



TRIBHUVAN UNIVERSITY

Lumbini City College

Divertole, Tilottama, Rupandehi

An Internship Report

on

“SnapDrive: An Orientation Detection Mobile App”

At

(KonnectCraft)

Submitted By

Kishmat Bhattarai (20530/075)

*An Internship Report Submitted in partial fulfillment of the requirement of **Bachelor of Science in Computer Science & Information Technology (BSc.CSIT) 8th Semester of Tribhuvan University, Nepal***

September, 2023

Acknowledgement

The successful completion of this internship report would not have been possible without the support and assistance of many individuals and organizations. I feel immensely blessed to have gotten this during the course of my internship program. I would like to take this opportunity to offer my earnest admiration to each and every one of them.

I express my sentiment of gratitude to **Mr. Anish Byanjankar**, who has been a continuous source of inspiration as my intern mentor. Without his constant guidance and suggestions, this report would have been nowhere near completion. My gratitude for his trust and generosity goes beyond words. I am indebted and thankful to my learned and revered supervisor for his upbeat personality, kindness; encouraging support and willingness to help have tangibly and greatly improved the quality of my Internship Report and brought it up to its present status.

Thanks to our colleagues who helped us directly or indirectly to accomplish our work. I am especially grateful to **Mr. Kamal Bahadur(Shrish) Thapa**(Supervisor) and **Mr. Manoj Pokharel**(Coordinator), who has also been a positive and encouraging influence on our research efforts. Finally, I thank all our teachers and colleagues, who were the people who prepared us for this endeavor.

With respect,

Kishmat Bhattarai (20530/075)

Abstract

The purpose of the project, SnapDrive is to deliver a comprehensive web and mobile application that assists vehicle management process. With its features and integration of Machine Learning technology, SnapDrive offers a robust solution for efficient organization and management of vehicle-related information. The application empowers users to capture their vehicle's image with precise orientation, following a specific order, using orientation detection model. The orientation detection model is created using the Transfer Learning techniques where the pretrained model is used and trained with multiple vehicle's orientation images. SnapDrive is synchronized between the web and mobile platforms, it ensures a seamless user experience and streamlined data flow. SnapDrive optimizes the vehicle management process, making it easier than ever to keep track of vehicle details and capture accurate visual representations.

KEYWORDS: *Vehicle data, Image Orientation, Machine Learning, Inventory*

Table of Contents

Acknowledgement	i
Abstract	ii
List of Abbreviations	v
List of Figures	vi
List of Tables	vii
Chapter 1: Introduction	1
1.1. Introduction	1
1.2. Problem Statement	1
1.3. Objectives.....	2
1.4. Scope and Limitation	2
1.4.1. Scope.....	2
1.4.2. Limitation.....	2
1.5. Report Organization	2
Chapter 2: Organization Details and Literature Review	4
2.1. Introduction to Organization	4
2.2. Organization Hierarchy	4
2.3. Working Domains of Organization	5
2.4. Description of Intern Department/Unit	5
2.4.1. Placement	5
2.4.2. Duration	5
2.5. Literature Review	6
Chapter 3: Internship Activities	8
3.1. Roles and Responsibilities	8
3.2. Weekly Log	8
3.3. Description of the Project Involved During Internship	10
3.3.1. System Requirements.....	10
3.3.2. System Design	12
3.4. Tasks Performed.....	22
3.4.1. System Implementation	22
3.4.2. System Testing.....	25

Chapter 4: Conclusion and Learning Outcomes	31
4.1. Conclusion.....	31
4.2. Learning Outcomes	31
References.....	32
Appendices.....	33

List of Abbreviations

AI	-	Artificial Intelligence
API	-	Application Programming Interface
HTTP	-	Hypertext Transfer Protocol
JS	-	JavaScript
REST	-	Representational State Transfer
SQL	-	Structured Query Language
SSD	-	Single Shot Multi-Box Detector
UI	-	User Interface
UML	-	Unified Modeling Language
YOLO	-	You Only Look Once

List of Figures

Figure 2.1: Organization Hierarchy	4
Figure 3.1: Use Case Diagram (Mobile App).....	12
Figure 3.2: System Architecture of SnapDrive.....	13
Figure 3.3: Class Diagram	14
Figure 3.4: State Diagram (Web App).....	15
Figure 3.5: State Diagram (Mobile App).....	16
Figure 3.6: Sequence Diagram (Web App).....	17
Figure 3.7 Sequence Diagram (Mobile App).....	18
Figure 3.8: Activity Diagram (Web App).....	19
Figure 3.9: Activity Diagram (Mobile App).....	20
Figure 3.10: Component Diagram	21
Figure 3.11: Deployment Diagram	22
Figure 3.12: Incremental Delivery Model	24
Figure 3.13: Confusion Matrix for Orientation Detection Model.....	27
Figure 3.14: Accuracy per Epochs Graph for Orientation Detection Model.....	28

List of Tables

Table 2.1: Intern Duration in KonnectCraft.....	6
Table 3.1: Weekly Log	8
Table 3.2: Test Case for Installation	25
Table 3.3: Test Case for Application Login.....	25
Table 3.4: Test Case for Orientation Detection Model.....	26
Table 3.5: Test Case for Data Input	28
Table 3.6: Test Case for Capturing Image	28
Table 3.7: System Testing.....	29

Chapter 1: Introduction

1.1. Introduction

This report provides an overview of the tasks performed by the intern and highlights the various related tasks completed during the internship period. The internship took place at KonnectCraft Inc, where the intern had the privilege to work on the project called "SnapDrive". Specifically, the intern was involved in teams responsible for building web and mobile applications using frameworks like ReactJS, Flutter, and Django.

SnapDrive is a web and mobile application designed for vehicle management process. SnapDrive offers a robust solution to efficiently manage and organize vehicle related information as well as allows users to capture their car image with proper orientation in particular order with the help of Machine Learning. The app detects the presence of a vehicle and prompts users to capture ten different types of vehicle orientations. SnapDrive ensures seamless integration and synchronization between the web and mobile applications.

SnapDrive uses state-of-the-art Machine Learning practices like transfer learning. Transfer learning is used for building the model to detect the orientation of the vehicle based on the image stream provided by the camera of mobile devices. Similarly, for vehicle detection another model is used, called SSD MobileNet.

1.2. Problem Statement

Manual record-keeping and scattered information make it challenging to effectively manage and organize vehicle related data. Additionally, there is a lack of streamlined processes for capturing and organizing vehicle images with proper orientation. Current method for such use case can be inconsistent and error prone, which leads to inaccurate assessments. Therefore, there is a need for a comprehensive and user-friendly vehicle management solution that combines efficient record keeping and organizing vehicle images.

By addressing these challenges, SnapDrive aims to optimize the management of vehicle inventories, improve decision making processes and enhance overall operational efficiency for individuals in the automotive industry.

1.3. Objectives

- To build a Deep Learning model for orientation detection.
- To integrate Deep Learning model with mobile application to ensure image accuracy.

1.4. Scope and Limitation

1.4.1. Scope

This report describes the intern's activities and contributions throughout his internship at KonnectCraft Inc. The primary focus of his work during this period was the development of a Vehicle Inventory Management System. The report provides an outline of the project's framework, implementation details, and testing methodologies. Additionally, it highlights the intern's acquired knowledge and experiences during his time in the organization.

1.4.2. Limitation

The limitations of this internship report and project includes the following:

- **Confidentiality:** There were some restrictions in disclosing some information as it was assumed to be confidential. Therefore, intern could not include those information's in his report.
- **Limited Features:** The project may have limitations in terms of the features and functionalities provided like lack of interface for comparing features of multiple vehicles and limited set of fields for adding vehicle related data.
- **Accuracy:** The system may misidentify or struggle to accurately detect certain views as well as for object detection since the primary focus of the system is to detect cars so it may misidentify other real-world objects.

1.5. Report Organization

The report on "SnapDrive" is based on four chapters.

Chapter 1 Introduces the internship project along with problem statements, objectives, scope and limitations of the project.

Chapter 2 Introduces the organization, organizational hierarchy along with its working domains, description of intern department/unit and literature review of similar projects.

Chapter 3 Consists of the internship activities like roles and responsibilities, weekly log, description of the projects involved during the internship.

Chapter 4 Includes the conclusion section and also learning outcomes from the internship.

Chapter 2: Organization Details and Literature Review

2.1. Introduction to Organization

KonnectCraft is a Managed IT and VoIP Services Provider that has been providing quality services to small to mid-sized businesses (SMB) across US & Canada since 2012. The company understands that not all businesses are equal and each have different needs and requirements. The company also work with the clients to deliver a tailored solution to meet their goals and challenges. KonnectCraft also engages with clients across diverse industries such as Automotive, Real Estate, Finance, Accounting, Manufacturing, and Legal fields. As an authorized partner of 3CX, team comprises certified engineers with expertise in various products and vendors including Avaya, Microsoft, Cisco, Fortinet, and Netgate pfSense Security Appliances.

Apart from Managed IT and VoIP services, company also provide Hardware Procurement, Web Development and Systems Integrations Services to essentially become a one stop solution for all IT and Business communication needs for the customers. (Byanjankar, Byanjankar, & Shrestha, 2022)

2.2. Organization Hierarchy

KonnectCraft Inc. includes several groups and departments. These includes the design and development group, testing group, accounts, marketing and sales, business development, technology division, and managing director.

