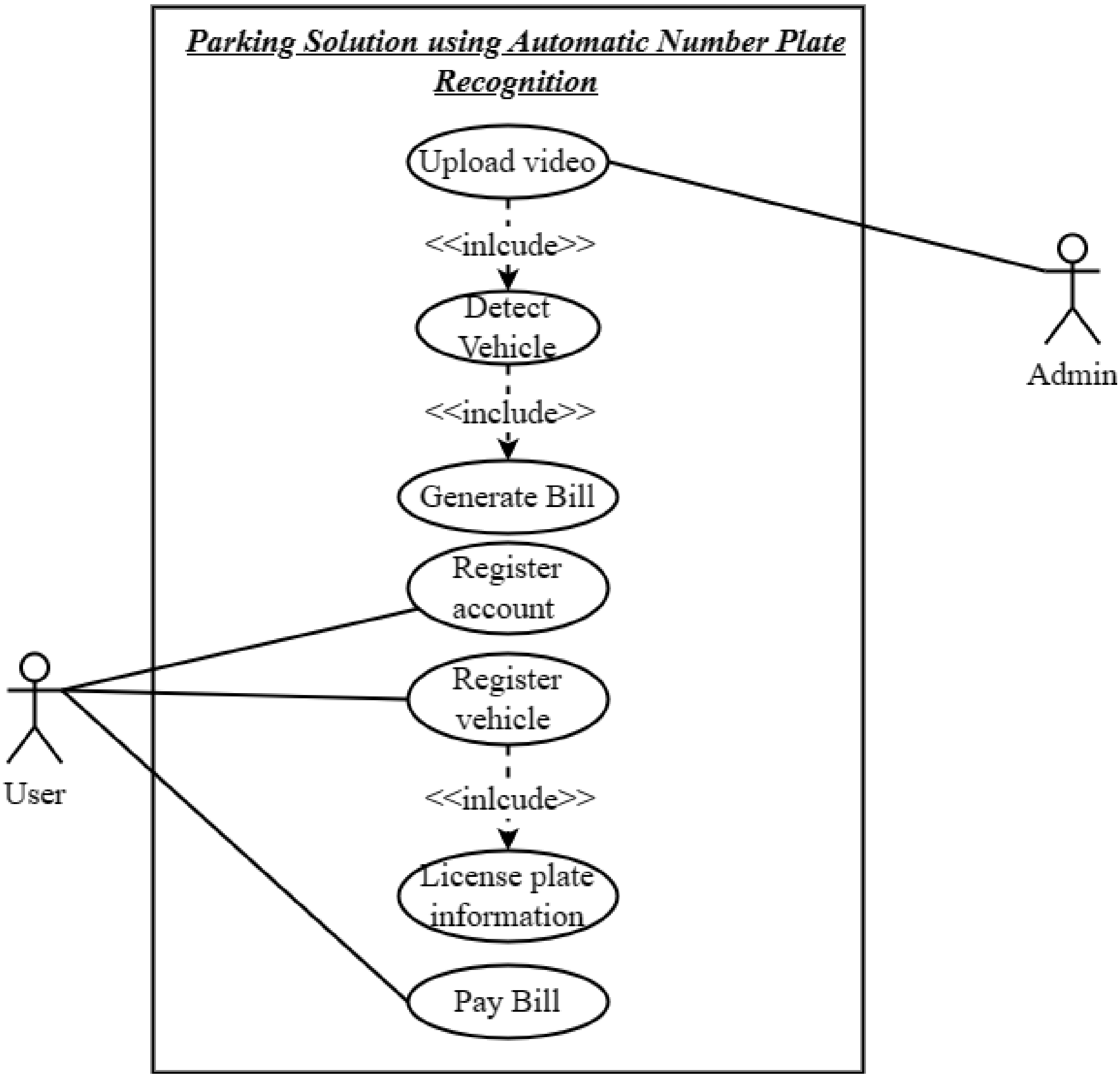


Figure 3.1: Use Case Diagram of Parking solution using ANPR

The system should facilitate user registration by allowing individuals to create an account. Users must be able to register their vehicles within the system. The registration process should detect and extract license plate information from the uploaded videos. Upon the successful detection of license plates, the system is responsible for generating bills to the user corresponding to the identified vehicles. Users should have the ability to access their generated bills within their accounts. The system must incorporate a secure payment gateway, allowing users to make payments online.

Admins are granted the privilege to upload videos containing moving vehicles for processing. These uploaded videos should be securely stored within the system and made accessible for processing. The system is responsible for generating bills to the admin corresponding to the identified vehicles.

3.1.2. Non-functional requirements

i. Performance

The system is designed for swift and accurate vehicle detection, license plate recognition, and authentication. Advanced algorithms prioritize speed and accuracy, ensuring efficient data processing to meet dynamic parking environment demands.

ii. Availability

To guarantee a seamless user experience, the system is engineered for continuous and uninterrupted operation, providing 24/7 availability. Redundancy measures have been strategically implemented to minimize downtime and maintain service reliability, ensuring users can access the system whenever needed.

iii. Security

The system prioritizes user data protection with robust security measures, including encryption for confidentiality. Multi-factor authentication enhances access control, and regular security audits address potential vulnerabilities, creating a secure environment for both user data and system functionality.

iv. Usability

The parking system is crafted with a user-centric approach, featuring a user-friendly interface designed for easy navigation and accessibility. Tailored to accommodate both registered and non-registered users, the system provides clear and concise instructions to guide users seamlessly through the registration and parking processes. Accessibility features are integrated to cater to a diverse user base, ensuring an inclusive and user-friendly experience.

3.2. Feasibility Analysis

3.2.1. Technical Feasibility

The technical feasibility assessment involves evaluating the availability and compatibility of essential technologies, such as YOLOv8 for object detection and EasyOCR for OCR, to ensure they meet the system's performance and accuracy requirements. Compatibility checks are also vital to guarantee the seamless integration of the parking solution using ANPR with existing parking management systems, databases, and other relevant technologies. Furthermore, it is imperative to verify that the system adheres to legal and regulatory requirements concerning data privacy, security, and other pertinent standards in the implementation region. This comprehensive evaluation ensures the technical viability of the parking solution using ANPR, meeting both performance expectations and regulatory compliance.

3.2.2. Economic Feasibility

The economic feasibility of implementing the Parking Solution using Automatic Number Plate Recognition (ANPR) involves a comprehensive evaluation of both initial and ongoing costs in comparison to the anticipated benefits. This assessment includes calculating the expected return on investment over a specified period, factoring in elements such as reduced manpower requirements, increased revenue from non-registered vehicles, and potential scalability benefits. A crucial aspect of this process is the performance of a cost-benefit analysis to determine the project's financial viability. This analysis will provide insights into the economic sustainability of the parking solution using ANPR, helping stakeholders make informed decisions about its implementation.

3.2.3. Operational Feasibility

Addressing the training needs for staff is essential for the effective operation and maintenance of the system. A thorough evaluation of the system's scalability concerning the increasing number of users and vehicles is crucial to ensure it can handle growth seamlessly. Additionally, assessing the feasibility of maintaining and supporting the system over time is paramount for long-term success. This includes considerations for ongoing training, technical support, and adapting the system to evolving requirements. A comprehensive approach to these aspects will contribute to the system's sustained performance and responsiveness to changing demands.

3.2.4. Project Planning and Scheduling

The work breakdown and time in week as well as days required to complete the project is shown below:

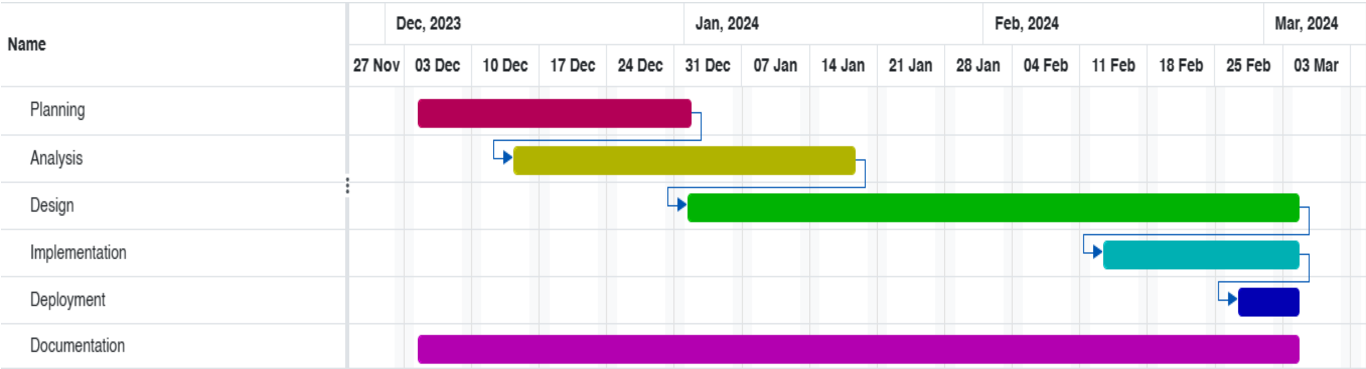


Figure 2.2: Gnatt Chart of Parking solution using ANPR

According to the timeline of our planned project, encompassing design, implementation, deployment, and documentation, is depicted in this diagram. Similarly, the Analysis phase was concluded 15 days after the completion of the planning phase. The design portion of the project is currently 95% complete, and documentation is almost finished with the overall project progress.

