INTELLIGENT VEHICLE DAMAGE ASSESMENT AND COST ESTIMATOR FOR INSURANCE COMPANIES

PNT2022TMID50649

INDEX

1. INTRODUCTION

1.1 Project Overview

The project “Intelligent Vechile Damage Assessment and Cost Estimator for Insurance Companies” is a responsive web application powered by aritifical Intelligence and IBM Watson Cloud. Deep Learning model is trained with the various damaged car images in various views and the VGG16 from the TensorFlow library is used for the better Deep Learning model architecture. An attractive front end can be developed using HTML and CSS. The pages such as Index.html , login.html, logout.html, register.html and prediction.html are created and embedded with the IBM cloud databse using python framework called flask. The web application takes the image input and estimate the cost for the insurance companies based on the damages in the car.

1.2 Purpose

The project is based on the domain of Artificial Intelligence and powered by the IBM watson cloud. A responsive web application can be developed using the HTML and CSS which is connected to waston cloud. In the cloud, a database service by availing the service Instance of the IBM cloud and the database API key is collected and connected with the front-end using flash which is an python framework for desiging the backend. Pages such as index.html, login.html, logout.html and prediction.html are used to interact with the web application. The user can register and the data of the user is saved in the databse of the IBM cloud, during the time of login, the login ID is compared with the ID in the databse and allow the user to the next page. The Deep Learning model is build using the VGG16 which is present in the keras library and the model is trained with the images of mulitple car with various level cum types of damages. The model is deployed in the back-end using the flask and the prediction.html page is setted to collect the image from the user. The prediction algorithm is used treat the image and estimated the cost for the user. The project is based on the various components which helps to handle the back – end and Front – end. Then front – end is build using html and css which is connectedback – end which is build using the python and IBM cloud. The project is powered by the IBM Watson cloud and is based in the artificial intelligence field. With the use of HTML and CSS and the Waston Cloud, a responsive web application may be created. The database API key is gathered and connected with the front-end using flash, which is a python framework for designing the backend.

2. LITERATURE SURVEY

2.1 Existing Problem

- Damage Assessment of a vehicle and Insurance Reclaim: This paper presents a system using CNN and image classification to assess the severity of damage to an automobile, which takes a user's input as an image to test the severity of the damage, which happens in two steps. The first step is image classification, where the user's input is used by the neural network to determine whether or not an automobile is damaged. The region and severity of the damage are determined in the second step using object detection on the flattened input that was received as the output in step one. The area may be the back, the front, or the side, and the severity may be classified as minor, moderate, or major. A report is filed and delivered to the user and the insurance company when the R-CNN network determines the extent of the damage. With little human contact, the user will be able to receive payment based on the results of the models.

- Convolutional Neural Networks for vehicle damage detection: In this paper, a model for detecting vehicle damage is created, and it is divided into twelve categories. A deep learning model that can accurately detect and classify vehicle damages is created and evaluated in a specially designed light street, indicating that strong reflections complicate the detection performance. The proposed model outperforms other existing models in the classes Bend and Cover Damage. FSSD with Darknet-53 and YOLO v3 with Darknet-53 yield the best results. The drawback of the proposed approach is the robustness against different light conditions

- Car Damage Assessment for Insurance Companies: In this paper a neural network-based solution for car detection, managing the problem of car damage analysis, prediction of car damage location and severity of the damage is proposed. The proposed system is intended to help insurance companies to analyze car damage a lot more successfully and well organized, and it quickly performs car damage detection by sending the image containing a damaged car for visual inspection. This system utilizes a machine learning approach along with computer vision to decide the damage analysis, the location of the damage as well as the severity of the damage

2.2 Problem Statement Definition

- Assessing Car Damage with Convolutional Neural Networks: This study focuses on automotive damage estimation, with auto insurers as their main potential clients. Three different Transfer Learning techniques are employed to do this, each of which identifies the existence, location, and degree of damage. Convolutional Neural Networks, which are adapted to maximize accuracy, serve as the foundation for the algorithms used. Each approach is analyzed and varying degrees of accuracy were achieved across different models deployed ranging from 68% to 87%. In this work, accuracy as high as 87.9% was attained. This study improves a number of existing methods and creates opportunities for collaboration in image recognition, notably in the field of auto insurance.

- Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning: This paper proposes deep learning-based methods for the classification of car damage types - MobileNet to classify vehicle damage into three groups: medium damage, enormous damage, and no damage. The extent of the damage to the vehicle determines its severity, ranging from medium to huge. The damage categories are based on typical damage kinds including shattered glass, dents on the front or back, damaged lamps or bumpers, etc. Automation in real-time applications, however, faces several challenges. Instead of capturing a picture of a car in real time, users can upload fake pictures. Making fake photos can involve using image-editing software to cover up flaws, getting images from the internet, or even taking screenshots of other devices' screens. To deal with these kinds of fake photographs, a hybrid strategy is also suggested in this research. To determine whether an image has been altered or is a screenshot, metadata analysis, and image editing software signature detection are used. It is suggested that moiré effect detection be used to determine whether an image was captured from the screen of another device, such as a computer screen when a mobile phone was used to snap a photo of an automobile.

- Car damage detection and classification: In this paper, a CNN model is developed and trained on the ImageNet dataset. After fine-tuning the dataset, transfer learning with L2 regularization is applied. In the proposed system, a Pre-trained VGG model not only detects the damaged part of a car but also assesses its location and severity. With the use of transfer learning and L2 regularisation, the proposed system achieves an accuracy of 95.22% of VGG19 and 94.56% of VGG16 in damaged detection, 76.48% of VGG19 and 74.39% of VGG16 in damage localization, and 58.48% of VGG19 and 54.8% of VGG16 in damage severity.

2.3 References

- Vaibhav Agarwal, Utsav Khandelwal, Shivam Kumar, Raja Kumar, Shilpa M, "Damage Assessment Of A Vehicle And Insurance Reclaim", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 4, pp.e197-e201, April 2022, Available at :http://www.ijcrt.org/papers/IJCRT2204483.pdf

- R. E. van Ruitenbeek and S. Bhulai, “Convolutional Neural Networks for vehicle damage detection,” Machine Learning with Applications, vol. 9. Elsevier BV, p. 100332, Sep. 2022. doi: 10.1016/j.mlwa.2022.100332.

- Mandara G S and Prashant Ankalkoti, “Car Damage Assessment for Insurance Companies,” International Journal of Advanced Research in Science, Communication and Technology. Naksh Solutions, pp. 431–436, Jun. 23, 2022. doi: 10.48175/ijarsct-5048.

- H. Bandi, S. Joshi, S. Bhagat, and A. Deshpande, “Assessing Car Damage with Convolutional Neural Networks,” 2021 International Conference on Communication information and Computing Technology (ICCICT). IEEE, Jun. 25, 2021. doi: 10.1109/iccict50803.2021.9510069.

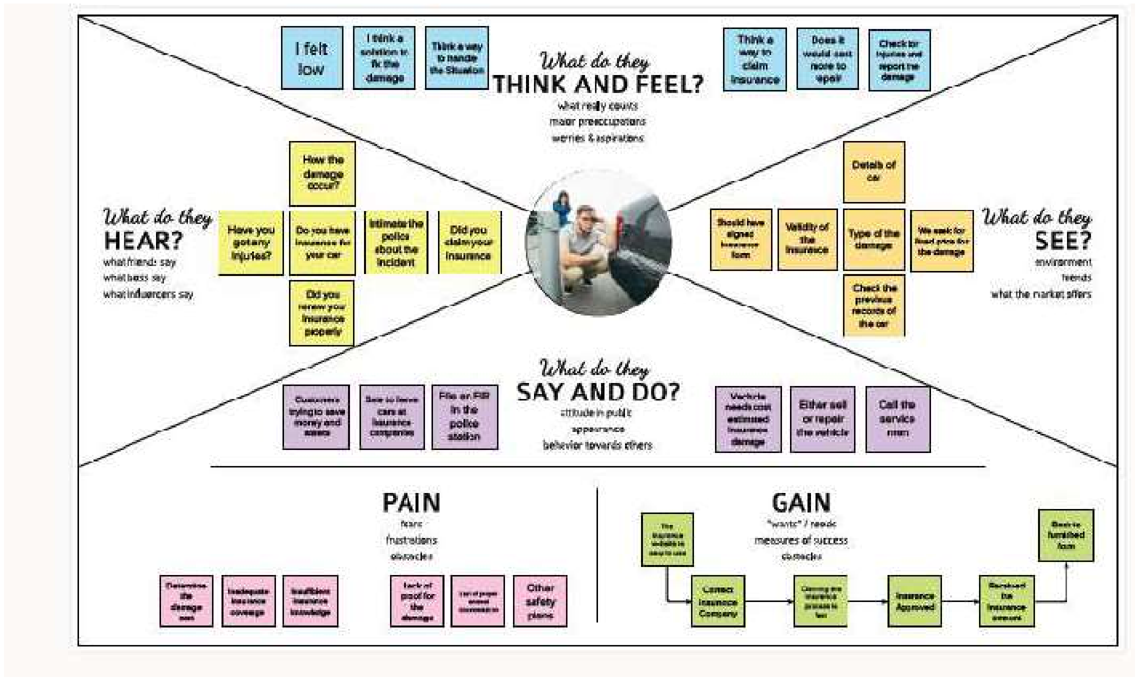
- U. Waqas, N. Akram, S. Kim, D. Lee and J. Jeon, "Vehicle Damage Classification and Fraudulent Image Detection Including Moiré Effect Using Deep Learning," 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2020, pp. 1-5, doi: 10.1109/CCECE47787.2020.9255806.

- M. Dwivedi et al., “Deep Learning-Based Car Damage Classification and Detection,” Advances in Intelligent Systems and Computing. Springer Singapore, pp. 207–221, Aug. 14, 2020. doi: 10.1007/978-981-15-3514-7_18.

