INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER, UK

FixMe:
Severe Vehicle Damage Identification and Damage Cost Estimation System

A dissertation by

W.M.T.P.Wanninayake

Supervised by

Torin Wirasingha

Submitted in partial fulfillment of the requirements for the
BSc (Hons) in Computer Science Degree
Department of Computing

May 2016

Declaration

I hereby certify that this project report and all the artefacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any degree programme.

Full Name of Student	: Wanninayake Mudhiyanselage Tharindu Prabhashwara Wanninayake
IIT Registration No	: 2015294
UOW Registration No	: w1608486

Signature : .... Date : ....

Abstract

According to the Ministry of Transport & Civil Aviation in Sri Lanka, vehicle collisions per year in Sri Lanka is around 5000 (Ministry of transport, 2018). After a collision, vehicle owners spend lot of time and money to get the vehicle repaired. Sometimes, damage can be too high to repair the vehicles. This is where the Vehicle insurance companies come.

Vehicle insurance companies offer insurance for vehicles. They provide financial protection against physical damage and liabilities that could arise from the collision. Today, in the car insurance industry, a lot of money is wasted due to claims leakage. (Ernst & Young LLP,2015) As a solution, insurance agencies send their agents to accident locations to evaluate the vehicle damage at the location itself.

This whole process takes lot of time and energy from both the customer and the insurance company. And also, the traffic situation made by the collision takes the time from the people who are travelling through the road. Author believes by reducing the time for the evaluation process of the damage can become helpful in reducing the time that spent on the road after a collision happen. In this document author address a method to evaluate the vehicle damage without going to the collision location by developing a system for insurance agencies that can validate the damage using image processing technology.

Keywords- Image Processing, Vehicle Damage Classification, Deep Learning

Acknowledgement

Lot of effort and dedication was used for the completion of this project. My supervisor, Mr. Torin Wirasingha gave lot of feedback and guidance to overcome many obstacles that I faced during the development of this project. It would not have been possible without the support Mr. Torin Wirasingha and I would like to thank my supervisor for providing me guidance during this project. I would also like to thank Mr. Kaneeka Vidanage, who helped us understand the concepts of a research project. Last, but not least, I would like to thank my parents for their continuous support.

Table of Contents

Table of Figures

List of tables

List of Abbreviations

Abbreviation	Definition
CAD	computer-aided design
SIFT	Scale-invariant feature transform
ORB	Oriented FAST and rotated BRIEF
SURF	speeded up robust features
CAN	Controller Area Network
PRNN	Pattern Recognition Neural Network
CNN	convolutional neural network
CAE	Convolutional Auto Encoders
API	Application programming interface
UI	User Interface
PRINCE	PRojects IN Controlled Environments

Chapter 1: Introduction

1.1. Chapter Overview

This chapter lays the foundation of the project background in order to give a clear understanding to the reader about the problem domain of the project. Later on, this chapter describes the project Aim, Scope, objectives and the motivation.

1.2. Project Background

According to the Ministry of Transport & Civil Aviation in Sri Lanka, vehicle collisions per year in Sri Lanka is around 5000 (Ministry of transport, 2018). After a collision, vehicle owners spend lot of time and money to get the vehicle repaired. Sometimes, damage can be too high to repair the vehicles. This is where the Vehicle insurance companies come.

Vehicle insurance companies offer insurance for vehicles. They provide financial protection against physical damage and liabilities that could arise from the collision. Today, in the car insurance industry, a lot of money is wasted due to claims leakage. (Ernst & Young LLP,2015) As a solution, insurance agencies send their agents to accident locations to estimate the damage at the location itself.

Mila Araujo(Araujo, 2018) explains the insurance claiming process. With that information, insurance calming process can be divided in to 3 main steps.

1. Informing the insurance company about the collision that happened.
   After the collision, Client needs to inform about the accident to the insurance company. Then an insurance agent will come to assist the vehicle owner to the location. Agent needs to confirm that a damage has happened and give a damage row estimation for the vehicle.
2. Taking the damage vehicle to a repair center.
   After getting a damage estimation to the vehicle, Client needs to take the damaged vehicle to a repair shop.
3. Repairing process of the vehicle.
   Repair shop also gives a damage estimation and insurance company needs to confirm that both estimations are similar, to start with repairing the vehicle.

This whole process takes lot of time and energy from both the customer and the insurance company. And, when a traffic collision happens lot of people that are in the road get affected by this. In this document author address a method to get faster with insurance calming process by automating the damage estimation process.

1.3. Aim

To reduce the time that spent on the street after a vehicle collision by developing a system that can validate the damage and estimating a cost for that.

1.4. Motivation

The author has witnessed the hassle people have to go through in the event of an accident on many occasions. The time that an evaluation officer from the respective insurance corporations takes is also a contributing problem. By implementing a system that can automate the process of the damage estimation, the author sees that waiting time and the traffic made by the accidents can be reduced.

1.5. Scope

Estimating severe damages of a vehicle from images can be hard due to its nature of being mixed with dents and scratches. The Project address a method to estimates severe damages. Research gap is to properly identify the severe damages that took in a vehicle from an image and estimate its cost.

1.6. Objectives

Objective Id	Objective Description
Obj 01	To identify damage severity levels in a damaged vehicle.
Obj 02	To analyze severity levels and find out the most accurate severity of the damage.
Obj 03	Generate a minimum damage estimation for severe damages.
Obj 04	To increase the accuracy of identifying severe vehicle damages from images.
Obj 05	To implement a system that can estimate severe damages in a vehicle from images and generate cost estimation for the severe damages.

Table 1: List of Objectives of the Project FixMe

1.7. Rich Picture of the Proposed Solution

Rich picture of the proposed solution that is going to be discussed throughout this project.

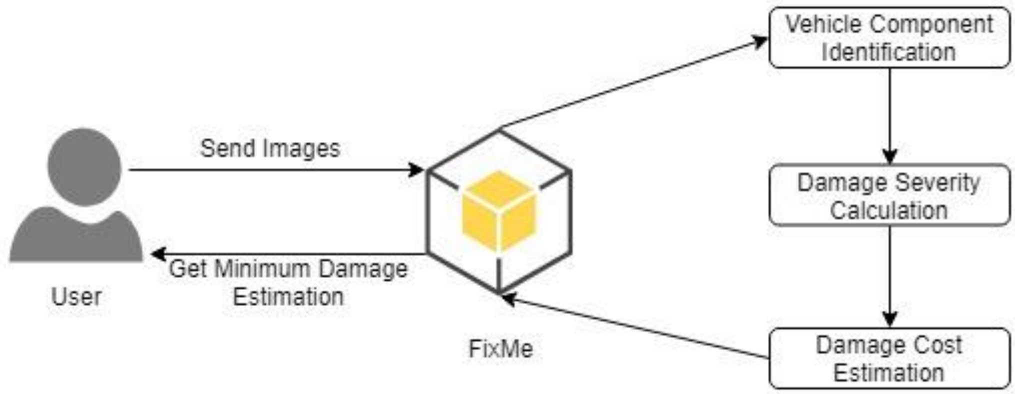


Figure 1: Rich Picture of the Proposed Solution

1.8. Outline of the Thesis Chapters

Chapter 1: Introduction	Discuss about the project background.
Chapter 2: Literature Review	Provides a review of past literature and existing system.
Chapter 3: Project Management	Discuss about the details about project management aspect of the project.
Chapter 4: Requirement Specifications	Discuss about the system requirements for the application.
Chapter 5: System Design Methodology	Discuss about the designing aspect of the provided solution.
Chapter 6: Implementation	Discuss about the implementation process of the provided solution.
Chapter 7: Testing	Discuss about the testing aspect of the prototype.
Chapter 8: Evaluation	Discuss about the evaluation of the prototype and the project process.
Chapter 9: Conclusion	Conclusion on this project.
Chapter 10: References	References.

Table 2:Out line of the Thesis Chapters

Chapter 2: Literature Review

2.1. Chapter Overview

Previous chapter discussed about the problem domain of the project. This chapter describes the past published literature and existing systems that are related to the problem domain. Beginning of the chapter explains about the types of the vehicle damages and why author choose to estimate vehicle damage. Reviewing of the past literature & algorithms will be discussed after that.

2.2. What Are the Types of Vehicle Damage?

When a damage happened to a vehicle, it deforms the appearance of it. Even though there are not any universal deformation types to vehicles, With the information provided by *Libertymutual.com* (Anon, n.d.), deformation of a vehicle can be divided in to categories.

a. Minor Damage - scratches, scrapes or dings. For example, a cracked headlight or small dent in your hood.
b. Moderate damage - Large dents in the hood, fender or door of your car. If the doors won't open, or if airbags have deployed, you likely have a moderate amount of damage to your car.
c. Severe damage - very heavy damage to the vehicle body. Broken axles and bent or twisted frames. Mixture of dents and Scratches.
d. Critical damage- In a Critical damage situation, air bags have almost always deployed. Examples include if your car has rolled over, or an internal damage was happened to the car.

There are current researches to identify the miner and moderate damages through the image processing technology. But there is no research or a product to identify severe and critical damages that took in place in a vehicle.

2.3. Why Estimate Vehicle Damage?

A lot of money is wasted due to claims leakages in the car insurance industry (Ernst & Young LLP,2015). Claims leakage is the difference between the actual claim payment that have been made and the amount that should have been paid. To reduce such effects, insurance companies use visual inspection and validation. Due to this reason, insurance companies send their agents to the collision location to estimate a value for the damage. Owners of the damaged vehicle need to wait for the agent to arrive and it creates a traffic situation on the road. If this process can be automated, lot of time and money for both the customer and the insurance company can be saved.

2.4. Reviewing Past Literature & Algorithms

This selection will be discussing about the past similar approaches to address the problem. Author will be first discussing about the similar projects that was done in the problem domain due to current existing systems does not provide enough information about their approaches.

2.4.1. Image Based Automatic Vehicle Damage Detection

The PhD thesis of Jayawardena (Jayawardena, 2013) aims directly at automating vehicle damage detection through images. According to Srimal Jayawardena (2013), using predefined 3D computer-aided design (CAD) models of undamaged vehicles and comparing them with images of damaged vehicles can help to identify the damage details through an image.

Jayawardeana(2013) uses 3D computer-aided design models (CAD) to get ground truth information about the vehicles that have not been damaged and uses a 3D pose estimation algorithm to register them in the database. After registering them, Jayawardena compares them with images from damaged vehicle and extract the deformation details from it.

3D pose estimation

After getting an image of a damaged vehicle, Jayawardeana(2013) registers a predefined 3D computer aided design (CAD) model of that vehicle over the image. It identifies the differences in the vehicle and shows how it would look like if it were undamaged. The final 3D pose is made by calculating the distance and minimizing it between the image and the 2D projection of the 3D model.

 *Figure 2: initial pose*	 *Figure 3: Final Pose*

Table 3: 3D pose estimation procedure developed by Jayawardena (2013) , Before and After optimization

3D segmentation

The second step is to use recovered 3d pose to identify the vehicle components. Jayawardena (2013) uses image segmentation to identify vehicle components. Image segmentation is the process of dividing a digital image into multiple segments. By using segmentation, it will simplify an image into something that is more meaningful and easier to analyze. After 3d pose estimation jayawardena uses 3D segmentation in to the image. Result is shown in Figure 3 with a car divided in to sub segments.

Figure 4: The figure shows '3D Model Assisted Segmentation' in Jayawardena's project

Reflection Detection

Jayawardena's (2013) final step is to compare the identified components of the vehicle with the 3D CAD model. False image edges can be included in the image due to the reflective vehicle's bodies. Image edges which are not in the 3D CAD model can be considered as damage. Jayawardena applies multi view geometry techniques on two images of the damaged area from two different viewpoints to overcome this problem. By improving the details in the 3D CAD model, Jayawardena (2013) was able to improve the accuracy of the system and reduce the miscalculated damages. Vehicle bodies are reflective and due to this, reflection in the image can be also miscalculated as damage Jayawardena's (2013) developed a prototype that led to Controlexpert's EasyClaim app (Controlexpert, 2015). More details of the product EasyClaim will be discussed under "Existing Products" topic.

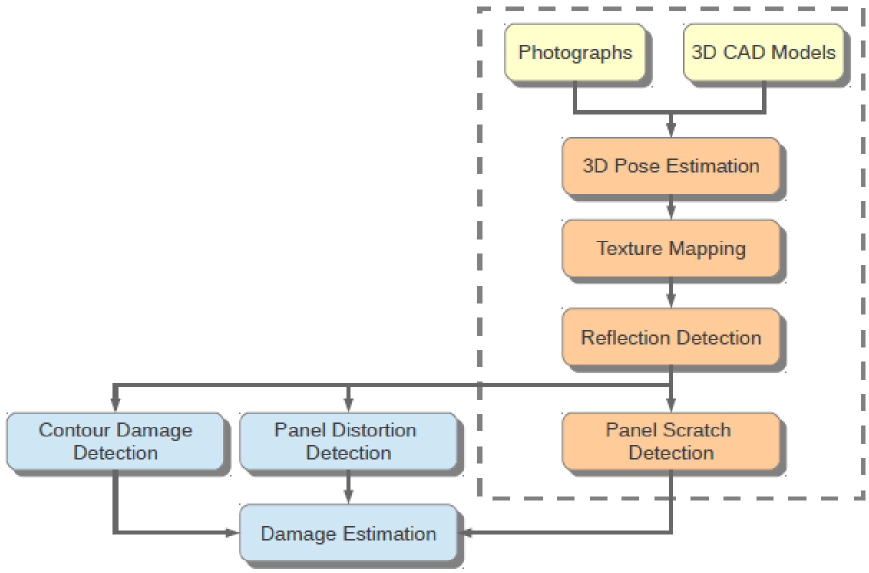


Figure 5:The technical flow of the system of Jayawardena's (2013) research

2.4.2. Image Processing Based Severity and Cost Prediction of Damages in the Vehicle Body

Rukshala Harshani (Harshani, 2017) uses Scale-invariant feature transform (SIFT) techniques to extract the deformation details from images of a damaged vehicle and predicts minor damages in a vehicle body.

Using SIFT, Harshani (2017) extracts the local features from an image and she uses it to divide the damage of the vehicle in to different severity classes. These severity classes take a crucial part in damage estimation process. Harshani uses a set of cost rules to produce the damage cost. These cost rules are conditional statements and it has several parameters such as severity of the damage, vehicle type, vehicle model and year of manufacture.

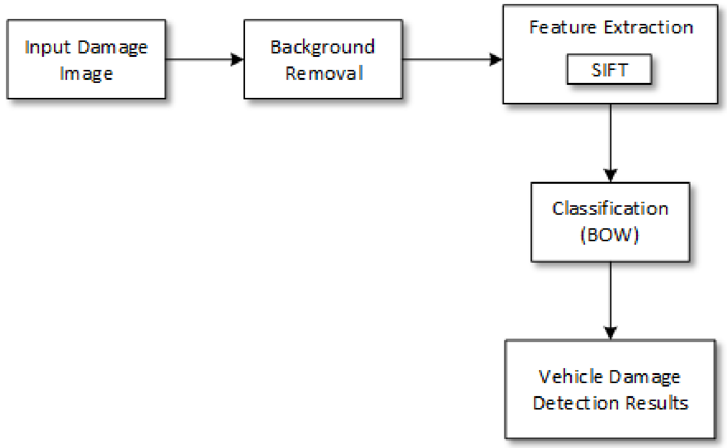


Figure 6:The technical flow of the system of Harshani's (harshani,2013) research.

SIFT (Scale-invariant feature transform)

There are couple of algorithems that can use for feature extraction and they are ORB(Oriented FAST and rotated BRIEF), SIFT(Scale-invariant feature transform) and SURF(speeded up robust features). According to Ebrahim Karami, Siva Prasad, and Mohamed Shehata (Karami, Prasad and Shehata, 2016), SIFT performs the best in most scenarios.

They showed that ORB is the fastest algorithm while SIFT performs the best in the most scenarios. For the special cases when the angle of rotation is proportional to 90 degrees, ORB and SURF outperforms SIFT. ORB and SIFT show almost similar performances in the noisy images, In ORB, the features are mostly concentrated in objects at the center of the image while in SURF, SIFT and FAST key point detectors are distributed over the image. Research also show that SIFT algorithm has very high accuracy comparatively to the SURF and also faster than the SURF when working on the Realtime applications. Due to those reasons using SIFT is the best for feature extraction.

2.4.3. Minor Damage Identification in Vehicle Bodies Using Adaptive Sensor Data Processing

Group of researchers Sergei Gontscharova, Hauke Baumgärtela, Andre Kneifela and Karl-Ludwig Kriegera(Gontscharova, Baumgärtela, Kneifela and Kriegera, 2014) worked on fully-automatic detection and recognition of minor vehicle body damages using acoustical noise level of the vehicle. The main goal of the research project is to identify external structural damages to the vehicle body, their type and severity from structure borne sound analysis.

Sounds are taken by specially developed sensor nodes that are attached to the car parts. When a damage happened to the vehicle body, the system classifies the structural damage type and collects data about the damage such as damaged component and severity of the damage. Then an embedded sensor node network based on Controller Area Network (CAN) communicates the data to a central data processing unit and calculates the damage. After calculating the damage, it will be notified to the user through a mobile application.

To get the sound data, researchers use piezoelectric sensors on the inside of the vehicle body. It is for measuring vibrations and damages. Sensors will generate a voltage relative to the mechanical vibrations in the vehicle body. These analog signals will be digitalized and use to process the damage based on the implemented algorithms. To evaluate the damage, researchers use two algorithm blocks. First to calculate relevant signal feature and then to classify detected events in to damage classes “dent” and Scratch”.

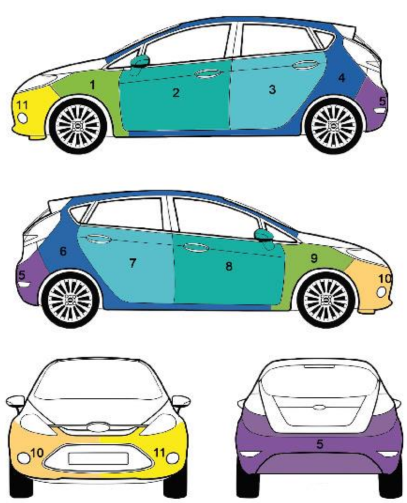


Figure 7: Sensor node positions used on the vehicle in research "Algorithm development for minor damage identification in vehicle”

Pattern Recognition Neural Network

Pattern Recognition is a recognition algorithm that used to recognize patterns and regularities in data. After getting the data from the sensor node, data will be available to a probability based PRNN classifier (Pattern Recognition Neural Network). The duration of the damage sound and the sound that makes will be taken to classify the damage type and damage.

Problems that might arise from this method is un necessary sounds such as rain and sound waves of passing vehicles can be mistaken as damage. To overcome this problem researchers have used an already existing method that was found by Baumgärtel H, Gontscharov S, Kneifel S and Krieger K-L.(2014) A research about detection of dents in vehicle bodies by sound emission analysis

In order to get a qualified damage estimation from this method, the sensor nodes must be trained individually. And also to get the best use of PRNN classification process, lot of training data captured from real time signals is required. Researchers state that “In practice, a full coverage is not attainable” due to the fact that a sensor node can capture only a 360 area. And also, there can be miscalculated damage caused by car noise and car structure noise. Authors state that is possible to detect damages on the entire vehicle body and reduce the miscalculated damage with the help of specific algorithms. And they are yet to be implemented. Overall to implement such a system needs sound analyzing sensors and a larger set of training data results in a higher probability and sensors needed. To authors point, it is not practical to implement this method because the sound of the damage one car making is different from another car brand. Therefore, will have to make algorithms for every car model and brand.

2.4.4. Deep Learning Based Car Damage Classification

Kalpesh Patil, Mandar Kulkarni and Shirish Karande (Patil, Kulkarni, Karande, 2016) researched about using convolutional neural network (CNN), domain-specific pre-training followed by fine tuning and transfer learning to help with the automation of insurance processing.

Convolutional Neural Network

CNN is a class of deep neural networks and it is most commonly applied to analyzing visual imageries. Firstly, authors trained the CNN with random initialization and compared with an augmented data initialization. Table 2 shows the results they got and it can be seen that data augmentation is needed to improve the performance and generalization.

Convolutional Autoencoder

"Autoencoders are a family of neural networks for which the input is same as the output. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation." (Chablani, 2017)

Convolutional AutoEncoders (CAE) provides a better alternative because of a smaller number of parameters they use due to low amount of connections and weight sharing (Masci, Meier, Cires, Schmidhuber, 2011). To train the CAE, Authors used unlabeled image set and used it with CNN and results are shown in the table 2.