3.3. Analysis (Structure System)

3.3.1. Data Modeling using ER diagram.

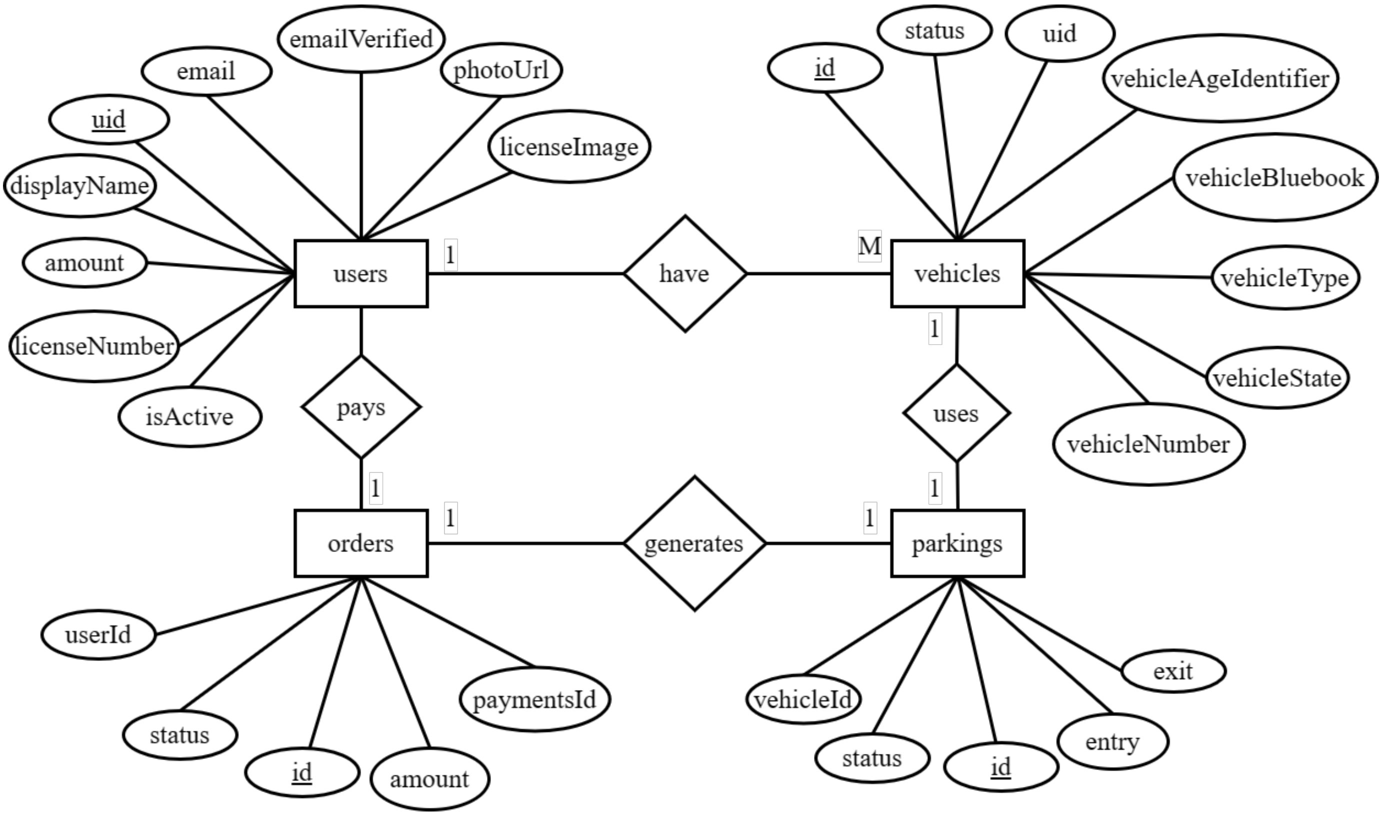


Figure 3.3: ER Diagram of Parking solution using ANPR

Figure 3.3 illustrates four entities: User, Vehicle, Order, and Parking, with associated attributes and relationships. Users can have multiple vehicles (one-to-many), and each vehicle has a one-to-one relationship with a license plate. Additionally, vehicles and registrations share a one-to-one connection. Finally, parking generates the bill using the entry and exit times of the vehicle and the bill is paid by the user.

3.3.2. Process Modeling using DFD (Data Flow Diagram)

DFD Level-0

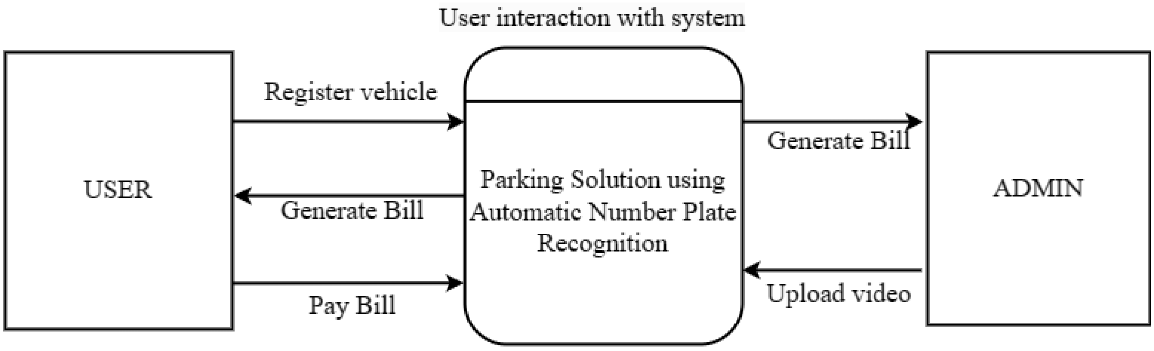
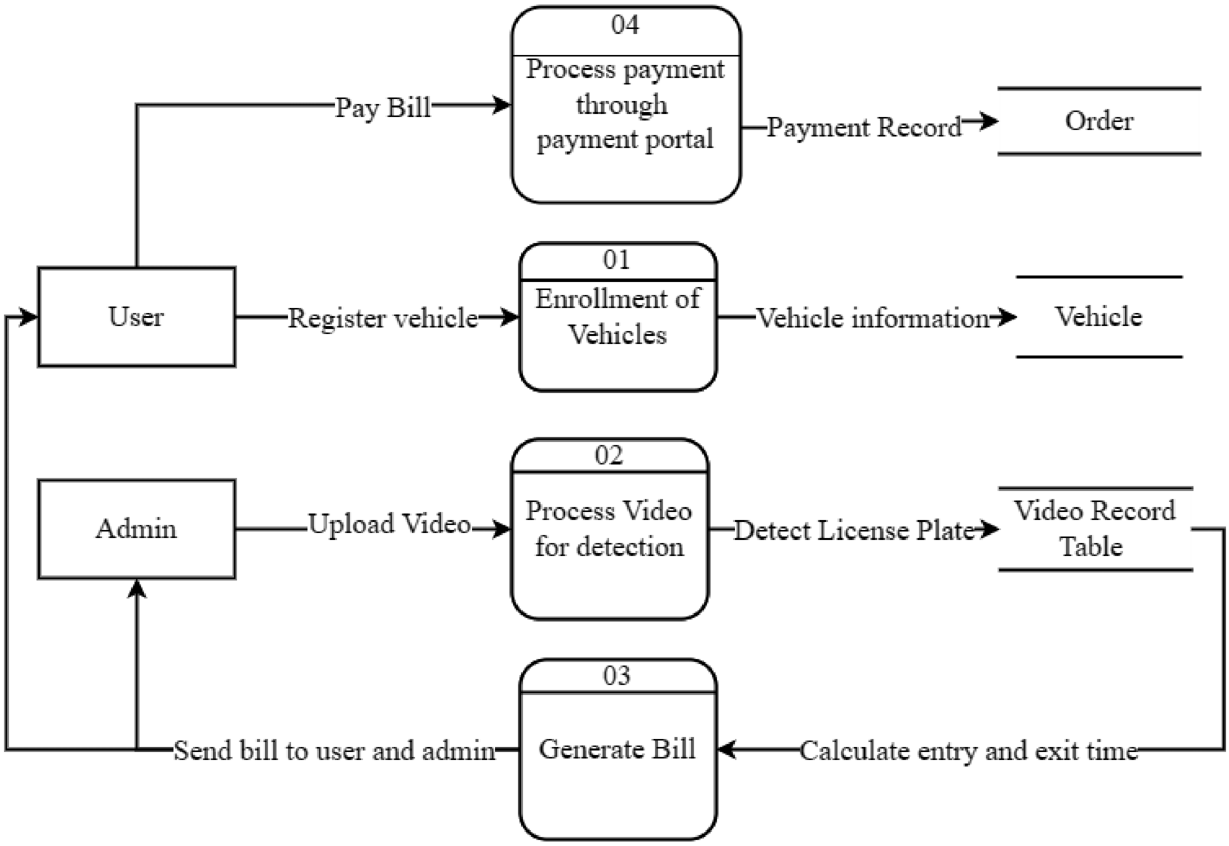


Figure 3.4: Context Diagram of Parking solution using ANPR



DFD Level-1

Figure 3.5: Level-1 Data-flow Diagram of Parking solution using ANPR

Figure 3.5 shows the sequential actions for completing a bill payment via a payment portal in a parking system. The process commences with users registering their vehicles, followed by admin uploading videos for analysis. The system then analyzes the video content to recognize license plates and fetches relevant vehicle details. Subsequently, utilizing entry and exit timestamps, the system generates a bill, which is then forwarded to both the user and administrator for payment processing.

Chapter 4: System Design

4.1. Design

4.1.1. Database Design

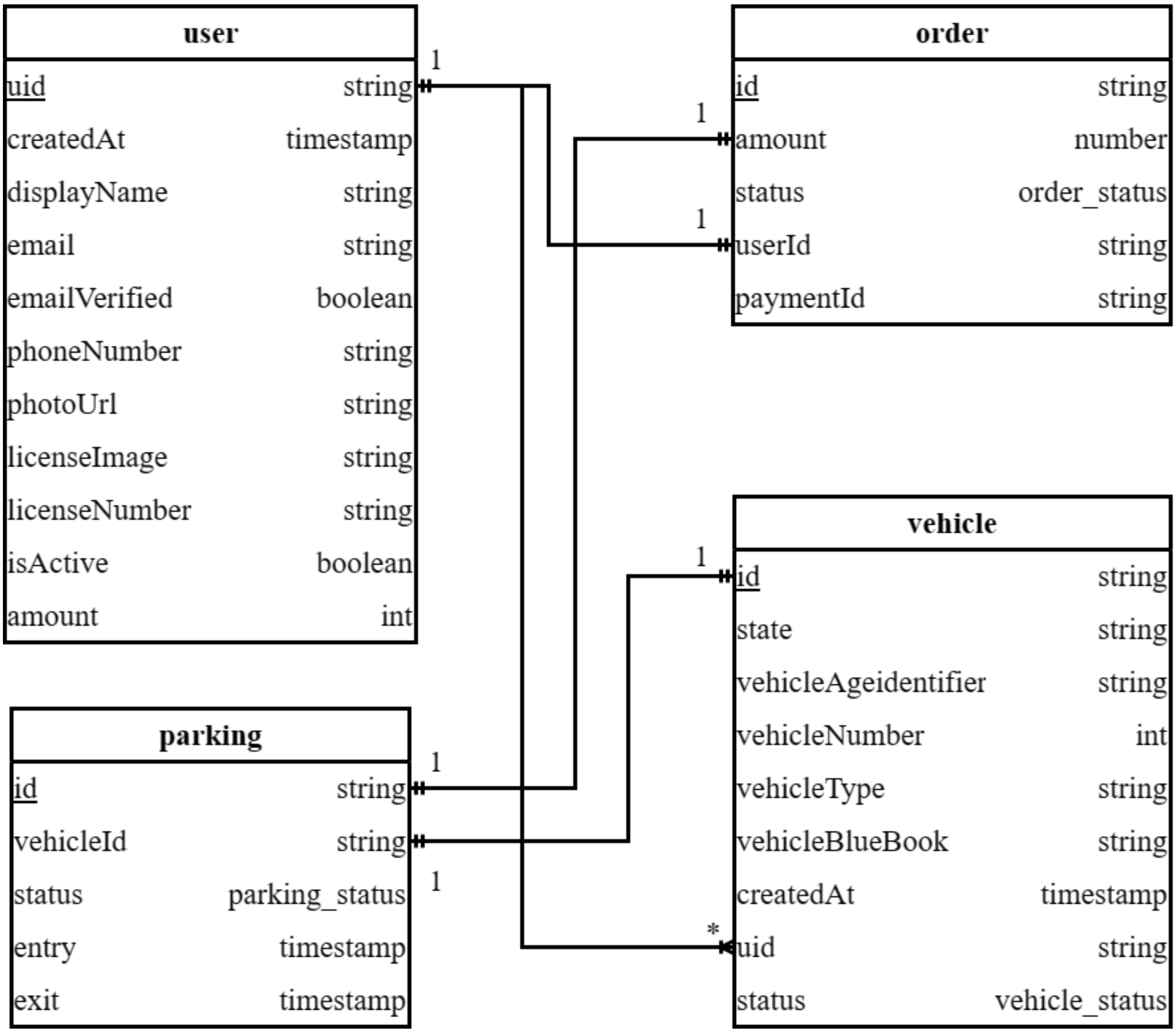


Figure 4.1: Database Design of Parking solution using ANPR

In figure 4.1, the schema defines four tables: user, vehicle, parking, and order. These tables are linked together by foreign key relationships.