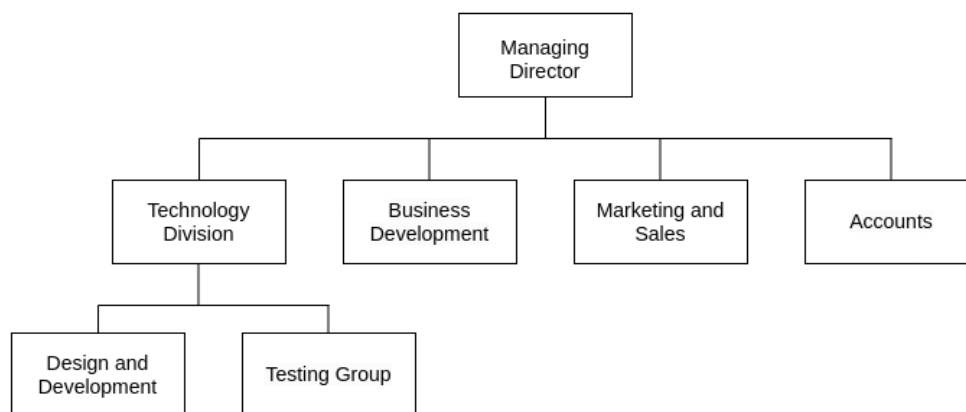


Figure 2.1: Organization Hierarchy

2.3. Working Domains of Organization

Within its diverse working domains, the organization offers a comprehensive range of services to address the IT and business communication needs of SMBs. The overview of the key working domains, highlighting the specialized solutions and services provided by the organization are as follows:

- **Web App Development:** The organization specializes in web app development, creating robust and user-friendly applications that run on web browsers. These applications are customized to meet the requirements of businesses, providing functionality and interactivity.
- **Mobile App Development:** The organization also offers expertise in mobile app development. Utilizing the latest technologies and platforms, they design and develop mobile applications for operating system such as Android.
- **VOIP Phone System and Managed IT Services:** The organization provides VOIP (Voice over Internet Protocol) phone systems and comprehensive managed IT services. The VOIP phone systems leverage internet connectivity to enable efficient and cost-effective communication for businesses. Additionally, managed IT services, ensures the smooth operation, security, and maintenance of IT infrastructure, networks, and systems for the clients.

2.4. Description of Intern Department/Unit

2.4.1. Placement

KonnectCraft Inc. gives equivalent open doors to new commoners in the tech market and experiences one as well. Interns are recruited through CV screening to determine if candidate has necessary qualifications and then technical interview is taken where the lead takes an interview and asks questions related to software engineering. After that interview performance of the candidates are evaluated and is followed by HR round where certain questions are asked to the participants. Participants can also ask if they have any queries regarding company. Certain candidates are selected for the position.

2.4.2. Duration

The duration and relevant details of internship are as follows:

Table 2.1: Intern Duration in KonnectCraft

Internship Start Date:	15 th May, 2023
Internship End Date:	15 th Aug, 2023
Duration	3 Months
Office Hour:	9:00 AM to 6:00 PM
Working Days:	Sunday to Thursday

2.5. Literature Review

Object detection and Image classification are two fundamental tasks in the field of computer vision that play a crucial role in various applications. Several object detection and image classification apps have been developed each with its unique approach and strengths. For this purpose there are different model are available and few of the models which are widely used and known are YOLO, SSD, MobileNet etc. The researchers and practioners have used these model for multiple purposes like for building a module or doing some research activity.

Among the multiple models built using deep learning technique, MobileNet model has been found to have wide applications in various vision tasks due to its efficiency and effectiveness. This model has streamlined architecture that utilizes depth-wise convolutions. With extensive experimentation, MobileNet has resulted impressive performance in comparison to other popular models. By leveraging MobileNet model and using transfer learning method it can be used for development of various vision-based systems and applications. (Andrew G. Howard, 2017)

MobileNet model is used by multiple researchers and practioners. This model became very successful due to its ability to capture the feature of the image properly which can be very beneficial for developing the model which it was intended for. Apart for MobileNet itself at a model, this model is very popular in the context where it can be used as base model and create a completely different model with only few numbers of images provided with the help of Transfer Learning method. With transfer learning method, basically first the pretrained convolutional layer is taken from the pre-trained model which basically captures the property of the image like edges and use this feature extracting layer to train for the multiple use cases.

Researchers in the past have used object detection models and image classification models for various real time computer vision tasks. “Real Time Object Detection and Recognition using MobileNet-SSD with OpenCV” was one where the researchers used MobileNet-SSD Model to detect objects on video stream via webcam. They used models for their use case of real-time detection and for webcam streaming to detect objects in a video stream. (Mr. Harshal Honmote, 2022)

Similarly, research done on fruit image classification, where the deep learning model identifies the pattern of the fruit images from a large number of image dataset and classifies the newer fruit image. This task of detecting the fruit required expert and detailed inspection which was easily automated with the help of a deep learning model. Researchers used MobileNetV2 as the base model and trained on images using Transfer Learning Technique. (Gulzar, 2023)

In the same way another research on bird image classification was done by the researchers in past, where they used deep learning models to identify the bird species. This research was done with intention to identify the endangered birds and helps as the initial steps on preserving them. Researchers used different models and did transfer learning to see the outcomes of the prediction with base models as Resnet152V2, Inception V3, Densenet201, and MobileNetV2. (Manna, et al., 2023).

Chapter 3: Internship Activities

3.1. Roles and Responsibilities

At first the supervisor divided the work into teams and assigned the task to the teams. Then for few weeks basic knowledge of React, Django, Docker, was learned to get started. In first week, concepts like react Components, Props, Hooks such as useState, useEffect, Routes were learned. In second, third, fourth, fifth week, concepts like docker container, docker image, docker compose, different commands related to docker, Django, useContext hook for state management, fetch and axios for API call, antDesign as rich UI components were learned and used to create vehicle inventory management web application. During the development period the team members pushed their work into git repository and also the application was dockerized.

After finishing the web application, another project was assigned to the teams. For few weeks basic knowledge of Flutter and Dart was explored. In sixth week, concepts like Widgets, Stateless Widgets, Stateful Widgets, Navigation were learned. In seventh and eight-week, UI of the app was developed followed by integration of API call to the backend using package like http. Further shared_preferences, intl were also used for local storage and date formatting. In rest of the week, AI model was created using keras and the model was integrated in flutter app using different packages like Camera, tflite, Image. Then the application was hosted in the Virtual Machine in cloud.

3.2. Weekly Log

Table 3.1: Weekly Log

Weeks	Activity Topic	Activity Log
First	Basic Overview of React, Django	Research and implementation of concepts like react Components, Props, Hooks such as useState, useEffect, Routes and Django REST Framework, SQLite database.

Second	Basic Overview of Docker	Research and implementation of concepts like docker container, docker image, docker compose, dockerfile.
Third	Frontend (React)	Sign in, Signup, Homepage, Add-vehicle page, Edit-vehicle page, Vehicle-details page using antdesign.
Fourth	Authentication	API Testing with POSTMAN. Local Storage to store user info, useContext for State Management.
Fifth	Backend Integration (React)	Fetch, axios for API request for sign in, sign up, add vehicle data, edit vehicle data, delete vehicle data.
Sixth	Basic Overview of Flutter and Dart	Research of Widgets, Stateless widgets, Stateful widgets, Navigation.
Seventh	Frontend (Flutter)	Login screen, Register screen, Home screen, Add vehicle screen.
Eight	Backend Integration (Flutter)	API request using http package, shared_preferences, Intl to store user info and format date.

Ninth	AI Model (Keras)	Created Datasets, used transfer learning to train model, model evaluation.
Tenth	Model Integration (Flutter)	Camera screen, Orientation screen. Using Camera package for camera screen, tflite package to run model.
Eleventh	Backend Integration (Flutter)	API request to add vehicle data. Prepare image data using image package and base64Encode.
Twelfth	Hosting	System Testing and Deployed web application in Virtual Machine

3.3. Description of the Project Involved During Internship

3.3.1. System Requirements

System requirements refer to the specific functionalities, characteristics, and conditions that a software or system must meet to fulfill client's needs. They guide the design, development, and testing of the system. There are different types of requirements, including functional and non-functional requirements.

3.3.1.1. Functional Requirements

The functional requirement defines the system. It specifies what should the system do. Functional requirements of SnapDrive include the followings:

- Register by the user
- Login by the user
- Adding, editing and deleting vehicle data
- Capturing vehicle image
- Detect vehicles and its different orientations
- Sending captured image and vehicle data to the database

Use Case Diagram

Use case diagram is a graphical depiction of a user's possible interactions with a system. It is used to gather the requirements of a system including internal and external influences.