Method	Without Augmentation			With Augmentation		
	Acc	Prec	Recall	Acc	Prec	Recall
CNN	71.33	63.27	52.5	72.46	64.03	61.01
AE-CNN	73.43	67.21	55.32	72.30	63.69	59.48

Figure 8:Test accuracy with CNN training and (CAE + finetuning) mentioned in "Deep Learning Based Car Damage Classification" Research paper

Transfer Learning

Machine Learning Mastery (machinelearningmastery,2018) gives a gentle introduction to Transfer Learning as "A machine learning method where a model developed for a task is used as the starting point of another task". When using Transfer Learning it is possible that some data can be specific for individual domains and some data can be common between different domains. This might help to improve the performance for the current ongoing task. In cases where the selected domain and current ongoing task domain are not related, a force transfer can be unsuccessful and that can lead to low quality performance.

To overcome this problem authors used convolutional neural network models that are trained on the Imagenet dataset. They say that, imagenet dataset contains car as an object and because of that they expect the transfer to be useful. They validate that by experimenting with multiple pretrained models. Fig. 7 shows the transfer learning setups that researchers used.

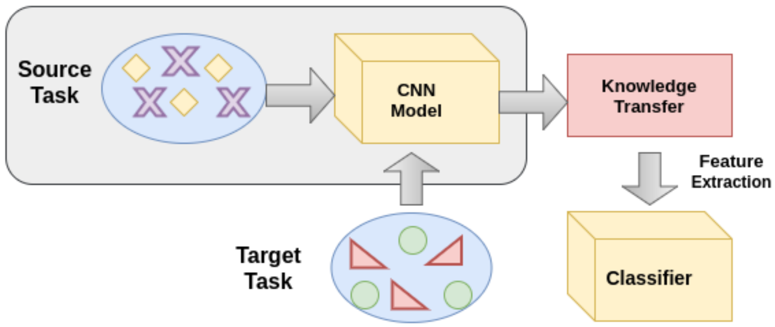


Figure 9: Transfer learning steps used in research "Deep Learning Based Car Damage Classification"

Most of the supervised methods need large amount of data and resources. Unsupervised techniques such as Autoencoders proved to improve the generalization performance of the classifier in case of small number of labeled samples. For images, author can see that convolutional AutoEncoders (CAE) have shown promising results. (Table 2) Researchers also point out that "car specific features do not affect in damage classification and because of that it points out the superiority of features that represent in the image learned from the lager training set."

2.5. Existing Systems

With the developments of the image processing technology, some insurance companies offer express insurance claim options by allowing customers to upload pictures of the damaged vehicles taken by mobile devices. Below there are couple of companies that use mobile express options for to damage calming.

2.5.1. American-Autobody (American-autobody.com, 2018)

American Autobody is a vehicle repair company and it allows people to use a mobile app to contact them and inform about damaged vehicles to start with the repairing process. Users of this mobile app can fill their contact details and vehicle info, then take photos of the damaged vehicle and upload them to their company site. Users can also upload and estimation if they got one from their insurance agency and one of the agents in American Autobody will contact them with their estimate of the damage and necessary details to repair the vehicle. If the both parties agree upon the damage estimate, American-Autobody will start repairing process of the vehicle.

To get more accurate results, the images will be sent to their damage estimation team. And they will assist the user with damage estimation. (Bodyshop.systems, 2018) And to get more details and clear accurate pics about damaged vehicles, mobile app guides the user through precise photo angles.

2.5.2. EasyClaim-ControlExpert (ControlExpert, n.d.))

Easy Claim is a mobile app developed by Control Expert to estimate vehicle damages.

After reporting about the collision to the insurance company, the customer will get a link via e-mail or SMS to the Easy Claim application. It can be opened any mobile device with internet access. If the claim is applicable, customer need to take photos of their vehicle registration certificate and photo of the vehicle dashboard displaying mileage information. This information is to register the vehicle with damages. Then customer is needed to take pictures of the damaged car and add a short description of how the accident circumstances occurred. Finally, user can submit the information and within few hours the claimant receives a calculation of the repair costs including spare parts, paintwork and labor costs. Users can choose to get the vehicle repair or receive a payout for the damage. App offers the feature to arrange appointments for the nearby repair centers.

2.5.3. Automotive-Estimate (Gocanvas, 2018)

Automotive Estimate mobile app is one of Gocanvas apps specifically designed to vehicle repairs. This application is used by auto repair shops, body shops and auto repair centers. App provides a professional estimate that can be submitted to an insurance company or given to customers.

This application is a management tool that can capture auto repairs specifics such as owner information and vehicle details, and summarize the labor, parts, quantities, costs and more needed to complete the repair. One of the key benefits using this application is to estimate vehicle repair costs before beginning the job.

2.6. Conclusion of Existing Systems

Automotive Estimate from the Gocanvas is a good management tool for damage estimation. In order to use the app, user need to have a good knowledge about the damage that happened to the vehicle and prices for those damages. This application lacks the ability to identify damages and because of that user needs to be near the vehicle in order to calculate the damage. Due to this reason, insurance company or the repair man need to go to the collision location.

Both the American Autobody and Easy Claim by Control Expert use images to identify damages. By give the user the ability to upload pictures from their mobile, they have reduced the time and money cost of travelling. Both the companies use image processing technology to identify damages. American Autobody uses x-ray filters to get a better look of the damages while a team supporting to estimate that damages by looking at them. (Bodyshop.systems, 2018) Control Expert uses the technology founded by Srimal Jayawardena (Jayawardena, 2013) to identify minor damages and a team to estimate the damages.

The above-mentioned products that can identify damages only have minor damage identification capabilities and have human assistance for to estimate the damage. This makes the application a semi-automatic application. And, above products lacks the ability to identify and estimate severe damages. This makes an opportunity to develop an application that can identify and estimate value for the severe damages.

2.7. Research Gap

There are current researches to identify the miner and moderate damages in a vehicle body. But there is no research or products to identify severe damage in a vehicle body. Severe damage is a combination of moderate and minor damages in a vehicle body and beyond. Research gap is to properly identify severe damages, improve the accuracy of identyfing the damage levels and deliver a minimum damage estimation.

Choosing a Suitable Technology

According to the part literature and reviews, there are two ways to provide a solution for current existing problem.

- Sensor technology
- Image processing technology

Using sensor technology can be cost highly (depends on the sensors) and not suitable for large scale projects. Attaching the sensors to every car to identify damage is not possible due to the number of vehicles that are currently in the market therefore it is not practical. It is a limitation for sensor based approch. (Gontscharova, Baumgärtela, Kneifela and Kriegera, 2014)

Image processing approach is not only can overcome that limitation easily, but also can overcome the practicality due to the increase usage of modern phones with camera capabilities (Parasuraman et al., 2017). And, most of the past literature to this approach provides a solid foundation to conduct a good research and provide a solution for the problem. Due to these reasons, image processing technology was selected as the approach of this project.

2.8. Chapter Summary

Insurance companies use visual inspection and validation to reduce claims leakages in the car insurance industry (Ernst & Young LLP,2015). Visual inspection and validation are currently carried out by insurance company agents and they need to travel to the collision location to estimate a value for the damage. The time for the agent to arrive is based on the cost, distance and traffic along the route. If this process can be automated, lot of time and money for both the customer and the insurance company can be saved.

With the help of modern technology, many have come up with the idea to identify and estimate costs for vehicle damages. Many of them were only attempting to identifying minor and moderate damages since severe damages cannot be easily captured using an image. Since it is important to identify and estimate values for severe damages, it is choose as the research gap and this research attempts to identify possible methods that could be used to achieve this using image processing.

Chapter 3: Project Management

3.1 Chapter Overview

A project is highly likely to fail without a proper project management process. This chapter discusses about the project management aspect of the project.

3.2 Project Management Methodology

Project Management Methodology is to plan and guide the progress of the project from beginning to the end. It is widely used to control the complexity of the project development process. Project Management Institute divides the process in to five stages. (Project Management Institute, 2015))

1. Initiation
2. Planning
3. Executing
4. Controlling
5. Closing

One of the widely known project management mythology is PRINCE2(PRojects IN Controlled Environments) methodology. It is a process-based method for effective project management. It is widely recognized and practiced in the private sector in the UK and internationally (PRINCE2, 2015). It provides a good framework to manage projects. To manage the management aspect of the project, author decided to use the PRINCE2 methodology. Below figure 9 explains the main phases of PRINCE2 methodology.

Figure 10: Phases of PRINCE2 Methodology

3.3 Software Development Model

There are many software development process models that describes the process of software development. (eg: Waterfall, Agile, Spiral Models) They are design to describe the phases of the software development and the order of the phases that are being executed.

There are six phases in every Software development life cycle model.

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

Software development process models are being different based on how they manage each phase. Table no:4 describes the comparison between popular software development models.

Model Name	Advantages	Disadvantages
Waterfall	• Uses clear structure with the system design to maintenance • Determines the end goal early • Transfers information well • Works best for the smaller projects	• Making changes are difficult • Focuses very little on end user • Delays testing until completion
Agile	• Responding to change, easy to do changes • Accepting uncertainty • Faster review cycle • Flexibility in releasing features	• Lack of predictability • Cannot do large deliverables • Flexibility can go off track
RAD	• Development time is reduced • Flexibility in accepting changes	• Not suitable for small projects

	• Reusable capability in coding. • Manual coding is reduced.	• Not suitable when the technical risk is high • High development skills • High cost
Spiral	• Additional functionality or changes can be added at later stage • Early prototype development • Repeated development helps in risk management • Fast development	• Not suitable for small projects • Needs risk assessment expertise • Sometimes cost for the project can be high

Table 4: Comparison Between Software Development Models

Following facts were considered when selecting a software development model for this project.

- Model should be suitable for medium scale project
- Prototype of the project should be developed.
- It should be able to easily do changes through the development process

Based on the comparison and the requirement this project needs, Spiral Model was selected as the software development model for this project.

3.4 Research Methodology

Research methodology is a systematic way for conducting researches. Qualitative and Quantitative research methods will be used to get data from experiments, observations and surveys researches. Quantitative methods aim to classify features and create statistical models to explain theories while Qualitative methods aim for a complete, detailed description to explain theories. Table no:5 shows the differences between Qualitative and Quantitative research methods.

Qualitative Evidence	Quantitative Evidence
Powered by words and descriptions	Powered by numbers and statistical analysis
Based on inductive reasoning	Based on deductive reasoning
More subjective	More objective
Associated with soft sciences (sociology, social work, linguistics, etc.)	Associated with hard sciences (math, physics, chemistry, etc.)

Table 5: Differences between Quantitative and Qualitative Researches

Due to the availability of past literature and the availability to conduct surveys, author has used both the qualitative and quantitative research methods to gather data for this research.

3.5 Task and Time Allocation

Time plan was prepared for the tasks to complete the project within the given time. Table no:6 shows the initial time plan for the project. Rifer Appendix: A for the Gantt chart.

ID	Task	Duration
01	Project Initiation	61 days
02	Literature Review	142 days
03	System Requirement Specification	28 days
04	System Design	31 days
05	Interim Report	17 days
06	Prototype Development	56 days
07	Final Prototype and Report (Thesis)	142 days

Table 6: Initial Time Plan for the Project "FIxMe"

3.6 Risk Mitigation and Limitation Analysis

Potential risks, limitations and mitigations for each risk are listed in the table no:7

Risk Id	Risk level	Risk Description	Mitigation Plan
01	Medium	Technological changes in the domain are not up-to-date	Stay alert on the new updates in the domain area.
02	Medium	Being unable to gather enough training images.	Start requirement gathering process early.
03	High	Hardware/Software failures	Maintaining backups. Using better Hardware.

04	High	Time limitation	Create a time plan for each phase and follow it.
05	Medium	Lack of domain knowledge.	Talk to experts on subject area. Study past literature that are belong to the domain area.

Table 7: Risk Mitigation and Limitations

3.7 Chapter Summary

The project management aspect of the project was discussed throughout this chapter. PRINCE2 methodology was selected as the project management methodology and the Spiral Model was selected for the Software Development Model for this project. Tasks and time for the tasks were identified and allocated through a Gantt at the end of the chapter.

Chapter 4: Requirement Specification

4.1. Chapter Overview

This chapter mainly focused on requirement analysis phase of the project. Gathered data from the past literature and survey questionnaires will be analyzed to identify the functional and non-functional requirements.

4.2. Stakeholder Analysis

Figure 12 shows the Onion model of the stakeholders for the project

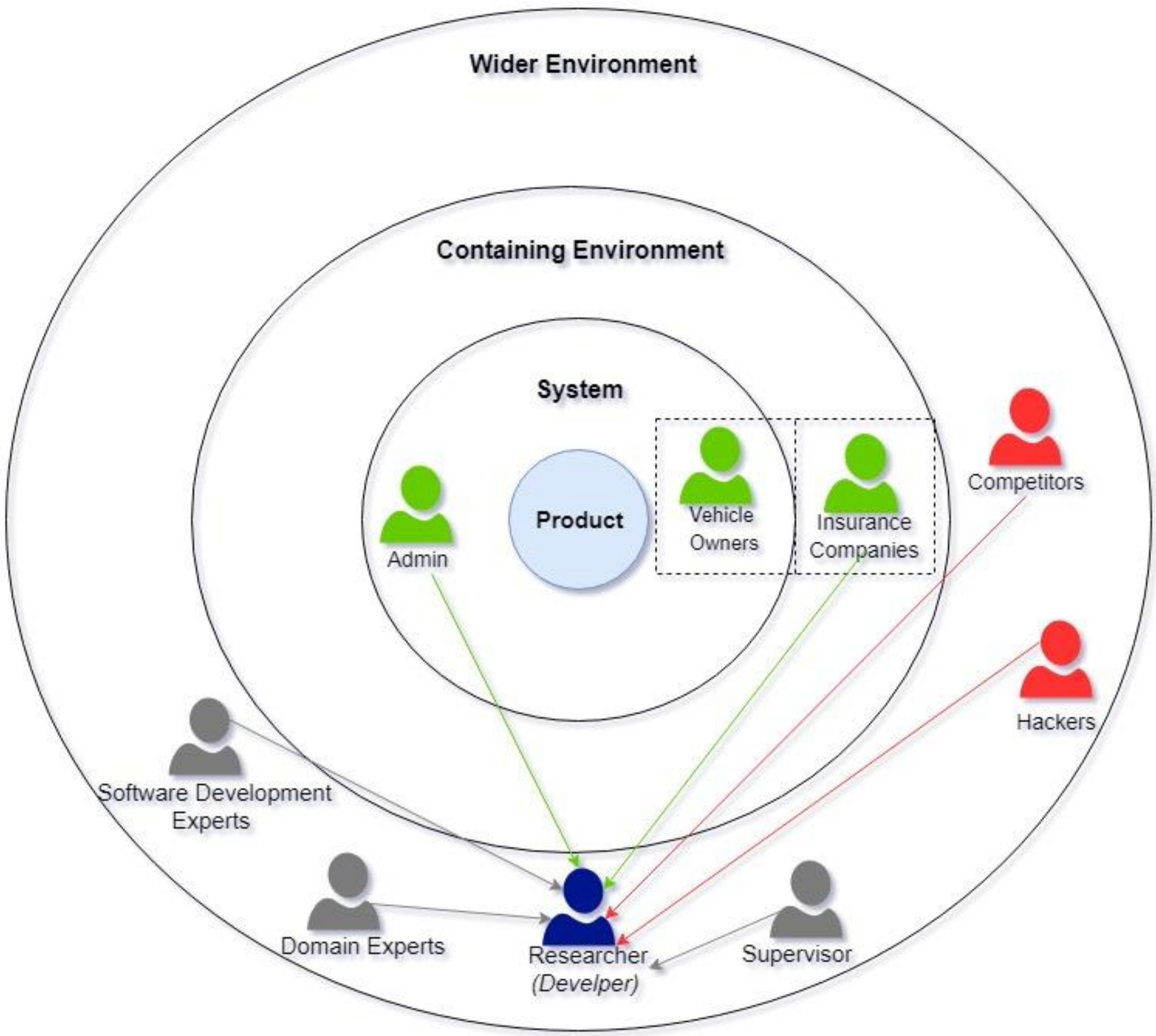


Figure 11: ONION Model of Stakeholders

Table down below(Table 4:) shows the Stakeholder viewpoints.

Stakeholder	Stakeholder Role	Viewpoint
Admin	Functional Beneficiary	Wants to moderate the users and functions of the system.
Vehicle Owners	Functional Beneficiary	Wants to be able to upload pictures of the damaged vehicle and get damage estimation.
Insurance Companies	Financial Beneficiary	Wants to be able to benefit from the users who request damage estimation via mobile.
Supervisor	Experts	Wants the researcher to complete the project with flying colors.
Software Development Experts	Experts	Wants to provide development knowledge about the image processing technology.
Domain Experts	Experts	Wants to provide domain knowledge about the vehicle damage estimation.
Competitors	Negative	Wants to build a better system than the researcher's system.
Hackers	Negative	Wants to break down the system.

Table 8: Table of Stakeholder viewpoints.

Description of the roles

- Functional Beneficiary– Stakeholders who directly interact with the functionality of the system.
- Financial Beneficiary- Stakeholders who benefit financially from the system.
- Experts- Stakeholders who are a strength in developing the system.
- Negative- Stakeholders who want to disrupt the behavior of the system.

4.3. Techniques for Requirement Gathering

The techniques that were used for requirement gathering are listed below in the table no:

Literature Review
Researching about the background of the project and existing work of the subject domain.
Online Survey Questionnaires
A survey which carries questionnaires to get peoples responses online.
Formal Interviews with The Insurance Agents
Interviewing domain experts to get an understanding of the problem and possible solutions for the problem.
Brainstorming
Brainstorming with colleagues and domain experts to gather requirements based on self-experience and knowledge on the subject area.
Observation
Observing the current existing solutions and similar solutions related to the domain.

Table 9: Requirement Gathering Techniques

4.4. Online Survey Questionnaire

The Questionnaire was directed to vehicle users who encountered a vehicle accident. Targeted response count was 100 and 84 responses were recorded.

4.4.1. Summary of the Survey

Question 1&2
1. Do you regularly use a vehicle to travel other than public transportation? 2. Have you ever encountered a vehicle accident while traveling?
Aim
To get information about the regular vehicle users and ones who encounter an accident while travelling.
Observation

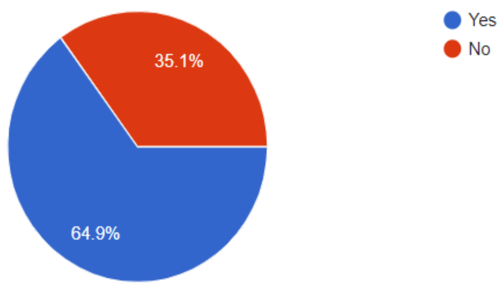


Figure 12: Survey Result for Requirement Gathering 1

According to the response, 64.9% of the people travel using a vehicle other than public transportation.

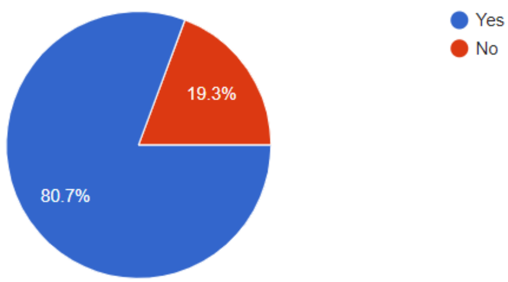


Figure 13: Survey Result for Requirement Gathering 2

From the ones who travel from a vehicle, more than 80.7% people encountered an accident while traveling.

Conclusion
Majority of the people who regularly travel in vehicles have a 80.7% chance to encounter an accident.