- P. M. Kyu and K. Woraratpanya, “Car Damage Detection and Classification,” Proceedings of the 11th International Conference on Advances in Information Technology. ACM, Jul. 2020. doi: 10.1145/3406601.3406651.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map


I felt low
I think a solution to fix the damage
Think a way to handle the Situation
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Think a way to claim Insurance
Does it would cost more to repair
Check for injuries and report the damage
What do they HEAR?
what friends say
what boss say
what influencers say
How the damage occur?
Have you got any injuries?
Do you have insurance for your car
Intimate the police about the incident
Did you claim your insurance
Did you renew your insurance properly
What do they SEE?
environment
friends
what the market offers
Details of car
Should have signed insurance form
Validity of the insurance
Type of the damage
We seek for fixed price for the damage
Check the previous records of the car
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Customers trying to save money and assets
File an FIR in the police station
Vehicle needs cost estimated insurance damage
Either sell or repair the vehicle
Call the service man
PAIN
fears
frustrations
obstacles
Determine the damage cost
Inadequate insurance coverage
Insufficient insurance knowledge
Lack of proof for the damage
Other safety plans
GAIN
"wants" / needs
measures of success
obstacles
Contact Insurance Company
Insurance Approved
Received the insurance amount

3.2 Ideation and Brainstorming

Sabin Rakesh

Computer vision based estimation for car

With image processing the user gets notified severity of the damage

Neural network extracting Image feature

Machine Learning allows to predict accurate cost for thr damage

VGG model achieves almost 92% accuracy in image Net

Jeyanth Kumar

We can estimate the minor,major,and severe damage of the car

With the help of the image we can detect the dent

We can eliminate the labour cost

Fake image Detection

VGG model achieves almost 92% accuracy in image Net

Karthik

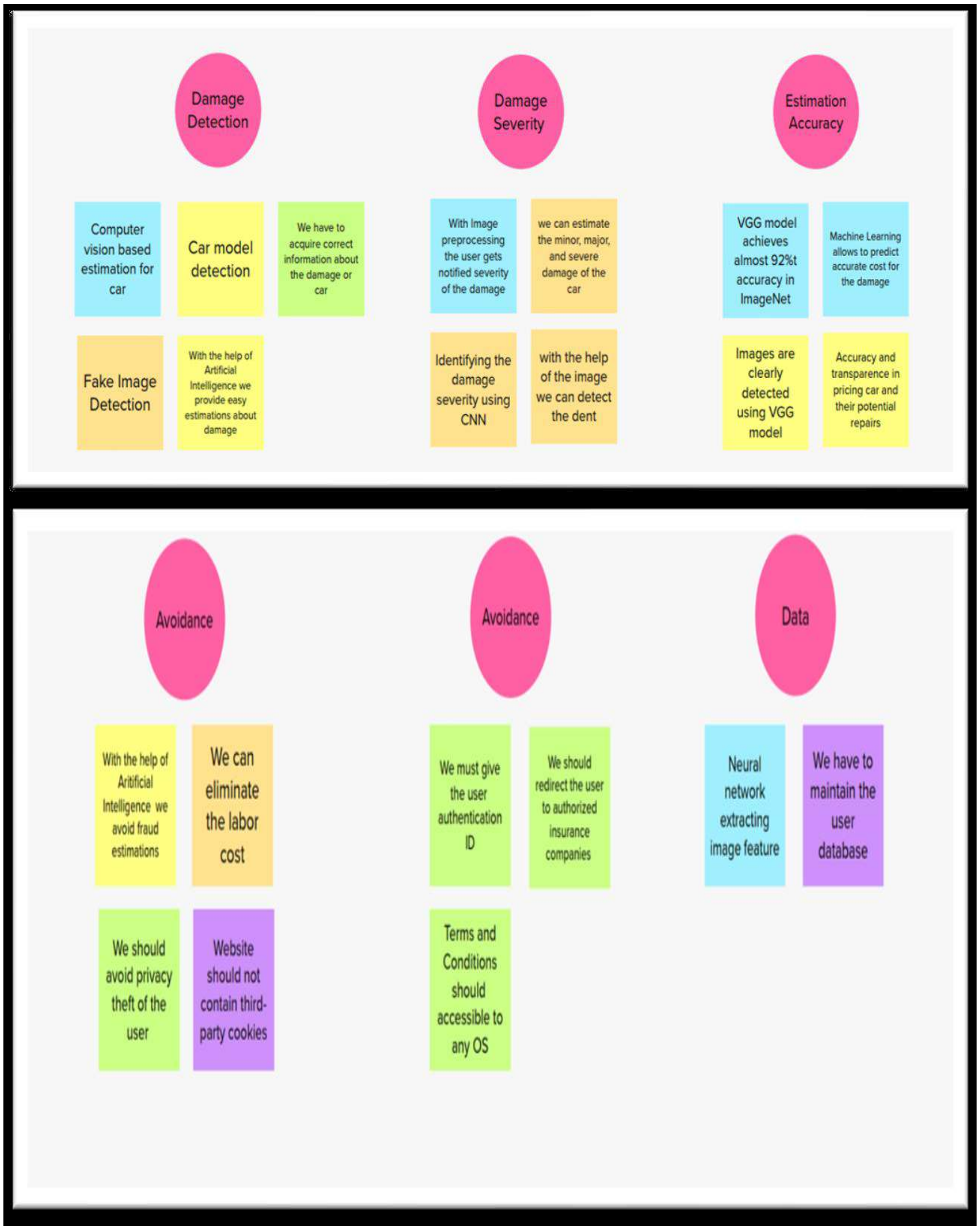
- With the help of AI we provide easy estimation about damage
- Images are clearly detected using VGG model
- With the help of AI we avoid fraud estimations
- Car Model Detection
- Accuracy and transparence in pricing car and their potential repairs

Viswa

- We have to acquire the correct information about the damage of car
- We must give the user authentication ID
- We should avoid privacy theft of the user
- We should redirect the user to authorized insurance companies
- Terms and Conditions should Accessible to any OS

Arun Kumar

- Website should not contain third-party cookies
- We have maintain the User Database
- Response time should be quick
- The website should be attractive to the user
- Website consists a feedback note



Damage Detection

Computer vision based estimation for car

Car model detection

We have to acquire correct information about the damage or car

Fake Image Detection

With the help of Artificial Intelligence we provide easy estimations about damage

Damage Severity

With Image preprocessing the user gets notified severity of the damage

we can estimate the minor, major, and severe damage of the car

Identifying the damage severity using CNN

with the help of the image we can detect the dent

Estimation Accuracy

VGG model achieves almost 92%t accuracy in ImageNet

Machine Learning allows to predict accurate cost for the damage

Images are clearly detected using VGG model

Accuracy and transparence in pricing car and their potential repairs

Avoidance

With the help of Artificial Intelligence we avoid fraud estimations

We can eliminate the labor cost

We should avoid privacy theft of the user

Website should not contain third-party cookies

Avoidance

We must give the user authentication ID

We should redirect the user to authorized insurance companies

Terms and Conditions should accessible to any OS

Data

Neural network extracting image feature

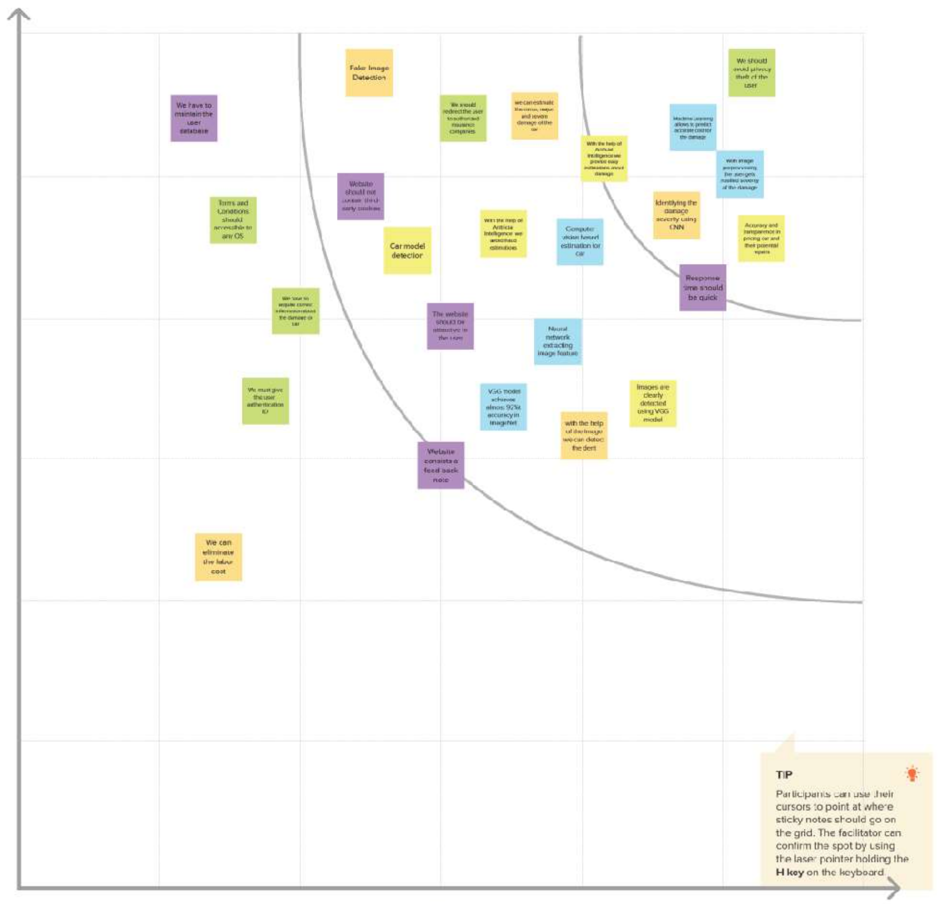
We have to maintain the user database

4

Prioritize

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



TIP

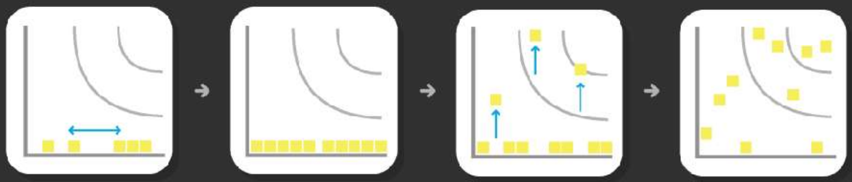
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

