

**MACHINE LEARNING APPROACH TO DETECT &
ANNOTATE DISEASES USING RETINAL IMAGGES**

2023-162

Status Document II

Lakshith G. P. R.

IT20165666

B.Sc. (Hons) Degree in Information Technology
Specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

Table of Contents

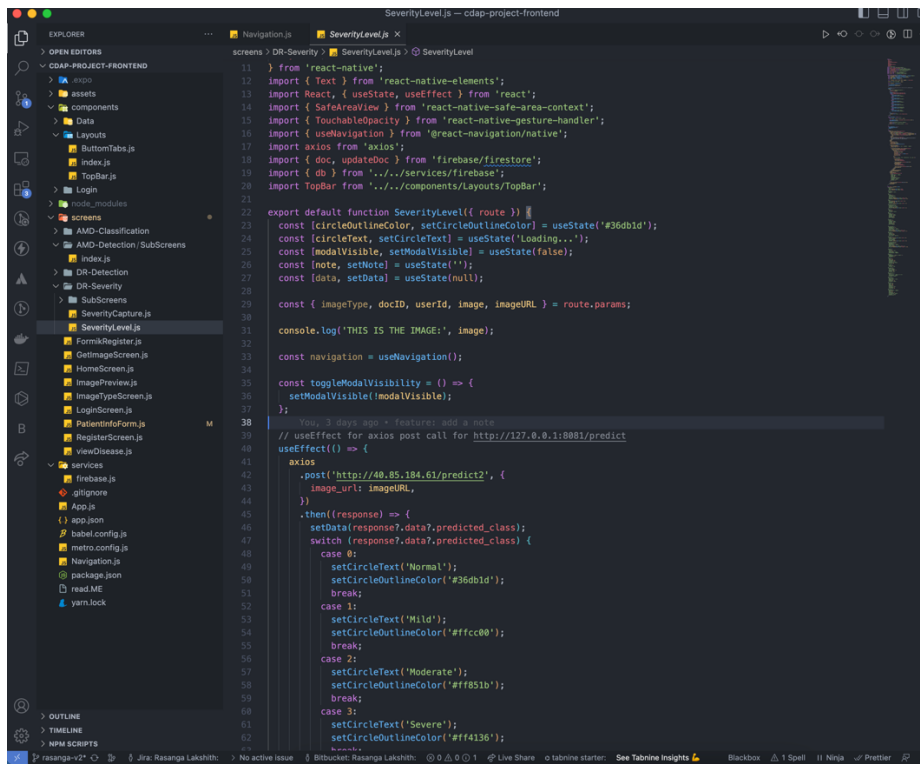
1. PROJECT PROGRESS	3
1.1. FRONT-END IMPLEMENTATION.....	3
1.2. BACK-END IMPLEMENTATION	4
1.3. MOBILE APP UI DESIGNS	6
2. PROJECT VIEW	7
3. GANTT CHART	8
4. SCREENSHOTS OF CONVERSATIONS AND CALLS – MS TEAMS.....	8

Table of Figures

Figure 1: Front-end code snip 1	3
Figure 2: Front-end code snip 2	3
Figure 3: Back-end Flask code.....	4
Figure 4: Back-end Service code	4
Figure 5: Custom made lightweight model.....	5
Figure 6: Preprocessed images	5
Figure 7: Planner - board view.....	7
Figure 8: Planner - chart view	7
Figure 9: Planner - schedule view	7
Figure 10: Gantt chart.....	8

