SETHU INSTITUTE OF TECHNOLOGY, KARIAPATTI
An Autonomous Institution

B.E. ELECTRONICS AND COMMUNICATION ENGINEERING

INTELLIGENT VEHICLE DAMAGE ASSESSMENT AND COST ESTIMATION FOR INSURANCE COMPANIES

INNOVATIVE PROJECT REPORT

Submitted by

P.VISHNU PRASATH	92172019104175
P.YOGESH	92172019104178
N.VIGNESH	92172019104172
B.ROSHAN	92172019104127

TEAM ID: PNT2022TMID17420

for the course of

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP
(Naalaiya Thiran Program)

INDEX

1. INTRODUCTION

1.1 Project Overview

Vehicles are significantly rising in today's globe. Because there are more cars on the road, accidents happen more frequently because individuals are driving them at high speeds. When an accident occurs, the people file a claim with their auto insurance for the necessary funds to repair the car, because to inaccurate claims, the corporation behaves improperly and doesn’t make payments now. This occurs as a result of claims leakage, which is the discrepancy between the sums secured by the firm and the sums that it should have secured in accordance with the claims. Even if the car's damage is easily seen, the claim procedure will take longer than usual in accordance with company policy. Despite the company's best efforts, there is a delay in the claims procedure. Differentiate the suggested approach to perhaps speed up the process of assessing automotive damage. Instead of taking hours to accomplish automotive damage detection if it were visually inspected, a system may perform it in a minute by just providing a picture of a damaged vehicle. The system can determine the analysis of the damage, the position of the damage, and the degree of the damage using machine learning and computer vision.

1.2 Purpose

In Advanced Artificial Intelligence (AI), machine learning and deep learning algorithms can help to solve these kinds of problems for insurance industries. we apply deep learning-based algorithms, VGG16 and VGG19, for car damage detection and assessment in real world datasets .The algorithms detect the damaged part of a car, assess its location and severity .After analyzing and implementing our models, we can find out that the results of using transfer learning and regularization can work better than those of fine-tuning. Then we apply transfer learning in pre-trained VGG models and use some techniques to improve the accuracy of our system

2. LITERATURE SURVEY

2.1 Existing problem

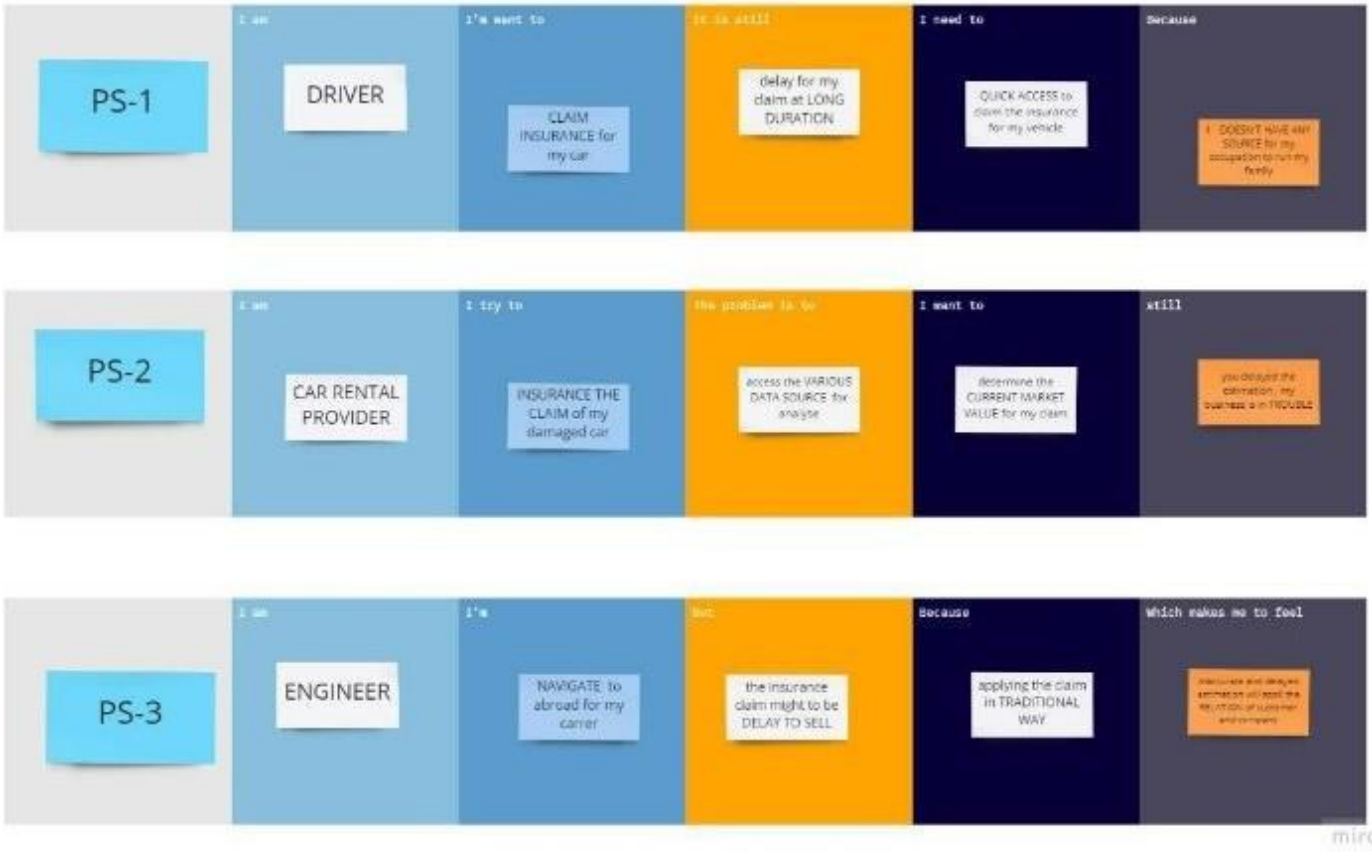
Today's world is seeing a substantial increase in automobiles. Because there are more automobiles on the road and more people are driving them at high speeds, accidents happen more frequently. When an accident happens, the parties involved submit a claim with their auto insurance to obtain the money needed to repair the vehicle since, according to false claims, the company acts inappropriately and withholds payments.

