PROJECT REPORT

On

Advanced Car Price Prediction Model

Submitted by

Rutul Patel (IU2041220072)
Nidhi Patel (IU2041220065)
Unnati Rawat (IU2041220080)

In fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

Information Technology

INSTITUTE OF TECHNOLOGY AND ENGINEERING
INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ
AHMEDABAD-382115, GUJARAT, INDIA.
WEB: www.indusuni.ac.in

APRIL 2024

PROJECT REPORT

ON

Advanced Car Price Prediction Model

AT

In the fulfillment of the requirement for the degree of

Bachelor of Technology
in
Information Technology

PREPARED BY

Rutul Patel (IU2041220072)
Nidhi Patel (IU2041220065)
Unnati Rawat (IU2041220080)

UNDER GUIDANCE OF

Internal Guide
Prof. Sejal Thakkar
Assistant Professor, CE Department
Indus Institute of Technology & Engineering
INDUS UNIVERSITY

SUBMITTED TO

INSTITUTE OF TECHNOLOGY AND ENGINEERING
INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ
AHMEDABAD-382115, GUJARAT, INDIA,
WEB: www.indusuni.ac.in
APRIL 2024

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Advanced Car Price Prediction Model**" is my own work conducted under the supervision of the guide **Prof. Sejal Thakkar.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B. Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Rutul Patel (IU2041220072)

Guide: **Prof. Sejal Thakkar**
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Advanced Car Price Prediction Model**" is my own work conducted under the supervision of the guide **Prof. Sejal Thakkar.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B. Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Nidhi Patel (IU2041220065)

Guide: **Prof. Sejal Thakkar**
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad
State: Gujarat

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Advanced Car Price Prediction Model**" is my own work conducted under the supervision of the guide **Prof. Sejal Thakkar.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B. Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Unnati Rawat (IU2041220080)

Guide: **Prof. Sejal Thakkar**
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad
State: Gujarat

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING

INFORMATION TECHNOLOGY

2023 -2024

CERTIFICATE

Date:19/04/2024

This is to certify that the project work entitled "**Advanced car price prediction model"** has been carried out by **Rutul Patel** , **Nidhi Patel** and **Unnati Rawat** under my guidance in partial fulfillment of degree of Bachelor of Technology in **Information Technology (Final Year)** of Indus University, Ahmedabad during the academic year 2023 – 2024.

Guide: Prof. Sejal Thakkar	Dr. Seema Mahajan
Assistant Professor,	Head of the Department,
Department of Computer Engineering,	Department of Computer Engineering,
I.T.E, Indus University	I.T.E, Indus University
Ahmedabad	Ahmedabad

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of my Bachelor of Technology Project.

Firstly, I would like to thank my project guide, prof. **Sejal Thakkar**, for their guidance and support throughout the project. Their valuable insights and suggestions have been instrumental in shaping my work.

I am also grateful to the faculty members of the department of information technology engineering for their support and encouragement. Their feedback and suggestions have been invaluable in improving the quality of my work.

I would like to express my gratitude to my family and friends for their unwavering support and encouragement throughout the project. Their constant motivation has been a source of inspiration for me.

Lastly, I would like to thank the university for providing me with the necessary resources and facilities to complete my project successfully.

Thank you all for your support and guidance.

Rutul Patel, Nidhi Patel, Unnati Rawat
IU2041220072, IU2041220065, IU2041220080
Computer Engineering

TABLE OF CONTENT

ABSTRACT

Determining the fair price of a used car, especially when factoring in potential damages, can be a significant challenge. Our project, Advanced Car Price Prediction Model, provides users with a tool to determine accurate used car prices. Unlike traditional models that rely solely on car features, our model utilizes a cutting-edge object detection system powered by computer vision.

The web interface provides the user with seamless support to find the worth of their used car. It is integrated with a direct data pipelining, which enhances the user experience and supports them in detecting damages on their car immediately after assessing the price. The user's journey begins with an input form via the interface medium, where they can enter the features of their used car, like mileage, kilometers driven, fuel type, manufacturer, transmission, etc. On form submit, the machine learning model that we have trained will calculate the base price of the car using the features that were provided by the user. Later on, users will be prompted to indicate if they have any damages on their car. If they agree to that question, then they will be directed to the Damage Detection Model to upload images where damage is present. Post-assessment, a final price will be shown to the user that will be a price lower than the base price that was calculated previously.

This project stands as an innovative idea with a huge potential in the automobile industry. It aims to build trust between sellers and buyers of used cars by bringing a sense of transparency and honesty between the parties involved.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

Abbreviations used throughout this whole document for Survey Application are:

HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Java Script
XML	Extensible Markup Language
UML	Unified Modeling Language
CSS	Cascading Style Sheet
DBMS	DataBase Management System
My SQL	My Structured Query Language
XAMPP	X-operating system, Apache, Mysql, Php, Perl
AJAX	Asynchronous JavaScript And XML
HOD	Head of the Department

CHAPTER 1

INTRODUCTION

- **PROJECT SUMMARY**
- **PROJECT PURPOSE**
- **PROJECT SCOPE**
- **OBJECTIVES**
- **TECHNOLOGY AND TOOLS**
- **SYNOPSIS**

1.1 PROJECT SUMMARY

The used car market often lacks transparency, making it challenging for both buyers and sellers to determine fair prices, especially when potential damages are involved. Our Advanced Car Price Prediction Model introduces an innovative solution to this problem. It combines a traditional machine learning-based price prediction model with a ***state-of-the-art*** computer vision-powered damage detection system. This integrated approach provides a comprehensive assessment of a used car's value, taking into account both its standard features and any existing damages. This technology has the potential to transform the used car industry. It fosters trust by providing both buyers and sellers with accurate, unbiased valuations, leading to fairer transactions and potentially reducing the need for time-consuming negotiations.

Our project empowers users with a straightforward web interface. Initially, users enter their car's details, generating a base price prediction from our machine learning model. Subsequently, they're given the option to utilize the damage detection system. By uploading images of their vehicle, our object detection model identifies and analyzes any damage present. This analysis results in a final adjusted price, reflecting both the initial predicted value and deductions for detected damages. Ultimately, our project aims to revolutionize the used car market by promoting transparency, fairness, and efficiency for all parties involved.

1.2 PROJECT PURPOSE

The used car market often operates with significant information asymmetry between buyers and sellers. This project seeks to bridge this gap by providing a comprehensive car valuation tool that combines traditional price prediction with cutting-edge damage detection. Our goal is to empower users with the knowledge they need to make confident and informed decisions about buying or selling a used vehicle. Our key focus areas are to:

- Develop a robust machine learning model to accurately predict used car prices, considering factors beyond standard features.
- Utilize computer vision to identify and assess various types of car damage, ensuring no surprises for buyers or sellers.
- Provide a detailed breakdown of the factors influencing a car's valuation, fostering trust and understanding.
- Create an intuitive web interface that simplifies the valuation process for all users, regardless of technical expertise.
- Aim to increase overall confidence in used car transactions by reducing uncertainty and potential for disputes.

1.3 PROJECT SCOPE

The Advanced Car Price Prediction Model prioritizes creating a user-friendly system empowering both buyers and sellers in the used car market through data-driven valuations. This project focuses on demonstrating the core functionalities of the backend models. These functionalities include:

- A user-friendly interface that guides users through entering car details and receiving base price estimates.
- Damage detection with price adjustments, utilizing a YOLOv4 model to identify and classify damage in uploaded pictures, and reflecting the impact on the car's value.
- Data-driven valuations based on extensive car sale datasets carefully curated from reputable sources. These sources include car dealerships, auction websites, and automotive pricing databases, ensuring the model considers a broad range of real-world factors influencing car value.

It's important to note that this project demonstrates the capabilities through a limited front-end integration, and doesn't encompass a fully-fledged, industry-ready website. Additionally, hardware limitations restricted training to the YOLOv4 model on our laptops, preventing exploration of potentially more advanced algorithms. Despite these limited scope for the project, it successfully establishes a foundation for a comprehensive used car valuation system, particularly targeting individual car buyers and sellers seeking a more objective valuation process. This system can empower them to make informed decisions in the used car market, fostering a more transparent and trustworthy experience for all parties involved.

1.4 OBJECTIVES

This project aims to revolutionize the used car valuation process by addressing the challenges of uncertainty and potential bias. We will develop, integrate, and rigorously evaluate cutting-edge technologies in machine learning and computer vision to achieve these goals. The project's objectives encompass the creation of accurate predictive models, the development of a user-centric interface, and the assessment of the system's broader impact on the used car market. Moreover, we have provided more of an interface than a website, implying that it can be integrated and deployed anywhere, as per the requirements.

1.4.1 MAIN OBJECTIVES

The main objective of the project on Advanced Car Price Prediction Model was to design and train a machine learning model that integrates car feature data with damage assessment from the image-based detection system, and thoroughly evaluate model performance using relevant metrics.

Moreover, we have built an intuitive web interface that seamlessly guides users through the valuation process, from feature input to damage detection and final price estimation.

1.4.2 SECONDARY OBJECTIVES

To analyze the potential of this technology to reshape the used car market, exploring aspects such as increased transparency, fairer pricing, and enhanced trust between buyers and sellers, with a specific focus on market side objectives. The buyers can be rest assured that the car they want to purchase will be worth the price that is listed on the app/website, and that the seller isn’t just any unreliable source of information. Finally, the integration of machine learning and computer vision in our valuation tool helps eliminate subjectivity and potential bias from the pricing process.

1.5 TECHNOLOGY AND TOOLS

Front-end technologies used in this project are:- HTML, CSS, Javascript and Jinja. Back-end technologies used in this project are:- Python and Flask.

1.5.1 FRONT-END TECHNOLOGY

- **HTML5:** The core of web development, HTML5 defines the content structure and semantic meaning of web pages. Its advancements in multimedia support (audio, video), offline storage capabilities, and improved form elements have transformed how web applications function and interact with users.
- **CSS3:** CSS (Cascading Style Sheets) extends the core styling capabilities of web pages, introducing features like gradients, animations, and flexible layouts (e.g., flexbox, grid). This allows developers to create visually stunning, responsive interfaces that adapt seamlessly to different screen sizes and devices.
- **Javascript:** JavaScript powers the interactive nature of modern websites. It handles everything from simple button clicks and form validations to complex animations and data fetching from the backend. JavaScript's ability to manipulate the DOM (Document Object Model) enables the creation of dynamic and engaging user experiences.
- **Jinja**: Jinja is a robust Python-based templating engine. It simplifies the process of constructing HTML pages by providing placeholders for variables, conditional logic, and reusable blocks of code. Jinja promotes clean separation between presentation logic and backend data, enhancing code maintainability.

1.5.2 BACK-END TECHNOLOGY

- **Python:** Python's popularity stems from its readability, vast ecosystem of libraries, and suitability for diverse programming tasks. In web development, Python shines due to its clear syntax and beginner-friendliness, making it an excellent choice for rapid prototyping and web application development. Its extensive standard library provides essential tools for building web servers, handling HTTP requests, and

interacting with databases. Additionally, Python's strength in data manipulation and analysis make it a perfect fit for web applications that process and leverage large datasets. This is particularly advantageous for projects involving machine learning or computer vision, where Python offers a rich set of libraries like NumPy, Pandas, Scikit-learn, Tensorflow and PyTorch.

- **Flask:** Flask is a microframework known for its flexibility and ease of use. It offers essential components for web development while leaving many design choices to the developer. This allows for highly customized applications built with only the necessary components, leading to a more lightweight and performant final product.

1.6 SYNOPSIS

This project tackles the uncertainty and potential for unfairness in the used car market by developing a comprehensive valuation tool. Utilizing machine learning (Random Forest), Object detection (YOLOv4), and web technologies (HTML, CSS, JavaScript, Flask), we built a system that predicts car prices based on standard features while also detecting and factoring in damages. The user-friendly web interface guides users through the valuation process, providing a transparent breakdown of factors influencing the final price. This project aims to promote fairer transactions, increase trust, and potentially transform how used cars are valued.

CHAPTER 2

LITERATURE SURVEY

- **INTRODUCTION OF SURVEY**
- **WHY SURVEY?**

2.1 INTRODUCTION OF SURVEY

Our inspiration for this project idea came from some research papers, some of which are discussed below.

- In the paper [Integrated Deep Learning System for Car Damage Detection and Classification Using Deep Transfer Learning], the idea is to delve into the application of deep learning models, particularly convolutional neural networks (CNNs), for car damage detection and classification. A key focus of the research lies in the use of transfer learning, where pre-trained CNN models (like VGG16 and ResNet 50) are adapted to the specific task. The system is trained on a dataset of car images labeled with various damage types and severity. The paper demonstrates the effectiveness of this approach through performance metrics such as accuracy, precision, and recall, suggesting that transfer learning could be a valuable time-saving technique for any similar project. We got our idea to use this approach instead of a start from scratch.