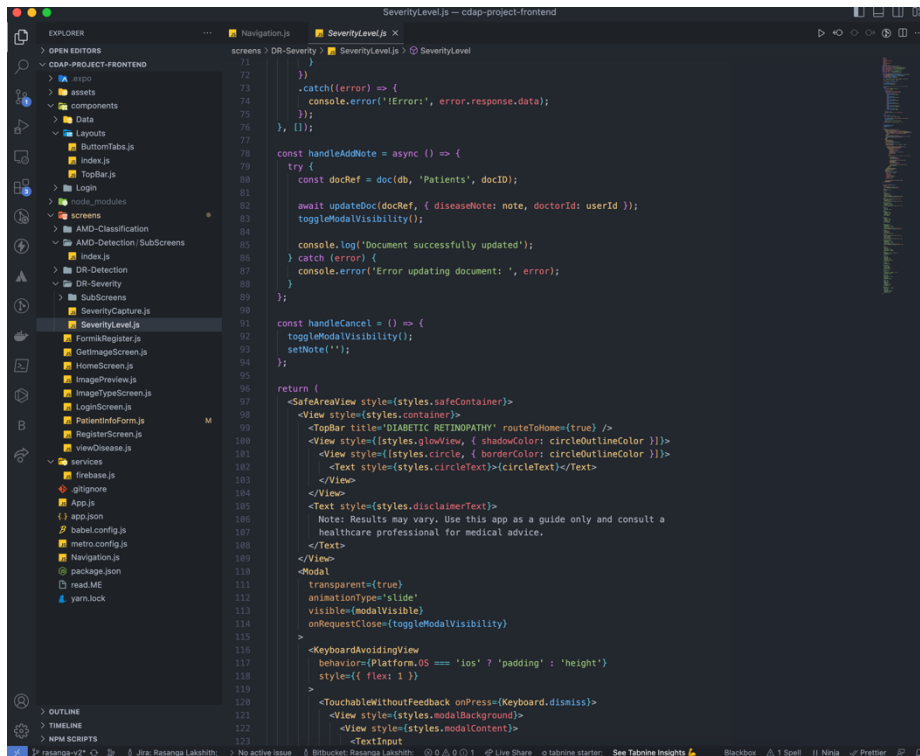
1. Project progress

1.1. Front-end Implementation



```
SeverityLevel.js — cdap-project-frontend
11  } from 'react-native';
12  import { Text } from 'react-native-elements';
13  import React, { useState, useEffect } from 'react';
14  import { SafeAreaView } from 'react-native-safe-area-context';
15  import { TouchableOpacity } from 'react-native-gesture-handler';
16  import { useNavigation } from '@react-navigation/native';
17  import axios from 'axios';
18  import { doc, updateDoc } from 'firebase/firestore';
19  import { db } from '../services/firebase';
20  import TopBar from '../components/Layouts/TopBar';
21
22  export default function SeverityLevel({ route }) {
23    const [circleOutlineColor, setCircleOutlineColor] = useState('#36dbd1');
24    const [circleText, setCircleText] = useState('Loading...');
25    const [modalVisible, setModalVisible] = useState(false);
26    const [note, setNote] = useState('');
27    const [data, setData] = useState(null);
28
29    const { imageType, docID, userID, image, imageURL } = route.params;
30
31    console.log('THIS IS THE IMAGE:', image);
32
33    const navigation = useNavigation();
34
35    const toggleModalVisibility = () => {
36      setModalVisible(!modalVisible);
37    };
38
39    // UseEffect for axios post call for http://127.0.0.1:8881/predict
40    useEffect(() => {
41      axios
42        .post('http://40.85.184.61/predict2', {
43          image_url: imageURL,
44        })
45        .then((response) => {
46          setData(response.data?.predicted_class);
47          switch (response.data?.predicted_class) {
48            case 0:
49              setCircleText('Normal');
50              setCircleOutlineColor('#36dbd1');
51              break;
52            case 1:
53              setCircleText('Mild');
54              setCircleOutlineColor('#ffcc00');
55              break;
56            case 2:
57              setCircleText('Moderate');
58              setCircleOutlineColor('#ff851b');
59              break;
60            case 3:
61              setCircleText('Severe');
62              setCircleOutlineColor('#ff4136');
63            default:
64              break;
65          }
66        });
67    }, [imageURL]);
68
69    const handleAddNote = async () => {
70      try {
71        const docRef = doc(db, 'Patients', docID);
72
73        await updateDoc(docRef, { diseaseNote: note, doctorId: userID });
74        toggleModalVisibility();
75
76        console.log('Document successfully updated!');
77      } catch (error) {
78        console.error('Error updating document: ', error);
79      }
80    };
81
82    const handleCancel = () => {
83      toggleModalVisibility();
84      setNote('');
85    };
86
87    return (
88      <SafeAreaView style={styles.safeContainer}>
89        <View style={styles.container}>
90          <TopBar title="DIABETIC RETINOPATHY" routeToHome={true} />
91          <View style={styles.glowView, { shadowColor: circleOutlineColor }}>
92            <View style={styles.circle, { borderColor: circleOutlineColor }}>
93              <Text style={styles.circleText}>{circleText}</Text>
94            </View>
95          </View>
96          <Text style={styles.disclaimerText}>
97            Note: Results may vary. Use this app as a guide only and consult a
98            healthcare professional for medical advice.
99          </Text>
100        </View>
101        <Modal
102          transparent={true}
103          animationType="slide"
104          visible={modalVisible}
105          onRequestClose={toggleModalVisibility}>
106          <KeyboardAvoidingView
107            behavior={Platform.OS === 'ios' ? 'padding' : 'height'}
108            style={{ flex: 1 }}>
109            <TouchableWithoutFeedback onPress={Keyboard.dismiss}>
110              <View style={styles.modalBackground}>
111                <View style={styles.modalContent}>
112                  <TextInput
113                    style={styles.textInput}
114                    value={note}
115                    onChangeText={setNote}
116                    placeholder="Enter your note here"
117                    multiline={true}
118                    numberOfLines={5}
119                  />
120                  <View style={styles.buttonContainer}>
121                    <Text style={styles.addNoteText}>Add Note</Text>
122                    <Text style={styles.cancelText}>Cancel</Text>
123                  </View>
124                </View>
125              </View>
126            </KeyboardAvoidingView>
127          </Modal>
128        </SafeAreaView>
129      </View>
130    );
131  }
132
133  const styles = {
134    safeContainer: {
135      flex: 1,
136      padding: 10,
137    },
138    container: {
139      flex: 1,
140      alignContent: 'center',
141      justify-content: 'center',
142    },
143    glowView: {
144      width: 100,
145      height: 100,
146      borderRadius: 50,
147      shadowOffset: { width: 0, height: 0 },
148      shadowOpacity: 0.5,
149      shadowRadius: 10,
150    },
151    circle: {
152      width: 80,
153      height: 80,
154      borderRadius: 40,
155      border: 2px solid '#36dbd1',
156    },
157    circleText: {
158      position: 'absolute',
159      top: 50%,
160      left: 50%,
161      transform: 'translate(-50%, -50%)',
162      color: 'white',
163      font-weight: 'bold',
164      font-size: 24,
165    },
166    disclaimerText: {
167      margin: 10,
168      color: 'gray',
169      font-size: 12,
170      text-align: 'center',
171    },
172    modalBackground: {
173      width: '100%',
174      height: '100%',
175      backgroundColor: 'white',
176    },
177    modalContent: {
178      padding: 20,
179    },
180    textInput: {
181      width: '100%',
182      height: 40,
183      border: 1px solid #ccc,
184      border-radius: 5px,
185      margin-bottom: 10,
186    },
187    buttonContainer: {
188      display: 'flex',
189      justify-content: 'space-around',
190    },
191    addNoteText: {
192      color: '#36dbd1',
193      font-weight: 'bold',
194    },
195    cancelText: {
196      color: 'gray',
197      font-weight: 'bold',
198    },
199  };
200
```