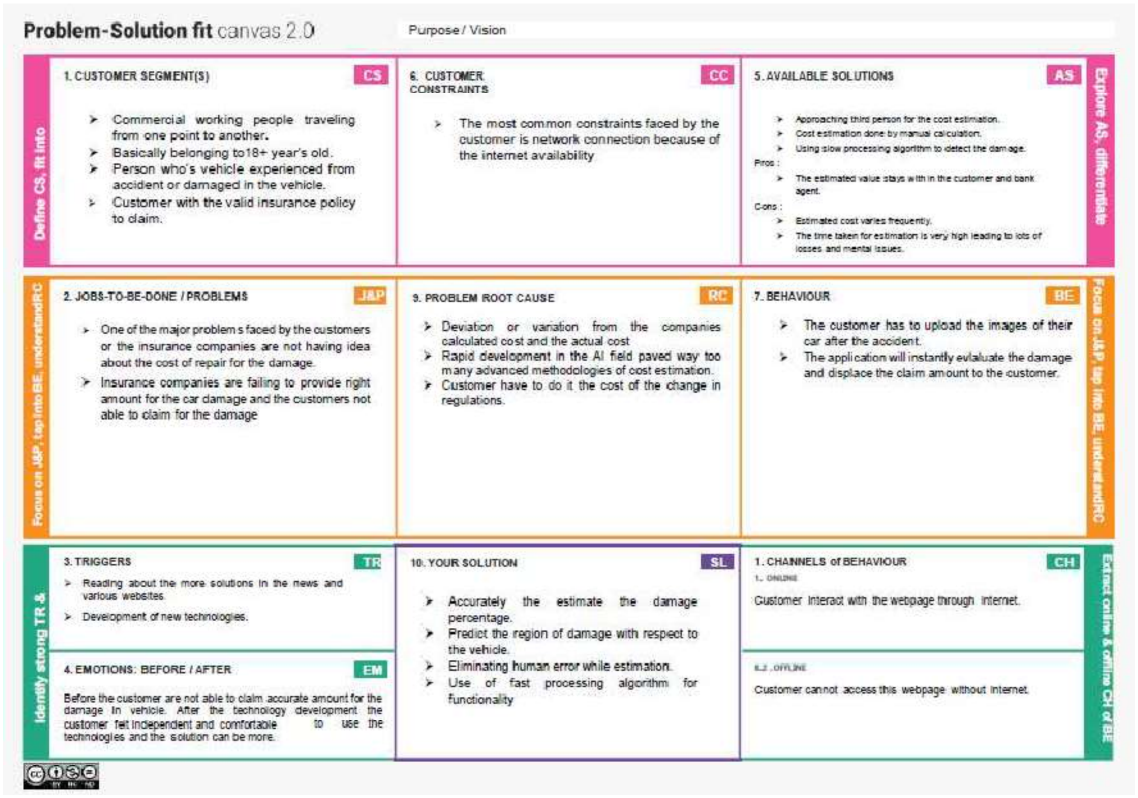
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)



3.3 Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Insurance company frequently suffer loses. Because they did not provide a proper explanation regards the estimation of the damage to the customer.
2	Idea / Solution Description	1. We create an AI Model to sense and detect the precise amount damage that occurred in the vehicle. 2. Then we create a user accessible portal and securely store the data provided by the user. 3. Finally compare the gathered damage percentage with the statistical cost estimation value to predict the cost.
3	Novelty / Uniqueness	1. The AI Model automatically calculates the damaged vehicle cost. 2. The deep learning algorithm provides progressively higher level features.
4	Social Impact / Customer Satisfaction	1. It's the user friendly website. 2. All the images and personal data will be secured in the cloud data security.
5	Business Model (Revenue Model)	Insurance companies have two primary sources of income Underwriting & Investment income. Financial investments including Listed shares, Government bonds, and Corporate bonds, make up the majority of insurance firms' assets. By estimating the level of car damage using our AI model and providing insurance accordingly, they can save more money and invest it in their businesses.
6	Scalability of the Solution	1. With the use of advanced machine learning techniques analyze damaged vehicles with high accuracy levels and keep on improving the learning ability of the model. 2. Our AI model can operate at the scale, speed, and complexity required for the aim.

3.4 Proposed Solution Fit



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into

1. CUSTOMER SEGMENT(S) CS

- Commercial working people traveling from one point to another.
- Basically belonging to18+ year's old.
- Person who's vehicle experienced from accident or damaged in the vehicle.
- Customer with the valid insurance policy to claim.

6. CUSTOMER CONSTRAINTS CC

- The most common constraints faced by the customer is network connection because of the internet availability

5. AVAILABLE SOLUTIONS AS

- Approaching third person for the cost estimation.
- Cost estimation done by manual calculation.
- Using slow processing algorithm to detect the damage.

Pros :

- The estimated value stays with in the customer and bank agent.

Cons :

- Estimated cost varies frequently.
- The time taken for estimation is very high leading to lots of losses and mental issues.

Explore AS, differentiate

Focus on J&P, tap into BE, understandRC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- One of the major problem s faced by the customers or the insurance companies are not having idea about the cost of repair for the damage.
- Insurance companies are failing to provide right amount for the car damage and the customers not able to claim for the damage

9. PROBLEM ROOT CAUSE RC

- Deviation or variation from the companies calculated cost and the actual cost
- Rapid development in the AI field paved way too many advanced methodologies of cost estimation.
- Customer have to do it the cost of the change in regulations.

7. BEHAVIOUR BE

- The customer has to upload the images of their car after the accident.
- The application will instantly evlaluate the damage and displace the claim amount to the customer.

Focus on J&P, tap into BE, understandRC

Identify strong TR &

3. TRIGGERS TR

- Reading about the more solutions in the news and various websites.
- Development of new technologies.

4. EMOTIONS: BEFORE / AFTER EM

Before the customer are not able to claim accurate amount for the damage in vehicle. After the technology development the customer felt independent and comfortable to use the technologies and the solution can be more.

10. YOUR SOLUTION SL

- Accurately the estimate the damage percentage.
- Predict the region of damage with respect to the vehicle.
- Eliminating human error while estimation.
- Use of fast processing algorithm for functionality

1. CHANNELS of BEHAVIOUR CH

1. ONLINE

Customer interact with the webpage through internet.

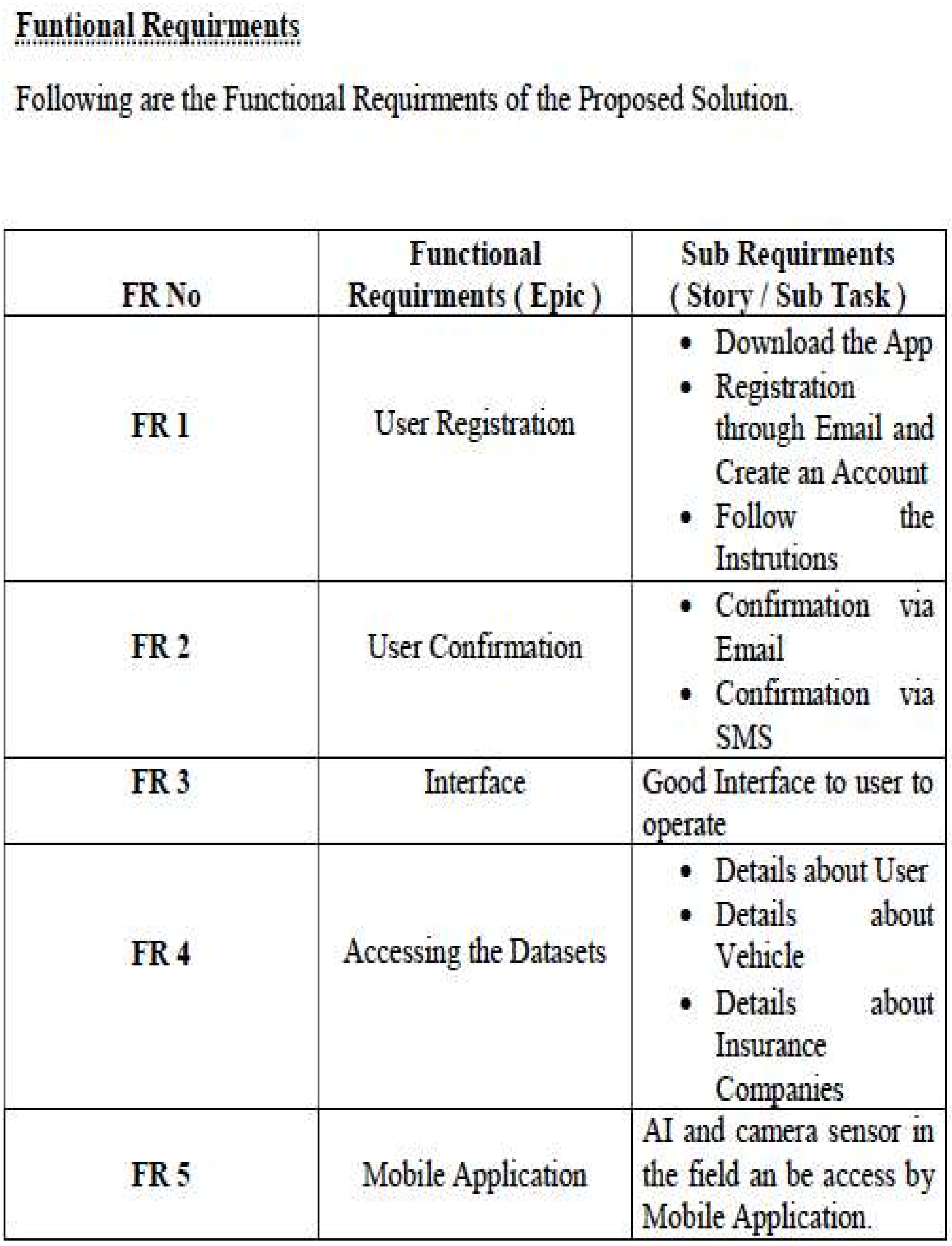
8.2 OFFLINE

Customer cannot access this webpage without internet.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