4.1.2. Forms and Report Design

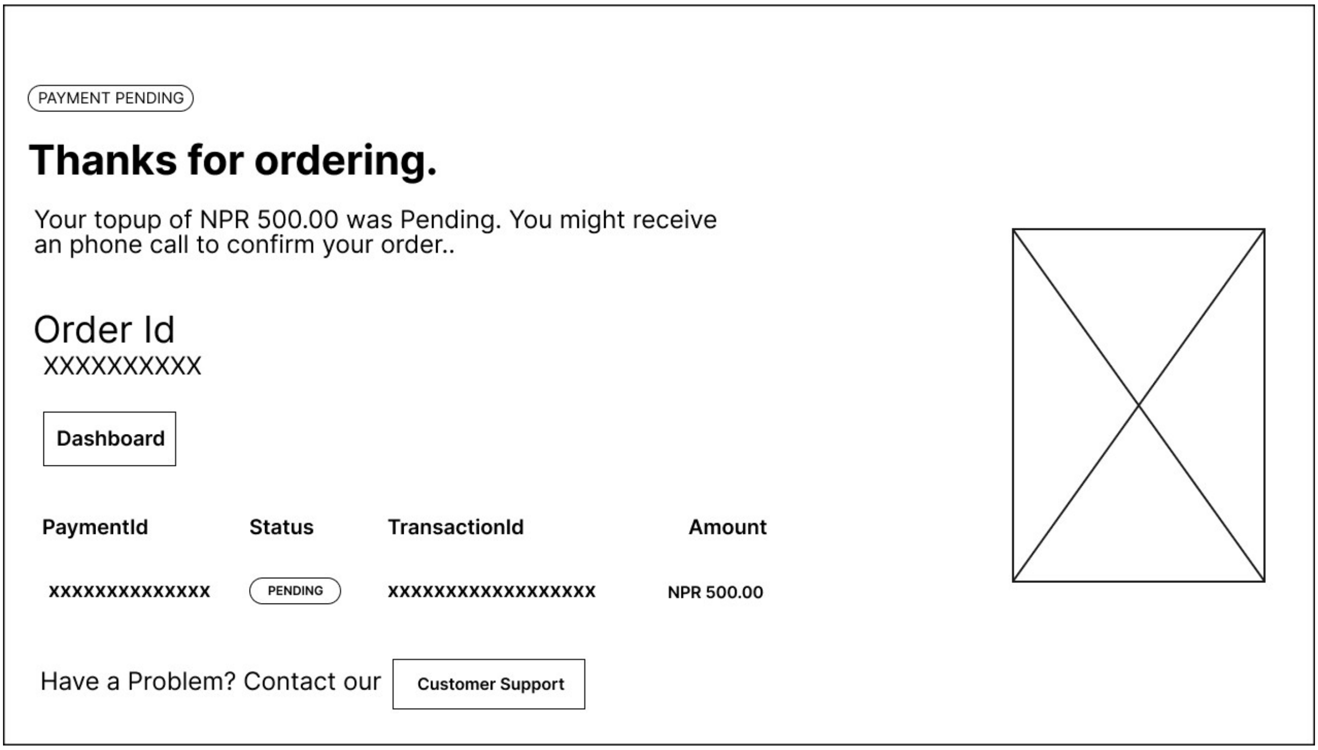


Figure 4.2: Wireframe of Payment Success Page

Figure 4.2 shows the wireframe depicting the payment success page, users are promptly informed about the status of their transaction following an order placement. The page displays vital details including the order ID, payment ID, transaction status, and the amount paid. Users can easily see whether the payment is pending or completed, ensuring transparency and clarity throughout the transaction process. Additionally, the page offers an option to contact customer service in case of any issues or queries, thereby providing users with immediate assistance and support when needed.

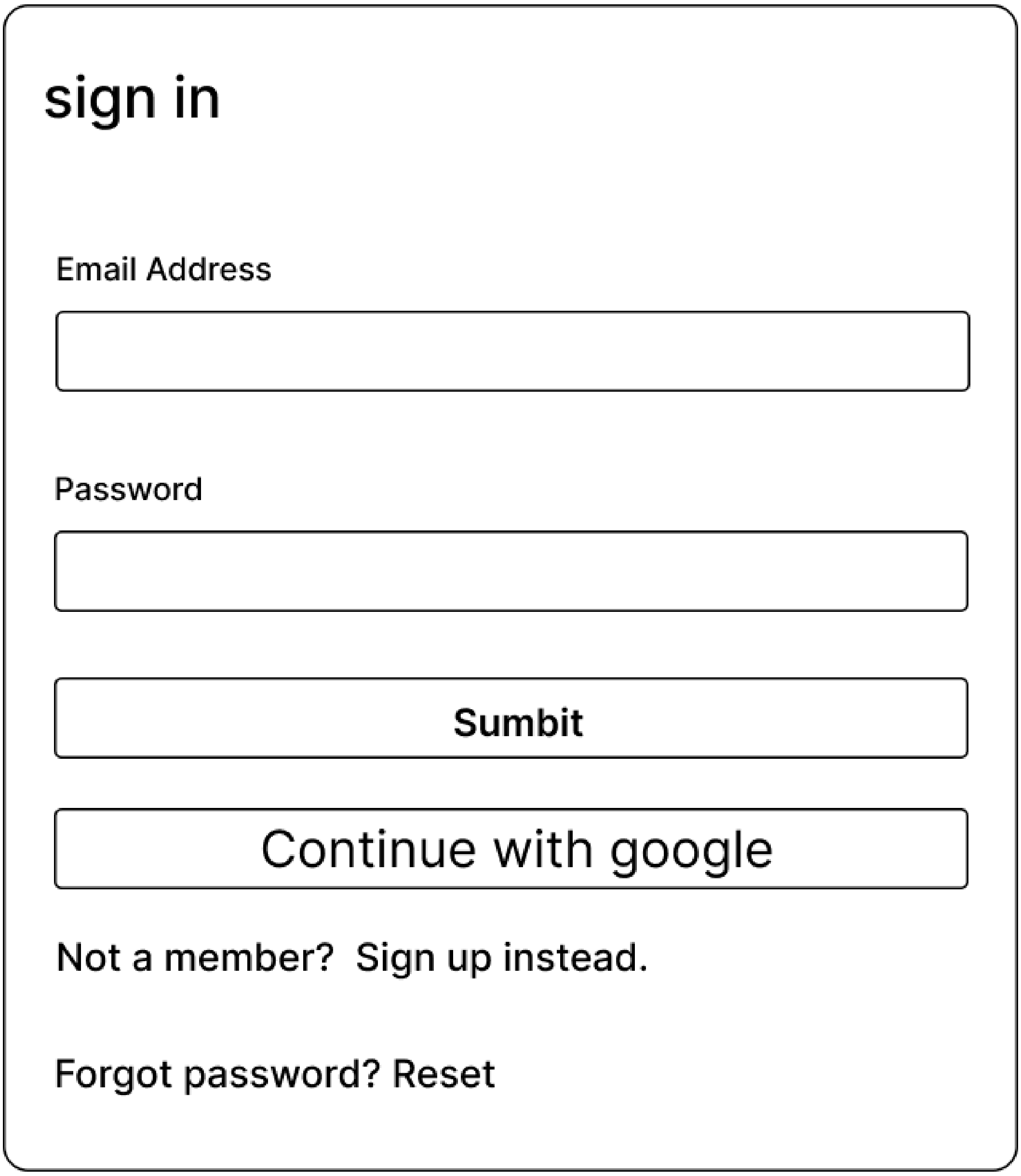


Figure 4.3: Wireframe of Sign In page

Figure 4.3 shows the wireframe of the sign-in page, users encounter a straightforward interface facilitating platform access. The form requests users to input their email address and password for logging into the system. Alternatively, users are provided with the convenience of signing in through their Google account. For individuals who are not yet registered members, the option to sign up is prominently displayed. Furthermore, in the event of a forgotten password, users are offered the opportunity to initiate a password reset, ensuring a seamless and user-friendly sign-in experience.

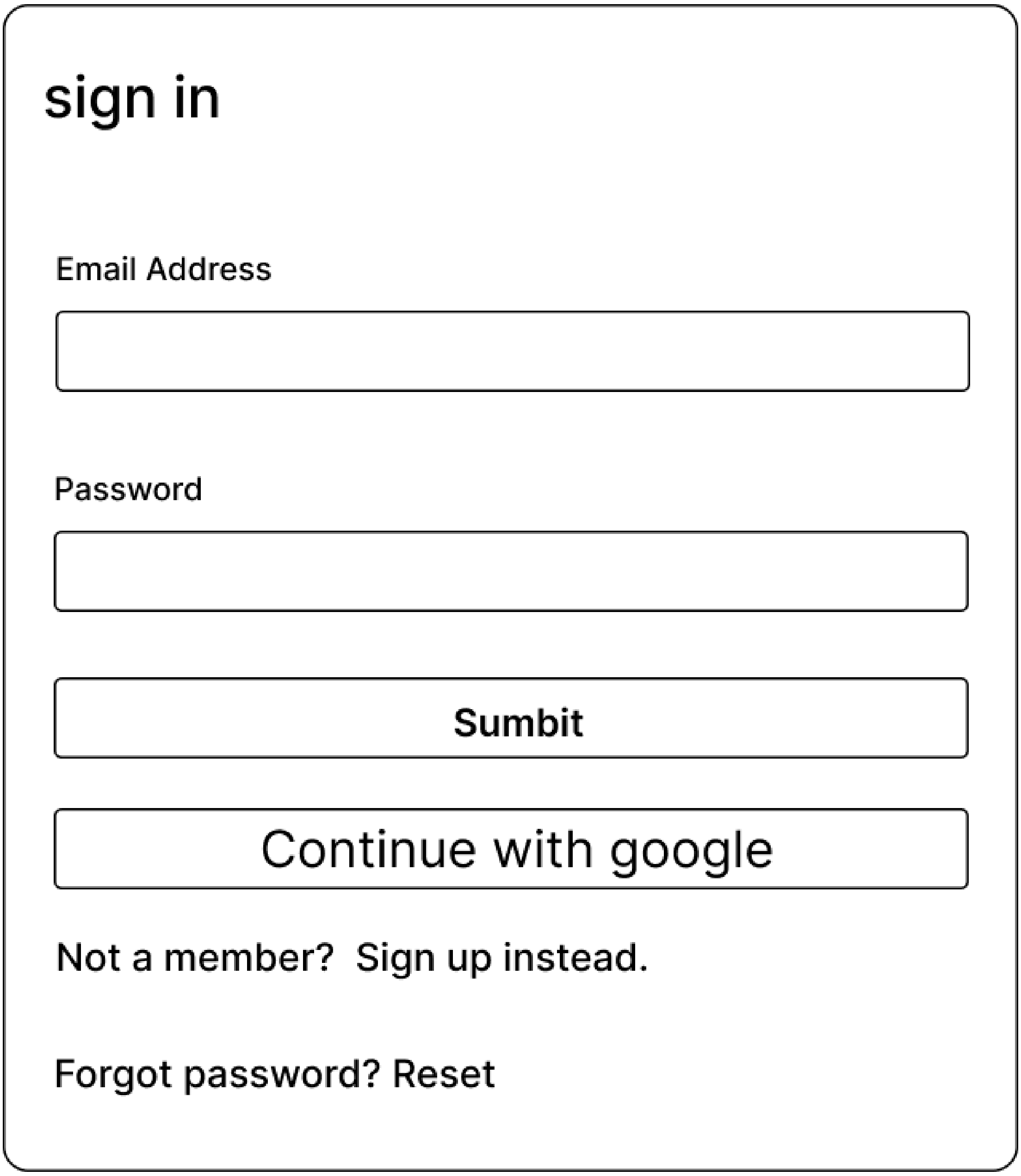


Figure 4.4: Wireframe of Sign-Up page

Figure 4.4 shows a wireframe that represents the signup page, users are presented with a simple form to register for the platform. The form prompts users to enter their email address and create a password. Upon completion of the form, users can proceed to sign up for the service, enabling them to access the platform's features and functionalities.