2.2 References

TITLE	METHODOLOGY	AUTHOR/ PUBLICATION YEAR	ADVANTAGE	DISADVANTAGE
Applying image analysis to auto insurance Triage	Image analysis and pattern recognition are applied to automatically identify and characterize automobile damage.	Li Ying & Dorai Chitra, 2012	Because of the advancement of image analysis and pattern recognition technologies, the auto insurance industry could significantly benefit.	The drawback is that the automobile damaged can be analyzed only having white background otherwise it will be not able to give thedesired results.
Image based automatic vehicle damage detection	This approach requires 3D computer aided design (CAD) modes of the considered vehicle to identify how it would look if it were undamaged.	Srimal Jayewardene' ,2013	Automatically detecting the damage of the vehicle using photographs clicked at the accident site is extremely functional as it can greatly decrease the rate of processing insurance claims, and it will also providegreater	Vehicles have very reflective metallic bodies the photographs taken insuch an uncontrolled environment can be expected to have a certain amount of inter object reflection. Application of standard computer vision.

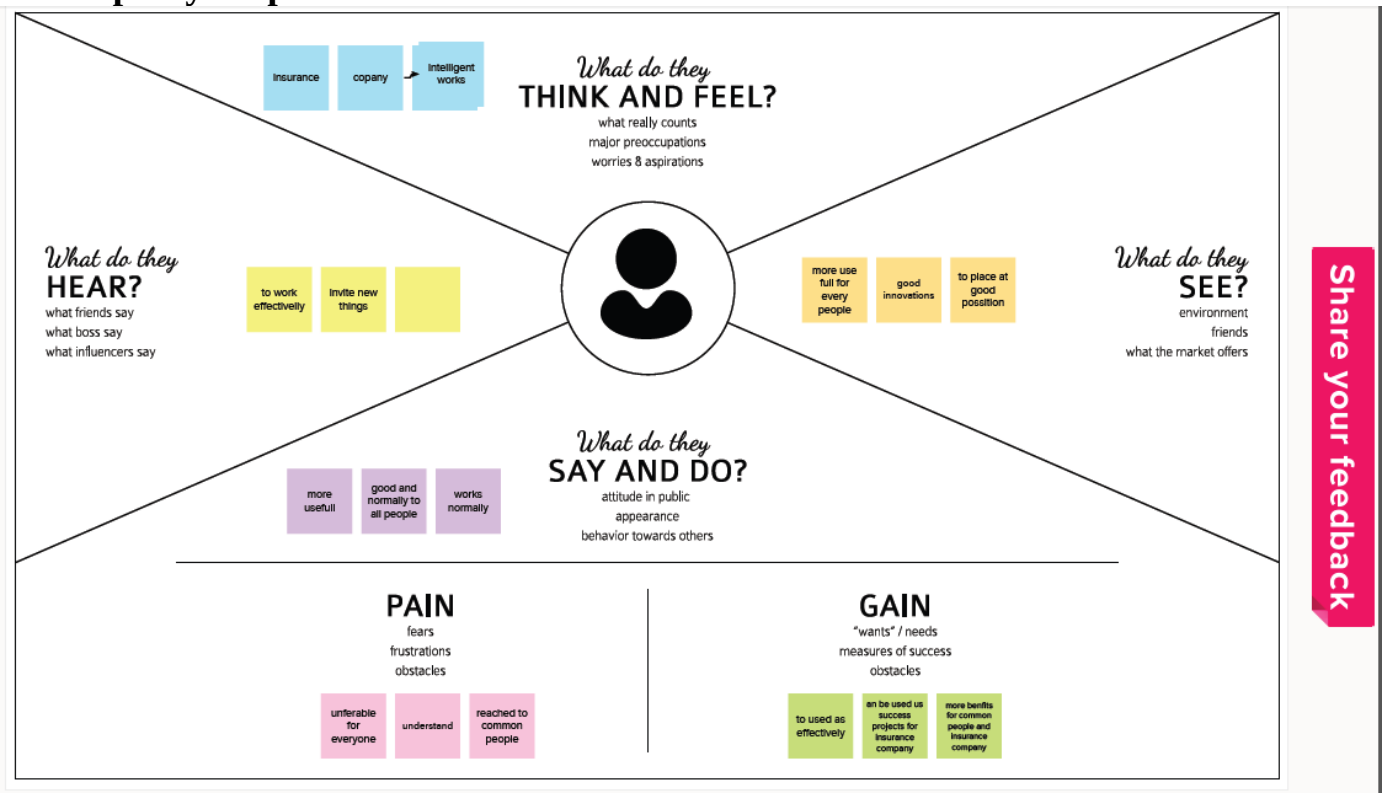
A Secure AI-driven Architecture for Automated Insurance Systems Fraud Detection and Risk Measuremen t	Block chain, data analysis, machine learning, AI for damageidentificati on.	M.Wassel,2019	Proposed classifiers ensure not only the best accuracy in detecting fraudulent claims but alsocan classify different types of fraud for insurance unlike the existing solutions	The major drawback of the proposed model isthat it only identifies the physical visible damage and not ofthe internal or the interior damage.
Car damage detection and classificati on	CNN model is trainedon Image Net dataset.After fine tuning the dataset, transfer learning with L2 regularization is Applied	Phyu Mar Kyu, Kuntpong Woraratpanya, 2020	Pre-trained VGG modelnot only detect damagedpart of a car but also assess its location and severity.	Transfer learningand regularization can work better than those of finetuning.

