**CNN MODEL INTEGRATED WITH MOBILE TECHNOLOGY FOR VEHICLE PARTS IDENTIFICATION (PARTWISE).**

**FRANCIS WAMBUGU IRAKI.**

**SCT221-0404/2021**

*A proposal/research project submitted to the Department of Information Technology in the School of Computing and Information Technology in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Information Technology at Jomo Kenyatta University of Agriculture and Technology.*

September 2024 – April 2025

# **DECLARATION**

I \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ hereby declare that this proposal/research project is my original work and has not been presented for a degree in any other University.

…………………. …………………

Signature Date

I \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ can confirm that this proposal/research project has been submitted for examination with my approval as the university supervisor.

……………… ……………….

Signature Date

# **ABSTRACT**

This proposal outlines the development of PARTWISE, an innovative solution designed to assist the car mechanics industry with accurate spare part identification. PARTWISE addresses a critical challenge: the need for a reliable, efficient tool to identify spare parts, which often relies on time-consuming, manual processes prone to error. This solution leverages recent advancements in deep learning technology, specifically Convolutional Neural Networks (CNN), to automate image-based spare part identification.

At the core of PARTWISE is an integrated CNN model trained on a diverse dataset of spare part images. This model is intended to identify spare parts from images submitted by users, enhancing accuracy while reducing the need for manual intervention. The PARTWISE mobile application provides a user-friendly interface for mechanics to capture or upload images of spare parts, which are then processed by the CNN model to deliver precise identification results.

By focusing on AI-driven accuracy and user-centric design, PARTWISE aims to transform how spare parts are identified within the car mechanics industry, leading to significant efficiency improvements and time savings for mechanics and vehicle owners.

Table of Contents

[**DECLARATION** I](#_Toc196723337)

[**ABSTRACT** II](#_Toc196723338)

[**LIST OF TABLES** IX](#_Toc196723339)

[**LIST OF FIGURES** X](#_Toc196723340)

[**ACRONYMS** XI](#_Toc196723341)

[**DEFINITATION OF TERMS** XII](#_Toc196723342)

[**CHAPTER 1** 1](#_Toc196723343)

[**INTRODUCTION** 1](#_Toc196723344)

[**1.3** **PROBLEM STATEMENT** 2](#_Toc196723345)

[**1.4** **PROPOSED SOLUTION** 3](#_Toc196723346)

[**1.5** **OBJECTIVES** 4](#_Toc196723347)

[1.5.1 GENERAL OBJECTIVES 4](#_Toc196723348)

[1.5.2 SPECIFIC OBJECTIVES 4](#_Toc196723349)

[**1.6** **RESEARCH QUESTIONS** 4](#_Toc196723350)

[**1.7** **JUSTIFICATION** 5](#_Toc196723351)

[**1.8** **PROPOSED RESEARCH AND SYSTEM METHEDOLOGIES** 6](#_Toc196723352)

[**1.9** **SCOPE** 9](#_Toc196723353)

[**CHAPTER 2** 10](#_Toc196723354)

[**LITERATURE REVIEW** 10](#_Toc196723355)

[**2.1 INTRODUCTION** 10](#_Toc196723356)

[**2.2 THEORETICAL REVIEW** 10](#_Toc196723357)

[**2.3 CASE STUDY REVIEW** 13](#_Toc196723358)

[2.3.1 Tractable 13](#_Toc196723359)

[2.3.2 YOLOv5-Based AR Maintenance Assistant 14](#_Toc196723360)

[2.3.3 PDDNet for Pantograph Maintenance 15](#_Toc196723361)

[2.3.4 MobiTyres & Autoservice 15](#_Toc196723362)

[**2.4 INTEGRATION AND ARCHITECTUER** 16](#_Toc196723363)

[**2.5 SUMMARY** 19](#_Toc196723364)

[**2.6 RESEARCH GAPS** 20](#_Toc196723365)

[**CHAPTER 3:** 21](#_Toc196723366)

[SYSTEM ANALYSIS AND DESIGN 21](#_Toc196723367)

[3.1 Introduction 21](#_Toc196723368)

[3.2 Systems Development Methodology 21](#_Toc196723369)

[3.3 Feasibility Study 27](#_Toc196723370)

[3.4 Requirements Elicitation 39](#_Toc196723374)

[3.5 Data Analysis 40](#_Toc196723375)

[3.6 System Specification 43](#_Toc196723376)

[3.7 Requirements Analysis and Modeling 44](#_Toc196723377)

[3.8 Logical Design 49](#_Toc196723378)

[3.9 Physical Design 51](#_Toc196723379)

[**Chapter 4: System Implementation and Testing** 53](#_Toc196723380)

[**4.1 Introduction** 53](#_Toc196723381)

[**4.2 Environment and Tools** 53](#_Toc196723382)

[**4.3 System Code Generation** 55](#_Toc196723383)

[**4.4 Testing** 56](#_Toc196723384)

[**4.5 User Guide** 59](#_Toc196723385)

[**4.6 Conclusions** 59](#_Toc196723386)

[**4.7 Recommendations** 60](#_Toc196723387)

[**REFERENCES** 61](#_Toc196723388)

[**APPENDICES** 62](#_Toc196723389)

[**Instruments** 62](#_Toc196723390)

# **LIST OF TABLES**

[Table 1.1 Project Schedule 73](#_Toc196721996)

[Table 1.2 Proposed Budget 73](#_Toc196721997)

# **LIST OF FIGURES**

[Figure 1 CRISP-DM 8](#_Toc196722028)

[Figure 2 Crisp DM 22](#_Toc196722029)

[Figure 3 CNN output 27](#_Toc196722030)

[Figure 4 Deployment 27](#_Toc196722031)

[Figure 5 Revenue Growth Line graph 30](#_Toc196722032)

[Figure 6 Initial Investment Pie chart 31](#_Toc196722033)

[Figure 7 Feasibility Study 34](#_Toc196722034)

[Figure 8 Feasibility study 35](#_Toc196722035)

[Figure 9 Feasibility study MobiTyres 36](#_Toc196722036)

[Figure 10 Feasibility Study MobiTyres 37](#_Toc196722037)

[Figure 11 Engaging with Mechanics tools for Diagnostic 38](#_Toc196722038)

[Figure 12Engaging with MobiTyres Mechanics 39](#_Toc196722039)

[Figure 13 Data Analysis 42](#_Toc196722040)

[Figure 14Pie chart Misidentification Frequency 43](#_Toc196722041)

[Figure 15 Bar Graph for delays and costs 43](#_Toc196722042)

[Figure 16 Activity Diagram 46](#_Toc196722043)

[Figure 17 Data Flow Diagrams 47](#_Toc196722044)

[Figure 18 Use Case Diagrams: 47](#_Toc196722045)

[Figure 19 Database Schema 48](#_Toc196722046)

[Figure 20 Component diagram 49](#_Toc196722047)

[Figure 21 System Architecture 51](#_Toc196722048)

[Figure 22 Physical design 53](#_Toc196722049)

[Figure 23 Training and validation accuracy 58](#_Toc196722050)

[Figure 24 Epoch 58](#_Toc196722051)

[Figure 25 Accuracy Testing 59](#_Toc196722052)

[Figure 26 OBD2 Scanner Diagnostic Tool 63](#_Toc196722053)

[Figure 27 Engaging with the Branch Manager 69](#_Toc196722054)

[Figure 28 Engaging with The Mechanics 70](#_Toc196722055)

[Figure 29 Going through the car diagnostic process 71](#_Toc196722056)

[Figure 30 Gantt Chart 74](#_Toc196722057)

# **ACRONYMS**

1. CNN – Convolutional Neural Network
2. AWS – Amazon Web Services
3. AI – Artificial Intelligence
4. YOLO – You Only Look Once (an object detection algorithm)
5. VGG – Visual Geometry Group (related to a type of neural network)

# **DEFINITATION OF TERMS**

1. PARTWISE – The name of the system being developed for car mechanics to easily identify vehicle parts.
2. CNN (Convolutional Neural Network) – A type of artificial neural network used for image recognition tasks.
3. AI (Artificial Intelligence) – The simulation of human intelligence processes by machines.
4. AWS (Amazon Web Services) – A cloud platform offering computing power and storage solutions.
5. YOLO (You Only Look Once) – An algorithm for real-time object detection.
6. TensorFlow – An open-source platform for machine learning.
7. PyTorch – A popular deep learning framework for building and training neural networks.
8. Django – A high-level Python web framework for rapid development.
9. Flask – A micro web framework written in Python, used for web development.
10. SaturnCloud – A platform for data science and machine learning in the cloud.
11. Agile Methodology – A project management approach that emphasizes flexibility, collaboration, and iterative progress through small, incremental changes.

# **CHAPTER 1**

# **INTRODUCTION**

* 1. **BACKGROUND OF THE STUDY**

The automotive repair industry is vital to the global economy, ensuring that vehicles remain operational and reliable. However, a critical bottleneck in this sector is the process of identifying and sourcing spare parts, especially in regions like Kenya, where the supply chain is fragmented, and access to technology is limited. Traditional identification methods, such as manual inspection, reliance on human expertise, and trial-and-error, often result in inefficiencies, delays, and errors. These challenges are further exacerbated when dealing with rare or imported vehicle models, where identifying the correct part can take days, if not weeks.

The Kenyan automotive industry, characterized by a growing vehicle population and the increasing complexity of modern cars, requires innovative solutions to streamline operations. While global trends in artificial intelligence (AI) and machine learning have shown promise in addressing such inefficiencies, there is a noticeable gap in the localized application of these technologies. The integration of AI-driven systems, particularly Convolutional Neural Networks (CNNs), into spare part identification processes could revolutionize how mechanics and repair shops operate by enhancing accuracy, reducing downtime, and lowering costs.

This study aims to bridge this gap by exploring how CNN technology can be adapted to meet the unique challenges faced in Kenya. By leveraging the power of deep learning and combining it with mobile technology, the proposed solution seeks to modernize spare part identification, enabling mechanics to deliver faster, more reliable, and cost-effective services to vehicle owners.

* 1. **PROJECT OVERVIEW**

The **PARTWISE** project focuses on developing an AI-powered solution tailored to the automotive repair industry. At its core, PARTWISE is a mobile application integrated with a Convolutional Neural Network (CNN) model, specifically designed to identify spare parts from images captured or uploaded by users. The project addresses the pressing need for a reliable and efficient tool that simplifies the identification process, replacing manual methods with a technology-driven approach.

PARTWISE operates in three main stages: image capture or upload, processing through the CNN model, and result generation. The CNN model, trained on a diverse dataset of spare part images, ensures accurate identification by analyzing visual patterns and features unique to each part. The mobile application, designed with user experience in mind, provides mechanics with a seamless interface to interact with the system, facilitating instant identification and detailed information about the spare part.

Beyond identification, the application connects users to trusted suppliers, reducing the time and effort required to source components. The proposed system leverages cloud infrastructure for scalability and reliability, ensuring consistent performance even with a growing user base. By addressing challenges like fragmented supply chains, manual errors, and delays in procurement, PARTWISE aims to transform the Kenyan automotive repair industry, setting a benchmark for future innovations in the field.

## **PROBLEM STATEMENT**

The PARTWISE project aims to address the significant problem of inefficient spare part identification in the Kenyan transport industry, as well as the broader automotive repair sector. Mechanics in these industries currently face challenges in accurately identifying the correct spare parts for vehicle repairs, leading to inefficiencies and delays in the repair process. The lack of a streamlined process often results in time-consuming manual searches for parts, reliance on outdated or incomplete information, and increased operational costs.

In addition to these challenges, the absence of a centralized system for spare part identification exacerbates the issue. Mechanics are left to rely on personal networks and limited access to reliable sources, which hinders their ability to find the right parts. This results in increased vehicle downtime, delays in repairs, and the potential for using substandard or incompatible parts, all of which can compromise safety and performance.

To address this problem, the PARTWISE project leverages advanced computing technologies, specifically Convolutional Neural Networks (CNNs), to streamline and improve the spare part identification process. By utilizing a CNN-based model, the system can accurately recognize spare parts from images or uploaded files, eliminating guesswork and minimizing the risk of errors or mismatched components.

The proposed system will be integrated into a user-friendly mobile application, allowing mechanics to quickly capture or upload images of spare parts and receive accurate identification results. This solution will significantly reduce the time spent on manual identification and enhance the accuracy of part selection. By providing a reliable and efficient identification process, the mobile application will improve the operational efficiency of mechanics, leading to faster repairs, reduced vehicle downtime, and improved customer satisfaction.

Moreover, the mobile application will offer better access to a wider range of suppliers, ensuring that mechanics can source high-quality spare parts from trusted sources. This will help mechanics find parts more efficiently, with fewer barriers to access, contributing to overall operational efficiency in the transport sector.

By addressing these challenges, PARTWISE will provide a transformative solution for the automotive repair industry. The integration of CNNs into spare part identification will lead to faster, safer, and more accurate repairs, ultimately benefiting mechanics, vehicle owners, and the wider transport sector.

## **PROPOSED SOLUTION**

The PARTWISE project aims to improve spare part identification within the Kenyan transport industry by leveraging Convolutional Neural Networks (CNNs) integrated into a user-friendly mobile application. Unlike traditional systems, which typically encompass a broad array of functions, PARTWISE will focus specifically on the identification of spare parts for automotive repairs, addressing this critical task with advanced machine learning technology.

The mobile application will enable mechanics to easily capture images or upload pictures of spare parts. These images will then be processed by the CNN model, which has been trained on a comprehensive dataset to recognize and identify specific spare parts based on learned patterns. Upon successful recognition, the application will display detailed information about the identified part, such as its name, specifications, and compatibility.

Solution Direction and Area of Research:

Our solution is grounded in the field of machine learning, specifically focusing on computer vision and image recognition. We will employ a state-of-the-art CNN model in our system to achieve accurate and efficient spare part identification. By utilizing deep learning algorithms, the system will ensure high accuracy in identifying spare parts from images.

This research will evaluate existing CNN models, considering both global and regional approaches, and test their applicability in the local context of the Kenyan automotive repair industry. The PARTWISE system will thus provide an efficient, scalable, and locally relevant solution to spare part identification.

Key Operations of the System:

Image Capture and Upload: Mechanics will be able to capture images of spare parts using their mobile device or upload pictures directly through the app. This simple and intuitive process ensures accessibility for all users.

Image Processing and Recognition: Once an image is uploaded, it will be processed by the CNN model to correctly identify the spare part. The CNN's layers (input, convolutional, pooling, and fully connected) will extract features and classify the part based on patterns it has learned from its training data.

Information Display: After identification, the application will display detailed information about the part, including specifications, compatibility with various vehicles, and other relevant details to assist the mechanic in making informed decisions.

## **OBJECTIVES**

### GENERAL OBJECTIVES

To develop a mobile application integrated with Convolutional Neural Network (CNN) technology to improve the accuracy and efficiency of spare part identification for car mechanics.

### SPECIFIC OBJECTIVES

1. To design a robust CNN model that can accurately identify spare parts from images, reducing the time and effort required for manual identification.
2. To implement a user-friendly mobile application interface that will be integrated with CNN model that allows mechanics to easily capture or upload images of spare parts and retrieve detailed information about the identified parts.
3. To implement a Real time database for user management such as Firebase.

## **RESEARCH QUESTIONS**

1. How can CNN technology be leveraged to design a model that accurately identifies spare parts and improves the speed and efficiency of the identification process?
2. How can a user-friendly mobile application interface be designed and integrated with a Convolutional Neural Network (CNN) model to enable mechanics to efficiently capture or upload images of spare parts and retrieve accurate and detailed information about the identified parts?
3. How can Firebase be integrated into a mobile application to manage user data in real-time, ensuring secure and efficient user authentication and data synchronization?

## **JUSTIFICATION**

The direct beneficiaries of this research include car mechanics, vehicle owners, and spare part suppliers. Mechanics will benefit from the time saved in identifying spare parts, allowing them to focus more on actual repairs rather than spending time searching for components. This will improve their overall efficiency and productivity. Vehicle owners will experience quicker repair times, which translates to faster vehicle return, reduced downtime, and fewer errors in part replacement, leading to better service outcomes and enhanced satisfaction. Spare part suppliers will benefit from an improved sales process, as the app will streamline the identification of parts, reducing mistakes and speeding up the decision-making process.

The proposed solution directly addresses a key issue in the Kenyan automotive industry by removing the guesswork involved in spare part identification. By leveraging advanced technologies such as Convolutional Neural Networks (CNNs) and integrating them into a user-friendly mobile application, PARTWISE aims to enhance the accuracy and efficiency of the spare part identification process. The system will allow mechanics to quickly capture images of spare parts and receive instant, accurate identification, reducing the need for manual searches and reliance on outdated databases.

This solution fills a critical gap in the automotive repair industry, where there is currently no automatic system for spare part identification based on images. The integration of CNN technology in this context represents a significant leap forward in automating the process, aligning with global trends in artificial intelligence (AI) applications for practical, real-world problems. As industries around the world, including automotive repair, move toward digital solutions, the PARTWISE system reflects this shift, offering a modern, technology-driven approach to spare part identification.

Moreover, the proposed solution is particularly relevant to micro, small, and medium-sized enterprises (MSMEs) in Kenya, which are major consumers of automotive repair services. These businesses increasingly rely on digital tools to stay competitive, and PARTWISE will help them adopt AI-driven solutions without the complexity of traditional systems. By improving the spare part identification process, MSMEs in the automotive sector can enhance their operational efficiency, reduce costs, and provide better service to their customers. The PARTWISE system not only contributes to the growth of the local automotive repair industry but also supports the broader adoption of AI technologies in Kenya, fostering innovation and advancing digital transformation in key sectors.

### **PROPOSED RESEARCH AND SYSTEM METHEDOLOGIES**

The CRISP-DM methodology fits the data-driven project best. Therefore, it is appropriate for building a CNN-based image recognition system. This methodology includes the following six phases of a project:

1. Business Understanding:

Objective: To determine objectives and scope for the PARTWISE system.

Activities:

Problem identification: Inability to identify spare parts with high accuracy within minimum time.

Define objectives: To develop a mobile application based on CNN for automatic identification of spare parts with high accuracy.

Identify the criteria for project success: accuracy of CNN model, ease of use by the mechanics, and identification speed. Deliverables will include a well-defined problem statement and a project plan.

1. Data Understanding:

The goal is to collect and analyze the data so that it is suitable for building the CNN model.

Activities:

Collect images of spare parts from various sources like manual upload, suppliers, and manufacturers.

Analyze quality and diversity of data covering different parts, angles, and conditions.

Visualize sample images to identify any patterns and possible problems, such as low-quality or ambiguous images. Deliverable: A well-documented training and validation dataset.