4.1.3. Interface and Dialogue Design

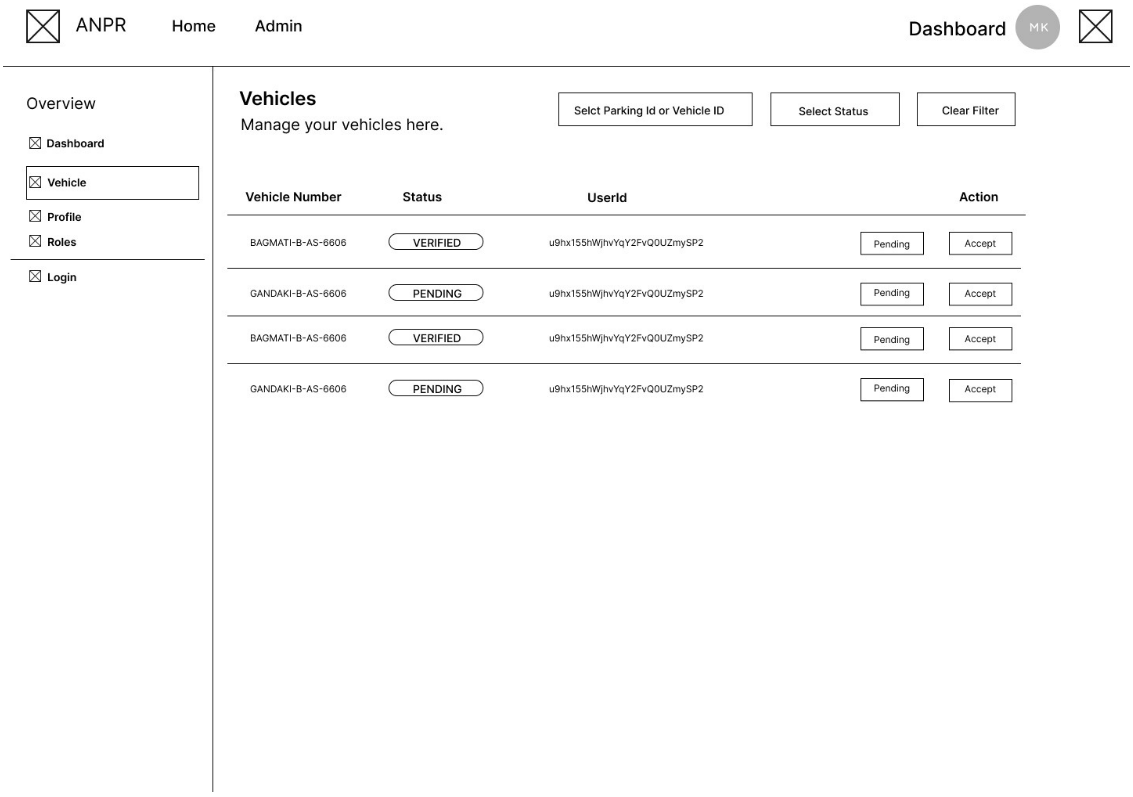


Figure 4.5: Wireframe of Vehicle Management Page

Figure 4.5 shows a wireframe model in which administrators are empowered to oversee vehicle management operations. This includes overseeing the list of vehicles entering the system, each accompanied by a unique user ID and status indication. Administrators are afforded the capability to take decisive actions, such as accepting or rejecting vehicles for inclusion within the database.

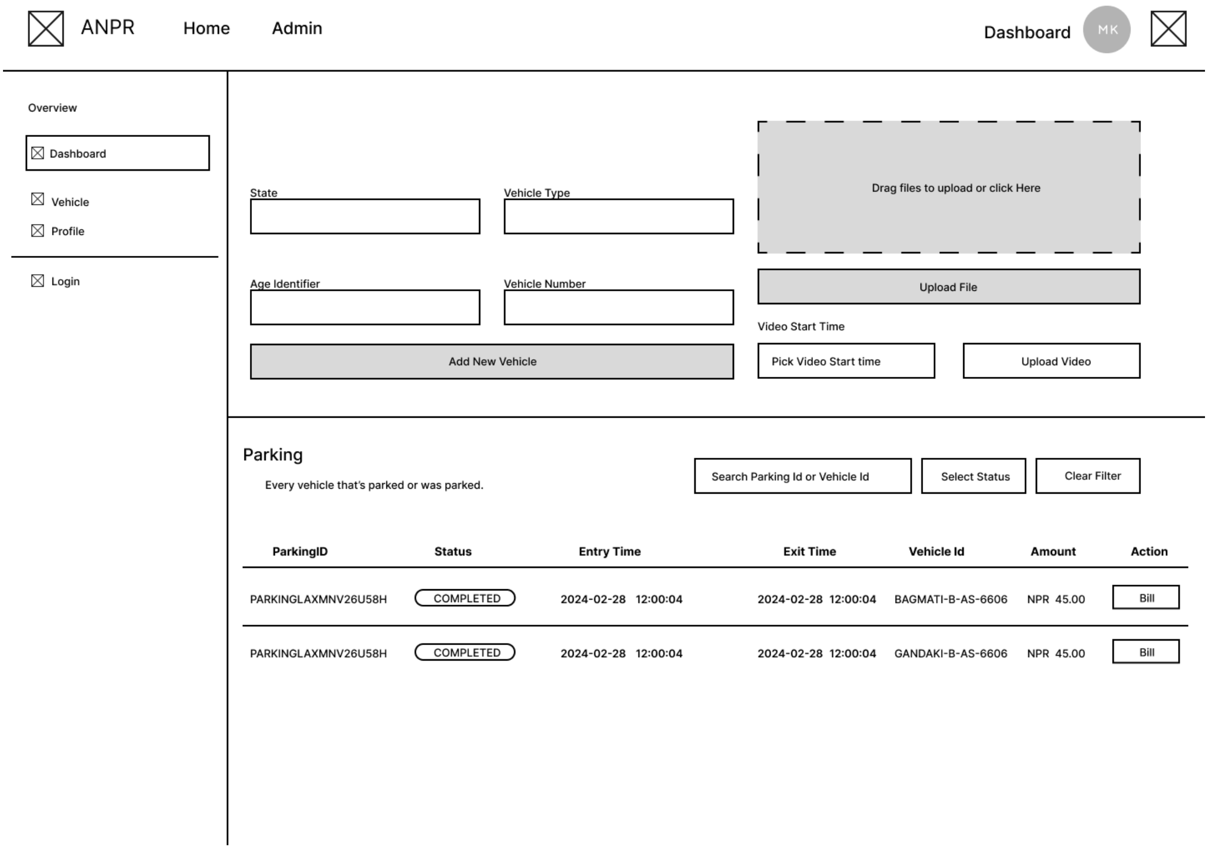


Figure 4.6: Wireframe of Admin upload video page

Figure 4.6 shows a wireframe model in which administrators possess the ability to upload videos capturing vehicle entry into the parking area, with the flexibility to select the precise start date and time. Additionally, administrators can manually input new vehicle entries. A comprehensive table format displays the status of vehicles parked within the premises, showcasing each vehicle's parking ID, status indicating whether it remains parked or has exited, entry time, exit time, vehicle ID, and corresponding user payment amount. Furthermore, the system generates a bill dialogue box for each transaction, ensuring clarity and transparency in financial transactions.

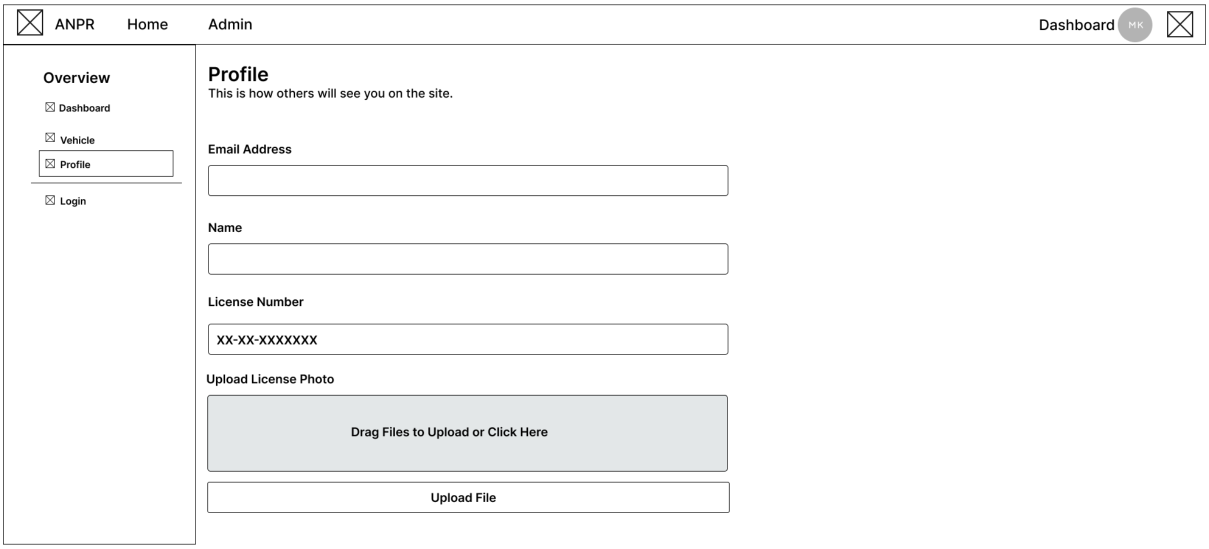


Figure 4.7: Wireframe of User profile page

Figure 4.7 shows the wireframe depiction of the user profile page, users are provided with the functionality to personalize and manage their profile information. Users can input essential details such as their email address, name, license number, and include a license photo. This curated profile serves as a representation of the user within the platform, allowing others to view their information and identity with ease.

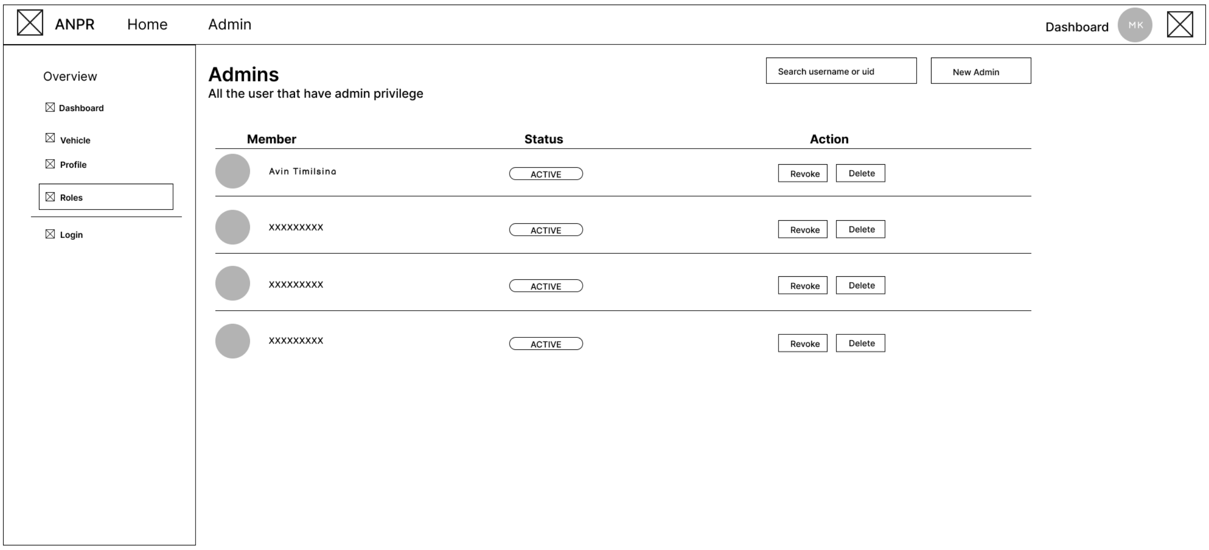


Figure 4.8: Wireframe of Admin roles page

Figure 4.8 shows admin page wireframe model, an interface is provided to display all users granted admin privileges. Each member's profile showcases their name alongside a status indicator, denoting their activity level, whether active or inactive. Administrators are equipped with actionable controls, enabling them to manage user privileges efficiently. These actions include options to revoke or delete admin privileges for specific members.

