National College of Ireland

BSHC4

Software Development

2021-2022

Filip Munteanu

x18359843

EZMechanic

Technical Report

Contents

Executive Summary 2

1.0 Introduction 3

1.1. Background 3

1.2. Aims 3

1.3. Technology 3

1.4. Structure 4

2.0 System 4

2.1. Requirements 4

2.1.1. Functional Requirements 4

2.1.1.1. Requirement 1 Scanning of car part 4

2.1.1.2. Description & Priority 4

2.1.1.3. Use Case 4

2.3.1.1. Requirement 2 Register users in database 6

2.3.1.2. Description & Priority 6

2.3.1.3. Use Case 6

2.3.1.4. Requirement 3 Login in users to the app 8

2.3.1.5. Description & Priority 8

2.3.1.6. Use Case 8

2.3.1.7. Requirement 4 View profile details of logged in user 10

2.3.1.8. Description & Priority 10

2.3.1.9. Use Case 10

2.3.1.10. Requirement 5 Update details in settings 12

2.3.1.11. Description & Priority 12

2.3.1.12. Use Case 12

2.3.2. Data Requirements 14

2.3.3. User Requirements 14

2.3.4. Usability Requirements 14

2.4. Design & Architecture 14

2.5. Implementation 15

2.6. Graphical User Interface (GUI) 24

2.7. Testing 33

2.8. Evaluation 34

3.0 Conclusions 34

4.0 Further Development or Research 34

5.0 References 34

6.0 Appendices 35
6.1. Project Proposal 35
7.0 Objectives 37
Core/Main objective 37
Side objective: Registers users 37
Side objective: Display users details 37
Side objective: Update users details 37
Extra objective: Book a mechanic and leave a rating ..... **Error! Bookmark not defined.**
Storage of data 38
Machine learning 38
8.0 Background 38
9.0 State of the Art 38
10.0 Technical Approach 38
11.0 Technical Details 39
12.0 Special Resources Required 39
13.0 Project Plan 39
14.0 Testing 41
Bibliography 42
14.1. Reflective Journals 42
14.2. Other materials used 47

Executive Summary

The report will provide a general overview of the aim of EZMechanic , it will explain the necessary functional requirements that application needs to have and also some non-functional requirements. It will look at the implementation and the user interface.

The core focus of the project is to use machine learning to be able to detect what a car part is from the image that a user will upload.

Other focuses will be the creation and storage of user credentials to gain access to the application .

Once the object is scanned, the application will update automatically to inform the user what the car part is and also display the location of the object with bounding boxes.

The conclusion of the project is that app is simple to use, and users can save time searching as to what a part may be by using the application . The application may not be able to detect between very similar parts. It can detect between substantially different parts of the car.

1.0 Introduction

1.1. Background

I chose to undertake this project because of my interest in cars and how they work . I have good knowledge about the parts of a car and the purpose of each part. I could Identify the different parts of a car if they were shown to me in a picture. However, I also know that most people do not have this knowledge and only know how to drive the car and may not know how the engine, for example, works. With the ever-growing increase in autonomous vehicles being designed and developed , it is clear that machine learning will begin to play a significant role in the automotive industry . Machines already play a big role in building cars in factories, so it is not long before machine learning begins to root itself deep in the use of cars and the activities surrounding it. Due to the increasing production of cars , the demand for safety has also been highlighted. Car defect detection used to be manual work carried out by experienced factory workers not so long ago . Following on from this we begun moving away from the human eye and using machines to detect defective car parts. Based on a research paper , machine learning has already begun to be experimented with in detecting defective car parts. A study had shown that a deep learning model could have an accuracy of around 95% of detecting defective parts (Liqun, Jiansheng and Dingjin, 2020). It is clear that machine learning will play a big part in detecting vehicle parts.

1.2. Aims

The project aims to utilise machine learning to use object detection to detect different car parts when a picture is uploaded or taken on a mobile Android application .

1.3. Technology

For the creation of my custom object detection model , I have decided to use TensorFlow. This is an open-source library used for machine learning as well as artificial intelligence and is developed by Google. It has many applications but puts particular emphasis on training deep neural networks. Essentially a neural network is a collection of algorithms that can recognize correlations/relationships between sets of data similar to the way the human brain operates hence the name neural. EZMechanic is a mobile application and so Android Studio will be used as the IDE. The logic for the android application is coded in Java and XML is used for the user interface. For the storage of the users data , Firebase Realtime database will be utilised to store the user login credentials which grant them access to the app. This allows for seamless real time updates as users will create an account and then proceed to login.

TensorFlow module is imported into the script to begin training.

Python is used for the creation and training of the machine learning model and then after the model is trained and evaluated for accuracy , it is then exported to a .TFLite format which is used for mobile apps. For the labelling of the machine learning data, I have used LabelImg , which is a free open-source labelling tool that is written in Python (Joseph Nelson, 2020).

1.4. Structure

Section 2 will outline the system , data , user , usability requirements,a visual approach to the system's architecture will be used.An explanation of some user interface pages will be provided, and the code relating to particular features will be shown. The rest of section 2 will look at testing and evaluation. Section 3 will be a conclusion . Section 4 will be possible further development of EZMechanic . Section 5 and 6 will be references and appendices.

2.0 System

2.1. Requirements

2.1.1. Functional Requirements

2.1.1.1. Requirement 1 Scanning of car part

2.1.1.2. Description & Priority

This use case is the user uploading a picture from their phone to be used to for the machine learning model to detect the car part object being displayed and update the picture to what the part may be. This is of the highest importance as it is the core element of the project

2.1.1.3. Use Case

ID : scanImage

Scope

The scope of the use case is to scan the image uploaded by the user and detect the object.

Description

This use case will describe how the image will be detected by the trained model and present results to the suer.

Use Case Diagram

Flow Description

Precondition

- The phone is turned on
- The application is installed
- The user is logged into the app with their credentials
- The user goes to scan section to upload a picture
- The user has granted camera/gallery access permissions
- The user has a camera to take pictures
- The user has images in their gallery of car parts

Activation

The use case starts when the user goes to the scan section

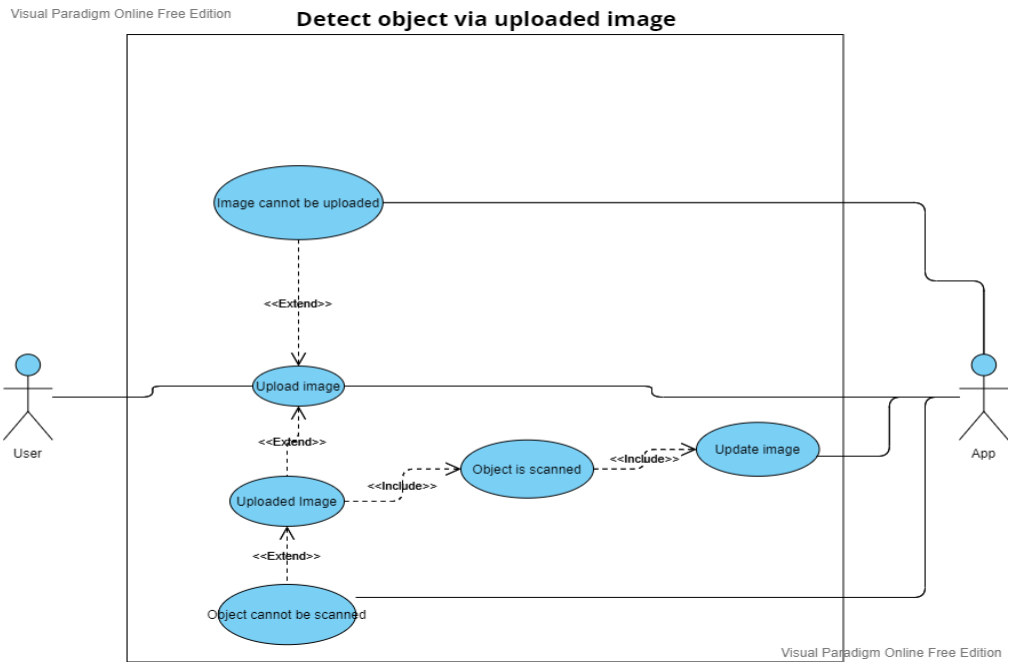
Main flow

1. The user is at the scan section of the app
2. The <User> uploads an image (See A1)
3. The system informs the user image is uploaded successfully .
4. The <User> clicks to scan the image (See A2)
5. The system displays the new image with the object detected with bounding boxes surrounding the object. (See E1)

Alternate flow

A1 : <Image cannot be uploaded>

1. The system displays an error to inform the user that the image cannot be uploaded.



2. The <User > then tries again
3. The use case continues at position 2 of the main flow.

A2 : <Object cannot be scanned>

1.The system displays an error to inform the user that the object cannot be detected by the model.

1. The system then clears the image
2.The use case continues at position 2 of the main flow.

Exceptional flow

E1 : <Object scanned shows wrong car parts >

1The system displays the wrong part detected

2. The <User> exits the app and logs back in
3. The use case continues at position 1 of the main flow

Termination

The system presents the detected object.

Post condition

The system goes into a waiting state for a user response.

2.3.1.1. Requirement 2 Register users in database

2.3.1.2. Description & Priority

This use case is the user registering their information to be able to gain access to the app via the login screen . Once registered they will be able to access the app . This is of second highest importance as without registration a user will not be allowed to access the app and therefore won't be able to use the core functionality .

2.3.1.3. Use Case

ID : registerUser

Scope

The scope of the use case is to register a user to the Firebase Real time database.

Description

This use case will describe how the user will register their details to Firebase Real time database and indicate if it was successful or not.