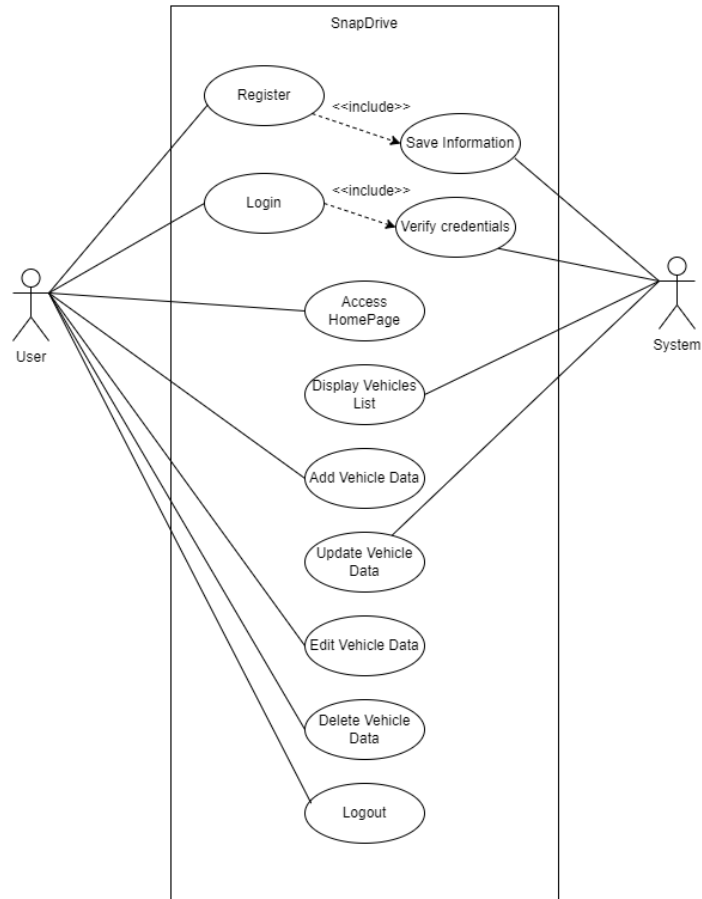


Figure 3.1: Use Case Diagram (Web App)

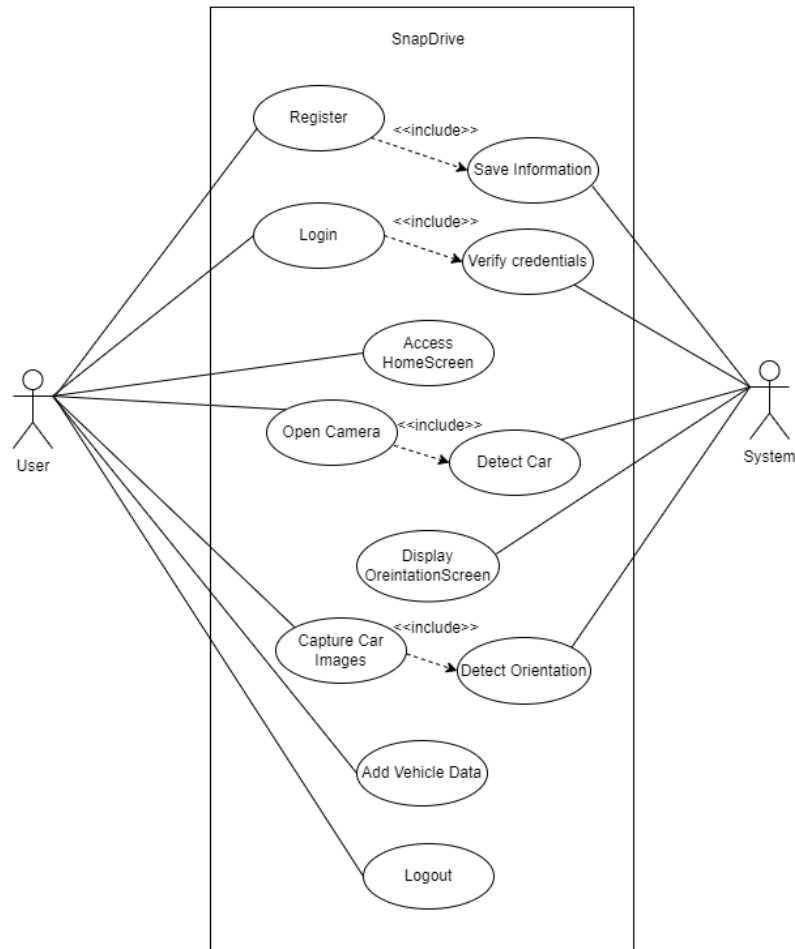


Figure 3.1: Use Case Diagram (Mobile App)

3.3.1.2. Non-Functional Requirements

It specifies how the system performs. The non-functional requirements include the following:

- Performance
- Security
- Availability
- Reliability

3.3.2. System Design

System design is the process of representing architecture, interfaces, components that are included in the system. i.e., system design can be seen as the application of system theory to product development. System design can be understood as a blueprint that outlines the

architectural design of the system. In essence, system design serves as a visual representation of how the system will function and its underlying structure.

3.3.2.1. Architectural Design

The application architecture explains the patterns and techniques used to design and build an application. Among numerous architectures, SnapDrive follows Three-Tier architectural design.

The three-tier architecture divides applications into three distinct computing layers: the presentation layer or client layer which is responsible for the user interface, the application layer which is responsible for data processing and implementing the business logic and finally the database layer which is responsible for storing and managing the application's data. In SnapDrive, the user interacts with the application through the user interface via web browser or mobile application which is connected to the application layer through API Gateway. Application Layer at the middle handles the request from the user, process the request and send the response back to the user. Apart from that application layer also communicates with database layer for storing the processed information which can be retrieved later when required.

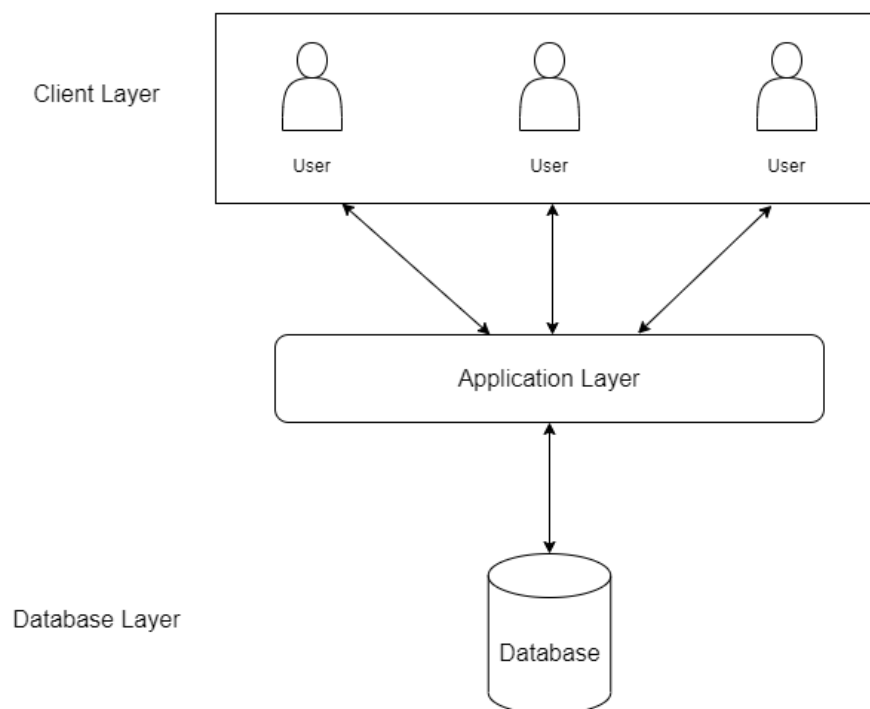


Figure 3.2: System Architecture of SnapDrive

3.3.2.2. Analysis

Systems analysis involves the study of a system to gain an overall understanding of its components, their interactions, and potential areas for enhancement. It takes an entire perspective, considering the system as a unified entity and identifying the interconnections among its elements. The primary objective of systems analysis is to identify existing issues and inefficiencies within the system and propose appropriate solutions for enhancement.

3.3.2.2.1. Object modelling using Class Diagrams

Class diagram in the UML is a type of static structure diagram that describes the structure of a system, it shows system's classes along with their attributes, operations and relationships among objects.

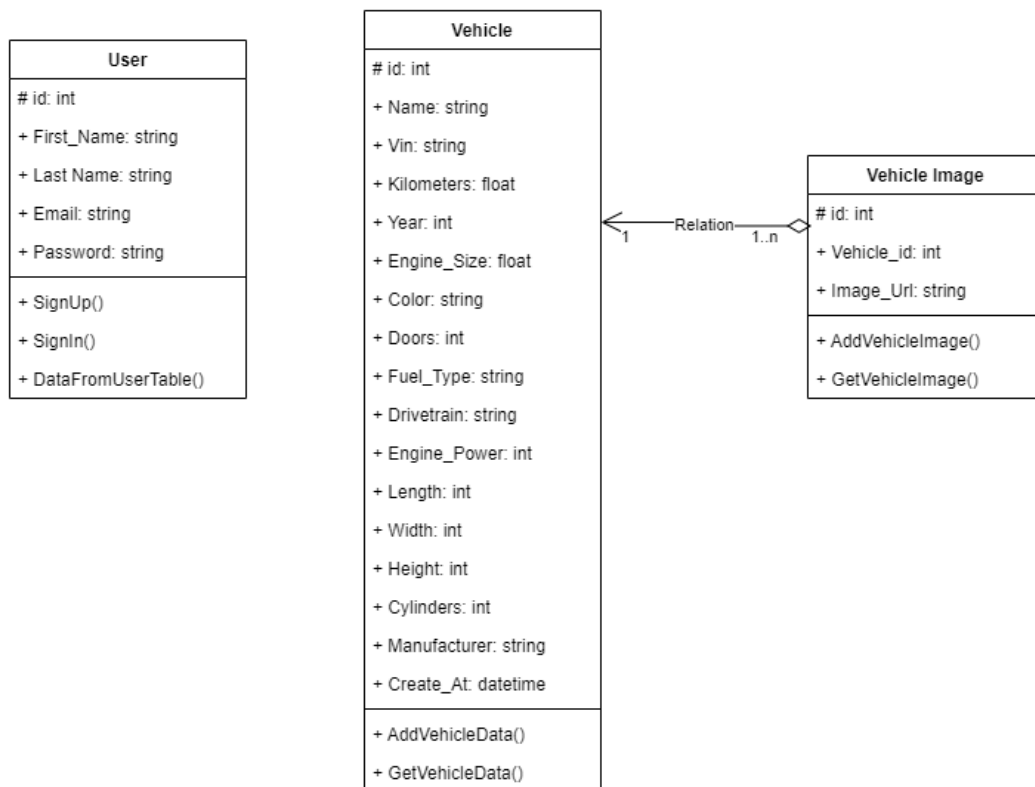


Figure 3.2: Class Diagram

The above class diagram represents the different classes in the system and their relationships. The classes are User, Vehicle, VehicleImage. Each class contains 3 compartments where, the first compartment contains the class name itself, second compartment contains individual properties and the third compartment contains different operations. Here, the Vehicle class has one to many relations as well as aggregation relationship with VehicleImage class.