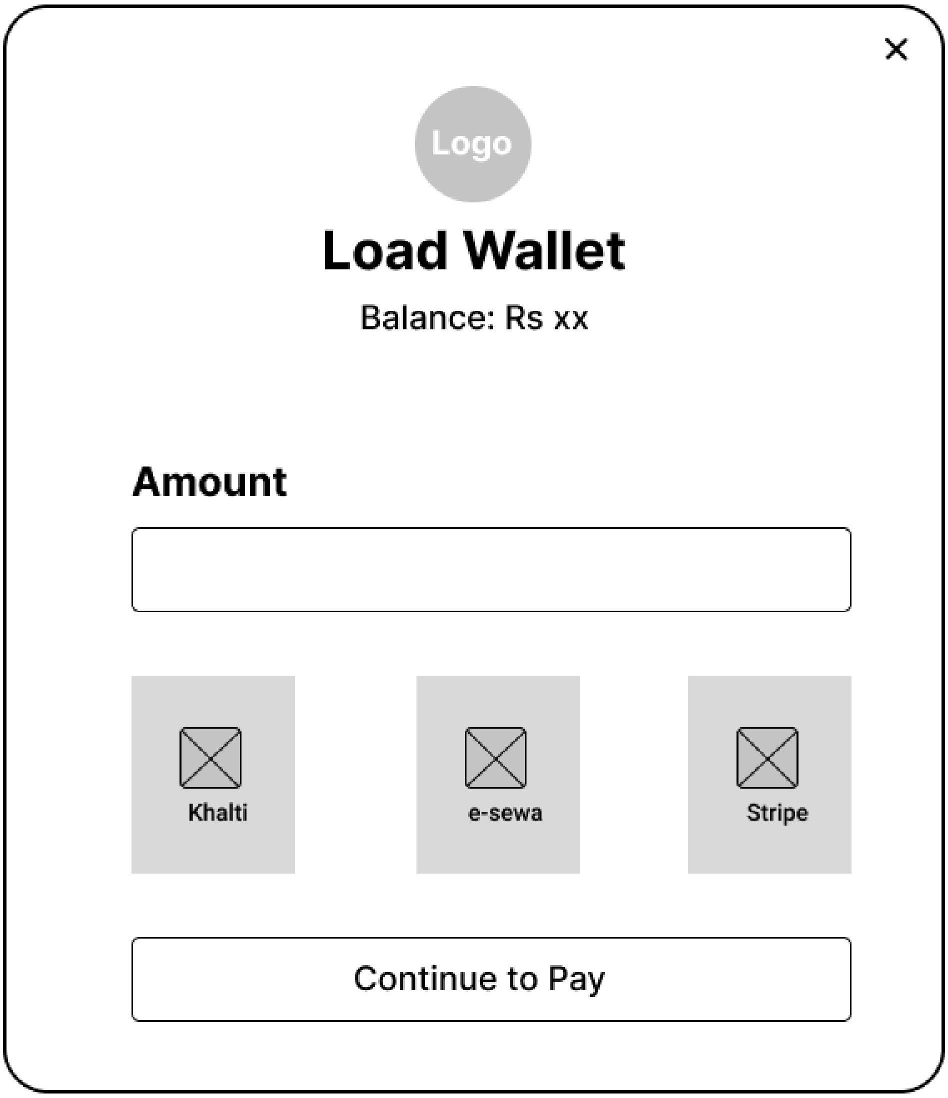


Figure 4.9: Wireframe of load wallet model

Figure 4.9 shows the wireframe for the load wallet page dialogue, users are provided with a seamless payment experience for topping up their wallet balance. Integration with trusted payment gateways such as Khalti, ESewa and Stripe enables users to conveniently add funds to their wallet, facilitating hassle-free payment for parking bills. Additionally, the dialogue displays the user's total balance, ensuring transparency and easy tracking of available funds for future transactions.

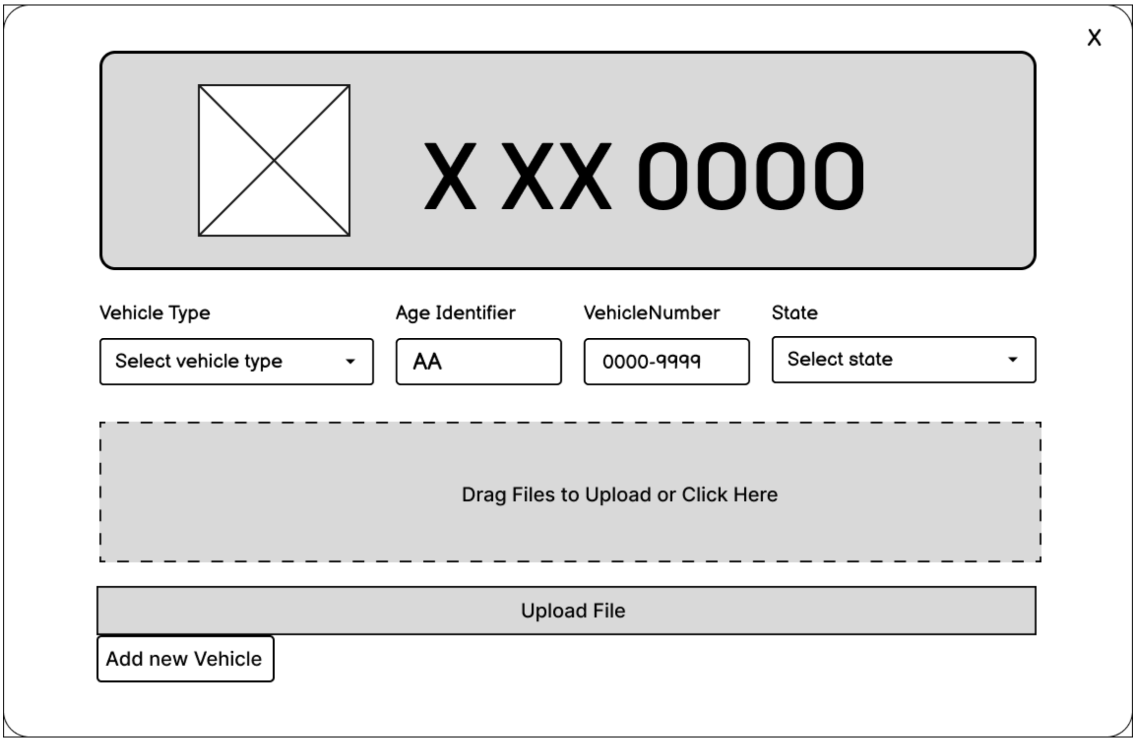


Figure 4.10: Wireframe of Vehicle registration Dialogue

Figure 4.10 shows the wireframe design of the vehicle registration dialogue, users are presented with a streamlined process for registering their vehicles. The dialogue prompts users to select their vehicle type and provide pertinent details such as a unique identifier, vehicle number, and the state of registration. Additionally, users are required to upload a photo of their license plate for verification purposes. Upon completion of these steps, users have the option to add the newly registered vehicle to their profile.

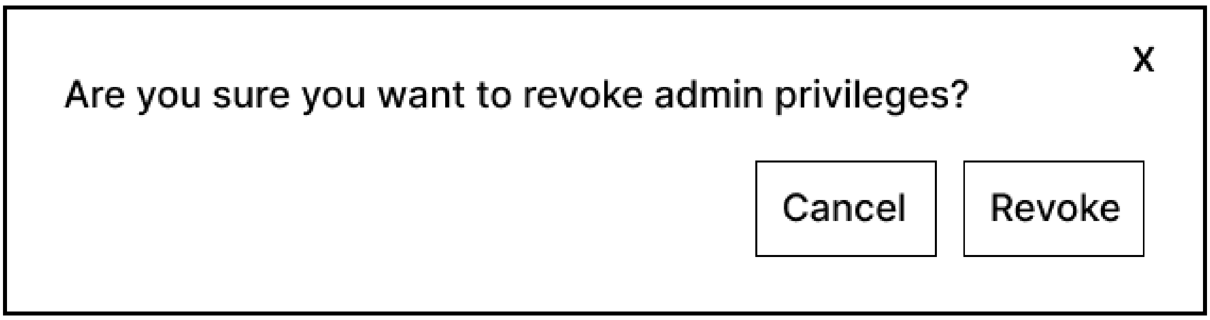


Figure 4.11: Wireframe of Admin Privileges Dialogue

Figure 4.11 shows the wireframe design of the admin privileges dialog box, The dialog box allows administrators to revoke admin privileges.

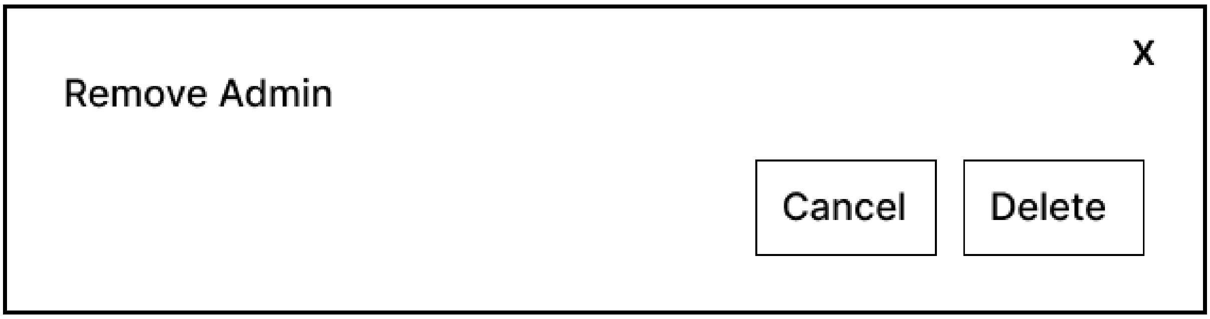


Figure 4.12: Wireframe of Remove Admin Dialogue

Figure 4.12 shows the wireframe design of the remove admin dialog box, the dialog box shows whether to remove the user from admin or not.