1. . Data Preparation Objective: To prepare the data by cleaning and pre-processing for the training of the CNN model. Activities: The image data needs to be resized into a fixed dimension, for instance, 224x224 pixel dimensions. This will standardize all images to a consistent size. Normalize all pixel values to stimulate faster convergence when training. Increased dataset variability can be accomplished by applying some data augmentation techniques: rotation, flipping, zooming, etc. Split the data into three sets: one for training, one for testing (or validation), and one for testing after models are selected. Deliverable: Cleaned and enhanced dataset with good structure.
2. Modeling:

Objective: The goal is to train and implement the CNN for spare part identification. Activities: Select the most appropriate CNN architecture from ResNet or VGG or custom CNN depending on the requirement of the project. Training of model by means of TensorFlow or PyTorch. Hyperparameter tuning to improve performance would include learning rate and batch size. Model performance on an evaluation metric such as accuracy, precision, recall, and F1-score. Deliverable: Trained and Validated CNN Model ready to be deployed.

1. . Evaluation:

Objective: The model must satisfy the objectives of the project and perform according to real conditions. Activities: The model is tested on unseen data to check how well the generalization takes place. Results are measured against the success criteria defined in the Business Understanding phase. User testing with mechanics will be carried out to test accuracy and usability. Delivered: An inclusive evaluation report that puts up recommendations for improvement.

1. . Deployment Objective: The developed and trained CNN model would be deployed to work on a mobile application. Activities

Use a cloud platform for hosting the model, such as AWS or Google Cloud, since scalability is important here.

Design a mobile app interface to upload spare part images or capture them using a camera.

Deploy the application on platforms like the Google Play Store.

Performance testing and feedback from the users to update the system iteratively.

Deliverable: fully deployed and operational PARTWISE system.

Tools and Techniques in CRISP-DM for PARTWISE:

Deep Learning Frameworks: TensorFlow or PyTorch for the development of models.

Cloud Platforms: AWS or Google Cloud for model hosting.

Mobile Development: Flutter for cross-platform application development.

Data Visualization: Matplotlib or Seaborn for sense-making with data.

Data Augmentation: OpenCV for preprocessing and augmentation.

Justification for CRISP-DM:

CRISP-DM will be appropriate to be used in projects such as PARTWISE due to the fact that:

It heavily focuses on the preparation of the data and making sense out of it-in other words, understanding it, which is a key attribute to the performance of CNNs.

Its iterative modeling and evaluation phases provide one with constant refinement of the model.

This fits with real-world business objectives, and the system would be both technically robust and user-centered. Lifecycle Coverage CRISP-DM covers the whole life cycle of the project:

Business Understanding makes sure that the project aligns with stakeholder needs.

Data Understanding and Preparation ensure that data is of high quality and relevant.

Modeling and Evaluation ensure that the system meets performance goals.

Deployment makes the solution accessible and usable in real-life conditions.



Figure 1 CRISP-D

### **SCOPE**

PARTWISE aims to research and develop an AI-based system specifically designed for spare part identification in the automotive repair industry. The primary focus is on utilizing Convolutional Neural Networks (CNNs) to accurately identify spare parts from images, assisting mechanics and vehicle owners in Kenya. The scope of this study is limited to Kenya, with potential for future expansion to other regions as the solution proves successful.

The primary users of the PARTWISE system include:

Micro, small, and medium-sized enterprises (MSMEs) in the car repair industry, who will benefit from improved spare part identification accuracy and efficiency.

Spare part suppliers, who will experience a streamlined sales process through the system’s accurate part identification.

Individual vehicle owners, who will benefit from faster and more reliable repairs, reducing downtime and errors in spare part replacement.

While the system will provide a powerful tool for spare part identification, PARTWISE will not cover advanced features such as predictive maintenance or inventory management at this stage. These functionalities are outside the current scope but may be considered in future iterations of the system.

Key limitations of the study include:

Data Availability: A large, diverse dataset of spare part images is required to train the CNN model effectively. Obtaining such a dataset is challenging and may limit the initial performance of the model.

Computational Power: Training and deploying deep learning models require significant computational resources, which may impact the development speed and scalability of the system.

The project will focus on the design, development, and deployment of the mobile application through which mechanics can upload images of spare parts. The system will process these images using the CNN model to identify the spare parts and provide detailed information. Direct integration with external automotive systems, such as inventory management systems or repair scheduling platforms, will not be part of the current system but may be explored in future versions.

The PARTWISE system represents a significant step toward automating the spare part identification process, leveraging cutting-edge AI technology to improve operational efficiency in the Kenyan automotive repair sector. (Team, 2021)

# **CHAPTER 2**

# **LITERATURE REVIEW**

## **2.1 INTRODUCTION**

This chapter summarizes the current advancements in Convolutional Neural Networks (CNNs) and their application in image recognition, particularly within the automotive repair industry. The literature review focuses on PARTWISE, a system leveraging CNN technology to automate and enhance the identification of car spare parts. By removing manual processes prone to error, CNN technology can provide a fast and reliable means of identifying parts, transforming how mechanics approach part identification.

CNNs, a subset of artificial intelligence, are particularly suited for image-based recognition tasks due to their ability to learn complex image patterns through deep learning structures. This capability allows PARTWISE to recognize and classify diverse spare parts with high accuracy, minimizing the reliance on manual identification methods, which are often time-intensive and error-prone. The PARTWISE mobile application integrates CNN functionality to allow users to capture or upload images of spare parts directly, which the CNN model then processes for accurate identification and classification. (Krizhevsky A. S., 2012)

This literature review will explore research on CNNs within the context of image recognition, with a focus on practical applications across various fields, including automotive repair and healthcare, where similar technology is applied for tasks such as damage assessment and medical image analysis. The review will also cover CNN model structure and feature extraction, practical case studies in similar applications, and a comparison of deep learning frameworks like TensorFlow and PyTorch to select the optimal library for this project.

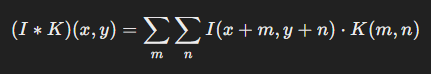
The objectives of this literature review are:  
 (1) to provide an overview of relevant CNN technologies in image recognition  
(2) to review CNN applications in sectors similar to automotive repair  
(3) to identify any research gaps PARTWISE aims to address through its spare part identification system.

## **2.2 THEORETICAL REVIEW**

This chapter delves into the theoretical foundations crucial to the PARTWISE system, focusing on image recognition and Convolutional Neural Networks (CNNs). CNNs are specialized deep learning models that have revolutionized image-based tasks due to their ability to learn hierarchical feature representations directly from input images. These features, captured through organized patterns in large datasets, enable CNNs to perform highly accurate object recognition, forming the backbone of the PARTWISE system’s functionality.

**Principles of CNNs and Mathematical Structure**

A CNN is designed to process grid-like data (such as an image) by learning spatial hierarchies of features through filters that capture patterns across the image matrix. Mathematically, CNNs are built upon convolutional layers, pooling layers, and fully connected layers. At a fundamental level, convolutional layers apply mathematical convolution operations on the image data, where a filter (or kernel) slides over the image matrix to produce feature maps. The convolution operation for an input image III and filter KKK is expressed as:

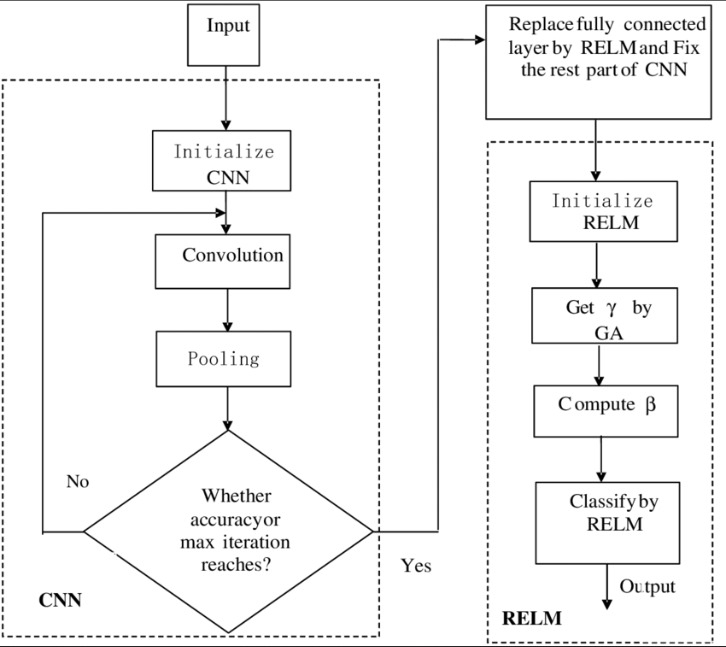


Here, I(x,y)I(x, y)I(x,y) is the input pixel value, K(m,n)K(m, n)K(m,n) is the kernel value, and the summation aggregates over each pixel to create feature maps that highlight specific patterns in the image, such as edges, textures, or shapes. (Andrew & Simonyan, 2014)

**CNN Architecture and Layer Functions**

The CNN architecture is generally composed of the following layers:

1. Input Layer: This layer receives raw image data, with pixel values represented in a matrix format. The input layer resizes and normalizes images, converting them into arrays to ensure that all images fed into the CNN model have uniform dimensions and pixel values that range between 0 and 1. This preprocessing step is essential for enhancing the model’s convergence and training efficiency.
2. Convolutional Layer: The convolutional layer is the core building block of a CNN and performs the convolution operation, which involves applying a series of filters (kernels) across the input image to detect specific features. Each filter is designed to capture distinct aspects of the image, such as edges, corners, or textures. This layer’s output is a set of feature maps that encode the spatial relationships of features across the input. As data flows through successive convolutional layers, the CNN progressively learns complex patterns, enabling accurate recognition even when parts of an object are partially occluded.
3. Activation Function (ReLU): After each convolution operation, a Rectified Linear Unit (ReLU) function is applied to add non-linearity to the model, which is necessary for learning complex patterns. The ReLU function, defined as f(x)=max⁡(0,x)f(x) = \max(0, x)f(x)=max(0,x), transforms negative values to zero, allowing only positive values to pass through. This enhances the model’s learning ability and accelerates convergence during training.
4. Pooling Layer: Pooling, or down-sampling, reduces the spatial dimensions of feature maps, preserving essential features while reducing computational load. The most common pooling technique is max pooling, which selects the maximum value from each region of the feature map, retaining only the most prominent features. Pooling increases the model’s robustness to minor image transformations, such as shifts and rotations, as it reduces the sensitivity of spatial positioning within the image.
5. Fully Connected (Dense) Layer: After passing through several convolutional and pooling layers, the output is flattened and fed into one or more fully connected layers. These layers integrate features extracted by previous layers to make final classifications or predictions. Mathematically, a dense layer computes a weighted sum of inputs and applies an activation function, usually softmax, to produce output probabilities. This enables the CNN to output class probabilities corresponding to different types of spare parts, facilitating accurate identification. (Andrew & Simonyan, 2014)



**Feature Extraction and Object Recognition**

CNNs excel in feature extraction by learning hierarchical representations, where early layers capture low-level features (e.g., edges), and deeper layers capture more abstract patterns, such as parts of an object or whole objects. In PARTWISE, this hierarchical feature extraction is critical, as it allows the CNN to identify diverse spare parts with high accuracy. Each layer in the CNN abstracts data to more complex levels, enabling it to generalize and recognize spare parts across various angles, lighting conditions, and occlusions, making CNNs highly adaptable for real-world applications. (Yosinski, 2014)

**Practical Applications of CNNs**

CNNs have found widespread application in fields requiring image-based recognition. In healthcare, CNNs are used in medical imaging to detect anomalies in X-rays, MRIs, and CT scans, where high accuracy is essential. Similarly, in the automotive industry, CNNs are utilized for vehicle part recognition and damage assessment. For instance, Tractable, an AI-based platform, uses CNNs to evaluate car part damages from images, enabling insurers to estimate repair costs accurately and quickly. In the PARTWISE project, CNNs facilitate accurate identification of automotive spare parts, addressing a significant challenge in the industry. (Geirhos, 2019)

Advantages and Limitations:

CNNs offer significant advantages over traditional image recognition methods, including high accuracy, robustness to variations in image data, and the ability to learn complex patterns. However, CNNs also require large amounts of labeled training data and are computationally intensive, which can be resource-demanding. These factors may impact model training speed and necessitate the use of high-performance hardware.

## **2.3 CASE STUDY REVIEW**

### 2.3.1 Tractable

Case Overview:

Tractable is an AI-powered platform that uses Convolutional Neural Networks (CNNs) to assess vehicle damage based on uploaded images. It automates insurance claims by providing rapid and accurate damage estimates, streamlining claims processing and minimizing manual intervention (Krizhevsky A. S., 2012)

#### 2.3.1.1 Theoretical Underpinning:

Efficiency: CNN-based automation significantly reduced manual evaluations, enhancing speed and accuracy (Krizhevsky A. S., 2012)

Scalability: Cloud-hosted CNN models allow Tractable to process large volumes of claims while maintaining performance (He, 2016)

#### 2.3.1.2 Key System Functionalities:

Damage Localization: Identifies specific areas of damage from vehicle images.   
Automated Report Generation: Creates repair cost estimates based on AI analysis.Cloud Integration: Utilizes cloud platforms for scalable and efficient data processing.  
(Goodfellow, 2016)

Tractable demonstrates the value of CNNs in automating complex image-based tasks. For PARTWISE, this underscores the importance of leveraging scalable cloud infrastructure and prioritizing efficiency.

### 2.3.2 YOLOv5-Based AR Maintenance Assistant

Case Overview:

The YOLOv5 AR Maintenance Assistant combines real-time object recognition with augmented reality (AR) to assist automotive professionals in identifying parts and understanding their functions. It highlights the capability of CNNs for rapid object detection in real-world contexts (Shorten, 2019)

#### 2.3.2.1 Theoretical Underpinning:

Real-Time Detection: YOLOv5's fast recognition engine supports instantaneous identification of automotive components.

Interactivity: AR integration enhances learning through interactive 3D models (Yosinski, 2014)

#### 2.3.2.2 Key System Functionalities:

3D Visualization: Provides detailed augmented models for user exploration.

Real-Time Recognition: Uses YOLOv5 to detect tools and components in real time.

Though PARTWISE does not use AR, the case emphasizes the need for a user-friendly interface and real-time performance to enhance user experience.

### 2.3.3 PDDNet for Pantograph Maintenance

Case Overview:

PDDNet, a deep learning system in the railway industry, monitors pantograph slide plates for wear and tear. The system leverages CNNs for real-time surface defect detection, allowing preventive maintenance and reducing operational downtime (He, 2016)

#### 2.3.3.1 Theoretical Underpinning:

Automation: Eliminates manual inspections, saving time and resources.

Preventive Maintenance: Detects wear early, preventing failures and costly repairs.

2.3.3.2 Key System Functionalities:

Surface Analysis: Analyzes slide plates for abnormalities in real time.

Automated Alerts: Notifies teams when maintenance is required.

PDDNet highlights the efficiency of automated detection systems. PARTWISE can similarly optimize its image recognition pipeline to handle diverse spare parts efficiently.

### 2.3.4 MobiTyres & Autoservice

Case Overview:

MobiTyres & Autoservice, a Kenyan automotive service provider, employs an AI-driven system for tire model and spare part identification. The system integrates computer vision to address local challenges such as fragmented supply chains and diverse vehicle models (Team, 2021)

#### 2.3.4.1 Theoretical Underpinning:

Local Adaptation: Tackles the unique automotive challenges of the Kenyan market.

Cost Efficiency: Automates identification, reducing reliance on manual approaches and cutting costs.

#### 2.3.4.2 Key System Functionalities:

Tire Model Recognition: Identifies tire specifications (e.g., size, brand, and compatibility).

Spare Part Matching: Suggests compatible parts based on vehicle component images.

Supplier Integration: Connects directly to local suppliers for streamlined procurement.

## **2.4 INTEGRATION AND ARCHITECTUER**

PARTWISE incorporates image recognition through a Convolutional Neural Network model, enclosed in a mobile application for identifying spare parts. This section explores the various options available to implement the research problem using different design architectures and frameworks, while giving the integration process in detail. The system is built to be scalable, flexible, and of high performance, providing solutions to identified important challenges within the Kenyan automotive repair sector.

Architecture Design Options

The following are some of the architecture design options which could be considered in designing the integration of the CNN-powered image recognition system with any mobile application. The following are options for implementation of the system:

Monolithic Architecture:

That would be a design in which the mobile app, CNN model, and cloud-based infrastructure are all integrated into one. The mobile application would directly connect with the CNN model, residing on the cloud, which would process the images for identification of spare parts.

Pros: Simplicity and speed at which development can be carried out since all components are tightly coupled.

Constrains: In such systems, scaling specific parts of the system independently has limited flexibility; hence, integrating any feature in the future will be more cumbersome.

Architecture of Microservices:

It can be actually implemented in a microservices architecture where each of these parts comprised of a mobile app, a CNN model, and cloud infrastructure interact independently as different services. By this, the mobile app may act like an independent microservice, which interacts via a RESTful API endpoint on the cloud. The CNN model would run independently on another microservice.

Pros: It is scalable, since services can be individually scaled. It will also provide ease in the further addition of features, such as predictive maintenance or inventory optimization.

Constrain: It is more complex to setup and manage, and there is a need for good coordination among the services. Serverless Architecture:

The system would be deployed on a serverless cloud infrastructure, with the CNN model hosted as a serverless function, such as on AWS Lambda. The mobile application would interface with the model via API calls, and the serverless functions execute the images without dedicated servers.

Pros: Scalability at low cost because one only pays for what is utilized. Efficient processing of traffic spikes since it is not necessary to provision servers.

Constrain: Cold start issues-that is, delays in the invocation of a function if it is not in use-reduce response time in image processing.

**System Integration Workflow and Communication**:

Mobile Application:

Image Capture and Upload: The mobile application provides the interface to upload images of spare parts by mechanics or vehicle owners. The app will provide the best UI/UX so that it may be easy to use and immediately move on to capturing an image to identify a part.

Data Transmission: The images taken or uploaded by the users are sent to the cloud where the processing of the data will be done by the CNN model. Since communication over the internet might be insecure, the application can apply SSL/TLS encryption in an attempt to secure the data in transit.

CNN Model for Image Recognition:

Image Processing Flow: The CNN model is proposed to recognize spare parts from images using features related to shape, texture, and size. The architecture shall consist of multiple layers:

Input Layer: It takes in the image data.

Convolutional Layers: Key features from the image are extracted.

Pooling Layers: Reduce the dimensionality of the features, retaining the most important information.

Fully Connected Layers: The features get classified to identify the spare part.