3.3.2.2.2. Dynamic modelling using State and Sequence Diagrams

State diagram in the UML, also known as a state machine diagram is a type of diagram that illustrates the possible states of the system and transitions between those states.

Sequence diagram in the UML is a type of diagram that illustrates the sequence of messages between objects in an interaction. It consists of a group of objects represented by lifelines and message they exchange during the interaction denoted by arrow symbols.

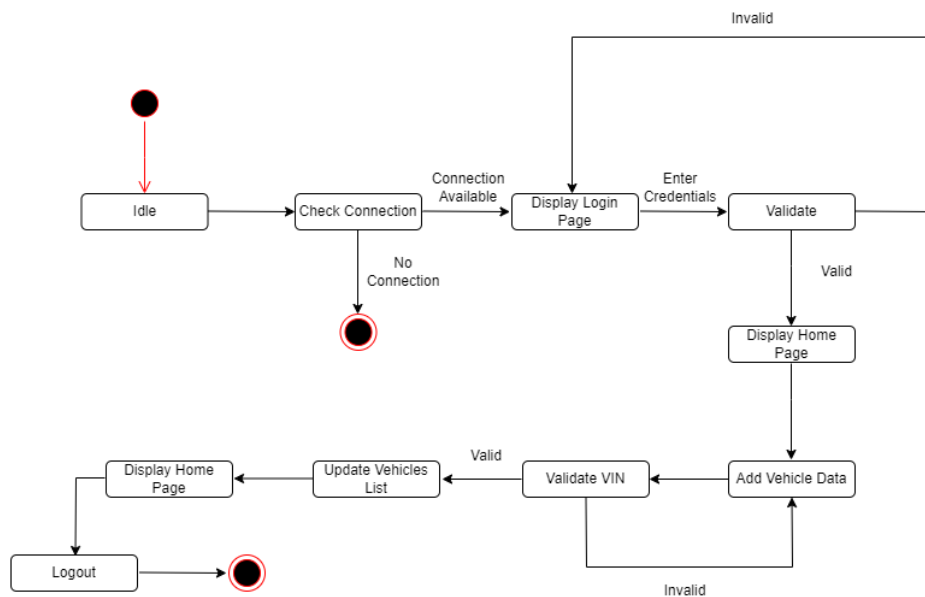


Figure 3.4: State Diagram (Web App)

The above state diagram for the web app consists of different states and events. First the connection is checked. If the connection is available then the system displays a login page and prompts the user to enter their login credentials. If the user enters valid credentials, then the user is prompted to homepage otherwise the user is redirected to the login page. The user can add vehicle data and if the VIN is valid then the vehicles list is updated and the list and details of the vehicle is displayed.

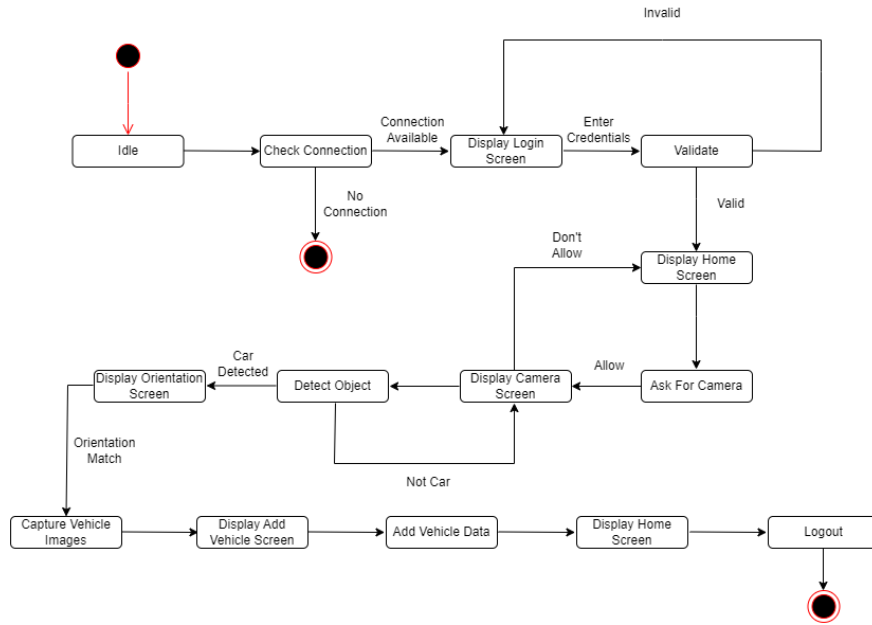


Figure 3.5: State Diagram (Mobile App)

The above state diagram for the mobile app consists of different states and events. First the connection is checked. If the connection is available then the system displays a login screen and prompts the user to enter their login credentials. If the user enters valid credentials, then the user is prompted to home screen otherwise the user is redirected to the login screen. Then, the user is asked to provide the access for camera. If access for camera is provided by the user, then the camera screen is displayed. In camera screen the object is detected and if car detected, the user can route to the orientation screen. Here, images of vehicles are captured in a particular order. After capturing vehicle images, the user is prompted to add vehicle data screen where different data related to vehicles can be added.

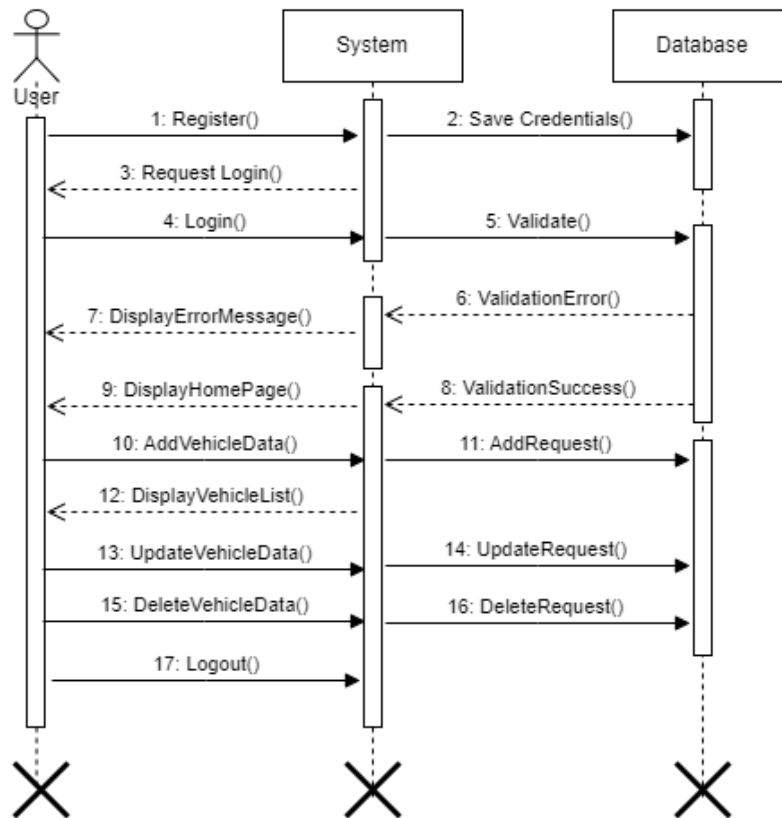


Figure 3.6: Sequence Diagram (Web App)

The above sequence diagram for web app consists of actor i.e., user, and two objects i.e., system and database. The user registers with valid credentials and the system stores it in the database. When the user tries to login, the provided info is validated and if the user is successfully verified then the access is granted otherwise an error message is displayed. Then homepage is displayed to the valid user. User can add vehicle data through add vehicle page, that data is stored in the database and the vehicles list is also updated which can be displayed to the user in the home page. Similarly, the vehicle data can be edited, updated and deleted respectively. During these requests, the database is updated accordingly.