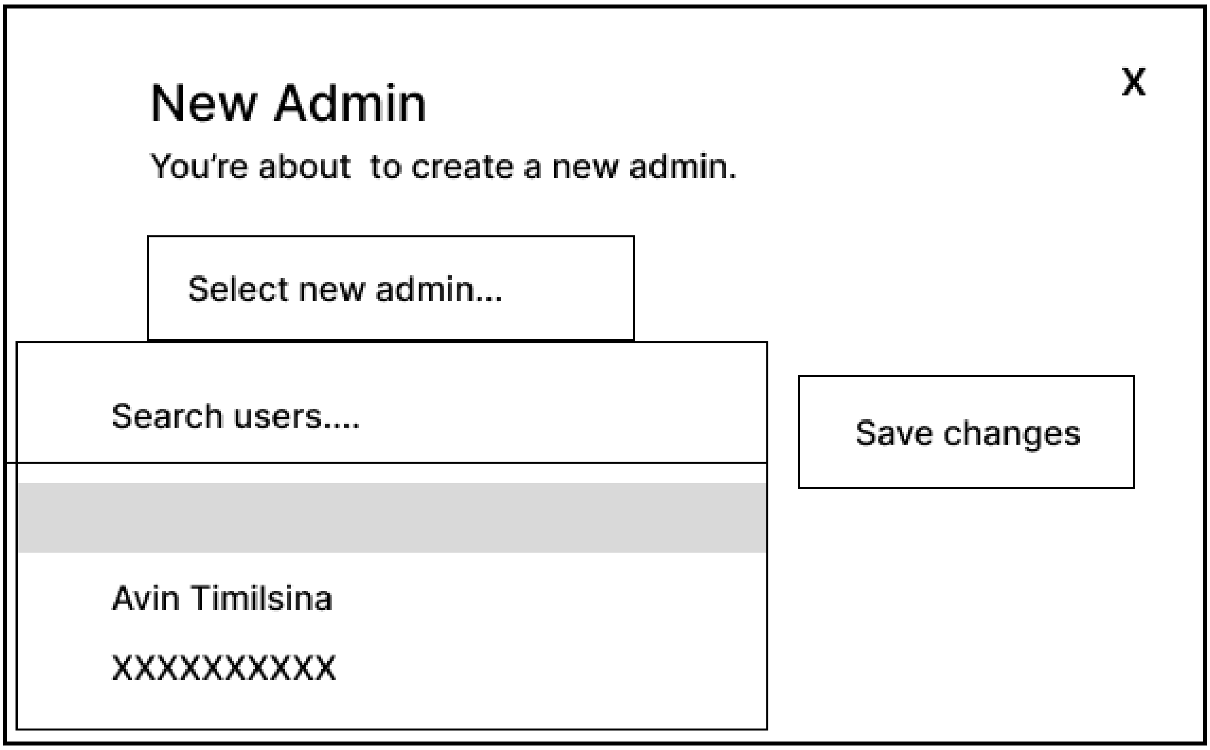


Figure 4.13: Wireframe of Add new Admin Dialogue

Figure 4.13 shows the wireframe design of the new admin dialog box. Administrators are presented with a user-friendly interface for adding additional administrators to the system. The dialog box allows administrators to select and designate new individuals for admin privileges.

4.2. Algorithm

4.2.1. YOLO Algorithm

YOLO, which stands for "You Only Look Once," is a popular object detection algorithm in computer vision and deep learning. The primary advantage of YOLOv8 is its ability to perform real-time object detection with high accuracy. This object detection is done as a regression problem.

It is used to predict various classes and bounding boxes simultaneously. Most of the license plates have different background and foreground color. Need to train the model with YOLOv8 custom weights to successfully complete the localization of the license plate. The recognition and localization of vehicle license plates is a critical task for an ANPR system.

Working of YOLO Algorithm

In the YOLO algorithm, the input image is divided into an n x n grid, and each grid cell is responsible for predicting multiple bounding boxes (typically 5). This means that the total number of bounding boxes predicted by the algorithm is n x n x 5.

Each bounding box prediction includes the coordinates of the box's center (x, y), width (w), height (h), and a confidence score. It gives a confidence score which shows how confident the algorithm is that the object to detect exists in the bounding box. The score doesn't tell the type of object but tells the confidence score if it is in that bounding box or not.

For example, there are 13*13 grid cells, and each cell detects 5 bounding boxes so total bounding boxes becomes 13*13*5 which makes 845 bounding boxes. Now the beauty of the algorithm is that these total 845 boxes were predicted all at once as the name suggests "You Only Look Once".

The simultaneous prediction of all bounding boxes in a single pass through the network is a key characteristic of YOLO, and it contributes to the algorithm's efficiency and real-time processing capabilities. This contrasts with some other object detection approaches that involve multiple stages or steps in the detection process.

YOLOv8 (You Only Look Once) is a popular object detection algorithm that operates on the principle of treating object detection as a regression problem. The basic idea behind YOLOv8 is to divide the input image into a grid and predict bounding boxes and class probabilities directly from the entire image.

Here is a simplified mathematical representation of the YOLOv8 loss function:

$$\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2 + \left(\sqrt{w_i} - \sqrt{\hat{w_i}}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h_i}}\right)^2 \right] + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C_i}\right)^2 + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c=0}^{C} \left(p_i(c) - \hat{p_i}(c)\right)^2$$

Here:

- (λ_{coord}) and (λ_{obj}) are hyperparameters that control the importance of the different components of the loss.

- ($\mathbb{1}_{\text{ij}}^{\text{obj}}$) is an indicator function that equals 1 if object (j) is present in cell (i) and 0 otherwise.

-((x_i, y_i, w_i, h_i, C_i)) are the predicted bounding box coordinates and class probabilities for cell (i).

- (($\hat{x_i}, \hat{y_i}, \hat{w_i}, \hat{h_i}, \hat{C_i}$))are the ground truth bounding box coordinates and class probabilities.

-($p_i(c)$)is the predicted probability of class (c) for cell (i), and ($\hat{p_i}(c)$) is the corresponding ground truth.

This loss function combines terms for bounding box coordinates, abjectness, and class probabilities, and it is used during the training process to update the model parameters.

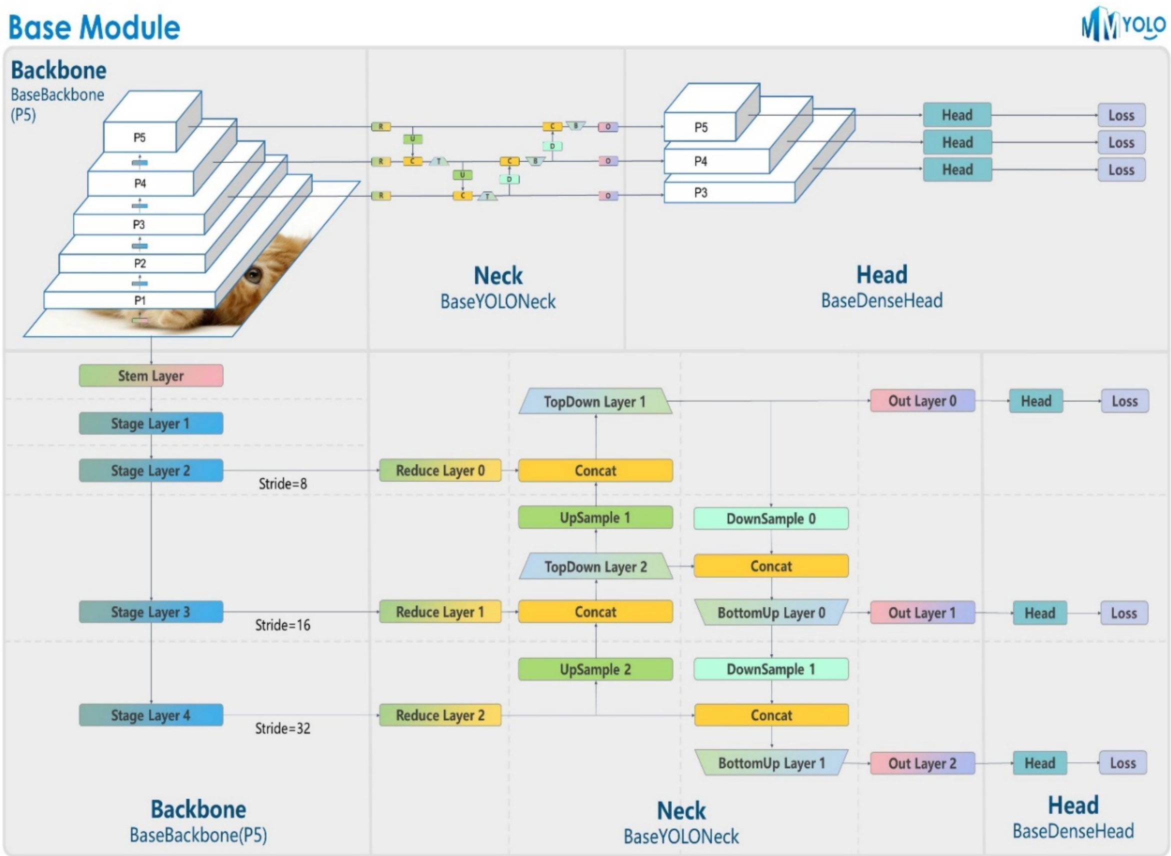


Figure 4.14: Architecture Diagram of YOLOv8

Steps of YOLOv8 detection algorithm

i. Receive a single frame from video as input to process for car and license plate detection.

ii. Pass the input frame through the backbone network, such as Darknet to extract features.

iii. Extract feature maps at different scales (e.g., P2, P3, P4, P5) using convolutional layers.

iv. Implement a feature pyramid structure like FPN to combine feature maps from different scales.

v. Use top-down and bottom-up pathways to enhance features and preserve spatial information across resolutions.

vi. Apply multiple detection heads at different levels of the feature pyramid each head predicting bounding boxes, class labels, and confidence scores for objects.

vii. Use anchor boxes technique to define potential object locations and attributes.

viii. Compute the loss for each detection head based on predicted and ground-truth bounding boxes, class labels, and confidence scores.

ix. Backpropagate the loss through the network to update the weights of the backbone, neck, and heads.
x. Post-process the predicted bounding boxes to eliminate duplicates and refine their positions.
xi. Generate the final set of detections with bounding boxes, class labels, and confidence levels.
xii. Repeat the process for multiple images or batches to perform object detection.

4.2.2. EasyOCR

EasyOCR, a Python-based Optical Character Recognition (OCR) module, stands out for its flexibility and user-friendly design, making OCR implementation accessible even to those without a background in OCR or computer vision. This module proves valuable for tasks such as data entry automation and image analysis, enabling computers to identify and extract text from photos or scanned documents. With multiple language support, pre-trained text detection and identification models, and a focus on speed and efficiency in word recognition within images, EasyOCR prioritizes user convenience. Its versatility in handling various typefaces and text layouts, coupled with a commitment to accuracy and speed, makes it a reliable choice for Python developers. By simplifying the extraction of text from photos, EasyOCR facilitates the integration of OCR capabilities into diverse Python projects, including desktop software and online applications, allowing developers to focus on the unique requirements of their products.

Working Of EasyOCR

EasyOCR operates by taking an image as input from diverse sources such as scanned documents, images, or photos. Users can specify the language(s) for character recognition to enhance accuracy. The input image undergoes preprocessing, including resizing, normalization, and noise reduction, to optimize OCR accuracy. The system then employs text detection methods to identify regions containing text, delineating bounding boxes or regions of interest. Subsequently, text recognition occurs, where OCR algorithms convert image-based text into machine-readable text. EasyOCR offers support for multiple OCR engines and models, potentially utilizing deep learning approaches for character recognition within the identified regions of the image.