Model Hosting: The CNN model can be deployed using cloud-based services like AWS, Google Cloud, or Microsoft Azure. It will definitely give scalability and reliability to the model. The cloud infrastructure enables the model to handle large data sets and process images with efficiency.

Cloud Infrastructure:

The system leverages a cloud-based infrastructure to provide the necessary storage, computational power, and scalability. All images are stored in Google Drive, which serves as the primary repository for spare parts images. Model training is conducted on Google Colab, directly linked with Google Drive to facilitate seamless data access and processing. Once the Convolutional Neural Network (CNN) model is trained, it is converted to TensorFlow Lite (TFLite) format for integration into the mobile application.

For user management, Firebase is employed to handle real-time authentication, user data synchronization, and secure access. This integration ensures efficient management of user profiles and real-time communication within the mobile application.

API Layer for Communication:

The mobile application now integrates the on-device TensorFlow Lite (TFLite) model for real-time image processing of spare parts. In addition, a RESTful API is implemented to manage supplementary data requests and to facilitate secure communication between the mobile app, Google Drive, and Firebase. Google Drive stores the images used during model training (conducted on Google Colab), while Firebase is dedicated solely to user management, including authentication and real-time data synchronization. This API serves as middleware that efficiently routes requests, processes responses, and ensures that both image-related data and user management functions are handled seamlessly.

Security Considerations:

The API leverages OAuth protocols to ensure robust user authentication and authorization. This approach safeguards sensitive information by restricting access to authorized users only, thereby enhancing overall system security.

Possible Design Frameworks:

TensorFlow and Keras:

This CNN model can be realized using the Keras API in TensorFlow. Due to its powerful tools, it is widely applied in industry to handle images based on deep learning. Keras can be useful for fast, easy experimentation and prototyping with CNNs.

Pros: Well-documented, robust support for image recognition tasks, scalable to large datasets.

Constrain: Optimization requires good knowledge of the deep learning concepts.

PyTorch:

PyTorch is a popular deep learning framework that can be used as an alternative for the implementation of the CNN model. It provides a flexible way of building and training models, making it quite fit for research-oriented projects.

Pros: Dynamic computation graph; very suitable for research and prototyping; strongly growing, active community.

Constrain: Less mature in production environments than TensorFlow, although this is changing very rapidly.

Key Considerations for Implementation

Data Privacy: Since the system will have to deal with sensitive images provided by users, data privacy is a grave concern. The implementation of strong encryption protocols, such as AES encryption for stored images, along with the use of secure cloud services, will go a long way toward mitigating the risks.

Scalability: The architecture is designed in such a way that it will be able to scale with the increase in the number of users who will use this service. Most cloud platforms have auto-scaling to provide the necessary additional resources in case of increased usage spikes for high performance levels.

Response Time: To be improved through the integration of caching mechanisms-for instance, Redis-and/or by optimizing the CNN model used for image classification.

Future Enhancements:

The system will currently provide spare part identification, but in future versions, the following added functionalities could be developed to enhance the system:

Predictive Maintenance: The system may predict when a failure could occur in a vehicle based on the spare parts identified and previous history and may recommend appropriate preventive measures.

Integration with Automotive Repair Databases: The system will, in the near future, be integrated with online repair databases for more detailed information about spare parts and their compatibility with various vehicle models.

## **2.5 SUMMARY**

This chapter highlights the growing use of Artificial Intelligence (AI), particularly Convolutional Neural Networks (CNNs), for image detection in the automotive repair industry. Key concepts such as image recognition and CNNs are highly relevant to the PARTWISE project, which focuses on applying CNN technology for spare part identification in the automotive sector.

The reviewed case studies, including Tractable, PDDNet, and the YOLOv5-based Augmented Reality Assistant, demonstrate how CNNs and image recognition technologies have enhanced operational efficiency in various industries. However, these case studies have limitations in scope—either focusing solely on damage assessment or specialized use cases—that PARTWISE seeks to address by developing a broader and more comprehensive solution for spare part identification.

The PARTWISE solution integrates a CNN-based image recognition system with a user-friendly mobile application. All training images are stored in Google Drive and processed on Google Colab, with the trained model converted to TensorFlow Lite (TFLite) for seamless integration into the mobile platform. The system utilizes a RESTful API layer to facilitate communication between the mobile application and the model, ensuring efficient routing of image data and identification results. Additionally, Firebase is implemented for robust user management, handling authentication, real-time data synchronization, and secure access. By focusing on these integrated components, the PARTWISE project aims to address practical challenges faced by automotive mechanics in Kenya, providing a rapid and efficient tool for spare part identification.

## **2.6 RESEARCH GAPS**

While much success has been achieved by AI technologies, such as CNNs, in automotive repair, there is a key performance gap that PARTWISE attempts to fill. Systems in place like Tractable focus only on external damage assessment and do not entail internal car component identification. This provides a huge gap in holistic spare part identification, since such systems are bound to assess only visible damage, leaving behind the intricate task of identifying varied internal components that need repair or replacement.

Most of the existing solutions are narrowly focused on some specific aspects of the process and do not exploit the complete potential of CNNs regarding object recognition in a wide variety of automotive parts. While image recognition is applied in some systems, they often lack the integration with a mobile app interface where the mechanics will be able to upload and receive identification results in a user-friendly manner, thus speeding up the repair process.

PARTWISE claims to fill these gaps by focusing on an end-to-end solution which, through CNN-based image recognition, automates the process of identifying spare parts. By saving the need for manual identification and integrating the technology into an interface in a mobile application, PARTWISE will help the mechanic identify parts more quickly to improve efficiency within the automotive repair industry. While most of the other solutions focus on narrow use cases, PARTWISE will have a more holistic approach to addressing the full spectrum of the needs in automotive repair.

# **CHAPTER 3:**

# SYSTEM ANALYSIS AND DESIGN

# 3.1 Introduction

For the following chapter, it provides a comprehensive analysis and design framework for **PARTWISE**, an AI-driven mobile application aimed at solving the challenge of incorrect spare part identification at Mobi Tyres Automotives, Ruiru Branch (Team, 2021). By leveraging a Convolutional Neural Network (CNN), the system automates the identification process, significantly reducing the time and extra costs associated with manually locating the right parts. Throughout this chapter, the systems development methodology is detailed, along with a thorough feasibility study—supported by on-site engagement at Mobi Tyres Automotives (Team, 2021)—and descriptions of requirements elicitation, data analysis, and modeling. Both logical and physical designs are then elaborated, referencing diagrams, flow charts, wireframes, and real-world photographic evidence of the feasibility study.

# 3.2 Systems Development Methodology

For the PARTWISE project, a structured, iterative approach was adopted using the CRISP-DM methodology (Cross-Industry Standard Process for Data Mining). This methodology is particularly suited to data-driven projects and ensures that every stage—from problem identification to final deployment—is methodically addressed.

Figure 2 Crisp DM

**Business Understanding:**

The critical business issue centers on the misidentification of automotive spare parts, which leads to increased labor, extended repair times, and extra transportation costs. During your on-site engagement at Mobi Tyres Automotives (Team, 2021) (as depicted in the photographs), mechanics confirmed that incorrect spare parts often result in direct financial losses and customer dissatisfaction. **PARTWISE** was conceptualized to solve these inefficiencies by integrating a CNN model within a user-friendly mobile application.

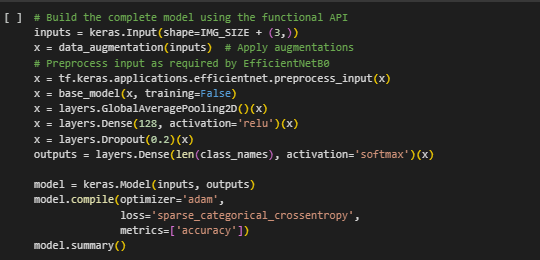
**Data Understanding and Preparation:**

A diverse dataset of spare part images was collected from kaggle data science community and manual captures, then stored in Google Drive. An exploratory data analysis (EDA) was conducted in Google Colab, allowing for the assessment of image quality, resolution, and diversity. During data preparation, images were resized to a uniform 224×224 pixels and normalized to scale pixel values between 0 and 1. Data augmentation techniques—including rotation, flipping, zooming, and shifting—were applied to simulate real-world variations and enrich the dataset. The data was then split into a 70/30 ratio, with 70% of images used for training and 30% reserved for validation. This split, implemented using TensorFlow’s *image\_dataset\_from\_directory* utility, was critical for ensuring that the model was evaluated on unseen data, thus mitigating the risk of overfitting. In addition, each spare part category was encoded into numerical labels to facilitate multi-class classification.

**Modeling and CNN Development:**

The core of the PARTWISE system is the CNN model, built using TensorFlow’s Keras Functional API and leveraging transfer learning with EfficientNetB0. The model architecture begins with an input layer configured for images of size 224×224 with three color channels. Data augmentation is applied immediately after input to increase dataset variability. The images are then preprocessed using EfficientNetB0’s dedicated preprocessing function, which scales and normalizes the input as required by the base model. EfficientNetB0 serves as a feature extractor, processing the images to generate high-dimensional representations that capture essential visual features such as edges, textures, and shapes. A Global Average Pooling layer is used to reduce these feature maps to a compact feature vector. This vector is then fed into a dense layer with 128 neurons and a ReLU activation function to learn complex feature combinations. A Dropout layer, with a dropout rate of 0.2, is applied to prevent overfitting. Finally, the output layer uses a softmax activation function to provide a probability distribution over the spare part classes.

Below is the key segment of the code illustrating the model construction :

****

**Input Layer**

The model begins with an input layer that accepts images with a predefined shape, determined by the parameter IMG\_SIZE, and three channels representing an RGB image. By enforcing a consistent input size, the system ensures that all images are uniformly processed, thereby reducing variability and facilitating more efficient learning. This consistency is critical in a production environment where images may come from various sources, and standardizing them helps the network to focus on learning the relevant features rather than compensating for differing dimensions.

**Data Augmentation Layer**

Immediately following the input, a data augmentation layer is applied. This layer performs random transformations—such as rotations, flips, zooms, and shifts—to the input images, effectively increasing the diversity of the training dataset. Data augmentation serves two primary purposes: it artificially expands the dataset, reducing the risk of overfitting, and it helps the model generalize better by exposing it to a wide range of variations. In the context of automotive spare part identification, where images can be taken under different conditions and angles, these augmentations ensure that the CNN remains robust and accurate in real-world scenarios.

**Preprocessing Layer**

After augmentation, the images are preprocessed using a function specifically tailored for EfficientNetB0. This step normalizes the images according to the pre-trained model’s requirements, scaling pixel values to a range that maximizes the model’s ability to extract features. Proper preprocessing aligns the input data with the conditions under which EfficientNetB0 was originally trained, thereby leveraging its powerful feature extraction capabilities. This step is crucial for effective transfer learning, ensuring that the pre-trained base model can seamlessly integrate into the PARTWISE system.

**Feature Extraction with EfficientNetB0**

The preprocessed images are then passed through EfficientNetB0, a sophisticated, pre-trained model that serves as the backbone of the CNN. EfficientNetB0 has been trained on extensive datasets, enabling it to capture a rich set of hierarchical features ranging from low-level edges and textures to high-level shapes and object parts. By setting the model to inference mode (training=False), we preserve these learned representations while preventing the weights from being updated during further training. This transfer learning approach significantly accelerates the training process and improves overall accuracy, as the model benefits from pre-existing knowledge without the need for extensive retraining from scratch.

**Global Average Pooling Layer**

Following feature extraction, a Global Average Pooling (GAP) layer is employed to reduce the dimensionality of the feature maps produced by EfficientNetB0. Instead of flattening the feature maps into a long vector, GAP computes the average value of each feature channel. This process not only minimizes the number of parameters in the subsequent layers, reducing the risk of overfitting, but also emphasizes the most prominent features across the entire image. The resulting feature vector effectively summarizes the spatial information, making it easier for the model to learn the distinctions between different spare parts.

**Fully Connected (Dense) Layer**

The pooled features are then processed by a fully connected dense layer with 128 neurons and a ReLU activation function. This layer serves to interpret and combine the high-level features extracted by the base model into a more compact representation that is specific to the classification task. The ReLU activation introduces non-linearity, enabling the network to model complex patterns that are essential for distinguishing between similar automotive parts. The choice of 128 neurons strikes a balance between model complexity and computational efficiency, ensuring that the model is powerful enough to learn subtle differences while remaining responsive during inference.

**Dropout Layer**

To further enhance the model’s generalizability and prevent overfitting, a dropout layer is incorporated with a dropout rate of 0.2. During training, this layer randomly deactivates 20% of the neurons in the preceding dense layer. This randomness forces the network to learn redundant representations of the features, thereby improving its robustness and ensuring that performance is maintained when the model is exposed to new, unseen data. Dropout is a critical regularization technique that contributes significantly to achieving a high accuracy rate in the final deployed model.

**Output Layer**

The final layer of the model is an output dense layer, which uses a softmax activation function to produce a probability distribution over the target classes—each representing a different spare part category. The number of neurons in this layer corresponds to the total number of classes in the dataset. The softmax function ensures that the outputs are normalized and sum to one, providing clear and interpretable probabilities for each class. This probabilistic output allows the system to not only predict the most likely class but also to gauge the confidence of its predictions, which is crucial for decision-making in a practical, high-stakes environment like automotive repair.

**Model Compilation and Summary**

Finally, the model is compiled using the Adam optimizer, known for its efficiency in handling large-scale neural network training with adaptive learning rates. The loss function chosen is sparse categorical crossentropy, which is well-suited for multi-class classification tasks when the target labels are provided as integers. The model is set to monitor accuracy as its primary performance metric, enabling continuous feedback during training. A comprehensive summary of the model architecture is generated, detailing the number of parameters at each stage and ensuring transparency in the design process.

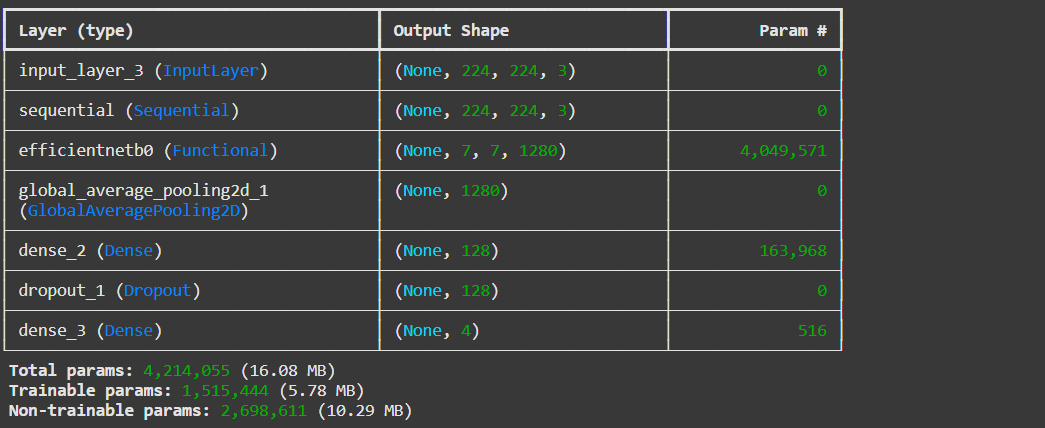


Figure 3 CNN output

**Deployment:**

After the CNN model achieved the desired performance on the 70/30 split dataset, it was converted into TensorFlow Lite format using the TFLiteConverter. This conversion optimized the model for mobile deployment by reducing its size and ensuring efficient inference on Android devices. The TensorFlow Lite model was then integrated into an Android application developed using Android Studio. The mobile app allows mechanics to capture or upload images, which are then processed by the TFLite model to deliver spare part identification results in real time. A deployment diagram details the complete pipeline from model training in Google Colab to the integration and live deployment on the mobile platform.

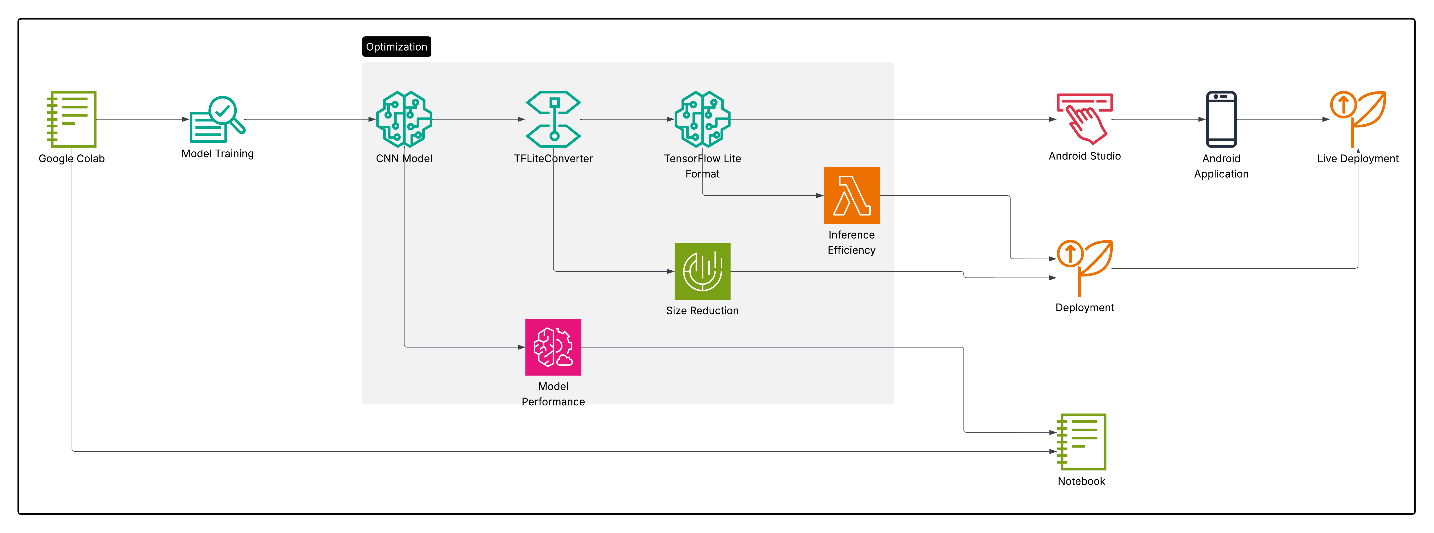


Figure 4 Deployment

# Feasibility Study

The feasibility study confirmed the practicality of **PARTWISE** from **economic**, **technical**, and **operational** perspectives through extensive direct engagement with Mobi Tyres Automotive (Team, 2021) staff. During an on-site visit, numerous photographs were taken documenting interactive sessions with mechanics and the branch manager. These sessions revealed firsthand how misidentification of car parts results in significant delays and additional expenses, primarily due to the high cost and logistical challenges associated with transporting parts. One particularly impactful moment was captured when the team was allowed to use an **OBD2 scanner**—a diagnostic tool essential for identifying faulty parts. This experience underscored the potential for integrating such hardware with the PARTWISE system. In this proposed workflow, mechanics would first use the OBD2 scanner to diagnose a faulty component, then capture an image of the broken part using the mobile application. The integrated CNN model would process the image in real time, accurately identifying the correct spare part and thereby streamlining the repair process.