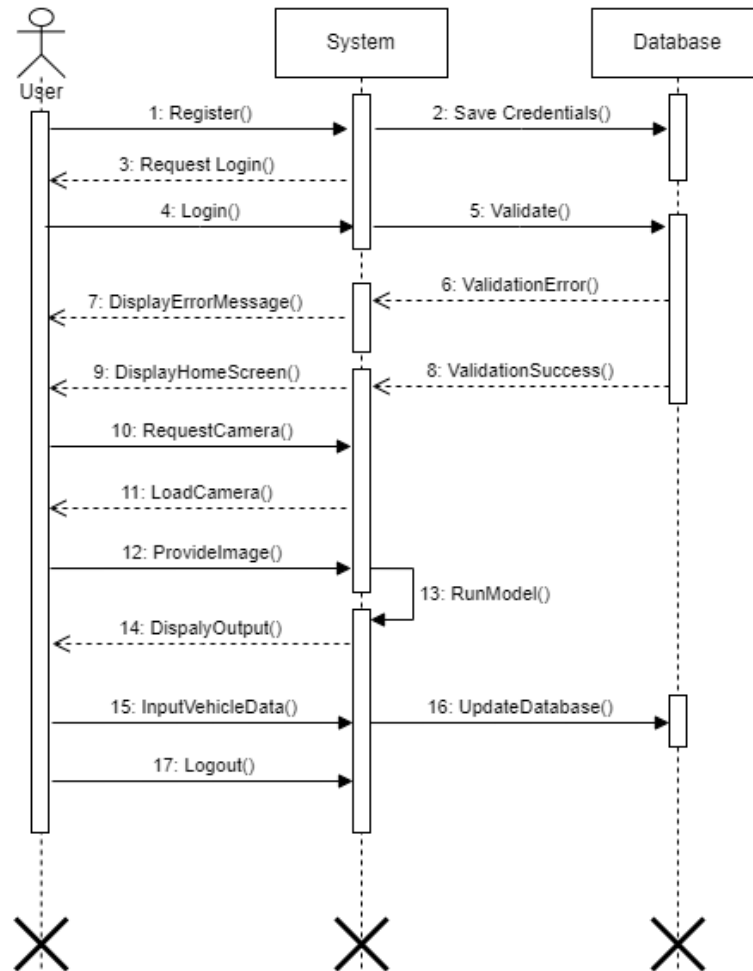


Figure 3.7 Sequence Diagram (Mobile App)

The above sequence diagram for mobile app consists of actor i.e., user, and two objects i.e., system and database. The user registers with valid credentials and the system stores it in the database. When the user tries to login, the provided info is validated and if the user is successfully verified then the access is granted otherwise an error message is displayed. Then home screen is displayed to the valid user. User can request camera through the camera screen and the camera is loaded by the system. Through orientation screen vehicle images are captured and are provided by the user to the system. Model is run by the system continuously to detect the object and different orientations. The output is then displayed to the user based on the image. At last, the vehicle data are added by the user and hence the database is updated.

3.3.2.2.3. Process modelling using Activity Diagram

Activity diagram in the UML is a type of diagram that visually presents a series of actions or flow of control in a system. It shows workflows of stepwise activities with support for choice, iteration and concurrency.

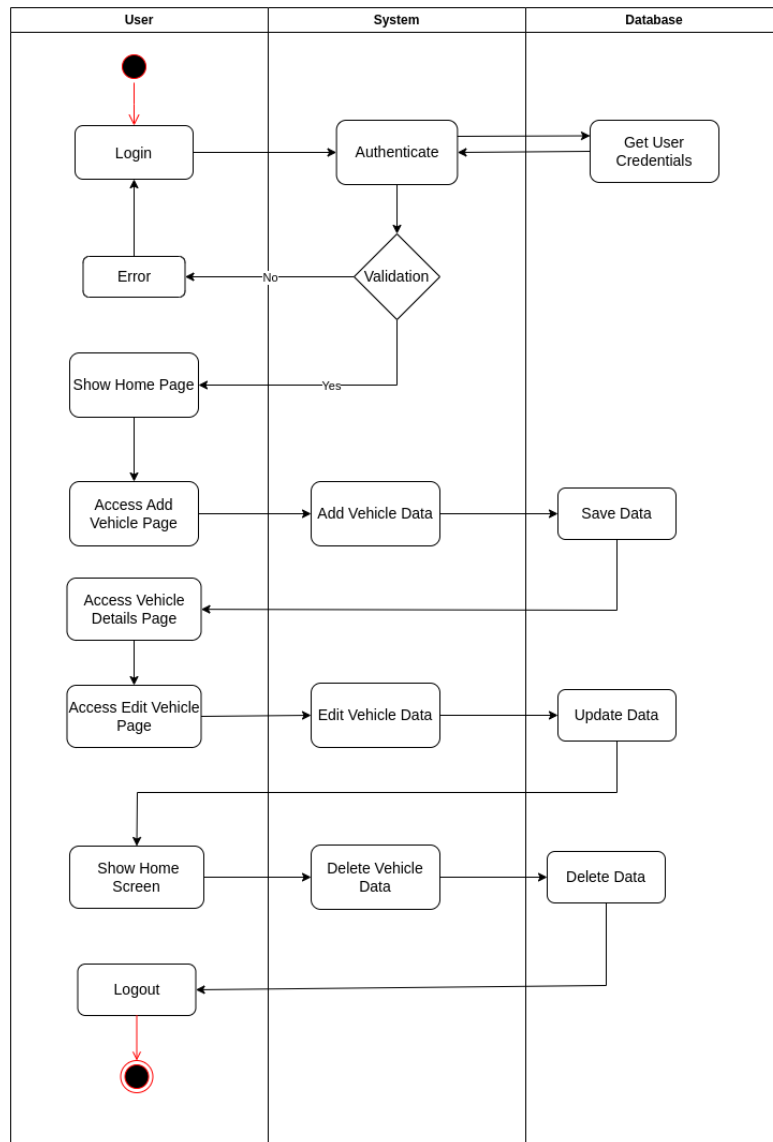


Figure 3.8: Activity Diagram (Web App)

The above activity diagram for web app shows the flow of control from User, System and Database. First the user credentials are passed to the system for authentication. Then if the user is valid the user is taken to home page, and shows error if the credentials provided is not valid. User can access the add vehicle page to add vehicle data which is then sent to the database for saving those data. Similarly, user can edit and delete vehicle data. During this process, the database is updated accordingly.

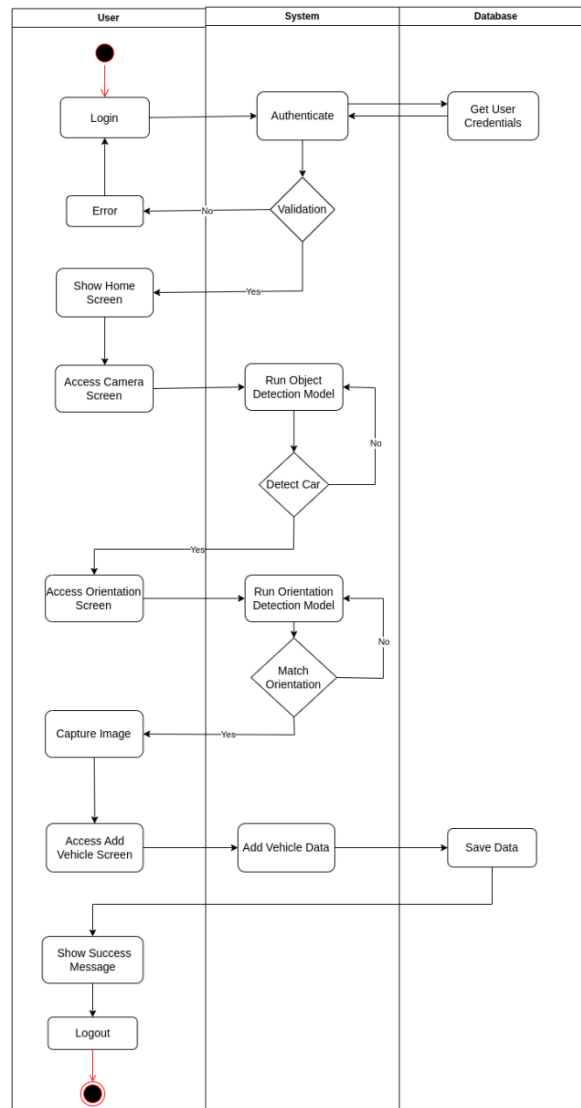


Figure 3.9: Activity Diagram (Mobile App)

The above activity diagram for mobile app shows the flow of control from User, System and Database. First the user credentials are passed to the system for authentication. Then if the user is valid the user is taken to home screen, and shows error if the credentials provided

is not valid. User can access the camera screen, for detecting object the system runs a model and if the detected object is car, then the user can access the orientation screen. In orientation screen, another model is run by the system for detection of different orientations. If the required orientation matches, then the user is allowed to capture image. After capturing all the orientation of vehicle, user is prompted to add vehicle screen where user can enter vehicle data which is saved in the database.

3.3.2.2.4. Component Diagram

Component diagram are UML structural diagrams that is used in modeling the physical aspects of object-oriented systems. It shows the relationship between various components of the system which helps to visualize, specify and document those components-based systems.