Question 3
How long did it take to clear the road after the accident?
Aim
To get information about the time spent during a vehicle collision.
Observation

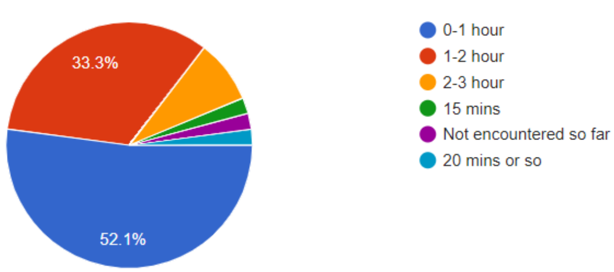
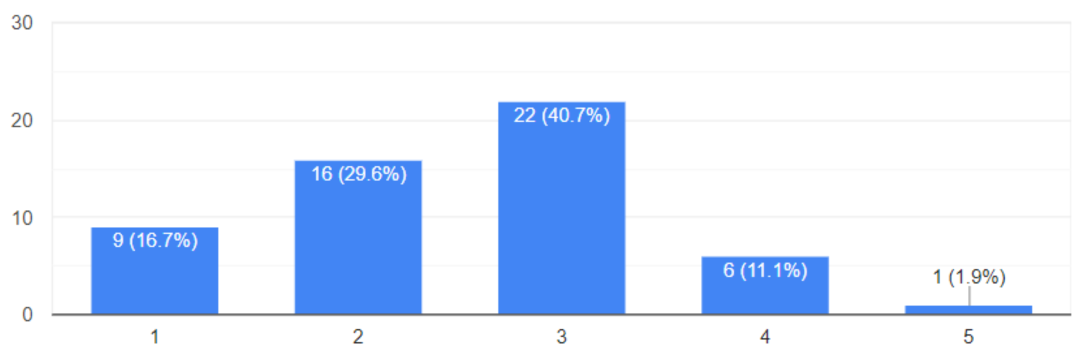


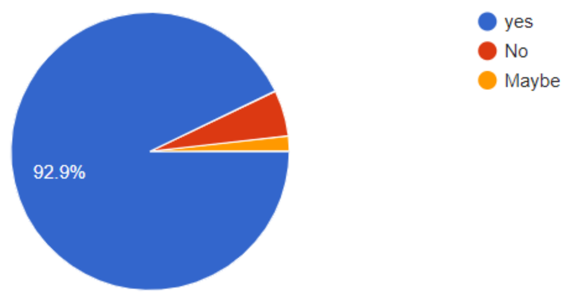
Figure 14: Survey Result for Requirement Gathering 3

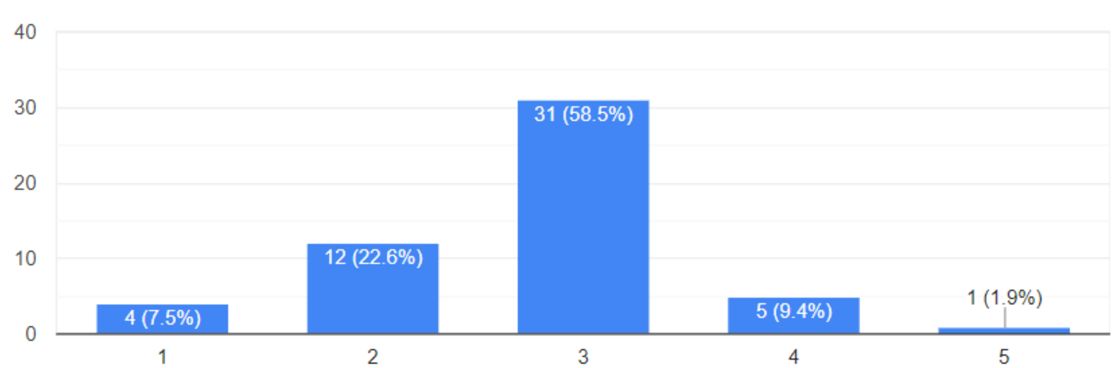
More than half of the people only spent between 0 to 1 hour during a collision. The second majority of the people (33.3%) spent between 1-2 hours and a small number of people spend more than 2 hours at the collision.

Conclusion
After a collision, vehicle owners have to wait until the insurance agent arrives to remove the vehicle from the collision location. According to the observation it takes around 0-1 hour and sometimes between 1-2 hour. This makes a traffic situation.

Question 4
How satisfied were you with the waiting time till the insurance agent arrives to your location?
Aim
To get information about people's opinion about the waiting time till the insurance agent arrive.
Observation
 *Figure 15: Survey Result for Requirement Gathering 4* 1- Very dissatisfied / 5- Very satisfied
Conclusion
Observation shows that only a small number of people are satisfied with the waiting time while majority of the people(46.3%) are unsatisfied with the waiting time.

Question 5
Do you think, use of a mobile application to calculate damage estimation would help to save time in the damage claiming process? (This includes waiting time till the insurance agent arrives to your location)

Aim
To get peoples opinion and the reaction towards the use of a mobile application to get faster with the damage claiming process.
Observation
 *Figure 16: Survey Result for Requirement Gathering 5*
Conclusion
Majority of people(92.9%) agrees with the use of a mobile application to calculate damage estimation will save time.

Question 6
How satisfied were you with the Damage estimation given by the repair center?
Aim
To get people's opinion about the damage estimation that was given by repair centers.
Observation
 *Figure 17: Survey Result for Requirement Gathering 6*

1- Very dissatisfied / 5- Very satisfied
Conclusion
Most of the people are good with the damage estimation they got from the repair centers and the second majority of people are looking forward to a better estimation.

Question 7
How satisfied will you be if you get a detailed report on the damage estimation and the components that are going to be repaired in your vehicle via mobile?
Aim
To get people’s opinion about getting a detailed estimation on the spot where the collision happened.
Observation

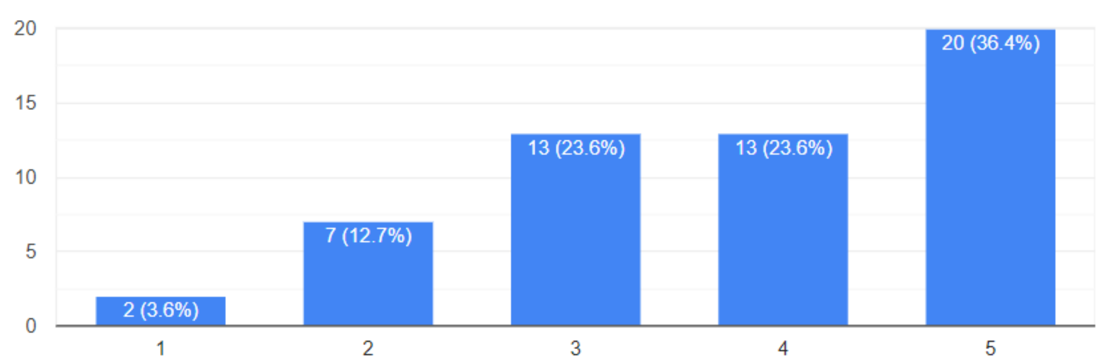


Figure 18: Survey Result for Requirement Gathering 8

1- Very dissatisfied / 5- Very satisfied.

Conclusion
Majority of the people are looking forward to a mobile system that can estimate damage on the spot and show detailed reports of the components that took damage.

4.5. Requirements Specifications

Functional and nonfunctional requirements were made by analyzing the survey, past literature and brainstorming

4.5.1. Functional Requirements

Functional Requirements that describes the behavior of the application.

No.	Requirement	Priority	Related Use Case
FR1	User should be able to login to the system	Low	Login
FR2	User should be able to sign up to the system	Low	Sign Up
FR3	User should be able to upload multiple images of the damaged car.	Medium	Upload Images
FR4	System should be able to filter out the vehicle damages	Very High	Classify Vehicle Damages
FR5	System should be able to identify severe vehicle damages	Very High	Classify Damage Severity
FR6	System should be able to get the latest price of the damaged components.	Very High	Get Price of Damaged Components
FR7	System should be able to estimate the damage cost	Very High	Calculate Damage

Table 10: Functional Requirements of "FixMe"

4.5.2. Non- Functional Requirements

Non-Functional requirements that shows the system characteristics are listed below according to the user needs.

No.	Requirement	Description
NF1	Accuracy	Application should be able to analyze images and give damage estimation accurately.
NF2	Performance	Application should be able to respond within 1 minute after requesting a damage estimation.
NF3	Usability	User Interaction with the UI of the application will be minimal. Simple UI to upload images and a graphical view to show damage estimation and details of the components.
NF4	Security	User authentication system.
NF5	Reliability	Servers will be always functional.

Table 11: Non-Functional Requirements of "FixMe"

4.6. Use Case Diagram

Use Case Diagram of the Project - Severe Damage Identification from Image Processing and Damage Cost Estimation. Functional requirements that displays in the table4 is mapped to the use case diagram. (Figure 12)

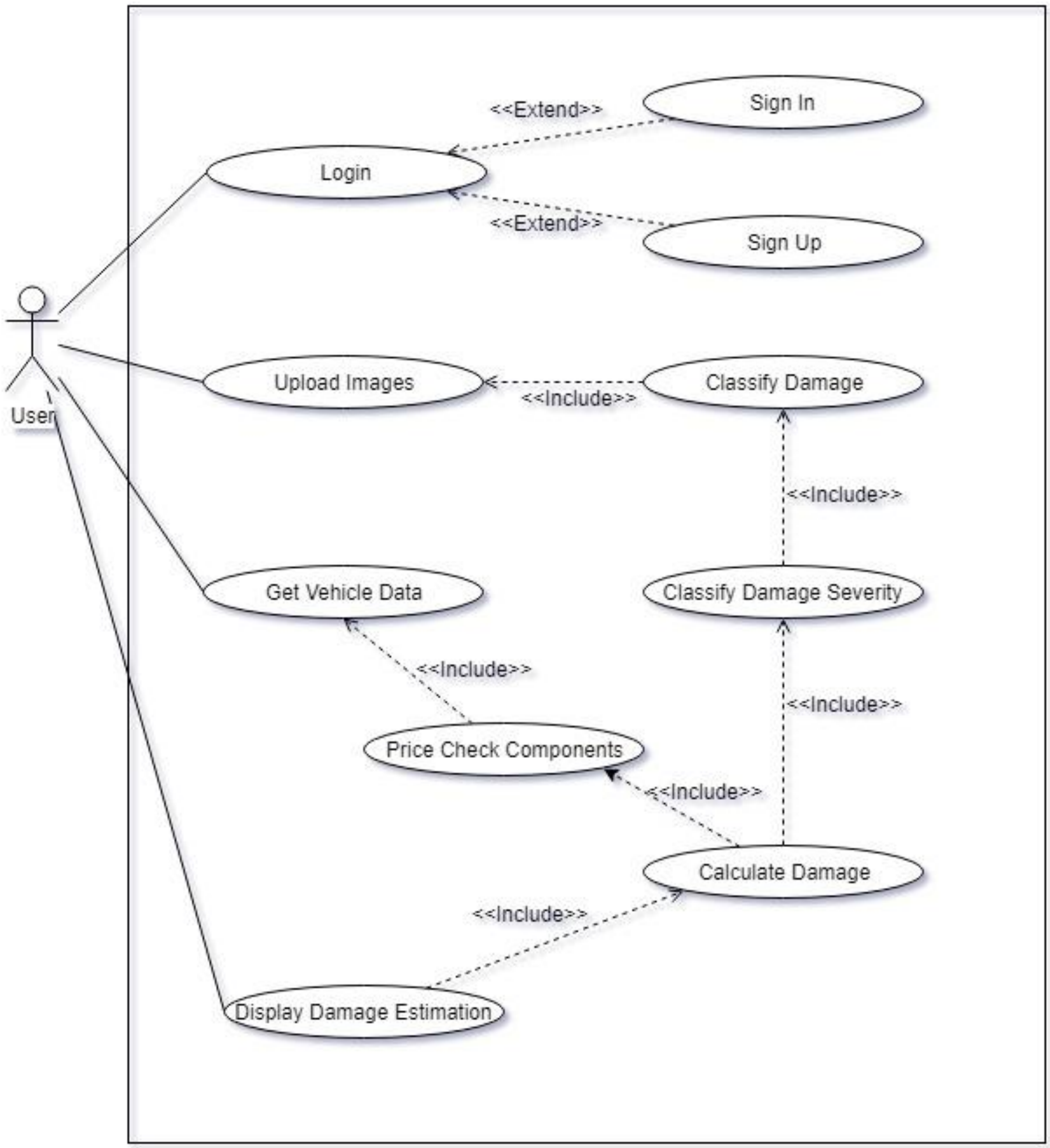


Figure 19:UseCase Diagram of "FixMe"

4.6.1. Use Case Description

Descriptions for the use cases are given in this section. Refer Appendix: C for more detailed descriptions for the use cases.

Use Case: Classify Damage

Use Case	Classify Damage
Id	001
Description	Allows FixMe to classify the damage for each severity type
Primary actor	FixMe
Supporting actor	none
Pre-condition	Image need to be uploaded in tot the system.
Post-condition	After classifying the damage, classifying of damage severity will start. (Use case: Classify Damage Severity)
Trigger	User wishes to proceed to damage estimation.
Main Success scenario	1) user opens the app. 2) User sign in 3) User upload image 4) Start classifying damage process.
Variations	1)If user didn't upload an Image- Error message will be shown 2)If user upload a corrupted Image – Error message will be shown 3)If user select another file type than images– Error message will be shown 4)User can log out from the system.

Table 12:Use Case Description for Classify Damage

Use Case: Classify Damage Severity

Use Case	Classify Damage Severity
Id	002
Description	Allows FixMe to choose the best reliable(accurate) damage type for the given image.
Primary actor	FixMe
Supporting actor	none
Pre-condition	Classification process for each damage type should have been completed.
Post-condition	Gives FixMe the most accurate severity level for the input image.
Trigger	Classify Damage
Main Success scenario	1) Get the damage classification accuracy of each damage severity. 2) Sort out the damage severities. 3) Select the most accurate severity.

	4) Start calculating damage process.
Variations	1) Accuracy below 20% will be neglected and if all the accuracy levels are below 20%, Error message will be shown to recheck the image.

Table 13:Use Case Description for Classify Damage Serverity

Use Case: Calculate Damage

Use Case	Calculate Damage
Id	003
Description	Allows FixMe to calculate a damage cost for the vehicle.
Primary actor	FixMe
Supporting actor	none
Pre-condition	Damage severity need to be calculated.
Post-condition	Display prices for the vehicle component that was selected for damage estimation.
Trigger	Classify Damage Severity
Main Success scenario	1) Get the most accurate damage severity. 2) If the severity was severe, display a WebView containing price of the vehicle component that was damaged.
Variations	1) Only the accuracy of the minor and moderate damage types will be shown

Table 14:Use Case Description for Calculate Damage

4.7. Chapter Summary

This chapter went through various techniques to analyze requirements and the requirements have been documented as functional and nonfunctional requirements. Stake holder analysis was done at the beginning and requirement gathering techniques were discussed after that. With those techniques, list of functional and non-functional requirements was created and documented. A use case diagram for those functions was created and its description was given after that. Next chapter will be discussing about the designing aspect of the project.

Chapter 5: Design & Architecture

4.5. Chapter Overview

The previous chapter identified the requirements of FixMe. This chapter is going to discuss about the design and architecture of FixMe. It is very important to have a solid design for the software before going into the implementation phase. Therefore, Design takes a very crucial phase in the software life cycle. This chapter will discuss about the design of FixMe.

4.6. Design Goals

Design of the FixMe was done to achieve following design goals.

Design Goal	Description
Flexibility	Requirements of the projects can be change during the implementation stage even the implementation is begun after a thorough analysis of the requirements. To overcome this problem the system should be very flexible to adapt to new changes and go forward without failures.
Reusability	Code of a well-designed system should be able to re-use and apply to similar projects.
Efficiency	Efficiency is one of the major goals in designing systems prior to implementation. A well-designed system should make the greatest use of the available memory, processing power and network speed.
Accuracy	Accuracy one of the highest priorities of the project. The system must be designed so as not to adversely affect accuracy.
Scalability	The prototype should be implemented in a way that developers can add more enhancement algorithms and classifiers without any difficulty in the future.

Table 15:Design Goals of "FixMe"

4.7. System Design Methodology

One of the major reasons to fail a project is not following a proper system design methodology. A methodology can be simply defined as a set of methods that are used in a particular area of study from the beginning to the completion of the process. The nature of a methodology relies on variety of things. Some of these things are software development environment, practices of the organization's, the necessities of the users, the nature or type of the software being developed, the availability of hardware and software resources, the availability of existing design modules, the budget and therefore the time schedule. There have been a rapid increase of the software design methodologies since the 1970s. Different methodologies have been developed to resolve different types of problems. (Khoo, 2011)

4.7.1. Object Oriented Analysis and Design (OOD)

OOD focuses on objects that are different states and data and code representations with a high focus on reusability. OOD is widely used in the software industry and its strength lies in its ability to cope with data mutations easily. This means that several entities can access and modify the same piece of data simultaneously without any problems OOD principles such as encapsulation allows objects to remain hidden inside while allowing external processes to work with them. OOD is considered the best choice for large, high-risk projects due to its ability to meet constantly changing requirements.

4.7.2. Structured System Analysis and Design (SSAD)

SSAD follows a top-down design approach with a higher focus on data manipulation processes. SSAD gives more attention to the changes of the data and not on the code that manipulates it. This reflects its strength and efficiency in the use of immutable data. But when its come to working with mutable data, SSAD is at a disadvantage. It cannot manage multiple entities at the same time constantly manipulating its data. Because of these reasons, SSAD is better for working with the projects that have stable requirements with a well-defined scope. In recent years, there has been a growing popularity in functional-oriented languages such as Scala and Haskell (SSAD-favoring languages).

4.7.3. Selection of the Design Methodology

The following key pointers are used to select the design methodology for the "FixMe"

- Application will take images and vehicle details one by one
- Application will process the images and produce an output
- All data used by the application will remain immutable
- No external entities or systems will be trying to process the data at the same time.

The overall meaning of the key points is that the application's next process will not start until completion of the current process. This gives a strong emphasis on the data that is passing through the system and therefore, following a structured system analysis and design methodology is much more suitable for the project than the object-oriented design methodology.

4.8. High Level Design

This section highlights the high-level design of FixMe. The high-level design will be used to describe the major components of the system and the ways that they interact with each other.

4.8.1. Rich Picture of FixMe

A Rich Picture of a system is a collection of symbols that are arranged in a way of diagrams to explore and acknowledge a situation. It Figure 20 shows the rich picture for the "FixMe"

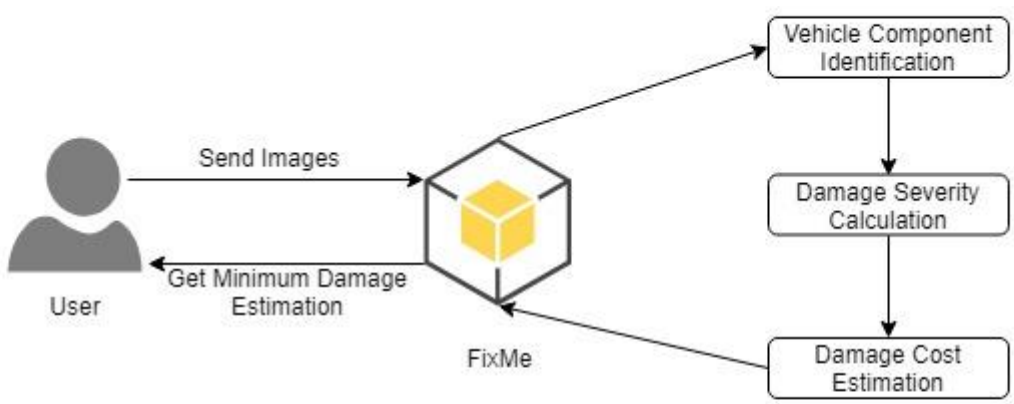


Figure 20:Rich Picture diagram of "FixMe"

Rich picture visualizes the User sending images of the damaged vehicle and vehicle information through the web interface of FixMe. The System first identifies the model of the car and the damaged component of the car and sends images to calculate the damage severity. After the calculation of the damage severity, the cost estimation process will start and send the calculated estimation of the damage to the user.

4.9. Design Diagrams

Selected design methodology for "FixMe" is Structured System Analysis and Design Methodology (SSAD) and therefore, Object Oriented Design (OOD) favored diagrams such as class diagram and domain models are not suitable for this application. The reason for this is that functional applications do not model objects and mutable states. Instead of the class diagram and domain models, the following diagrams were selected to design the "FixMe".

4.9.1. Context Diagram

A context diagram also called a level-0 data flow diagram, shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, data stores, etc.). Context diagrams can be broken in to smaller processes named level-1 Data Flow Diagrams. Data Flow Diagram (DFD level-1) is a graphical visualization of the data flow through a process in context diagram. This diagram also can be broken in to smaller processes such as DFD (level-2), and so on. Figure:21 shows the context diagram of the "FixMe"

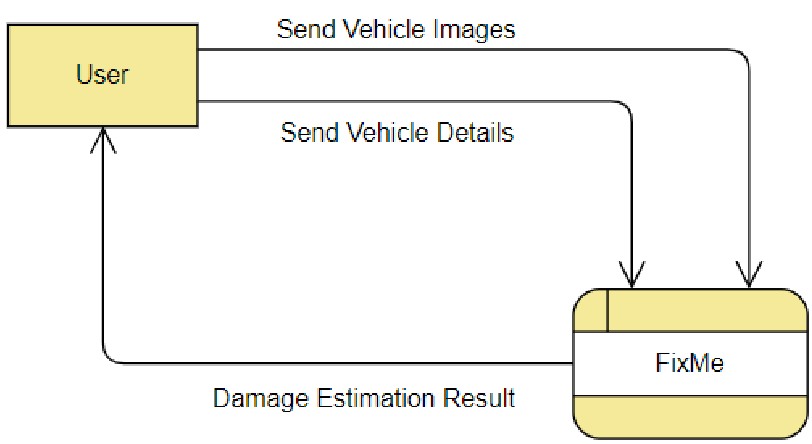


Figure 21:Context Diagram for "FixMe"

4.9.2. Data Flow Diagram(level-1)

Figure: 22 Shows the level 1 data flow diagram for "FixMe". This figure visualizes all the components and data flow through the system.