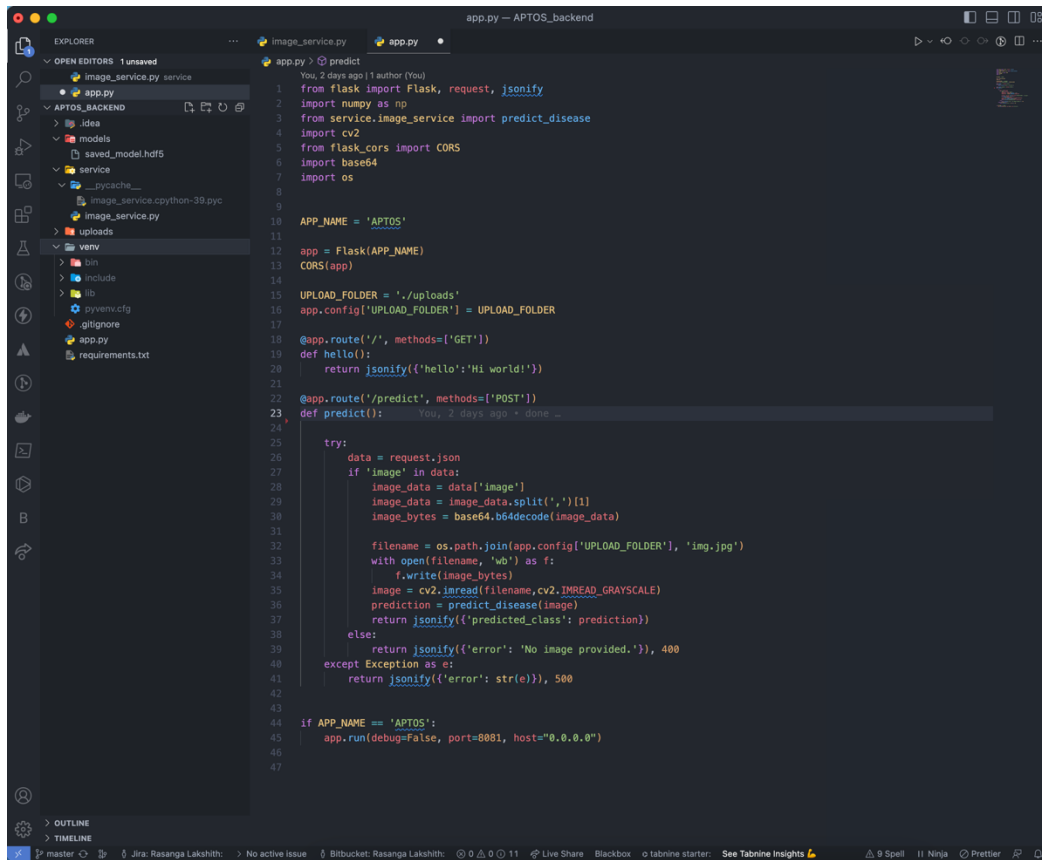
Figure 1: Front-end code snip 1



```
SeverityLevel.js — cdap-project-frontend
71  }
72  }
73  }
74  <catch(error) => {
75    console.error('Error:', error.response.data);
76  });
77  }, {});
78
79  const handleAddNote = async () => {
80    try {
81      const docRef = doc(db, 'Patients', docID);
82
83      await updateDoc(docRef, { diseaseNote: note, doctorId: userID });
84      toggleModalVisibility();
85
86      console.log('Document successfully updated!');
87    } catch (error) {
88      console.error('Error updating document: ', error);
89    }
90  };
91
92  const handleCancel = () => {
93    toggleModalVisibility();
94    setNote('');
95  };
96
97  return (
98    <SafeAreaView style={styles.safeContainer}>
99      <View style={styles.container}>
100        <TopBar title="DIABETIC RETINOPATHY" routeToHome={true} />
101        <View style={styles.glowView, { shadowColor: circleOutlineColor }}>
102          <View style={styles.circle, { borderColor: circleOutlineColor }}>
103            <Text style={styles.circleText}>{circleText}</Text>
104          </View>
105        </View>
106        <Text style={styles.disclaimerText}>
107          Note: Results may vary. Use this app as a guide only and consult a
108          healthcare professional for medical advice.
109        </Text>
110      </View>
111      <Modal
112        transparent={true}
113        animationType="slide"
114        visible={modalVisible}
115        onRequestClose={toggleModalVisibility}>
116        <KeyboardAvoidingView
117          behavior={Platform.OS === 'ios' ? 'padding' : 'height'}
118          style={{ flex: 1 }}>
119          <TouchableWithoutFeedback onPress={Keyboard.dismiss}>
120            <View style={styles.modalBackground}>
121              <View style={styles.modalContent}>
122                <TextInput
123                  style={styles.textInput}
124                  value={note}
125                  onChangeText={setNote}
126                  placeholder="Enter your note here"
127                  multiline={true}
128                  numberOfLines={5}
129                />
130                <View style={styles.buttonContainer}>
131                  <Text style={styles.addNoteText}>Add Note</Text>
132                  <Text style={styles.cancelText}>Cancel</Text>
133                </View>
134              </View>
135            </TouchableWithoutFeedback>
136          </KeyboardAvoidingView>
137        </Modal>
138      </SafeAreaView>
139    </View>
140  );
141
142  const styles = {
143    safeContainer: {
144      flex: 1,
145      padding: 10,
146    },
147    container: {
148      flex: 1,
149      alignContent: 'center',
150      justify-content: 'center',
151    },
152    glowView: {
153      width: 100,
154      height: 100,
155      borderRadius: 50,
156      shadowOffset: { width: 0, height: 0 },
157      shadowOpacity: 0.5,
158      shadowRadius: 10,
159    },
160    circle: {
161      width: 80,
162      height: 80,
163      borderRadius: 40,
164      border: 2px solid '#36dbd1',
165    },
166    circleText: {
167      position: 'absolute',
168      top: 50%,
169      left: 50%,
170      transform: 'translate(-50%, -50%)',
171      color: 'white',
172      font-weight: 'bold',
173      font-size: 24,
174    },
175    disclaimerText: {
176      margin: 10,
177      color: 'gray',
178      font-size: 12,
179      text-align: 'center',
180    },
181    modalBackground: {
182      width: '100%',
183      height: '100%',
184      backgroundColor: 'white',
185    },
186    modalContent: {
187      padding: 20,
188    },
189    textInput: {
190      width: '100%',
191      height: 40,
192      border: 1px solid #ccc,
193      border-radius: 5px,
194      margin-bottom: 10,
195    },
196    buttonContainer: {
197      display: 'flex',
198      justify-content: 'space-around',
199    },
200    addNoteText: {
201      color: '#36dbd1',
202      font-weight: 'bold',
203    },
204    cancelText: {
205      color: 'gray',
206      font-weight: 'bold',
207    },
208  };
209
```