2.3 Problem Statement Definition



3. IDEATION & PROPOSED SOLUTION

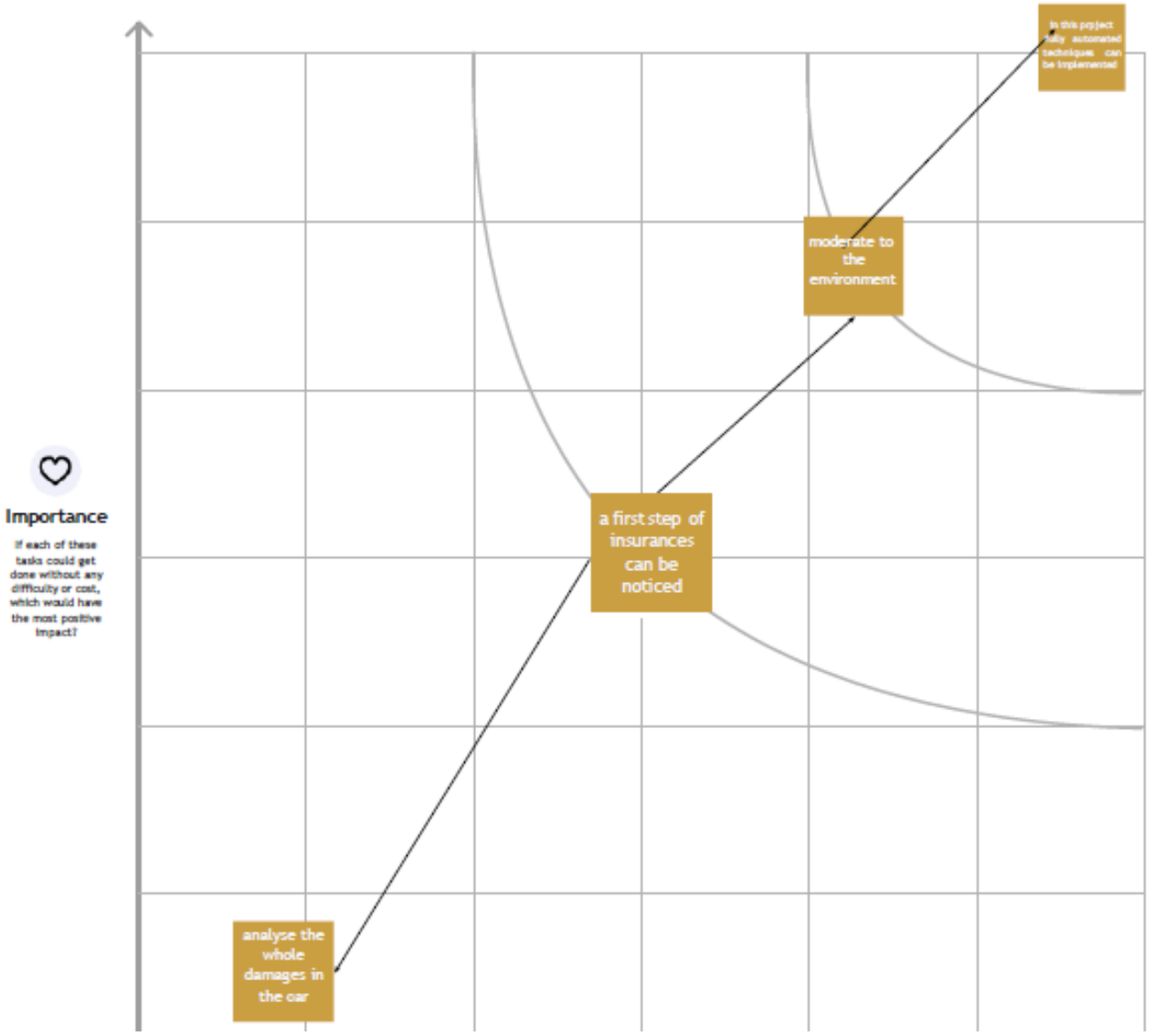
3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