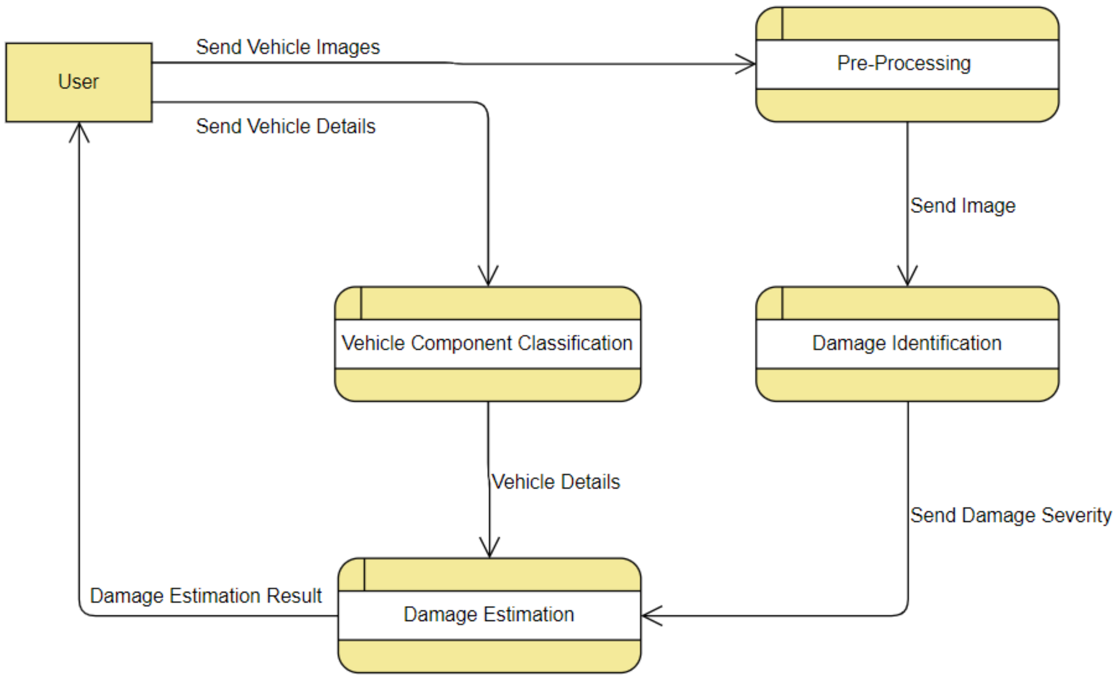


Figure 22: Data Flow Diagram (Level-1) for "FixMe"

4.9.3. Data Flow Diagram (level-2)

For the data flow diagram(level-1), main components in the DFD(level-1) were broken down in detail.

Damage Estimation: Figure:23 Shows the Damage Estimation process.

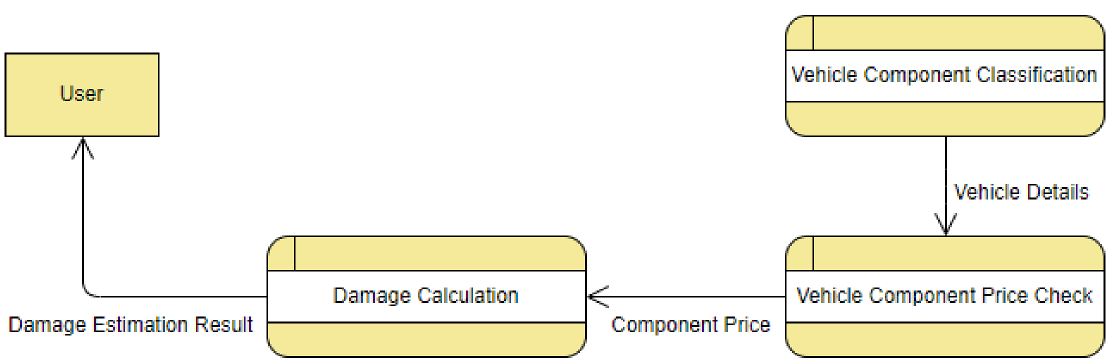


Figure 23:Data Flow Diagram (level-2) Damage Estimation process of "FixMe"

Damage Identification: Figure:24 Shows the Damage Identification process.

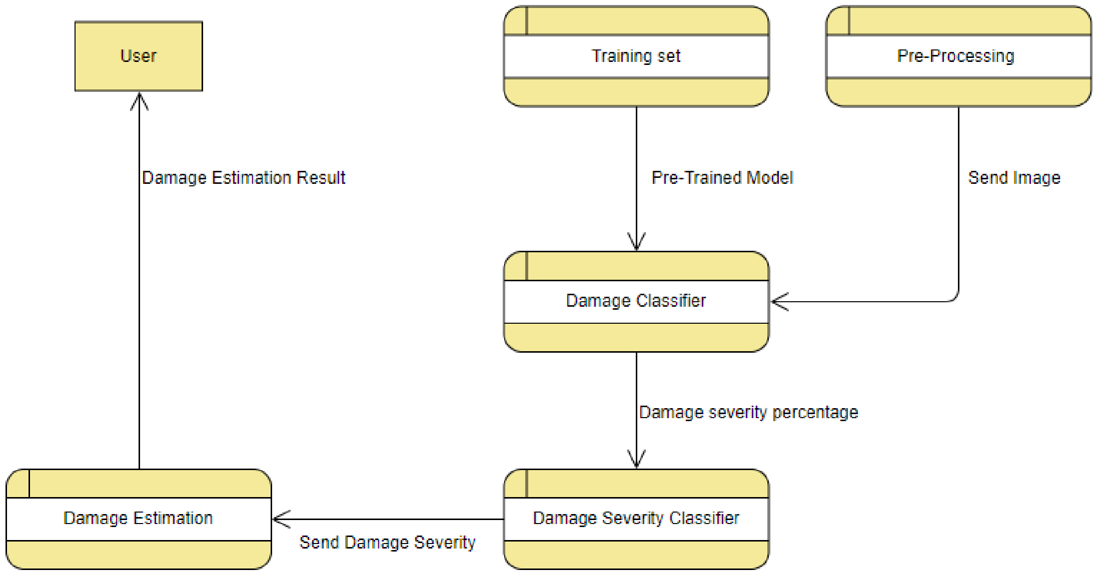


Figure 24:Data Flow Diagram (Level-2) Damage Identification of "FixMe"

4.10. User Interface Design

UI mock-ups were created to describe the finished product's look and feel. Refer Appendix: B for the UI Previews.

4.11. Chapter Summary

This chapter justified SSAD's selection as the system's design methodology. A rich picture diagram was used to discuss the overall functionality of the system. A data flow diagram was designed in several layers using these identified functionalities to help confirm the system structure. With these design specifications, the implementation process of the prototype will be discussed in the next chapter.

Chapter 6: Implementation

6.1. Chapter Overview

The Implementation phase of the system will be discussed in this chapter. According to the Design & Architecture of the system discussed in the previous chapter appropriate technology selections, development language, frameworks and related facts will be discussed in detail throughout this chapter.

6.2. High Level Architecture

The first phase in the implementation was to create a high-level architecture of "FixMe" system. Figure:25 explains the main components in "FixMe"

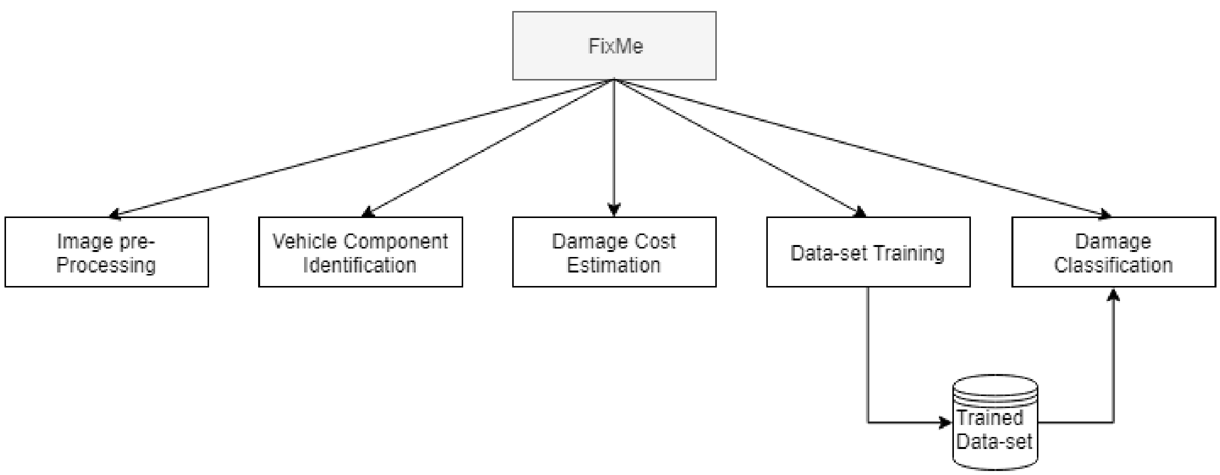


Figure 25:The High-Level Architecture of "FixMe"

6.3. Selection of the Technologies

This section includes explanations and justification for selecting for the technologies that are used in this project. For the implementation of this prototype, several image processing technologies were considered. Flowing factors were taken in to consideration when selecting a proper technology.

Factors	Description
Features	Technologies are not created for a single purpose and therefore every technology has a number of limitations. Selected technology must have the features that are needed to implement the system.
Learnability	Technology should be easy to learn due to the limited time given.
Performance	Speed and efficiency are needed.
Compatibility	Technology must be compatible with the requirements in the SRS.

Table 16: Factors and Description of a Proper Technology

Because of these factors, several image processing technologies were considered for the implementation of "FixMe". Each technology has their unique advantages and disadvantaged and table below is used to summaries those details.

	OpenCV	MATLAB	TensorFlow
Features	One of the popular open-source image processing libraries. Written in c++. Has bindings for python, Java.	Has pre-defined tool boxes for image processing and pattern recognition. Written in C, C++, Java. Has several inbuilt functions (code bulks) to make development time shorter. Supports all the required functions.	Open-source software library for machine learning and deep neural network. Written in Python, C++, CUDA. Has inbuilt functions to make the development time shorter.
Learnability	Challenging for a beginner. There's a large community support. Learning resources are available.	High level language. Learning resources are available.	Easy to learn. Learning resources are available. Has a large community support.

Performance	Faster than other other two technologies. Can be further optimized to increase the speed of the implementation.	Has its own compilers and it's slower than the other two technologies.	Faster and can be optimized to increase the speed.
Compatibility	Compatible with the requirements in SRS.	Compatible with the requirements in SRS.	Compatible with the requirements in SRS.

Table 17:Differances Between the Technologies- OpenCV/MATLAB/TensorFlow

TensorFlow

TensorFlow is a free, open source machine learning and deep neural network library designed by the Google brain team. It is based on the graphs that show the data flow of the system. Graphs are made from nodes which are mathematical operators and multi-dimensional arrays. These nodes are also known as tensors. (Deep Learning with TensorFlow - Introduction to TensorFlow, 2016).

TensorFlow is well known for its deep learning capabilities and the deep learning has become successful in image classification than manually extracting features from the images and comparing them. TensorFlow provides a classification library called "TensorFlow for Poets". It uses a pre-existing classifier called "Inception" and it is known as one of the best google opensource image classifiers. It is also trained on 1.4 Million images on 1000 categories (TensorFlow, 2019). And also, it uses transfer learning which is a machine learning technique which will be used to create a new classifier using feature extraction details from images. Which means that it can create a new classifier from a new set of images.

Due to these factors and above-mentioned factors when choosing a technology, TensorFlow was identified as the most suitable technology for this project.

6.4. Selection of Development Languages

Selecting a development language needs to be made very carefully. It is one of the most important aspects in a project because the whole project depends on the capabilities of the development language. Before choosing a good development language for “FixMe”, following factors were taken into consideration.

Factors	Description
Popularity	Popularity of the language needs to be taken in to consideration when choosing a proper development language for the project. It becomes more easier to find tutorials, guides, reference materials and people to collaborate in the project.
Language-Domain match	Choosing a popular language is useless if the language doesn’t match with the domain of the project. From analyzing similar projects that are done in the past and capabilities of the development languages, it can be decided the development language matches the problem domain or not.
Availability of Libraries	Implementing a good existing solution for a function from the scratch is a waste of time. Choosing a language that has a better range of libraries to address your problem area is good. More libraries will be developed and available for use when a language becomes popular. Therefore, the language's popularity affects a language's library availability.
Efficiency	Before choosing a language, consideration should be given to the effectiveness of compilers and interpreters.
Project’s Scope	Depending on the scope of the project, language selection may differ from a simple high-level language to a heavy low-level language.
Supported Tools	Tools make it easier and faster to develop. The developer can carry on the development process without any trouble with a language with a great tool support.
Familiarity	Familiarity of the development language to the developer is also an important factor when selecting a development

	language. It doesn't matter if the language is very popular and have many libraries if the user is not familiar with it. Developer should be able to be comfortable with the development language,

Table 18:Factors and Discription of Proper Development Languages

In view of all the above facts, python has been selected as the development language for "FixMe" back-end development. JavaScript & HTML have been selected as the front-end development languages.

6.5. Development Environment

Development was performed in a computer with moderate performance specifications.

- Intel Core i7 / 2.2GHz
- 8GB RAM
- 80GB of free HDD space

6.6. Implementation

Implementation aspect and the steps of the project will be discussed throughout this section.

6.6.1. Training the Image Classifier

TensorFlow is well known for its deep learning capabilities and for this implementation widely used image recognition model named "Inception model" is used. Inception V3 was able to attain greater than 78.1% accuracy on the ImageNet dataset (Google Cloud, 2019) and because of that Inception V3 model will be use in this classifier.

6.6.2. Pre-Processing of the test images

Pre-processing was done to set the parameters of the testing and improve and enhance the image data.

```
# training steps till ending
HOW_MANY_TRAINING_STEPS = 1000

# learning rate
LEARNING_RATE = 0.01

# percentage of images to use as a test set
TESTING_PERCENTAGE = 10

# percentage of images to use as a validation set
VALIDATION_PERCENTAGE = 10

# evaluation interval between the training results
EVAL_STEP_INTERVAL = 10

# images size at a time
TRAIN_BATCH_SIZE = 100

# How many images to test on. This test set is only used once, to evaluate the final accuracy of the model after
# training completes.  A value of -1 causes the entire test set to be used, which leads to more stable results.
TEST_BATCH_SIZE = -1

# Image size in an evaluation batch. This validation set is used much more often than the test set, this is an indicator
# of how accurate the model is during training. A value of -1 causes the entire validation set to be used, which leads
# to more stable results across training iterations, but may be slower on large training sets.
VALIDATION_BATCH_SIZE = 100

# whether to print out a list of all misclassified test images
PRINT_MISCLASSIFIED_TEST_IMAGES = False

# Path to classify_image_graph_def.pb, imagenet_synset_to_human_label_map.txt, and imagenet_2012_challenge_label_map_prot
MODEL_DIR = os.getcwd() + "/" + "model"

# Path to cache bottleneck layer values as files
BOTTLENECK_DIR = os.getcwd() + '/' + 'bottleneck_data'

# the name of the output classification layer in the retrained graph
FINAL_TENSOR_NAME = 'final_result'
```

Figure 26:Screenshot of Initial step of pre-processing

6.6.3. Downloading the model

Used architecture was Inception V3 model and below code snippet in figure:27 explains the downloading step of the v3 model.

```
ARCHITECTURE = 'inception_v3'
#check model
model_info = create_model_info(ARCHITECTURE)
if not model_info:
    tf.logging.error('Did not recognize architecture flag')
    return -1
#download model
print("downloading model  . . .")
downloadModelIfNotAlreadyPresent(model_info['data_url'])
print("creating model graph . . .")
graph, bottleneck_tensor, resized_image_tensor = (create_model_graph(model_info))
```

Figure 27:Screenshot of setting the setting up the Inception V3 Model

Create_model_info() was given in the figure down below. Model will be saved as "classify_image_graph_def.pb"

```
if architecture == 'inception_v3':
    data_url = 'http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz'
    bottleneck_tensor_name = 'pool_3/_reshape:0'
    bottleneck_tensor_size = 2048
    input_width = 299
    input_height = 299
    input_depth = 3
    resized_input_tensor_name = 'Mul:0'
    model_file_name = 'classify_image_graph_def.pb'
    input_mean = 128
    input_std = 128
elif architecture.startswith('mobilenet_'):
    parts = architecture.split('_')
    if len(parts) != 3 and len(parts) != 4:
        tf.logging.error("Couldn't understand architecture name '%s'", architecture)
        return None
```

Figure 28: Screenshot of Saving the V3 Model

6.6.4. Creating image list

After downloading the model, A list of images will be created. Below code snippet is for creating the list of images and folder checking validations. This will go through folders and store it as labels.

```
# create lists of all the images.
print("creating image lists . . .")
image_lists = create_image_lists(TRAINING_IMAGES_DIR, TESTING_PERCENTAGE, VALIDATION_PERCENTAGE)
class_count = len(image_lists.keys())
# folder validation
if class_count == 0:
    tf.logging.error('No valid folders of images found at ' + TRAINING_IMAGES_DIR)
    return -1
if class_count == 1:
    tf.logging.error('Only one valid folder of images found at ' + TRAINING_IMAGES_DIR + ' - multiple classes are needed for classification.')
    return -1
```

Figure 29:Screenshot of Image List Creation and Folder Validation

Bellow figure displays the code snippet in create_image_lists() function. Accuracy depends on the size of the image set. Therefore, the image sizes of each containing folder will be checked and if the folders which have images lesser than 20 will prompt with a warning. Refer Appendix: F for more detailed figure of the function.

```
    if len(file_list) < 20:
        tf.logging.warning('WARNING: Folder has less than 20 images, which may cause issues.')
    elif len(file_list) > MAX_NUM_IMAGES_PER_CLASS:
        tf.logging.warning('WARNING: Folder {} has more than {} images. Some images will never be selected.'.format(dir_name, MAX_NUM_IMAGES_PER_CLASS))
```

Figure 30: Screenshot of Giving Warning Messages to for the Folders that have Images Lesser than 20

Figure down below displays the code snippet for performing the decoding process. It's a pre-processing step where the image properties will be change and also distortions will apply in order to cache the image summaries.

```
with tf.Session(graph=graph) as sess:
    # Set up the image decoding sub-graph.
    print("performing jpeg decoding . . .")
    jpeg_data_tensor, decoded_image_tensor = add_jpeg_decoding( model_info['input_width'],
                                                                model_info['input_height'],
                                                                model_info['input_depth'],
                                                                model_info['input_mean'],
                                                                model_info['input_std'])
    print("caching bottlenecks . . .")
    distorted_jpeg_data_tensor = None
    distorted_image_tensor = None
    if doDistortImages:
        #  setup the operations to apply distortions.
        (distorted_jpeg_data_tensor, distorted_image_tensor) = add_input_distortions(FLIP_LEFT_RIGHT, RANDOM_CROP, RANDOM_SCALE,
                                                                                     RANDOM_BRIGHTNESS, model_info['input_width'],
                                                                                     model_info['input_height'], model_info['input_depth'],
                                                                                     model_info['input_mean'], model_info['input_std'])
    else:
        # cache 'bottleneck' image summaries
        cache_bottlenecks(sess, image_lists, TRAINING_IMAGES_DIR, BOTTLENECK_DIR, jpeg_data_tensor, decoded_image_tensor,
                          resized_image_tensor, bottleneck_tensor, ARCHITECTURE)
```