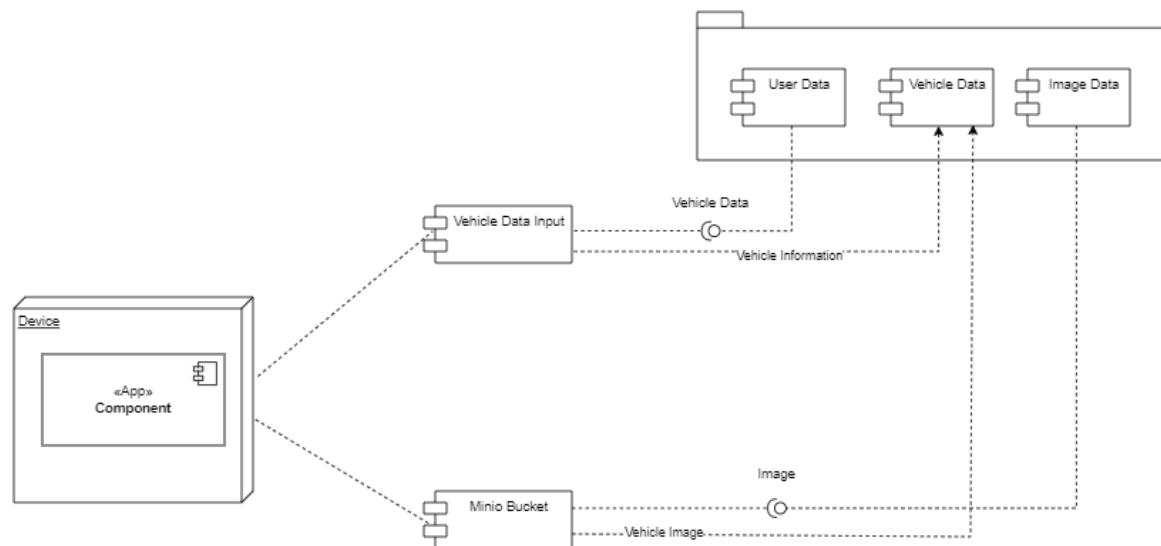


Figure 3.10: Component Diagram

The above component diagram consists of different components, such as Vehicle Data Input, User Data, Vehicle Data, Image Data and Minio Bucket. It includes interfaces such as Vehicle Data, Vehicle Information, Image and Vehicle Image. There is a dependency between Vehicle Data Input component and Vehicle Data component, also between Minio Bucket component and Vehicle Data component. The vehicle data, vehicle information and image are provided to the Vehicle Data component and Minio Bucket respectively.

3.3.2.2.5. Deployment Diagram

Deployment diagram are UML structural diagrams that shows the relationships between the hardware and software components in the system and the physical distribution of the processing i.e., Deployment diagram are used to visualize the topology of the physical components of the system where software components are deployed.

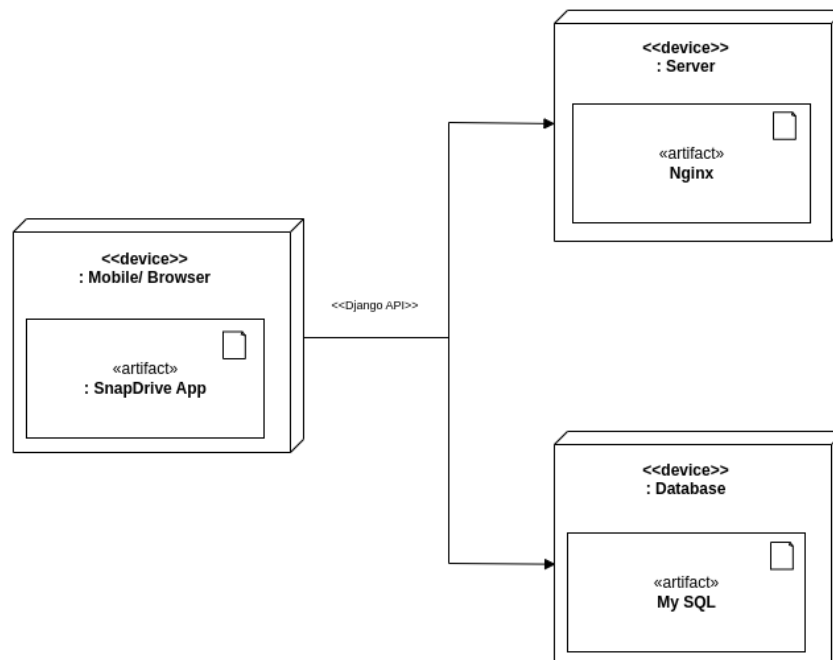


Figure 3.11: Deployment Diagram

The above deployment diagram consists of nodes and artifacts. The Mobile/Browser node is linked to server node and database node through Django API. This API enables the both mobile and web app to send and receive data from the server i.e., Nginx and database i.e., My SQL.

3.4. Tasks Performed

3.4.1. System Implementation

System implementation is the process of turning a conceptual design into a functional system by developing and integrating hardware, software, and other components. The implementation process includes several steps such as gathering requirements, designing the system architecture, developing the necessary software and hardware components, testing the system to ensure its reliability and functionality, deploying it to the production

environment. Effective project management, collaboration, and following the best practices are essential for a successful system implementation.

3.4.1.1. Front End Tools

- React JS

React is a free and open-source front-end JavaScript library for building user interfaces based on components. React was used to develop frontend for SnapDrive web application. It provides different hooks such as useState, useEffect, useContext which helps to build interactive and dynamic interface.

- Ant Design

Ant Design is a React UI library that contains easy-to-use components that are useful for building interactive user interfaces. It is very easy to use as well as integrate. It is used to render form, pop confirm, notifications, loading indicators and so on.

- Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications from a single codebase for any web browser. Flutter was used to develop SnapDrive mobile application. It consists of different widgets that describes what the view should look like given the current configuration and state.

3.4.1.2. Back End Tools

- Django

Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. Django was used for building API for SnapDrive. It is robust, secure, including batteries for essential web functionalities.

- MySQL

MySQL is a widely used relational database management system (RDBMS). MySQL is free and open-source. It is ideal for both small and large applications. It was used as database management system in SnapDrive application for storing user and vehicle data.

- MinIO

MinIO is an object storage solution that provides an Amazon Web Services S3-compatible API and supports all core S3 features. MinIO is built to deploy anywhere - public or private cloud, baremetal infrastructure, orchestrated environments, and edge infrastructure. It was used in SnapDrive application for storing vehicle images.

3.4.1.3. Other Tools

- Docker

Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. It is designed to help developers build, share, and run modern applications. Docker was used in SnapDrive application for containerizing the different services within the application.

- Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform. It allows machine learning developers to narrate visualizations along with sharing the code and data sets. Jupyter Notebook was used during development of SnapDrive mobile application. It was used to create model for vehicle orientation detection.

- Postman

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

3.4.1.4. Development Methodology

The SnapDrive is based on the Incremental Delivery Model. This approach allows to deliver working versions of the project in stages, with each stage building on the previous one. This allows for regular feedback and adjustments to be made throughout the development process. In SnapDrive application the intern started development process according to client requirements and the new requirements were adopted as well as implemented incrementally. In this way, the clear project schedule and milestones was set up to ensure that the project is delivered on time and within budget.

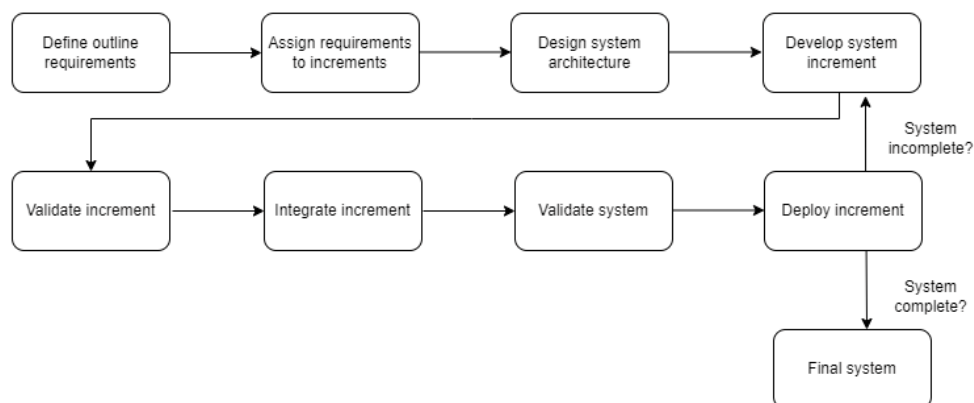


Figure 3.12: Incremental Delivery Model

3.4.2. System Testing

System testing is a crucial phase in the software development life cycle (SDLC) that focuses on verifying and validating the entire integrated system to ensure that it meets its specified requirements and functions as expected.

3.4.2.1. Test Case Design for Unit Testing

Unit testing is the part of the testing methodology which includes testing of individual software modules as well as the components that make up the entire software. The purpose is to validate each unit of the software code so that it performs as expected. In this project, unit testing was performed to validate the functionality of individual modules and components, such as database operations, user interfaces, and business logic. This process helped to identify and rectify any errors or bugs early in the development cycle, ensuring the final product was of high quality and met the desired specifications.

3.4.2.2. Test Case Execution for Unit Testing

The below table is divided into seven column S.N. for Serial Number, Test Case ID for representation of Test Cases, features column for features of that particular test case, test description for detail description of that test case, expected result column for representation of desired outcome, actual result for obtained outcome and finally remarks that signifies the success or failure of the test.