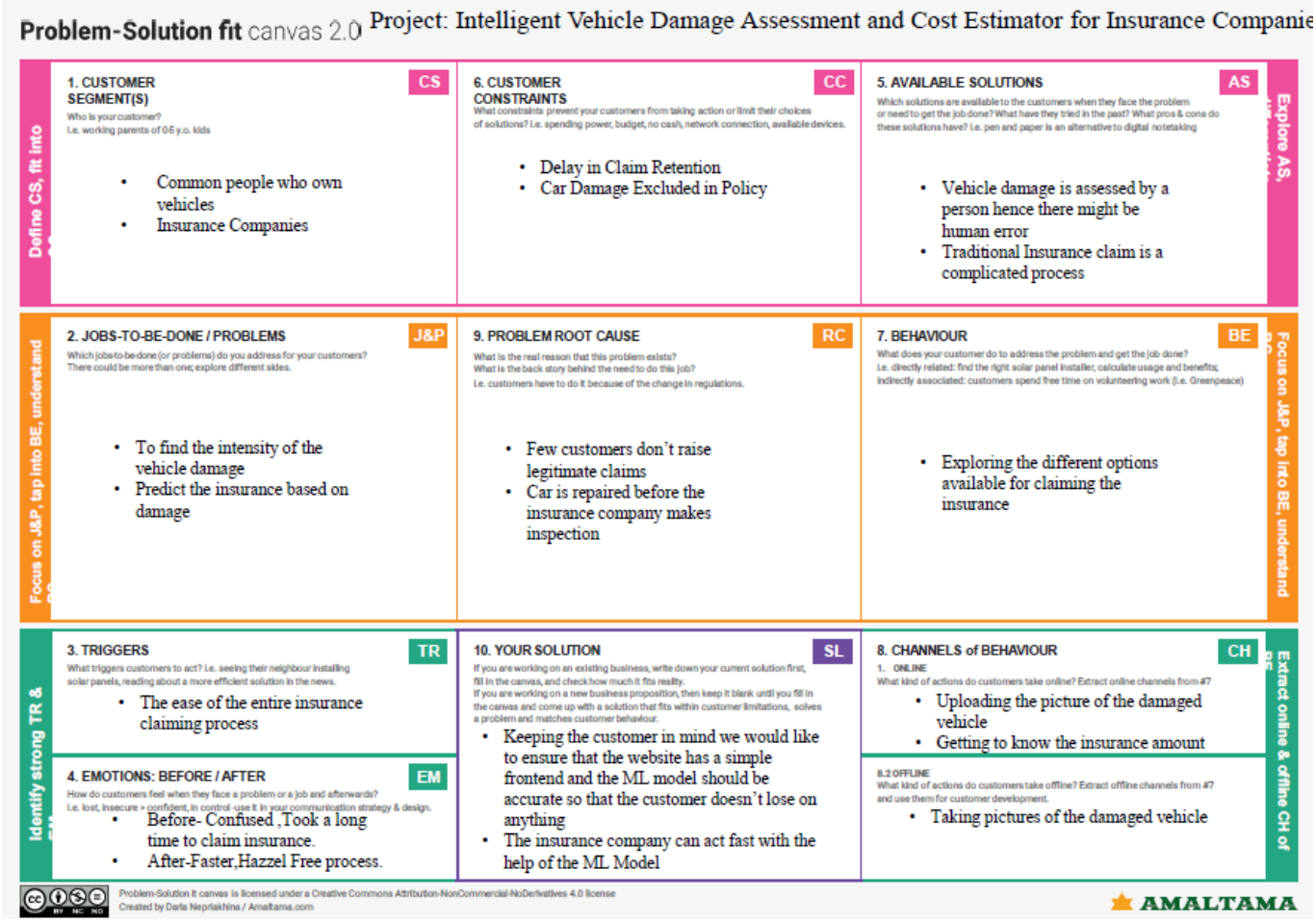
20 minutes



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop an Intelligent Vehicle Damage Assessment & Cost Estimator for Insurance Companies.
2.	Idea / Solution description	Estimate the cost of damage due to the accident. Easy way to claim the insurance. It contains several categories to detect the damage.
3.	Novelty / Uniqueness	Locating the damage occurred at a place with accordance to the specific cost of the damage. It find the exact damaged location to predict the cost. Its objective is to mechanically observe damages in vehicles, find them, classify their severity levels, and visualize them by contouring their precise locations.
4.	Social Impact / Customer Satisfaction	Easy to predict the accurate cost for the damage Everyone gets the exact details for their damage. AI has proved its efficiency in fraud detection for suspected collusion claims.
5.	Business Model (Revenue Model)	The Algorithms notice the broken area of an automobile and assess its location thus its severity.
6.	Scalability of the Solution	AI detects the accurate damaged area and predicts their cost to insure.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

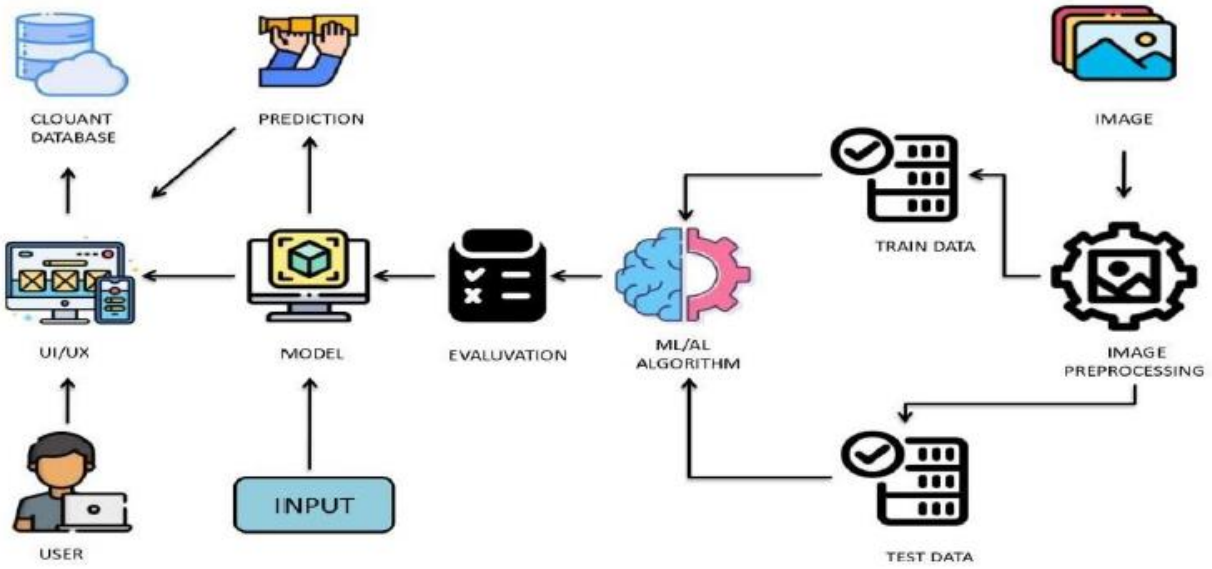
4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Interface	Login System,Dashboard,Uploading Image,Review and Analyze the results.
FR-4	Collection of datasets	Information about the user and their vehicle. Information about Insurance plans.
FR-5	Results	The model must be structured with high accuracy. The results obtained from the model will be displayed for the user to understand easily.