- On a similar yet different practical application, this paper [Preventing Car Damage using CNN and Computer Vision] proposes a system utilizing CNNs to detect car damage, aiming to prevent insurance fraud. It introduces a customized CNN model for classifying a range of damages and an additional component for estimating repair costs. While the primary focus is on fraud prevention, the techniques demonstrated for damage detection hold relevance for our used car valuation project. Moreover, the concept of estimating repair costs could spark ideas in the future for industry use systems.

- More support to the idea of success of this topic was given by this paper [Car Damage Detection and Classification]. This research explores the creation of a CNN-powered system for detecting and classifying various car damages, including scratches, dents, and broken glass. The paper emphasizes the importance of a dataset with diverse damage examples in training a successful model. The promising performance reported in the study offers validation for our decision to use CNNs in our damage detection module.

- In [Car Damage Detection and Analysis Using Deep Learning Algorithm For Automotive], the paper presents a deep learning system for car damage detection and analysis aimed at streamlining insurance claim processes. Using

Convolutional Neural Networks (CNNs), the system can identify and classify various types of car damages from images. This technology has the potential to significantly accelerate claim assessments by automating a portion of the inspection process. The ability to automatically analyze damage images could reduce manual effort and minimize subjective judgment calls, potentially leading to faster and more consistent insurance claim resolutions. Thus, this gave us the idea to implement a system where the same model (Damage Detector), can be used for the valuation of used cars.

2.2 WHY SURVEY?

Conducting surveys plays a crucial role in refining the damage detection model for this project. In the realm of computer vision and object detection, obtaining feedback from potential users is invaluable. Gathering insights about the real-world prevalence of different damage types, their characteristics, and variations in how they appear in images directly supports the model development process. Surveys ensure that the model is trained on an extensive and representative dataset, addressing the range of damages that are likely encountered in practical use. Furthermore, they can uncover users' pain points or expectations regarding damage detection accuracy and the level of detail provided. This user-centric feedback guides the optimization of the model, ensuring it not only identifies damages but presents the results in a manner that aligns with the needs of those assessing the used car's condition. Ultimately, a survey-driven approach helps bridge the gap between technical performance and real-world utility, enhancing the value out project delivers to both buyers and sellers.

CHAPTER 3

PROJECT MANAGEMENT

- **Project Planning objectives**
- **Project Scheduling**
- **Risk management**

3.1 PROJECT PLANNING OBJECTIVES

The project planning objectives include selecting software, examining its scope, and developing a project development approach using the Software Development Life Cycle (SDLC).

- The primary purpose of any project is to be kept at utmost priority, and hence, outline the features and functionalities that will be within the project's boundaries.
- Factoring into the price prediction for a fair evaluation, we selected- kilometers driven, mileage, fuel type, transmission, build type and manufacturer. Those account for calculating the initial base price.
- For the object detection model, the damage detection model analyzes images and finds damaged areas accurately, categorizing them into 14 categories of where the damage is located on the car.

3.1.1 SOFTWARE SCOPE

Visual Studio Code is an easy to use source code editor. Its strength lies in its ability to seamlessly integrate both Python and HTML, the cornerstones of our back-end and front-end respectively. This eliminates the need to juggle between separate environments for each language, boosting our productivity significantly. VS Code's built-in Python support offers features like syntax highlighting, code completion, and debugging tools, making Python development a breeze. For the front-end, HTML benefits from VS Code's intelligent code formatting and code snippets, speeding up the creation of well-structured and efficient user interfaces. This seamless integration between languages fosters a smooth workflow, allowing us to concentrate on the logic and functionality of our project without getting dragged down by technical hurdles.

3.1.2 PROJECT DEVELOPMENT APPROACH

Software development often follows a structured approach known as the Software Development Life Cycle (SDLC). This model outlines distinct phases, each with specific goals and outcomes. The output of one phase becomes essential input for the next. Starting with analyzing requirements, developers transform them into detailed designs. This design then guides the coding or development phase. Alongside development, the testing team follows a parallel process – the Software Testing Life Cycle (STLC) – to rigorously evaluate the software's quality and functionality; but that comes later. There are following six phases in every Software development life cycle model:

- Requirement Gathering and Planning
- Analysis
- Design
- Implementation or Coding
- Testing and Integration
- Deployment and Maintenance

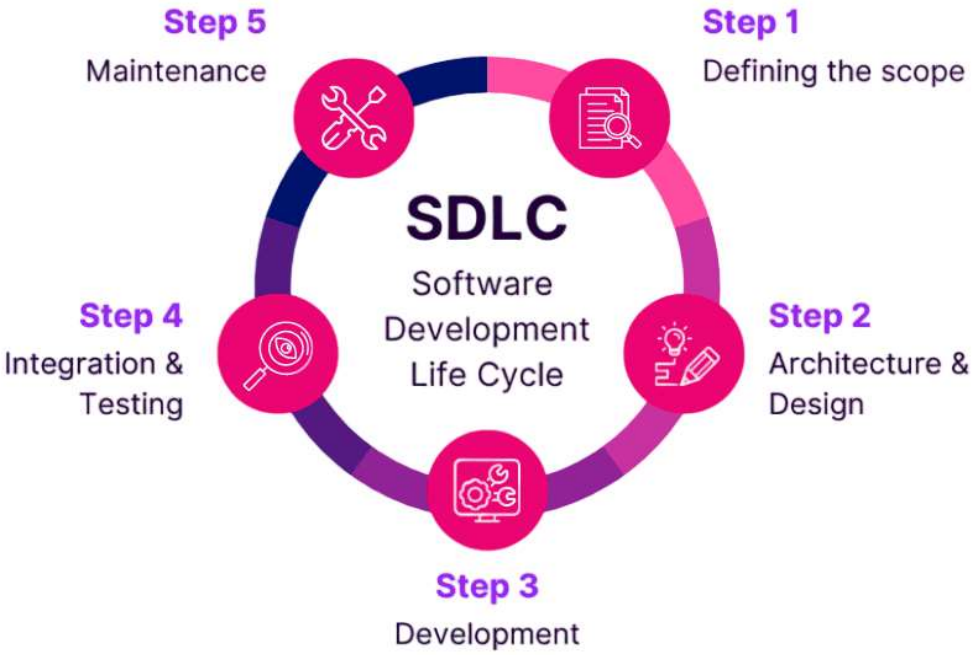

Fig 3.1.2.1: SDLC

3.2 PROJECT SCHEDULING

Project schedule means a mechanism that is used to communicate and know the tasks that are needed and have to be done or performed. Effective project scheduling leads to success, reduced cost, and increased customer satisfaction. Scheduling in project management means listing out activities, deliverables, and milestones within a project that are delivered.

3.2.1 BASIC PRINCIPLE SCOPE

The project began by defining its scope and objectives. This included gathering requirements and selecting the appropriate languages and frameworks. Next, we conducted thorough research on the project concept. As this research yielded insights, we iteratively refined the project's direction. Ultimately, we solidified the basic principle scope, finalizing our choices for front-end technologies (HTML, CSS, JavaScript, and Jinja) and back-end tools (Python, Flask, OpenCV, Keras, and Darknet).

3.2.2 PROJECT ORGANIZATION

Project Planning: Project organization and planning were instrumental to the success of this endeavor. We employed a structured approach, incorporating a timeline with milestones for each key phase (data collection, model development, integration, testing). Task tracking tools aided in monitoring progress and identifying potential bottlenecks. Clear documentation of requirements, design choices, and code ensured maintainability and knowledge sharing. Regular communication fostered collaboration and facilitated timely problem-solving. A risk assessment plan helped anticipate challenges and implement mitigation strategies, ensuring the project remained on track.

Project Scheduling: Software project scheduling is an activity that distributes estimated efforts across the planned duration by allocating the effort to specific software engineering tasks. Proper Scheduling requires:

1. All tasks appear in the system.
2. Effort and timing are intelligently allocated to each task.
3. Interdependencies between tasks are properly indicated, thus increasing efficiency.
4. Resources are allocated for the work to be done, alongside dividing work between the members.

3.2.3 TIMELINE CHART

3.2.3.1 Time Allocation

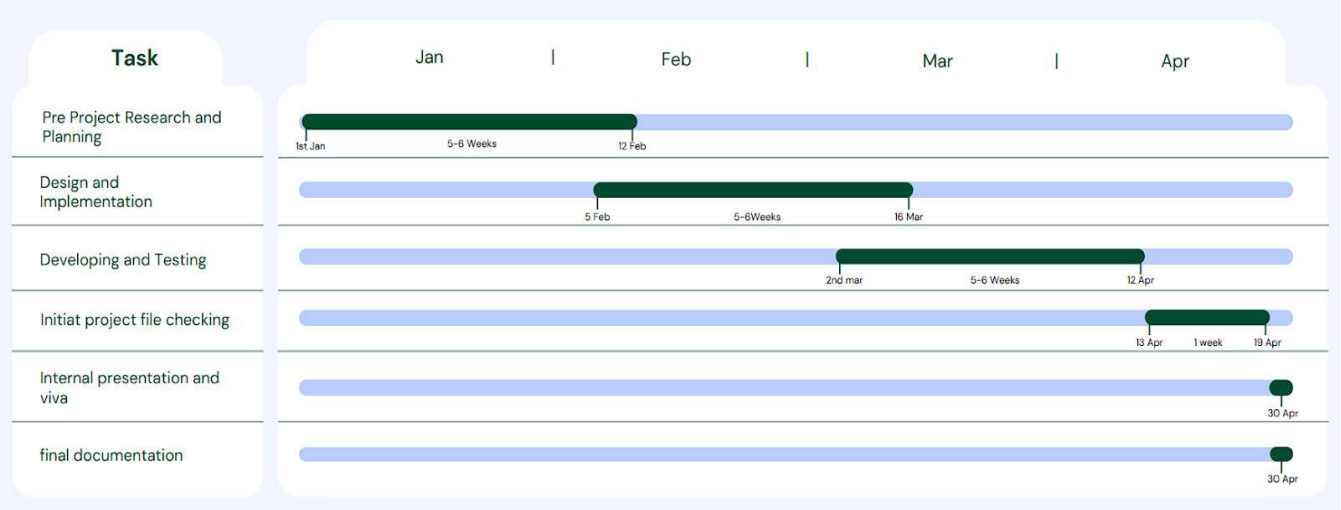

Fig 3.2.3.1: Time Allocation Chart

3.2.3.2 Task Sets

PROJECT CRITERIA	PRESENTATION
Pre Project Research and Planning	project definition, project scope and objectives , a detailed project plan including timelines and methodologies.
Design and data collection	Design system architecture and UI, gather and prepare the data, Implement basic functionalities.
Developing and Testing	Model Selection, Initial Model Training, integrate components, and implement algorithms, Evaluate the performance of project
Initial project file checking	Review and finalize project design documents, data collection plans, and code documentation.
Internal presentation and viva	a comprehensive presentation summarizing project, including objectives, methodologies, results, and conclusions.
final documentation	Final hard copy of the file after the changes in the previous phases.

Fig 3.2.3.2: Task Sets

3.3 RISK MANAGEMENT

Risks are unexpected events that can throw your project off track. Software development is tricky – even the best code can sometimes misbehave. Things break for reasons we don't always predict, and that's just how it is for any developer. That's why it's important to identify potential risks early on. By understanding where things could go wrong, we can better manage those uncertainties and keep the project on course.

3.3.1 RISK IDENTIFICATION

The following are the possible risks associated with this project:

1. Technical risks: If the internet connection is poor, there may be some visual bugs. A future update on the website may lead to its shutdown. The website may be slow on some devices. Also, the number of requests at a time to the server may affect the processing times, deteriorating performance.
2. Project Risks: Limited car image datasets or inconsistent labeling could hinder the damage detection model's training and performance. Variations in how real-world car prices fluctuate over time might decrease price prediction model accuracy. Overly complex machine learning models could be prone to overfitting or might need long training times, as it happened with us in the beginning. Integration issues like difficulties in smoothly integrating front-end components with the back-end (damage detection, price prediction models) could lead to a less user-friendly experience.

3.3.1.1 Risk Identification artifacts

Risks vary significantly across software systems. Some projects might face internal risks, such as developer oversights or inadequate testing of critical components. Others might grapple with less obvious risks tied to non-functional aspects, like performance bottlenecks that prevent the system from handling large volumes of requests simultaneously.