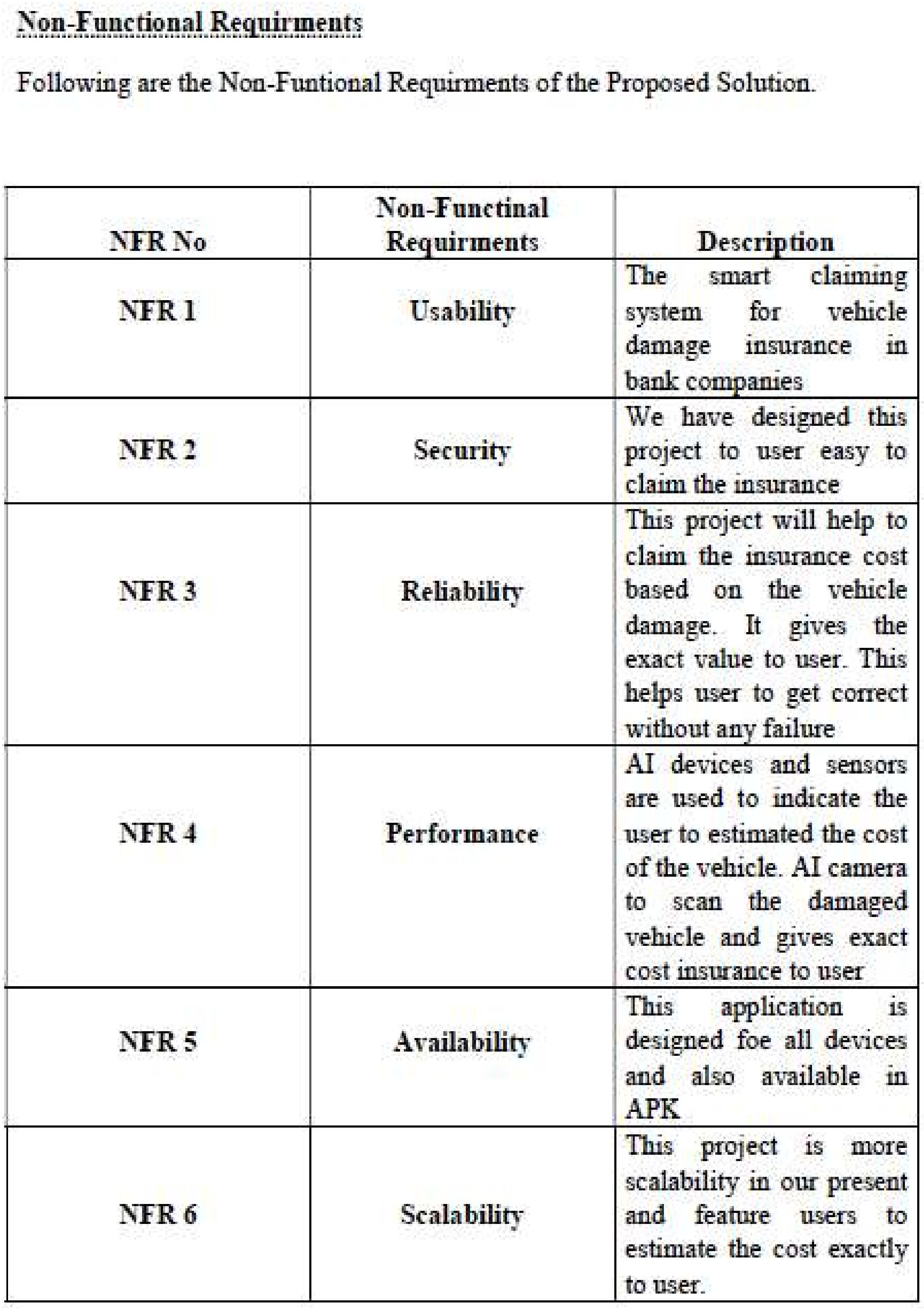


Funtional Requirments

Following are the Functional Requirments of the Proposed Solution.

FR No	Functional Requirments (Epic)	Sub Requirments (Story / Sub Task)
FR 1	User Registration	• Download the App • Registration through Email and Create an Account • Follow the Instrutions
FR 2	User Confirmation	• Confirmation via Email • Confirmation via SMS
FR 3	Interface	Good Interface to user to operate
FR 4	Accessing the Datasets	• Details about User • Details about Vehicle • Details about Insurance Companies
FR 5	Mobile Application	AI and camera sensor in the field an be access by Mobile Application.

4.2 Non Functional Requirements



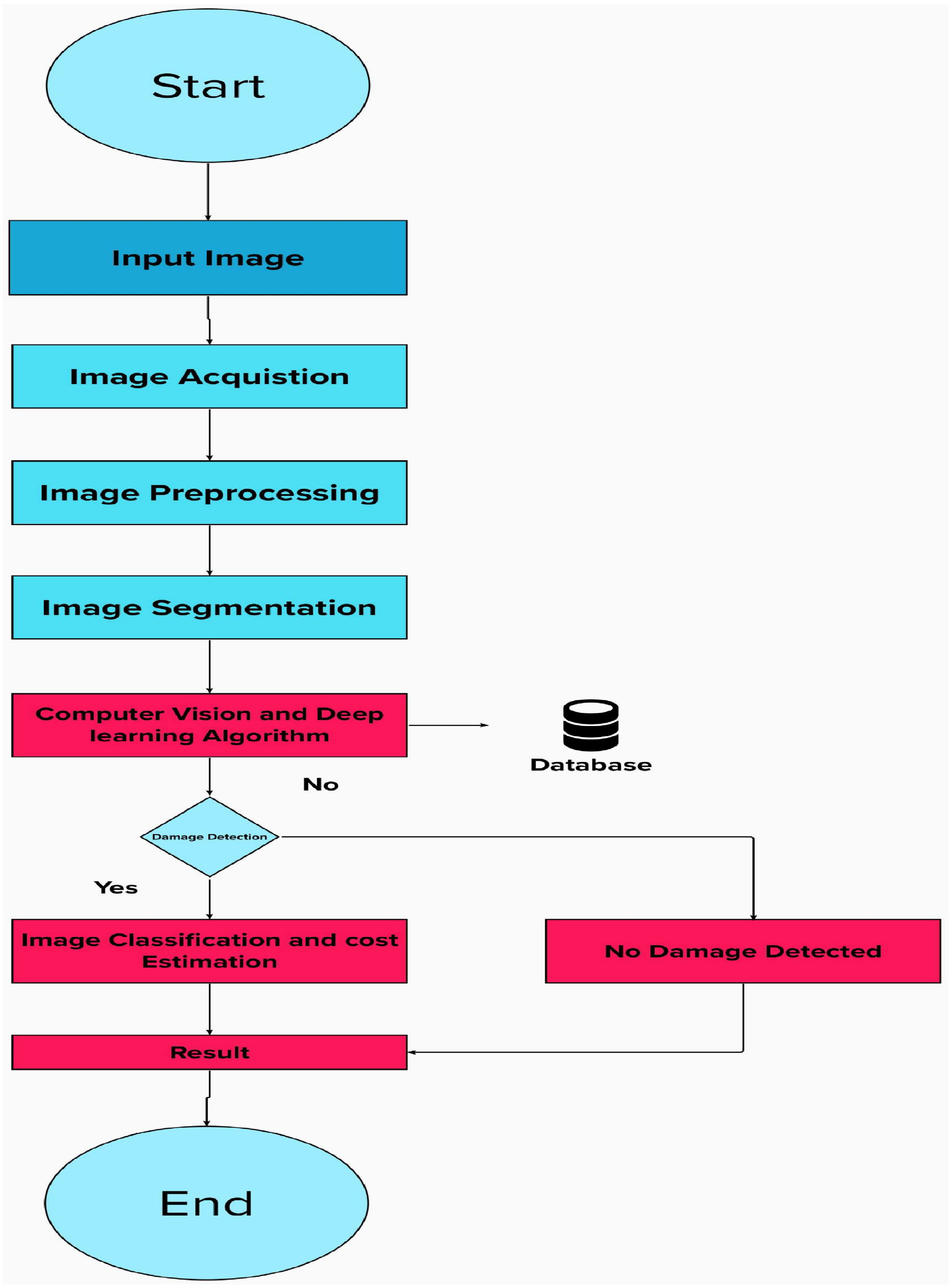
Non-Functional Requirments

Following are the Non-Funtional Requirments of the Proposed Solution.

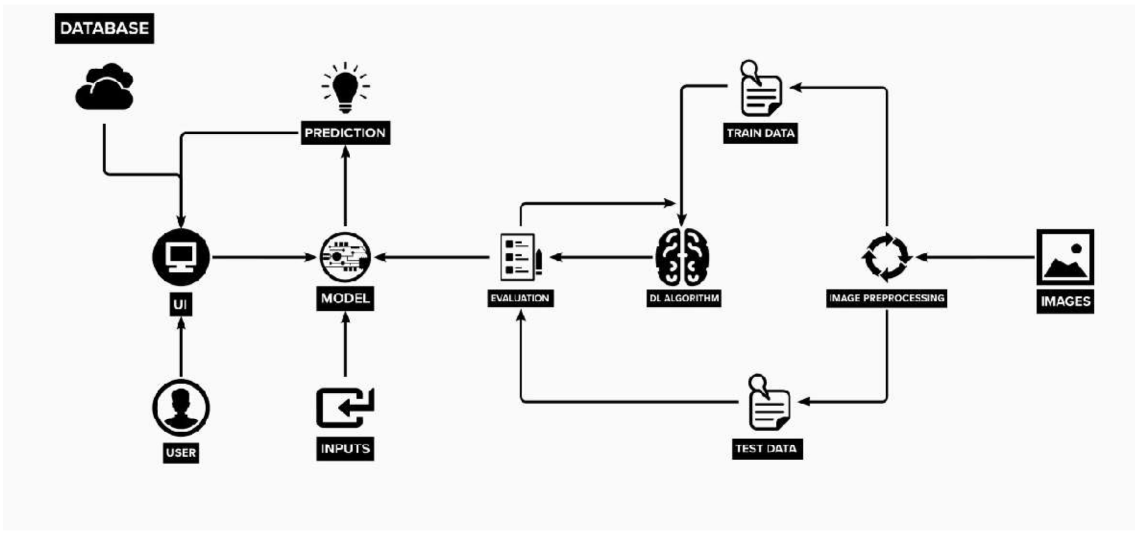
NFR No	Non-Functinal Requirments	Description
NFR 1	**Usability**	The smart claiming system for vehicle damage insurance in bank companies
NFR 2	**Security**	We have designed this project to user easy to claim the insurance
NFR 3	**Reliability**	This project will help to claim the insurance cost based on the vehicle damage. It gives the exact value to user. This helps user to get correct without any failure
NFR 4	**Performance**	AI devices and sensors are used to indicate the user to estimated the cost of the vehicle. AI camera to scan the damaged vehicle and gives exact cost insurance to user
NFR 5	**Availability**	This application is designed foe all devices and also available in APK
NFR 6	**Scalability**	This project is more scalability in our present and feature users to estimate the cost exactly to user.