Use Case Diagram

Flow Description

Precondition

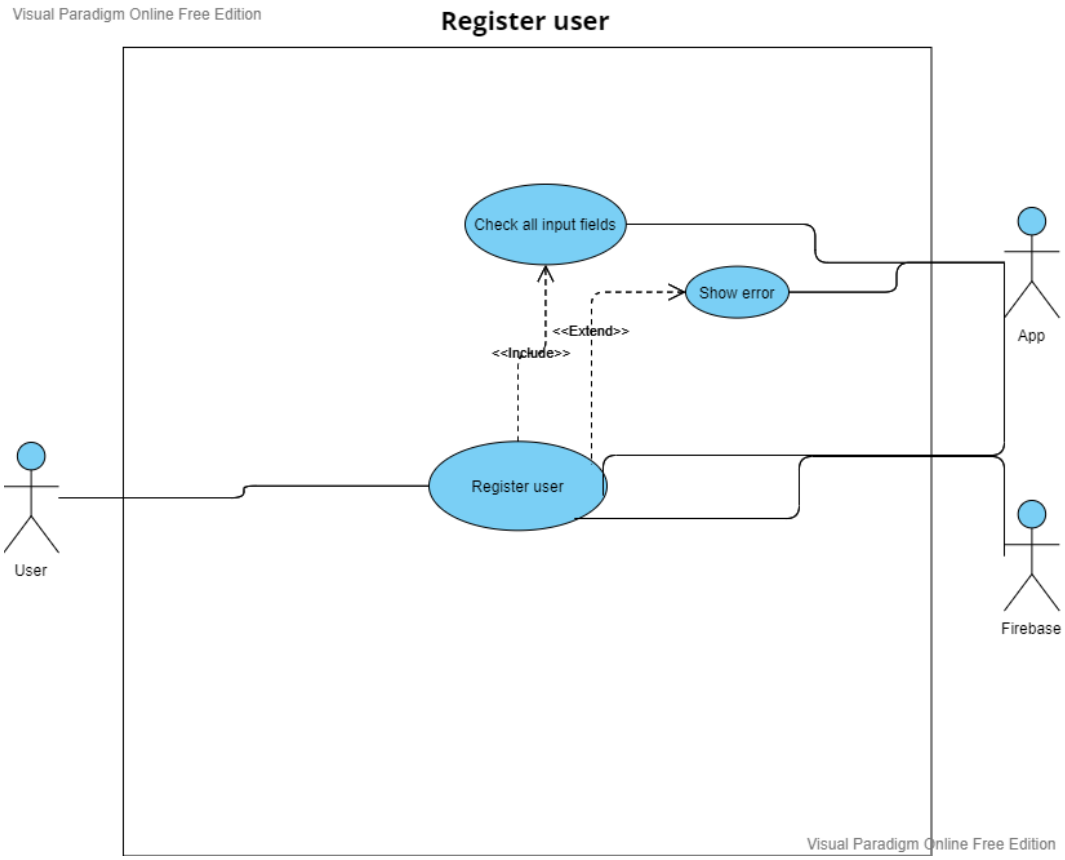
- The phone is turned on
- The application is installed
- The user opened the Register page by clicking the Register button on the login page

Activation

The use case starts when the user goes to the register page

Main flow

1. The <User> enters all the details required in the input fields (See A1)
2. The system informs the user that they have registered successfully .



3. The database is updated with the users information . (See E1)
4. The system brings the user to Profile page in the app.(See A2)

Alternate flow

A1 : <Details are not entered or entered incorrectly>

1. The system displays an error if details are entered wrong or if a particular field is empty .
2. The use case continues at position 1 of the main flow.

A2 : <System does not bring user to the Profile page of the app>

1.The system does not bring the user to the profile page and is left on the register page.

2.The use case ends.

Exceptional flow

E1 : <User information has not been stored in the database>

1. The database does not display the new user information
2. The <User> has to enter details again .
3. The use case continues at position 1 of the main flow.

Termination

The system has successfully registered the user to the Firebase Realtime database.

Post condition

The system goes to the Profile section of the application.

2.3.1.4. Requirement 3 Login in users to the app

2.3.1.5. Description & Priority

This use case is the user logging in with their information that they have registered . Once logged in they will be able to access the app . This is of equal importance as registration .

2.3.1.6. Use Case

ID : logIn

Scope

The scope of the use case is to login a user using their credentials.

Description

This use case will describe how the user will login with their details from Firebase Real time database and indicate if it was successful or not.

Use Case Diagram

Flow Description

Precondition

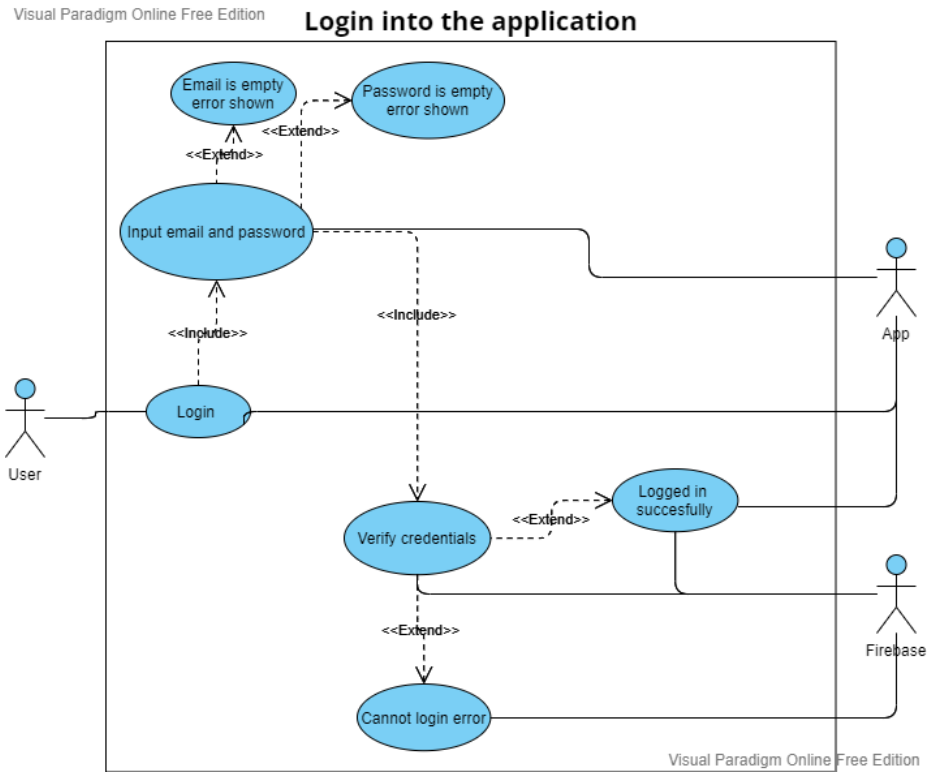
- The phone is turned on
- The application is installed
- The user has registered their credentials to Firebase Realtime database
- The user is not visually impaired and can type using the keyboard

Activation

The use case starts when the user is on the login page

Main flow

1. The <User> enters their credentials (See A1)



2. The credentials are verified by Firebase . (See A2)
3. The system brings the user to Profile page in the app.(See A2 and E1)
4. The system informs the user they have logged in successfully.

Alternate flow

A1 : <Details are not entered or entered incorrectly>

1. The app informs the user if the email is of an incorrect format or if the email or password is empty.
2. The use case continues at position 1 of the main flow.

A2 : <Details cannot be verified>

1.The system informs the user that they cannot log in.

2.The use case continues at position 1 of the main flow.

Exceptional flow

E1 : <User credentials do not exist but the user gains access regardless>

1. The <User> has gained access to application .
2. The use case ends.

Termination

The system has successfully logged in the user using their credentials that are stored in Firebase Realtime database.

Post condition

The system goes to the Profile section of the application.

2.3.1.7. Requirement 4 View profile details of logged in user

2.3.1.8. Description & Priority

This use case is the user clicking on the Profile button to view their details. Once logged in the users details are displayed in the Profile section. This is of slightly lower importance than registering and logging in.

2.3.1.9. Use Case

ID : viewProfile

Scope

The scope of the use case is to have a user's details displayed on the profile page.

Description

This use case will describe how the user will have their details displayed.

Use Case Diagram

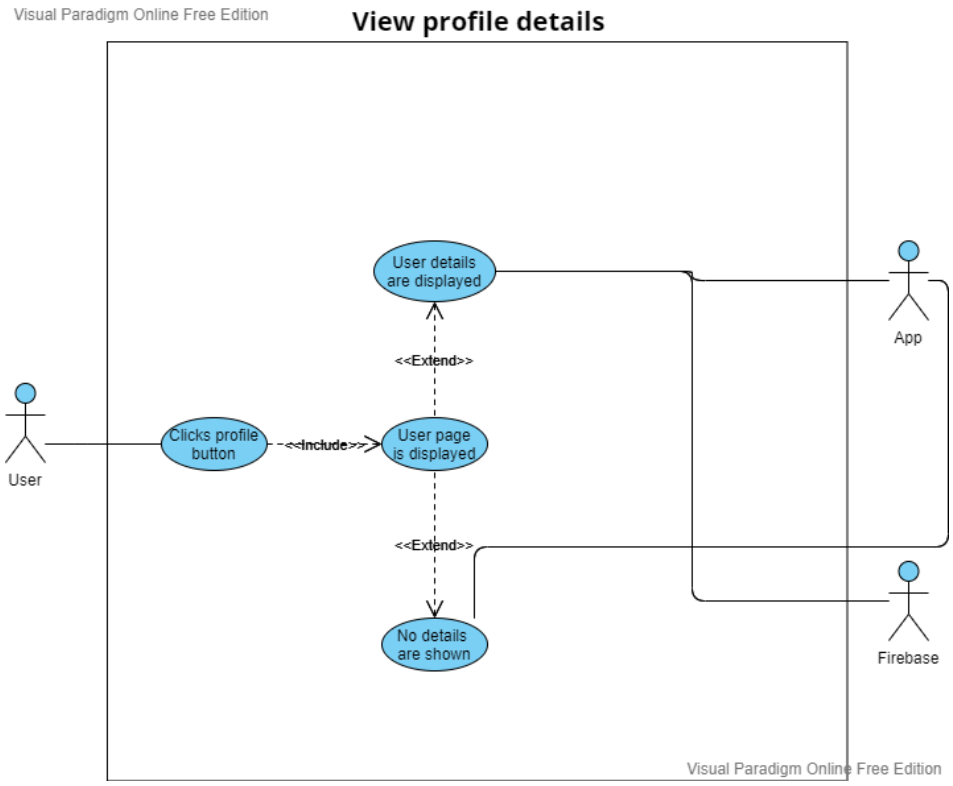
Flow Description

Precondition

- The phone is turned on
- The application is installed
- The user has registered their credentials to Firebase Realtime database
- The user has successfully logged in to the application

Activation

The use case starts when the user is on the profile page



Main flow

1. The <User> clicks on the Profile button .
2. The system brings the user to Profile page in the app .(See A1)
3. The system displays the users credentials that is stored in Firebase Realtime database. (See E1)

Alternate flow

A1 : <Profile page does not display>

1. The use case continues at position 1 of the main flow.

Exceptional flow

E1 : <User credentials cannot be displayed from the database>

1. The use case ends.

Termination

The system has successfully displayed the users credentials from Firebase.

Post condition

The system goes into a waiting stage.

2.3.1.10. Requirement 5 Update details in settings

2.3.1.11. Description & Priority

This use case is the user clicking on the Settings button to update their details. Once logged in the users should be able to change their password. This is of equal importance as displaying the users details.