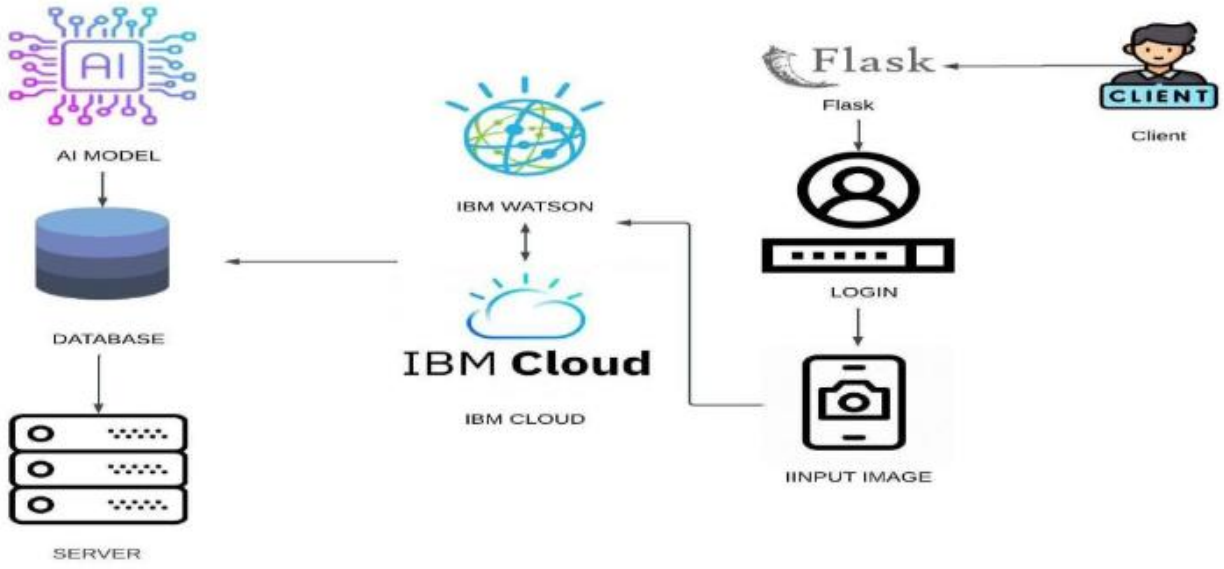
4.2 Non-Functional requirements

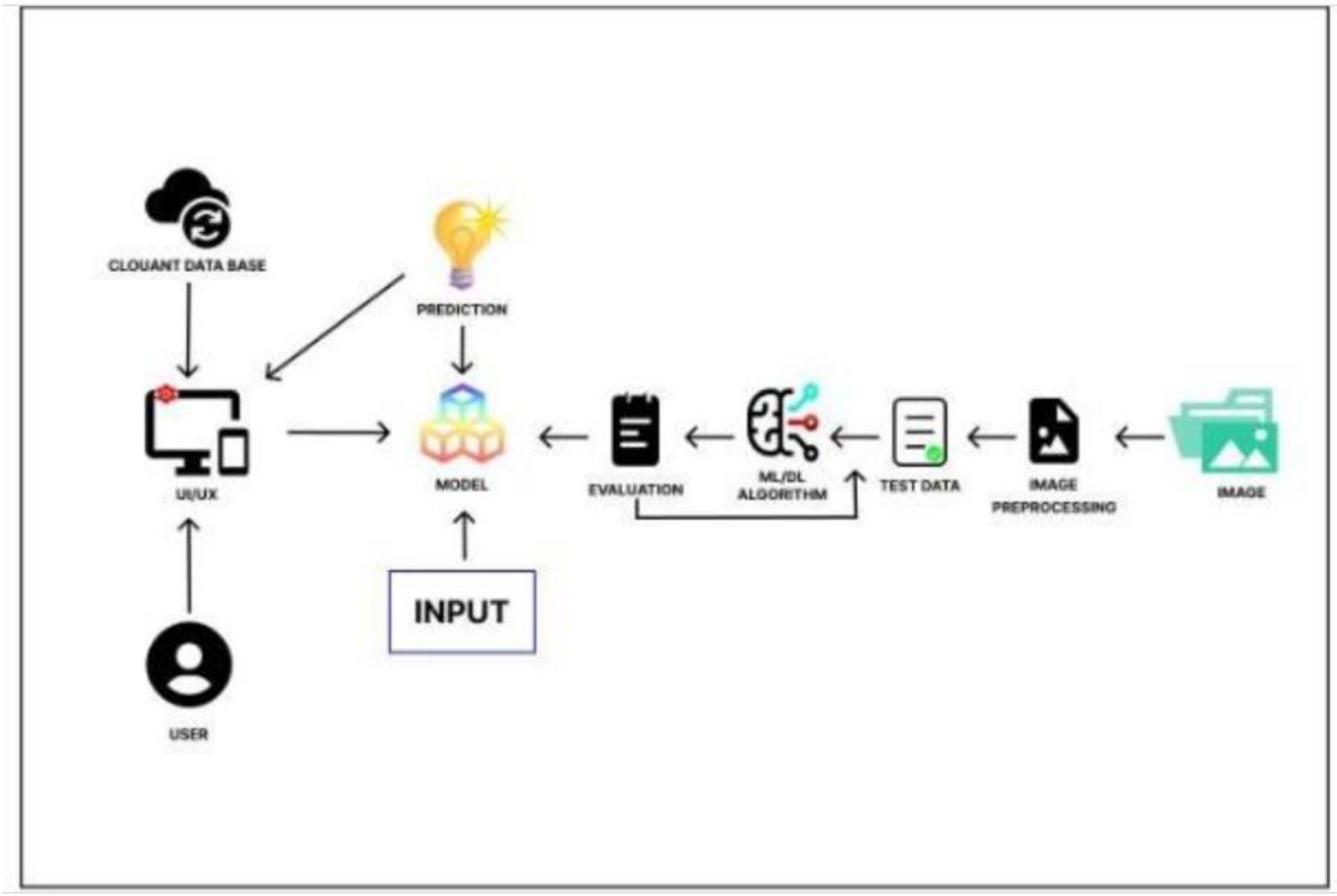
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Intelligent model for damage assessment in vehicle and cost estimate provided by insurance company.
NFR-2	**Security**	The authenticity of the user and the confidentiality of the user's details relating to his vehicle must be preserved.
NFR-3	**Reliability**	This project needs to achieve good accuracy in damage assessment as well as cost estimation so that users receive an accurate and unbiased amount of insurance.
NFR-4	**Performance**	Abide images should be captured and uploaded to a website where the proposed model will perform a damage assessment and quote the appropriate insurance costs.
NFR-5	**Availability**	The webpage must be compatible with web browsers on mobile phones and computers.
NFR-6	**Scalability**	The proposed solution will be scalable in the future due to more efficient and faster analysis and accurate cost forecasting.