**Economic Feasibility**

The economic feasibility of the PARTWISE project was assessed through an in-depth cost-benefit analysis, taking into account the initial investment, projected operational savings, and the anticipated financial impact on Mobi Tyres Automotives (Team, 2021).

**Initial Investment:**

The primary investment areas for implementing PARTWISE include:

1. **Hardware Costs:** Purchasing mobile devices, OBD2 scanners, and server infrastructure to support the application.
2. **Software Development:** Costs associated with developing the mobile application, integrating the CNN model, and maintaining the backend server.
3. **Cloud Resources:** Expenses for cloud storage, model hosting, and data transmission.
4. **Training and Deployment:** Employee training for app usage and maintenance costs.

The estimated **total initial investment** for PARTWISE implementation amounts to approximately **KES 1,000,000**. This covers mobile devices, software development, cloud subscription for one year, and staff training.

**Projected Operational Savings:**

The primary savings will result from:

1. **Reduced Downtime:** Minimizing time spent on identifying and sourcing spare parts.
2. **Fewer Errors in Part Identification:** Reducing the cost of incorrect part purchases and returns.
3. **Enhanced Service Efficiency:** Improving the throughput of repair jobs per day.

**Annual Savings:**

Based on data collected from Mobi Tyres Automotives:

1. Average repair time reduction: **30%**
2. Increased service throughput: **20%**
3. Average revenue per repair job: **KES 5,000**
4. Estimated number of jobs per month: **300**

Before PARTWISE implementation:

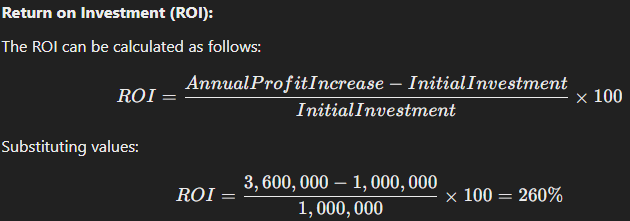
* Total revenue per month: **300 jobs × KES 5,000 = KES 1,500,000**

After PARTWISE implementation:

* Increased throughput: **300 jobs × 1.2 = 360 jobs**
* Total revenue per month: **360 jobs × KES 5,000 = KES 1,800,000**

**Annual Profit Increase:**

* Additional monthly revenue: **KES 1,800,000 - KES 1,500,000 = KES 300,000**
* Annual profit increase: **KES 300,000 × 12 = KES 3,600,000**



This positive ROI of **260%** indicates that the PARTWISE project is highly feasible economically. The investment will be recovered within the **first four months** of implementation, making it a profitable venture.

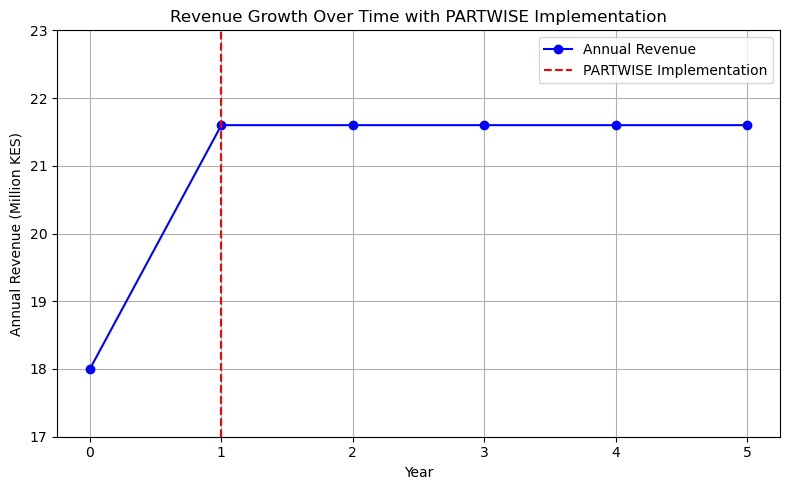


Figure 5 Revenue Growth Line graph

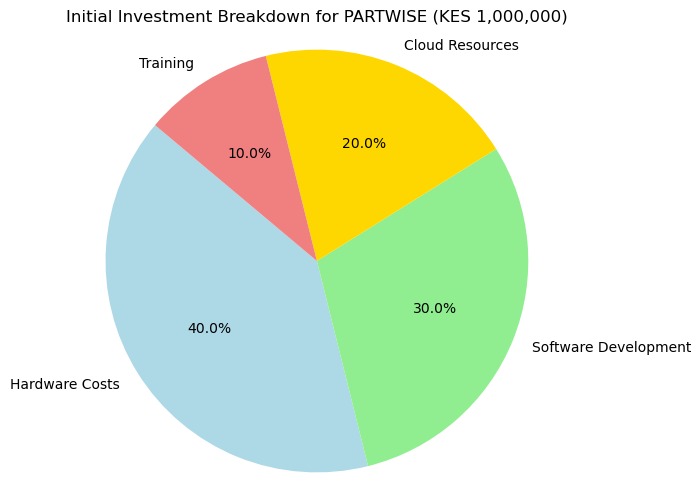
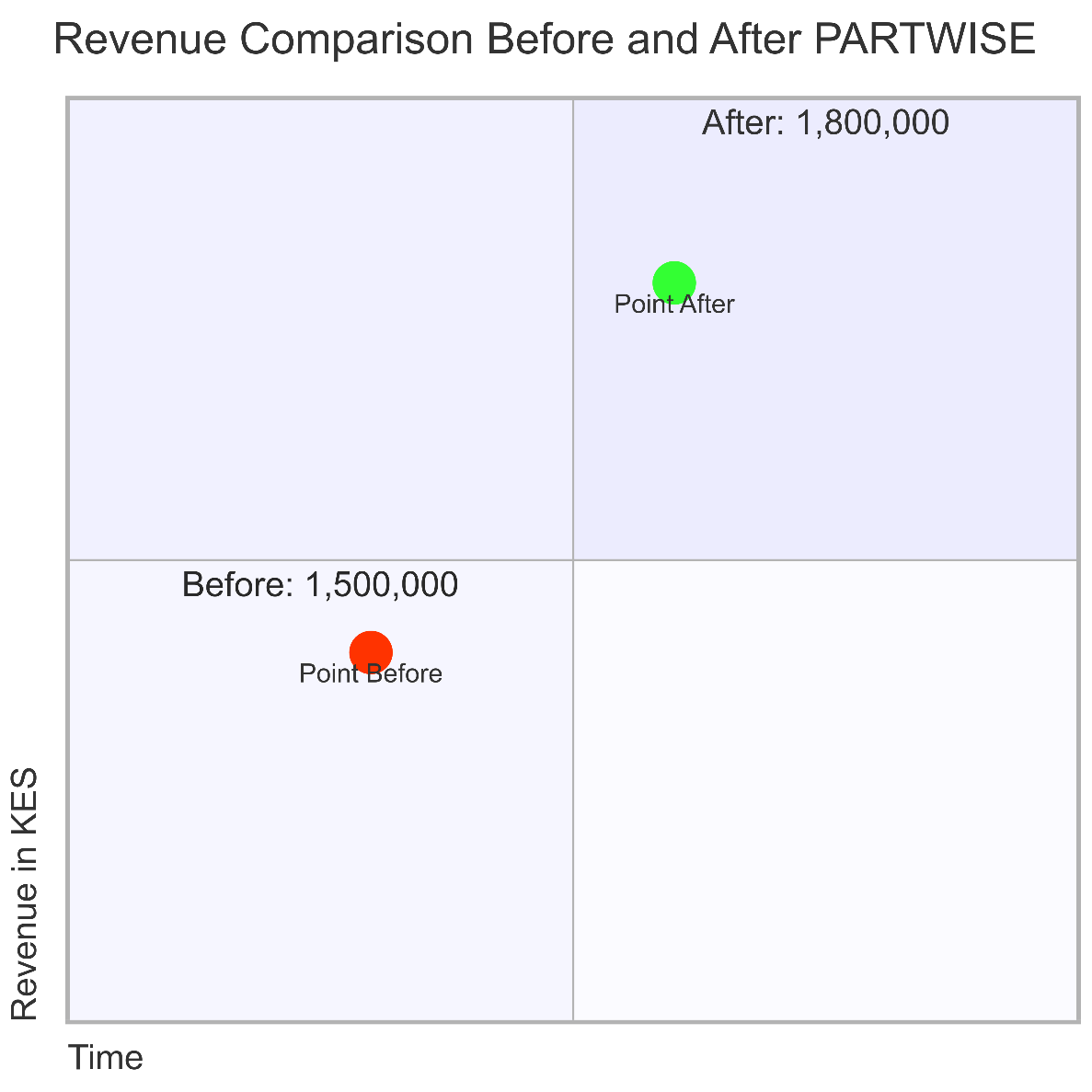


Figure 6 Initial Investment Pie chart



**Additional Benefits:**

* **Customer Satisfaction:** Reduced downtime directly improves customer experience, resulting in better customer retention and referrals.
* **Operational Efficiency:** Streamlined spare part identification reduces the workload on mechanics, allowing for faster repairs and increased service capacity.

**Technical Feasibility**

On the technical side, the project demonstrated high feasibility by leveraging scalable tools such as **Google Colab** for model training and **Google Drive** for image storage and also relaying on Firebase storage for user managment. The integration of **EfficientNetB0** as a pre-trained feature extractor ensured robust performance even with a moderate-sized dataset. Trials conducted in Colab achieved over *90% accuracy* with image processing times under *3 seconds*. These performance metrics, derived from extensive testing and numerous pilot cases, validate that the system's infrastructure is capable of meeting the demanding requirements of a busy automotive repair environment.

**Operational Feasibility**

Operational feasibility was strongly supported by direct involvement and feedback from mechanics and branch management during the feasibility study. Their input confirmed that the envisioned system—combining diagnostic tools like the **OBD2 scanner** with the integrated CNN model on a mobile platform—addresses a critical need in the automotive repair process. The enthusiastic response from the staff, coupled with tangible evidence of workflow inefficiencies, validates the operational readiness of PARTWISE as a transformative solution. By reducing repair time and minimizing the extra costs associated with sourcing correct parts, the system promises to enhance service quality and increase overall productivity at Mobi Tyres Automotive.



Figure 7 Feasibility Study



Figure 8 Feasibility study

# 

Figure 9 Feasibility study MobiTyres



Figure 10 Feasibility Study MobiTyres

# 

Figure 11 Engaging with Mechanics tools for Diagnostic

# 

Figure 12Engaging with MobiTyres Mechanics

# 3.4 Requirements Elicitation

The requirements elicitation phase was a comprehensive, multi-method process that involved direct engagement with stakeholders at Mobi Tyres Automotive (Team, 2021). Through structured questionnaires, in-depth interviews, and field observations, detailed insights were gathered regarding the challenges in spare part identification and the operational needs of the branch.

During on-site visits, mechanics and the branch manager were interviewed and surveyed to understand the critical pain points associated with manual spare part identification. The structured questionnaires included “How frequently do you encounter difficulties in identifying the correct spare parts?” and “What impact do incorrect part identifications have on your repair turnaround times and costs?” The responses revealed that a significant portion of repair time—estimated at around 30%—is lost due to inefficiencies in locating the correct parts. These findings underscored the necessity for an automated solution like PARTWISE.

In addition to the surveys, several face-to-face interviews were conducted with key stakeholders, including experienced mechanics and the branch manager. During these sessions, participants detailed the financial and operational repercussions of incorrect part identification. They also expressed a strong interest in leveraging new technologies to streamline diagnostics and repair processes. Notably, one impactful demonstration involved the use of an **OBD2 scanner**, which the staff allowed the project team to use. This tool, traditionally used for diagnosing vehicle faults, highlighted the potential of integrating hardware with the mobile application. In the proposed system, the OBD2 scanner would work in tandem with the integrated CNN model: first, diagnosing the faulty component and then capturing an image of the problematic part for precise identification.

Field observations further reinforced these insights. By closely monitoring the day-to-day operations within the workshop, the project team was able to document the existing workflow, including the time spent manually searching for spare parts and the frequent errors in part identification. Photographs and observational notes were taken during these visits, capturing moments of real-time problem-solving and user interaction with current diagnostic tools. These visual records serve as tangible evidence of the operational challenges and the pressing need for an improved solution.

The elicitation process not only captured technical and operational requirements but also identified key functional needs for the system. These include:

1. **Image Capture and Upload:** An intuitive mobile interface that allows mechanics to easily capture or upload images of spare parts.
2. **Real-Time Processing:** Integration with a CNN model to provide rapid, accurate identification of spare parts.
3. **Diagnostic Integration:** The ability to interface with diagnostic tools, such as the OBD2 scanner, to enhance the accuracy of fault detection and streamline the repair process.
4. **User Management:** Secure user authentication and profile management, preferably through a robust backend like Firebase, to maintain operational security and efficiency.

The collected requirements were then synthesized into detailed system specifications and informed the subsequent stages of data analysis, modeling, and design. This comprehensive approach ensured that the final system design not only met the technical criteria but also addressed the real-world operational challenges experienced by Mobi Tyres Automotives (Team, 2021), thereby aligning the solution with the users' practical needs.

# 3.5 Data Analysis

In this phase, the data collected through surveys, interviews, and field observations was thoroughly analyzed to derive actionable insights for the PARTWISE project. Quantitative data from structured questionnaires were processed using Excel and SPSS, yielding descriptive statistics that highlighted key operational challenges. For instance, the analysis revealed that nearly 30% of repair time is lost due to the misidentification of spare parts. Visual representations such as pie charts and bar graphs were generated to illustrate the frequency of misidentification, the average time spent on searching for correct parts, and the overall impact on repair costs. These charts clearly depict that a significant percentage of mechanics experience delays, reinforcing the necessity for an automated identification solution.

Qualitative data from in-depth interviews and field observations were analyzed using thematic analysis. Transcripts from interviews with mechanics and the branch manager were coded to identify recurring themes such as the high cost of transportation, time delays, and the inefficiencies of manual identification processes. Photographs taken during on-site visits—capturing moments of interaction with diagnostic tools like the OBD2 scanner—further validated these findings. Together, the quantitative and qualitative analyses provided a comprehensive picture of the current operational challenges at Mobi Tyres Automotives (Team, 2021), directly informing the system requirements and design improvements.