2.3.1.12. Use Case

ID : updateDetails

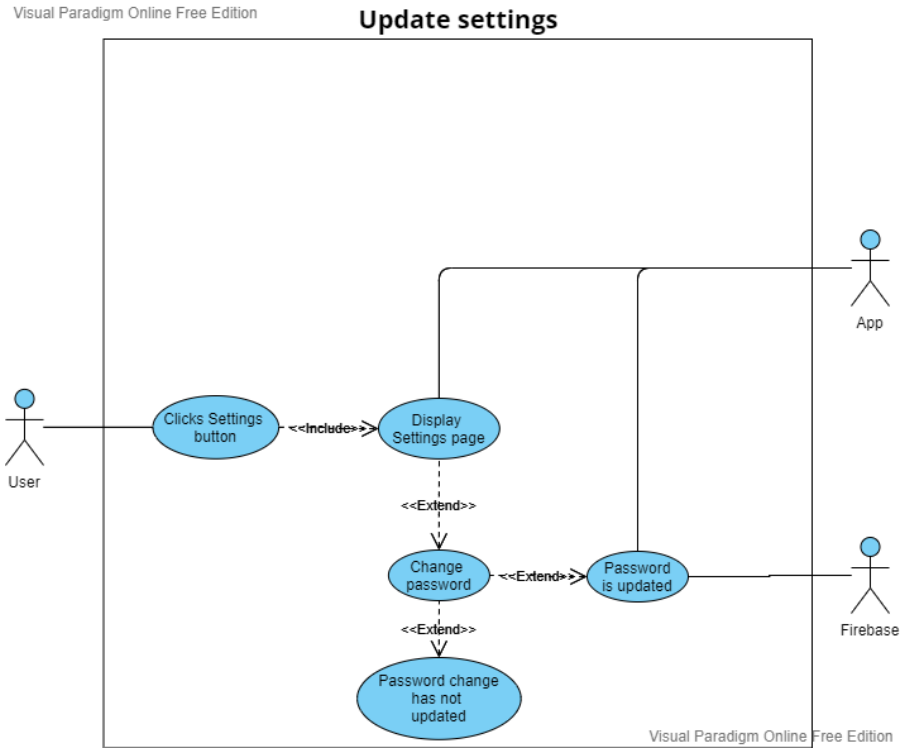
Scope

The scope of the use case is to update a user's details on the settings page.

Description

This use case will describe how the user will update their password.

Use Case Diagram



Flow Description

Precondition

- The phone is turned on
- The application is installed
- The user has registered their credentials to Firebase Realtime database
- The user has successfully logged in to the application

Activation

The use case starts when the user is on the settings page

Main flow

1. The <User> clicks on the settings page. (See A1)
2. The <User> clicks to reset password button. (See A2)
3. The system updates the password in Firebase Realtime database. (See E1)
4. The system informs the user that the password has updated.

Alternate flow

A1 : <Settings page is not displayed>

1. The use case continues at position 1 of the main flow.

A2 : <Update password popup display does not display >

1. The use case ends.

Exceptional flow

E1 : <User password is not updated>

1.The use case ends.

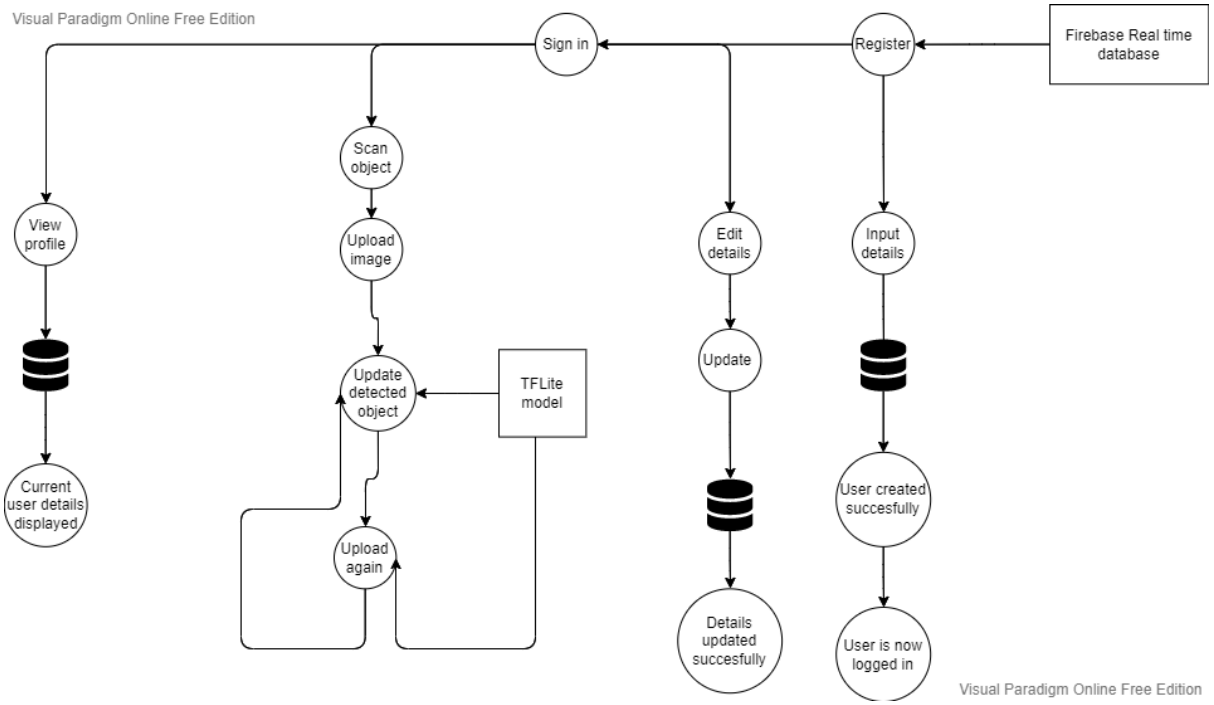
Termination

The system has successfully updated the users password in Firebase .

Post condition

The system goes into a waiting stage.

2.3.2. Data Requirements



The user data that is being registered must be securely sent and stored in Firebase . The data must also be sent and updated in Realtime .

Passwords that users will create when registering must be over 6 characters with a minimum of an uppercase letter, a number, and a special character . Password validation will be in place.

An email needs to consist of an email pattern , i.e., letters with or without numbers followed by a @ symbol , a domain and . to close the email. Emails that do not follow this pattern will not be allowed.

The users data must be retrieved securely and displayed in real time as the user logs into the application. Other users data must not be displayed .

The user must be able to update their details and update this in real time.

2.3.3. User Requirements

A user needs to have basic skills of using a mobile and applications . Users must be familiar with taking pictures on their camera . A user must be familiar with the process of registering their details and logging into an application.

2.3.4. Usability Requirements

User needs to be able to type on a keyboard to input their credentials when logging in.

User needs to allow access to camera and gallery permissions.

2.4. Design & Architecture

Figure 1

Figure 1 is the system architecture overview of how the complete application is integrated with the model.

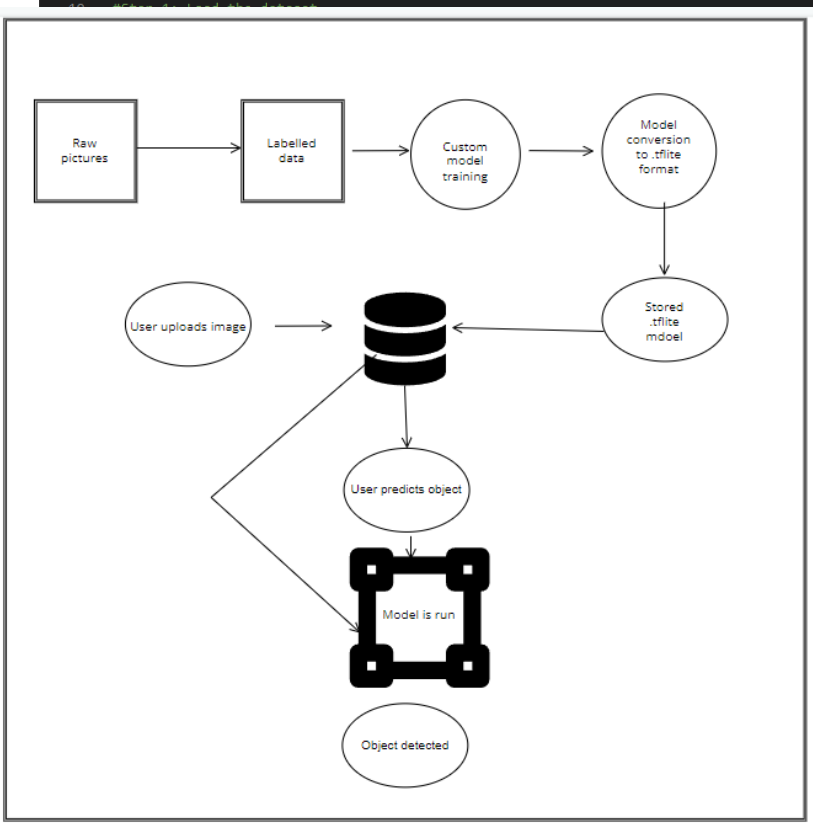
Figure 2

Figure 2 is the system architecture of how the workflow of the model is within the application.

2.5. Implementation

Script.py is the script which I have wrote for the creation of the TensorFlow lite model. I

```
#Import the required packages.

import numpy as np
import os

from tflite_model_maker.config import ExportFormat, QuantizationConfig
from tflite_model_maker import model_spec
from tflite_model_maker import object_detector

from tflite_support import metadata

import tensorflow as tf
assert tf.__version__.startswith('2')

tf.get_logger().setLevel('ERROR')
from absl import logging
logging.set_verbosity(logging.ERROR)
```




```
at it hasn't seen before.

model=True, epochs=20, validation_data=val_data)
```

created and trained a custom object model with a python library from TensorFlow called Model maker (tensorflow.org, 2021). Using this library, I have been able to simplify the process of creating a custom model. The library uses a technique which is referred to as

Model architecture	Size(MB)*	Latency(ms)**	Average Precision***
EfficientDet-Lite0	4.4	146	25.69%
EfficientDet-Lite1	5.8	259	30.55%
EfficientDet-Lite2	7.2	396	33.97%
EfficientDet-Lite3	11.4	716	37.70%
EfficientDet-Lite4	19.9	1886	41.96%

transfer learning which allows the training time to be reduced and also the volume of data needed (Brownlee, 2017).

Figure 3 represents the script.py code used for the creating of the model.

Line 23-26 is where the training data is declared to be used. There are 4 objects that will be detected as part of my model , these are turbo, piston, wheel, mirror. You can also see it loads in data in Pascal VOC format . Pascal VOC unlike other labelling formats is contained in an XML File and a file is created for every image that is used in the training dataset. It has a couple of attributes to it such as name , truncated , difficult and bounding box (Khandelwal, 2019).

Line 29-32 is used for the model to evaluate how accurate the model is with new images that have not been used in training.

Figure 4

Line 36 is choosing the model architecture . There are 4 options to choose from and choosing the EfficientDet-Lite4 means better accuracy but a much higher latency time. I decided to choose a balance between accuracy and latency therefore I chose EfficientDet-Lite2 as my architecture for the model (Khanh LeViet, 2021).

Line 39 is finally where the model is actually created and trained with the inputted dataset.

The epochs is set to 20 as the value . It will go through my custom dataset 20 times .

Batch size is set to 4 meaning it will take 4 images at a time to train through and train_whole_model is set to true so has the ability to tine tune my model in an effort to improve the accuracy .

Line 42 the model is evaluated to see how accurate it is.