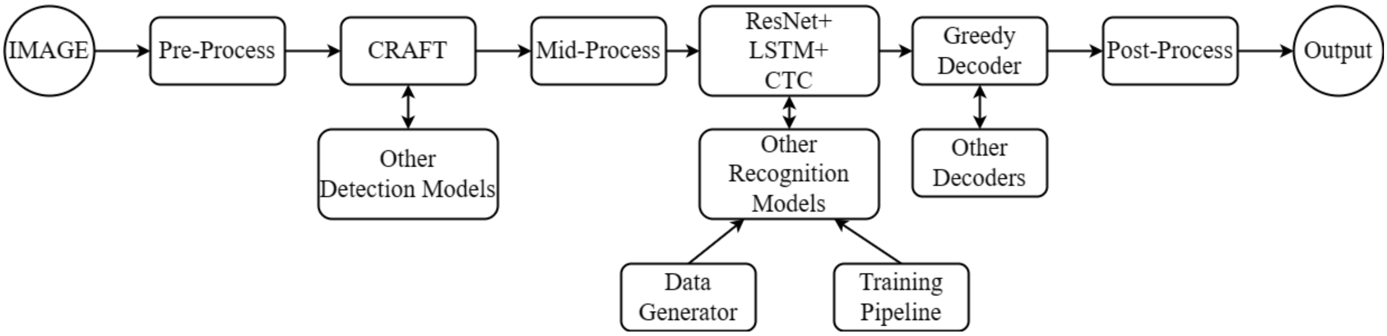


Figure 3.15: EasyOCR Framework

Steps of EasyOCR algorithm

i. Receives preprocessed license plate image as input
ii. Apply noise reduction techniques.
iii. Adjust contrast for improved text visibility.
iv. Utilize the Character Region Awareness For Text Detection (CRAFT) model to detect text regions in the preprocessed images.
v. Generate bounding boxes around the identified text regions.
vi. Extract visual features from the text regions using a CNN, typically based on ResNet.
vii. Apply a Long Short-Term Memory (LSTM) network to analyze the sequential relationship between the extracted features.
viii. Use the Connectionist Temporal Classification (CTC) algorithm to translate the sequential features into a raw text output.
ix. Apply a greedy decoding algorithm to determine the most probable text sequence from the CTC output.
x. Select the most likely character at each step of the decoding process.
xi. Refine the recognized text by correcting spelling errors and apply basic formatting adjustments.
xii. Finally recognized text is generated as the output of the EasyOCR algorithm.

Chapter 5: Implementation and Testing

5.1. Implementation

Python was utilized as the primary programming language for model training and object detection, employing the YOLO v8 model and EasyOCR for character recognition. The front end of this system is constructed with a React-based project, which adopts a component-based approach to building extensive User Interfaces. Each component functions as an independent entity, amalgamating to form meaningful User Interfaces. Shadcn UI was applied for visually styling most of the components, while framer motion was employed for animations. The UI design of the website was crafted using Figma. On the backend, the focus was on generating normalized database tables, implemented through Firebase. Firebase served the dual purpose of handling authentication and storing data in database tables for user information, vehicle information and license plate information. Nextjs were utilized for server-side logic. Finally, Khalti and ESewa were chosen as the payment gateway options.

5.1.1. Tools

Web Application Development Tool

Nextjs (React), Shadcn UI for Styling

Back-End Tools

Python (for model training and detection using YOLO and EasyOCR), Firebase for User information, vehicle information and license plate information for Authentication, Database and Storage, Khalti for Payment Gateway

Other Tools

Vercel for Deployment, Git for Version Control System & GitHub for Collaboration, VS Code for IDE

5.1.2. Implementation Details of Modules

Video Processing Module

It takes input in the form of a JSON object containing a video URL. The function uses the OpenCV library to read frames from the specified video. It then employs a pre-trained COCO model for object detection, focusing on vehicles with specific class IDs (2, 3, and 5) and a confidence score threshold of 0.5. For each detected vehicle, the function extracts the bounding box and subsequently extracts license plates using another custom neural network (np_model). The license plate images are processed to obtain the text through the read_license_plate function. The results, including frame number, track ID, vehicle bounding box, license plate bounding box, license plate number, and corresponding scores, are stored in results_df. The function then computes the total license text score for each unique license plate, identifies the maximum scoring license plate for each track, and calculates the average time per track. Finally, the function returns a JSON response containing the license plate number, text score, track ID, and average time for the highest-scoring license plate in each track, or an error message if any exception occurs during the process.

Bill Generation Module

The entry time is saved in the database upon entry of vehicle in the parking. Then when the vehicle exits the parking lot, the time difference between the exit and entry time is calculated in minutes and then the bill amount is calculated based on the time difference.

The vehicle details are also cross referenced in the database for registered vehicles, and if it matches with one of the registered vehicles, it deducts the balance from the vehicle owner digital wallet, else the status is changed to payment required and can be exited upon processing payment manually.

Vehicle Registration Module

The vehicles in the user dashboard sections shows the list of vehicles for the currently logged in user. When the users add a new vehicle, filling up the vehicle form with bluebook image the information is stored in the database with status as pending. Upon verification of vehicle information with bluebook the status can be changed to Verified in the Admin Dashboard.

The vehicle status verified signifies the registration of the vehicle on the platform.

Payment Module

The module begins by generating unique order and payment IDs, which are then used to create corresponding documents in the Firestore database. The order document includes information such as order amount, user ID, and status, while the payment document contains details like payment amount, order ID, and payment status. Subsequently, a request is made to an external API ("/api/payment") to initiate the payment process, providing necessary data such as order and payment IDs, payment amount, and user information. The response from the API is then examined based on the payment provider (Khalti, Stripe, or ESewa), and the appropriate actions are taken. For example, if the payment provider is Khalti, the user is redirected to the provided payment URL. Similarly, for ESewa, a form is dynamically created and submitted to ESewa's payment endpoint with the required parameters. The module effectively manages the flow of order and payment information between the Firestore database and external payment providers, ensuring a seamless and secure payment process for users.

5.2. Testing

5.2.1. Unit Testing

In unit testing we have tested each module is tested and is analyzed to ensure that all the information flows in and out of the program. Individual components are tested to ensure that operate correctly. Each component is tested independently without other system components. The test cases for unit testing are listed below:

i. User Login

Table 5.1: Test case for User Login

Test Case ID	Test Case	Input data	Expected Output	Actual Output	Status
1	Login with incorrect email and password	Email, password	User Login Successfully	Incorrect Credentials	Fail
2	Login with correct email and password	Email, password	User Login Successfully	As expected,	Pass

ii. Video Processing

Table 5.2: Test case for Video Processing

Test Case ID	**Test Case**	**Input data**	**Expected Output**	**Actual Output**	**Status**
1	Processing low quality video	Upload low quality video	License plate detected	License plate not detected	Fail
2	Processing High quality video	Upload high quality video	License plate detected	As expected,	Pass

iii. Entry and Exit of Vehicles

Table 5.3: Test case for Entry and Exit of Vehicles

Test Case ID	**Test Case**	**Input data**	**Expected Output**	**Actual Output**	**Status**
1.	Vehicle entering parking space	Upload a high-quality video, Time, Date	Vehicle successfully entered.	Video processing failed, unable to grant entry access to vehicle	Fail
2.	Vehicle entering parking space	Upload a high-quality video, Time, Date	Vehicle successfully entered.	As expected,	Pass

3.	Vehicle exiting parking space	Exit time, video	Vehicle successfully exited.	Bill not paid , insufficient balance	Fail
4.	Vehicle exiting parking space	Exit time, video	Vehicle successfully exited.	As expected ,	Pass

iv. Bill Generation

Table 5.4: Test case for Bill generation

Test Case ID	**Test Case**	**Input data**	**Expected Output**	**Actual Output**	**Status**
1.	Generate bill for a vehicle with paid balance	Vehicle ID, Entry Time, Exit Time	Bill generated successfully with correct amount	Bill generated with correct amount	Pass
2.	Generate bill for a vehicle with insufficient balance	Vehicle ID, Entry Time, Exit Time	Bill generated with notification of insufficient balance	Bill generated with notification of insufficient balance	Pass
3.	Generate bill for a vehicle with unpaid balance	Vehicle ID, Entry Time, Exit Time	Bill generated with notification of unpaid balance	Bill generated with notification of unpaid balance	Pass

4.	Generate bill for a vehicle with missing entry or exit data	Vehicle ID, Entry Time or Exit Time (missing), Exit or Entry Time (available)	Bill generation failed due to incomplete data	Bill generation failed due to incomplete data	Fail
5.	Generate bill for a vehicle with incorrect entry or exit time	Vehicle ID, Incorrect Entry Time or Exit Time, Correct Exit or Entry Time	Bill generation failed due to incorrect time	Bill generation failed due to incorrect time	Fail
6.	Generate bill for a vehicle with invalid vehicle ID	Invalid Vehicle ID, Entry Time, Exit Time	Bill generation failed due to invalid vehicle ID	Bill generation failed due to invalid vehicle ID	Fail

5.2.2. System Testing

System testing for a parking solution using Automatic Number Plate Recognition (ANPR) system involves validating the system as a whole to ensure it meets the specified requirements and functions correctly in its intended environment. The test cases for unit testing are listed below:

i. Admin Functionality

Table 5.5: Test Case For Admin Functionality

Test Case ID	**Test Case**	**Input data**	**Expected Output**	**Actual Output**	**Status**
1.	Upload Video	Upload a high-quality video	Video uploaded successfully	Video format not accepted	Fail

2.	Upload Video	Upload a high-quality video	Video uploaded successfully	As expected,	Pass
3.	Enter new vehicle	State, Vehicle type, age identifier, vehicle number, entry time	Vehicle successfully entered.	Invalid vehicle number	Fail
4.	Enter new vehicle	State, Vehicle type, age identifier, vehicle number, entry time	Vehicle successfully entered.	As expected ,	Pass
5.	Verified registered vehicles	Vehicle number, user id	Status changed to approve	As expected,	Pass
6.	Verified registered vehicles	Vehicle number, user id	Status changed to pending	As expected,	Pass
7.	Verified registered vehicles	Vehicle number, user id	Status changed to rejected	As expected,	Pass

8.	Revoke admin privileges	Member name	Revoke admin privileges	As expected ,	Pass

ii. User Functionality

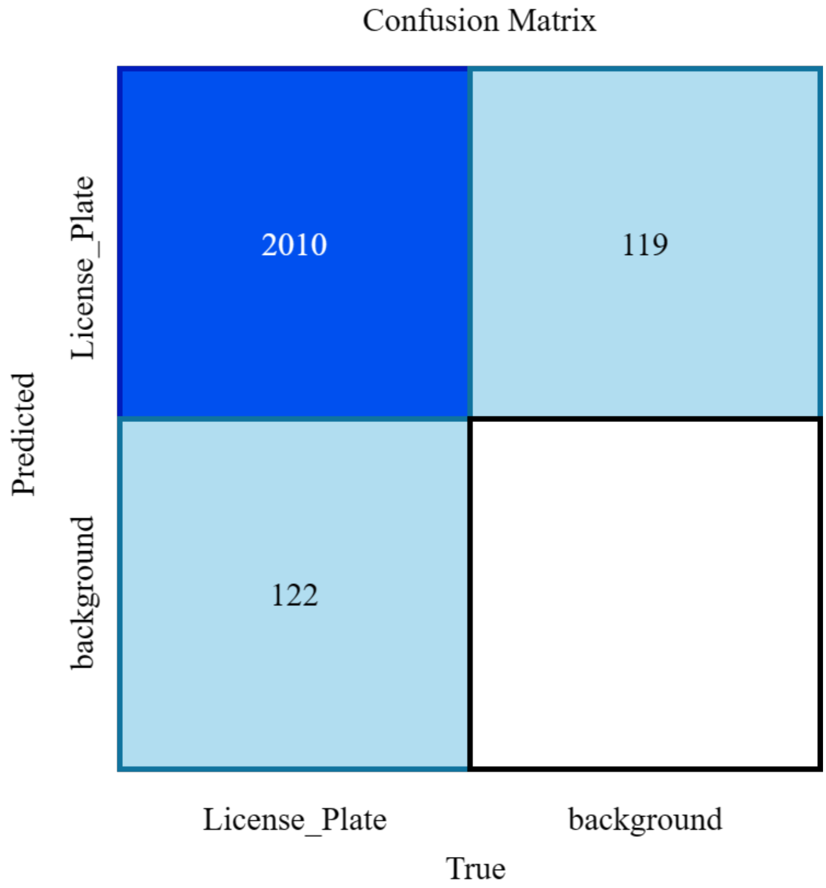
Table 5.6: Test case for User Functionality