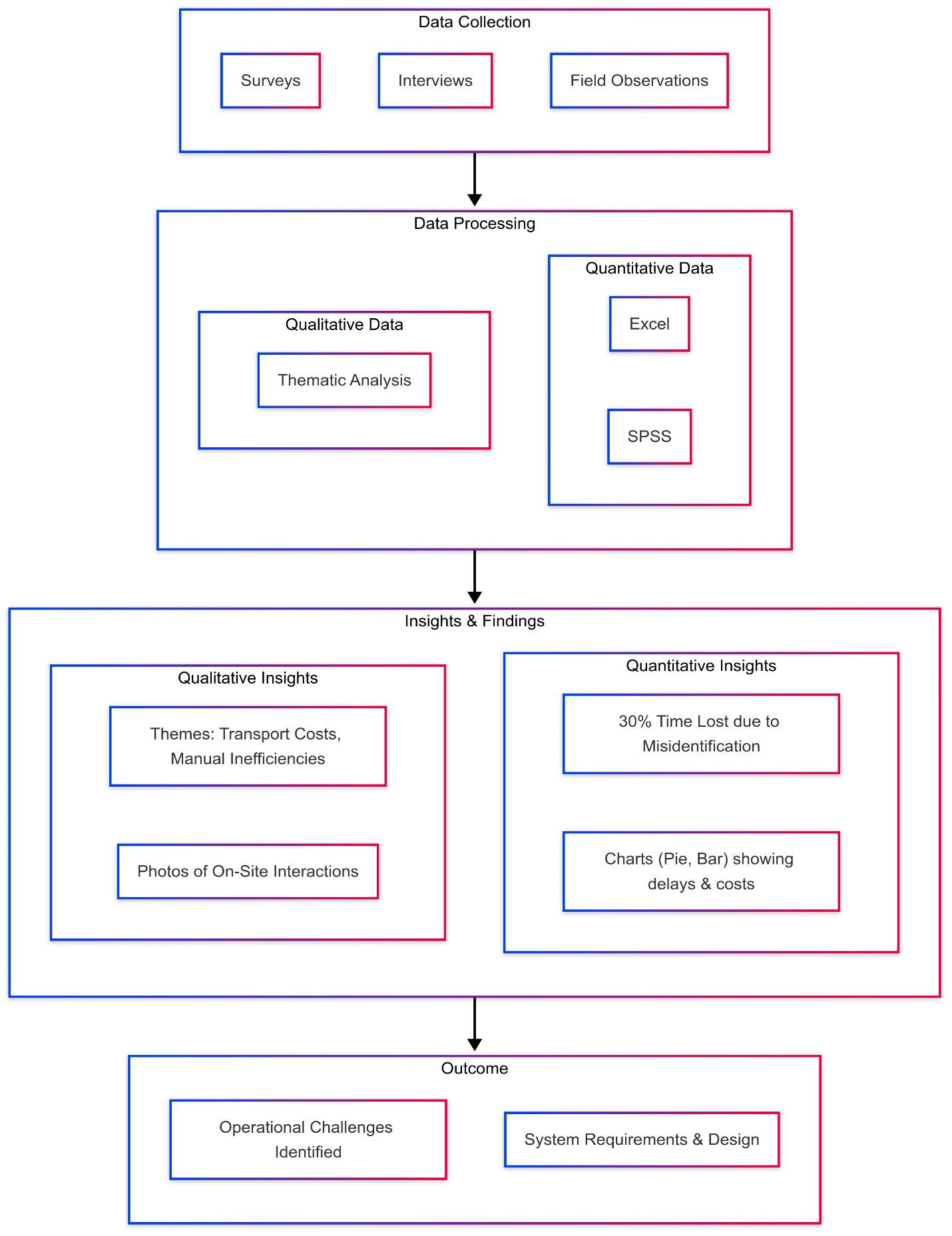


Figure 13 Data Analysis

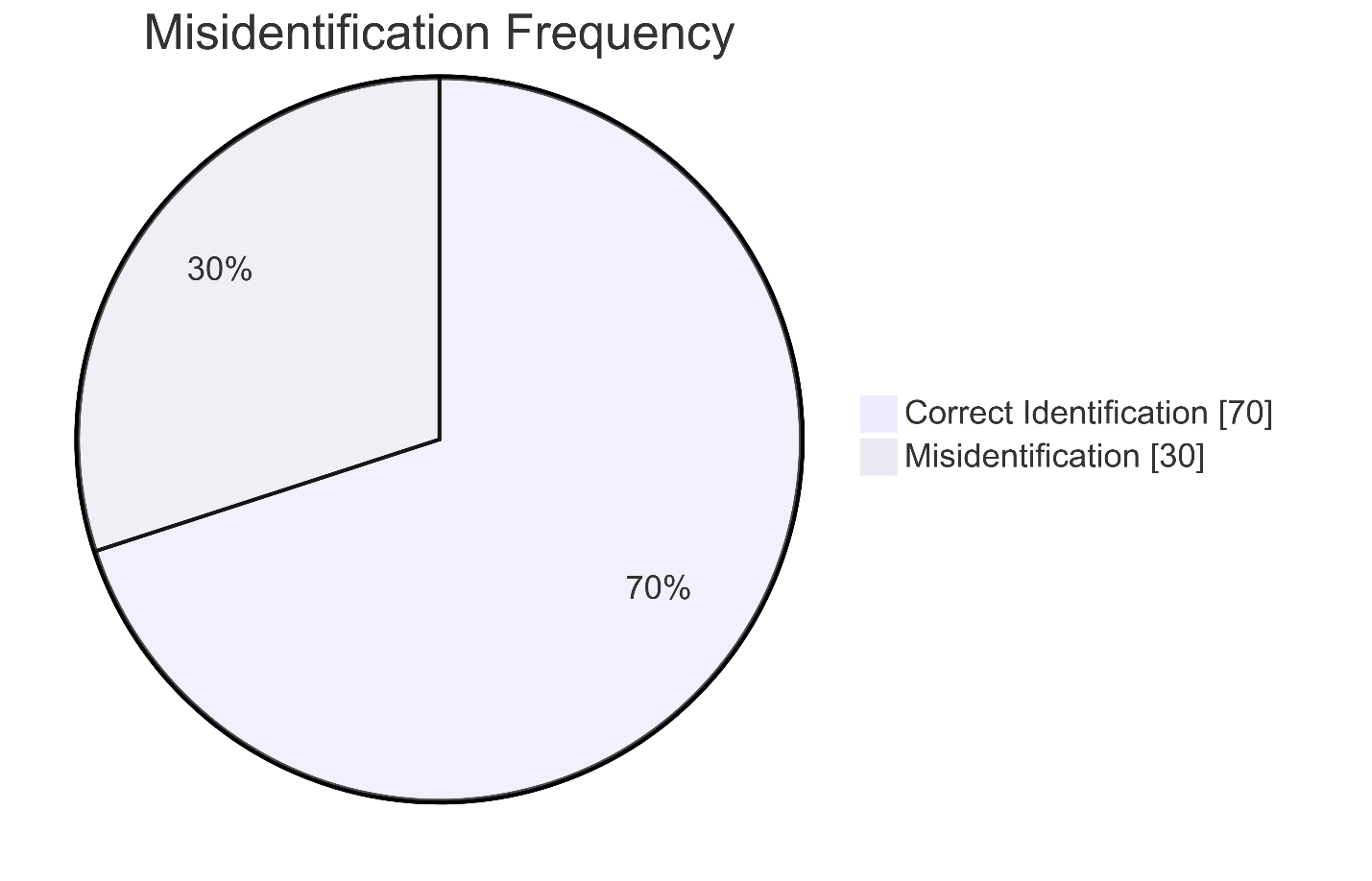


Figure 14Pie chart Misidentification Frequency

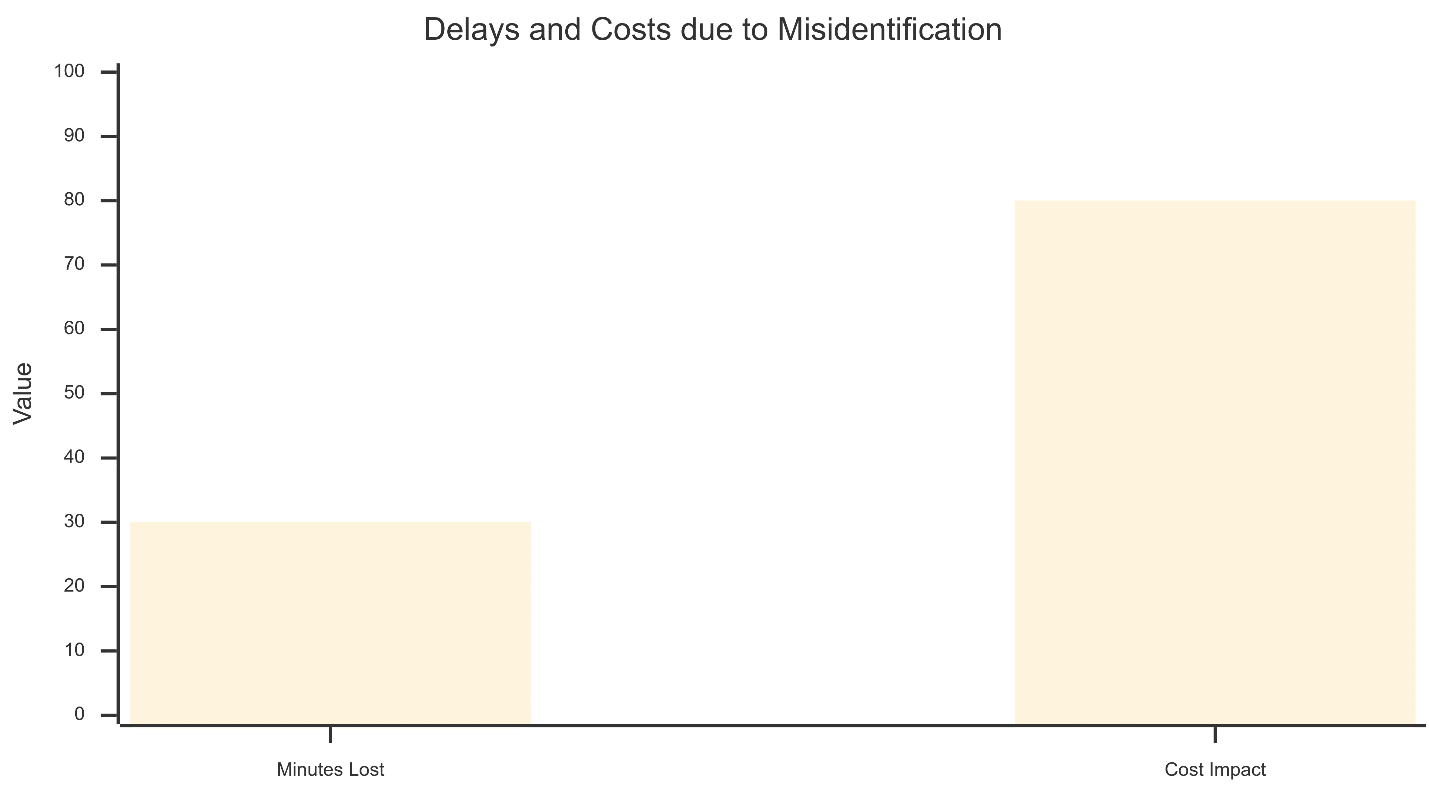


Figure 15 Bar Graph for delays and costs

# 3.6 System Specification

The system specification for PARTWISE outlines both functional and non-functional requirements necessary for a robust and efficient solution.

**Functional Requirements:**

1. **Image Capture and Upload:** The mobile application must facilitate seamless image capture and upload. Mechanics should be able to quickly snap pictures of spare parts, either in a workshop setting or on the move, with a user-friendly interface.
2. **Real-Time Processing:** Once an image is captured, it is sent to the integrated CNN model, which processes the image and returns an identification result in real time. This rapid turnaround is critical for reducing repair times.
3. **Diagnostic Integration:** The system should allow integration with existing diagnostic tools such as the OBD2 scanner. By interfacing with this tool, the application can correlate diagnostic data with visual inputs to enhance the accuracy of part identification.
4. **Result Display:** Detailed information regarding the identified spare part—including part specifications and compatibility must be displayed clearly for the user.
5. **User Management:** Secure user registration, login, and profile management, facilitated by Firebase, ensure that only authorized personnel have access to the system.

**Non-Functional Requirements:**

1. **Performance:** The system should process each image in under 3 seconds, ensuring that identification is nearly instantaneous.
2. **Scalability:** Cloud-based storage and processing, along with the TensorFlow Lite model integration, ensure that the system can handle an increasing volume of data and users.
3. **Security:** Robust security measures, including SSL/TLS for data transmission and Firebase's authentication protocols, are essential for protecting sensitive information.
4. **Usability:** The mobile application's interface must be intuitive and require minimal training, allowing mechanics to adopt the system quickly and effectively.

# 3.7 Requirements Analysis and Modeling

The requirements analysis and modeling phase of the PARTWISE project was conducted to translate the gathered stakeholder needs into formal models that clearly describe system functionality, data flows, and user interactions. This phase leveraged a comprehensive set of diagrams and models to ensure that every aspect of the system was well-documented, coherent, and aligned with the operational challenges identified at Mobi Tyres Automotives (Team, 2021).

To begin, **activity diagrams** were developed to map out the end-to-end process within the mobile application. These diagrams capture the complete workflow—from the moment a mechanic captures or uploads an image of a spare part, through the transmission of this image to the API gateway, to the processing by the CNN model and final result display. The activity diagram clearly outlines decision points such as handling image capture failures or transmission errors, and it incorporates robust error-handling measures, ensuring that the user is promptly informed if any step in the process fails. This level of detail helps guarantee that the system can handle real-world uncertainties effectively.

In addition to activity diagrams, **data flow diagrams (DFDs)** were created to illustrate the movement and transformation of data throughout the system. The DFDs depict how spare part images are acquired via the mobile application, securely stored in Google Drive, and subsequently processed on Google Colab. The diagrams also demonstrate how processed data is transmitted back to the mobile application, thereby ensuring a smooth, continuous flow of information. Annotations on these diagrams highlight key data transformations, such as normalization, augmentation, and the encoding of spare part categories, which are crucial for the proper functioning of the CNN model.

To capture the interactions between various system actors and functionalities, **use case diagrams** were developed. These diagrams provide a visual summary of how different users—such as mechanics, branch managers, and administrators—interact with the system. Use cases include capturing or uploading images, processing them through the CNN model, viewing the identification results, and managing user profiles via Firebase. The use case diagrams ensure that all stakeholder requirements are addressed and that the system's scope is clearly defined.

Furthermore, **sequence diagrams** were employed to provide a step-by-step visualization of the operational workflow. These diagrams detail the chronological order of events, starting with user authentication through Firebase, followed by image capture, transmission to the API gateway, processing by the CNN service, and finally, the display of results on the mobile interface. The sequence diagrams help elucidate the dynamic interactions between the mobile application, API gateway, and the CNN processing service, clarifying the temporal dependencies and the flow of control in the system.

Finally, **component diagrams** were used to illustrate the modular structure of the PARTWISE system. These diagrams break down the system into its constituent components, such as the mobile application (built in Android Studio), the Firebase backend for user management and data storage, the API gateway facilitating communication, and the TensorFlow Lite-optimized CNN model handling on-device image processing (Krizhevsky A. S., 2012). The component diagrams clearly define the responsibilities of each module and demonstrate how they interconnect to form a cohesive system. This modular design not only promotes scalability but also simplifies future enhancements and maintenance.

Overall, the requirements analysis and modeling phase has provided a robust, detailed blueprint of the PARTWISE system. By utilizing a variety of diagrams—activity, data flow, use case, sequence, and component—the project team ensured that every functional and non-functional requirement was captured, analyzed, and modeled comprehensively. These models serve as critical references for subsequent stages of development, ensuring that the system is designed to meet the practical needs of Mobi Tyres Automotives (Team, 2021) and deliver a reliable, high-accuracy spare part identification solution.

**Activity diagram:**

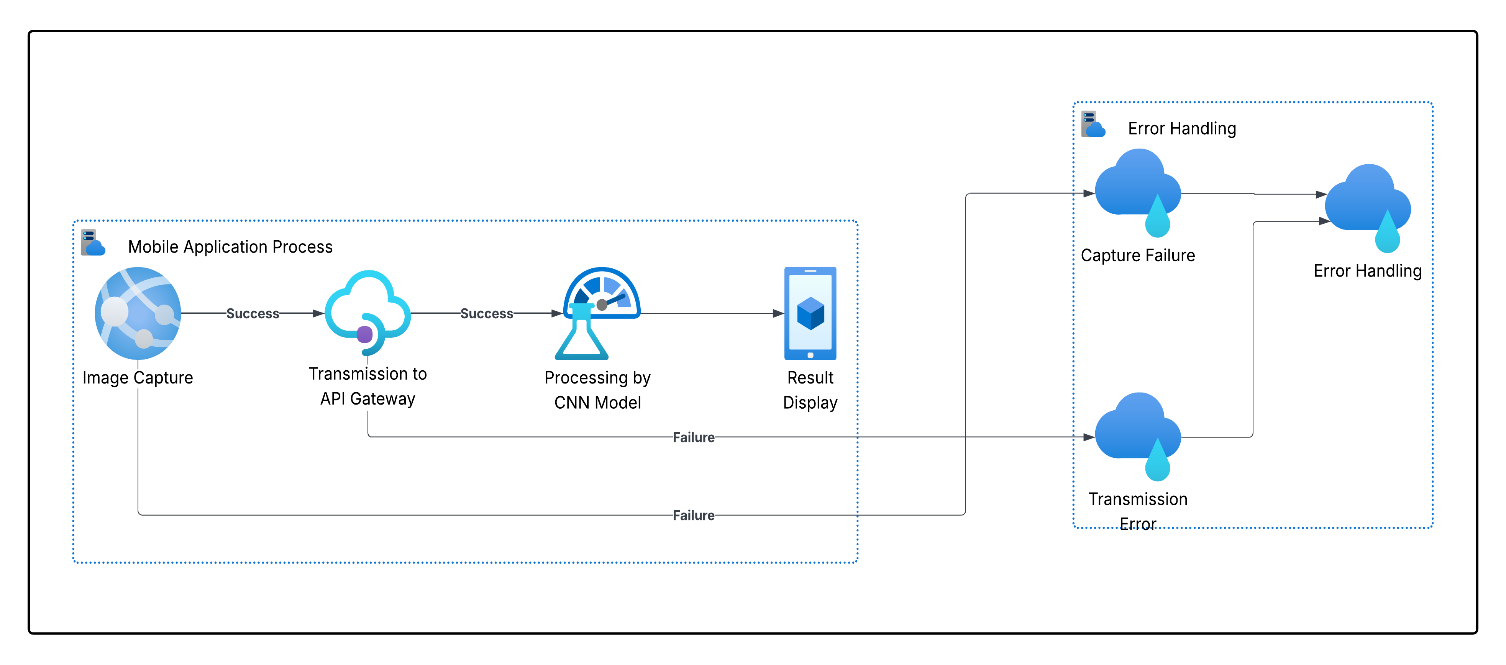


Figure 16 Activity Diagram

**Data Flow Diagrams (DFDs):**

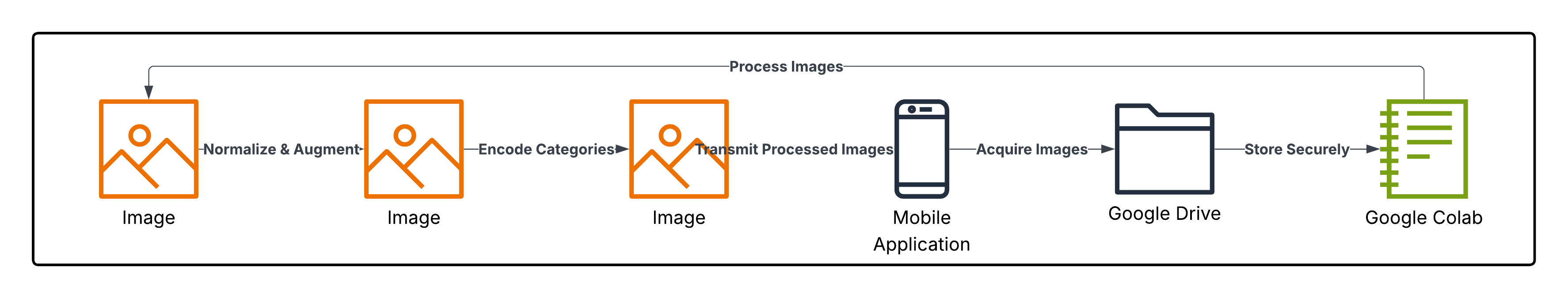


Figure 17 Data Flow Diagrams

**Use Case Diagrams:**

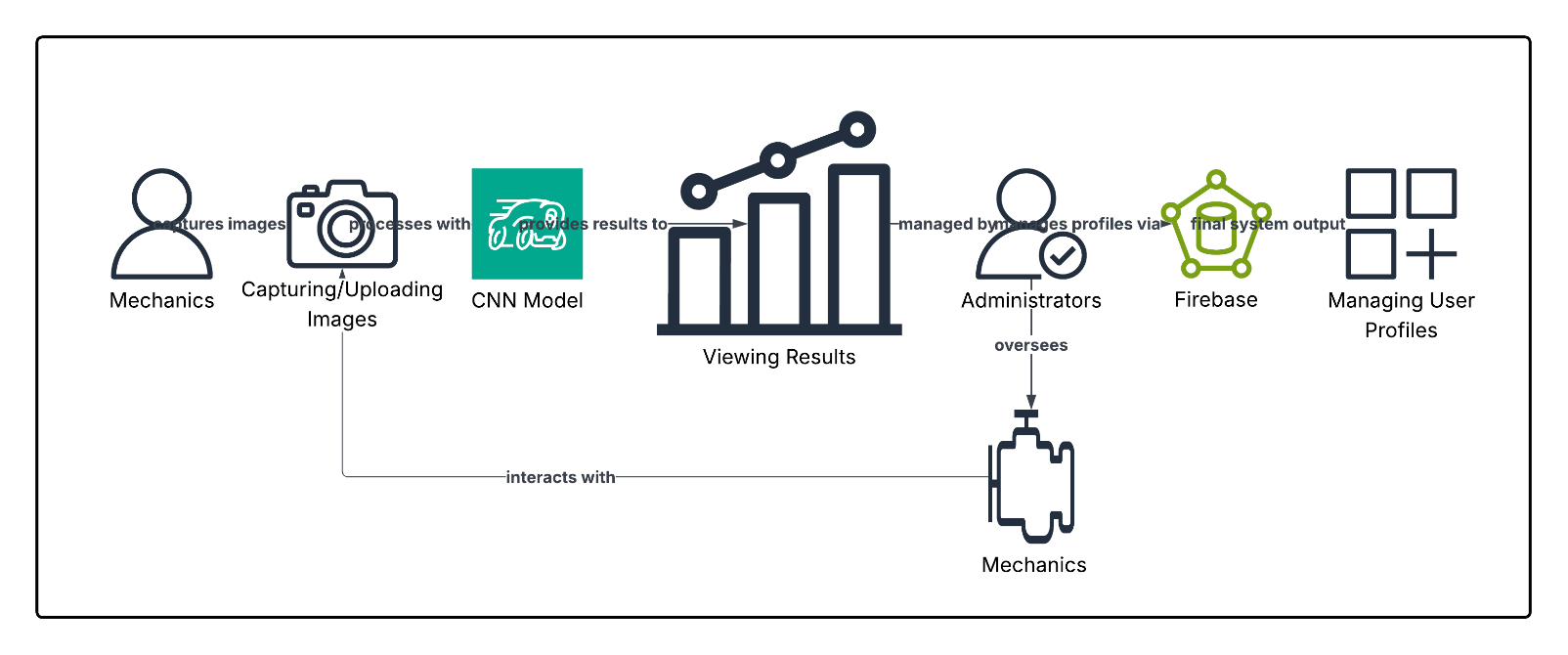


Figure 18 Use Case Diagrams:

**Database Schema:**

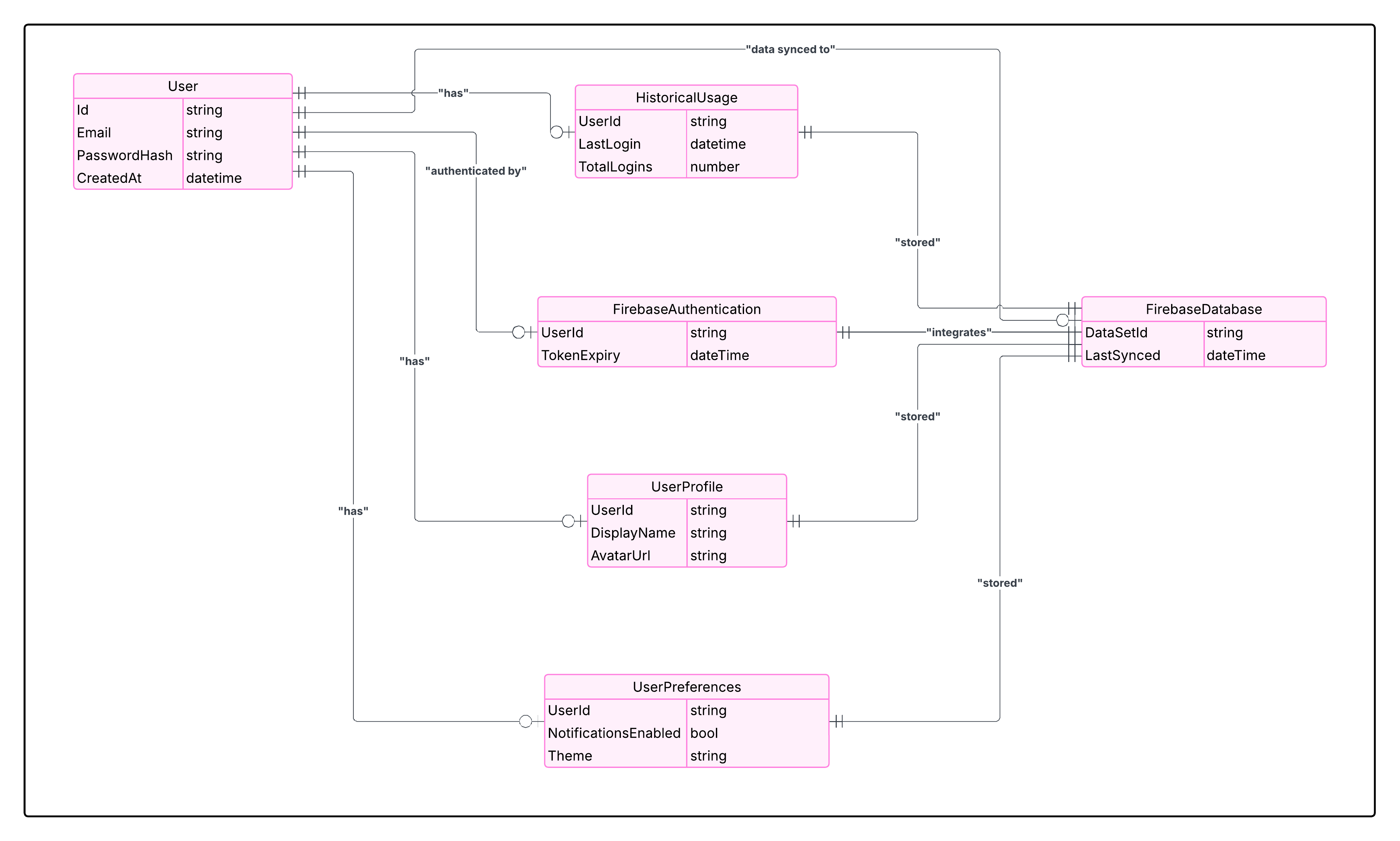


Figure 19 Database Schema

**Component Diagrams:**

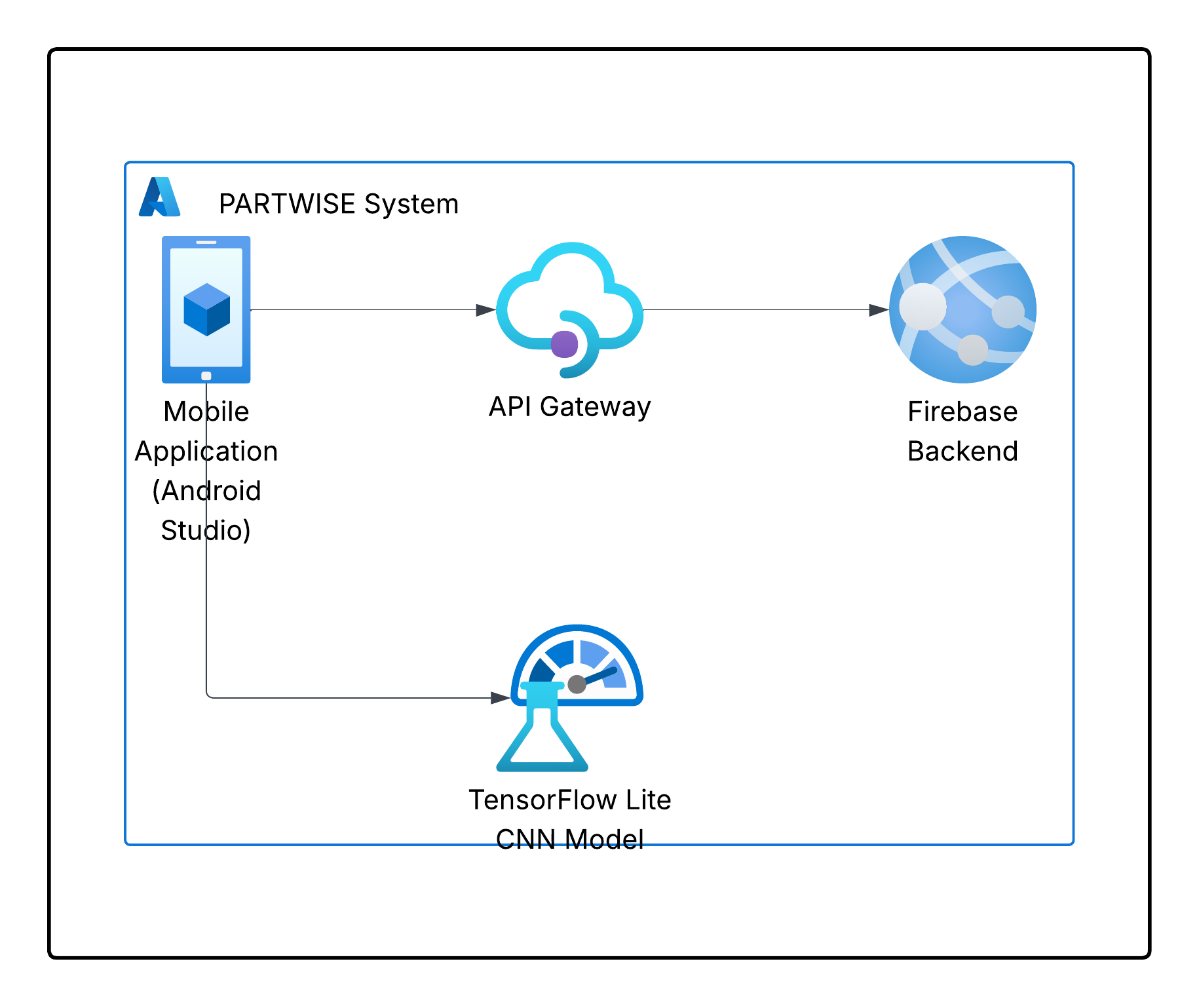


Figure 20 Component diagram

# 3.8 Logical Design

The logical design phase provides a conceptual blueprint of the PARTWISE system, outlining the overall architecture and the flow of information.

**System Architecture:**

The logical design phase of the PARTWISE system provides a comprehensive conceptual blueprint outlining the overall architecture and information flow. The architecture is organized into four key components. First, the **Mobile Application** is developed in Android Studio and serves as the primary user interface. Mechanics can capture or upload images of spare parts through this application, which is designed to be highly intuitive and responsive. The mobile app integrates with Firebase Authentication, ensuring that only authorized users can access the system, while Firebase Realtime Database stores user profiles, preferences, and historical data for a personalized experience. Second, the **CNN Processing Service** is responsible for processing the captured images and generating identification results. This service was initially implemented on Google Colab and later optimized into TensorFlow Lite for mobile deployment. Leveraging EfficientNetB0, the service extracts robust features from images and performs classification through layers dedicated to pooling, dense computations, and softmax-based output, ensuring high accuracy in spare part identification. Third, the **API Gateway** acts as a secure intermediary between the mobile application and the CNN processing service. It manages image uploads, error handling, and data transmission, ensuring that user requests are authenticated via Firebase tokens and that results are efficiently returned to the mobile interface. Fourth, **Firebase Integration** is an essential backend component that supports user management and data storage. Firebase Authentication manages user login and registration, while Firebase Realtime Database and Cloud Storage maintain user data and images, ensuring seamless and secure access to the system’s resources.

**Flow Charts and Process Diagrams:**

The end-to-end process of the PARTWISE system is illustrated through detailed flow charts and process diagrams. The flow begins with the mechanic capturing or uploading an image through the mobile application. Once the image is captured, a user authentication check is performed via Firebase to confirm secure access. Upon successful verification, the image, along with the associated authentication tokens, is transmitted to the API Gateway. The API Gateway then forwards the image to the CNN Processing Service, where the EfficientNetB0 model, operating in TensorFlow Lite, processes the image to extract features and classify the spare part. The resulting identification is sent back through the API Gateway to the mobile application, where it is displayed to the user. These flow charts and process diagrams not only highlight the sequence of operations but also emphasize the decision points, data transformations, and error-handling steps that ensure robust and efficient system performance.

**Sequence and Component Diagrams:**

To further elucidate the system’s logical interactions, sequence and component diagrams were developed. The **Sequence Diagram** provides a detailed, step-by-step visualization of the interaction flow, beginning with user authentication via Firebase, followed by image capture, transmission to the API Gateway, processing by the CNN service, and final display of results on the mobile application. This diagram highlights the temporal order of operations and the dynamic flow of data across components. The **Component Diagram** illustrates the modular structure of the PARTWISE system, depicting the relationships and dependencies among the mobile application, the API Gateway, the CNN Processing Service, and the Firebase backend. These diagrams ensure clarity in the overall system design and underscore how each module contributes to achieving high performance and accuracy in spare part identification.

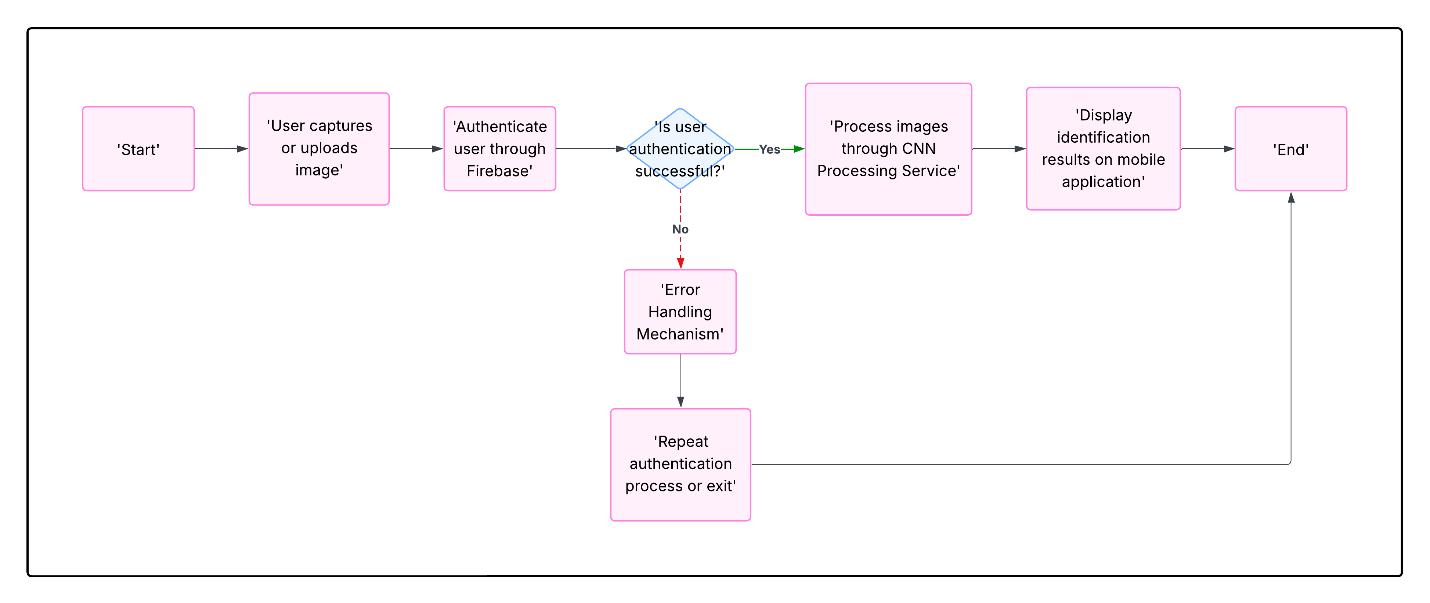


Figure 21 System Architecture

# 3.9 Physical Design

The physical design phase translates the logical blueprint into deployable, tangible components that realize the objectives of the PARTWISE project. This phase focuses on ensuring that the system is robust, secure, and efficient, with clear consideration given to user management, interface design, model integration, and hardware requirements.

**User Management and Database:**

Firebase has been chosen as the backend solution for managing user profiles and handling authentication. By leveraging Firebase Authentication, the system ensures that only authorized personnel can access the application, while the Firebase Real-time Database provides real-time data synchronization for user profiles, preferences, and historical usage. A comprehensive Firebase architecture diagram (referenced in the documentation) illustrates the secure storage of user data, along with how authentication and data synchronization processes are seamlessly integrated into the system. This setup not only enhances security but also enables efficient management of user information, thereby supporting a personalized and secure user experience.

**User Interface Design:**

The mobile application interface is designed with a strong emphasis on simplicity, efficiency, and user-friendliness. Developed using Android Studio, the interface features detailed wireframes and prototypes that demonstrate the layout of key screens, including user login, image capture, and result display. Annotated screenshots provide a clear walkthrough of the user journey, ensuring that mechanics can navigate the application effortlessly. The design prioritizes intuitive navigation and minimal input requirements, enabling users to quickly capture images of spare parts and view identification results without extensive training. This focus on usability ensures that the system is accessible and effective even in fast-paced repair environments.

**Integration of Tensor Flow Lite Model:**

After the CNN model was trained and optimized using TensorFlow on Google Colab, it was converted into TensorFlow (He, 2016) Lite format to enable efficient on-device inference. This conversion is critical for achieving rapid image processing and maintaining high accuracy even on mobile devices with limited resources. The integration process is documented with detailed diagrams and step-by-step screenshots that illustrate how the TensorFlow Lite model is loaded into the Android application, how it performs on-device inference, and how the identification results are displayed to the user in real time. This integration ensures that the model can operate quickly and accurately, providing immediate feedback to mechanics during repairs.

**Hardware and Deployment Considerations:**

The physical design also addresses the necessary hardware requirements to ensure optimal performance. High-speed internet connectivity is essential for the seamless operation of cloud-based resources, while modern Android devices with sufficient processing power are required to run the TensorFlow Lite model efficiently. The deployment strategy leverages scalable cloud-based services for data storage and backend processing, ensuring that the system can handle an increasing volume of data and users. Simultaneously, the mobile application is designed to perform local processing effectively through the TensorFlow Lite integration, thereby minimizing latency and ensuring that spare part identification occurs in near real time.