Finally line 44 I convert the model to a .tflite format to be used in the android application.

Below will be the code from the android application .

```
private void PerformAuth() {
    String firstName = inputName.getText().toString();
    String secondName = inputSName.getText().toString();
    String email = inputEmail.getText().toString().trim();
    String password = inputPassword.getText().toString().trim();
    String confirmPassword = inputConfirmPassword.getText().toString().trim();
    // some form validation

    if(firstName.isEmpty()&&secondName.isEmpty()&&email.isEmpty()&&password.isEmpty()&&confirmPassword.isEmpty()){
        Toast.makeText( context: RegisterActivity.this, text: "All fields cannot be empty!", Toast.LENGTH_SHORT).show();
    }

    if(firstName.isEmpty()){
        inputName.setError("Enter a first name !");
        inputName.requestFocus();
    }
    if(secondName.isEmpty()){
        inputSName.setError("Enter a surname !");
        inputSName.requestFocus();

    }
    if(!email.matches(emailPattern)){
        inputEmail.setError("Enter a proper email!");
        inputEmail.requestFocus();
    }
    else if(password.isEmpty()){
        inputPassword.setError("Password cannot be empty!");
        inputPassword.requestFocus();
    }
    else if(password.length()<6){
        inputPassword.setError("Password cannot be less than 6 characters ");
        inputPassword.requestFocus();
    }
    else if(!password.equals(confirmPassword)){
        inputConfirmPassword.setError("Passwords must match!");
        inputConfirmPassword.requestFocus();
    }
```

Figure 5

Figure 5 is code that is part of “RegisterActivity” . The PerformAuth() method is used for a user to register an account to Firebase. It contains data validation for the records.

```
        mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    User user = new User(firstName, secondName, email);
                    // add the User object just created into our Realtime database under " Users "
                    FirebaseDatabase.getInstance().getReference( path: "Users")
                            .child(Objects.requireNonNull(FirebaseAuth.getInstance().getCurrentUser()).getUid())
                            .setValue(user).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                //if task is okay then go to next activity which is login page
                                progressDialog.dismiss();
                                nextActivity();
                                Toast.makeText( context: RegisterActivity.this,  text: "Registered Successfully!", Toast.LENGTH_SHORT).show();
                            } else {
                                progressDialog.dismiss();
                                Toast.makeText( context: RegisterActivity.this,  text: "" + task.getException(), Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                } else if (!task.isSuccessful()) {
                    progressDialog.dismiss();
                    Toast.makeText( context: RegisterActivity.this,  text: "" + task.getException(), Toast.LENGTH_SHORT).show();
                }
            }
        });


    }

} // END OF REGISTER METHOD
```

Figure 6

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //initialise them
    inputEmail = findViewById(R.id.enterEmailTV);
    inputPassword = findViewById(R.id.enterPasswordTV);
    registerBtn = findViewById(R.id.registerBtn);
    loginBtn = findViewById(R.id.loginBtn);
    progressDialog = new ProgressDialog( context: this);
    mAuth = FirebaseAuth.getInstance();
    mUser = mAuth.getCurrentUser();

    loginBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { PerformAuth(); }
        private void PerformAuth() {

            String email = inputEmail.getText().toString();
            String password = inputPassword.getText().toString();
            // some data validation
            if(!email.matches(emailPattern)){
                inputEmail.setError("Enter a proper email!");
                inputEmail.requestFocus();
            }
            else if(email.isEmpty()){
                inputEmail.setError("Email cannot be empty!");
                inputEmail.requestFocus();
            }
            else if(password.isEmpty()){
                inputPassword.setError("Password cannot be empty!");
                inputPassword.requestFocus();
            }
```

Figure 7

In Figure 6, the createUserWithEmailAndPassword method is used to enter the users details into Firebase. It creates a new User object which is POJO class , to be entered into the database.

Figure 7, it is code from the “MainActivity” . This activity is where the user is able to login to application or else create a new account. First it gets an instance of Firebase and an gets the current user of that instance. It checks for data validation before it runs a query to Firebase.

```
else{
    progressDialog.setMessage("Please wait while it's logging in....");
    progressDialog.setTitle("Logging in");
    progressDialog.setCanceledOnTouchOutside(false);
    progressDialog.show();

    mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()){
                FirebaseDatabase.getInstance().getReference( path: "Users")
                        .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                        .addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot snapshot) {
                                progressDialog.dismiss();
                                nextActivity();
                                Toast.makeText( context: MainActivity.this,  text: "Logged in Successfully!", Toast.LENGTH_SHORT).show();
                            }

                            @Override
                            public void onCancelled(@NonNull DatabaseError error) {

                            }
                        });
            } else{
                progressDialog.dismiss();
                Toast.makeText( context: MainActivity.this,  text: ""+task.getException(), Toast.LENGTH_SHORT).show();
            }
        }
    });


}
```

Figure 8

It runs a method called signInWithEmailAndPassword , which checks if the user exists or if the data is incorrect such as the password. It is correct it brings you to the profile activity. Otherwise, it displays the error and stays on the same activity.

Figure 9

```
//get the firebase user who is logged in
authProfile = FirebaseAuth.getInstance();
FirebaseUser firebaseUser = authProfile.getCurrentUser();

//if the firebase user is empty then print an error
if (firebaseUser == null) {
    Toast.makeText( context: ProfileActivity.this,  text: "Error!", Toast.LENGTH_LONG).show();
} else {
    //otherwise show the users profiles
    showUserProfile(firebaseUser);
}
```

Figure 9 is code part of the “ProfileActivity” it gets the current user who is logged into the application from Firebase. It the user is not empty it will run the showUserProfile method.

```
//method to display the currently logged in user
private void showUserProfile(FirebaseUser firebaseUser) {
    //create a string with the userid
    String userID = firebaseUser.getUid();
    //Extract user reference from database for "Users"
    DatabaseReference referenceProfile = FirebaseDatabase.getInstance().getReference( path: "Users");
    referenceProfile.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            User thisUser = snapshot.getValue(User.class);
            if (thisUser != null) {
                //if user is not null , then we grab his details.
                userEmail = firebaseUser.getEmail();
                fName = thisUser.firstName;
                sName = thisUser.secondName;
                // Set the text views based on the data
                firstName.setText(fName);
                secondName.setText(sName);
                email.setText(userEmail);

            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            //print an error
            Toast.makeText( context: ProfileActivity.this,  text: "Something went wrong!", Toast.LENGTH_LONG).show();
        }
    });
}
```

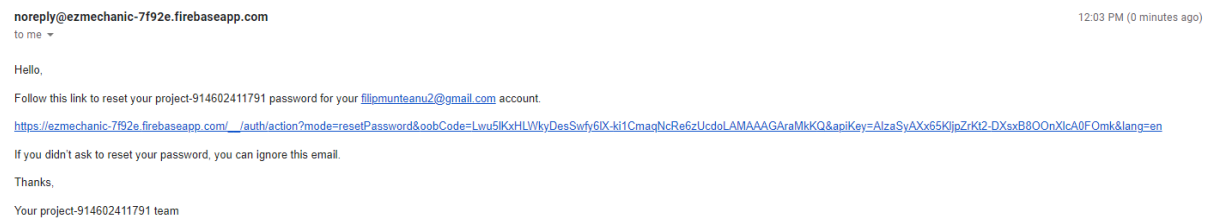
Figure 10

Figure 10 , extracts a specific user based on String userID from Firebase and gets the data from it. If the user is not empty, then it will retrieve his details and append to the information to the text views of the profile view.

```java
//reset password method
private void resetPassword(String email) {
    authProfile = FirebaseAuth.getInstance();
    authProfile.sendPasswordResetEmail(email).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Toast.makeText( context: SettingsActivity.this,  text: "Password reset has been sent to your email", Toast.LENGTH_SHORT).show();
                nextActivity();
            } else {
                try {
                    throw task.getException();
                } catch (FirebaseAuthInvalidUserException e) {
                    inputEmail.setError("User no longer exists or is no longer valid, register again.");
                } catch (Exception e) {
                    Log.e(TAG, e.getMessage());
                    Toast.makeText( context: SettingsActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                }

            }
            progressBar.setVisibility(View.GONE);
        }
    });
}
```

Figure 11

In Figure 11, is the code from the SettingsActivity where a user can reset their password in order to receive a password reset link sent to their email.



noreply@ezmechanic-7f92e.firebaseapp.com 12:03 PM (0 minutes ago)
to me

Hello,

Follow this link to reset your project-914602411791 password for your filipmunteanu2@gmail.com account.

https://ezmechanic-7f92e.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=Lwu5lKxHLWkyDesSwfy6lX-ki1CmaqNcRe6zUcdoLAMAAAGAraMkKQ&apiKey=AIzaSyAXx65KljpZrKt2-DXsxB8OOnXlcA0FOmk&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-914602411791 team

Finally, the core of the application is below.

```java
// method to set the image the user uploaded
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode == 100)
    {
        //set the image view with image data
        imageBox.setImageURI(data.getData());

        Uri uri = data.getData();
        try {
            // create a bitmap to be used later
            img = MediaStore.Images.Media.getBitmap(this.getContentResolver(), uri);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Figure 12

In Figure 12, is where we set the imageBox variable with the image the user has selected.

We also create this image into a bitmap to be used later during the detection process.

In Figure 13 is the method that is called upon to draw the bounding boxes of the objects location within the image.

```
//the method where the bounding boxes are drawn
void drawBoundingBox(Canvas canvas, RectF location) {
    // Draw the rectangle on the canvas of our image
    //set the paint attributes
    Paint boxPaint = new Paint();
    boxPaint.setColor(Color.RED);
    boxPaint.setAlpha(200);
    boxPaint.setStyle(Paint.Style.STROKE);
    canvas.drawRect(location, boxPaint);
}
```

Figure 13

```
86          //Predict the object when you click this button
87          predictBtn.setOnClickListener(new View.OnClickListener() {
88
89              @Override
90              public void onClick(View v) {
91
92                  //if the image is empty then let the user know to upload one
93                  if(img == null){
94                      Toast.makeText( context: CameraActivity.this,  text: "Upload an image first!", Toast.LENGTH_SHORT).show();
95                  } // otherwise run the custom tensorflow lite model based on what the image is
96                  else if(img != null){
97                      //if image is not null scale the bitmap
98                      img = Bitmap.createScaledBitmap(img,  dstWidth: 281,  dstHeight: 324,  filter: true);
```

Figure 14

First, we check if there is an image and if there is we created a scaled bitmap to be used.

```
100                 try {
101                     //Create a new canvas to draw on
102                     Canvas canvas = new Canvas(img);
103                     //create a model to use
104                     Android model = Android.newInstance(getApplicationContext());
105                     // Creates inputs for reference.
106                     TensorImage image = TensorImage.fromBitmap(img);
107                     // Runs model inference and gets result.
108                     Android.Outputs outputs = model.process(image);
109                     Android.DetectionResult detectionResult = outputs.getDetectionResultList().get(0);
110
111                     // Gets result from DetectionResult.
112                     float score = detectionResult.getScoreAsFloat();
113                     RectF location = detectionResult.getLocationAsRectF();
114                     String category = detectionResult.getCategoryAsString();
115                     //draw on the canvas with the given location from the model
116                     drawBoundingBox(canvas, location);
117                     // Releases model resources if no longer used.
118                     model.close();
119                     // here we will print out the results of the object to text views based on the image that is inputted by the user
120                     // we print out object type, accuracy score and location of the object on the image
121                     objecttv.setText(category);
122
123                     scoretv.setText(Float.toString(score));
124                     textBox.setText(newText);
125                     imageBox.setImageBitmap(img);
126                     //let user know we detected an object
127                     Toast.makeText( context: CameraActivity.this,  text: "Object detected! ", Toast.LENGTH_SHORT).show();
128                 } catch (IOException e) {
129                     //print out an error to the user and restart the activity.
130                     Toast.makeText( context: CameraActivity.this,  text: "An error happened " + e, Toast.LENGTH_SHORT).show();
131                     sameActivity();
132                 }
133             } // end of if else
134         }
135     }
136     );
```

Figure 15

Line 102 , I create a clear canvas to be used for drawing based on the image bitmap.