3.3.2 RISK PROJECTION

During the risk projection, each identified risk is considered in turn, and a judgement is made about the probability and seriousness of the risk.

Risk	Probability	Effects
Major upgrade in Python	High	Normal
Security issue	Medium	Moderate
Failure to train satisfactory predictive models	High	Catastrophic
Performance Bottlenecks	Low	Normal

CHAPTER 4

SYSTEM REQUIREMENTS

- **User Characteristics**
- **Functional Requirement**
- **Non Functional Requirement**
- **Hardware and Software Requirement**

4.1 USER CHARACTERISTICS

The car price prediction software is a straightforward tool designed to estimate the value of used cars, taking into account potential damages. Accessible through a website, it can be used by anyone with a web browser and an internet connection. This ensures the system functions seamlessly on a wide range of devices.

4.2 FUNCTIONAL REQUIREMENT

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. The functional requirements of *Advanced Car Price Prediction* are:

- User-Friendly Interface: The system will provide a clear and intuitive interface, guiding users through the valuation process with helpful instructions and visual cues.
- Core Price Prediction: The system will allow users to enter essential car details (make, model, mileage, year, etc.) and generate a baseline price prediction using a reliable machine learning model.
- Damage Detection and Assessment: Users will be able to upload images of their vehicle. The system will employ an accurate object detection model to identify and categorize various types of car damage.
- Integrated Price Adjustment: The system will incorporate detected damages into the price calculation. It will transparently present the estimated price reduction due to each damage type, ensuring a fair and justifiable valuation.
- Guidance and Explainability: The system will provide brief explanations alongside the price prediction and damage assessments. This will help users understand the factors influencing the valuation.

4.2.1 ACTIVITY AND PROPOSED SYSTEM

The proposed system will enable users to initiate the car valuation process by entering basic car information (make, model, mileage, condition, etc.). They will have the option to upload images of their car for damage assessment. The system will then employ machine learning models to predict a baseline

price and identify any damages present. Based on the detected damages, the system will calculate an adjusted price, providing a transparent breakdown of the price reduction factors. Finally, the system will present the final valuation to the user, along with a visual analysis of any detected damages.

4.3 NON FUNCTIONAL REQUIREMENT

Non-functional requirements are the characteristics and constraints that a system must meet to satisfy its intended use. Non-functional requirements of *Advanced Car Price Prediction* includes:

- Performance: The system should provide timely price predictions and damage assessments to ensure a responsive user experience.
- Accuracy: The price prediction and damage detection models should achieve a high level of accuracy to instill trust in the valuations.
- Usability: The user interface should be intuitive and easy to navigate, regardless of the user's technical expertise.
- Reliability: The system should operate consistently and predictably, minimizing downtime or unexpected errors.
- Scalability: The system should be able to handle an increasing number of users and image uploads without significant performance degradation.
- Security: User data (if collected) and images should be protected with appropriate security measures.

4.4 HARDWARE AND SOFTWARE REQUIREMENT

4.4.1 HARDWARE REQUIREMENT

- **Memory:** 8 GB
- **CPU:** AMD Ryzen 5 4600H with Radeon Graphics
- **GPU:** NVIDIA GeForce GTX 1650 Ti
- **Storage**: None

4.4.2 SOFTWARE REQUIREMENT

- **Operating system :-** Windows, Linux, Android, MAC
- **Software system** :- Web Browser (Any)
- **Windows system :-** Windows XP or higher

CHAPTER 5

SYSTEM ANALYSIS

- **Study of Current System**
- **Problems in Current System**
- **Requirement of New System**
- **Process Model**
- **Feasibility Study**

5.1 STUDY OF CURRENT SYSTEM

In the current system, assessing the value of a used car often involves a time-consuming and location-dependent process. Traditionally, sellers rely on visiting dealerships or mechanics for appraisals, which can necessitate travel and scheduling challenges. Furthermore, the subjectivity inherent in human valuation can lead to inconsistent pricing. Most "used car seller" systems currently function online, and they depend on the seller estimating a price, introducing the risk of subjectivity. Buyers might initially like the listed price but could discover damages upon physical inspection that the seller hadn't factored into the asking price. This lack of transparency impacts the valuation process, potentially leading to dissatisfaction on both sides. Moreover, sellers might struggle to understand the rationale behind an appraisal, particularly regarding the impact of any existing damages on the car's worth. Additionally, the current system often presents challenges in managing and maintaining records of past valuations and supporting documentation. For individual sellers, keeping track of past appraisals and associated paperwork can be cumbersome and prone to loss or misplacement. This lack of a structured data management system hinders efficient transactions within the used car market, making it difficult for both buyers and sellers to access clear and reliable information.

5.2 PROBLEMS IN CURRENT SYSTEM

The current used car valuation system presents several issues. The process often lacks transparency, with sellers struggling to obtain reliable price estimates and buyers facing uncertainty about the true condition of a vehicle. Existing online pricing platforms might be prone to subjective valuations, failing to fully account for damage that could significantly impact a car's value. Moreover, without tools to objectively assess car condition, buyers and sellers risk encountering mismatches between expectations and reality. The absence of standardized damage detection techniques can lead to time-consuming negotiations and potentially unfair transactions. Additionally, maintaining accurate records of past valuations and supporting evidence (such as images or inspection reports) can be disorganized and

difficult for individual sellers, hindering historical reference and market analysis. This overall lack of a centralized, data-driven system creates inefficiencies and a potential for frustration on both the buyer and seller sides of the marketplace.

5.3 REQUIREMENT OF NEW SYSTEM

To overcome the limitations of the current system, a new solution is needed that prioritizes transparency, accuracy, and efficiency for both buyers and sellers. The system should feature a user-friendly interface to streamline the valuation process, employ a robust machine learning model for objective pricing, and integrate damage detection capabilities for accurate assessments. Transparency is key, with the system providing clear explanations of price adjustments due to detected damages. Data-driven guidance should accompany the valuation to build user trust. Optional features like record-keeping for past valuations and strong security measures would further enhance the system's usability and value, and can be implemented on the go.

5.4 PROCESS MODEL

The Spiral model is a flexible Software Development Life Cycle (SDLC) approach known for its emphasis on risk management. Visually, it resembles a spiral with multiple loops, where each loop represents a phase of development. The number of phases can be tailored by the project manager based on the project's specific risk profile. This model breaks down development into iterative modules, allowing requirements, design, implementation, and testing to occur within each module. Subsequent releases build upon previous ones, resulting in a progressively complete system.

The spiral's radius symbolizes the cumulative project cost, while its angular dimension indicates progress within the current phase. Each phase has four quadrants:

Objectives and Alternatives: This quadrant initiates the phase by gathering requirements from stakeholders. These requirements are then analyzed and refined to define clear, measurable objectives for the current development cycle. Once the objectives are established, the team explores potential solutions (technical approaches,

design concepts) that could achieve those goals. This exploration might involve brainstorming sessions, feasibility studies, or preliminary prototyping to assess the viability of different options.

Risk Analysis and Resolution: With a set of potential solutions identified, the team moves into risk assessment. Here, they meticulously evaluate each solution to identify potential risks associated with development, deployment, or user adoption. Once risks are recognized, the team develops mitigation strategies to address them. These strategies could involve adopting specific technologies, implementing additional testing measures, or even revising the chosen solution if the risks are deemed too high. Prototyping often occurs in this quadrant to validate the feasibility of the chosen solution and further refine risk mitigation strategies.

Development and Testing: Having identified the optimal solution and addressed potential risks, the team embarks on development. This quadrant focuses on building the software components according to the chosen solution and established requirements. Rigorous testing is performed throughout this stage to ensure the developed software meets quality standards and user expectations. The outcome of this phase is a new, functional version of the software, ready for deployment or further iteration.

Review and Planning: The final quadrant of the spiral phase focuses on evaluation and planning for the next iteration. Stakeholders and the development team collaboratively assess the newly developed software version. This evaluation considers factors like functionality, usability, and alignment with project objectives. Feedback gathered during this review is then incorporated into the planning process for the subsequent development phase, ensuring the project continues to evolve in a controlled and iterative manner.

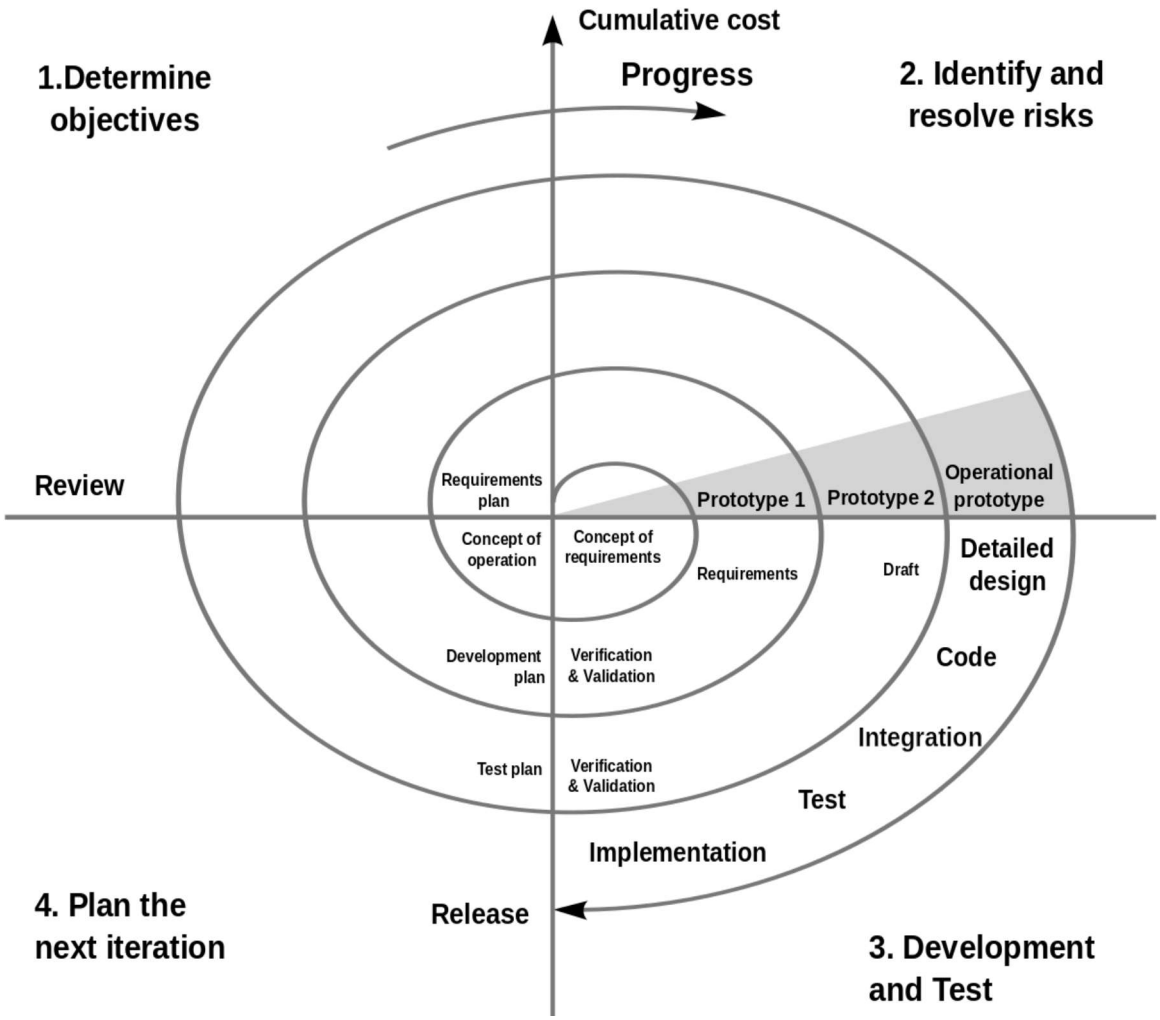

Fig 5.4.1: Spiral Model

5.5 FEASIBILITY STUDY

5.5.1 TECHNICAL FEASIBILITY

The system leverages robust and widely-used technologies to ensure a successful implementation. For the backend, Python offers a rich ecosystem of libraries and frameworks well-suited for data manipulation, machine learning, and web development tasks. Specific libraries utilized in this project include *Flask, Numpy, OpenCV* and *Keras*. Other supporting frameworks used are: *Darknet.* For the frontend, *HTML* and *CSS* provide a solid foundation for creating user-friendly, responsive interfaces across devices. Additionally, the incorporated JavaScript with Jinja templating enhances dynamic elements and interactions. A cloud-based platform (like AWS, Google Cloud, Azure) will ensure scalability, accommodating fluctuations in user traffic and image storage needs. Cloud services can also streamline model deployment and potentially offer pre-built machine learning tools and infrastructure.