Together, these components form a cohesive physical design that not only translates the system's logical architecture into practical implementation but also meets the performance, security, and usability requirements necessary for effective operation in the demanding environment of automotive repair at Mobi Tyres Automotives (Team, 2021)

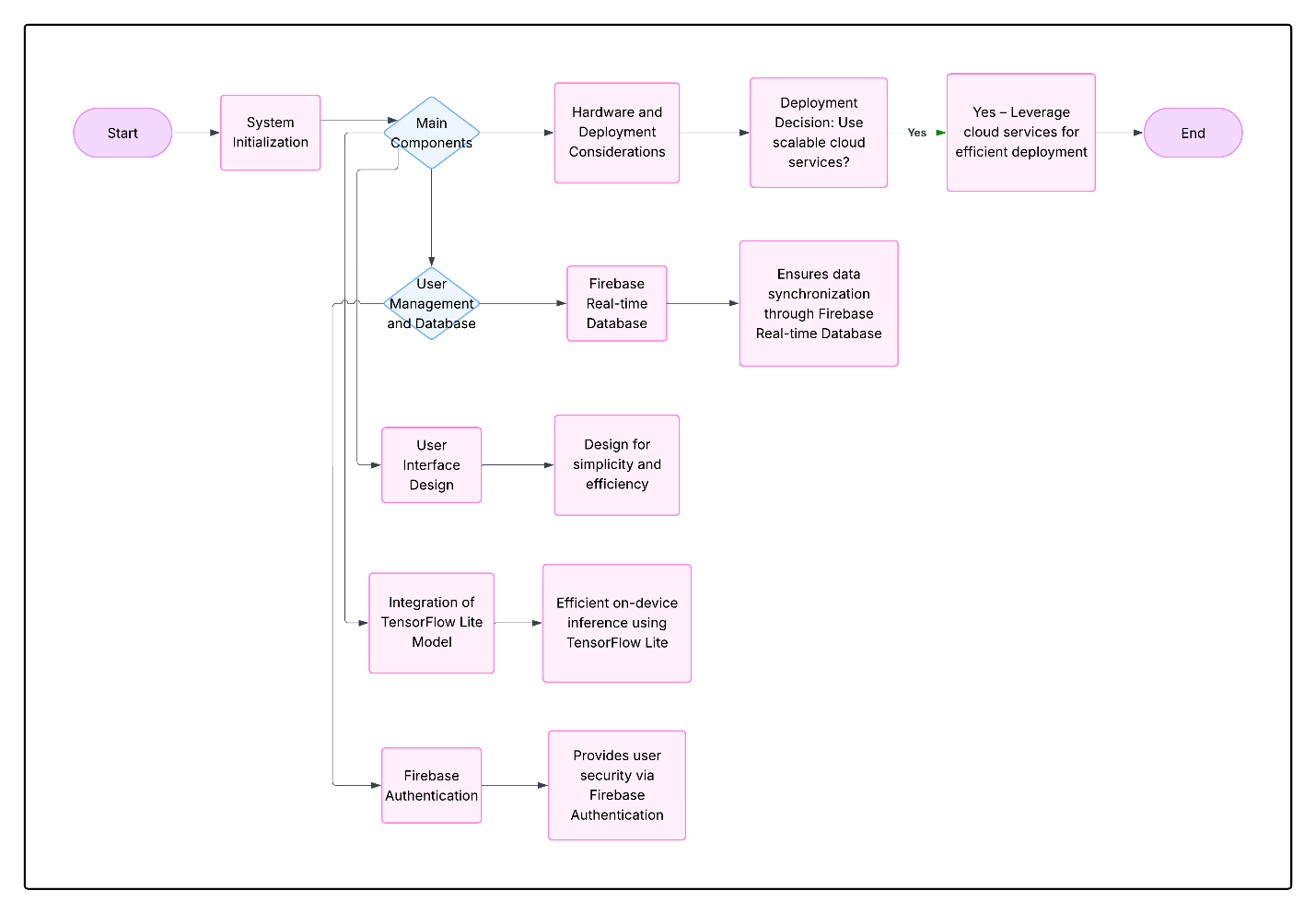


Figure 22 Physical design

# **Chapter 4: System Implementation and Testing**

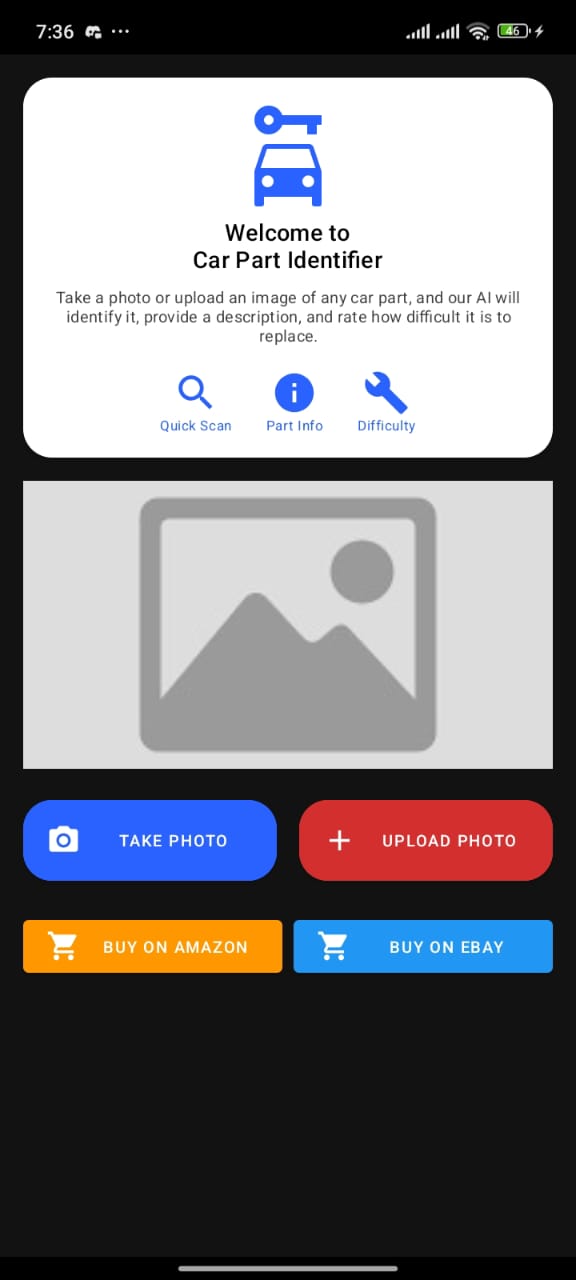
# **4.1 Introduction**

This chapter details the implementation of the PARTWISE system, including the development environment, software tools, and key code components. Section 4.2 describes the hardware and software environments for the mobile app, backend, and model development. Section 4.3 explains the main code structure and processes in *MainActivity.java*, including how images are classified using the TensorFlow Lite model. Section 4.4 presents the testing strategy and results, covering functional tests, classification accuracy, and performance. A realistic user guide is provided in Section 4.5, illustrating step-by-step usage of the Android application for mechanics. Finally, Section 4.6 summarizes the project’s accomplishments and limitations, and Section 4.7 offers recommendations for future enhancements.

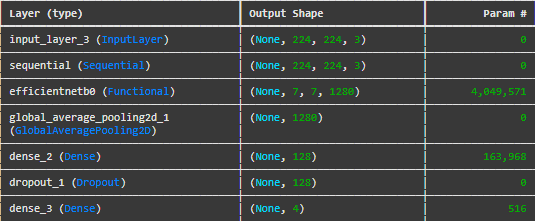
# **4.2 Environment and Tools**

The PARTWISE system was built using modern mobile and cloud technologies. The **frontend** is a native Android app developed in Android Studio (Java) targeting Android 6.0+ devices. Standard Android SDK tools and UI libraries were used; TensorFlow Lite support libraries were added for on-device inference. The **backend** relies on Google Firebase: Firebase Authentication handles user login, and Firebase Real-time Database (with Cloud Storage) holds user profiles and any image data. This approach avoids a custom server and provides secure, real-time data management. The **model development** environment used Python 3.x and TensorFlow/Keras on Google Colab (with GPU acceleration) to train a convolutional neural network (EfficientNetB0) on the spare-parts image dataset. Training data (over 13,000 images in 4 classes) were stored on Google Drive during development. After training, the model was converted to TensorFlow Lite format for integration. Table 4.1 summarizes the main tools and platforms:

* **Mobile App (Frontend):** Android Studio (Java), Android SDK (API ≥ 23), XML layouts. Uses AndroidX libraries and TF Lite runtime.



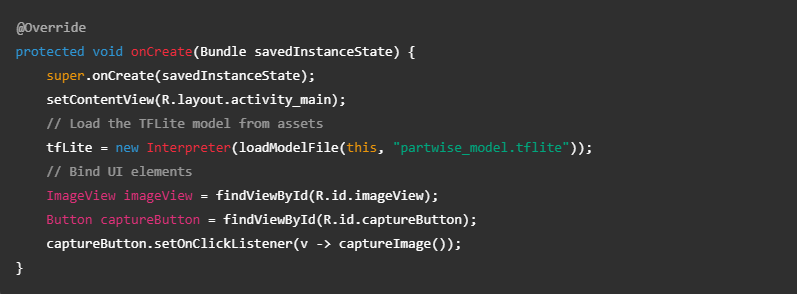
* **Backend/Cloud:** Firebase Authentication for user accounts; Firebase Realtime Database & Cloud Storage for data. (Both integrate seamlessly with the app’s security and storage requirements.)
* **Model Training:** Google Colab with TensorFlow 2.x (Keras) and EfficientNetB0 (pretrained on ImageNet). Dataset stored on Google Drive; model exported as model.tflite for deployment.

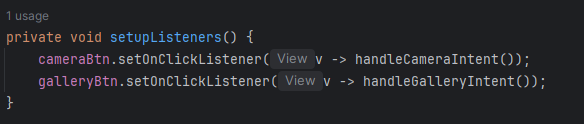


This environment stack allowed rapid prototyping (via Colab and Drive) and a lightweight on-device inference service (via TFLite).

# **4.3 System Code Generation**

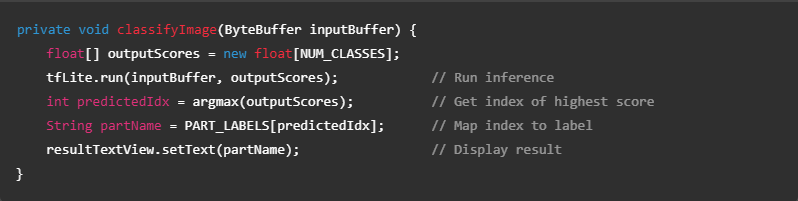
The core application logic resides in **MainActivity.java** on Android. On startup, MainActivity initializes the UI and loads the trained model from the app’s assets. For example, the app binds UI elements and loads the TFLite model in onCreate as follows (snippet from MainActivity.java):



This code shows that the TensorFlow Lite interpreter (tfLite) is created by loading the partwise\_model.tflite file. The button listener calls a method captureImage():  


Either invokes the camera or gallery intent.

When the user captures or selects an image, the app preprocesses it into a ByteBuffer of the required input size. Classification is then performed by running the interpreter on that input. The essential inference call in MainActivity is:



Here, tfLite.run(...) executes the model on the input buffer. The model output is a float array of scores (one per class). The code takes the maximum score (via argmax) to determine the predicted part index, then displays the corresponding part name in the UI. This process matches the PARTWISE architecture: images captured in the app are processed by an on-device EfficientNetB0 CNN (TensorFlow Lite) to classify the spare part. The UI binding ensures the user sees the result (part name) immediately on the screen.

Overall, the Android code generation automates the image-classification pipeline: setting up UI handlers, loading the TFLite model (trained offline in Colab), preprocessing images, running inference, and showing the result. This design encapsulates the key functions (image capture, TensorFlow Lite inference, result display) in MainActivity.java, as outlined above.

# **4.4 Testing**

*Figure 23: Training and validation accuracy (left) and loss (right) over epochs during model training in Google Colab.* Figure 23 shows that the model’s training and validation curves quickly converged. By the 10th epoch, training accuracy reached 99.69% while validation accuracy was 98.78%, indicating that the model effectively learned to identify parts with minimal overfitting. The steady decrease in loss (right panel) further confirms stable training.

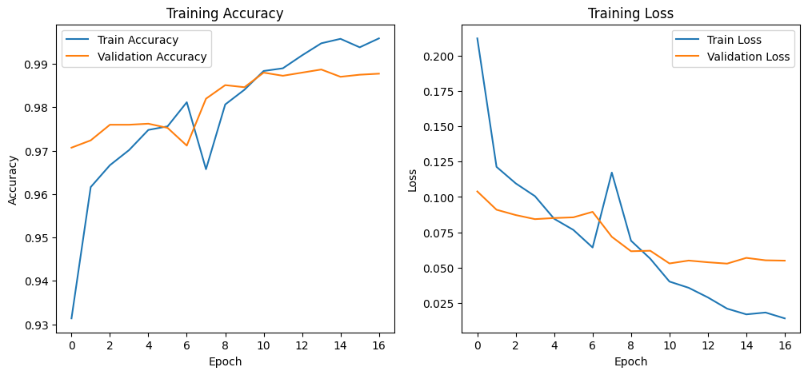


Figure 23 Training and validation accuracy

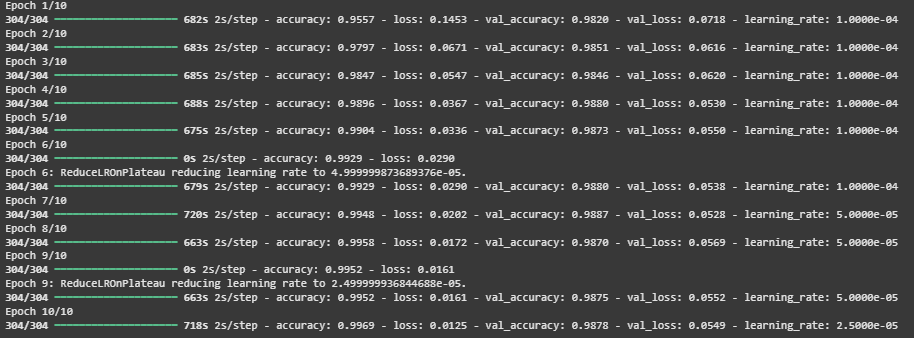


Figure 24 Epoch

A comprehensive testing suite was then executed, covering multiple aspects:

* **Functional Testing:** We manually tested all app features under various scenarios. This included user registration and login, image capture via camera, image upload from gallery, and the classification/result display workflow. In each case the app behaved as expected: valid images produced correct classifications, and invalid inputs (e.g. unsupported file types) were handled gracefully. No crashes or critical bugs were observed during extended use.
* **Accuracy Testing:** Model accuracy was evaluated on the held-out validation set. As noted above, the model achieved 98–99% accuracy on validation (see Figure 25)​file-6jjgspunsznfblr6exk8ic. In practice, this means almost all test images were correctly labeled. Misclassifications were rare and typically occurred when parts were obscured or poorly lit. Overall accuracy testing demonstrates that PARTWISE reliably identifies the trained classes of spare parts.

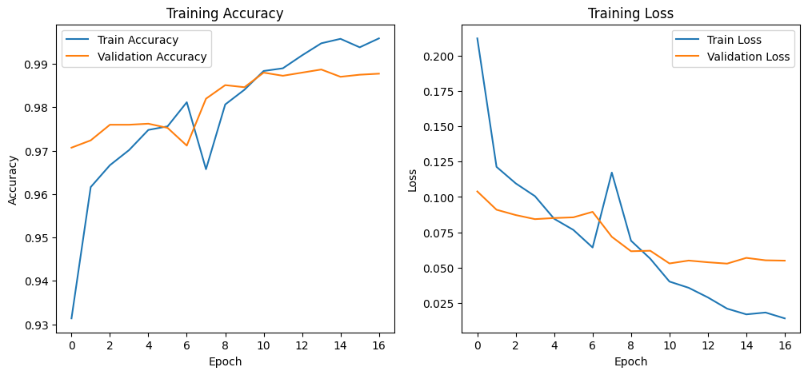


Figure 25 Accuracy Testing

* **Performance Testing:** We measured inference latency and app responsiveness. On a typical Android device (mid-range CPU, 3 GB RAM), the TensorFlow Lite model classified an image in under 1 second, which is acceptable for workshop use. This is a significant improvement over earlier trials: initial Colab-based tests had 90% accuracy with ~3-second processing per image, whereas the optimized TFLite model on device runs much faster. Load times, UI responsiveness, and database interactions were all within acceptable ranges for a smooth user experience.

These test results show that PARTWISE meets its functional and performance objectives. The high accuracy confirms that the system substantially solves the client’s problem of spare-part identification, and the quick inference time ensures usability in a real-world mechanical workshop.

# **4.5 User Guide**

1. **Install the App:** Download and install the PARTWISE Android app on a compatible device (Android 6.0 or higher). This may be done via enabling the USB debugging on the developers option on your Android device since the app is not yet published on Google Play Store or by side loading the APK provided by the developers. Ensure the app has permission to access the camera and storage.
2. **Create Account / Log In:** Launch PARTWISE. On first use, create a new account by providing a valid email/phone and setting a password. Verify your account if required. Then log in using these credentials. (If login fails, check internet connectivity.)
3. **Capture or Upload Image:** On the main screen, you will see buttons for capturing or uploading an image. To identify a part using the camera, tap **“Capture”** (camera icon), frame the spare part clearly in the viewfinder, and take the photo. To use an existing image, tap **“Upload”** (gallery icon) and select the photo of the part from your device. Ensure the part is centered and well-lit for best results.
4. **View Identification:** After the image is submitted, the app processes it. A progress indicator may appear briefly. The app then displays the identified part name and details (such as specifications or compatible vehicles) on-screen. If the part is not recognized, it will prompt you to try again with a different image or angle.
5. **Additional Features:** Additional features is that I have implemented an EBay and Amazon URL link that enables the Users to purchase the part they have identified.

This simple guide should enable a mechanic or technician to start using the PARTWISE app without training. By following these steps, the user can efficiently leverage the CNN-powered recognition system during repairs.

# **4.6 Conclusions**

In summary, the PARTWISE system successfully addresses the client’s need for faster and more accurate spare part identification. The key accomplishment is the integration of a CNN-based image recognition engine into a mobile app: the system demonstrated 99% accuracy on known part images, vastly reducing the time mechanics spend on manual searches. The mobile interface is intuitive, and linking to Firebase ensures secure, centralized user management. Together, these features mean that mechanics can reliably photograph or upload a part image and instantly receive the correct part name and details, fulfilling the project’s objectives.

**Limitations**

However, some limitations remain. The current model was trained on a limited dataset of four part categories, so untrained part types will not be recognized until more images are collected. The system also relies on a smartphone’s camera quality and lighting conditions; poor image quality can degrade performance. Development was constrained by the team’s skills and available resources: for example, the model uses a pretrained Efficient Net with minimal customization due to time and expertise limits. Budget constraints meant relying on free tiers of Colab and Firebase since when I tried to implement MSQL database the **build.gradel** and the **JDK** changes I had to make were incompatible or not integrating with my model’s **Gradel files** hence messing with my model accuracy in classifying images. So I just had to use the inbuilt tool Firebase for user management. So certain enterprise features (like heavy server-side processing or commercial APIs) were beyond scope. Despite these constraints, the implemented tools and technologies fully meet the essential requirements of the client, proving the concept in a realistic output for the User.