Line 104 , the model is selected to be used.

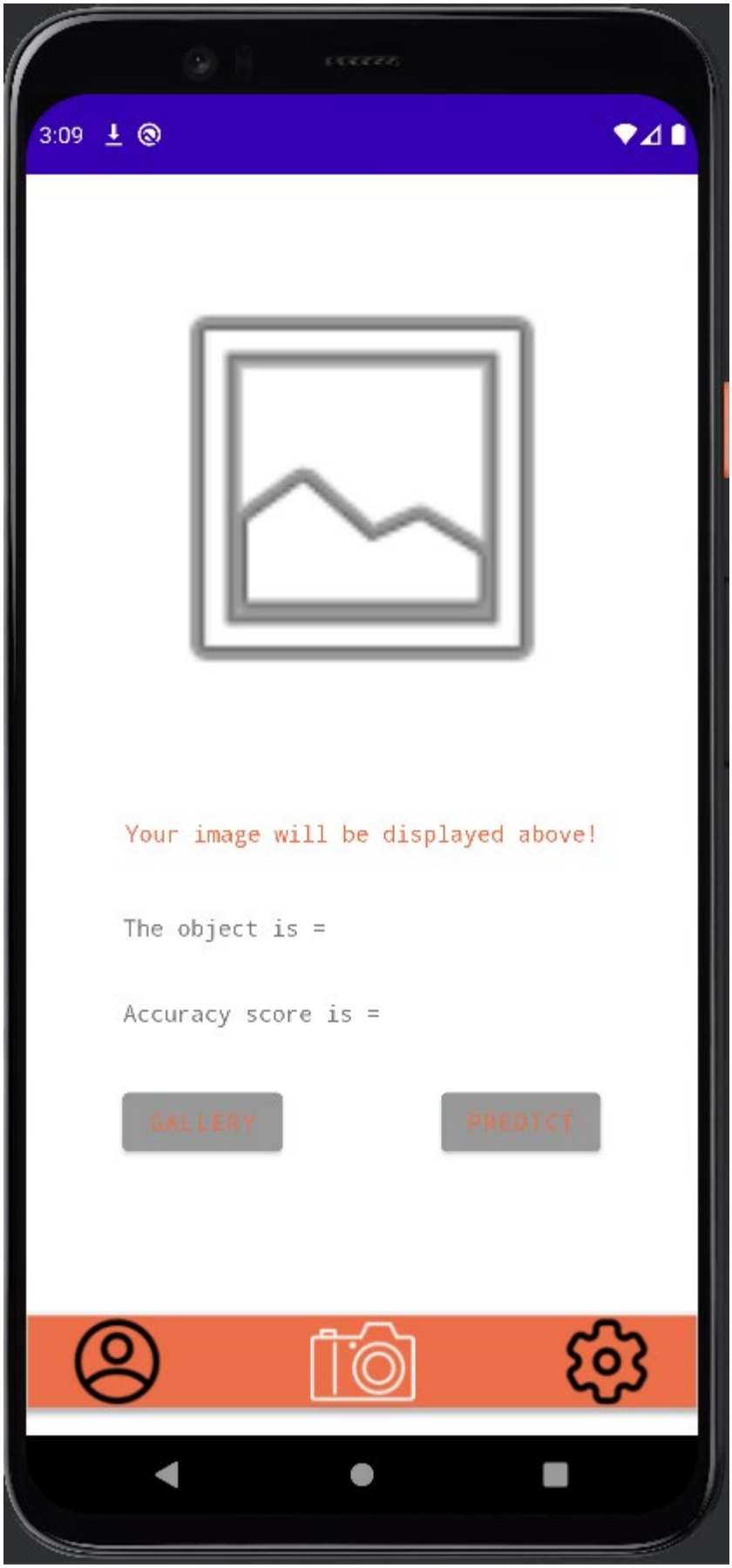
Line 106 and 108 a tensorflow image is created based on the bitmap . The model is then processed with the tensor image.

Line 112-114 results are taken from the detectionResult variable.

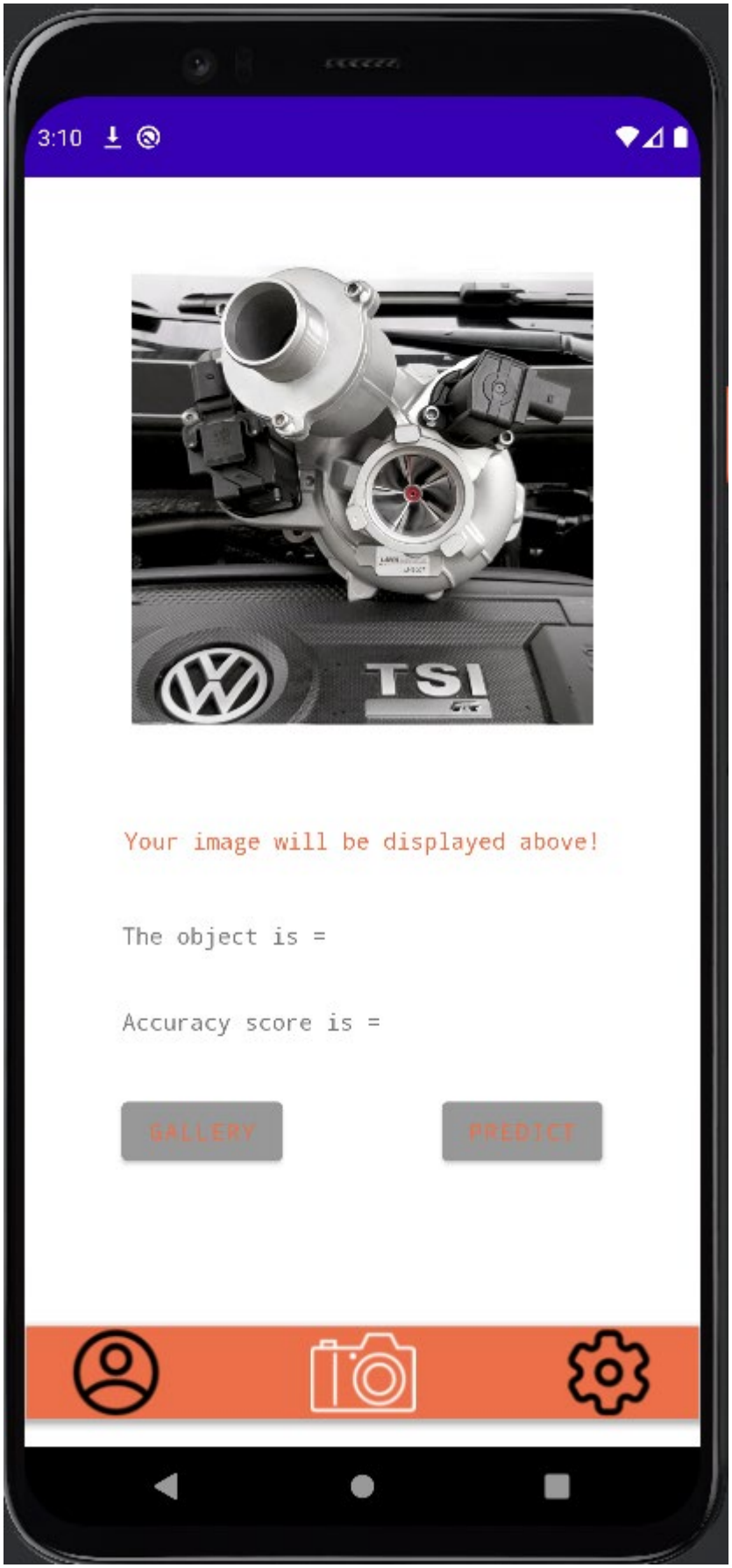
Line 116 the drawBoudingBox method is called to draw on a canvas with the location variable that we received. This allows to a canvas to be drawn with the correct coordinates of where the object is detected within the image.

Line 121-125 the results are displayed back within the application .