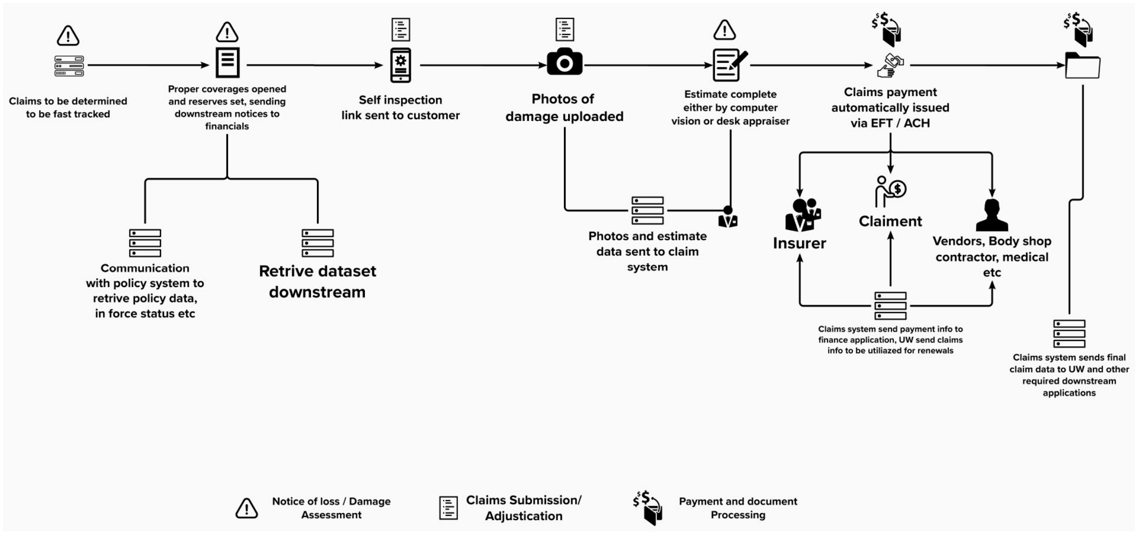
5. PROJECT DESIGN

5.1 Dataflow Diagrams

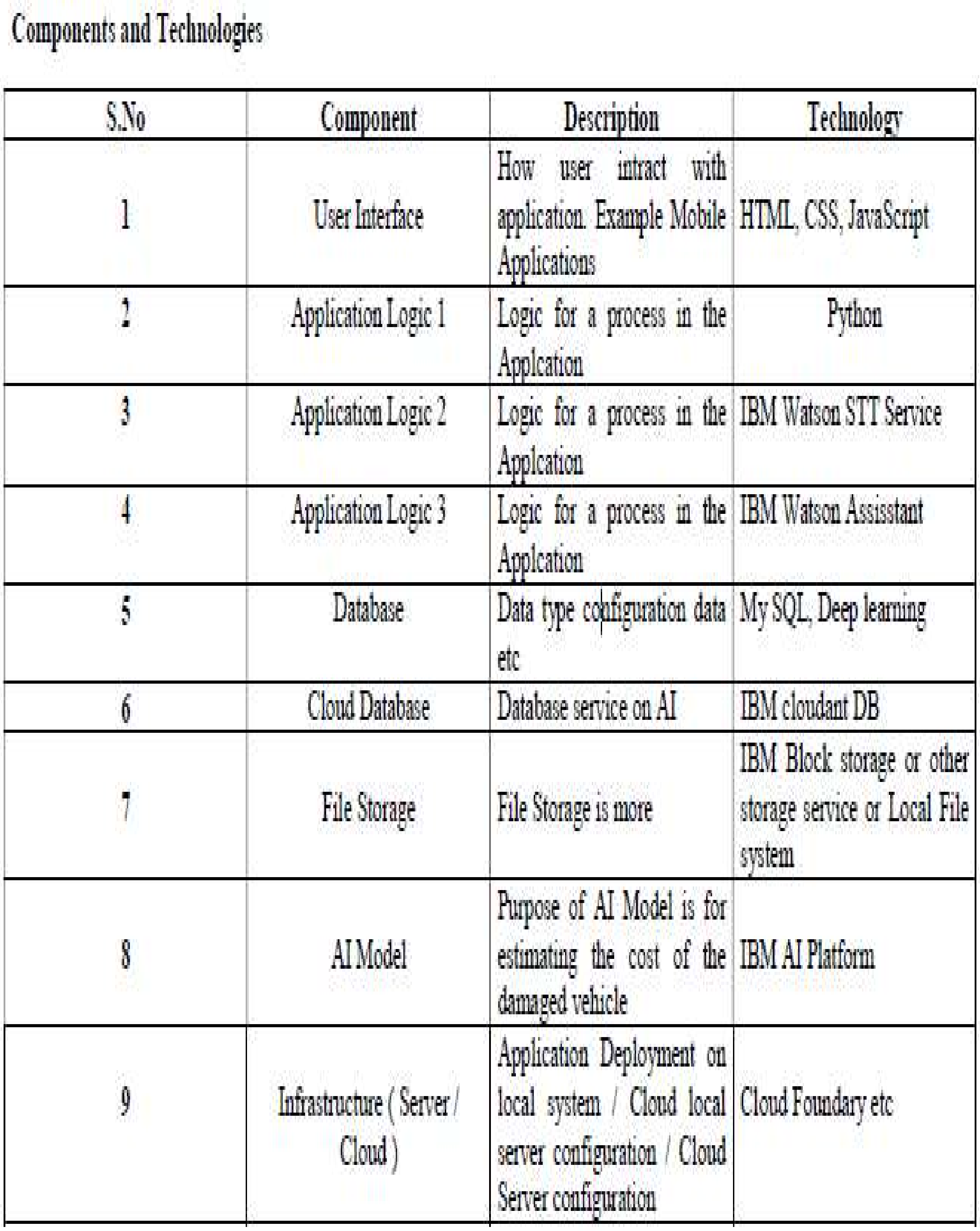


5.2 Solution and Technical Architecture




Claims to be determined to be fast tracked
Proper coverages opened and reserves set, sending downstream notices to financials
Self inspection link sent to customer
Photos of damage uploaded
Estimate complete either by computer vision or desk appraiser
Claims payment automatically issued via EFT / ACH
Communication with policy system to retrive policy data, in force status etc
Retrive dataset downstream
Photos and estimate data sent to claim system
Insurer
Claiment
Vendors, Body shop contractor, medical etc
Claims system send payment info to finance application, UW send claims info to be utiliazed for renewals
Claims system sends final claim data to UW and other required downstream applications
Notice of loss / Damage Assessment
Claims Submission/ Adjustication
Payment and document Processing

5.3 User Stories

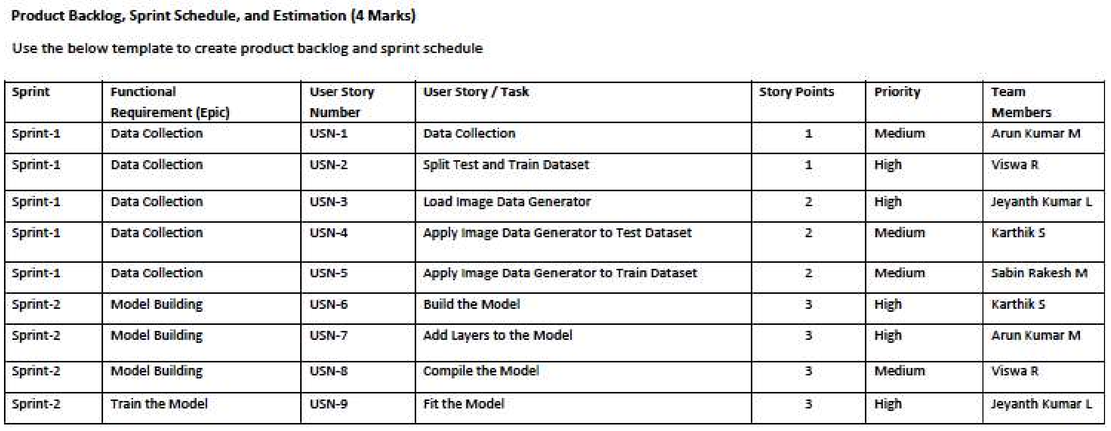


Components and Technologies

S.No	Component	Description	Technology
1	User Interface	How user intract with application. Example Mobile Applications	HTML, CSS, JavaScript
2	Application Logic 1	Logic for a process in the Applcation	Python
3	Application Logic 2	Logic for a process in the Applcation	IBM Watson STT Service
4	Application Logic 3	Logic for a process in the Applcation	IBM Watson Assisstant
5	Database	Data type configuration data etc	My SQL, Deep learning
6	Cloud Database	Database service on AI	IBM cloudant DB
7	File Storage	File Storage is more	IBM Block storage or other storage service or Local File system
8	AI Model	Purpose of AI Model is for estimating the cost of the damaged vehicle	IBM AI Platform
9	Infrastructure (Server / Cloud)	Application Deployment on local system / Cloud local server configuration / Cloud Server configuration	Cloud Foundary etc

6. PROJECT PLANING AND SCHEDULING

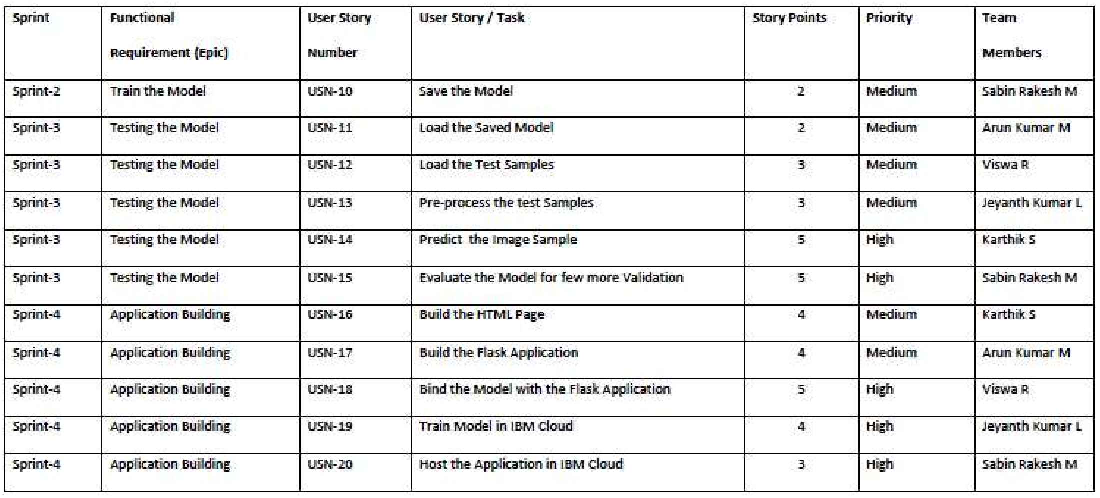
6.1 Sprint Planning and Estimation



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

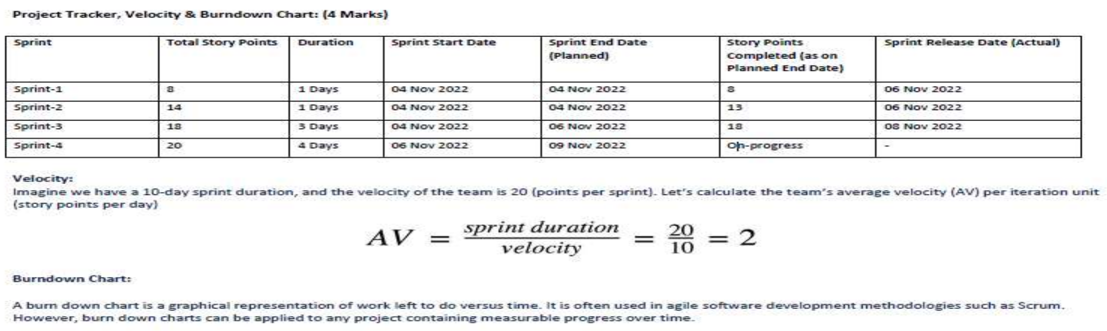
Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Data Collection	1	Medium	Arun Kumar M
Sprint-1	Data Collection	USN-2	Split Test and Train Dataset	1	High	Viswa R
Sprint-1	Data Collection	USN-3	Load Image Data Generator	2	High	Jeyanth Kumar L
Sprint-1	Data Collection	USN-4	Apply Image Data Generator to Test Dataset	2	Medium	Karthik S
Sprint-1	Data Collection	USN-5	Apply Image Data Generator to Train Dataset	2	Medium	Sabin Rakesh M
Sprint-2	Model Building	USN-6	Build the Model	3	High	Karthik S
Sprint-2	Model Building	USN-7	Add Layers to the Model	3	High	Arun Kumar M
Sprint-2	Model Building	USN-8	Compile the Model	3	Medium	Viswa R
Sprint-2	Train the Model	USN-9	Fit the Model	3	High	Jeyanth Kumar L