Figure 31:Screenshot for Decoding Process

6.6.5. Training Steps

Figure:32 displays the code snippet for training process of the images. Number of steps is pre-defined in the pre-processing step. In the training step, image feature extraction details will be saved under their label names. Accuracy of the trained model is discussed in the testing chapter. After the training process, overall accuracy (final test accuracy of 1000 steps) of the implemented classifier will be shown. Refer figure down below for log details.

```
INFO:tensorflow:2019-05-04 18:12:00.507030: Step 999: Train accuracy = 100.0%
INFO:tensorflow:2019-05-04 18:12:00.507030: Step 999: Cross entropy = 0.131229
running testing . . .
INFO:tensorflow:2019-05-04 18:12:00.648623: Step 999: Validation accuracy = 78.0% (N=100)
writing trained graph and labbels with weights
INFO:tensorflow:Final test accuracy = 82.1% (N=28)
INFO:tensorflow:Froze 2 variables.
INFO:tensorflow:Converted 2 variables to const ops.
done !!
```

Figure 32: Screenshot of the Log Details of the Training Process

```python
    for i in range(HOW_MANY_TRAINING_STEPS):
        # Get a batch of input bottleneck values, either calculated fresh every
        # time with distortions applied, or from the cache stored on disk.
        if doDistortImages:
            (train_bottlenecks, train_ground_truth) = get_random_distorted_bottlenecks(sess, image_lists, TRAIN_BATCH_SIZE, 'training',
                                                                                      TRAINING_IMAGES_DIR, distorted_jpeg_data_tensor,
                                                                                      distorted_image_tensor, resized_image_tensor, bottleneck_tensor)
        else:
            (train_bottlenecks, train_ground_truth, _) = get_random_cached_bottlenecks(sess, image_lists, TRAIN_BATCH_SIZE, 'training',
                                                                                      BOTTLENECK_DIR, TRAINING_IMAGES_DIR, jpeg_data_tensor,
                                                                                      decoded_image_tensor, resized_image_tensor, bottleneck_tensor,
                                                                                      ARCHITECTURE)
        # end if

        # Feed the bottlenecks and ground truth into the graph, and run a training
        # step. Capture training summaries for TensorBoard with the `merged` op.
        train_summary, _ = sess.run([merged, train_step], feed_dict={bottleneck_input: train_bottlenecks, ground_truth_input: train_ground_truth})
        train_writer.add_summary(train_summary, i)

        # Every so often, print out how well the graph is training.
        is_last_step = (i + 1 == HOW_MANY_TRAINING_STEPS)
        if (i % EVAL_STEP_INTERVAL) == 0 or is_last_step:
            train_accuracy, cross_entropy_value = sess.run([evaluation_step, cross_entropy], feed_dict={bottleneck_input: train_bottlenecks, ground_truth_input: train_ground_truth})
            tf.logging.info('%s: Step %d: Train accuracy = %.1f%%' % (datetime.now(), i, train_accuracy * 100))
            tf.logging.info('%s: Step %d: Cross entropy = %f' % (datetime.now(), i, cross_entropy_value))
            validation_bottlenecks, validation_ground_truth, _ = (get_random_cached_bottlenecks(sess, image_lists, VALIDATION_BATCH_SIZE, 'validation',
                                                                                               BOTTLENECK_DIR, TRAINING_IMAGES_DIR, jpeg_data_tensor,
                                                                                               decoded_image_tensor, resized_image_tensor, bottleneck_tensor,
                                                                                               ARCHITECTURE))
            # Run a validation step and capture training summaries for TensorBoard with the `merged` op.
            validation_summary, validation_accuracy = sess.run(
                [merged, evaluation_step], feed_dict={bottleneck_input: validation_bottlenecks, ground_truth_input: validation_ground_truth})
            validation_writer.add_summary(validation_summary, i)
            tf.logging.info('%s: Step %d: Validation accuracy = %.1f%% (N=%d)' % (datetime.now(), i, validation_accuracy * 100, len(validation_bottlenecks)))
        # end if

        # Store intermediate results
        intermediate_frequency = INTERMEDIATE_STORE_FREQUENCY

        if (intermediate_frequency > 0 and (i % intermediate_frequency == 0) and i > 0):
            intermediate_file_name = (INTERMEDIATE_OUTPUT_GRAPHS_DIR + 'intermediate_' + str(i) + '.pb')
            tf.logging.info('Save intermediate result to : ' + intermediate_file_name)
            save_graph_to_file(sess, graph, intermediate_file_name)
```

Figure 33: Screen shot of Training Process of the Model

After finishing with the training step, it is possible to proceed to identifying the damage severity step with the help of the trained classifying module.

6.6.6. Check the File Path and File Extension

First step calculating severity is to check the file past is correct. The figure down below displays the code snippet for checking the file path and generating relevant error messages.

```python
def checkFilePath():
    if not os.path.exists(TEST_IMAGES_DIR):
        print('')
        print('ERROR: TEST_IMAGES_DIR "' + TEST_IMAGES_DIR + '" does not exist')
        print('')
        return False

    if not os.path.exists(RETRAINED_LABELS_TXT_FILE_LOC):
        print('ERROR: RETRAINED_LABELS_TXT_FILE_LOC "' + RETRAINED_LABELS_TXT_FILE_LOC + '" does not exist')
        return False

    if not os.path.exists(RETRAINED_GRAPH_PB_FILE_LOC):
        print('ERROR: RETRAINED_GRAPH_PB_FILE_LOC "' + RETRAINED_GRAPH_PB_FILE_LOC + '" does not exist')
        return False

    return True
```

Figure 34: Screenshot of File Validations

The code snippet shown in the figure down below is used to filter out .jpg and .jpeg files and ignore other files. This is used as a validation to check image files.

```
# if the file does not end in .jpg or .jpeg ,  continue to next file
if not (fileName.lower().endswith(".jpg") or fileName.lower().endswith(".jpeg")):
    continue
```

Figure 35: Screenshot of File Validation(File Extensions)

After finding out the correct file with the extension (.jpg, .jpeg), open cv will be used to read the image file.

6.6.7. Estimating Damage Severity

After checking the file location, the classification process will start. Process will go through image files one by one and give the damage severity with the percentage of becoming that severity. OpenCV will be used here to read the image file and the OpenCV image (NumPy array) will be converted to a TensorFlow image. TensorFlow image will be use to classify the damage severity and it will be sort out to get the order from most accurate to least accurate. Most accurate one will be the severity of the damage and it will be taken out to calculate minimum damage estimation. Figure down below displays a code snippet used for this process.

```
# pen the image with OpenCV
openImage = cv2.imread(imageFileWithPath)
# if opening image was not successful, continue to next file
if openImage is None:
    print("unable to open " + fileName + " image")
    continue

# get the final tensor from the graph
finalTensor = sess.graph.get_tensor_by_name('final_result:0')
# convert the OpenCV image (numpy array) to TensorFlow image
tfImage = np.array(openImage)[:, :, 0:3]
# run the network to get the predictions
predictions = sess.run(finalTensor, {'DecodeJpeg:0': tfImage})

# sort predictions from most reliable to least reliable
sortedPredictions = predictions[0].argsort()[-len(predictions[0]):][::-1]
```

Figure 36: Screenshot for the Code Estimating the Damage Severity

6.6.8. Damage Estimation

In this project author only address the severe damage estimation and if the estimated severity of the prototype belongs to severe damages, estimating damage process will start. It will prompt with the search results of the relevant vehicle part and user can get the prices for the real time car part prices.

6.6.9. Web Interface Preview

Figure down below displays the homepage of "FixMe". Refer Appendix: B for Sign Up, Sign In, Damage Estimation interface previews.

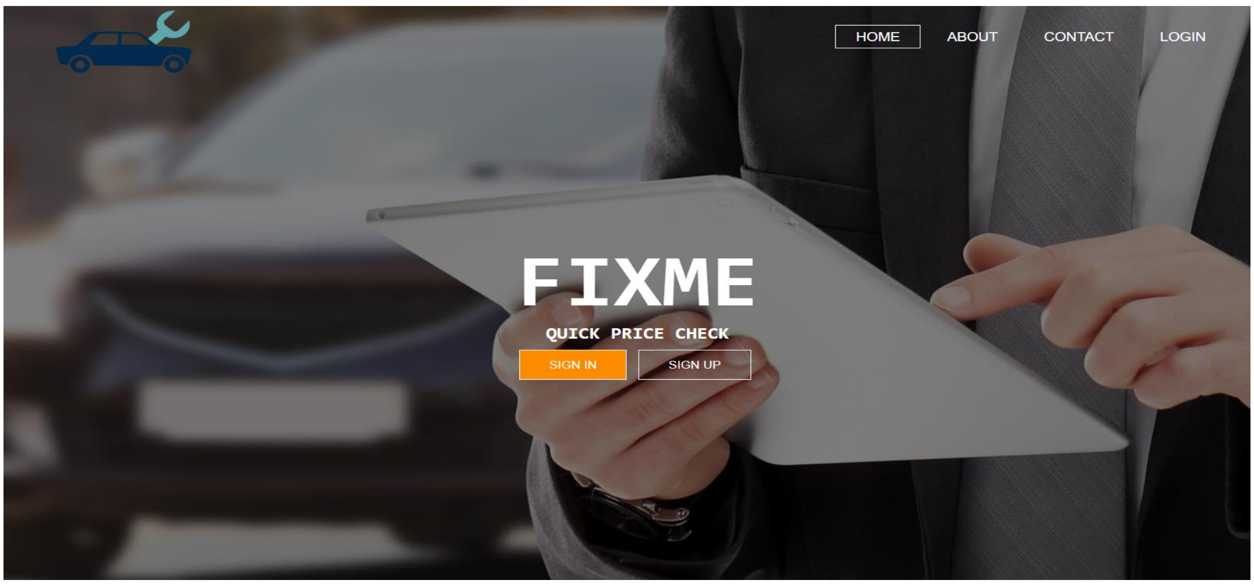


Figure 37: Preview of the Homepage of "FixMe"

6.7. Chapter Summary

This chapter is focused on finding out the suitable technologies and development languages to develop the provided solution. TensorFlow was selected as the main technology due to its learnability and preexisting libraries that can help to reduce the development time of "FixMe". Python was selected as the backend development language and html and JavaScript was chosen for the front-end development. Later, it discusses the implementation process and problems faced and the solutions that was provided for them. In the next chapter the testing process of the prototype will be discussed.

Chapter 7: Testing

7.1. Chapter Overview

Software testing is an important phase in the development of any application. An effective testing phase allows the prototype to be quickly analyzed and fix bugs. It also allows for a more effective evaluation. This chapter will first explain the selected testing strategies/justification for the selections and the testing phase of the application. Software testing is a trade-off between budget, time and quality. (Pan, 1999)

7.2. Introduction to Software Testing

The main objective of the software testing phase is to ensure the system works without errors or unexpected problems such as software crashing or performance issues. The following factors have been identified to clearly state what is expected and needed through a testing phase.

- Completion of the functional requirements.
- Completion of the non-functional requirements.
  - Accuracy testing
  - Reliability testing
- Software should be free of errors.

7.3. Software Testing Methodology

A software can be plagued with minor and major bugs. Minor bugs can be fixed quickly before a software is released but the major bugs causes more damages and cannot be fixed quickly. The table down below is a summarization of the testing methodologies that were used for testing the prototype.

Testing No	During Development
TN1	Unit Testing (Back-box and White-box)
TN2	Integration Testing (White-box)
Testing No	After Development
TN3	Functional Testing (Black-box)

TN4	System Testing
TN5	Accuracy Testing
TN6	Usability Testing

Table 19: Summarization of Testing Methodologies

7.4. Execution Environment

Testing was performed in a computer with moderate performance specifications.

- Intel Core i7 / 2.2GHz
- 8GB RAM
- 80GB of free HDD space

7.5. Testing and Results

This section will explain the carried-out testing of "FixMe"

7.5.1. TN1: Unit Testing

Unit Testing was carried out parallel to the development phase of "FixMe". The reason for using this test method was its ability to isolate and fix bugs easily due to the code involved is smaller in the development phase than the final prototype. In this testing stage, various components of "FixMe" were individually tested. This testing method was carried out in both black box and white box testing.

7.5.2. TN2. Integration testing

Because unit testing took place in both black box testing and white box testing, integration testing was performed in white box testing. White box testing (also known as open box testing) is a method of testing internal structures of an application. It focuses on the system-wide data flow and ensures that it is accurate and efficient. Each component is aggregated with its associated components and test cases are conducted to verify the application. One of the reasons to conduct this testing method is its ability to reveal serious incompatibilities that may have been forgotten in developing various components. There are several methods to conduct an integration testing. They are Top Down, Bottom Up and Sandwich. The following factors were taken in to consideration when choosing a method to conduct integration testing.

- Low level core components are quicker to test because they depend on fewer other functions
- When it comes to data focused applications like "FixMe", it is easier to test the transmission of data as it allows greater transparency on what functions are performed and how the data is affected.

Based on these factors Bottom Up method was selected to tun integration testing.

7.5.3. TN3: Functional Testing

Functional testing is focused of testing the functionalities of the system. Functions are tested by inputting relevant data and examining the output. Functional testing is a form of black box testing and therefore the internal structure of the program is not considered in this testing method. Each of the functionality of "FixMe" will be tested one by while checking the overall success rate of the process. Summarized version of the functional testing is shown in the table down below. Refer Appendix:C for detailed description of the test cases.

ID	Function name	Example Test Case	Success Rate (%)
F1	Login	Wrong Username and Password	100
F2	Sign Up	Adding a preexisting account	100
F3	Upload Images	Uploading a corrupt image	100
F4	Classify Vehicle Damage	Try to classify without an Image	100
F5	Classify Damage Severity	Try to classify without an Image	100
F6	Price Check Components	Price check without giving vehicle details	100
F7	Calculate Damage Estimation	Try to calculate estimation without an image	100

Table 20: Summarization of Functional Testing

7.5.4. TN4: System Testing

In this sections, Functional requirements that were specified in the initial stages of the project was tested. Table down below summarizes the results of the system testing.

FR No.	Requirement Details	Priority	Status
1	Allows user to input images to the system	High	pass
2	Enhance the input image to remove noise and resize to make the processing easier.	High	pass
3	Analyze the image and extract valid features	High	pass
4	Train an image classifier using an image dataset.	High	pass
5	Allows the user to input vehicle details manually.	High	pass
6	Classify an image through pre trained algorithm.	High	pass
7	Calculate the severity of the damage.	High	pass
8	Get the price check of the damaged component of the vehicle.	High	pass
9	Calculate the damage estimation for severe damages using vehicle details.	High	pass

Table 21: Summarization of System Testing

7.5.5. TN5: Accuracy testing

This section will be discussing about the accuracy testing of "FixMe" application. 120 images were used to test the accuracy of the system. Table down below shows the some of test cases used to classify the vehicle damage. Refer Appendix:C for more detailed description of the test cases.

TC Id	Vehicle Component	Input Image Variant	Expected output	Accuracy (%)			Output	Status
				Severe	Dent	Scratch		
1	Hood	Small	Severe	89.08	2.4	1.4	Severe	Pass
2	Head Light	Large	Severe	60.9	28.6	10.4	Severe	Pass
3	Door	Bright	Scratch	3.1	21.5	75.6	Scratch	Pass
4	Windshield	Dark	Severe	11.8	28.9	60.1	Scratch	Fail

5	Hood	High-quality	Dent	13.4	72.2	14.8	Dent	Pass
6	Hood	Low-Quality	Severe	58.1	34.4	13.2	Severe	Pass

Table 22: Summarization of Accuracy Testing

7.5.6. TN7: Reliability Testing

Reliability testing is defined as a type of software testing that checks whether the software can run a failure-free operation in a specified environment for a specified period of time. Reliability testing in this context involves running individual test cases and ensuring that the same results are obtained consecutively. For this prototype, the reliability measure averaged at 75% Performance Testing

7.6. Limitations of Testing

Testing phase makes sure a software is error-free and ready for the deployment phase. But even the testing process have its limitations. Following factors are some of the limitation that author faced during the testing phase.

- Accuracy of the output is based on the data set. Therefore, identification of the severe damages were lot more accurate than minor and moderate damages.
- Testing disclose many errors but never guarantees the absence of errors
- Manual testing process least accuracy.
- With a rich data set of images, test results can be differ from the current results.

7.7. Chapter Summary

This chapter started off with a clear objective and criteria about software testing process and went through the prototype to test it out. Test cases were designed to test the system requirements. This chapter also ensures that the prototype is free of errors and that it works as expected. Errors that were founded was fixed and Limitations of the testing was discussed at the end of this chapter. The results that were obtained through this chapter will be evaluated in the next chapter.

Chapter 8: Evaluation

8.1. Chapter Overview

Previous chapter discussed about the testing phase of “FixMe” and this chapter focusses on the evaluation of “FixMe”. This Chapter discusses about how evaluation process is followed, the criteria of the evaluation and the self-evaluation of “FixMe”

8.2. Evaluation Criteria

Evaluation can be divided into two major components. Evaluation of the Project Process and the Prototype.

1. **Evaluation of the Project Process-** The evaluation of the project process discusses both development of the prototype and the progress of the project.
2. **Evaluation of the Prototype-** Evaluation of the prototype can be done in two ways. They are Quantitative and Qualitative evaluation. More details will be discussed under the Evaluation of the Prototype topic.

8.2.1. Evaluation criteria for the Project Process

Evaluation Criteria	Process and Purpose
The aim and the scope of the project “FixMe”	Opinions about the aim of “FixMe”. Opinions about the scope of the application “FixMe”.
System Design	To find out the system design can be further improved than the current one.
Technologies	To find out the most ideal technologies that can be used to implement the prototype.
Functionalities	To find out the concept of the system is delivered by the functionalities.
Limitations and future enhancements	Limitations and future enhancements are evaluated to find out possibilities and importance of the enhancements by adding them in to the system.
Accuracy, Usability and security	Accuracy, Usability and Security was evaluated to find out the performance of the application.

Table 23: Evaluation Criteria and the Purpose

8.3. Evaluation of the Project Process

First phase of the evaluation process is to evaluate the project process.

8.3.1. Evaluation of Objectives

The following objectives are taken in to consideration when evaluating the project process.

- Literature Review
- Project Management Methodology
- Data Gathering Requirements
- Software requirement specification
- Software Design
- Prototype Development
- Testing
- Evaluation
- Documentation

Literature Review

- Research on the current existing products and research papers were done and documented successfully.
- Research on the image classification was done and documented successfully.
- Research on existing algorithms and libraries was done and documented successfully.

Project Management Methodology

- Spiral model was selected as the software development model along with the PRINCE2 and justifications were done and documented successfully.

Data Gathering Requirements

- Requirements were gathered using questionnaires, online surveys, self-evaluations and through literature reviews.
- Requirements have been analyzed and suitable conclusions were documented successfully.

Software Requirement specification

- Gathered data was used to create functional and non-functional requirements.

Software Design

- Structured System Analysis and Design was selected as the design methodology to implement the system

- Relevant context diagrams and data flow diagrams of the proposed system were drawn.

Prototype Development

- Python was selected as the development language along with the TensorFlow framework to implement the prototype.
- Coding best practices were followed throughout the development process.
- Steps that were involved in the implementation was documented.

Testing

- A test plan was created to go through all the functions and non-functions.
- Functional requirements and non-functional requirements were tested and documented successfully.
- Bugs detected throughout the testing process were fixed.

Evaluation

- Appropriate evaluation selections were done and evaluated successfully.
- Evaluation feedback from the domain experts were taken and future enhancements were documented.

Documentation

- Documentation was done parallelly to the implementation of the prototype and final report was finalized and documented successfully.

8.3.2. Evaluation of Deliverables

Deliverables that was agreed prier to the implementation was evaluated and status of the deliverables was listed in the table down below.

Deliverables	Description	Status
Project Initiation Document	Initial project idea was created and delivered.	Delivered
Literature Review	Literature review that covers all the subject areas were documented and delivered.	Delivered
SRS	System requirements for the prototype was identified and delivered.	Delivered
Software Design Methodology	Design aspect of the prototype was identified and delivered.	Delivered
Interim Report	Interim report was delivered before the deadline.	Delivered
Prototype	Working prototype of the system was delivered	Delivered
Final Report	Finalized report of the project was delivered	Delivered

Table 24: Evaluation of Deliverables

8.4. Product Evaluation

After evaluating the project process, the quantitative and qualitative aspects of the prototype was evaluated, and the evaluation process is documented in this section.

8.4.1. Quantitative Evaluation

End user survey was made and sent across potential users to estimate the usability of the system. Users were asked to create an account on FixMe, upload an image, compare results with the damage estimation provided by the vehicle garages and give feedback about their experience. Analysis for the results are given below. Refer to Appendix : E for End user survey questions.