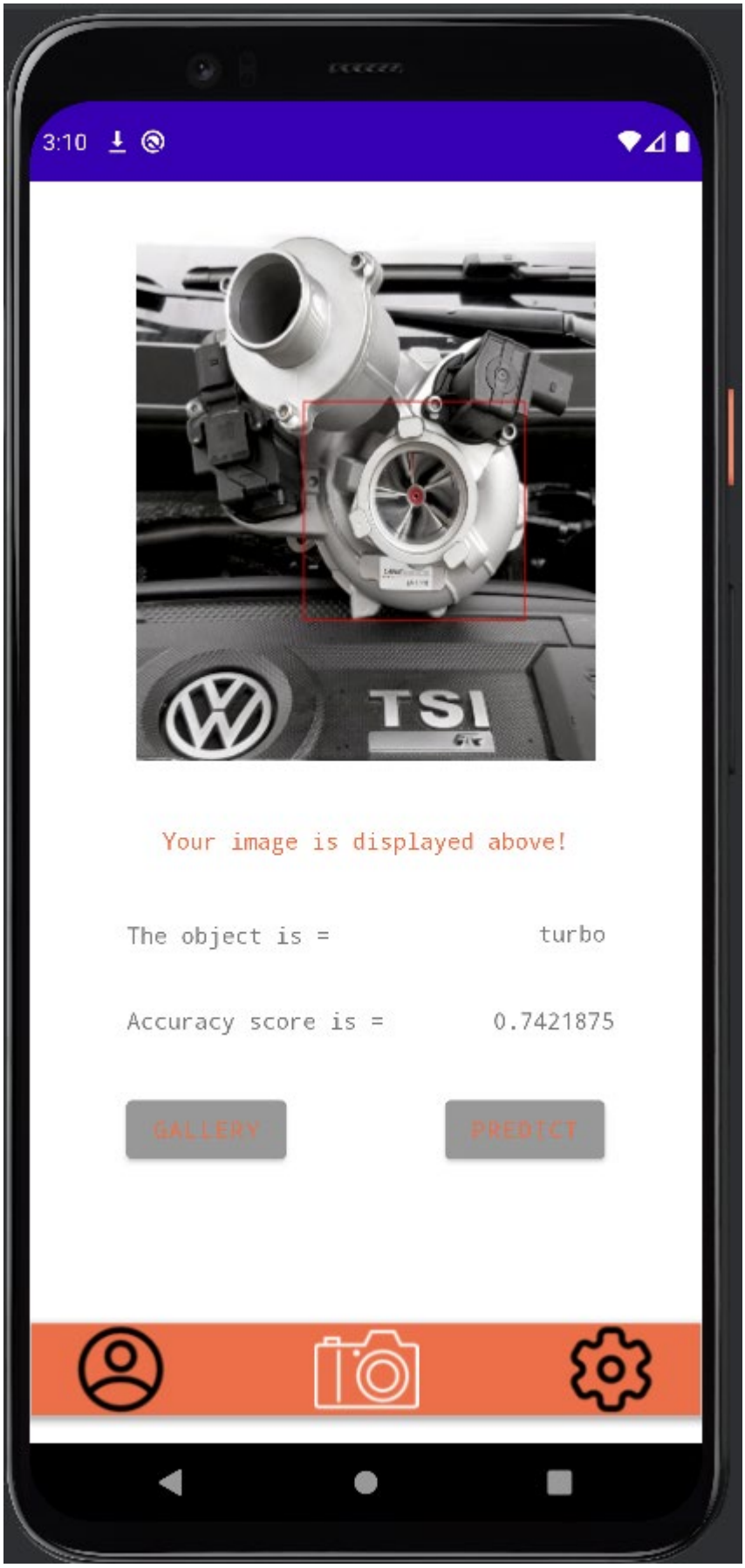
2.6. Graphical User Interface (GUI)



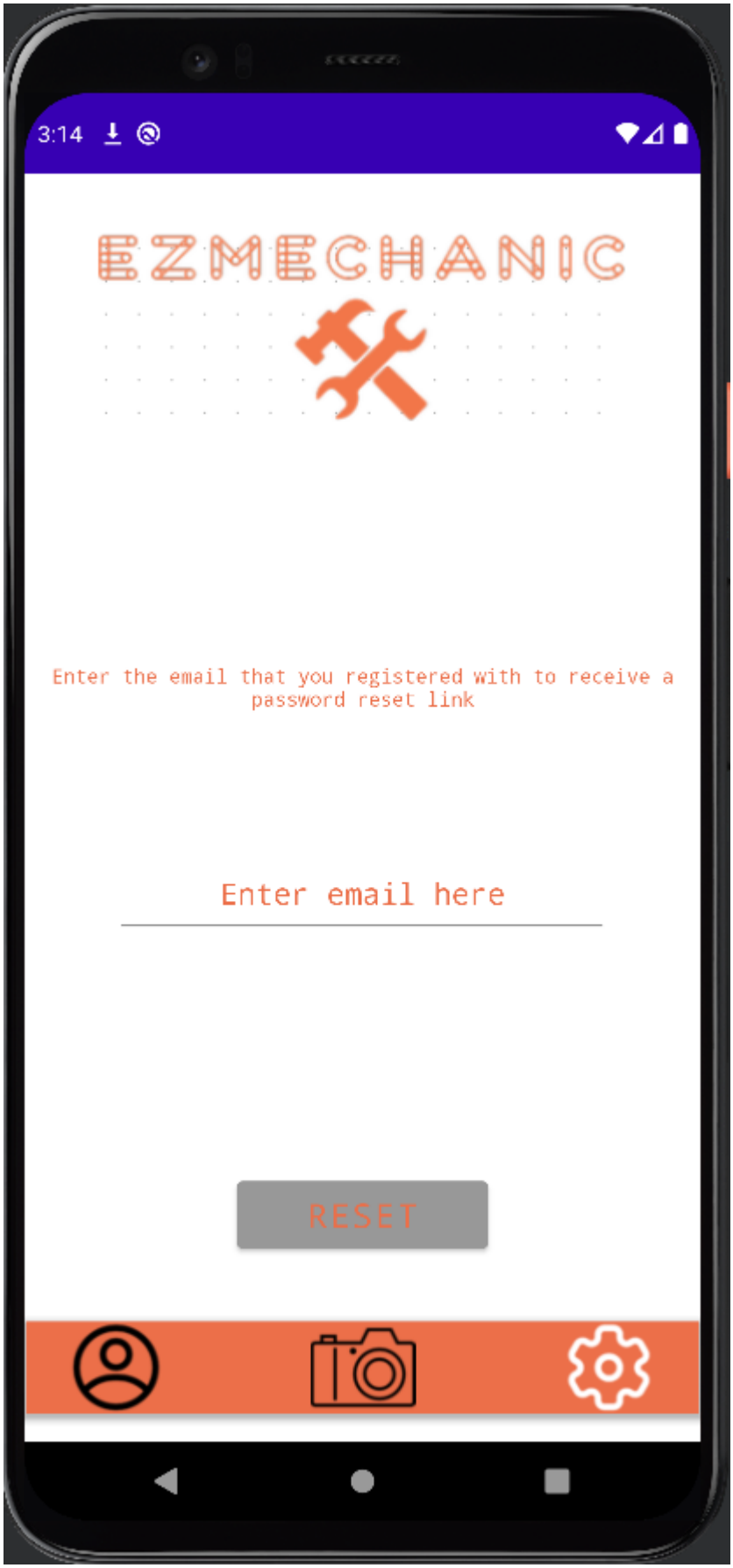
The first image is the CameraActivity where the image box is blank . You can see there is 2 buttons , one to upload an image from the gallery and a predict button.



This is how the view looks when an image is selected from the gallery.

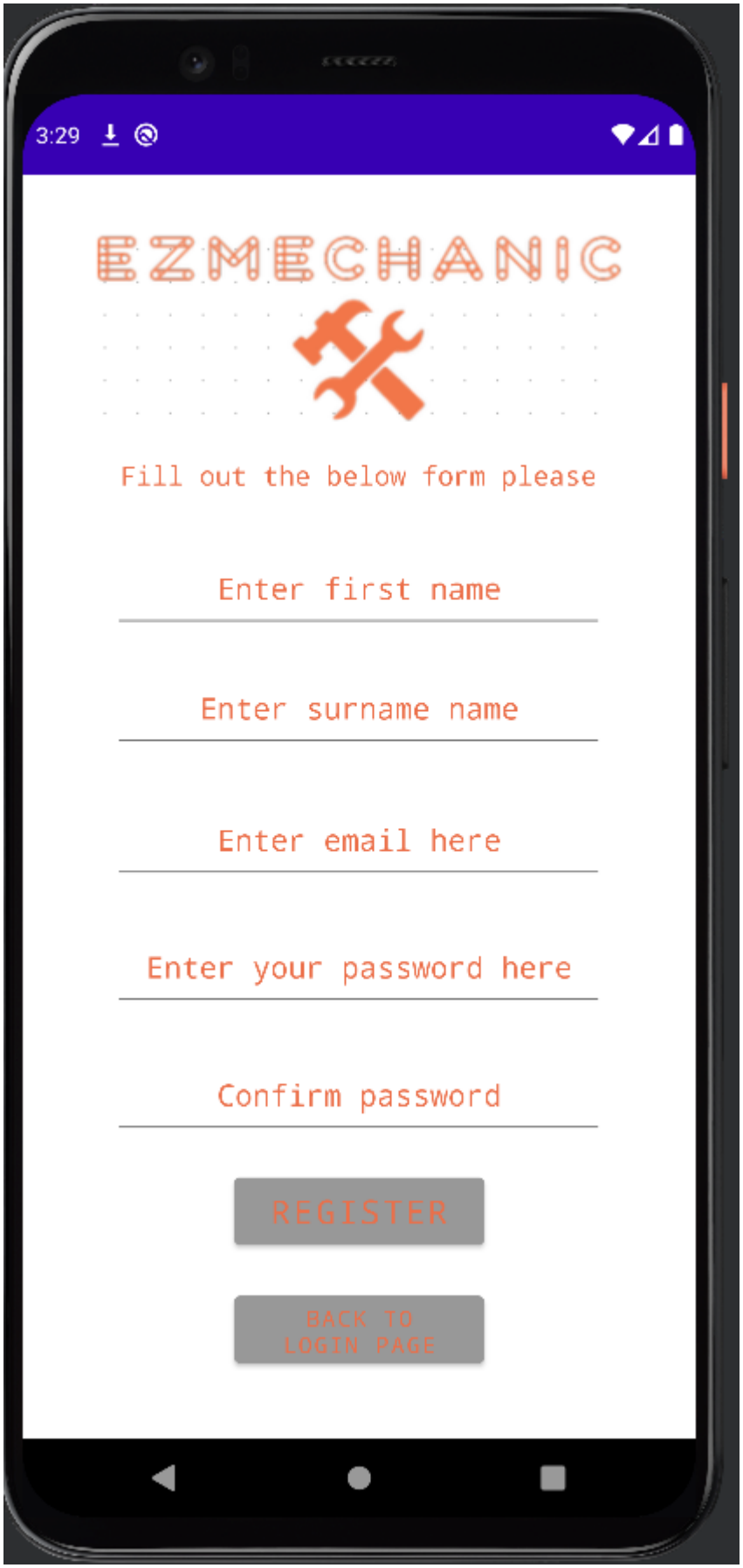


This image is how the view looks after a user uploaded an image, clicked the predict button and received back the predictions. The object being displayed is a turbo and the accuracy to which the object is detected at is around 74% . The drawing of the bounding boxes is also drawn on the image. The drawing is based on the models prediction of the location .