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Train the Model	USN-10	Save the Model	2	Medium	Sabin Rakesh M
Sprint-3	Testing the Model	USN-11	Load the Saved Model	2	Medium	Arun Kumar M
Sprint-3	Testing the Model	USN-12	Load the Test Samples	3	Medium	Viswa R
Sprint-3	Testing the Model	USN-13	Pre-process the test Samples	3	Medium	Jeyanth Kumar L
Sprint-3	Testing the Model	USN-14	Predict the Image Sample	5	High	Karthik S
Sprint-3	Testing the Model	USN-15	Evaluate the Model for few more Validation	5	High	Sabin Rakesh M
Sprint-4	Application Building	USN-16	Build the HTML Page	4	Medium	Karthik S
Sprint-4	Application Building	USN-17	Build the Flask Application	4	Medium	Arun Kumar M
Sprint-4	Application Building	USN-18	Bind the Model with the Flask Application	5	High	Viswa R
Sprint-4	Application Building	USN-19	Train Model in IBM Cloud	4	High	Jeyanth Kumar L
Sprint-4	Application Building	USN-20	Host the Application in IBM Cloud	3	High	Sabin Rakesh M

6.2 Sprint Delivery Schedule



Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	8	1 Days	04 Nov 2022	04 Nov 2022	8	06 Nov 2022
Sprint-2	14	1 Days	04 Nov 2022	04 Nov 2022	13	06 Nov 2022
Sprint-3	18	3 Days	04 Nov 2022	06 Nov 2022	18	08 Nov 2022
Sprint-4	20	4 Days	06 Nov 2022	09 Nov 2022	On-progress	-

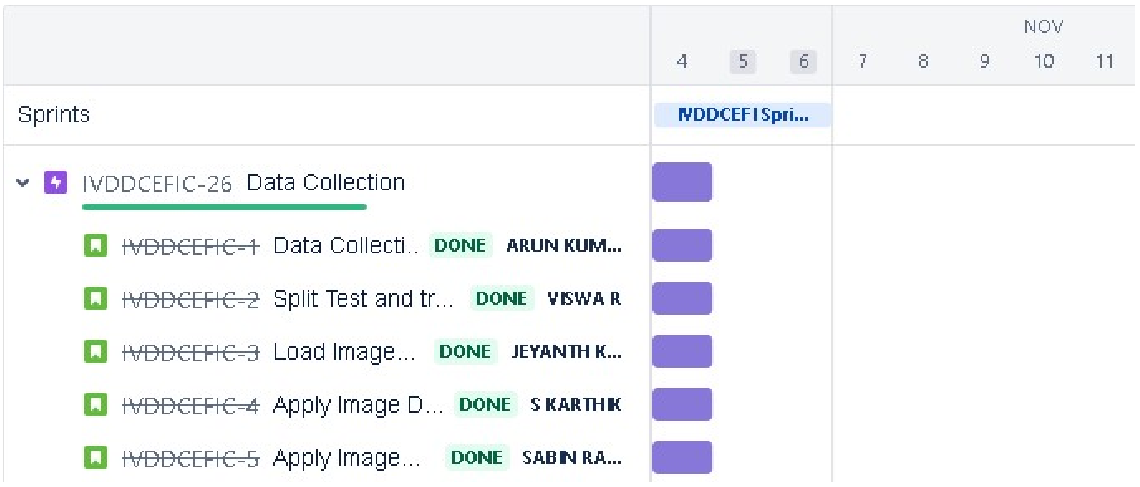
Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

6.3 Reports from JIRA



NOV 4 5 6 7 8 9 10 11

Sprints — IVDDCEFI Spri...

IVDDCEFIC-26 Data Collection

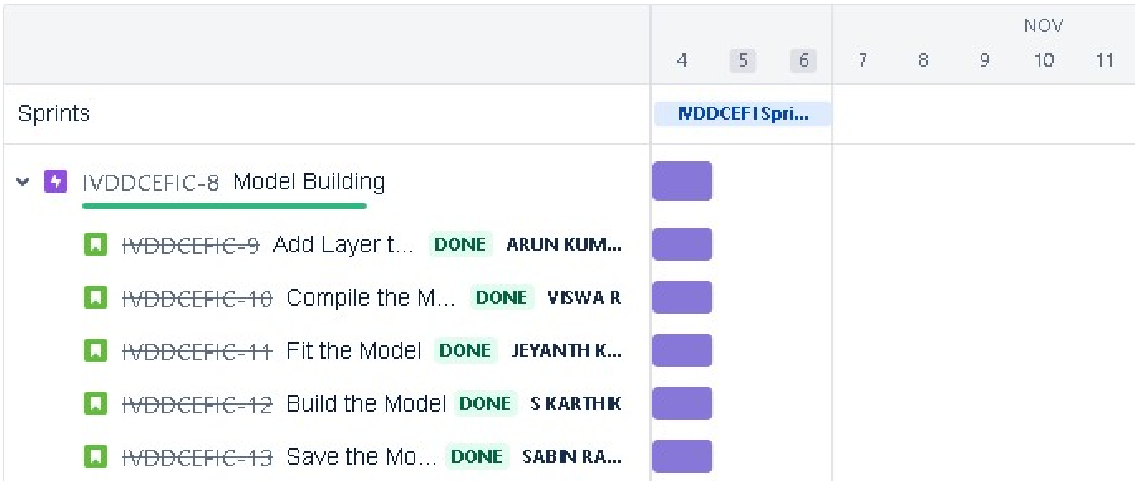
~~IVDDCEFIC-1~~ Data Collecti.. DONE ARUN KUM...

~~IVDDCEFIC-2~~ Split Test and tr... DONE VISWA R

~~IVDDCEFIC-3~~ Load Image... DONE JEYANTH K...

~~IVDDCEFIC-4~~ Apply Image D... DONE S KARTHIK

~~IVDDCEFIC-5~~ Apply Image... DONE SABIN RA...



NOV 4 5 6 7 8 9 10 11

Sprints — IVDDCEFI Spri...

IVDDCEFIC-8 Model Building

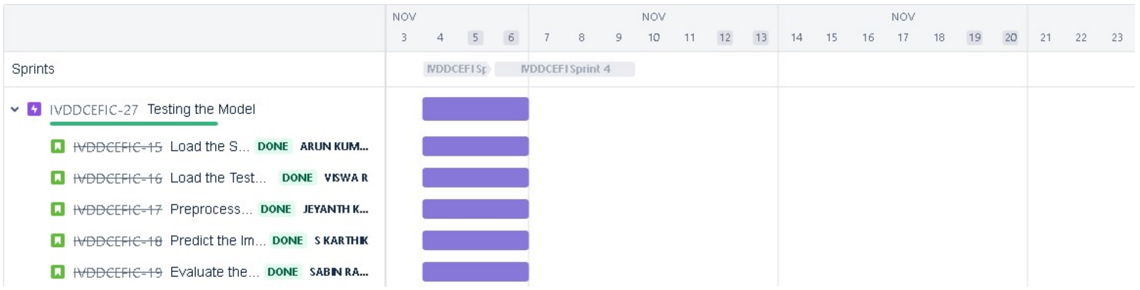
~~IVDDCEFIC-9~~ Add Layer t... DONE ARUN KUM...

~~IVDDCEFIC-10~~ Compile the M... DONE VISWA R

~~IVDDCEFIC-11~~ Fit the Model DONE JEYANTH K...

~~IVDDCEFIC-12~~ Build the Model DONE S KARTHIK

~~IVDDCEFIC-13~~ Save the Mo... DONE SABIN RA...



NOV 3 4 5 6 7 8 9 NOV 10 11 12 13 14 15 16 NOV 17 18 19 20 21 22 23

Sprints — IVDDCEFI Sp — IVDDCEFI Sprint 4

IVDDCEFIC-27 Testing the Model

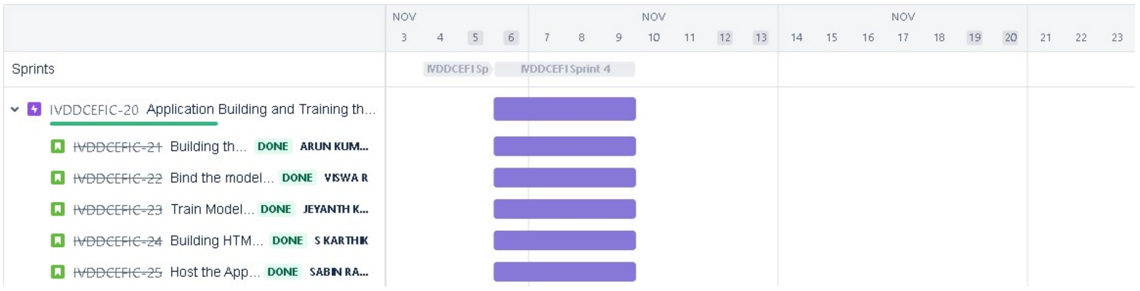
~~IVDDCEFIC-15~~ Load the S... DONE ARUN KUM...

~~IVDDCEFIC-16~~ Load the Test... DONE VISWA R

~~IVDDCEFIC-17~~ Preprocess... DONE JEYANTH K...

~~IVDDCEFIC-18~~ Predict the Im... DONE S KARTHIK

~~IVDDCEFIC-19~~ Evaluate the... DONE SABIN RA...



NOV 3 4 5 6 7 8 9 NOV 10 11 12 13 14 15 16 NOV 17 18 19 20 21 22 23

Sprints — IVDDCEFI Sp — IVDDCEFI Sprint 4

IVDDCEFIC-20 Application Building and Training th...

~~IVDDCEFIC-21~~ Building th... DONE ARUN KUM...