5.5.2 OPERATIONAL FEASIBILITY

The system is designed for ease of use and maintainability. Its intuitive interface and clear instructions will minimize the need for extensive user training. The modular design allows for seamless upgrades of the machine learning models (pricing, damage detection). As market trends or economic factors evolve, the system can easily incorporate improved models without taking down the system for a software update, and potentially disrupting the core user experience. Additionally, established technologies and cloud deployment ensure operational reliability and access to skilled resources if needed.

5.5.3 ECONOMICAL FEASIBILITY

The project's automated design offers significant cost advantages. The lack of a database and manual admin interface streamlines development, reducing both initial setup and ongoing maintenance costs. Additionally, the use of open-source technologies (Python, libraries) eliminates licensing fees. Cloud deployment allows for a pay-as-you-go cost model, ensuring expenses scale with usage. These factors, combined with the system's potential to enhance the efficiency of used car valuations, demonstrate a favorable cost-benefit ratio.

5.5.4 SCHEDULE FEASIBILITY

Schedule Feasibility an organization estimates how much time the project will take to complete. When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including: Internal Project Constraints: Technical, Technology, Budget, Resource, etc

CHAPTER 6

DETAIL DESCRIPTION

- **Customer Module**

6.1 CUSTOMER MODULE

The Customer Module is an essential part of any Car Price Prediction System. This module enables users to place orders and make payment through the platform.

6.1.1 Features - Input

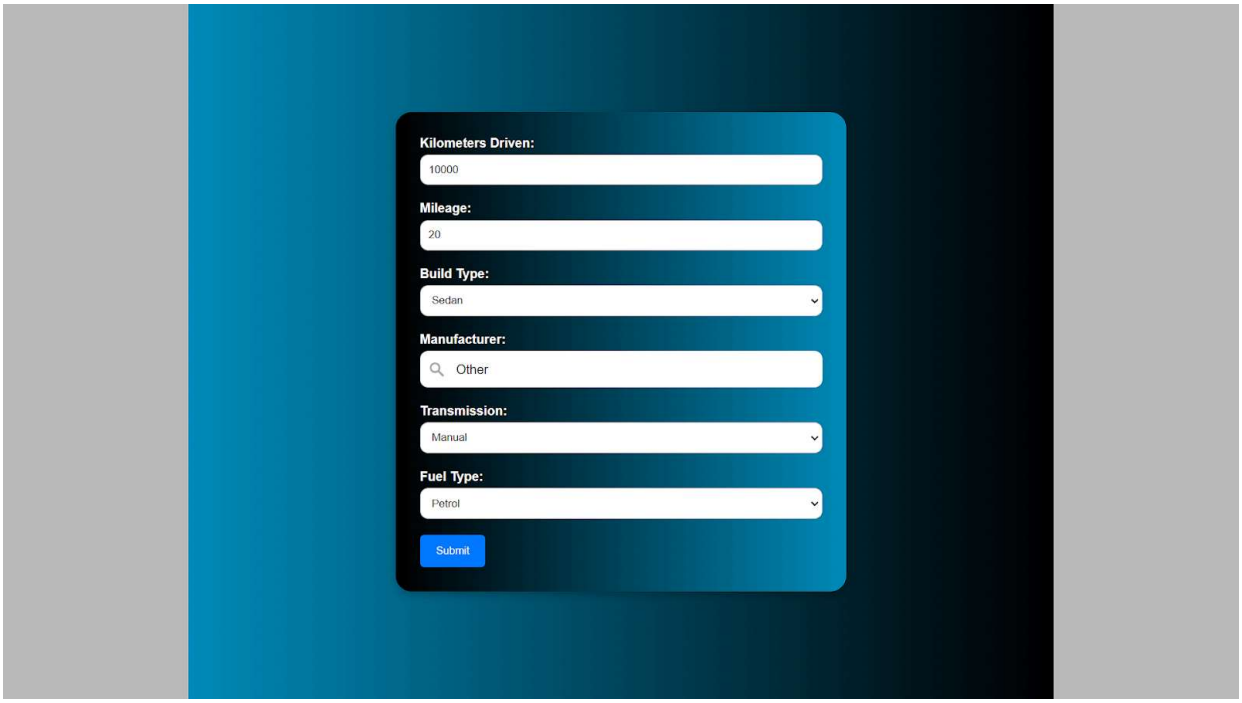

Fig 6.1.1: Feature Input Form

➔ This page acts as the entry point for users seeking a valuation for their used car.

➔ It presents a feature input form where users provide essential details about their vehicle.

➔ Upon submitting the form, the website calculates a preliminary valuation based on the provided data.

➔ Then, the backend returns a base price, which is displayed to the user.

6.1.2 Display Page: base_price

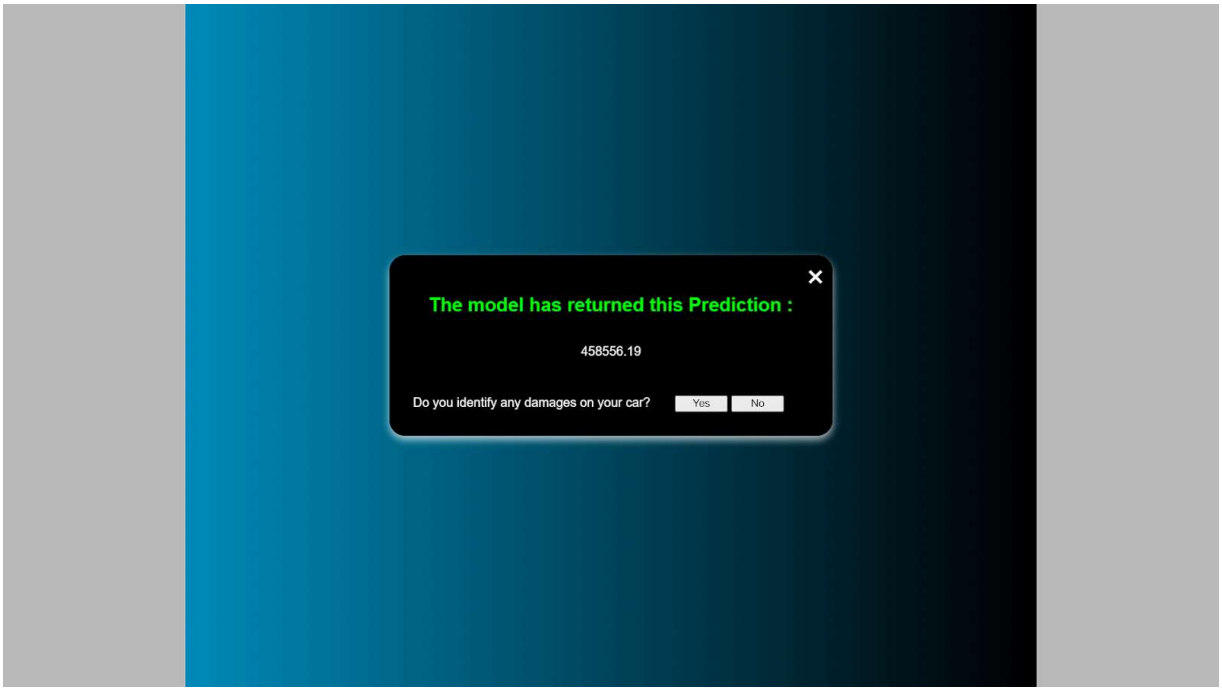

Fig 6.1.2: Display Base_price

➔ This screen represents the modal window in which the user is shown the calculated price from the machine learning model.

➔ It also prompts the user to select whether they have any damages on their car.

6.1.3 Select option: No

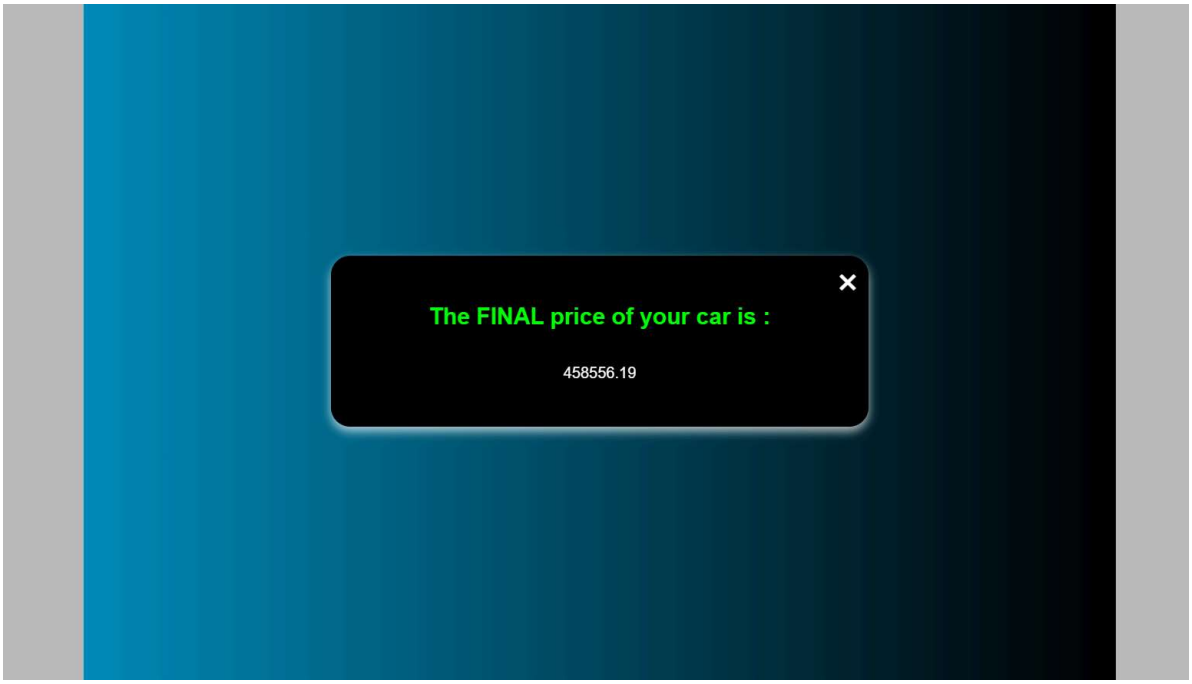

Fig 6.1.3: Select option: No

➔ This screen represents the modal window in which the user is shown the FINAL price of their car. This is shown only when the user clicks on the ‘No’ option from the previous screen.

6.1.4 Image upload form

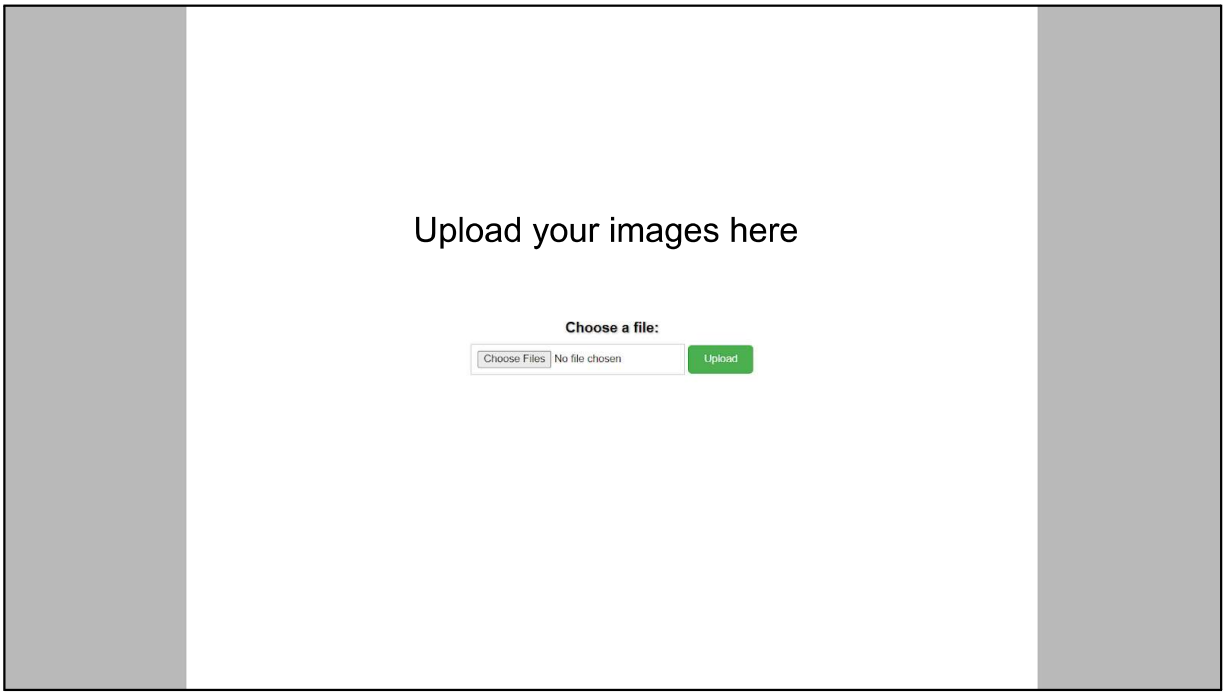

Fig 6.1.4: Image Upload screen

➔ This window will pop up if the user has selected the 'Yes' option in the display base_price screen. It will indicate that the user has damage on their car, and have opted to further assess any damages.