Table 2.2: Test Case for Installation

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
1	TC-INS-01	Application installation	Install SnapDrive app on mobile device	Application opens with splash screen page	Application executed with splash screen	Pass

Table 3.3: Test Case for Application Login

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
------	---------------	-------------------	------------------	------------------	---------------	---------

1	TC-LG-01	Login Success	Enter valid credentials in email and password	Navigate to home screen with success message	Displayed home page	Pass
2	TC-LG-02	Login failure	Enter invalid email or password	Display error message	Displayed “Invalid email or password”	Pass
3	TC-LG-03	Login with empty fields	Leave email or password or both fields empty	Display error message	Displayed “Required”	Pass

Table 3.4: Test Case for Orientation Detection Model

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
1	TC-ODM-01	Accuracy	Calculate accuracy using following formula: Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$	Above 75% accuracy was expected	78.33% accuracy was achieved	Pass
2	TC-ODM-02	Precision	Calculate precision using following formula:	Above 80% precision was expected	85.45% precision was achieved	Pass

			Precision = TP/(TP+FP)			
3	TC- ODM-03	Recall	Calculate recall using following formula: Recall = TP/(TP+FN)	Above 80% recall was expected	90.38% recall was achieved	Pass

Label	Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left	Driverseat	Dashboard
Front	6	0	0	0	0	0	0	0	0	0
Front-Right	1	4	1	0	0	0	0	0	0	0
Right	0	1	4	1	0	0	0	0	0	0
Back-Right	0	0	0	5	1	0	0	0	0	0
Back	0	0	0	0	5	1	0	0	0	0
Back-Left	0	0	0	0	3	3	0	0	0	0
Left	0	0	0	0	0	0	5	1	0	0
Front-Left	0	0	0	0	0	2	1	3	0	0
Driverseat	0	0	0	0	0	0	0	0	6	0
Dashboard	0	0	0	0	0	0	0	0	0	6

Predicted Label

Figure 3.13: Confusion Matrix for Orientation Detection Model

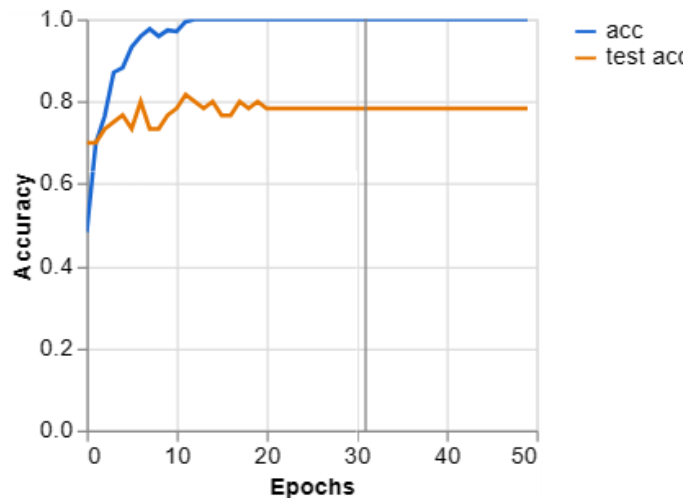


Figure 3.14: Accuracy per Epochs Graph for Orientation Detection Model

Table 3.5: Test Case for Data Input

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
1	TC-INP-01	Input Vehicle Data	Enter vehicle related data and click add to submit	Update vehicle table based on provided data	Vehicle table was updated	Pass

Table 3.6: Test Case for Capturing Image

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
1	TC-CAP-01	Object Detection	Open camera from home screen and detect car in front of camera view	Navigate to Orientation screen for car orientation detection	Orientation screen was displayed	Pass

2	TC-CAP-02	Orientation Screen	Capture the car image with different orientations	Allow to capture car image if orientation matches	Car image was allowed to capture when orientation matched	Pass
---	-----------	--------------------	---	---	---	------

3.4.2.3. Test Case Design for System Testing

System testing is an essential phase in the software development lifecycle. It is a type of testing that evaluates the complete and integrated software system to ensure that it meets the specified requirements and works as expected. The primary goal of system testing is to verify that all components of the software, when combined together, function correctly and interact as intended. In this project, system testing was performed to verify and validate overall functioning of the system. This process helped to identify potential defects, performance bottlenecks, security vulnerabilities, and compatibility issues which reduced the risk of critical failures and ensured that the software behaves as expected when used by end-users.

Table 3.7: System Testing

S.N.	Test Cases ID	Test Case/Feature	Test Description	Expected Results	Actual Result	Remarks
1	TC-INS-01	Application installation	Install SnapDrive app on mobile device	Application opens with splash screen page	Application executed with splash screen	Pass
2	TC-LG-01	Login Success	Enter valid credentials in email and password	Navigate to home screen with success message	Displayed home page	Pass

3	TC-INP-01	Input Vehicle Data	Enter vehicle related data and click add to submit	Update vehicle table based on provided data	Vehicle table was updated	Pass
4	TC-CAP-01	Object Detection	Open camera from home screen and detect car in front of camera view	Navigate to Orientation screen for car orientation detection	Orientation screen was displayed	Pass
5	TC-CAP-02	Orientation Screen	Capture the car image with different orientations	Allow to capture car image if orientation matches	Car image was allowed to capture when orientation matched	Pass

Chapter 4: Conclusion and Learning Outcomes

4.1. Conclusion

The Internship period in KonnectCraft Inc. has given an opportunity to learn and improve software development skills. It helped the intern to acquire new experience of working in professional field and particularly working as a member in a team. Besides that, it provided a chance to work in a project involving different programming languages and platforms connected with company's interest and specialization.

The Vehicle Inventory Management and Orientation Detection systems developed during the internship period helped the intern to get familiar with different frameworks and tools such as React, Flutter, Django, Docker. The application can be used to manage vehicle related data and capture vehicle images with different orientations in a particular order efficiently. In conclusion, this internship period has proved to be effective in enhancing the intern's skills, fostering teamwork, and rewarding for the intern.

4.2. Learning Outcomes

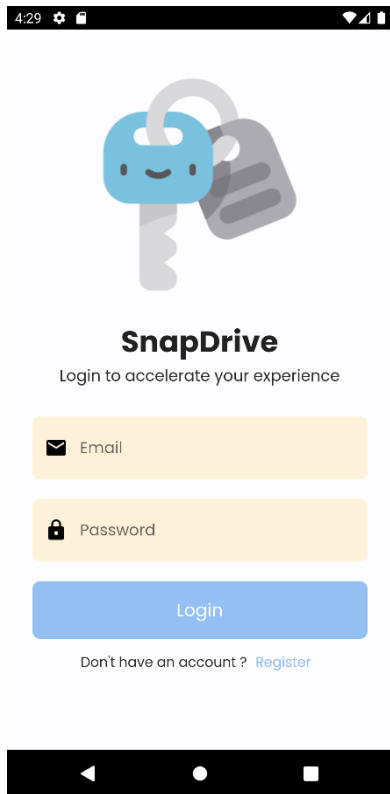
During the internship at KonnectCraft Inc., the intern gained valuable knowledge by working with different technologies like React, Django, Flutter, and Docker. The intern improved his skills in creating responsive and engaging user interfaces using frontend framework like React, building robust API with Django and testing API using Postman, developing intuitive and functional interface for mobile application with Flutter, and managing development and deployment environment using Docker.

Beyond technical skills, the intern also enhanced his problem-solving abilities, teamwork, and adaptability. It generally helps in developing skills such as solving real-world problems, following rules set by the company and working within specified deadlines. The projects and tasks assigned during internships provide technical knowledge and practical experience, which become valuable assets throughout intern's professional career.

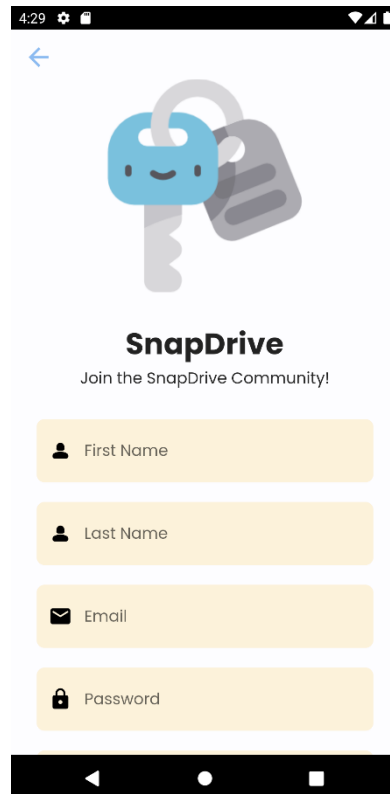
References

- Andrew G. Howard, M. Z. (2017, April 17). Retrieved from arxiv:
<https://arxiv.org/abs/1704.04861>
- Byanjankar, J., Byanjankar, A., & Shrestha, R. (2022). *KonnectCraft*. Retrieved from
KonnectCraft: <https://konnectcraft.com/>
- Gulzar, Y. (2023, January 19). Fruit Image Classification Model Based on MobileNetV2
with. *MDPI Open Access Journals*, 1-14. Retrieved from MDPI:
<https://www.mdpi.com/2071-1050/15/3/1906>
- Manna, A., Upasani, N., Jadhav, S., Mane, R., Chaudhari, R., & Chatre, V. (2023). Bird
Image Classification using Convolutional Neural . *International Journal of
Advanced Computer Science and Applications*, 1-11. Retrieved from thesai:
[https://thesai.org/Downloads/Volume14No3/Paper_97-
Bird_Image_Classification_using_Convolutional_Neural_Network.pdf](https://thesai.org/Downloads/Volume14No3/Paper_97-Bird_Image_Classification_using_Convolutional_Neural_Network.pdf)
- Mr. Harshal Honmote, M. S. (2022, January 01). Real Time Object Detection and
Recognition. *International Journal of Engineering Reasearch & Technology*, 2.
Retrieved from [https://www.ijert.org/research/real-time-object-detection-and-
recognition-using-mobilenet-ssd-with-opencv-IJERTV11IS010070.pdf](https://www.ijert.org/research/real-time-object-detection-and-recognition-using-mobilenet-ssd-with-opencv-IJERTV11IS010070.pdf)