Test Case ID	**Test Case**	**Input data**	**Expected Output**	**Actual Output**	**Status**
	Update profile	E-mail, Name, License number, License-card image	Profile updated successfully	Invalid email, License number, License-card image	Fail
	Update profile	E-mail, Name, License number, License-card image	Profile updated successfully	As expected,	Pass
	Add new vehicle	State, Vehicle type, age identifier, vehicle number, entry time	Vehicle successfully entered.	Invalid Bluebook Image, license number	Fail

4.	Add new vehicle	State, Vehicle type, age identifier, vehicle number, entry time	Vehicle successfully entered.	As expected ,	Pass
5.	Load Wallet	Amount, Phone number, PIN	Successful in loading wallet, Wallet balance changed	As expected,	Pass
6.	Load Wallet	Amount, Phone number, PIN	Successful in loading wallet, Wallet balance changed	Insufficient funds in Khalti or esewa or stripe	Fail
7.	Pay Bill	Parking ID, Vehicle ID, Amount	Bill paid ;vehicle allowed to exit	Insufficient fund in wallet	Fail
8.	Pay Bill	Parking ID, Vehicle ID, Amount	Bill paid; vehicle allowed to exit	As expected,	Pass

5.3. Result Analysis

5.3.1. Confusion Matrix of License Plate Detection Model



<u>Figure 5.1: Confusion Matrix of License Plate Detection Model</u>

In the evaluation of the model's performance on license plate detection, it was observed that, out of a total of 2400 images, the model correctly identified the presence of a license plate in 2010 instances and incorrectly predicted it in 119+122 instances, resulting in a total of 231 false positives. Notably, there were no true negatives recorded, indicating that the model did not successfully identify any images without a license plate. This deficiency poses a substantial limitation, as the model struggles to distinguish between images with and without license plates.

One significant issue is the elevated number of false positives, implying that the model frequently misidentifies objects as license plates when they are not. Additionally, the true negative rate is 0, signifying a complete inability to accurately identify images devoid of license plates. This presents a serious challenge, rendering the model unreliable for discerning images that do not contain license plates.

In summary, the confusion matrix highlights the model's poor performance, characterized by a high incidence of false positives and an absence of true negatives. Addressing these issues is imperative before considering the deployment of the model in real-world applications.

5.3.2. F1 Curve

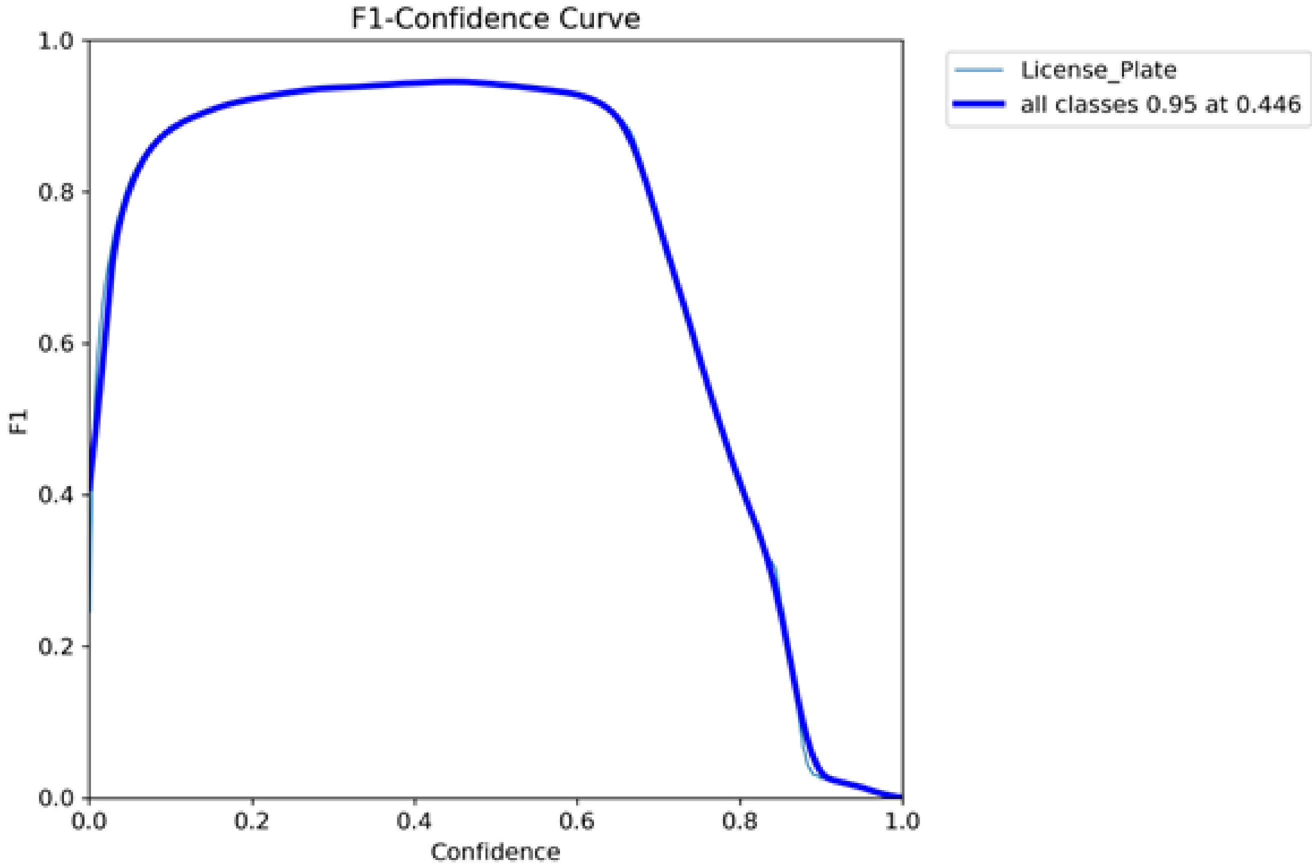


Figure 5.2: F1-Curve of License Plate Detection Model

The curve in the image shows that the F1 score of the neural network increases as the confidence threshold increases. This means that the model is more accurate at detecting license plates when it is only considering predictions with high confidence scores. However, as the confidence threshold increases, the number of predictions that the model makes also decreases. This is because the model is only considering a smaller subset of the data.

The text in the image shows that the F1 score of the neural network is 0.95 while the confidence threshold is 0.446. This means that the model is **95%** accurate at detecting license plates when it is only considering predictions with a confidence score greater than 0.446.

Chapter 6: Conclusion

In conclusion, the implementation of an Automatic Number Plate Recognition (ANPR) system in our parking solution has successfully achieved its objectives. Through the utilization of the YOLOv8 detection model and EasyOCR character recognition, we have effectively automated the process of license plate recognition from uploaded videos, enhancing efficiency and accuracy in vehicle identification. Additionally, the integration of time-based pricing for registered vehicles during check-in and checkout processes has been seamlessly introduced, providing a fair and dynamic pricing model for our users. By fulfilling these objectives, we have not only improved the overall functionality and effectiveness of our parking solution but also enhanced the user experience by streamlining the entry and exit processes while ensuring equitable pricing practices. This means that the model is **95%** accurate at detecting license plates when it is only considering predictions with a confidence score greater than 0.446. Moving forward, we remain committed to further advancements and innovations that will continue to optimize the parking experience for both operators and users alike.

6.1. Future Recommendation

Moving forward, several recommendations can be considered to enhance the parking solution using Automatic Number Plate Recognition (ANPR) system:

i. Enhanced Integration: Expand integration capabilities to include various parking management systems, payment gateways, and mobile applications to offer a seamless experience for users.
ii. Scalability: Ensure that the ANPR system is designed to scale effectively to accommodate increasing numbers of users and vehicles without sacrificing performance or accuracy.
iii. Real-time Monitoring: Implement real-time monitoring and alerts to promptly address issues such as unauthorized vehicle access, parking violations, or system malfunctions.
iv. Security Measures: Implement robust security measures to protect user data, prevent unauthorized access to the ANPR system, and ensure compliance with data protection regulations.

REFERENCES

[1]. K. Oo and A. A. Bakar, "A Deep Learning-Based License Plate Recognition System Using Turkish Dataset and Keras," 2019 International Conference on Advanced Science and Engineering (ICASE), 2019.

[2]. R. Kaur and R. Verma, "Automatic Number Plate Recognition using Machine Learning Approach," 2018 3rd International Conference on Computing and Communication Systems (ICCCS), 2018

[3]. P. S. Hiremath, D. S. Hunagundar, and V. C. Patil, "License Plate Recognition System Using Convolutional Neural Network," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016.

[4]. J. Li, Y. Wang, and J. Zhang, "An End-to-End Framework for Robust License Plate Recognition with Deformable Convolution and Attention Mechanism," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2023.

[5]. J. Wang, Y. Chen, and Z. Liu, "Illumination-Invariant License Plate Recognition Using Channel-Wise Attention and Dual Loss," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2022.

[6]. I. Kilic and G. Aydin, "Turkish Vehicle License Plate Recognition Using Deep Learning," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018.

[7]. J. V. Bagade, Sukanya Kamble, Kushal Pardeshi, Bhushan Punjabi, Rajpratap Singh, "Automatic Number Plate Recognition System: Machine Learning Approach," IOSR Journal of Computer Engineering (IOSR-JCE).

[8]. A. Patel, C. Patel, and D. Shah, "Automatic Number Plate Recognition System (ANPR): A Survey," International Journal of Computer Applications, vol. 69, no. 9, pp. 0975–8887, 2013.

[9]. Shraddha S Ghadage, Sagar R Khedkar, "A Review Paper on Automatic Number Plate Recognition System using Machine Learning Algorithms," International Journal of Engineering Research & Technology (IJERT), vol. 8, issue 12.

[10]. Marko Arsenovic, Srđan Sladojevic, Andras Anderla, Darko Stefanovic, "Deep Learning Driven Plates Recognition System," XVII International Scientific Conference on Industrial Systems, Novi Sad, Serbia, 2017.

[11]. Shrutika Saunshi, Vishal Sahani, Juhi Patil, Abhishek Yadav, Dr. Sheetal Rathi, "License Plate Recognition Using Convolutional Neural Network," IOSR Journal of Computer Engineering (IOSR JCE), e-ISSN: 2278-0661, p-ISSN: 2278-8727.

[12]. Selmi Zied, Ben Halima Mohamed, Alimi Adel, "Deep Learning System for Automatic License Plate Detection and Recognition," 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 1132-1138, 2017.