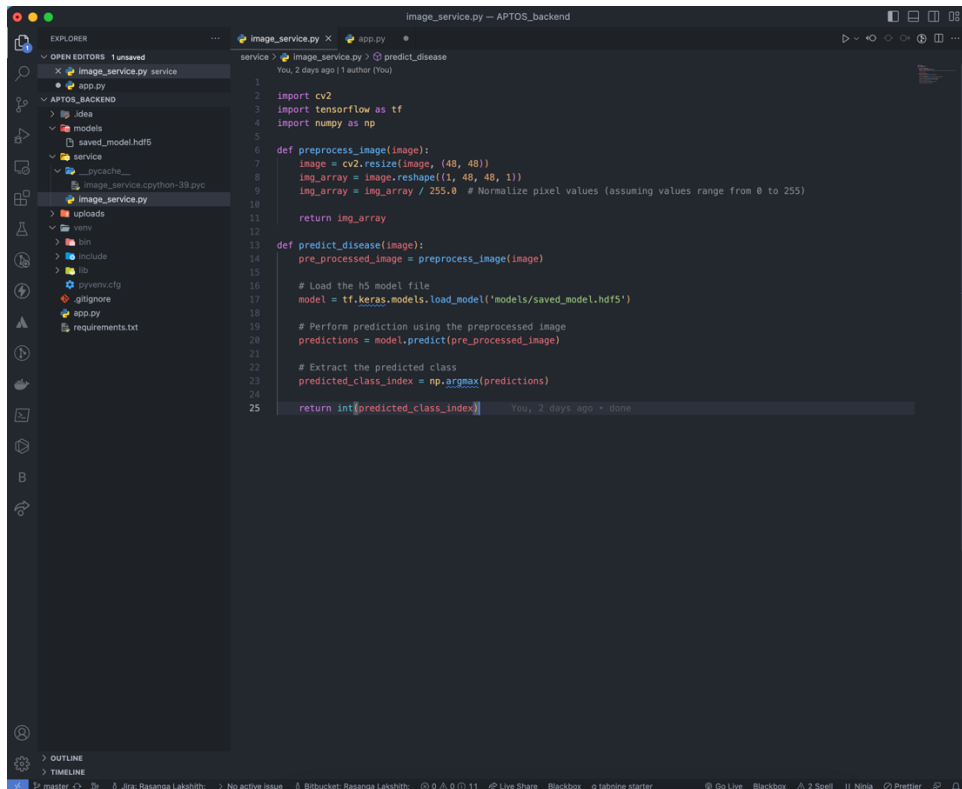
Figure 2: Front-end code snip 2

1.2. Back-end Implementation



```
app.py — APTOS_backend
1 from flask import Flask, request, jsonify
2 import numpy as np
3 from service.image_service import predict_disease
4 import cv2
5 from flask_cors import CORS
6 import base64
7 import os
8
9
10 APP_NAME = 'APTOS'
11
12 app = Flask(APP_NAME)
13 CORS(app)
14
15 UPLOAD_FOLDER = './uploads'
16 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
17
18 @app.route('/', methods=['GET'])
19 def hello():
20     return jsonify({'hello': 'Hi world!'})
21
22 @app.route('/predict', methods=['POST'])
23 def predict():
24     You, 2 days ago • done ...
25
26     try:
27         data = request.json
28         if 'image' in data:
29             image_data = data['image']
30             image_data = image_data.split(',')[1]
31             image_bytes = base64.b64decode(image_data)
32
33             filename = os.path.join(app.config['UPLOAD_FOLDER'], 'img.jpg')
34             with open(filename, 'wb') as f:
35                 f.write(image_bytes)
36             image = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
37             prediction = predict_disease(image)
38             return jsonify({'predicted_class': prediction})
39         else:
40             return jsonify({'error': 'No image provided.'}), 400
41     except Exception as e:
42         return jsonify({'error': str(e)}), 500
43
44 if APP_NAME == 'APTOS':
45     app.run(debug=False, port=8081, host="0.0.0.0")
46
47
```

Figure 3: Back-end Flask code



```
image_service.py — APTOS_backend
1 service > image_service.py > predict_disease
2 You, 2 days ago | 1 author (You)
3
4 1
5 2 import cv2
6 3 import tensorflow as tf
7 4 import numpy as np
8 5
9 6 def preprocess_image(image):
10 7     image = cv2.resize(image, (48, 48))
11 8     img_array = image.reshape((1, 48, 48, 1))
12 9     img_array = img_array / 255.0 # Normalize pixel values (assuming values range from 0 to 255)
13
14     return img_array
15
16 12 def predict_disease(image):
17 13     pre_processed_image = preprocess_image(image)
18 14
19 15     # Load the h5 model file
20 16     model = tf.keras.models.load_model('models/saved_model.hdf5')
21 17
22 18     # Perform prediction using the preprocessed image
23 19     predictions = model.predict(pre_processed_image)
24 20
25 21     # Extract the predicted class
26 22     predicted_class_index = np.argmax(predictions)
27 23
28 24     return int(predicted_class_index)
29 25
30 You, 2 days ago • done
```