➔ The system takes the uploaded image by the user to the backend application, where it will be preprocessed before passing it into the Damage Detection function.

6.1.5 Damage Assessment | review

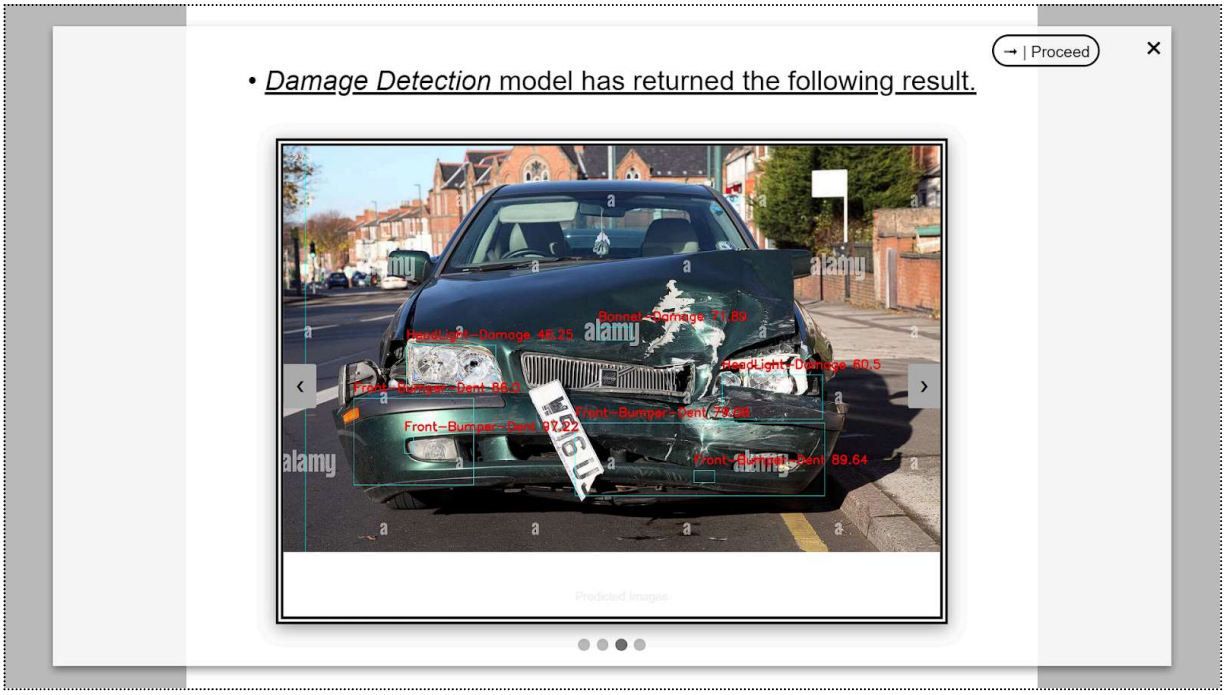

Fig 6.1.5: Damage Assessment & review

➔ Demonstration of an image, after all the damages were successfully detected.

➔ User can just confirm and go through all the images that they entered using the carousel, as shown.

➔ Finally, they can select the proceed button which will adjust the original base_price based on the severity of damages detected.

6.1.6 Display: Final Adjusted price

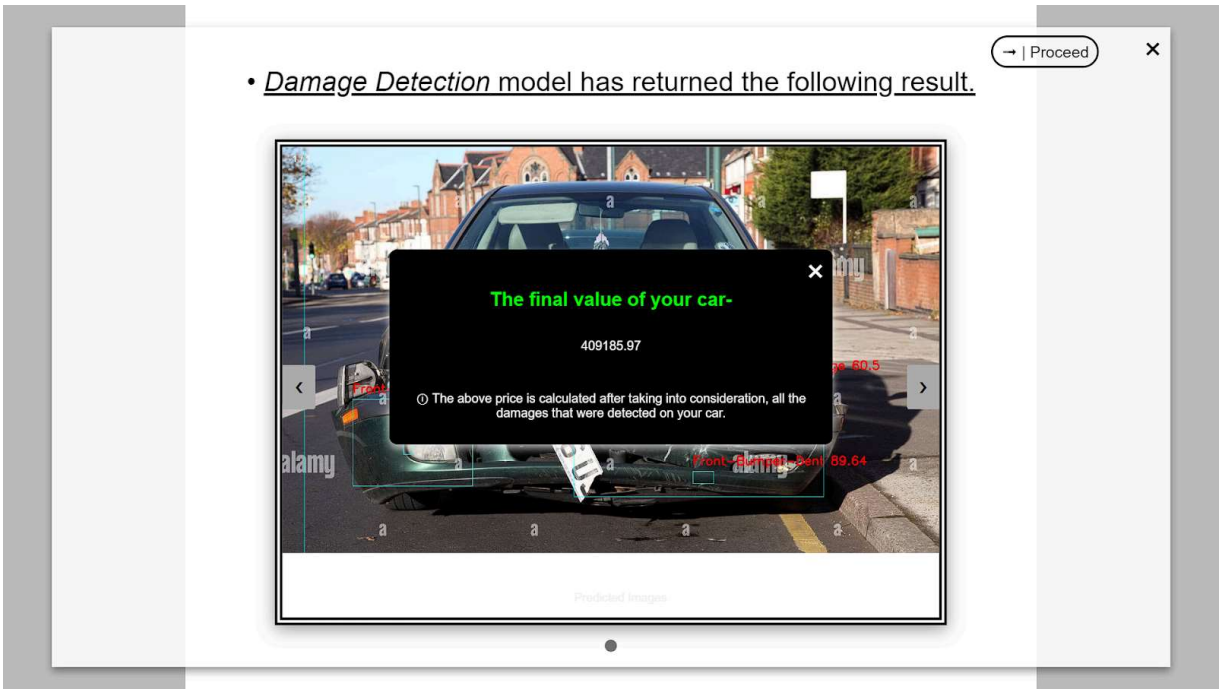

Fig 6.1.6: Final Adjusted Price display

➔ The final adjustment is made, and displayed in this window.

CHAPTER 7

TESTING

- **Black Box Testing**
- **White Box Testing**
- **Test Cases**

7.1 Black Box Testing

Black-box testing is a method of software testing that instead of examining its inner code, it focuses on whether the system delivers the expected results when given specific inputs. This approach works for various testing stages, from checking individual components to evaluating the entire system. It's sometimes called specification-based testing because the focus is on whether the software meets its intended functionality. During black-box testing, the tester uses the interface just like a regular user would, providing inputs and analyzing outputs without delving into the code behind the scenes.

Techniques for Black-Box Testing:

- **Equivalence Partitioning:** This technique involves grouping input values into categories (valid and invalid), and then selecting a few representative values from each category to test. The goal is to cover all possible scenarios without excessive testing.
- **Boundary Value Analysis:** This method concentrates on testing the edges of acceptable input ranges. For example, if a field accepts values 18-65, you'd test with values like 17, 18, 65, and 66 to see how the system handles these boundary conditions.

Advantages: -

- Since testers don't see the internal code, they approach the software from a true user's perspective. This can uncover bugs that developers might overlook due to familiarity with the system.
- Black-box testing can start early in the development cycle, even before detailed coding is complete, helping identify issues in requirements or design.
- Testers who may not have strong programming skills can still design and execute effective black-box tests, focusing on functional behavior.

Disadvantages: -

- Black-box testing might reveal a problem, but pinpointing the exact source of the bug within the code can be challenging.
- Without analyzing the code, it's difficult to ensure that all possible execution paths and error conditions are tested.

7.2 White Box Testing

White-box testing, also known as clear-box testing or glass-box testing, delves into the inner workings of the software. Testers with programming expertise meticulously examine the source code to identify potential problems. This method allows for highly focused tests that target specific code structures, logic branches, and error handling mechanisms. Since white-box testing leverages a deep understanding of the code's design and implementation, it can be highly effective in uncovering intricate bugs and ensuring code efficiency.

Advantages: -

- White-box testing allows testers to meticulously examine every line of code and design test cases that directly target specific code segments. This comprehensive approach can reveal bugs that might be missed by black-box testing.
- By analyzing the code's logic flow and decision points, white-box testing ensures all conditional branches and error handling scenarios are exercised, leading to a more robust codebase.
- Testing is more thorough, with the possibility of covering most paths.

Disadvantages: -

- Testers need strong programming skills and an understanding of the specific programming language used to effectively perform white-box testing.
- As the code evolves, white-box tests might need to be updated to reflect the changes, potentially increasing maintenance overhead.
- Developing comprehensive white-box tests can be time-consuming, especially for complex software projects that include more logical modules.

7.3 Test Cases

Test cases are the specific scenarios that are used to test the system's functionality and behavior. In the context of our project, the following are the test cases that were used:

1. Test Cases for Car Feature form

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Test Result
1	Verify user can access and enter all the fields of the feature form.	1. Launch the website. 2. Enter test values in each field.	User should be able to enter desired values in the input fields.	Same as expected.	Pass
2	Verify that an error message shows up when user enters invalid input.	1. Launch the website. 2. Enter invalid test data.	User should see an error message.	Same as expected.	Pass
3	Verify that no input field can be left empty on submit.	1. Launch the website. 2. Leave input fields empty simultaneously and click Submit.	Display error message that shows that no field should be empty.	Same as expected.	Pass
4	Verify the functionality of the submit button.	1. Launch the website. 2. Enter valid test data. 3. Click submit.	Form successfully submitted.	Same as expected.	Pass

Table 7.3.1 Feature form

2. Test Cases for Displayed predicted price modal

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Test Result
1	Verify that the price is displayed correctly and the damage prompt is visible.	1. Launch the website. 2. Enter test values in each field. 3. Click submit. 4. Check the visibility of the price and prompt.	Both the price and the prompt question are visible	Same as expected.	Pass
2	Verify that the price is calculated correctly as per the features submitted.	1. Launch the website. 2. Enter test values in each field. 3. Click submit. 4. Verify the predicted price using validation data.	The predicted price should be same as validation price.	Same as expected.	Pass
3	Verify that the 'No' button displays final price on click.	1. Launch the website. 2. Enter test values in each field. 3. Click submit. 4. Then, click on 'No' option in the next screen.	The final price should be displayed.	Same as expected.	Pass
4	Verify that the 'Yes' button redirects user to the damage assessment module.	1. Launch the website. 2. Enter test values in each field. 3. Submit features. 4. Then, click on 'Yes' option.	User is now redirected to the damage assessment module.	Same as expected.	Pass
5	Check the working of the close button in current modal window.	1. Launch the website. 2. Enter test values in each field. 3. Submit features. 4. Click on the close (x) button	The modal should disappear, leaving the user back to the previous feature form.	Same as expected.	Pass

Table 7.3.2 Predicted Price Modal

3. Test Cases for the image upload form.

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Test Result
1	Verify that the image upload form allow selecting multiple images at a time.	1. Select more than one images in the upload form.	System should allow uploading more than one images.	Same as expected.	Pass
2	Verify that user should not be able to submit the form without selecting atleast 1 image.	1. Click on submit.	The user should see error indicating that null form can not be submitted.	Same as expected.	Pass
3	Check whether the form submits; with the input images.	1. Select test image(s) and click on submit.	The upload form should successfully submit the selected images and pass them to the Damage Detector module.	Same as expected.	Pass

Table 7.3.3 Image upload form

4. Test Cases for Damage Assesment module

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Test Result
1	Check whether each submitted car image have returned with the assessment report.	1. Click submit in the previous section. 2. Check the displayed images.	The output images should have the damages marked on them,	Same as expected.	Pass
2	Check the	1. Click submit in the	The keys should	Same as	Pass

	navigation keys for image carousel works.	previous section. 2. Click the navigation keys.	swap the display images with the next/previous one.	expected.	
3	Verify that the close modal button works.	1. Click submit in the previous section. 2. Click the close button.	The modal should close.	Same as expected.	Pass
4	Verify that the proceed button is functional, and that it displays the final calculated price.	1. Click submit in the previous section. 2. Click the on the proceed button.	The reduction function should return the final adjusted price, and display it successfully.	Same as expected.	Pass

Table 7.3.4 Damage Assessment

CHAPTER 8

SYSTEM DESIGN

- **Class Diagram**
- **Use – Case Diagram**
- **Sequence Diagram**
- **Activity Diagram**
- **State Chart Diagram**
- **Data Flow Diagram**
- **ER Diagram**

8.1 CLASS DIAGRAM

A class diagram is a type of structural diagram within the Unified Modeling Language (UML) used in software engineering. It visually depicts the classes that make up a system and the relationships between them. Classes are like templates for creating objects in object-oriented programming. Class diagrams model the static structure of a system, focusing on what elements exist rather than how they interact over time.