5. PROJECT DESIGN
5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture




CLOUANT DATA BASE
PREDICTION
UI/UX
MODEL
EVALUATION
ML/DL ALGORITHM
TEST DATA
IMAGE PREPROCESSING
IMAGE
INPUT
USER

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard by entering valid credentails	High	Sprint-1
Customer Details	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer Uses	Dashboard	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-4
Customer Options	Details about insurance companies	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Facebook Gmail	Medium	Sprint-1
Customer usage	Login	USN-5	As a user, I can log into the application by entering email & password	I can log in and view my dashboard at my demand on any time	High	Sprint-1
Customer needs to do	Dashboard	USN-6	As a user I must capture images of my vehicle and upload it into the web portal	I can capture the entire vehicle and upload	High	Sprint-2
Customer (Web user)	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the cost estimated	I can get the estimated insurance cost	High	Sprint-3
Customer Care Executive	Details about Estimated cost Based on damage	USN-8	As a user, I need to get support from developers in case of queries and failure of service provided	I can have smooth user experiences and all the issues raised is sorted	Medium	Sprint-4
Administrator	Details about Estimated cost Based on damage	USN-9	We need to satisfy the customer needs in an efficient way and make sure any sort of errors are fixed	I can finish the work without any problems	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation ,Sprint Delivery Schedule

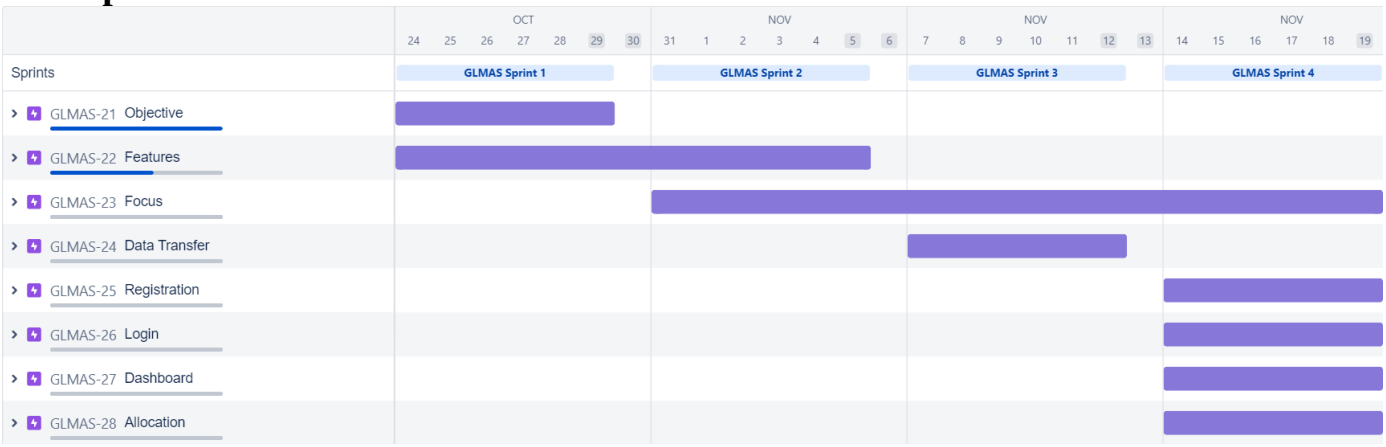
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN - 1	As a user, I can register for the application by entering my details of name, email, cars etc. verifying my Gmail account and creating new account with password	7	HIGH	Vishnu
Sprint-1	Login	USN -2	As a user,entering my email, and password, and confirming my password,I can login to my account.	7	HIGH	vignesh
Sprint-1	Dashboard	USN-3	As a user, I can clearly see data, point, graphs, charts and trends of my previous activity and global activity related to my views	2	LOW	Yogesh
Sprint-2	Details about insurance company	USN-4	As a user, I can register for theApplication through Gmail and account id.	8	MEDIUM	`vignesh
Sprint-1	repeated logins and logout	USN-5	As a user,I can log in and view my dashboard at my demand on any time	4	HIGH	roshan
Sprint-2	Webpage	USN-6	As a user, I must enter all details of car, accident, capture images of my vehicle and upload it into the web portal.	12	HIGH	vishnu
Sprint 3	Details about estimated cost based on damage	USN-7	As a user I must receive a detailed report of the damages present in the vehicle and the Cost estimated.	20	HIGH	Roshan Vishnu Vignesh yogesh
Sprint 4	Provide friendly and efficient	USN-8	As a user, I need to get support from developers in case of	10	MEDIUM	Vishnu vignesh

	customer support and sort out the queries.		queries and failure of service Provided by chat-box,mail or call.			
Sprint 4	overview the entire process and act as a bridge between user and developer	USN-9	As a Team member, We need to satisfy thecustomer needs in an efficient way and make sure any sort of errors are fixed	10	HIGH	Yogesh roshan

Project Tracker, Velocity & Burn-down Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 nov 2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING

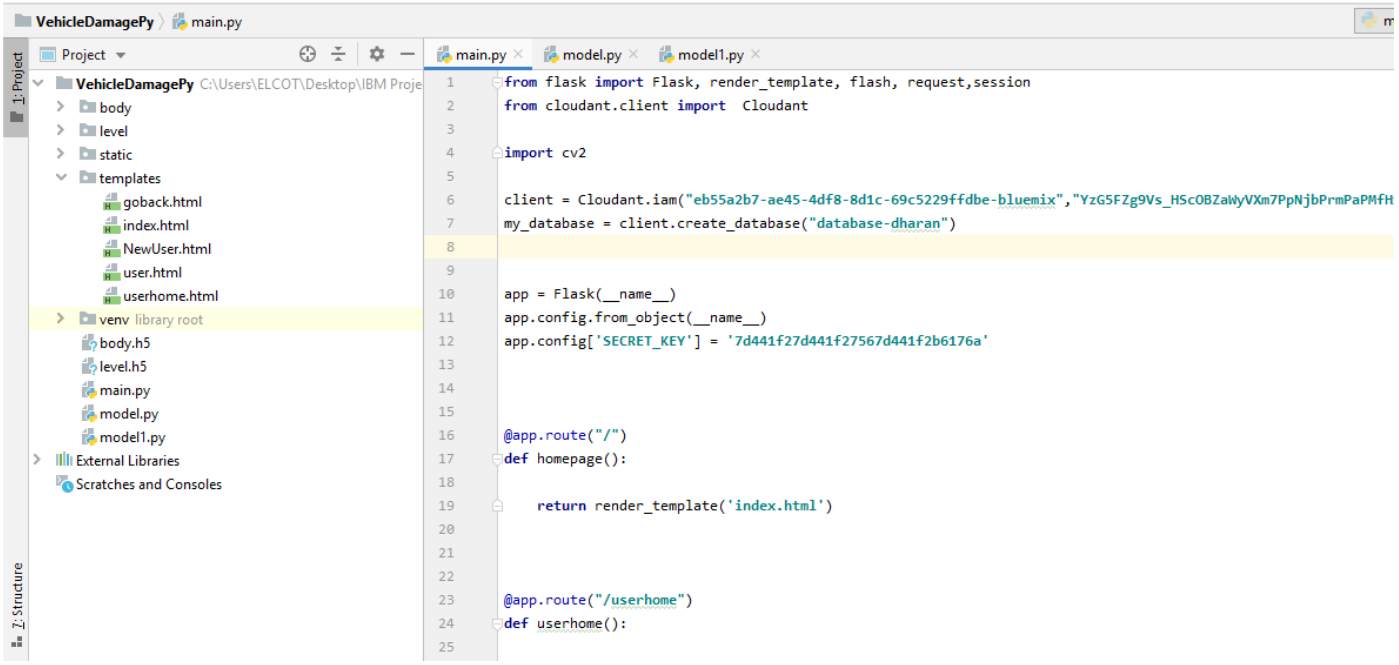
IMAGE PROCESSING CODE

```
{
         "cells": [
          {
           "cell_type": "markdown",
           "metadata": {},
           "source": [
            "Importing the Image Data Generator Libraries "
           ]
          },
          {
           "cell_type": "code",
           "execution_count": 2,
           "metadata": {},
           "outputs": [],
           "source": [
            "import tensorflow as tf \n",
            "import keras \n",
            "from keras.preprocessing.image import ImageDataGenerator"
           ]
          },
          {
           "cell_type": "markdown",
           "metadata": {},
           "source": [
            "Canfiguring The Image Data Generator "
           ]
          },
          {
           "cell_type": "code",
           "execution_count": 3,
           "metadata": {},
           "outputs": [],
           "source": [
            "#training images \n",
            "train_datagen = ImageDataGenerator(\n",
            "        rescale=1./255,\n",
            "        shear_range=0.2,\n",
            "        zoom_range=0.2,\n",
            "        horizontal_flip=True)\n",
            "\n",
            "#val images \n",
            "val_datagen = ImageDataGenerator(rescale=1./255)"
           ]
          },
          {
           "cell_type": "markdown",
           "metadata": {},
```

```
   "source": [
    "Applying the image generator to the body "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 979 images belonging to 3 classes.\n"
     ]
    }
   ],
   "source": [
    "body_train_generator = train_datagen.flow_from_directory(\n",
    "        'V:\\\\WorkSpace\\\\IBM-Project-23426-1659882722\\\\Dataset\\\\Car
damage\\\\body\\\\training',\n",
    "        target_size=(150, 150),\n",
    "        batch_size=32,\n",
    "        class_mode='categorical')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 171 images belonging to 3 classes.\n"
     ]
    }
   ],
   "source": [
    "body_val_generator = val_datagen.flow_from_directory(\n",
    "        'V:\\\\WorkSpace\\\\IBM-Project-23426-1659882722\\\\Dataset\\\\Car
damage\\\\body\\\\validation',\n",
    "        target_size=(150, 150),\n",
    "        batch_size=32,\n",
    "        class_mode='categorical')"
   ]
  },
  {
   "cell_type": "markdown",
```

```
   "metadata": {},
   "source": [
    "Applying the image generator to the level "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 979 images belonging to 3 classes.\n"
     ]
    }
   ],
   "source": [
    "level_train_generator = train_datagen.flow_from_directory(\n",
    "        'V:\\\\WorkSpace\\\\IBM-Project-23426-1659882722\\\\Dataset\\\\Car
damage\\\\level\\\\training',\n",
    "        target_size=(150, 150),\n",
    "        batch_size=32,\n",
    "        class_mode='categorical')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 171 images belonging to 3 classes.\n"
     ]
    }
   ],
   "source": [
    "level_val_generator = val_datagen.flow_from_directory(\n",
    "        'V:\\\\WorkSpace\\\\IBM-Project-23426-1659882722\\\\Dataset\\\\Car
damage\\\\level\\\\validation',\n",
    "        target_size=(150, 150),\n",
    "        batch_size=32,\n",
    "        class_mode='categorical')"
   ]
  },
  {
```

```
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3.7.9 64-bit (microsoft store)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.7.9"
  },
  "orig_nbformat": 4,
  "vscode": {
   "interpreter": {
    "hash": "b81d285a39ce6bce5fc3d0e228a6124a883049b7dbc9a3a4527a4b38646ff66a"
   }
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```


VehicleDamagePy
main.py
Project
VehicleDamagePy C:\Users\ELCOT\Desktop\IBM Proje
body
level
static
templates
goback.html
index.html
NewUser.html
user.html
userhome.html
venv library root
body.h5
level.h5
main.py
model.py
model1.py
External Libraries
Scratches and Consoles
main.py
model.py
model1.py
from flask import Flask, render_template, flash, request,session
from cloudant.client import Cloudant
import cv2
client = Cloudant.iam("eb55a2b7-ae45-4df8-8d1c-69c5229ffdbe-bluemix","YzG5FZg9Vs_HScOBZaWyVXm7PpNjbPrmPaPMfH
my_database = client.create_database("database-dharan")
app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
return render_template('index.html')
@app.route("/userhome")
def userhome():

8.TESTING

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a setof instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

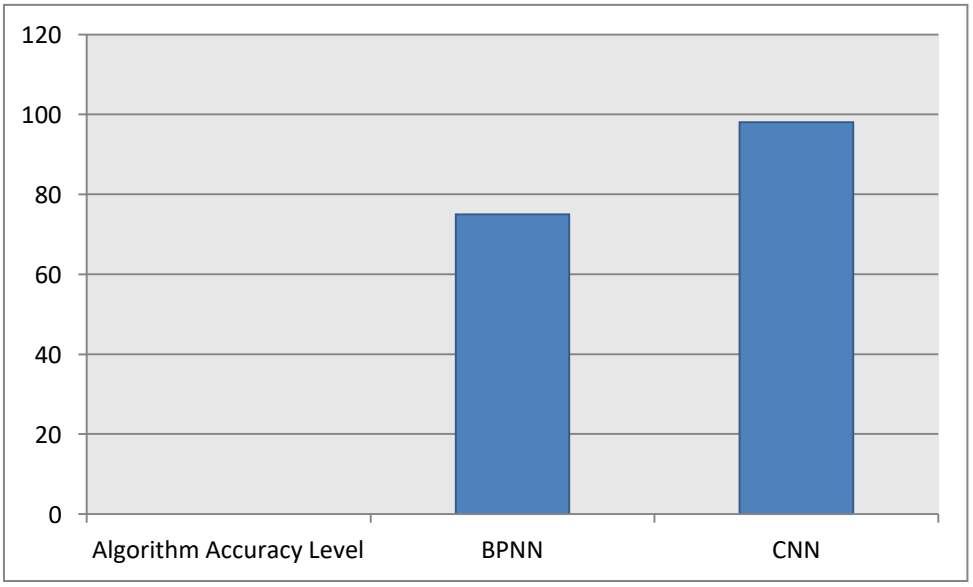
S.NO	Scenario	Input	Excepted output	Actual output
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload damaged vehicle image asa input	Detecting object and analyze for claim insurance	Details are stored in a database.

8.2 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

1. RESULTS

9.1 PERFORMANCE METRICS



2. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Digitalized claim process makes easy to use
- Give the accurate result of the damaged vehicle
- Helps the insurance company to analyze the damaged vehicle and also payment process.

DISADVANTAGE

- It will take more time to claim the insurance in manual process
- Because of incorrect claims, the company behaves badly and doesn’t make paymentscurrently.
- Poor customer support

3. CONCLUSION