Question 1
How would you rate the usability of the system?
Aim
To get user feedback on the easiness of the UI.
Observation

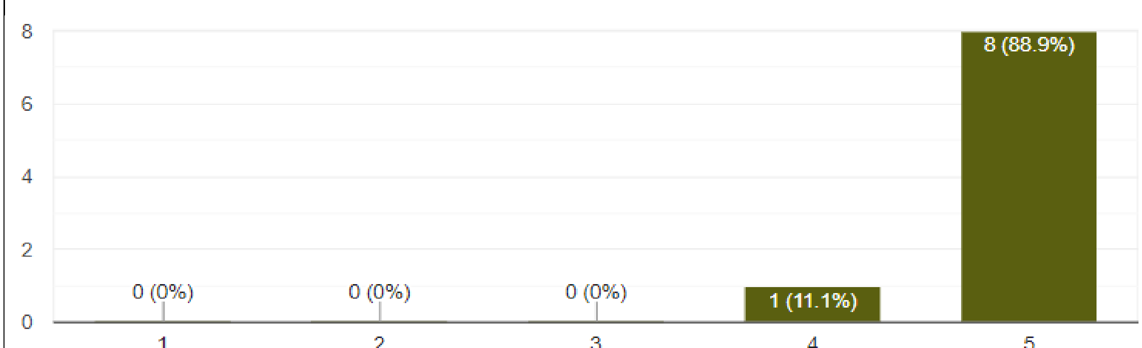
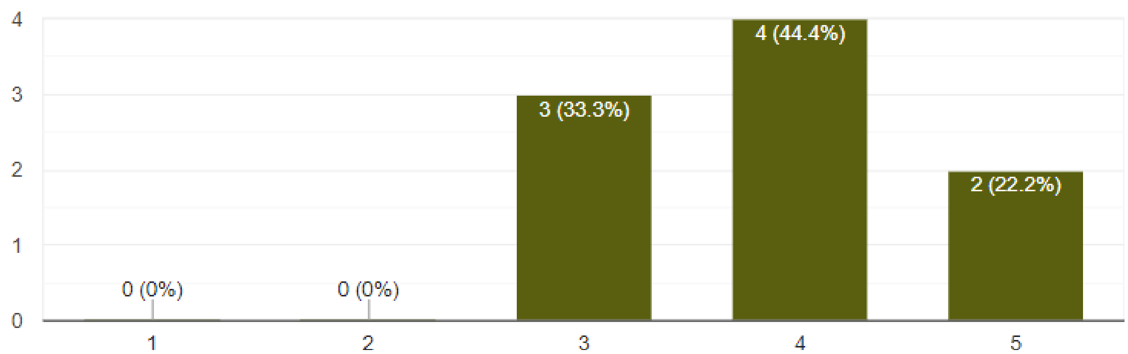
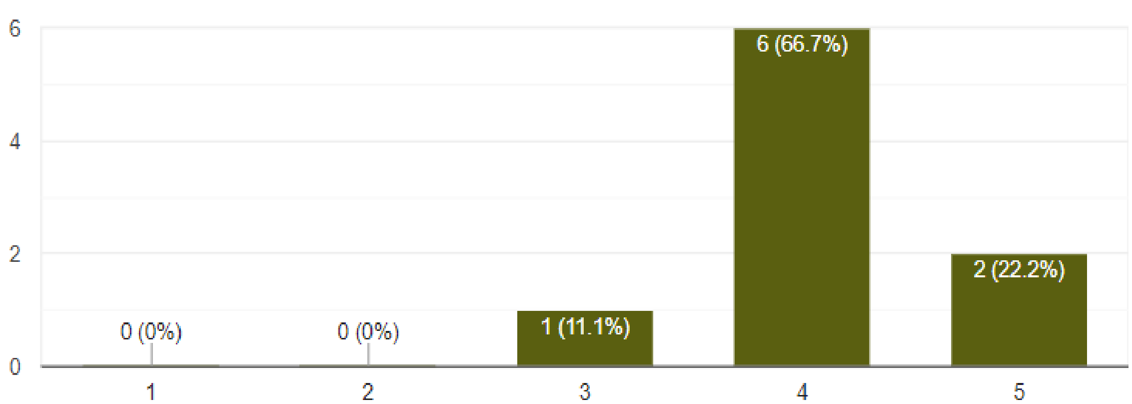


Figure 38: Evaluation Survey Result 1

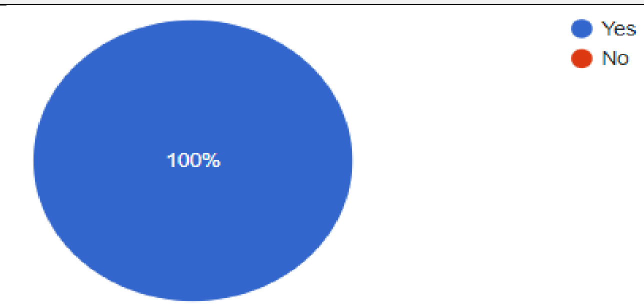
Analysis
88.9% of the users were very satisfied with the given UI and 11.1% of the users were satisfied. It is evident that most users find this product very easy to use.

Question 2
How would you rate the accuracy of the damage estimation given by "FixMe"?
Aim

"FixMe" only provides the minimum damage estimation for severe damages in a vehicle. Aim of this question is to get a comparison between the damage cost estimated by vehicle garages and the damage cost that was calculated by "FixMe"
Observation
 *Figure 39: Evaluation Survey Result 2*
Analysis
77.7% of users were somewhat satisfied with the result they got and only a 22.2% were satisfied of the result they got. Reason for this and additional information for this question was discussed in qualitative evaluation. (Question 2)

Question 3
In which scale do you think "FixMe" become helpful in damage estimation process?
Aim
To get the users opinion about the value of this type of applications.
Observation
 *Figure 40: Evaluation Survey Result 3*

Analysis
Majority of user's state that this type of application can become helpful when estimating the damage cost of a vehicle. Additional detail for this question was discussed in qualitative evaluation. (Question 3)

Question 4
Do you think this system is helpful to get a quick damage estimation of a damaged vehicle?
Aim
The main aim of the product "FixMe" is to get a quick damage estimation so that it can reduce the waiting time of the insurance agent after a collision happens. This question was asked to get the users opinion on the solution provided by "FixMe".
Observation
 *Figure 41: Evaluation Survey Result 4*
Analysis
All the users were agreed that "FixMe" is useful when conducting a quick damage estimation.

Question 5
How would you rate "FixMe"?
Aim
To get users likeness towards the application.
Observation

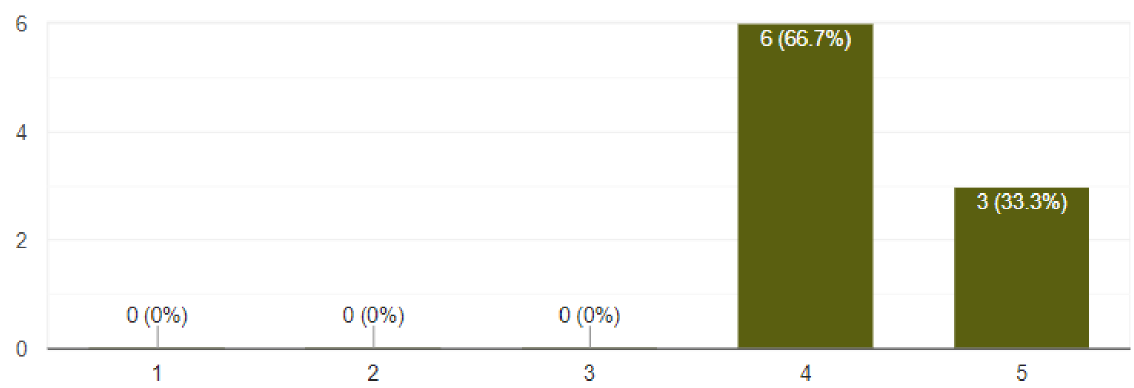


Figure 42: Evaluation Survey Result 5

Analysis
33.3% of the users were very satisfied with the product and 66.7% of users were satisfied with “FixMe”. It is evident that users are happy with the product and the qualitative information about the user feedback is discussed in Qualitative evaluation

8.4.2. Qualitative Evaluation

This section discussed about the qualitative evaluation process of the prototype and its findings.

Selection of the Evaluators

This section addresses the reason for selecting relevant evaluators for evaluating the prototype.

Evaluator	Description
General Users (Insurance agents/ Auto Mechanics)	Since the developed system is for the general users, system will be evaluated by few selected general users. This evaluation will be mainly focused on the aim and usability of the system.
Domain Experts	Few selected domain experts will be taken to evaluate the technologies that used to develop the system.
Software Engineers/Architects	System will be evaluated by few selected software engineers and architects in the industry. They will be evaluating all the previous criteria mentioned in this table.

Table 25: Selection Details of the Evaluators

Summary of Responses

Please be kind enough to contact the author to get details of the evaluators.

1) What are your thoughts on the prototype as a solution to the problem discussed in this project?

- *"The project concept of 'FixMe' meets a very important need of the society. Getting a quick damage estimation is indeed very helpful when reducing the time spent on the road after a collision."*

- Software Engineer

(CAMMS)

- "*Reducing the time spent on the road after a collision is very helpful for both the people on the road and the people who met with the accident. Therefore, calculating a quick damage estimate is indeed very helpful when it comes to insurance agents. The given solution addresses the problem very well."*

-Insurance Agent

- *"Given prototype is certainly impressive and interesting to use. It was able to accurately identify the damage type and give search results for the vehicle part prices."*

-Insurance Agent

Personal review: Most end users of "FixMe" were impressed by the concept and a positive feedback on the prototype was given as a solution for the problem in this project.

2) What are your thoughts on the prototype usability, reliability and accuracy?

- *"Prototype was able to accurately identify the damage severity. But for the damage cost, it only shows the market prices for each vehicle part. System is good for to get a quick estimation, but it would be better to add the repair cost also to the system."*

-Software Engineer

(CAMMS)

- *"Prototype was easy to use. User only have to select the image and it will be directly jump to the estimation process. 10/10 times it gave the correct damage severity and displayed prices for the vehicle part. Usability and reliability of the prototype is good. The given damage estimation for the damage and the damage estimation from the repair centers are different. Accuracy was around 65%."*

-Associate Software Engineer

(CAMMS)

- *"Prototype was easy to use. Damage severity was estimated successfully. Damage estimation was different from the repair center price. It would be better to include repair cost and delivery cost along with the price of the vehicle part for the damage estimation. System is good for a quick damage estimation"*

-Auto Mechanic

(Sajith Tyre Shop)

- *"prototype was easy to use. Damage severity was estimated successfully. Vehicle part prices was displayed correctly."*

-Insurance agent

Personal review: To estimate vehicle damage, it is necessary to include the vehicle part price, repair cost and the delivery charge. Repair center estimation include all these things and it is slightly different from the damage cost that was shown in "FixMe". "FixMe" only provides the minimum damage estimation for severe damages to get a quick idea of the damage that happened to the vehicle. Most of the users gave a good feedback on the quick damage estimation that was done by "FixMe"

3) Does "FixMe" become helpful when estimating the vehicle damages?

- *"It is helpful to find out the market prices for the vehicle parts. Without the tax, delivery cost and the repair cost, the damage estimation is not very reliable. It is good for a quick damage estimation."*

-Auto Mechanic
(Sajith Tyre Shop)

- *"It is helpful to get a quick damage estimation for the vehicle damage. It can reduce the time and money of insurance agents that are on the field."*

-Insurance agent

- *"Estimating vehicle damage need to have the price of the vehicle part in that area, repair cost and the time cost. Search results only give the global market values. It would be better to include those to the system to estimate vehicle damage. It is useful to get a quick damage estimation."*

- Software Engineer
(CAMMS)

Personal review: Estimating damage from the repair center perspective include the item price, time cost, repair cost and delivery cost. Both the delivery charge and the repair cost are depending on the repair center and the tax. Therefore, it is hard to give a good estimation for the damage directly. One evaluator point out that the price of the item depends on the area. It was taken into consideration for future enhancement.

4) Thoughts on limitations and future enhancements?

- *"System can be improved by adding local vehicle part prices. But it is hard to keep up to date with the prices. It would be good to manage a database rather than searching items across the web to get prices. System will be faster and can come up with a good UI to display results."*

- Software Engineer

(CAMMS)

- *"Estimating the critical damages is hard to perform due to the internal damages. It would be good to consider damage depth for those situations. Adding a that type of upgrade to find out critical damages will be awesome."*

-Associate Software Engineer

(CAMMS)

- *"It would be good to build a mobile application for the public, so they don't need to send their images to the insurance agencies and wait till it process. This method can save more time and you can overcome security issues by strengthening the security levels. Also, the insurance agencies can update the classification model when they needed and deploy it as updates in Playstore and Appstore."*

- Software Engineer

(CAMMS)

Personal review: Evaluators point out that adding critical damage estimation to the product will become more useful. Identifying the critical damage through images hard due to the fact that internal damages cannot be captured by images. But with the improvement in the image depth processing, it isn't impossible to give a minimum damage estimation for that. Calculating critical damage was taken in to consideration for future enhancements Another idea that come up in this session was to develop a app for the general public who drive vehicles. It is also taken in to consideration for future enhancements.

8.5. Self-Evaluation

Self-evaluation for each question was given as personal reviews at the end of each question and project process.

Maintaining a fixed aim is very important in the development process. Authors intension was to develop a system that can reduce the time wasted on streets after a collision happen. To address this problem a solution was made by the author to give a quick damage estimation for the damaged vehicle so that can reduce the travel time and money of the insurance agent. A development process was made to address to this scenario in this project.

Spiral model and PRINCE2 was selected as the software development and project management models because they proved to be the mode ideal models for the project. Survey questionnaires, past literatures and brain storming methods were used to gather system requirements because they proved to be the best way to gather information. TensorFlow was selected as the main technology due to its overall capabilities of performance and learnability.

After the development, the testing was done and displayed positive results for identifying the damage severity. And then the damage estimation process was done by searching relevant items in google search. Prototype was able to properly identify severities of a damage and give a quick estimation for the damage. By getting a quick estimation , insurance agents doesn't need appear at the collision location and they can check the authenticity of the damage without leaving the office. Aim of this project was to reduce time cost that spent on streets after a collision and considering every fact as a whole, the project "FixMe" can be declared as a success.

8.6. Chapter Summary

This chapter's objective was to evaluate and document the evaluation process of the project process and the prototype. Chapter started with a explanation of the evaluation criteria and the evaluation process was done by dividing the evaluation in to two separate evaluations. Evaluation of the project process was discussed at first and the prototype was discussed and documented after that. A self-evaluation was done at the end of the evaluation process.

Chapter 9: Conclusion

9.1. Chapter Overview

This final chapter will go through the whole process of the project briefly and summarize the findings and accomplishments of previous chapters. Aims, objectives and their achievement status will be highlighted. Problems and challenges faced will be discussed towards the end.

9.2. Achievement of Aims and Objectives

The Aim of the project is to "reduce the time that spent on the street after a vehicle collision by developing a system that can validate the damage and estimating a cost for that".

A complete research on the subject area was down and requirements were gathered using selected techniques. Solution for the problem was designed, developed and tested and evaluated within the given time range.

Objective Achievement

Objectives that were discussed throughout this project and their success status are shown in the table below.

Objective Id	Objective Description	Status
Objective 01	To identify damage severity levels in a damaged vehicle.	Success
Objective 02	To analyze severity levels and find out the most accurate severity of the damage.	Success
Objective 03	Generate a minimum damage estimation for severe damages.	Success
Objective 04	To increase the accuracy of identifying severe vehicle damages from images.	Success
Objective 05	To implement a system that can estimate severe damages in a vehicle from images and generate minimum damage cost estimation for the severe damages.	Success

Table 26: Project Objectives and Achivements

Identifying the severity levels and sorting out the most accurate severity level was successfully completed. Author witness the accuracy of the classification module can be increased by adding larger data set and increasing the training steps. Minimum damage estimation was given to the user by getting the relevant vehicle components from the user and performing web search. Due to these facts it can be safely declare that the objective no 01,02,03,04 and 05 were achieved successfully.

9.3. Problems and Challenges Faced

There were many challenges to face during the implementation process of this project and some of the solutions author had for this project went to a dead end. About the problems that author faced and the challenges will be discussed under this topic.

3D Segmentation to Address the Problem as a Solution

Authors first approach was to use 3d segmentation to address the problem as a solution. Author idea was to create a 3d mesh of the damaged vehicle and get the damage part as a 3d mesh by comparing it to a another 3d mesh model of an undamaged vehicle. Limiting the estimation for a single car variation, author was able develop a system to make a 3d mesh from images. The generated 3d mesh was not perfect. After struggling to create a 3d mesh, it was seen that the model generated cannot be used to compare damages. Figure:43 and 44 are the some of the pictures of 3d mesh that was generated by an undamaged vehicle. Refer Appendix: G for images that was used to generate this model.

Figure 43: 3d Mesh Model 1

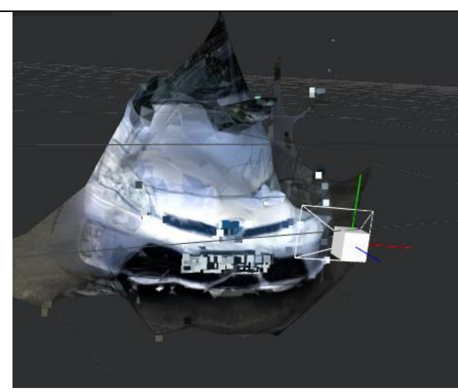


Figure 44: 3d Mesh Model 2

Lighting effects in the area, the surrounding small objects in the area, photo quality and many more effects were blocking the path of approaching this solution. Due to the limitation of the libraries and documents to this solution, this approach took a dead end.

Damage Cost Estimation

One of the problems author witness is when making the minimum damage cost estimation. At first the repair cost and item cost were taken in to the consideration for cost estimation but after conducting a research it was seen that the repair cost depends on the repair center. After conducting a research on what are the end user requirements, author decided only to take the vehicle component price for the minimum damage estimation.

In the evaluation, most of the users were very satisfied with the minimum damage estimation results but most of them said that it would be better to add the repair cost also.

Another problem author faced is that the price results were taken from the global market. Component prices can be slightly different due to the tax and delivery charges in each country. These results depend on the tax rate of each country. Therefore, the results can be cause accuracy change depending the end user's country.

Lack of Information

No proper products were released to the market to investigate and the current existing products also not provide the enough information to understand the product.

Time

Author was first focused on using the 3d segmentation to address the solution and took couple of months following that solution till it came to a dead end. Because of that the time spent on the new solution was limited to the author.

Product Prices

Available APIs to get the component prices were not free and at the end author had to direct the end user to a web search.

9.4. Learning Outcomes and New Skills Learnt

Lessons learned during the project process was unmeasurable. Lot of lessons were learned when overcoming the challenges author faced and the list below are the lessons and new skills that author learned during the project process.

Project management- Project management aspect was studied and applied to a project for the first time. It was a good experience and was able to learn lot of things related to conducting a proper project management.

Performing a literature review- Author was already familiar with the importance of a literature review and the knowledge of performing a literature review was increased.

Conducting surveys- conducted an online survey for a first time and understood the importance of the data gathering from surveys.

Web scraping techniques- For the first time, conducted web scraping in google search to get images of damaged vehicles and also learned how to do a word scraping in a web page.

Process of creating a classification model - Author was new to the deep learning techniques and this was a good opportunity to learn about that. For the first time, Author created a classification model to classify the severity of the vehicle damage and the experience that was able to get was priceless.

Meetings with the experts- Knowledge on conducting meetings with the experts were increased and learned how to properly address people in the industry and ask questions from them.

Documentation skills- Documentation skills were improved.

9.5. Limitations of the Solution

Cost estimation limitation 1- current existing literature provides solutions for minor and moderate damages. Therefore, author turned the main focuse of this project towards estimating the minimum damage cost for severe damages.

Cost estimation limitation 2- Minimum damage estimation depends on the tax rate of the end users. Therefore, accuracy can be change from the point of users country.

Critical damages- Internal damages cannot capture buy an image; therefore, the solution cannot classify critical damages.

Dataset limitation- There isn't any available data set for the damaged vehicle and user had to perform web scraping methods to get images.

Limitation of classifier – due to the dataset limitation, the accuracy of the trained classifier was reduced.

9.6. Future Enhancements

A software cannot be developed with a 100% successful rate. Every software has its own limitations and has solutions to overcome those limitations. Following table is a list of enhancements that can be used to improve "FixMe"

Enhancement	Description
Multiple image classification	System can be further improved to do multiple image classifications at the same time.
Better classifier	The Classifier can further improve by adding a larger dataset.
Increase accuracy and performance	Multiple classifiers can be used to increase the accuracy of the system.
Damage cost estimation	Minimum damage cost can be further improved by making an equation to calculate the repair cost and tax depending on the user's country.
Mobile platforms	Enabling the application to run through mobile platforms and making the general vehicle users as the target end users can be useful to reduce more time wasted on the streets after a collision happen.
Security	Authentication process can be further improved if the application was developed for mobile platforms. (Fingerprint, face detection, etc.)
Vehicle and component classification	A process to classify the vehicle type and the component can be added to the system to furthermore improve the quality of the service that provide by the application.