~~IVDDCEFIC-22~~ Bind the model... DONE VISWA R

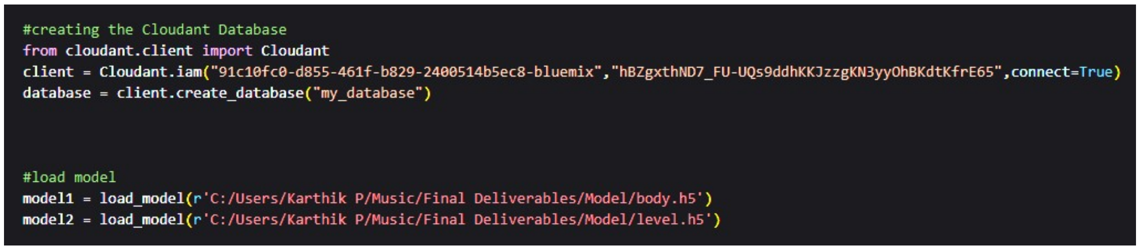
~~IVDDCEFIC-23~~ Train Model... DONE JEYANTH K...

~~IVDDCEFIC-24~~ Building HTM... DONE S KARTHIK

~~IVDDCEFIC-25~~ Host the App... DONE SABIN RA...

7. CODING AND SOLUTIONING

7.1 Feature 1



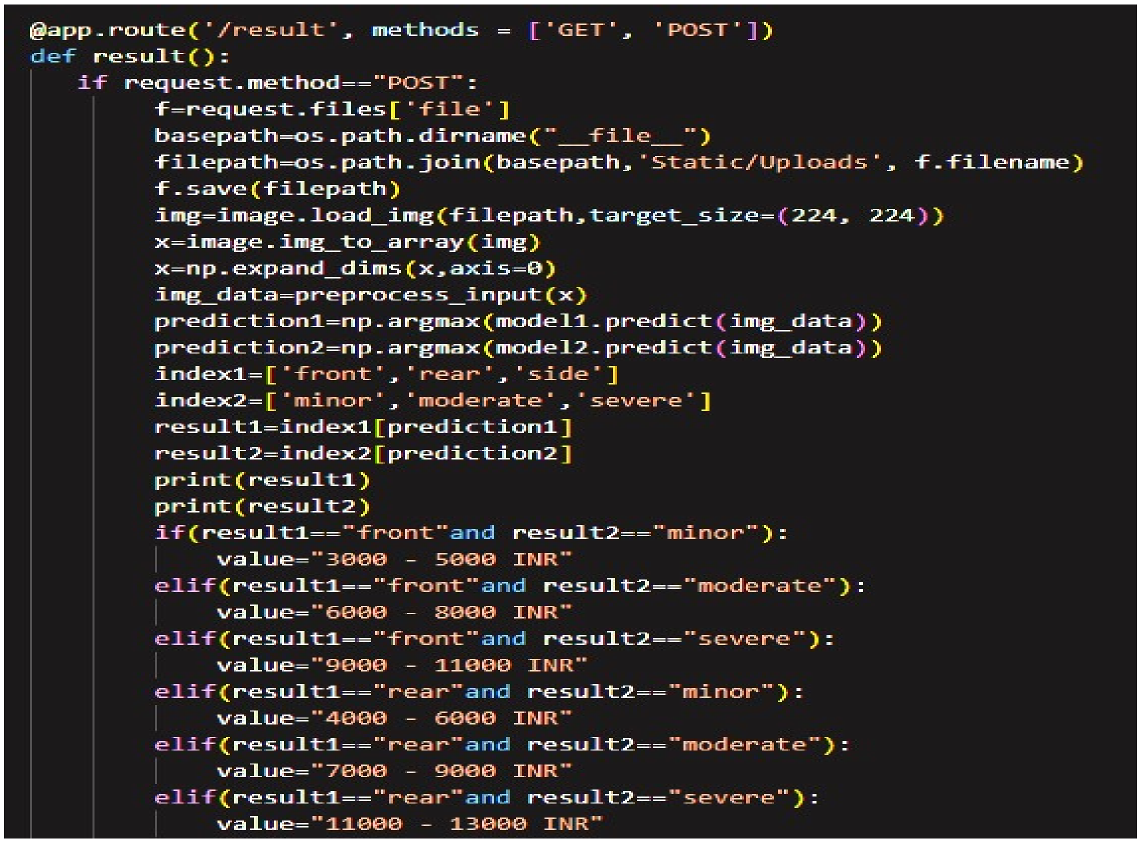
```
#creating the Cloudant Database
from cloudant.client import Cloudant
client = Cloudant.iam("91c10fc0-d855-461f-b829-2400514b5ec8-bluemix","hBZgxthND7_FU-UQs9ddhKKJzzgKN3yyOhBKdtKfrE65",connect=True)
database = client.create_database("my_database")



#load model
model1 = load_model(r'C:/Users/Karthik P/Music/Final Deliverables/Model/body.h5')
model2 = load_model(r'C:/Users/Karthik P/Music/Final Deliverables/Model/level.h5')
```

The feature 1 gives access to the trained deep learning models for predicting multiple damages in various areas in the vehicle and connected with the IBM Watson Database for storing the user data.

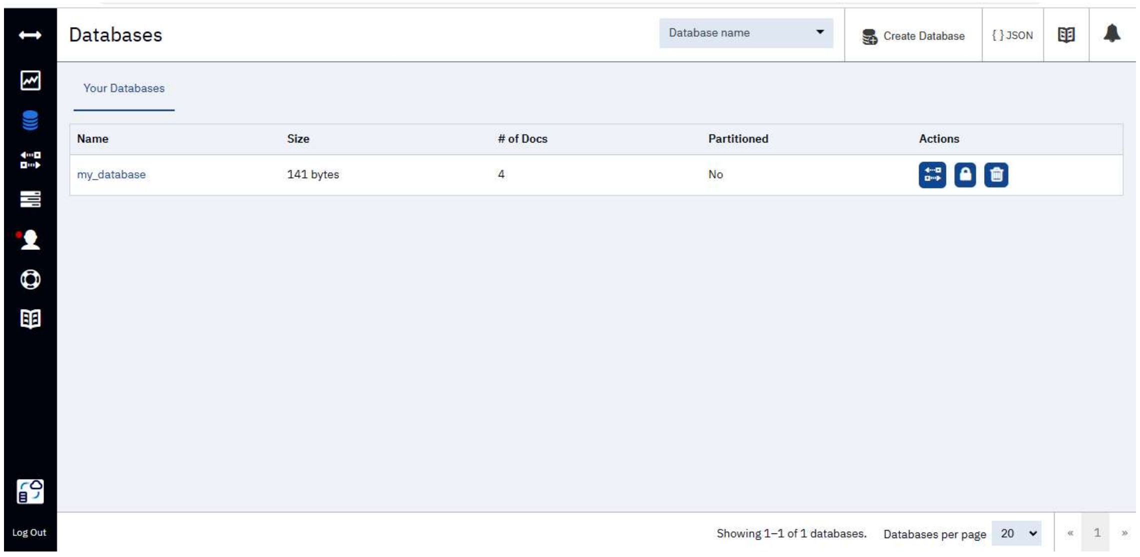
7.2 Feature 2



```
@app.route('/result', methods = ['GET', 'POST'])
def result():
    if request.method=="POST":
        f=request.files['file']
        basepath=os.path.dirname("__file__")
        filepath=os.path.join(basepath,'Static/Uploads', f.filename)
        f.save(filepath)
        img=image.load_img(filepath,target_size=(224, 224))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        img_data=preprocess_input(x)
        prediction1=np.argmax(model1.predict(img_data))
        prediction2=np.argmax(model2.predict(img_data))
        index1=['front','rear','side']
        index2=['minor','moderate','severe']
        result1=index1[prediction1]
        result2=index2[prediction2]
        print(result1)
        print(result2)
        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"
```

Feature 2 enables the web application to predict the incoming image from the user into the given labels. The code gets the image, convert into pixels and load into the model. Based on the predicted results, the algorithm will returns the value as the estimated cost.