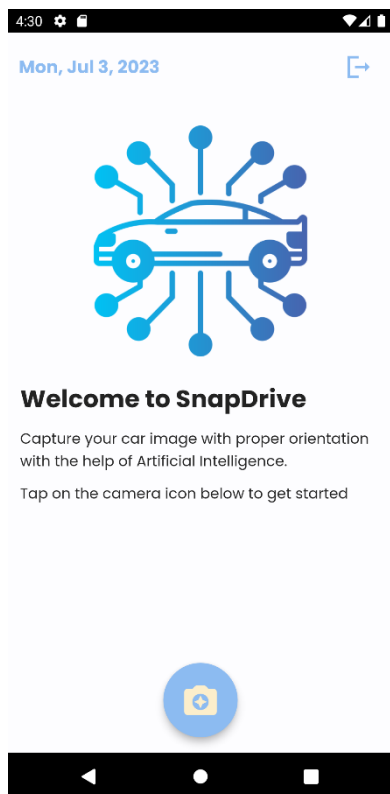
Appendices



Login Screen



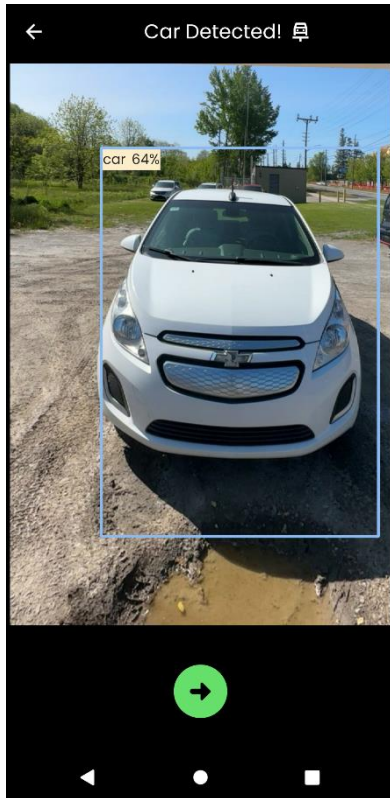
Register Screen



Home Screen



Object Detection



Car Detection



Front View Detection



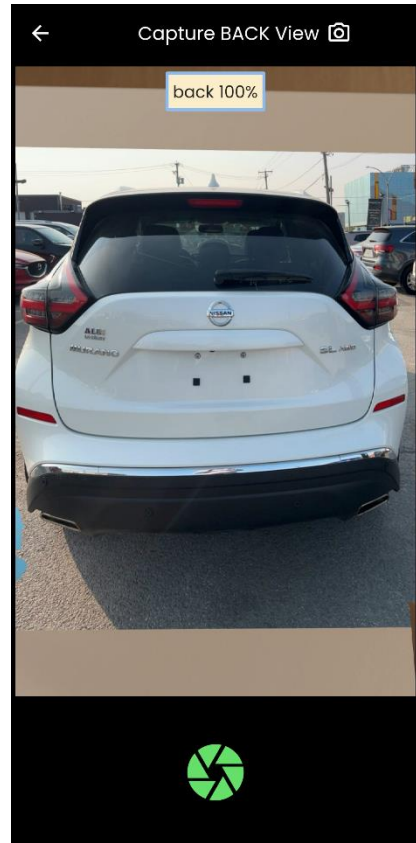
Front-Right View Detection



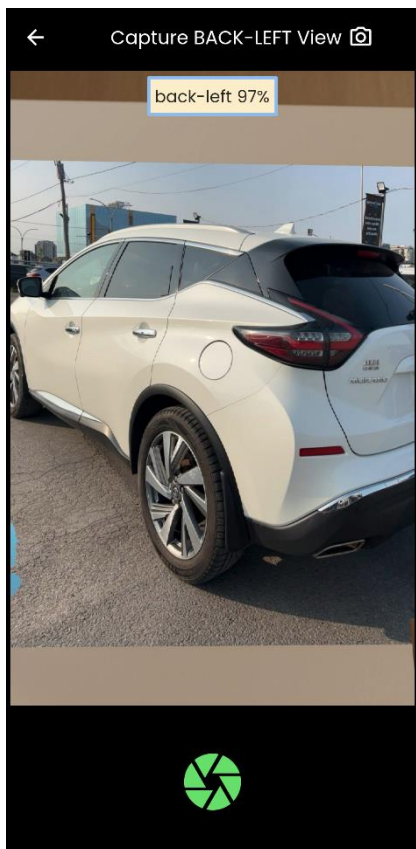
Right View Detection



Back-Right View Detection



Back View Detection



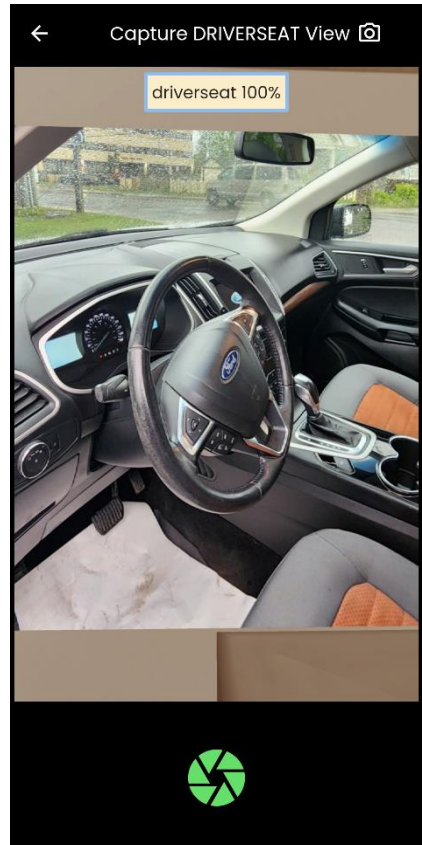
Back-Left View Detection



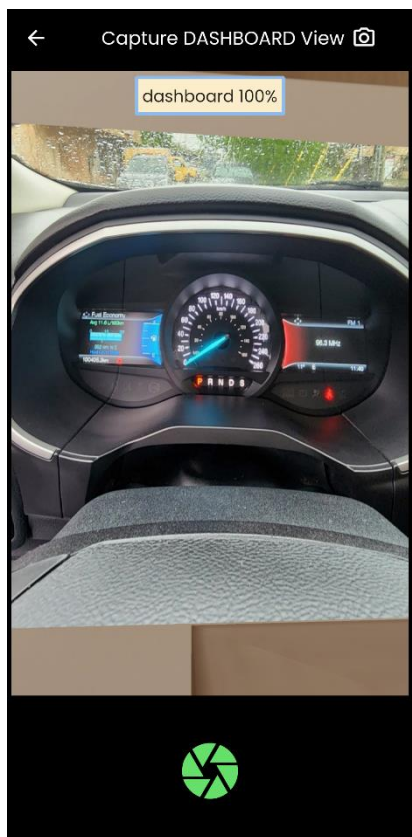
Left View Detection



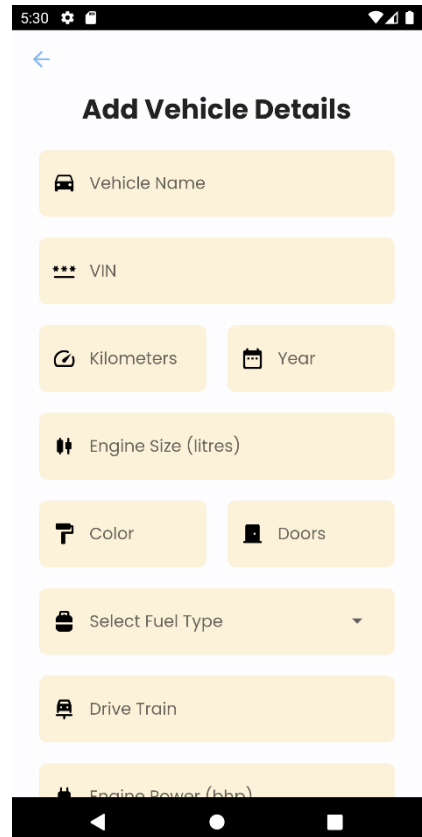
Front-Left View Detection



Driver-Seat View Detection



Dashboard View Detection



Add Vehicle Data



Email

Password

[Sign in](#)

Don't have an account? [Sign up](#)

Login Page

Sign up

First Name

Last Name

Email

Password

Confirm Password

[Sign up](#)

Already have an account? [Sign in](#)

Register Page



Provide Vehicle details below

Vehicle Name:

VIN:

Year:

Engine Size: Doors:

Color:

Fuel Type:

Drivetrain:

Engine Power: Length:

Height: Width:

Kilometers: Cylinders:




Manufacturer:

Images:

[Upload](#)

[Submit](#)

Add Vehicle Data

 <p>Mercedes-Benz C 300, C300-4M</p> <p>2018 123,124 km Gasoline 310 bhp</p> <p>View More Details</p> <p>Delete Edit</p>	 <p>Volkswagen Beetle, 1.8T, TDI</p> <p>2015 114,545 km Gasoline 260 bhp</p> <p>View More Details</p> <p>Delete Edit</p>	 <p>Ford Edge, SEL</p> <p>2018 100,406 km Gasoline 227 bhp</p> <p>View More Details</p> <p>Delete Edit</p>
--	--	---

Vehicle Inventory

Ford Edge, SEL



Year 2018	Color Silver	Kilometers 100,406 km	Doors 4	Fuel Type Gasoline	Engine Size 3.5 litres
Engine Power 227 bhp	Drivetrain Other				

Technical Specification

Engine	Manufacturer: Ford	Engine Size: 3.5 litres
	Cylinders: 6	Engine Power: 227 bhp
Size	Length: 4,778 mm	Width: 4,778 mm
		Height: 1,742 mm

Vehicle Details