Table 27:Future Enhancements and Descriptions

9.7. Concluding Remarks

Primary aim of "FixMe" was to reduce the time that spent on the street after a vehicle collision and the solution was to develop a system that can validate the damage and estimate a damage cost so that the evaluation agent from insurance company won't have spent time and money by travelling to the collision location. The development of the solution was positively completed, and the research objectives were successfully obtained. Due to all of the above facts, this project can be considered a success.

References

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105.Curran Associates, Inc., 2012.

- American-autobody.com. (2018). Submit Vehicle Damage, Receive Estimate. [online] Available at: https://www.american-autobody.com/services/get-our-app/ [Accessed 27 Oct. 2018].

- ARAUJO, M. (2018). How to File an Auto Insurance Claim in 5 Easy Steps. [online] Available at: https://www.thebalance.com/filing-auto-insurance-claim-2645925 [Accessed 30 Dec. 2018].

- Bengio Y. Lecun Y., Bottou L. and Haffner P., "Gradient-based learning applied to document recognition,"Proceedings of IEEE, vol. 86, no. 11, 1998.

- Belaid L. J., Mourou W. (2009) - Image Segmentation: A Watershed Transformation Algorithm

- Bodyshop.systems. (2018). Auto Body App. [online] Available at: http://www.bodyshop.systems/guidesktop.php?Customer=&BodyShopID=83&LocationID=0 [Accessed 30 Dec. 2018].

- Chablani, M. (2017). Autoencoders — Introduction and Implementation in TF.. [online]. Available at: https://towardsdatascience.com/autoencoders-introduction-and-implementation-3f40483b0a85 [Accessed 30 Dec. 2018].

- Controlexpert. (n.d.).. EasyClaim - Control€xpert. [online] Available at: https://www2.controlexpert.com/uk-en/highlights/easyclaim/ [Accessed 27 Oct. 2018].

- CQUniversity, n.d. Literature Review Tutorial. [Online]
  Available at: http://libguides.library.cqu.edu.au/litreview
  [Accessed 23 04 2019].

- Dumitru Erhan, Yoshua Bengio, Aaron Courville,Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio, "Why does unsupervised pre-training help deep learning?," Journal of Machine Learning Research, vol. 11, no. Feb, pp. 625–660, 2010.

- Ernst & Young LLP. (2015). Does your organization need to complete a Claims Leakage study. [online]
  Available at: https://www.ey.com/Publication/vwLUAssets/ey-does-your-firm-need-a-claims-leakage-study/$FILE/ey-does-your-firm-need-a-claims-leakage-study.pdf [Accessed 27 Oct. 2018].

- Gocanvas. (2018). Automotive Estimate. [online]
  Available at: https://www.gocanvas.com/mobile-forms-apps/459-Automotive-Estimate
  [Accessed 27 Oct. 2018].

- Google Cloud. (2019). Advanced Guide to Inception v3 on Cloud TPU | Cloud TPU | Google Cloud. [online]
  Available at: https://cloud.google.com/tpu/docs/inception-v3-advanced
  [Accessed 4 May 2019].

- Gontscharova,S. Baumgärtela,H. Kneifela,A. and Kriegera,K.(2014).(PDF) Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing[online] Sciencedirect.
  Available at: https://www.sciencedirect.com/science/article/pii/S2212017314001340 [Accessed 27 Oct. 2018]

- Harshani, R. (2017). [online] Available at:
  https://www.researchgate.net/publication/319793711_Image_Processing_Based_Severity_and_Cost_Prediction_of_Damages_in_the_Vehicle_Body_A_Computational_Intelligence_Approach
  [Accessed 28 Oct. 2018].

- Jayawardana, S. (2013). (PDF) Image Based Automatic Vehicle Damage Detection. [online] ResearchGate. Available at: https://www.researchgate.net/publication/263619076_Image_Based_Automatic_Vehicle_Damage_Detection [Accessed 27 Oct. 2018].

- Jonathan Masci, Ueli Meier, Dan Cires¸an, and J¨urgen Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in International Conference on Artificial Neural Networks. Springer, 2011, pp. 52–59.

- Karami, E., Prasad, S. and Shehata, M. (2016). (PDF) Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images.[online] Available at: https://arxiv.org/pdf/1710.02726 [Accessed 27 Oct. 2018].

- Ministry of transport. (2018). Statistics. [online] Transport.gov.lk. Available at: http://www.transport.gov.lk/web/index.php/statistics/national-council-for-road-safety.html [Accessed 28 Oct. 2018].

- TensorFlow. (2019). Transfer Learning Using Pretrained ConvNets | TensorFlow Core 2.0a | TensorFlow. [online] Available at: https://www.tensorflow.org/alpha/tutorials/images/transfer_learning [Accessed 5 May 2019].

- Tfhub.dev. (2019). TensorFlow Hub.[online] Available at: https://tfhub.dev/google/imagenet/inception_v3/classification/1 [Accessed 5 May 2019].

- The Ding Stinger. (n.d.). The Ding Stinger | Auto Body Repair Estimating App. [online] Available at: https://www.thedingstinger.com/the-ding-stinger-auto-body-repair-estimating-app/ [Accessed 27 Oct. 2018].

- Khoo, B. K. S., 2011. A Survey of Major Software Design Methodologies. [Online] Available at: http://userpages.umbc.edu/~khoo/survey2.html [Accessed 27 Oct. 2018].

- OpenSource.com, 2016. What is open source. [Online]
  Available at: https://opensource.com/resources/what-open-source
  [Accessed 07 04 2019].

- Pan, J., 1999. Carnegie Mellon University, 18-849b Dependable Embedded Systems. [Online]
  Available at: https://users.ece.cmu.edu/~koopman/des_s99/sw_testing/ [Accessed 03 01 2019].

- Parasuraman, S., Thamby Sam, A., Wong Kah Yee, S. and Lau Chik Chuon, B.
  (2017). Smartphone usage and increased risk of mobile phone addiction [online]
  Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5680647/
  [Accessed 5 Apr. 2019].

- PRINCE2, 2015. PRINCE2 Methodology. [Online]
  Available at: https://www.prince2.com/uk/prince2-methodology
  [Accessed 20 11 2015].

- Project Management Institute, 2015. On-Demand Webinars. [Online]
  Available at: http://www.projectmanagement.com/Webinars/webinarMainOnDemand.cfm
  [Accessed 10 09 2015].