7.3 Database Schema


Databases

Database name | Create Database | { } JSON

Your Databases

Name	Size	# of Docs	Partitioned	Actions
my_database	141 bytes	4	No	

Showing 1–1 of 1 databases. Databases per page 20

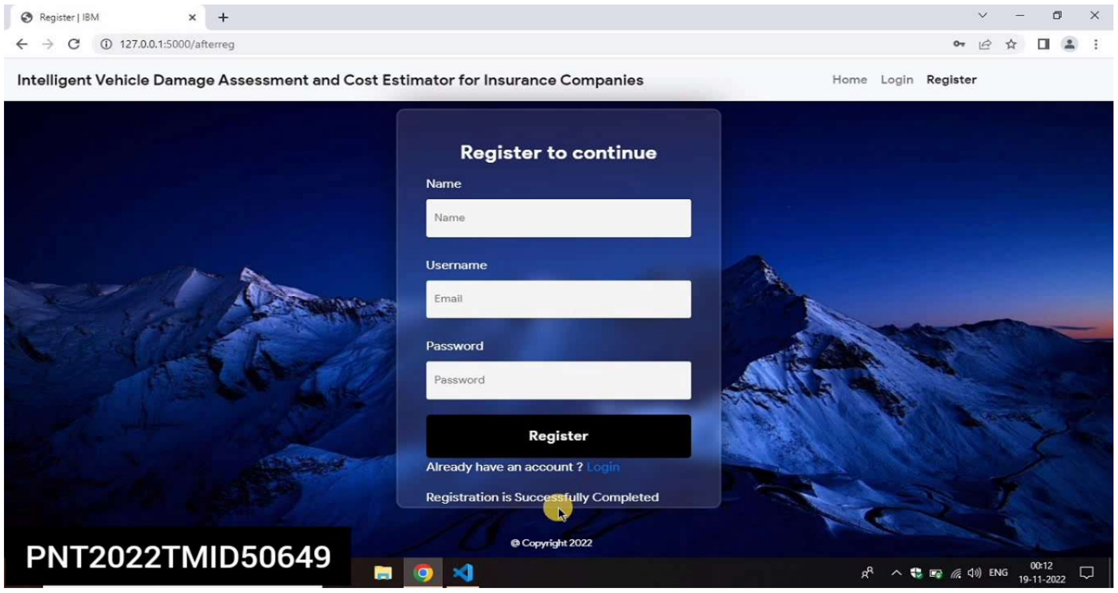
Log Out

8. TESTING

8.1 Test Cases

- User Login and Registration Test
- Database Update Test
- Prediction Test

8.2 User Acceptance Testing


Register | IBM

127.0.0.1:5000/afterreg

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

Home Login Register

Register to continue

Name: Name

Username: Email

Password: Password

Register

Already have an account ? Login

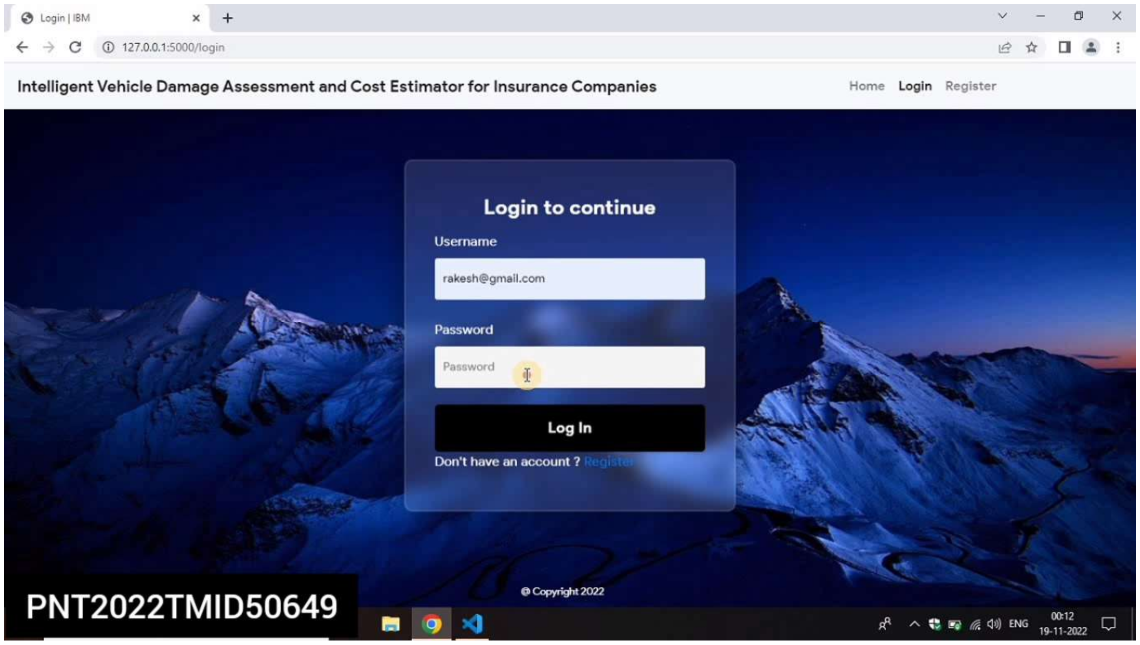
Registration is Successfully Completed

© Copyright 2022

PNT2022TMID50649

00:12 19-11-2022

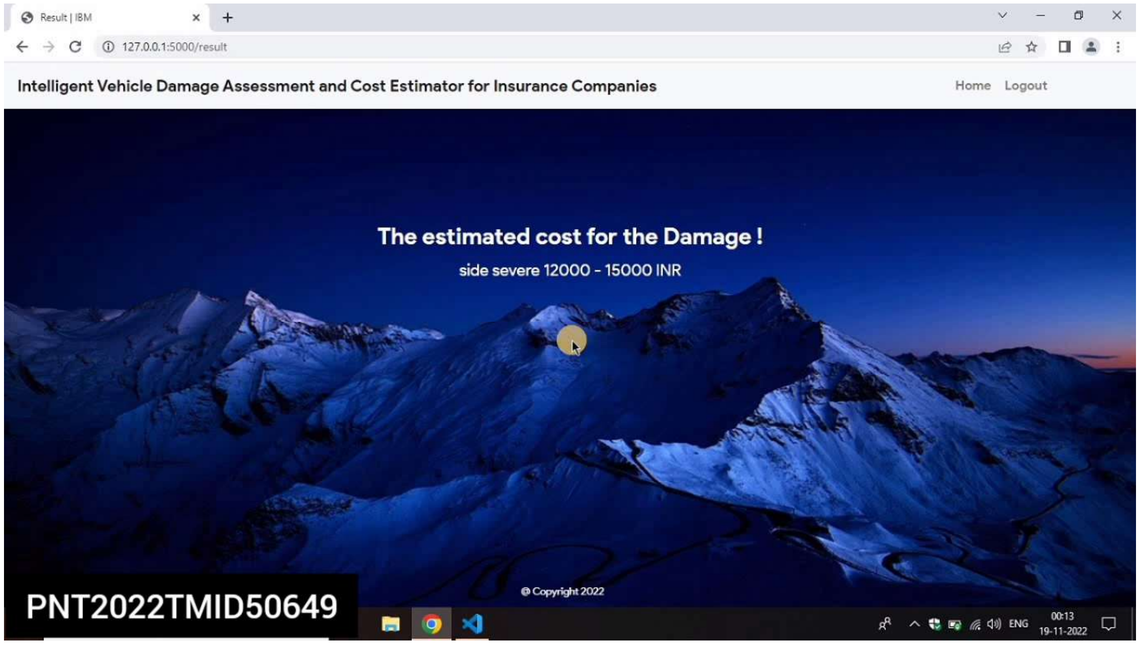
- The registration web page is tested with the already registered user information and hence it shows a message "You are already a member" by which the repeation of user information at database is prevented.


Login | IBM

127.0.0.1:5000/login

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

- The login web page is tested with the invalid user information to check the invalid login testing into the webpage.


Result | IBM

127.0.0.1:5000/result

Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies

The estimated cost for the Damage !

side severe 12000 - 15000 INR

- The prediction page is given with the test image of a damaged car to check the accuracy of the models.

9. RESULTS

9.1 Performance Metrics

The performance of the Cost estimator for insurance companies is tested and assessed with the latency check, which is run over the prediction page. The time taken to load the image and predict the cost based on the damages in the vehicle is checked. The results show that the web application took less than 10s to provide the estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance.

- Repair cost optimization, total loss and agreed value
- Quick assessment by phone – without the need for a visit by the professional inspector
- Overseeing the repair of the vehicle
- Establishing the monetary and residual value of vehicles
- Assistance in court
- Accident investigation to check all the data provided on the claim file
- Our reports and dataset are customized and adapted to your workflow, minimizing changes to your processes

The results show that the web application took less than 10s to provide the estimated cost of the given vehicle image. The model is tested with the various damaged car images which is not used during the training and validation of the model which also shows that the model works with the accuracy of about 98% in the overall performance.

10. ADVANTAGES

- The Advantage of having an Intelligent Cost Estimator based on the damages can save the time and resource of the user in automatically evaluating the images with the damages using the Deep Learning models trained with the various car images

- **Finding a proper data set-** Training machine learning models requires a sufficient data set of relevant images. The more varied the images are, the better the model will be able classify images appropriately

- Preprocessing image data sets is a crucial step in speeding up and obtaining better training results for models. This activity may span a variety of tasks: applying filters, removing noise, enhancing contrast, down sampling images, etc.

- **Building a model-** After you have a quality data set at hand, there are still some considerations when building a machine learning model.

DISADVANTAGES

- The Disadvantage of the project is expensive coding and time to develop the front end and back end of the web application
- Creating and training a model takes time
- **Optimizing performance and costs -** As insurance companies have to deal with damage assessment on a daily basis, the working solution also needs to demonstrate resonating performance

11. CONCLUSION

We conclude by suggesting this web application for damage assessment and cost estimation for the insurance companies. The web application is supported by the Deep Learning and IBM Watson cloud which stands for the complex image prediction and user information storage. The web application takes the user registration and login, The user can login into the prediction page using their ID and password. The prediction takes the image input and the model can predict the input based on the perviour knowledge about the damages. In future, The User Interface of the web application can be improved by updating the HTML and CSS coding. The improvement in UI can gives the better user experience in future, The model's accuracy over various images can increased by training with various damaged images. The Image processing methods can be improved to achieve higher performance of the model in the future.

12. FUTURE SCOPE

In future, The User Interface of the web application can be improved by updating the HTML and CSS codings. The improvement in UI can gives the better user exprience in future, The model's accuracy over various images can increased by trainning with various damaged images. The Image processing methods can be improved to achive higher performance of the model in the future.

13. APPENDIX

Github Link - https://github.com/IBM-EPBL/IBM-Project-45061-1660728082

Demo Video - https://vimeo.com/773670727

app.py

```
import numpy as np
import os
from flask import Flask, app,request,render_template,redirect,url_for,session
from tensorflow.keras import models
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from tensorflow.keras.applications.inception_v3 import preprocess_input
import requests
import webbrowser

os.add_dll_directory

#creating the Cloudant Database
from cloudant.client import Cloudant
client = Cloudant.iam("91c10fc0-d855-461f-b829-2400514b5ec8-
bluemix","hBZgxthND7_FU-UQs9ddhKKJzzgKN3yyOhBKdtKfrE65",connect=True)
database = client.create_database("my_database")



#load model
model1 = load_model(r'C:/Users/Karthik P/Music/Final
Deliverables/Model/body.h5')
model2 = load_model(r'C:/Users/Karthik P/Music/Final
Deliverables/Model/level.h5')

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

#login page setting

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST','GET'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id':{'$eq':user}}

    docs = database.get_query_result(query)
    print(docs)
    print(len(docs.all()))

    if(len(docs.all())==0):
        return render_template('login.html',message='The username is not
found')
```

```
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            return render_template("login.html",message="Invalid User
Details")


#Register page setting

@app.route('/register')
def register():
    return render_template('register.html')

@app.route('/afterreg',methods=['POST'])
def afterregister():
    x = [x for x in request.form.values()]
    print(x)
    data = {
        '_id':x[1],
        'name':x[0],
        'psw' : x[2]
    }
    print(data)

    query = {'_id':{'$eq' : data['_id']}}
    docs = database.get_query_result(query)

    if(len(docs.all())==0):
        url = database.create_document(data)
        return render_template('register.html', message="Registration is
Successfully Completed")
    else:
        return render_template("register.html", message="You are already a
member!")

#prediction

@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

#logout page

@app.route('/logout')
def logout():
    return render_template('logout.html')
```

```
#results
from os.path import join, dirname, realpath

@app.route('/result', methods = ['GET', 'POST'])
def result():
   if request.method=="POST":
        f=request.files['file']
        basepath=os.path.dirname("__file__")
        filepath=os.path.join(basepath,'Static/Uploads', f.filename)
        f.save(filepath)
        img=image.load_img(filepath,target_size=(224, 224))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        img_data=preprocess_input(x)
        prediction1=np.argmax(model1.predict(img_data))
        prediction2=np.argmax(model2.predict(img_data))
        index1=['front','rear','side']
        index2=['minor','moderate','severe']
        result1=index1[prediction1]
        result2=index2[prediction2]
        print(result1)
        print(result2)
        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("result.html",
prediction=value,ayya=result1,amma=result2)

if (__name__ == '__main__'):
    app.run(debug=True)
```