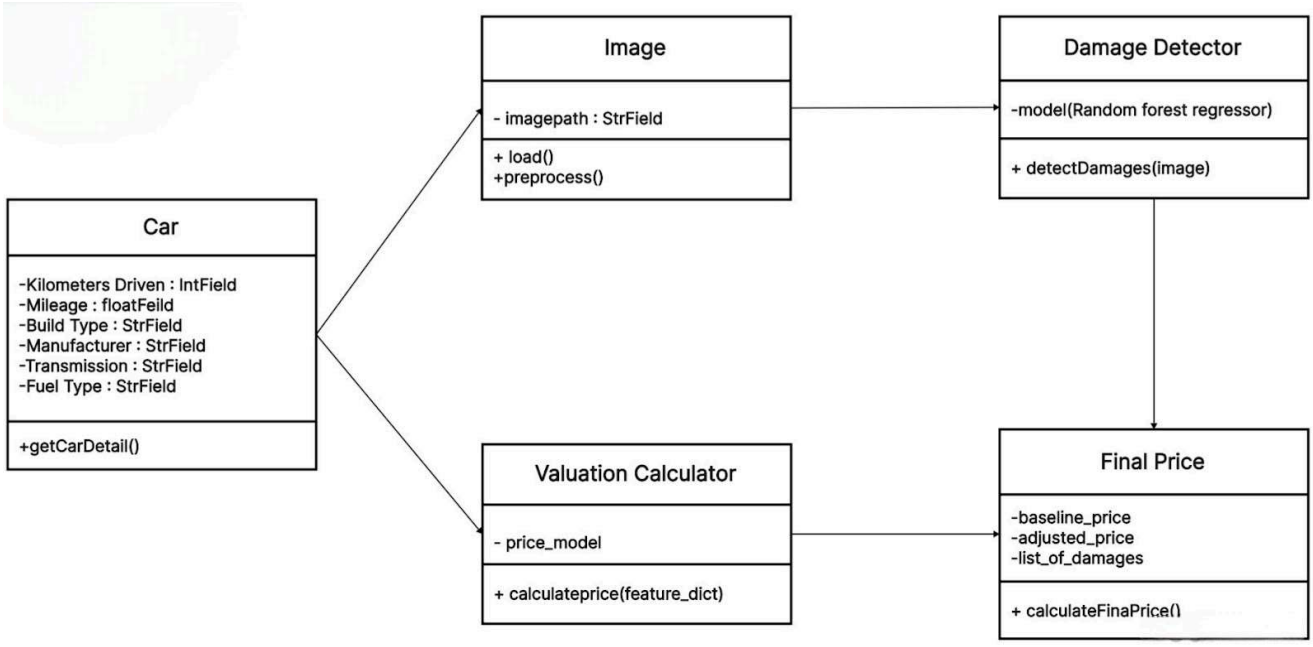

Fig 8.1.1: Class Diagram

8.2 USE – CASE DIAGRAM

A use case diagram visually represents how users interact with a system, showcasing the different use cases (actions or goals) within that system. It highlights user requirements and the expected outcomes from their perspective, without delving into the internal workings of the system.

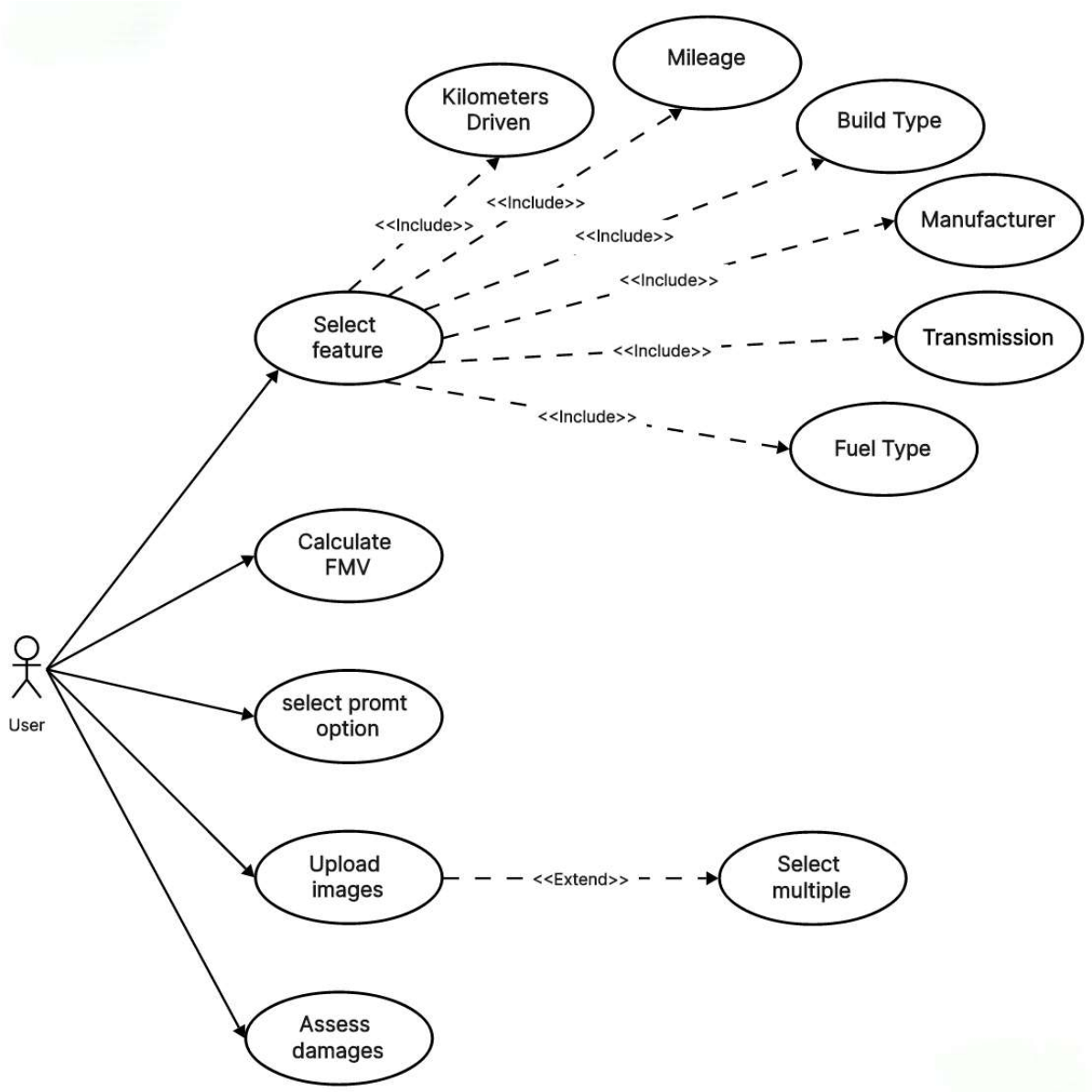

Fig 8.2.1: Use – Case Diagram of User

8.3 SEQUENCE DIAGRAM

Sequence Diagrams are interaction diagrams that detail how operations are carried out.

They capture the interaction between objects in the context of a collaboration.