Figure 4: Back-end Service code

```
# model
model = keras.Sequential()

model.add(tf.keras.layers.Conv2D(64, (3,3), input_shape = (48,48,1), padding="same"))
model.add(tf.keras.layers.MaxPooling2D((2,2)))

model.add(tf.keras.layers.Conv2D(64, (3,3), padding="same"))
model.add(tf.keras.layers.MaxPooling2D((2,2)))

model.add(tf.keras.layers.Dropout(0.2))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(5, activation = 'softmax'))

model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
conv2d_1 (Conv2D)	(None, 24, 24, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout (Dropout)	(None, 12, 12, 64)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 5)	46085

=====
Total params: 83,653
Trainable params: 83,653
Non-trainable params: 0

Figure 5: Custom made lightweight model

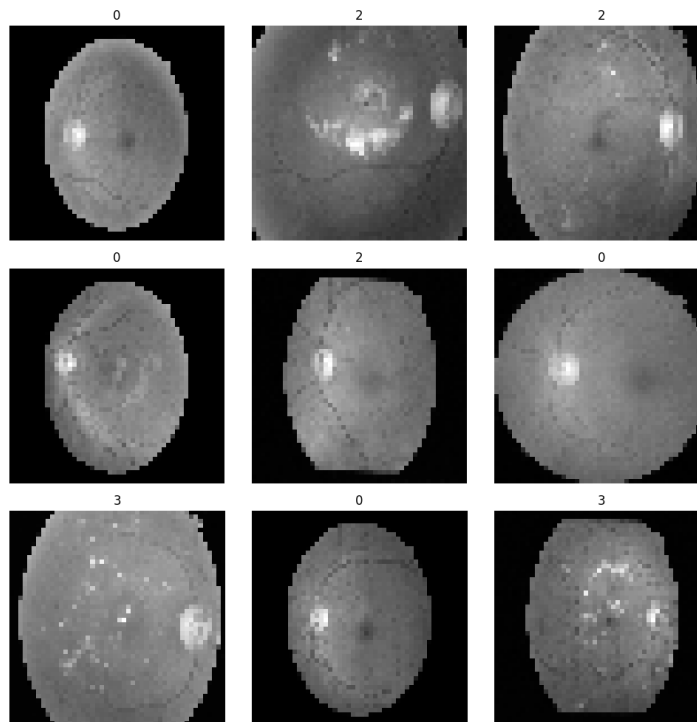
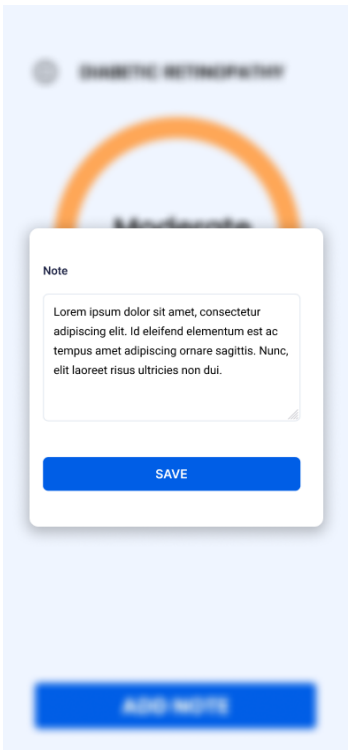
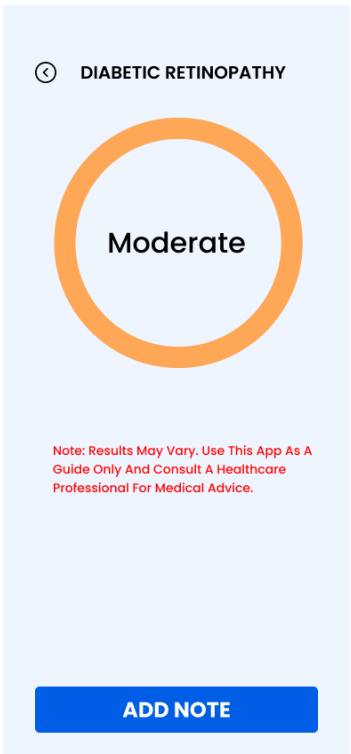
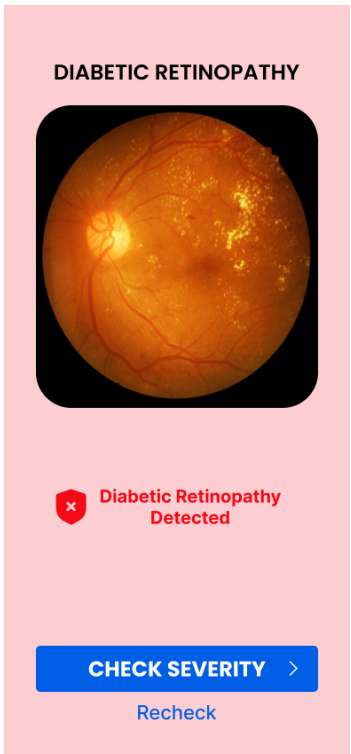


Figure 6: Preprocessed images

1.3. Mobile App UI designs



2. Project view

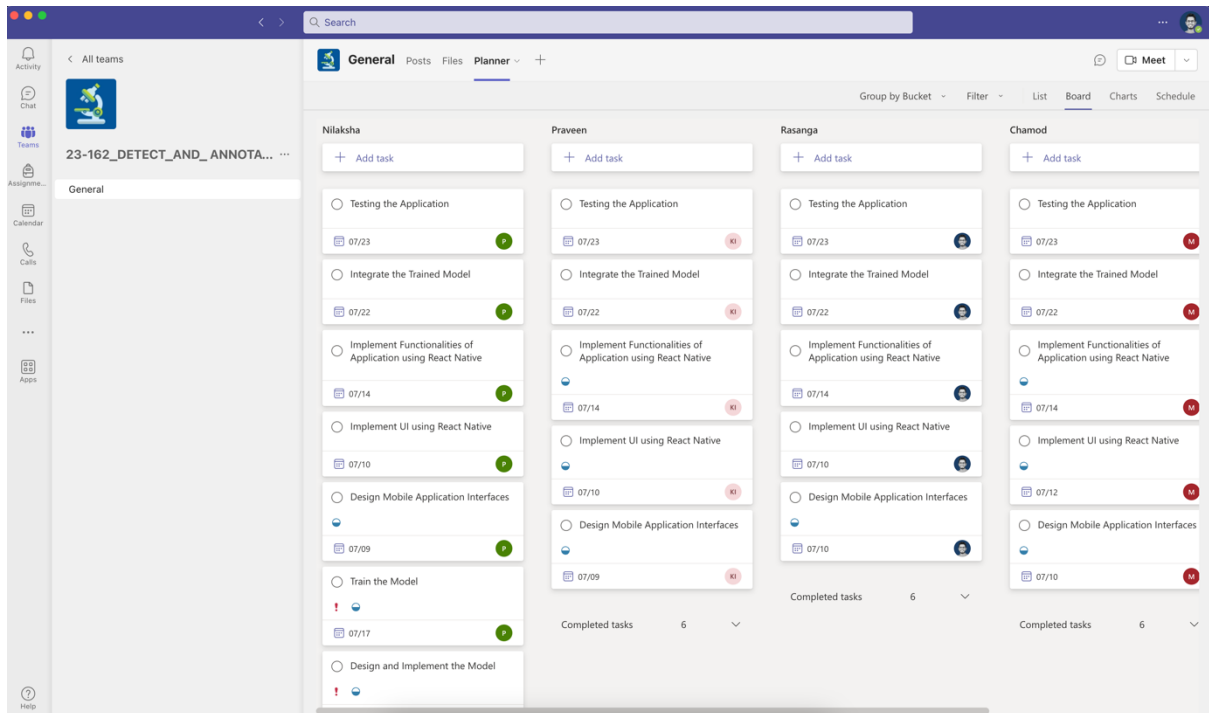