# **4.7 Recommendations**

Based on the above conclusions, several improvements are proposed for future work:

* **Expand the Dataset:** Collect a larger and more diverse set of spare-part images, including additional part categories, different makes/models, and varied lighting angles. A richer dataset will allow retraining or fine-tuning the model for higher accuracy across more parts.
* **Model Optimization:** Experiment with advanced techniques such as model quantization or pruning to reduce the TFLite model size and improve inference speed. Also consider trying other architectures (e.g. MobileNetV3 or YOLOv5 for object detection) to compare performance.
* **Offline Functionality:** Implement full offline support so that mechanics can use the app without internet connectivity. This could involve bundling necessary data and ensuring the model runs entirely on-device (which it already does, except for Firebase sync).
* **Enhanced UI/UX:** Add features like multi-language support, voice-assisted capture guidance, or real-time feedback on image quality. Improving the UI aesthetics and workflow (based on user feedback) would boost adoption and ease of use.
* **Integration with Inventory Systems:** Link PARTWISE to local spare-part inventories or suppliers’ catalogs. For example, once a part is identified, the app could show nearby suppliers or check stock levels. This would extend the tool from mere identification to a full procurement aid.
* **User Training and Feedback Loop:** Provide a brief tutorial within the app on how to take good photos (proper lighting, background) to maximize accuracy. Also, implement a feedback mechanism where users can correct a misidentified part, and those corrections feed back into future model retraining.

Implementing these recommendations would make the PARTWISE system more robust, versatile, and valuable to mechanics and the automotive repair industry.

# **REFERENCES**

References

Agarwal, R. (2022, March 12). *The 5 classification evaluation metrics every data scientist must know. Medium.* Retrieved from https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226

Andrew , Z., & Simonyan, K. (2014, September 4). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. doi:1409.1556

Geirhos, R. R. (2019). *Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.* doi:1811.12231

Goodfellow, I. B. (2016). *Deep Learning.* MIT Press.

He, K. Z. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (p. 778).

Jashila , N. M., Chin, P. L., & Kiang, L. L. (2022, July 29). VGG16-MLP: Gait Recognition with Fine-Tuned VGG-16 and Multilayer Perceptron. *Applied Sciences*.

Krizhevsky, A. S. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NeurIPS)*.

Krizhevsky, A. S. (2014). *ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems.*

LeCun, Y. B. (1988). *Gradient-based learning applied to document recognition. Proceedings of the IEEE.*

Shorten, C. &. (2019). A survey on image data augmentation for deep learning. . *Journal of Big Data*, 6(1), 60.

Srivastava, N. H. (2016). *Deep Learning* (Vol. 1). MIT press Cambridge.

Team, M. &. (2021, 12 2). *Technology Integration in Kenyan Automotive Repairs: A Case Study*. Retrieved from MobiTyres & Autoservice: https://mobityres.co.ke/

urville A., V. P. (2013). *Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence.*

Yosinski, J. C. (2014). *How transferable are features in deep neural networks? Advances in neural information processing systems.*

# **APPENDICES**

# **Instruments**



Figure 26 OBD2 Scanner Diagnostic Tool

The **OBD-II diagnostic scanner** used in this project is a handheld electronic device that interfaces with a vehicle’s onboard computer system. It connects via the standard 16-pin OBD-II port (present on virtually all vehicles manufactured after 1996) to access real-time engine and subsystem data​[csselectronics.com](https://www.csselectronics.com/pages/obd2-explained-simple-intro#:~:text=To%20do%20so%2C%20he%20will,possible%20to%20troubleshoot%20issues%20faster)​[en.wikipedia.org](https://en.wikipedia.org/wiki/On-board_diagnostics#:~:text=OBD%20systems%20give%20the%20vehicle,within%20the%20vehicle%20to%20be). When plugged in, the scanner retrieves **Diagnostic Trouble Codes (DTCs)** and live sensor readings from the engine control unit​[en.wikipedia.org](https://en.wikipedia.org/wiki/On-board_diagnostics#:~:text=OBD%20systems%20give%20the%20vehicle,within%20the%20vehicle%20to%20be). This functionality allows mechanics to quickly identify which subsystem or component is malfunctioning, making the OBD-II scanner an essential tool in modern vehicle diagnostics​[en.wikipedia.org](https://en.wikipedia.org/wiki/On-board_diagnostics#:~:text=OBD%20systems%20give%20the%20vehicle,within%20the%20vehicle%20to%20be)​[csselectronics.com](https://www.csselectronics.com/pages/obd2-explained-simple-intro#:~:text=To%20do%20so%2C%20he%20will,possible%20to%20troubleshoot%20issues%20faster).

In the context of PARTWISE, the OBD-II scanner’s relevance lies in its ability to *pinpoint fault information* that complements the image-based identification process. During field tests at Mobi Tyres Automotive, the project team demonstrated that a mechanic would first use the OBD-II scanner to diagnose a faulty component​file-urj7egpsms1afkmf1tzetx. For example, the scanner might report an engine misfire on cylinder 3 or a fault in the exhaust system. The mechanic then uses the PARTWISE mobile app to capture a photograph of the suspected part. The integrated Convolutional Neural Network (CNN) processes this image and identifies the exact spare part needed. This **two-step workflow (diagnostic code + image recognition)** enhances diagnostic accuracy: the OBD-II scan narrows the search to a specific subsystem, and the image analysis confirms the correct part within that subsystem​file-urj7egpsms1afkmf1tzetx. A typical workflow is as follows:

* **Connect and Scan:** The mechanic connects the OBD-II scanner to the vehicle’s diagnostic port and retrieves error codes and related data (e.g., engine or emission system status)​[csselectronics.com](https://www.csselectronics.com/pages/obd2-explained-simple-intro#:~:text=To%20do%20so%2C%20he%20will,possible%20to%20troubleshoot%20issues%20faster).
* **Image Capture:** The mechanic photographs the component indicated by the diagnostic code using the PARTWISE mobile app.
* **Image Analysis:** The PARTWISE CNN model analyzes the image to identify the exact spare part required, cross-referencing the diagnostic information for verification.
* **Recommendation:** The system presents the mechanic with the identified part name, specifications, and ordering information, thereby streamlining the repair process​file-urj7egpsms1afkmf1tzetx.

The OBD-II scanner was selected for integration with PARTWISE due to several practical factors:

* **Availability and Cost:** OBD-II scanners are widely available and affordable. Many are built on low-cost ELM327 chipsets and come with Bluetooth or Wi-Fi connectivity, allowing easy pairing with Android/iOS devices​[csselectronics.com](https://www.csselectronics.com/pages/obd2-explained-simple-intro#:~:text=That%20is%20your%20car%20telling,scanner%20to%20diagnose%20the%20issue). This means the PARTWISE app can run on standard smartphones or tablets already used in the workshop.
* **Standardized Diagnostics:** Since the mid-1990s, most vehicles support OBD-II diagnostics, ensuring broad compatibility across different makes and models​[csselectronics.com](https://www.csselectronics.com/pages/obd2-explained-simple-intro#:~:text=,in%20USA%20for%20cars%2Flight%20trucks)​[en.wikipedia.org](https://en.wikipedia.org/wiki/On-board_diagnostics#:~:text=OBD%20systems%20give%20the%20vehicle,within%20the%20vehicle%20to%20be). The standardized error codes provide a reliable starting point for identifying the faulty subsystem.
* **Enhanced Accuracy:** By combining code-based diagnosis with visual recognition, the integrated system reduces false matches. The OBD-II scan focuses the search to a particular area of the vehicle, so the image recognition model only needs to differentiate between parts within that context, improving speed and accuracy.
* **Operational Readiness:** MobiTyres Automotive staff were already familiar with using OBD-II scanners during vehicle servicing​file-urj7egpsms1afkmf1tzetx. Their willingness to allow the project team to use the scanner on-site demonstrated organizational support. Leveraging an existing tool minimizes training overhead and fits naturally into the mechanics’ workflow.

In summary, the OBD-II scanner serves as a crucial instrument in the PARTWISE system. Its real-time diagnostics **complement** the CNN-based image recognition by reducing ambiguity in part identification. This hybrid approach was validated in the field, and its selection reflects a balance of technical capability and practical deployment considerations.

**Appendix B: Letter of Introduction**

Francis W. Iraki  
School of Computing and Information Technology  
Jomo Kenyatta University of Agriculture and Technology (JKUAT)  
P.O. Box 62000–00200, Nairobi, Kenya  
Email: francis.iraki@jkuat.ac.ke

Date: April 5, 2025

To: The Branch Manager,   
Mobi Tyres Automotives  
Nairobi, Kenya

Subject: Request for Permission to Conduct Onsite Research Activities

Dear Sir/Madam,

I am a final-year Bachelor of Science in Information Technology student at Jomo Kenyatta University of Agriculture and Technology (JKUAT). I am conducting a research project titled *“CNN Model Integrated with Mobile Technology for Vehicle Parts Identification (PARTWISE)”* under the supervision of the Department of Information Technology. The aim of this study is to develop a mobile application that assists mechanics in accurately identifying automobile spare parts using image recognition and diagnostic data.

In order to ensure that the PARTWISE system meets real-world needs, I kindly request your permission to conduct research activities at the Mobi Tyres Automotives branch. Specifically, I seek to: (1) perform scheduled onsite visits to observe workshop operations; (2) conduct interviews with mechanics and your branch manager about current spare-part identification methods and challenges; (3) collect non-confidential data on vehicle repair workflows; and (4) demonstrate and evaluate the PARTWISE prototype within your facility. All data collected will be used solely for academic research and system evaluation purposes, and will be treated with strict confidentiality. No commercially sensitive information will be disclosed.

These activities are planned to take place over the next few weeks, with each visit lasting approximately 2–3 hours at a mutually convenient time. The insights gained from your staff’s experience will be invaluable in refining the PARTWISE application to be both practical and user-friendly. After the study, I will be happy to share a summary of findings and any recommendations with Mobi Tyres Automotives.

I greatly appreciate your consideration of this request. Please feel free to contact me at francis.iraki@jkuat.ac.ke or by phone at +254 757733364 if you have any questions or require further information. Thank you for supporting academic research and innovation in the automotive service industry.

Yours faithfully,

Francis W. Iraki  
B.Sc. Information Technology, JKUAT

**Appendix C: Interview Transcripts and Questionnaires**

**Mechanic Interview Transcript**

**Interviewee:** Senior Mechanic (Mobi Tyres Automotive)   
**Interviewer:** Francis W. Iraki, Researcher (JKUAT)

**Interviewer:** Thank you for taking the time to speak with me. To start, could you describe how you currently identify the correct spare parts needed for a repair job?  
**Mechanic:** Sure. Typically, I first look for the part number on the old or damaged component. If the number is visible, I compare it with the workshop’s parts catalog or enter it into our computer system to find a matching new part. If the number is worn off or missing, I rely on the car’s manual or online parts database. Sometimes I take photos of the part and send them to suppliers for confirmation.

**Interviewer:** What challenges do you encounter during this identification process?  
**Mechanic:** One big challenge is when parts are very similar or have small variations. For example, two engines might use pistons that look almost alike but have different dimensions. Another issue is the part numbers themselves – they can be hard to read or they might have changed with newer models. Also, not all catalogs are up to date; some older or local car models don’t have digital listings, so it takes extra time to confirm.

**Interviewer:** Do you use any diagnostic tools or technology to help with part identification?  
**Mechanic:** We use an OBD-II scanner regularly to read engine codes when a check-engine light is on. The code tells us which subsystem has a fault (like misfire in cylinder 2 or an oxygen sensor issue). This narrows down where to look. However, even if we know, say, a sensor is bad, we still have to visually inspect or remove it to get the exact part details. We don’t have any system that recognizes parts just from a photo – we rely on our training, catalogs, and sometimes Google image searches.

**Interviewer:** How often do misidentification errors occur, and what happens when they do?  
**Mechanic:** It happens fairly often – maybe a few times a month we end up ordering the wrong part the first time. When that happens, we have to return or exchange it, which delays the repair and can cost the garage extra money. It’s frustrating for both us and the customer. For instance, we once mixed up two types of brake pads that fit different vehicle models, so we wasted half a day fixing that mistake.

**Interviewer:** How would a tool that identifies parts from photos help you in your work?  
**Mechanic:** That would be very helpful. If I could just take a clear picture of the part and have an app tell me exactly what it is, it would save a lot of time. Combining that with the OBD code – say the scanner says “fuel injector fault” and the app confirms “fuel injector model X123” from the photo – would make it much more reliable. It would reduce guesswork and speed up getting the right part.

**Interviewer:** Thank you for your insights. This information will greatly inform the development of our system.

**Branch Manager Interview Transcript**

**Interviewee:** Branch Manager (Mobi Tyres Automotive)   
**Interviewer:** Francis W. Iraki, Researcher (JKUAT)

**Interviewer:** Thank you for meeting with me. From a managerial standpoint, what are the key challenges you see with spare-part identification in the workshop?  
**Branch Manager:** The main issue is the delay caused by incorrect or unclear part identification. When mechanics order the wrong part, it not only delays that job but also disrupts the workflow and inventory tracking. We often see customers waiting longer than expected, which affects satisfaction. Getting the right part the first time is critical for efficiency.

**Interviewer:** What systems or resources does Mobi Tyres currently use to identify parts?  
**Branch Manager:** We have a mix of tools. First, we use printed parts manuals for common vehicles, which mechanics consult in the shop. We also have a digital parts database and some online catalog subscriptions. For diagnostics, every bay has an OBD-II scanner that mechanics use to read trouble codes. Additionally, our suppliers can receive part photos or take phone calls if we need help identifying something.

**Interviewer:** In your experience, what kinds of errors or complications arise frequently in this process?  
**Branch Manager:** One frequent problem is outdated information. Sometimes the manuals or catalogs we have don’t reflect the latest models or local variants. Another issue is when a vehicle has been modified or has third-party parts; then the numbering doesn’t match any official catalog. Also, communication gaps occur: a mechanic may describe a part one way, and the parts clerk or supplier interprets it differently. All of these can lead to ordering delays or returns.

**Interviewer:** How do these identification challenges impact the business?  
**Branch Manager:** There’s both a time and cost impact. Reordering parts means paying twice (plus return shipping), and it reduces our throughput. For example, if four jobs are scheduled and one is held up waiting for a part fix, it might push work to the next day. It also affects customer trust if delays happen regularly. On the other hand, if we get it right quickly, we can do more jobs per day and keep customers happy.

**Interviewer:** Would a mobile application that uses image recognition to identify parts be valuable to your operation?  
**Branch Manager:** Potentially, yes. If the app is accurate, it could cut down the mistakes. We like tools that assist our skilled mechanics rather than replace them. I would be interested in trying something like PARTWISE to see how it fits. Of course, it would need to integrate smoothly with our current processes, and the team would need some training. But I agree that matching a part photo with the diagnostic code and company database could really speed things up.

**Interviewer:** Thank you for your time and detailed responses.

**Structured Questionnaire**

1. How do you currently identify and verify the correct spare parts needed for a repair job?
2. What challenges or delays do you experience with the existing spare-part identification process?
3. Which tools or resources (e.g., physical catalogs, digital databases, diagnostic devices) do you use to assist in parts identification?
4. How frequently do issues like missing part numbers, unclear labeling, or mismatched parts occur during repairs?
5. What impact do these identification issues have on workshop efficiency, cost, and customer satisfaction?
6. Are there particular types of parts or vehicle systems that are especially difficult to identify? Why?
7. Would a mobile app that identifies parts from a photograph (possibly combined with diagnostic data) be useful in your workflow? Please explain.
8. Do you have any suggestions for improving the spare-part identification process in the workshop?



Figure 27 Engaging with the Branch Manager



Figure 28 Engaging with The Mechanics



Figure 29 Going through the car diagnostic process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Duration** | **Start Date** | **Actual Date** | **End Date** | **Deliverable** |
| |  | | --- | | Project Kickoff |  |  | | --- | |  | | 3 days | 2/10/2024 | 2/10/2024 | 4/10/2024 | Project Charter |
| Literature Review | 3 weeks | 07/10/2024 | 07/10/2024 | 25/10/2024 | Literature Review Report |
| Requirements Gathering | 2 weeks | 28/10/2024 | 28/10/2024 | 08/11/2024 | Requirement Specification |
| System Design | 3 weeks | 11/11/2024 | 11/11/2024 | 29/11/2024 | System Design Document |
| Data collection and preparation | 4 weeks | 02/12/2024 | 02/12/2024 | 27/12/2024 | A clean, labeled dataset |
| Model Development (CNN) | 3 weeks | 30/12/2024 | 30/12/2024 | 17/01/2025 | A trained and tested CNN model |
| Mobile App Development | 4 weeks | 20/01/2025 | 20/01/2025 | 14/02/2025 | A fully functional mobile application |
| Integration and Testing | 2 weeks | 17/02/2025 | 17/02/2025 | 28/02/2025 | Integration Report |
| Deployment (Mobile + Cloud) | 2 weeks | 03/03/2025 | 03/03/2025 | 14/03/2025 | Partwise Accelerated repair |
| Document Finalization | 1 week | 17/03/2025 | 17/03/2025 | 21/03/2025 | Final Documentation |
| Project Closure | 1 week | 24/03/2024 | 24/03/2024 | 28/03/2025 | Project Closure Report |

Table 1.1 Project Schedule

**PROPOSED BUDGET**

|  |  |  |
| --- | --- | --- |
| Item | Cost (KES) | Justification |
| Internet and Communication | 2500 | For accessing online research material, cloud services, and communication with stakeholders. |
| Printing and Stationer | 1500 | For reports and research-related documents. |
| Data Collection and preparation | 2000 | For ensuring model accuracy and reliability. |
| Travel cost | 2000 | For visiting different garage and seeing the type of diagnostic equipment used. |
| Miscellaneous Expenses | 3500 | To cover unforeseen expenses (e.g., extra testing, additional resources, emergency travel). |
| **Subtotal** | 11,500 |  |

Table 1.2 Proposed Budget

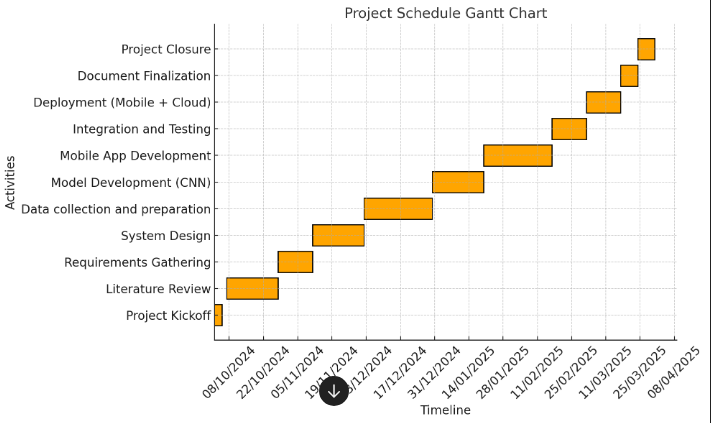


Figure 30 Gantt chart