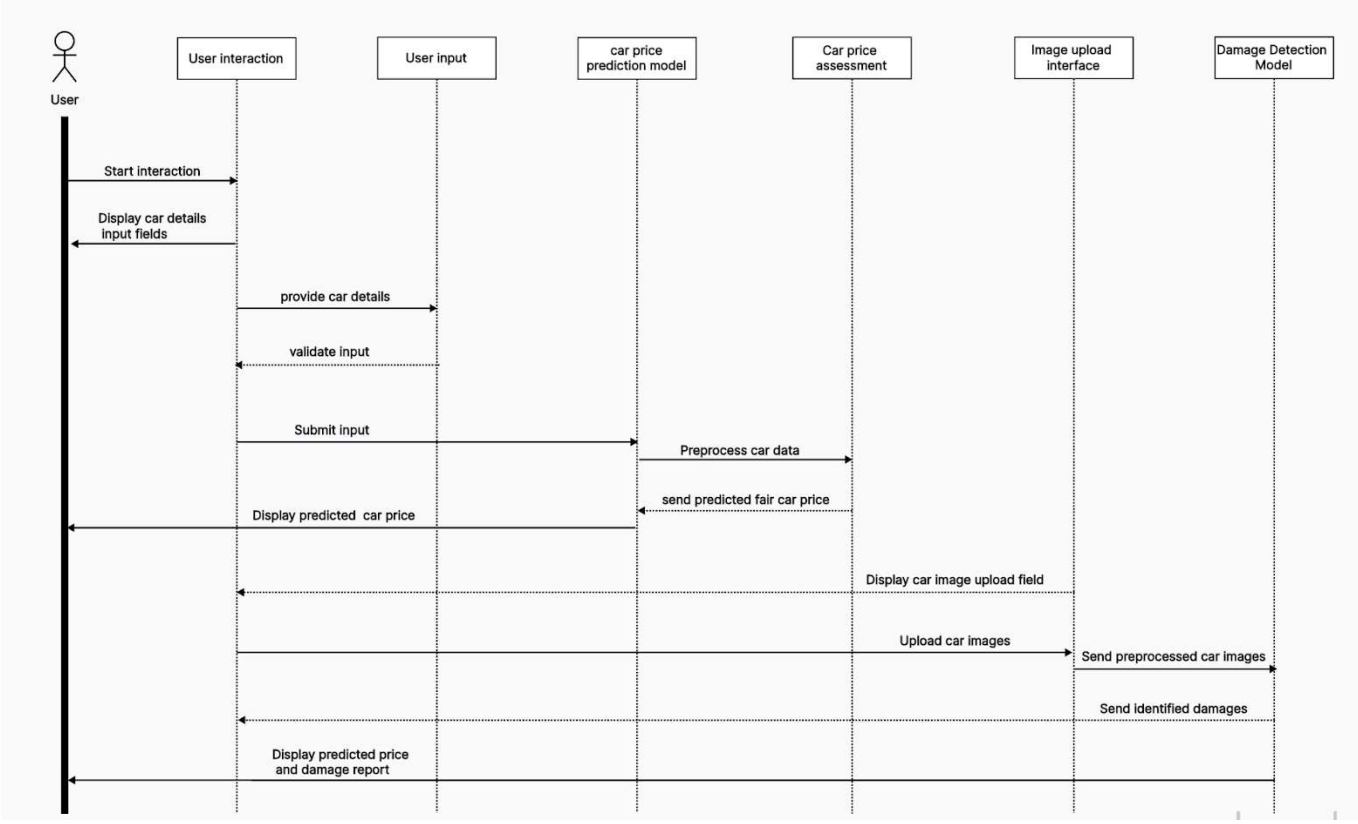

Fig 8.3.1: Sequence Diagram of User

8.4 ACTIVITY DIAGRAM

Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

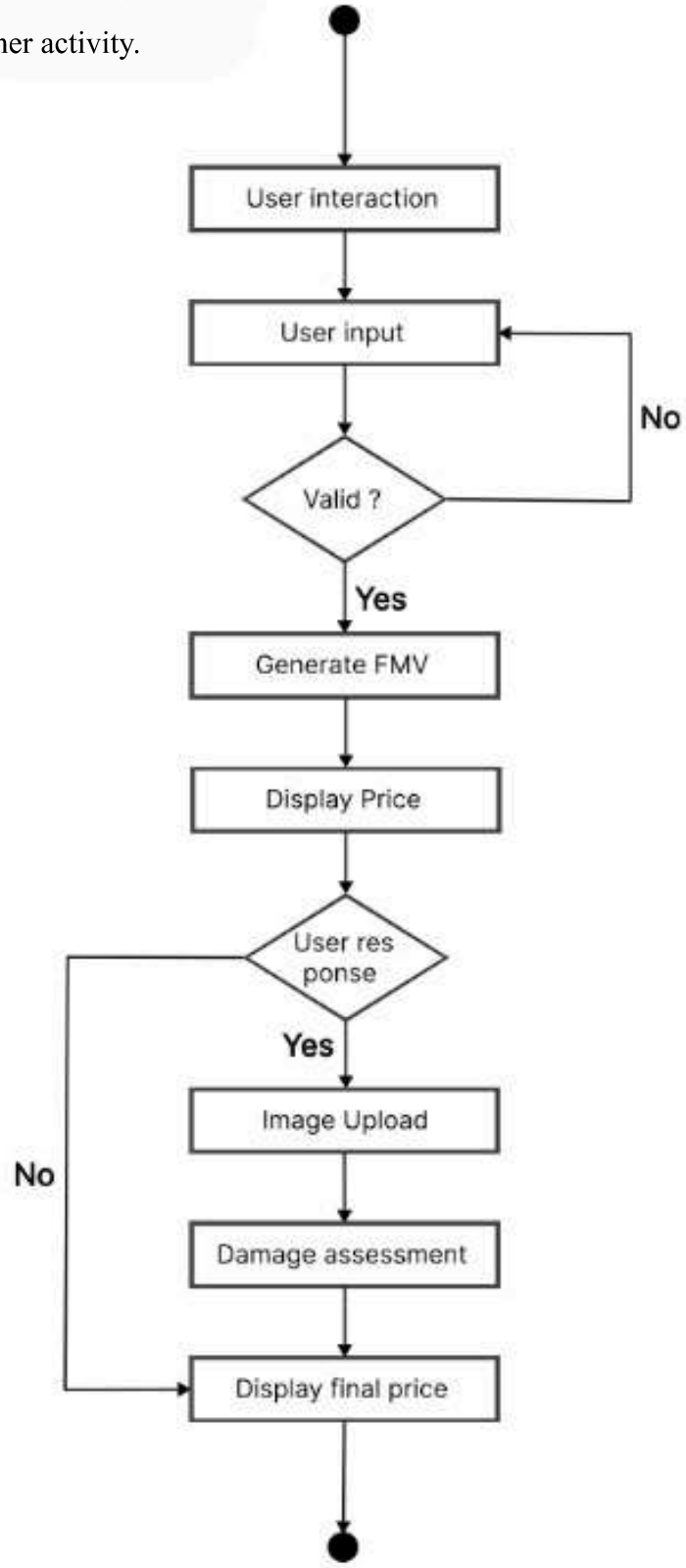

Fig 8.4.1: Activity Diagram of User

8.5 STATE CHART DIAGRAM

A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

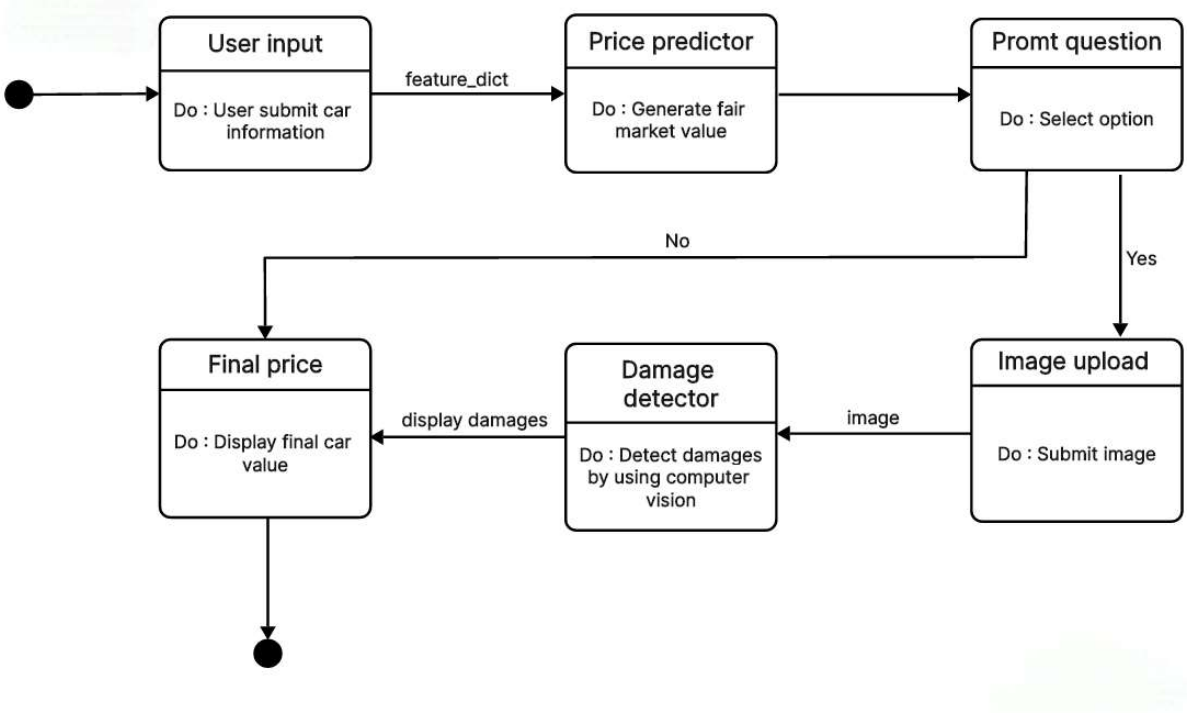

Fig 8.5.1: State Chart Diagram

8.6 DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system.

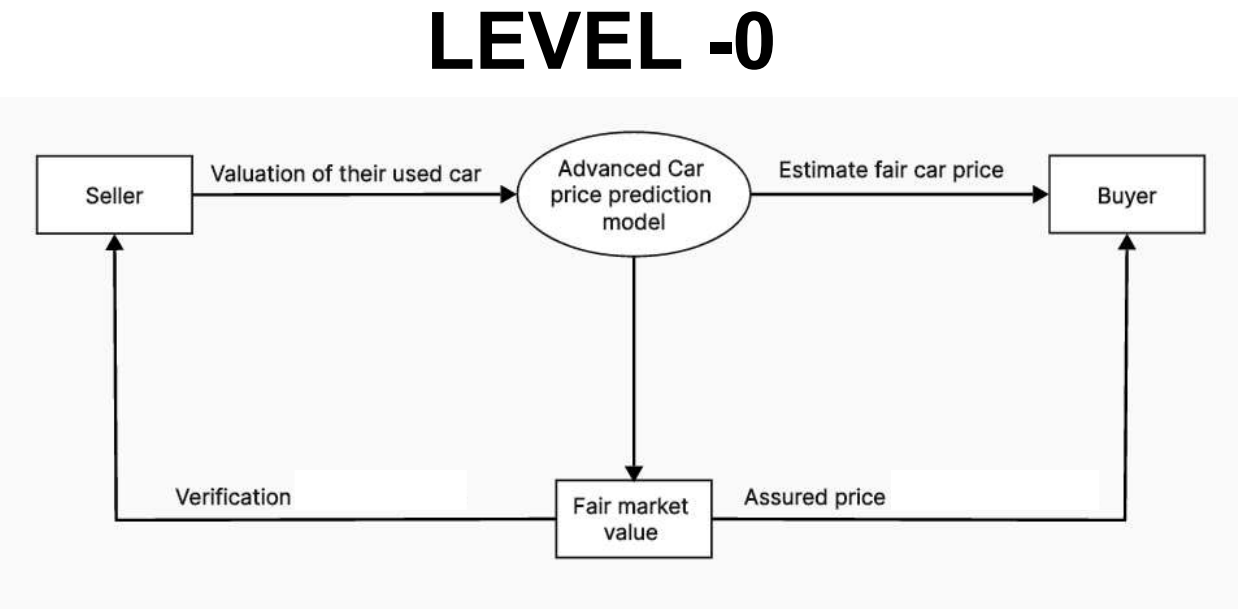

Fig 8.6.1: Data Flow Diagram LEVEL-0

LEVEL -1

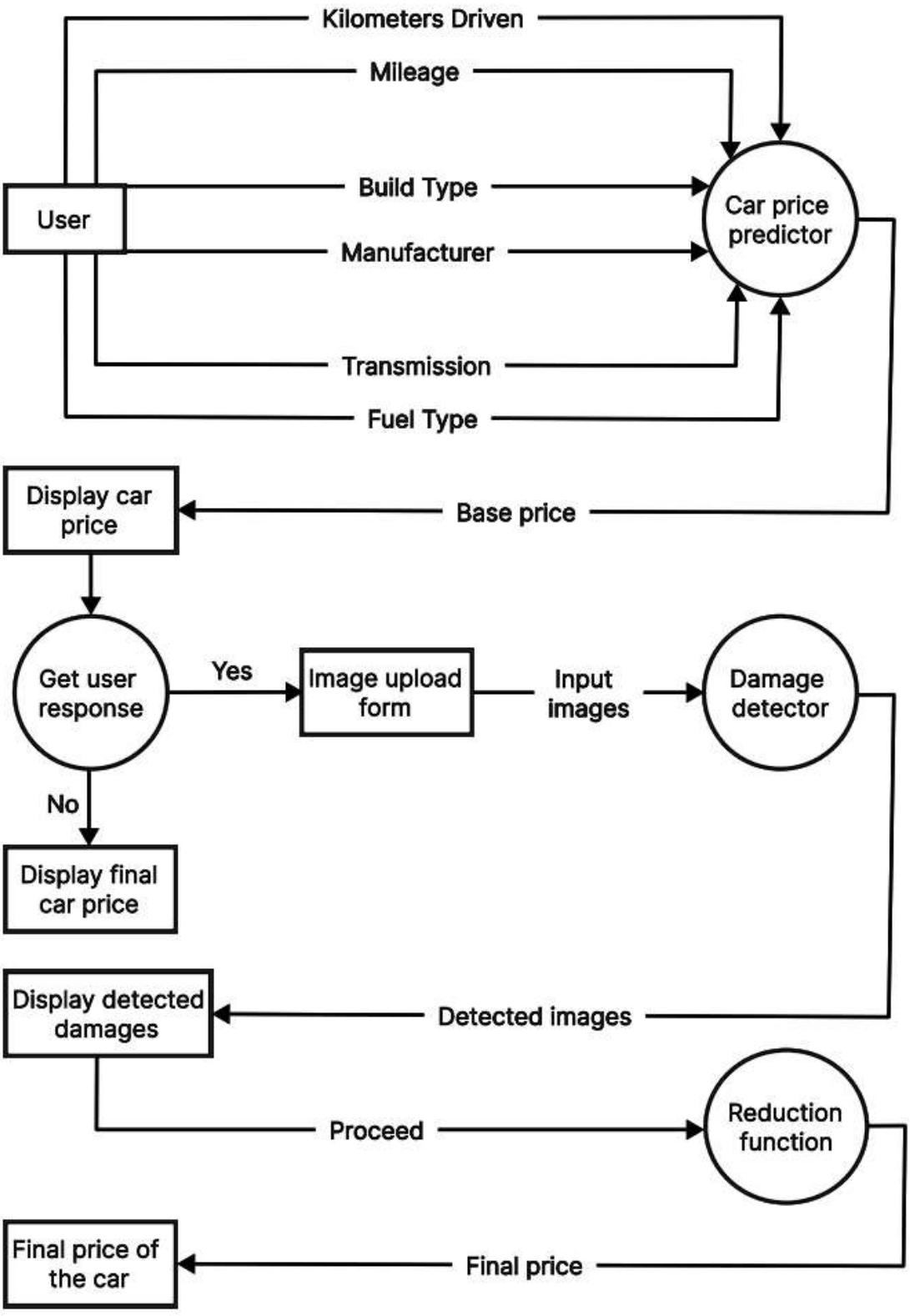

Fig 8.6.2: Data Flow Diagram LEVEL-1 for User

8.7 Entity Relationship Diagram

This model is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database and also it develops a very simple and easy to design view of data.