Appendix

Appendix A: Gantt Chart

Figure 45: Guntt chart of "FixMe" project

Appendix B: UI Designs

UI Design: Login

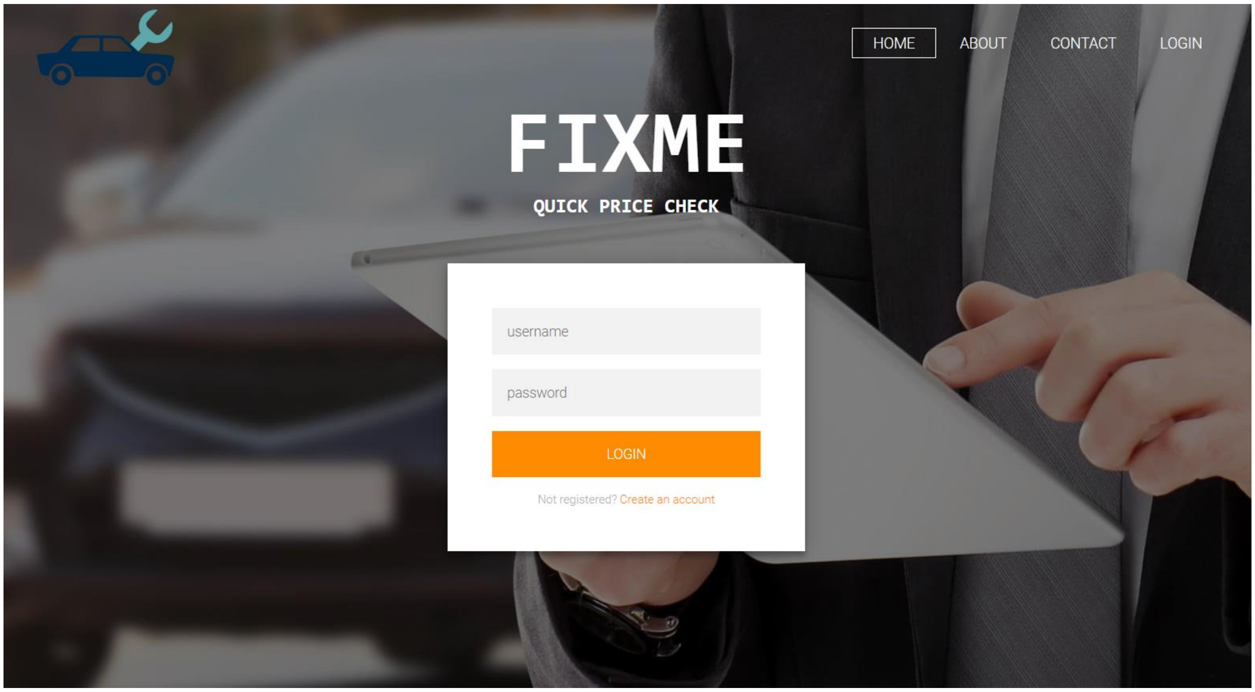


Figure 46: UI Design : Login

UI Design: Sign Up

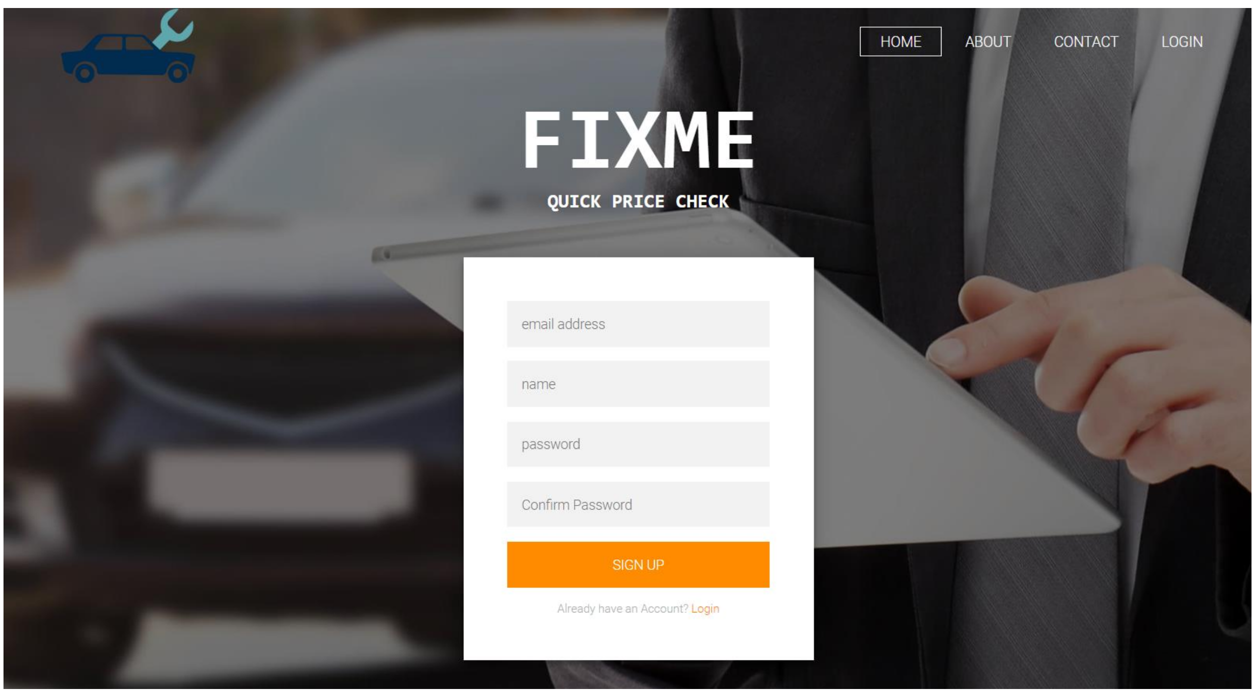


Figure 47: UI Design : SignUp

UI Design: Homepage

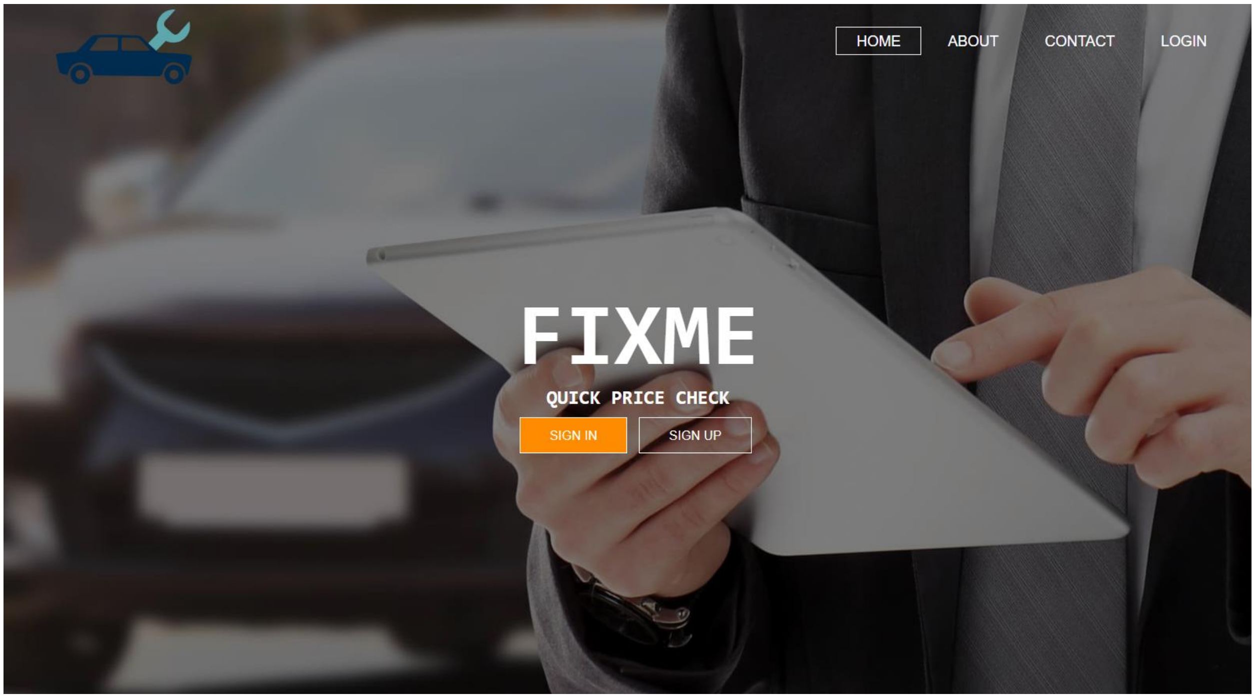


Figure 48: UI Design : HomePage

UI Design: FixMe Damage Estimation-1

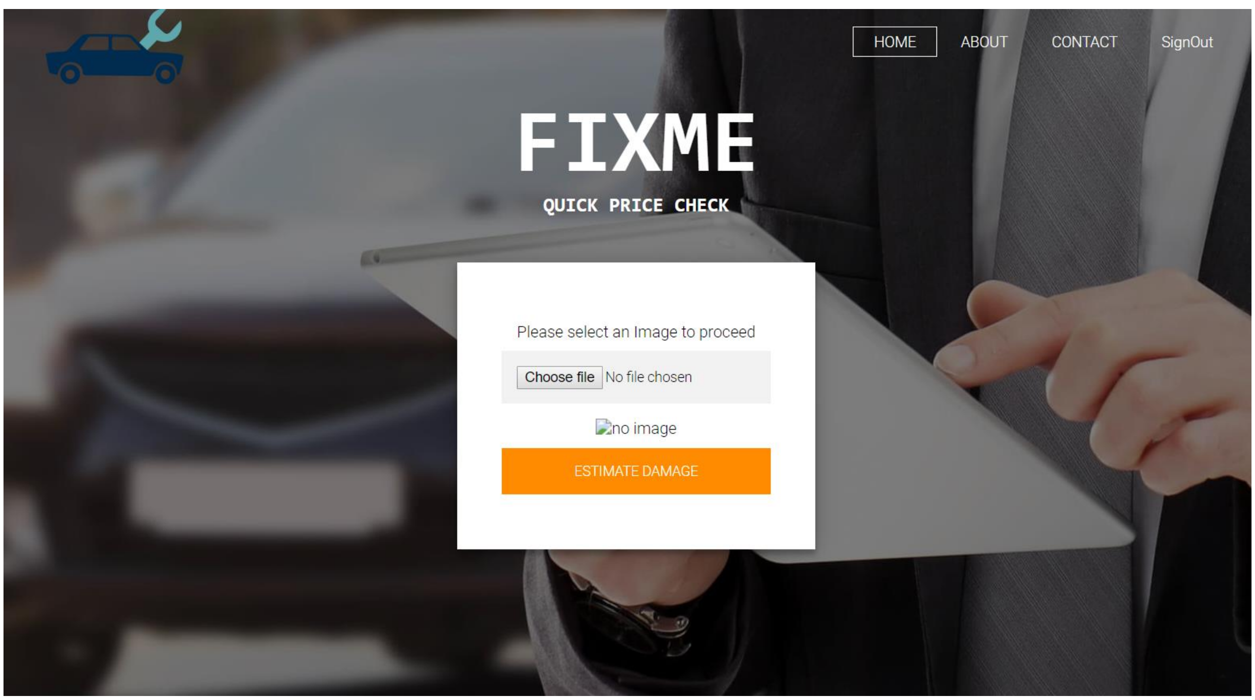


Figure 49: UI Design : Damage estimation-1

UI Design: FixMe Damage Estimation-2

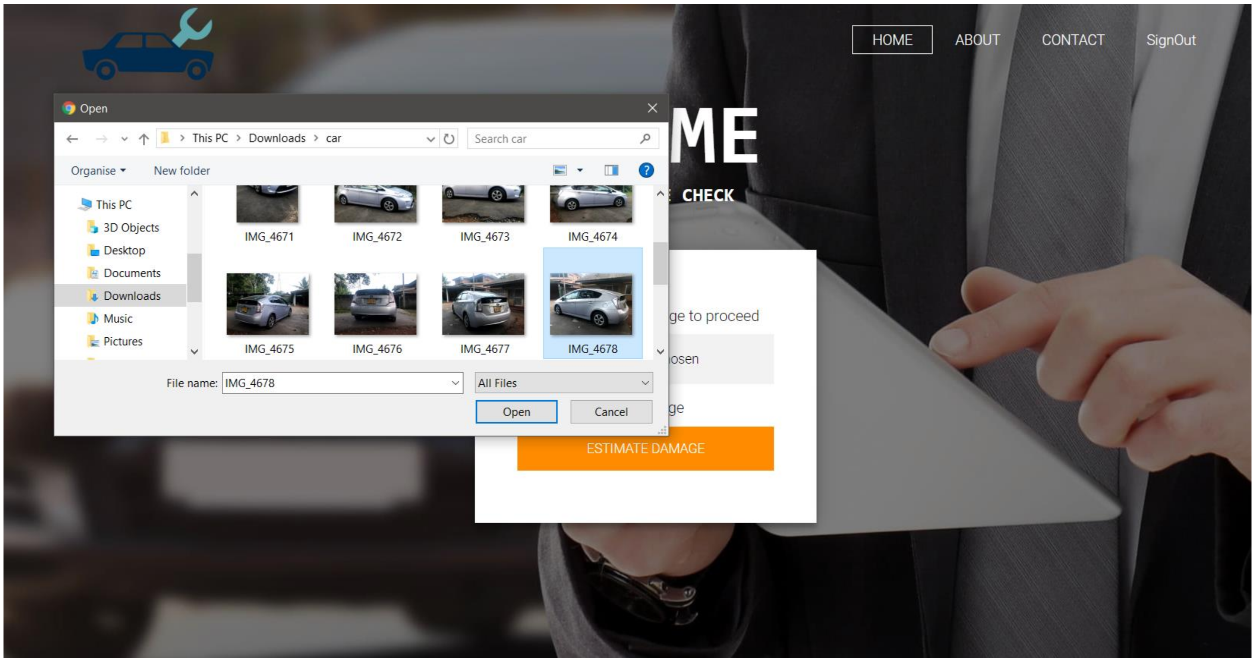


Figure 50UI Design : Damage estimation-2

UI Design: FixMe Damage Estimation-3

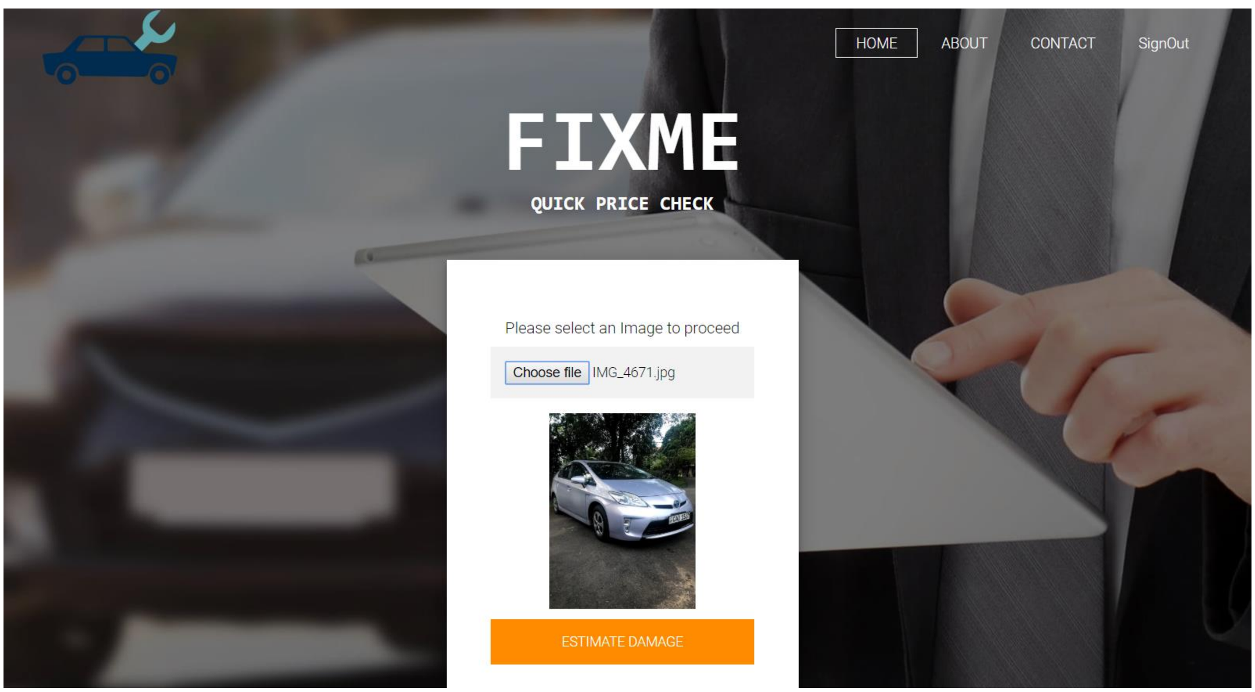


Figure 51: UI Design: Damage estimation-3

Appendix C: Use Case Descriptions

Use Case: Sign In

Use Case	**Sign in**
Id	004
Description	Allows user to sign in to the system
Primary actor	User
Supporting actor	none
Pre-condition	User needs to be registered
Post-condition	User will have access to adding income, expenses, reviewing and generating financial predictions.
Trigger	User wishes to sign in to the system
Main Success scenario	5) user opens the app. 6) User selects sign in 7) User authentication 8) Grants user access
Variations	If user authentication fails, a message will be shown. "register for a new user/ try again"

Table 28: Use Case Description: Sign In

Use Case: Sign Up

Use Case	**Sign up**
Id	005
Description	Allows user to register a new user in to the system
Primary actor	User
Supporting actor	none
Pre-condition	none
Post-condition	Allows user to sign in as a registered user
Trigger	User wishes to sign up
Main Success scenario	5) User opens app 6) User selects sign up 7) Authenticate new sign up 8) Register user
Variations	If there is already a user, message will be shown "Username or email already exists"

Table 29: Use Case Description: Sign Up

Use Case: Price Check Components

Use Case	**Price Check Components**
Id	006
Description	Allows system to check the prices of the vehicle components.
Primary actor	User
Supporting actor	none
Pre-condition	Vehicle details should be gathered before checking the price.
Post-condition	Give the price check to the system.
Trigger	User wishes to get the price check for the severity.
Main Success scenario	9) User selects the Get a quick price check. 10) Prices will be searched based on the given vehicle information 11) Prices will be shown.
Variations	1) User cannot proceed if the result damage severities are minor and moderate. System will only perform the price check for severe damages.

Table 30: Use Case Description: Price Check Components

Use Case: Upload Images

Use Case	**Upload Images**
Id	007
Description	Allows user to upload images.
Primary actor	User
Supporting actor	System
Pre-condition	User should be logged in to the system.
Post-condition	Proceed to classifying damage types.
Trigger	User wishes to upload an image.
Main Success scenario	1) User selects the Image. 2) Press proceed to go to the next scenario.
Variations	1) Error message will be shown when uploading corrupted images 2) Error message will be shown when uploading non image files

Table 31: Use Case Description: Upload Images

Use Case: Get Vehicle Data

Use Case	**Get Vehicle Data**
Id	008
Description	Allows user to input vehicle details and validates the detail entered by the user.
Primary actor	User
Supporting actor	System
Pre-condition	User should be logged in to the system.
Post-condition	Proceed to Price Check Components.
Trigger	User wishes to get a quick price check.
Main Success scenario	1) User inputs the vehicle details. 2) Validate vehicle details 3) Proceed to Price Check Component.
Variations	1) Error message will be shown for false data

Table 32: Use Case Description: Get Vehicle Data

Use Case: Display Damage Estimation

Use Case	**Display Damage Estimation**
Id	009
Description	Allows system to display damage cost.
Primary actor	System
Supporting actor	none
Pre-condition	Damage severity and the vehicle data need to be gathered.
Post-condition	None.
Trigger	Calculate Damage
Main Success scenario	1) Get the relevant vehicle component URL. 2) URL will be used to display the current price check
Variations	1) Error message will be shown for not getting any data for the URL

Table 33: Use Case Description: Display Damage Estimation

Appendix D: Test Cases

Functional Testing of "FixMe"

TC ID	Description	Expected Output	Actual Output	Result
F1	**Login**			
	Wrong username and password	Error Message to recheck username and password	Error Message to recheck username and password	Pass
	Correct username and password	User can login to the system	User can login to the system	Pass
	Username only	User can login to the system	User can login to the system	Pass
	Password only	User can login to the system	User can login to the system	Pass
F2	**Sign Up**			
	Adding a preexisting account	Error Message to remind user of an existing account	Error Message to remind user of an existing account	Pass
	Adding a non-preexisting account	User can register the account	User can register the account	Pass
F3	**Upload Images**			
	Without selecting an image	Error message to remind user to upload an image	Error message to remind user to upload an image	Pass
	Uploading a corrupt image	Error message: File can't be opened	Error message: File can't be opened	Pass
	Uploading a PNG image	Proceed to classifying process	Proceed to classifying process	Pass
	Uploading a JPEG image	Proceed to classifying process	Proceed to classifying process	Pass
F4	**Classify Vehicle Damage**			
	Classify without an image	Error message to notify to upload an image	Error message to notify to upload an image	Pass
	Classify without a dataset	Error message to notify to add dataset location	Error message to notify to add dataset location	Pass
	Classify only with severe damages	Classifying process will continues and accuracy of image being a severe damage will be given	Classifying process will continues and accuracy of image being a severe damage will be given	Pass
	Classify only with moderate damages	Classifying process will continues and accuracy of image being a moderate damage will be given	Classifying process will continues and accuracy of image being a moderate damage will be given	Pass

	Classify only with minor damages	Classifying process will continues and accuracy of image being a minor damage will be given	Classifying process will continues and accuracy of image being a minor damage will be given	Pass
	Classify with a rich dataset	Classifying process will continues and accuracy of image being a severe, moderate and minor damages will be given	Classifying process will continues and accuracy of image being a severe, moderate and minor damages will be given	Pass
F5	**Classify Damage Severity**			
	Without minor damage accuracy	Highest type of the damage type will be given	Highest type of the damage type will be given	Pass
	Without moderate damage accuracy	Highest type of the damage type will be given	Highest type of the damage type will be given	Pass
	Without severe damage accuracy	Highest type of the damage type will be given	Highest type of the damage type will be given	Pass
	Without damage classification	Error message to notify user that the classification process did not happen.	Error message to notify user that the classification process did not happen.	Pass
	With damage classification	Highest type of the damage type will be given	Highest type of the damage type will be given	Pass
F6	**Price Check Components**			
	Vehicle Hood price check	The relevant price will be given	The relevant price will be given	Pass
	Vehicle Door price check	The relevant price will be given	The relevant price will be given	Pass
	Vehicle Headlight price check	The relevant price will be given	The relevant price will be given	Pass
	Vehicle Mirrors price check	The relevant price will be given	The relevant price will be given	Pass
F7	**Calculate Damage Estimation**			
	Calculate without price check	Error message to show the missing vehicle data	Error message to show the missing vehicle data	Pass
	Calculate without severity	Error message to show severity miss match	Error message to show severity miss match	Pass
	Calculate with severity and price check	Display damage estimation to user	Display damage estimation to user	Pass

Table 34: Functional Testion of "FixMe"

Vehicle accuracy testing

Accuracy testing of Hood Component of the vehicle

TC Id	Vehicle Component	Input Image Variant	Expected output	Accuracy (%)			Output	Status
				Severe	Dent	Scratch		
1	Hood	Small	Severe	89.08	2.4	1.4	Severe	Pass
2	Hood	Large	Severe	60.9	28.6	10.4	Severe	Pass
3	Hood	Bright	Scratch	3.1	21.5	75.6	Scratch	Pass
4	Hood	Dark	Severe	11.8	28.9	60.1	Scratch	pass
5	Hood	High-quality	Dent	13.4	72.2	14.8	Dent	Pass
6	Hood	Low-Quality	Severe	58.1	34.4	13.2	Severe	Pass

Table 35: Accuracy Testing of the Component: Hood

Accuracy Testing of Head Light Component of the Vehicle

TC Id	Vehicle Component	Input Image Variant	Expected output	Accuracy (%)			Output	Status
				Severe	Dent	Scratch		
1	Head Light	Small	Severe	89.08	2.4	1.4	Severe	Pass
2	Head Light	Large	Severe	60.9	28.6	10.4	Severe	Pass
3	Head Light	Bright	Scratch	3.1	21.5	75.6	Scratch	Pass
4	Head Light	Dark	Severe	11.8	28.9	60.1	Scratch	Fail
5	Head Light	High-quality	Dent	13.4	72.2	14.8	Dent	Pass
6	Head Light	Low-Quality	Severe	58.1	34.4	13.2	Severe	Pass

Table 36: Accuracy Testing of the Component: HeadLight

Accuracy Testing of Door Component of the Vehicle

TC Id	Vehicle Component	Input Image Variant	Expected output	Accuracy (%)			Output	Status
				Severe	Dent	Scratch		
1	Door	Small	Severe	89.08	2.4	1.4	Severe	Pass
2	Door	Large	Severe	60.9	28.6	10.4	Severe	Pass
3	Door	Bright	Scratch	3.1	21.5	75.6	Scratch	Pass
4	Door	Dark	Severe	11.8	28.9	60.1	Scratch	Fail
5	Door	High-quality	Dent	13.4	72.2	14.8	Dent	Pass
6	Door	Low-Quality	Severe	58.1	34.4	13.2	Severe	Pass

Table 37: Accuracy Testing of the Component: Door

Accuracy Testing of Windshield Component of the Vehicle

TC Id	Vehicle Component	Input Image Variant	Expected output	Accuracy (%)			Output	Status
				Severe	Dent	Scratch		
1	Windshield	Small	Severe	89.08	2.4	1.4	Severe	Pass
2	Windshield	Large	Severe	60.9	28.6	10.4	Severe	Pass
3	Windshield	Bright	Scratch	3.1	21.5	75.6	Scratch	Pass
4	Windshield	Dark	Severe	11.8	28.9	60.1	Scratch	Fail
5	Windshield	High-quality	Dent	13.4	72.2	14.8	Dent	Pass
6	Windshield	Low-Quality	Severe	58.1	34.4	13.2	Severe	Pass

Table 38: Accuracy Testing of the Component: Windshield

Appendix E: End User Survey Questions

The below questions were passed across to end users to evaluate the product. Made from the google forms.

1. How would you rate the usability of the system?

   1[]   2[]   3[]   4[]   5[]

2. How would you rate the accuracy of the damage estimation given by "FixMe"?

   1[]   2[]   3[]   4[]   5[]

3. Additional feedback on accuracy of "FixMe"

   *paragraph

4. In which scale do you think "FixMe" become helpful in damage estimation process?

   1[]   2[]   3[]   4[]   5[]

5. Additional feedback on helpfulness in damage estimation

   *paragraph

6. Do you think this system is helpful to get a quick damage estimation of a damaged vehicle?

   Yes   /   No

7. How would you rate "FixMe"?

   1[]   2[]   3[]   4[]   5[]

8. Any additional feedback regarding "FixMe".

   *paragraph

Appendix F: Implementation Screenshots

create_image_lists() function

```python
        if len(file_list) < 20:
            tf.logging.warning('WARNING: Folder has less than 20 images, which may cause issues.')
        elif len(file_list) > MAX_NUM_IMAGES_PER_CLASS:
            tf.logging.warning('WARNING: Folder {} has more than {} images. Some images will never be selected.'.format(dir_name, MAX_NUM_IMAGES_PER_CLASS))

        label_name = re.sub(r'[^a-z0-9]+', ' ', dir_name.lower())
        training_images = []
        testing_images = []
        validation_images = []
        for file_name in file_list:
            base_name = os.path.basename(file_name)
            hash_name = re.sub(r'_nohash_.*$', '', file_name)
            hash_name_hashed = hashlib.sha1(compat.as_bytes(hash_name)).hexdigest()
            percentage_hash = ((int(hash_name_hashed, 16) % (MAX_NUM_IMAGES_PER_CLASS + 1)) * (100.0 / MAX_NUM_IMAGES_PER_CLASS))
            if percentage_hash < validation_percentage:
                validation_images.append(base_name)
            elif percentage_hash < (testing_percentage + validation_percentage):
                testing_images.append(base_name)
            else:
                training_images.append(base_name)
        result[label_name] = {'dir': dir_name, 'training': training_images, 'testing': testing_images, 'validation': validation_images,}
    return result
```

Figure 52: Screenshot for the Code - Image List Creation

Training Steps

```python
for i in range(HOW_MANY_TRAINING_STEPS):
    # Get a batch of input bottleneck values, either calculated fresh every
    # time with distortions applied, or from the cache stored on disk.
    if doDistortImages:
        (train_bottlenecks, train_ground_truth) = get_random_distorted_bottlenecks(sess, image_lists, TRAIN_BATCH_SIZE, 'training',
                                                                                   TRAINING_IMAGES_DIR, distorted_jpeg_data_tensor,
                                                                                   distorted_image_tensor, resized_image_tensor, bottleneck_tensor)
    else:
        (train_bottlenecks, train_ground_truth, _) = get_random_cached_bottlenecks(sess, image_lists, TRAIN_BATCH_SIZE, 'training',
                                                                                   BOTTLENECK_DIR, TRAINING_IMAGES_DIR, jpeg_data_tensor,
                                                                                   decoded_image_tensor, resized_image_tensor, bottleneck_tensor,
                                                                                   ARCHITECTURE)
    # end if

    # Feed the bottlenecks and ground truth into the graph, and run a training
    # step. Capture training summaries for TensorBoard with the `merged` op.
    train_summary, _ = sess.run([merged, train_step], feed_dict={bottleneck_input: train_bottlenecks, ground_truth_input: train_ground_truth})
    train_writer.add_summary(train_summary, i)

    # Every so often, print out how well the graph is training.
    is_last_step = (i + 1 == HOW_MANY_TRAINING_STEPS)
    if (i % EVAL_STEP_INTERVAL) == 0 or is_last_step:
        train_accuracy, cross_entropy_value = sess.run([evaluation_step, cross_entropy], feed_dict={bottleneck_input: train_bottlenecks, ground_truth_input: train_ground_truth})
        tf.logging.info('%s: Step %d: Train accuracy = %.1f%%' % (datetime.now(), i, train_accuracy * 100))
        tf.logging.info('%s: Step %d: Cross entropy = %f' % (datetime.now(), i, cross_entropy_value))
        validation_bottlenecks, validation_ground_truth, _ = (get_random_cached_bottlenecks(sess, image_lists, VALIDATION_BATCH_SIZE, 'validation',
                                                                                            BOTTLENECK_DIR, TRAINING_IMAGES_DIR, jpeg_data_tensor,
                                                                                            decoded_image_tensor, resized_image_tensor, bottleneck_tensor,
                                                                                            ARCHITECTURE))
        # Run a validation step and capture training summaries for TensorBoard with the `merged` op.
        validation_summary, validation_accuracy = sess.run(
            [merged, evaluation_step], feed_dict={bottleneck_input: validation_bottlenecks, ground_truth_input: validation_ground_truth})
        validation_writer.add_summary(validation_summary, i)
        tf.logging.info('%s: Step %d: Validation accuracy = %.1f%% (N=%d)' % (datetime.now(), i, validation_accuracy * 100, len(validation_bottlenecks)))
    # end if

    # Store intermediate results
    intermediate_frequency = INTERMEDIATE_STORE_FREQUENCY

    if (intermediate_frequency > 0 and (i % intermediate_frequency == 0) and i > 0):
        intermediate_file_name = (INTERMEDIATE_OUTPUT_GRAPHS_DIR + 'intermediate_' + str(i) + '.pb')
        tf.logging.info('Save intermediate result to : ' + intermediate_file_name)
        save_graph_to_file(sess, graph, intermediate_file_name)
```

Figure 53: Screenshot for the Code -Steps of Training the Model

Log details

Creating image lists

```
creating image lists . . .
INFO:tensorflow:Looking for images in 'dents'
INFO:tensorflow:Looking for images in 'scraches'
INFO:tensorflow:Looking for images in 'severe'
```

Figure 54: Screenshot for the Log Detail- Creating Image list

Perform training

```
performing training . . .
INFO:tensorflow:2019-05-04 18:09:11.075542: Step 0: Train accuracy = 43.0%
INFO:tensorflow:2019-05-04 18:09:11.075542: Step 0: Cross entropy = 1.045664
INFO:tensorflow:2019-05-04 18:09:15.231438: Step 0: Validation accuracy = 26.0% (N=100)
INFO:tensorflow:2019-05-04 18:09:16.704508: Step 10: Train accuracy = 69.0%
INFO:tensorflow:2019-05-04 18:09:16.704508: Step 10: Cross entropy = 0.961385
INFO:tensorflow:2019-05-04 18:09:16.843141: Step 10: Validation accuracy = 60.0% (N=100)
INFO:tensorflow:2019-05-04 18:09:18.257349: Step 20: Train accuracy = 76.0%
INFO:tensorflow:2019-05-04 18:09:18.257349: Step 20: Cross entropy = 0.833879
INFO:tensorflow:2019-05-04 18:09:18.394995: Step 20: Validation accuracy = 59.0% (N=100)
INFO:tensorflow:2019-05-04 18:09:19.805215: Step 30: Train accuracy = 78.0%
```

Figure 55: Screenshot for the Log Detail- Training the Classification Model

```
INFO:tensorflow:2019-05-04 18:12:00.507030: Step 999: Train accuracy = 100.0%
INFO:tensorflow:2019-05-04 18:12:00.507030: Step 999: Cross entropy = 0.131229
running testing . . .
INFO:tensorflow:2019-05-04 18:12:00.648623: Step 999: Validation accuracy = 78.0% (N=100)
writing trained graph and labbels with weights
INFO:tensorflow:Final test accuracy = 82.1% (N=28)
INFO:tensorflow:Froze 2 variables.
INFO:tensorflow:Converted 2 variables to const ops.
done !!
```

Figure 56: Screenshot for the Log Detail- End of the Training Session of the Model

Appendix G : Images Used for the Mesh Model

Images that was used to generate the 3d mesh model.

 *Figure 57: Image Used for Generating 3d Mesh Model 1*	 *Figure 58Image Used for Generating 3d Mesh Model 2*
 *Figure 59Image Used for Generating 3d Mesh Model 3*	 *Figure 60Image Used for Generating 3d Mesh Model 4*
 *Figure 61Image Used for Generating 3d Mesh Model 5*	 *Figure 62Image Used for Generating 3d Mesh Model 6*