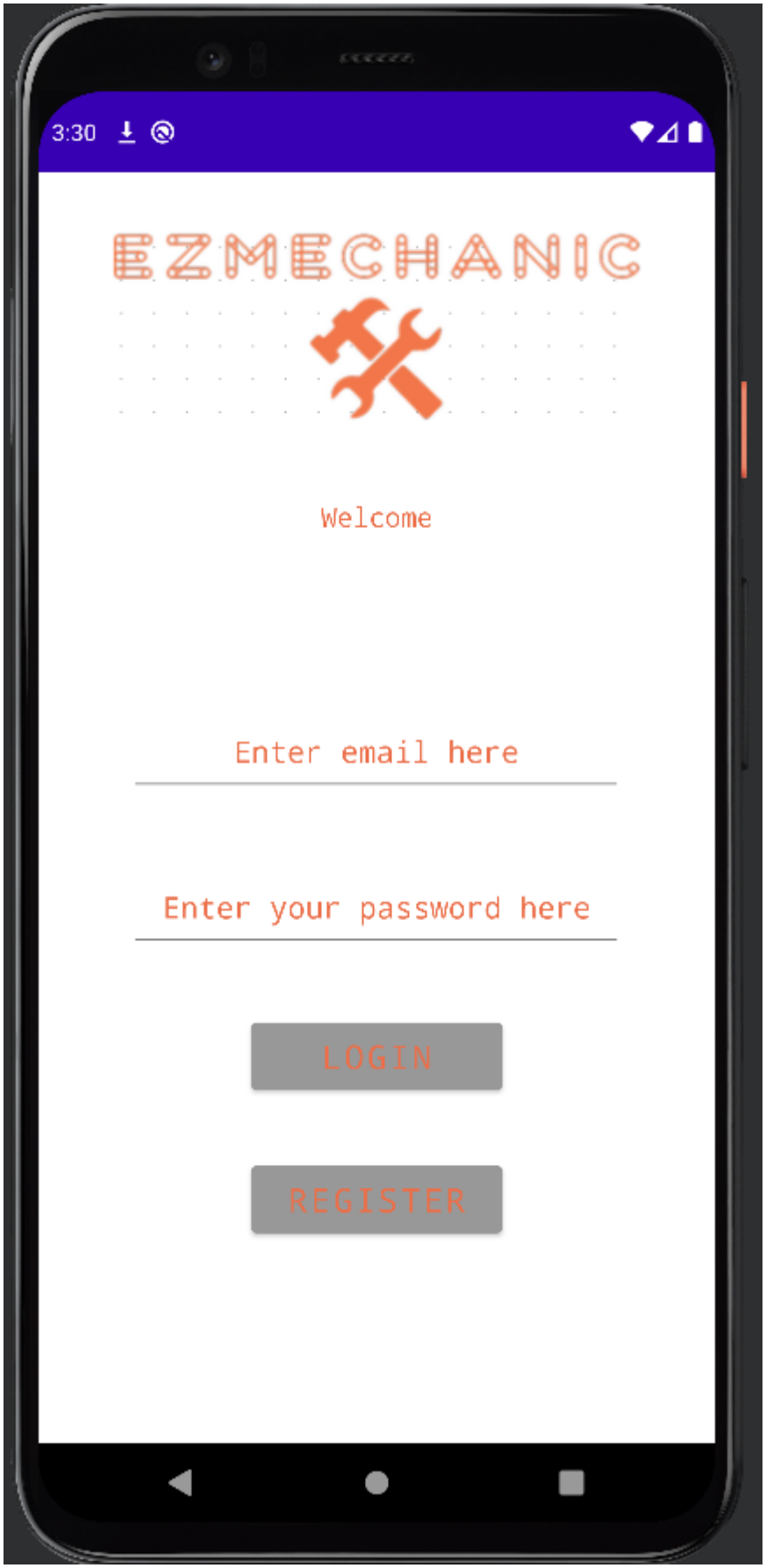

3:14
EZMECHANIC
Enter the email that you registered with to receive a password reset link
Enter email here
RESET

This image is the SettingsActivity and it is where a user can reset their password by receiving a password reset link sent to their email.

This is the ProfileActivity , where the users details are displayed back.


3:29
EZMECHANIC
Fill out the below form please
Enter first name
Enter surname name
Enter email here
Enter your password here
Confirm password
REGISTER
BACK TO LOGIN PAGE

The registration page, which is RegisterActivity , the users can enter their details and register an account to gain access to the application.



The MainActivity which is the login page for users to enter their credentials and sign in.

2.7. Testing

Unit testing:

Scenario 1: Create the user profile

- Can the user successfully create a profile with the details entered?
  If not, the test fails.
- Are the details sent to Firebase and stored correctly?
  If not, the test fails.
- Can the user create another profile with the same details?
  If yes, the test fails.
- Can the user login successfully using Firebase authentication?
  If not, the test fails

Scenario 2: Display user profile

- Are the details retrieved from Firebase and displayed correctly?
  If not, the test fails.

Scenario 3: Camera function

- Can you upload a picture with the camera functionality in the app?
  If not, the test fails.

Scenario 3: Login function

- Can the user successfully login with the correct details?
  If not, the test fails.
- Can the user successfully login with the incorrect details?
  If not, the test passes.

Scenario 3: Reset function

- Can the user successfully reset their password?
  If not, the test fails.
- Can the user receive a password reset link?
  If not, the test passes.

Integration testing:

Scenario 1:

- Does the android app connect successfully to Firebase?
  If not, the test fails.

Scenario 2:

- Does the TensorFlow Lite model detect objects successfully via the camera functionality of the android app?
  If not, the test fails.

Scenario 3:

- Does the app display the correct user details from Firebase in the profile section?
  If not, the test fails.

2.8. Evaluation

The system from an end user's perspective was tested by the developer as the focus was on the machine learning requirement. The developer tested the application to assess the accuracy of the object detection model by inputting new images that the model was not trained with. Data validation was evaluated by registering test accounts and observing if the data would be saved based on the validation requirements.

3.0 Conclusions

The advantage of the project is that a user can quickly scan a car part and determine what it is . They will not be able to receive information about the part and it's real life uses . The app is simple to use and no experience with complex user interfaces is required. This would free up an individual's time instead of guessing what a car part may be and they can find out this information quickly. The project is limited by the accuracy of scanning same parts that may look slightly different than one another.

4.0 Further Development or Research

For the further development of the project more car parts that could be detected would be added . It would also provide real-time detection so there would be no need to take a picture but rather open the camera and point it at a part and receive real time predictions . It would also look to add explanations so that once an object is detected , a user can also receive a formal description of what the car part is and what it's role is.

5.0 References

Brownlee, J. (2017) 'A Gentle Introduction to Transfer Learning for Deep Learning', *Machine Learning Mastery*, 19 December. Available at: https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (Accessed: 10 May 2022).

Joseph Nelson (2020) *LabelImg for computer vision annotation*, *Roboflow Blog*. Available at: https://blog.roboflow.com/labelimg/ (Accessed: 10 May 2022).

Khandelwal, R. (2019) *COCO data format for Object detection*, *Medium*. Available at: https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5 (Accessed: 10 May 2022).

Khanh LeViet (2021) 'Easier object detection on mobile with TensorFlow Lite', *Easier object detection on mobile with TensorFlow Lite*, 16 June. Available at: https://blog.tensorflow.org/2021/06/easier-object-detection-on-mobile-with-tf-lite.html (Accessed: 10 May 2022).

Liqun, W., Jiansheng, W. and Dingjin, W. (2020) 'Research on Vehicle Parts Defect Detection Based on Deep Learning', *Journal of Physics: Conference Series*, 1437, p. 012004. doi:10.1088/1742-6596/1437/1/012004.

tensorflow.org (2021) *TensorFlow Lite Model Maker*, *TensorFlow*. Available at: https://www.tensorflow.org/lite/guide/model_maker (Accessed: 10 May 2022).

6.0 Appendices

6.1. Project Proposal

National College of Ireland

Project Proposal

EZMechanic

November 7th, 2021

BSHC4 Computing

Software Development

Year 2021/2022

Filip Munteanu

x18359843

x18359843@student.ncirl.ie

Contents

1.0 Objectives 37

Core/Main objective 37

Side objective: Bookings for mechanics 37

Storage of data 38

Machine learning 38

2.0 Background 38

3.0 State of the Art 38

4.0 Technical Approach 38

5.0 Technical Details 39

6.0 Special Resources Required 39

7.0 Project Plan 39

8.0 Testing 41

Bibliography 42

7.0 Objectives

Core/Main objective

The application will be a mobile application developed in Android studio.

The goal of this project is to allow users to simply be able to take a picture/scan parts on their car using their phone and then object detection will come into effect to inform the user of the different individual objects. This application currently as it stands will be a free version.

Side objective: Registers users

The users will be able to register an account on the application. They will then be able to login into application .

Side objective: Display users details

The users will be able to view their details they have created in Firebase.

Side objective: Update users details

The users will be able to update their details that are stored in Firebase.

Extra functionality which can be implemented based on progress completion of core objective listed above*

Storage of data

The database storage which I plan to use will be Firebase. It is a Google pushed database and is of NO-SQL type. The database is in real-time which allows smooth creation of accounts and pulling of data for activities such as logging in users that already have an account (Rosencrance, 2019).

Machine learning

The application will make use of a machine learning open-source platform called TensorFlow.

8.0 Background

I chose to undertake this project because I have a passion for cars and would be well versed in the workings of them, but I realise many people are not. Often people's knowledge only goes as far as knowing how to put in fuel and drive. While there is a vast amount of information nowadays on the internet to browse through sometimes this can be very complicated and confusing for an average car user. EZMechanic sets out to help everyday car owners identify particular parts of a car without a complex technical explanation about how the part works. If you suspect your car has an issue within a particular area/part or it's making a weird noise, then you can use EZMechanic. It will help them by utilising machine learning object detection.

I will use TensorFlow which is Google's open-source machine learning platform. I will began to gather training images for the car parts. Afterwards I will need to train an object detection model and export that to a TensorFlow lite format which can be incorporated into Android studio for use.

Once I have the TensorFlow lite format, I need to create the camera functionality in the mobile application in Android studio and integrate the object detection.

9.0 State of the Art

On the marketplace there is few object detection apps, some of these include the more popular Google Lens application which can scan objects such as flowers and do a search based on the image. It can also read and translate foreign text in real time (Anon., 2018).

There is object detection for basic objects such as detecting coffee mugs or judging facial emotions on a human. We can detect cars vs bikes in traffic but there is nothing used for detecting different car parts. My aim of the app is to provide a more detailed breakdown of object detection for cars.

10.0 Technical Approach

For the development of EZMechanic, I plan on using a mix of Agile/Scrum methodology and Kanban approach to development.

By utilising both methodologies I can use the benefits of Agile's Sprint cycles and also Kanban's approach of having clarity over of the Workflow tasks that are in progress, to do and completed.

Also, by using Agile I can quickly adjust the applications UI/altering features etc. based on how the progress is evolving.

Since EZMechanic will be primarily focused on machine learning object detection,

requirements will be identified by first using a Prototype approach. This will give me a clear observation of what objects are being detected well and what other parts need improvement. Once testing is completed from a prototype, I can see how accurate the overall object detection model is.

The requirements of the project will be broken down into tasks based on time completion, so how long the given task will likely take to complete. Also, the importance of the tasks meaning which tasks are needed to be complete to finalize the project. Milestones will be comprised of completing the major tasks such as the object detection feature, midpoint, and final completion of the project.

11.0 Technical Details

For my project, it will be heavily revolved on machine learning which is derived from artificial intelligence.

The mobile app will be an android application so Android studio will be used as the primary IDE.

I plan to use Java as the main programming language for the Android app, I may switch to Kotlin if needed.

TensorFlow will be the open-source library for machine learning that I will utilise. A small amount of Python will be used for the creation and training of the machine learning training model. I plan to use Google Colaboratory as I can run my python code on the browser. This allows me to use their hardware/GPU's and it is free to use.