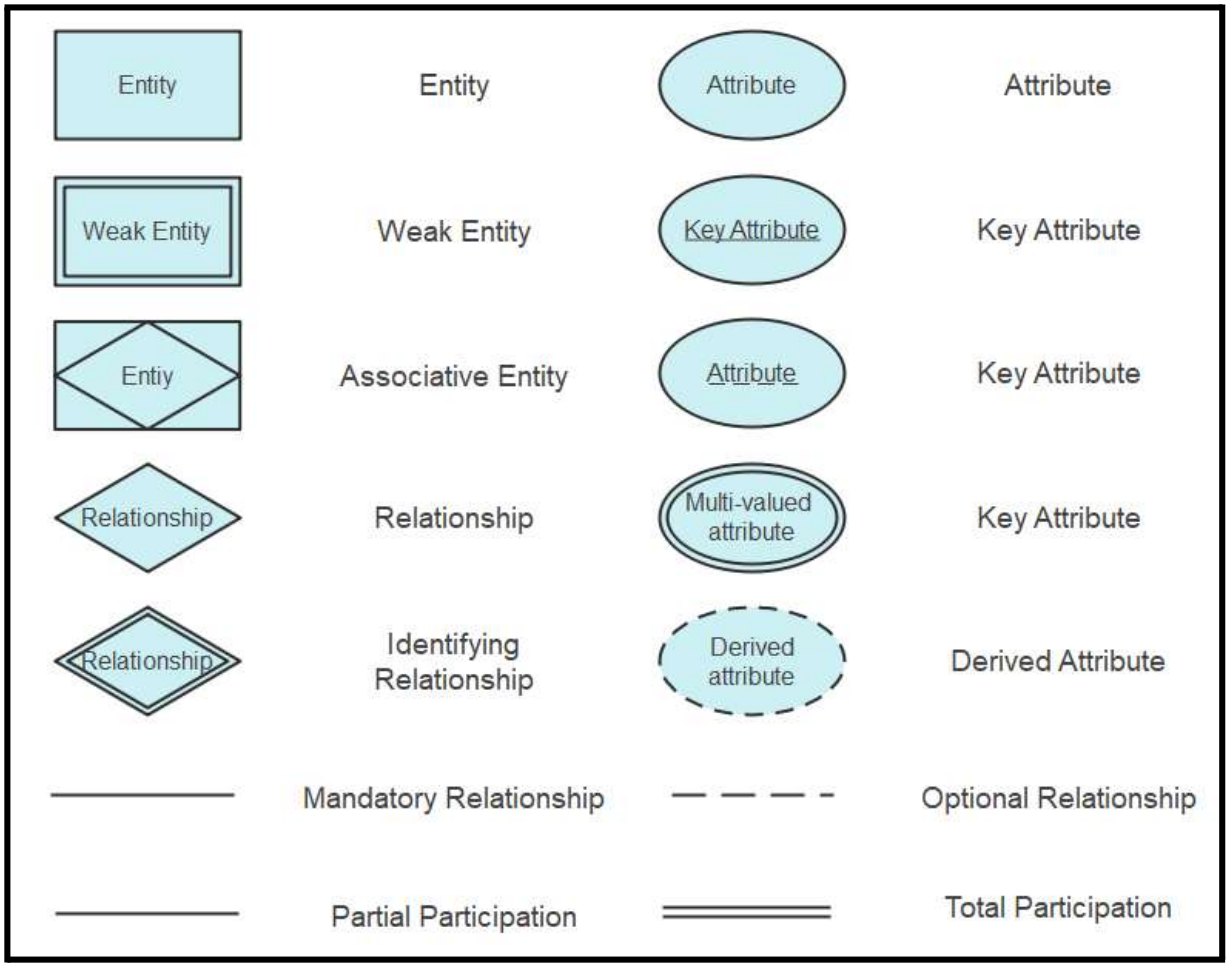

Fig: 8.7.1: ER Diagram Symbols

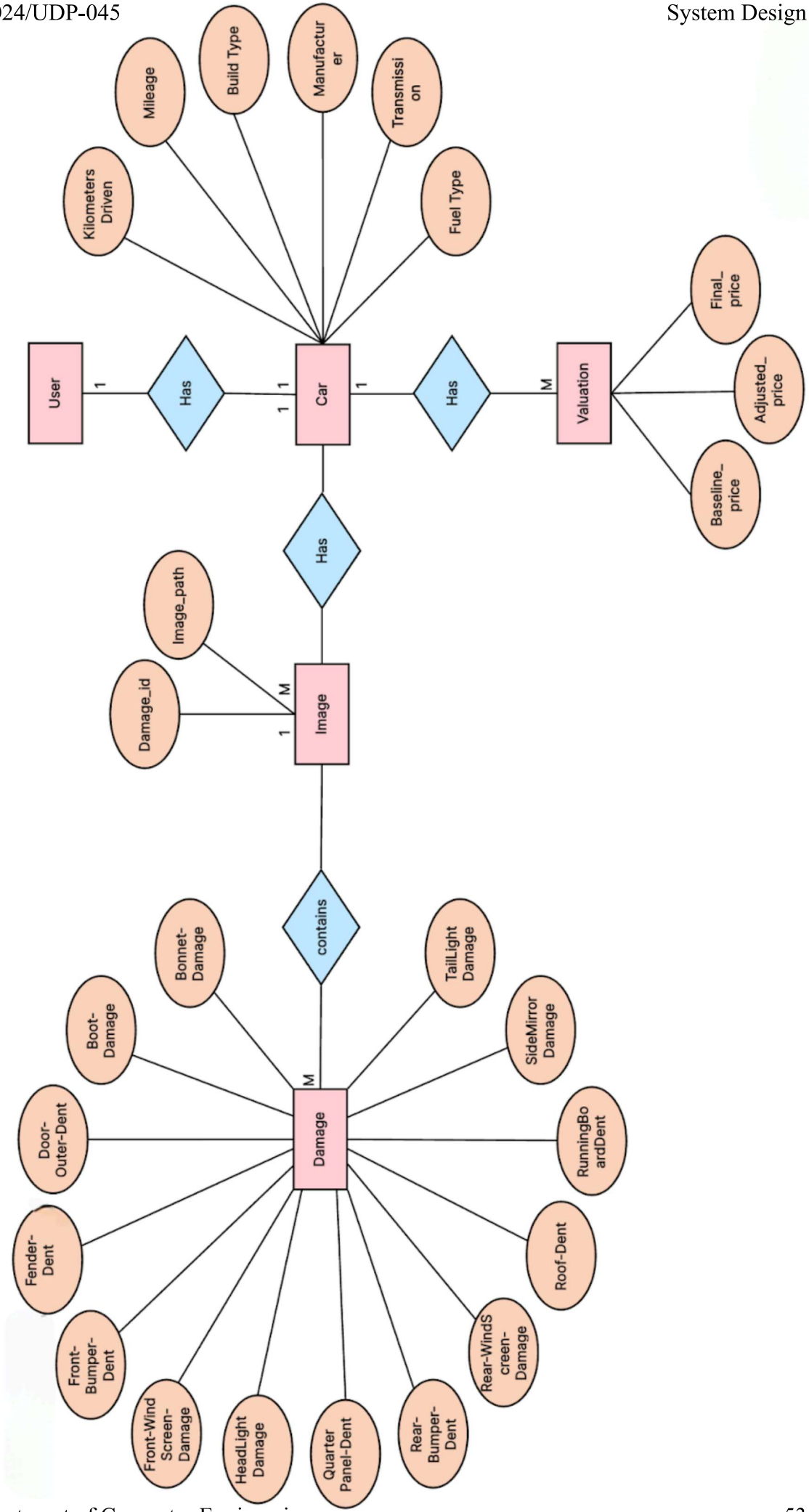

Fig: 8.7.2: ER Diagram

CHAPTER 9

LIMITATION AND FUTURE SCOPE

- **Limitation**
- **Future Enhancement**

9.1 LIMITATION

The current implementation operates under certain constraints that can be addressed in future iterations. Firstly, resource limitations during development led us to train our damage detection model using YOLOv4. While functional, recent advancements suggest that newer YOLO versions (v7, v8) might offer significantly improved accuracy and performance. Upgrading the object detection model should be a priority for subsequent versions to enhance the reliability of damage assessments. Additionally, the datasets used to train both the car price prediction model and the damage detection model might limit their generalizability. Expanding these datasets with a greater diversity of car images, damage types, environmental conditions, and reliable sources of used car valuations would likely increase the model's robustness in real-world scenarios. Researching and incorporating more reliable valuation data sources would be crucial for improving price prediction accuracy.

Due to project scope limitations, user registration and login functionalities were not implemented. This restricts users from saving past valuations or resuming their work from previous sessions. Future iterations should prioritize user account features to enhance the user experience and allow users to track historical valuations, compare past estimates, and potentially monitor market trends over time.

List of limitations in the *Advanced Car Price Prediction Model* are:

- Hardware Limitations: Our damage detection model, trained using YOLOv4 object detection, was hindered by insufficient hardware resources. Based on our research, we believe that more advanced state-of-the-art models, such as YOLOv7 and beyond, could significantly improve performance, accuracy, and precision in car damage detection, leading to increased reliability.
- Missing Valuation History: In its current phase, our website lacks user signup functionality. As a result, users cannot access and view past valuations generated for their cars.

9.2 FUTURE ENHANCEMENT

In a nutshell, it can be said that the future scope of the project revolves around maintaining information regarding:

- Enhanced Model Performance: Upgrading the damage detection model to a more recent YOLO architecture (v7, v8) should be prioritized for improved accuracy and speed. This would significantly enhance the reliability of the system's valuations.
- Robust Datasets: Sourcing larger and more diverse datasets for both car price prediction and damage detection. Incorporating a wider range of car conditions, image backgrounds, and valuation sources will increase the models' adaptability to real-world scenarios.
- User Account System: Implementing registration and login features to allow users to save past valuations, track market trends over time, and enhance personalization.
- Data Analytics: Adding a module to provide users with insights into market trends, price fluctuations based on damage severity, and potentially regional variations in car valuations.
- Cloud Deployment: Migrating the system to a cloud platform for scalability, ensuring the ability to handle increasing user traffic and potential future expansions.

CHAPTER 10

CONCLUSION

- **Conclusion**

10.1 CONCLUSION

This project has resulted in a user-friendly system designed to empower both buyers and sellers in the used car market. Through careful planning and iterative development, we have established a foundation that can be expanded and refined for even greater value. The system comprises both the front-end interface and an administrative area. The interface guides users intuitively through the valuation process, while the administrative side may allow for future functionalities such as the management of reference datasets or the analysis of system usage patterns. The primary focus was on providing car valuations, incorporating damage detection and price adjustment capabilities. We meticulously followed these key development stages:

- Clear Goals: From the outset, the project defined aims, objectives, and scope, ensuring a focused approach throughout development.
- Problem Definition: We carefully analyzed the challenges faced by used car sellers and buyers, identifying the need for objective valuation and transparent damage assessment.
- System Modeling: Through rigorous design, we created a model for the system, outlining core functionalities, user interactions, and the integration of machine learning components.
- User-Centric Interface: The design of a clear, intuitive user interface was paramount to ensure a positive user experience.
- System Implementation and Testing: The project followed established software development practices, with thorough testing to ensure the system's reliability and accuracy.

10.1.1 Future Vision

- This project serves as a strong foundation for further innovation within the used car valuation domain. It provides a framework upon which we can build more advanced analytics tools, potentially incorporating market trend analysis and valuation forecasting capabilities, further strengthening its value proposition in a dynamic marketplace.

BIBLIOGRAPHY

REFERENCES

- List of Literature Survey References:

[1] Jihad Qaddour, and Syeda Ayesha Siddiqa. “Automatic damaged vehicle estimator using enhanced deep learning algorithm.” 2023, https://www.sciencedirect.com/science/article/pii/S2667305323000170

[2] Mallios, Dimitrios & Xiaofei, Li & McLaughlin, Niall & Martinez-del-Rincon, Jesus & Galbraith, Clare & Garland, Rory. (2023). Vehicle Damage Severity Estimation for Insurance Operations Using In-The-Wild Mobile Images. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3299223.

[3] Jihad Qaddour, Syeda Ayesha Siddiqa, Automatic damaged vehicle estimator using enhanced deep learning algorithm, Intelligent Systems with Applications, Volume 18, 2023, 200192, ISSN 2667-3053, https://doi.org/10.1016/j.iswa.2023.200192. (https://www.sciencedirect.com/science/article/pii/S2667305323000170)

[4] Kyu, Phyu & Woraratpanya, Kuntpong. (2020). Car Damage Detection and Classification. 1-6. 10.1145/3406601.3406651.

[5] Rakshata P, Padma H V, Pooja M, Yashaswini H V ,Karthik V. “Car Damage Detection and Analysis Using Deep Learning Algorithm For Automotive.” vol. Volume 5, no. issue 6, Nov-Dec-2019.

- List of Web References

[1] https://www.python.org/

[2] https://github.com/pjreddie/darknet

[3] https://www.tensorflow.org/guide/keras