In this research proposal, a neural network-based solution for automobile detection will be used to address the issues of automotive damage analysis and position and severity prediction. This project does several tasks in one bundle. The method will unquestionably assist the insurance firms in conducting far more thorough and systematic analyses of the vehicle damage. Simply sending the system a photograph of the vehicle, it will evaluate itand determine whether there is damage of any type, where it is located, and how severe it is.

4. FUTURE SCOPE

In future work, need to use several regularisation methods with a big dataset in our next work. Anticipate the cost of a car damaged component more accurately and reliably if we have higher quality datasets that include the attributes of a car (make, model, and year of production), location data, kind of damaged part, and repair cost. This study makes it possibleto work together on picture recognition projects in the future, with a focus on the auto insurance industry. The study was able to accurately validate the presence of damage, its location, and its degree while eliminating human bias. These can be further enhanced by adding the on the fly data augmentation approaches.

5. APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request,session
from cloudant.client import  Cloudant
import cv2
client = Cloudant.iam("eb55a2b7-ae45-4df8-8d1c-69c5229ffdbe-
bluemix","YzG5FZg9Vs_HScOBZaWyVXm7PpNjbPrmPaPMfHx7w3X9",connect= True)
my_database = client.create_database("database-dharan")
app = Flask(_name_)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/userhome")
def userhome():
    return render_template('userhome.html')
@app.route("/addamount")
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')


@app.route("/user")
def user():


    return render_template('user.html')
@app.route("/newuse",methods=['GET','POST'])def
newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {
```

```
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
            url = my_database.create_document(data)
            return render_template('goback.html', data="Register, please login using your
details")
        else:
            return render_template('goback.html', data="You are already a member, please
login using your details")
@app.route("/userlog", methods=['GET', 'POST'])def
userlog():
        if request.method == 'POST':
            user = request.form['_id']
            passw = request.form['psw']
            print(user, passw)
            query = {'_id': {'$eq': user}}
            docs = my_database.get_query_result(query)
            print(docs)
            print(len(docs.all()))
            if (len(docs.all()) == 0):
                return render_template('goback.html', pred="The username is not found.")
            else:
                if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):return
                    render_template("userhome.html")
                else:
                    return render_template('goback.html',data="user name and password
incorrect")
```

```
@app.route("/predict", methods=['GET', 'POST'])def
predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        file.save('static/Out/Test.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))img1
        = cv2.imread('static/Out/Test.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        result = classifierLoad.predict(test_image)
        result1 = ''
        if result[0][0] == 1:
            result1 = "front"
        elif result[0][1] == 1:
            result1 = "rear"
        elif result[0][2] == 1:
            result1 = "side"
        file = request.files['fileupload1']
        file.save('static/Out/Test1.jpg')
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('level.h5')
        import numpy as np
        from keras.preprocessing import image
        test_image = image.load_img('static/Out/Test1.jpg', target_size=(200, 200))img1
        = cv2.imread('static/Out/Test1.jpg')
        # test_image = image.img_to_array(test_image)
```

```
            test_image = np.expand_dims(test_image, axis=0)result
            = classifierLoad.predict(test_image)
            result2 = ''
            if result[0][0] == 1:result2
               = "minor"
            elif result[0][1] == 1: result2 =
               "moderate"
            elif result[0][2] == 1:
               result2 = "severe"
            if (result1 == "front" and result2 == "minor"):value =
               "3000 - 5000 INR"
            elif (result1 == "front" and result2 == "moderate"):value
               = "6000 8000 INR"
            elif (result1 == "front" and result2 == "severe"):value
               = "9000 11000 INR"
            elif (result1 == "rear" and result2 == "minor"):value =
               "4000 - 6000 INR"
            elif (result1 == "rear" and result2 == "moderate"):value
               = "7000 9000 INR"
            elif (result1 == "rear" and result2 == "severe"):value
               = "11000 - 13000 INR"
            elif (result1 == "side" and result2 == "minor"):value =
               "6000 - 8000 INR"
            elif (result1 == "side" and result2 == "moderate"):value
               = "9000 - 11000 INR"
            elif (result1 == "side" and result2 == "severe"):value =
               "12000 - 15000 INR"
            else:
               value = "16000 - 50000 INR"
            return render_template('userhome.html', prediction=value)if
      name_______== '_main_':
app.run(debug=True, use_reloader=True)
```