12.0 Special Resources Required

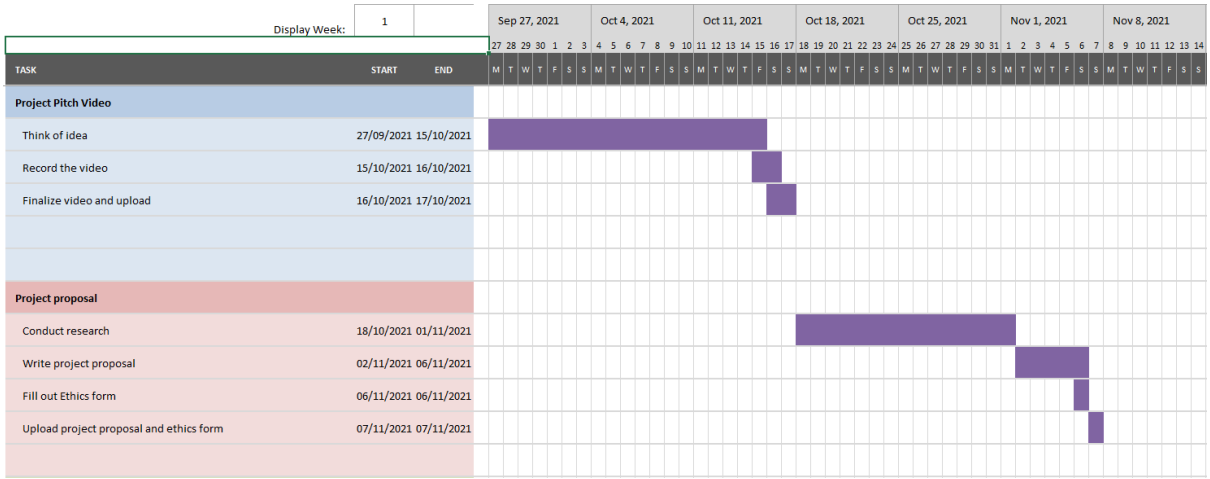
Android studio to be used for the android application.

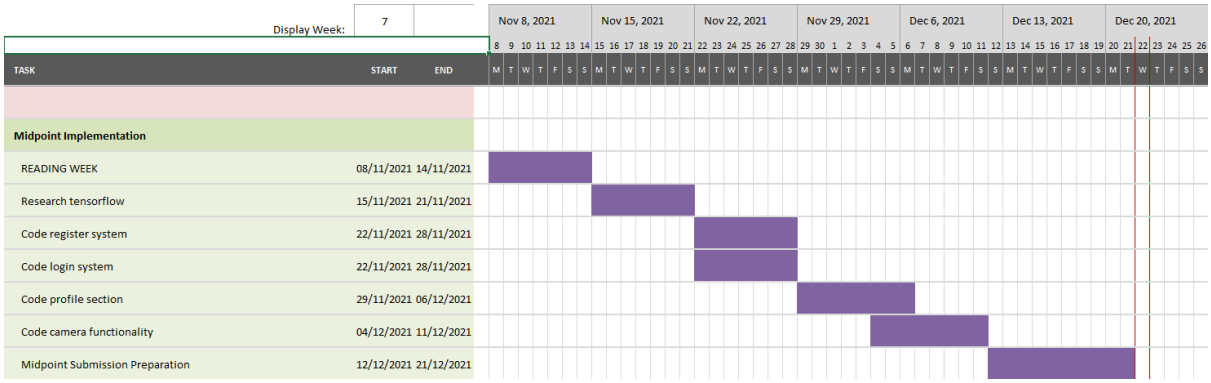
TensorFlow will be the machine learning resource I will require.

It is likely that I will use Google Colaboratory to run my python code, if this will not be possible then I will use my own pc environment.

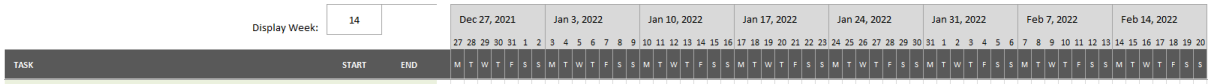
Images of car parts to use for my training model.

13.0 Project Plan

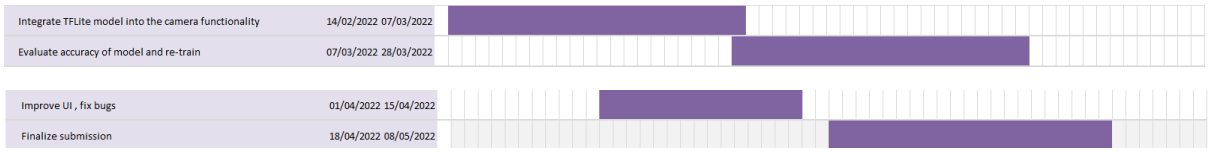




TASK	START	END
Midpoint Implementation		
READING WEEK	08/11/2021	14/11/2021
Research tensorflow	15/11/2021	21/11/2021
Code register system	22/11/2021	28/11/2021
Code login system	22/11/2021	28/11/2021
Code profile section	29/11/2021	06/12/2021
Code camera functionality	04/12/2021	11/12/2021
Midpoint Submission Preparation	12/12/2021	21/12/2021



TASK	START	END
Code profile section	29/11/2021	06/12/2021
Code camera functionality	04/12/2021	11/12/2021
Midpoint Submission Preparation	12/12/2021	21/12/2021
Final Implementation		
Christmas Break	20/12/2021	17/01/2022
Gather training images	17/01/2022	24/01/2022
Train Tensorflow model	24/01/2022	07/02/2022
Integrate TFLite model into the camera functionality	14/02/2022	07/03/2022
Evaluate accuracy of model and re-train	07/03/2022	28/03/2022
Improve UI , fix bugs	01/04/2022	15/04/2022
Finalize submission	18/04/2022	08/05/2022



Integrate TFLite model into the camera functionality	14/02/2022	07/03/2022
Evaluate accuracy of model and re-train	07/03/2022	28/03/2022

Improve UI , fix bugs	01/04/2022	15/04/2022
Finalize submission	18/04/2022	08/05/2022

14.0 Testing

Unit testing:

Scenario 1: Create the user profile

- Can the user successfully create a profile with the details entered?
  If not, the test fails.
- Are the details sent to Firebase and stored correctly?
  If not, the test fails.
- Can the user create another profile with the same details?
  If yes, the test fails.
- Can the user login successfully using Firebase authentication?
  If not, the test fails

Scenario 2: Display user profile

- Are the details retrieved from Firebase and displayed correctly?
  If not, the test fails.

Scenario 3: Camera function

- Can you take a picture with the camera functionality in the app?
  If not, the test fails.

Integration testing:

Scenario 1:

- Does the android app connect successfully to Firebase?
  If not, the test fails.

Scenario 2:

- Does the TensorFlow Lite model detect objects successfully via the camera functionality of the android app?
  If not, the test fails.

Bibliography

Anon., 2018. *netguru.* [Online]
Available at: https://www.netguru.com/blog/image-recognition-apps
[Accessed 7 November 2021].

Rosencrance, L., 2019. *Techtarget.* [Online]
Available at: https://searchmobilecomputing.techtarget.com/definition/Google-Firebase
[Accessed 7 November 2021].

14.1. Reflective Journals

October Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: OCTOBER 2021

What?

For the month of October, I was thinking of ideas to submit for my project pitch video that was due on October 17th. I initially had the idea of creating a video conference web application which would have video calling, chat function etc. After a brief discussion with Frances Sheridan, she suggested to think of another idea for my pitch, as the idea would lack in complexity and would not be enough. After this I came up with a mobile application idea that would be something of a mechanics app. The main point of the app was to allow users to take a picture/scan parts of their car and using object detection, the application would tell you the results.

I now had an idea, and I created my project pitch video and submitted it which was due on October 17th .

The supervisor list also came out and my supervisor for the Software project is Vanessa Ayala-Rivera which I will be contacting to discuss my project.

So What?

I have identified a project idea and I need to research and gather requirements to complete the project proposal form.

Now What? I will now be working on the project proposal form to complete and upload by November 7th .

Apprentice Signature	Filip Munteanu

November Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: NOVEMBER 2021

What? In the month of November , I have submitted my project proposal on November 7th . I had my first meeting with my supervisor, and she gave some feedback on the project pitch that was submitted and suggested that the main focus of the project which is the machine learning, would be sufficient for the scope of my project . I have been very busy with my other modules as I have a couple of assignments to complete and upload .
So What? I have not done much more work on my project because of the other module work. I will meet with my supervisor to discuss the mid-point submission and to see what she thinks will be sufficient functionality to have completed.
Now What? Once I meet with my supervisor which will be the 2nd of December , I will have a clearer path and what needs to be done and work towards the mid-point submission .

Apprentice Signature	Filip Munteanu

December Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: NOVEMBER 2021

What? In the month of December , I have submitted my midpoint deliverable on December 23rd .After meetings with my supervisor she suggested that a login, register system and the basic UI will suffice for the midpoint submission.	
So What? The mid-point submission deadline has now been reached and will start again in Semester 2.	
Now What? Once Semester 2 starts, I will need to begin focusing on the machine learning aspect of the project and began the work needed to get that done.	
Apprentice Signature	Filip Munteanu

January Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: January 2022

What? For the month of January , I've not done much regarding my project besides some further research on Tensor flow. I was busy with TABA's, and I also had my break after the exams ended.
So What? I need to start beginning work on Tensorflow and creating a machine learning model in order to even begin implementing into my android app.
Now What? I will create a list with all the tasks that are necessary to be done in order to complete my project in May , also I will identify project milestones . I need to contact my supervisor to arrange meetings for this second semester.

Apprentice Signature	Filip Munteanu

February Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: February 2022

What? For the month of February , I have gathered training images for my car parts and was able to successfully create and train a tensorflow model and export it to a .tflite format.
So What? I need to start beginning work on implementing the model into my android application . I also need to gather more training images in order to increase detection accuracy . As of now, there is only about 50% detection accuracy which is fine because I did not have many images to begin with.
Now What?

I will begin to gather more training images and retrain my model. I will also try to implement the detection functionality to see how it works.	
Apprentice Signature	Filip Munteanu

March Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: March 2022

What? For the month of March , I was busy with working on and completing other assignments from my modules and so did not work on my project.	
So What? I need to finish my other modules and projects so I can focus on my final year project.	
Now What? I will be gathering more training images and retrain my model and also be adding another object to my model. I will also try to implement the model into my android application .	
Apprentice Signature	Filip Munteanu

April Monthly Journal

Student Name	Filip Munteanu
Student Number	X18359843
Course	BSHC4

Month: April 2022

What? For the month of April , I was also busy with exams and finishing my modules, but I also successfully implemented my TensorFlow lite model into my android application and it detects the objects correctly and draws the bounding boxes on the image to show the location of the object.
So What? I need to refine the model by adding in some extra training images to increase accuracy .
Now What? I will be working on finishing up the applications interface and the refinement of the model. I will also work on completing the technical report , project poster and presentation for the end of May.

Apprentice Signature	Filip Munteanu

14.2. Other materials used