Figure 7: Planner - board view

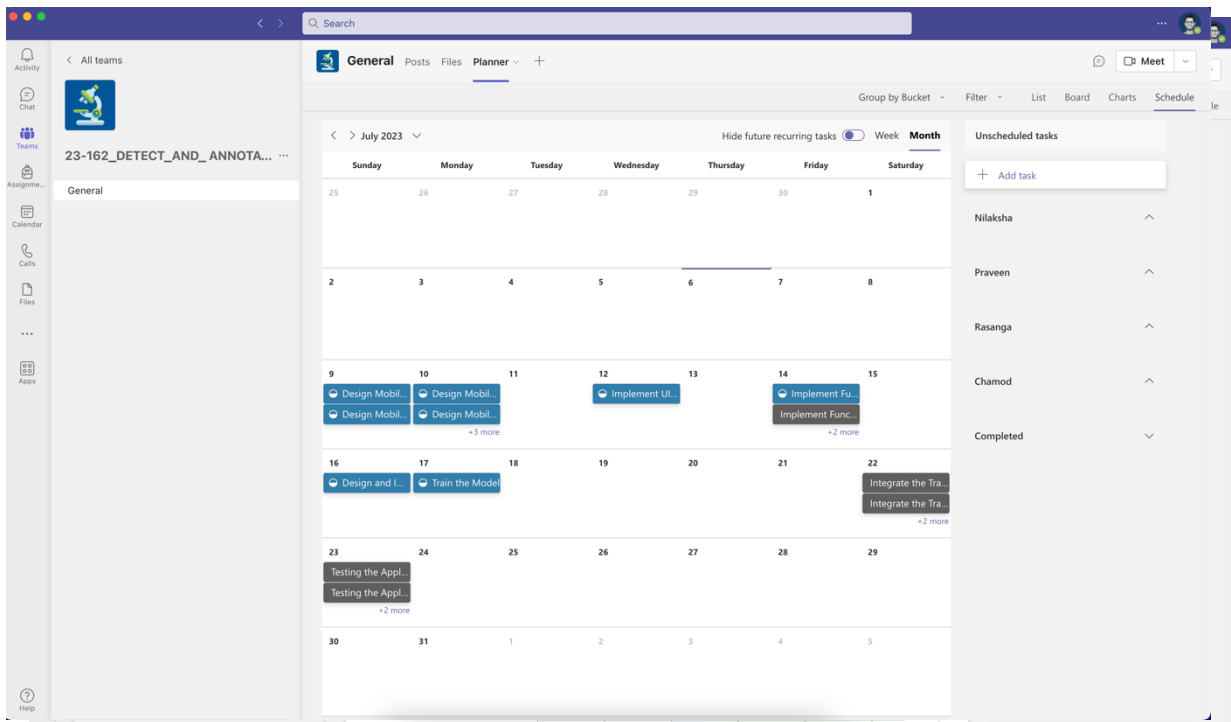


Figure 9: Planner - schedule view

3. Gantt chart

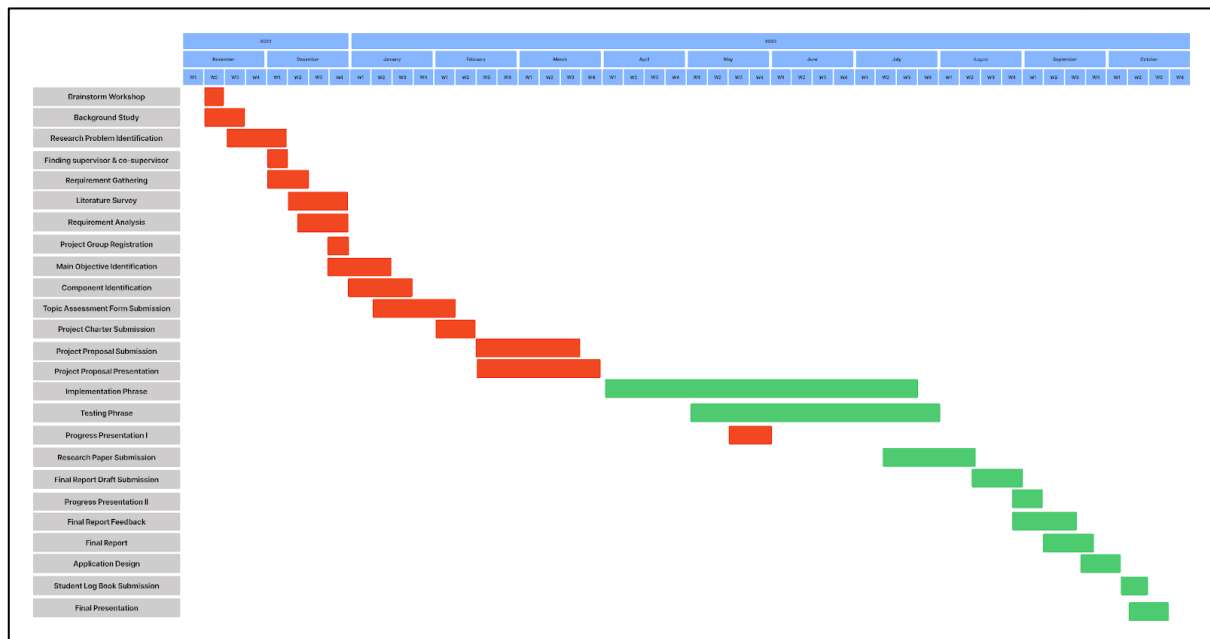
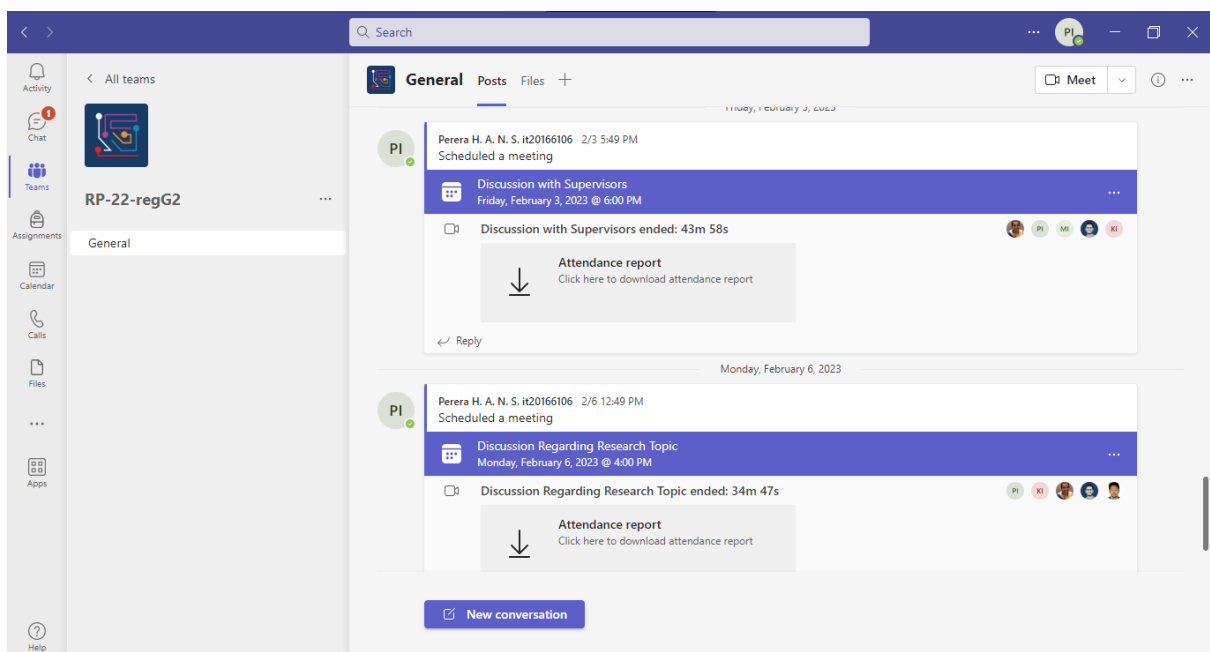
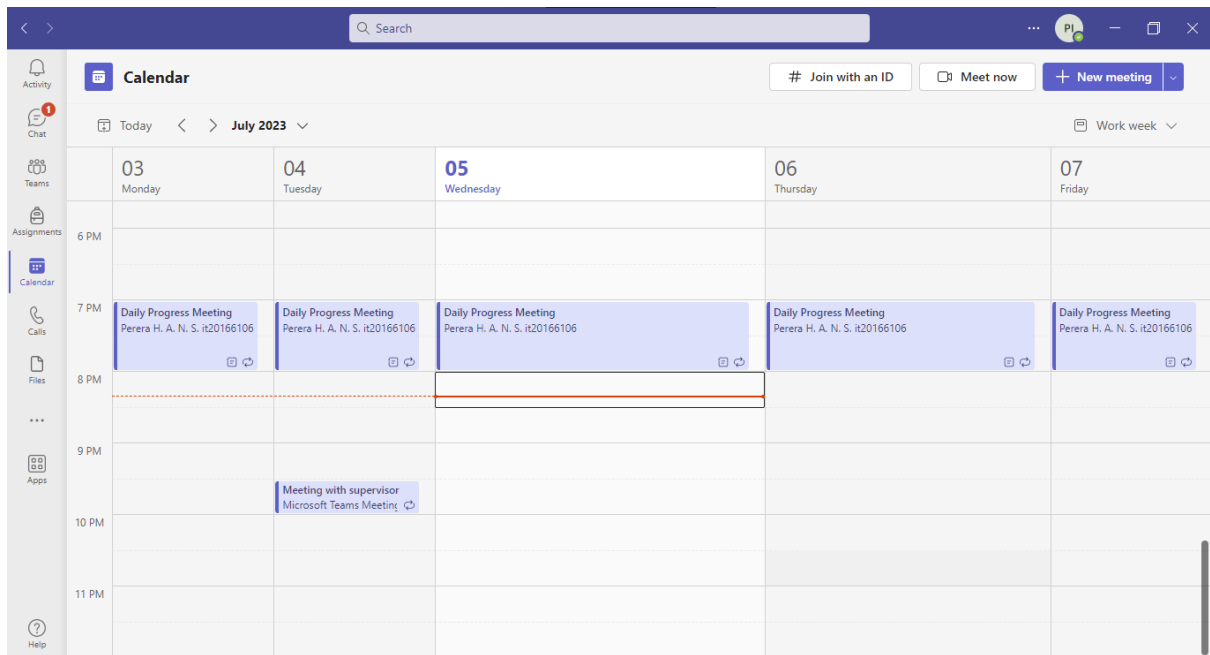


Figure 10: Gantt chart

4. Screenshots of conversations and calls – MS Teams





The screenshot shows a Microsoft Teams meeting window. The top bar includes a "Request control" button, a "Pop out" button, and icons for People, Chat, Reactions, More, Camera, Mic, and Share. The main area displays a whiteboard with a handwritten diagram. The diagram is a flowchart starting with "eye disease" at the top, which branches into "Human Computer" and "Fundus image". "Human Computer" leads to "Computer vision", which then leads to "Cataract Conjunctivitis". "Fundus image" leads to "Glaucoma", which then leads to "Dry eye solution" and "Blink detect". The bottom of the window shows a taskbar with various application icons, a system clock showing 9:29 PM, and a language indicator for "ENG INTL".