Pay As You Drive

Technical Manual

Rhyder Quinlan
C00223030

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW

Abstract

The purpose of the Pay As You Drive project is to develop a mobile application for the Android & iOS platforms that will, first and foremost, track driving habits of a user whilst travelling on a journey. Driving habits are recorded and used for motor insurance purposes, whereby a user pays a monthly insurance price based solely on how much and how well/poorly they drive. The application allows the user to view their past journey, control their profile details and vehicles through the app, and finally create and monitor insurance claims within the application.

Pay As You Drive is designed with a Peer 2 Peer business approach, whereby users are nodes in a network. Each month the nodes are billed for their driving, and this money is gathered into one central money pool. In the event of an accident, fire or theft to their motor vehicle an insurance claim can be made, and if approved, a transfer takes place from the money pool to the users wallet.

Table of contents

Table of Figures

Section 1 - Introduction

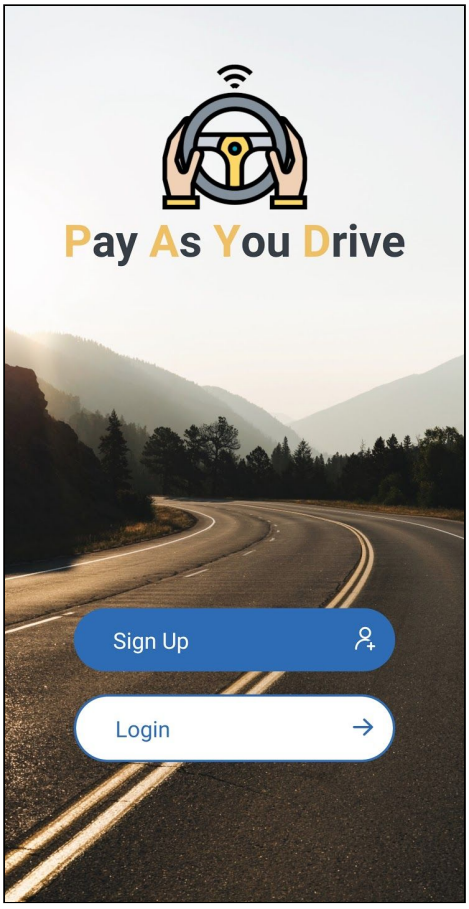
The purpose of this technical manual is to explain, in-depth, the functionality and design behind the final product of this project. This document will describe the navigational flow, what each component on the screen does and why it is there, and any other technical nuances related to the screens.

The document breaks the screens down into 8 high level sections, each containing the screens relating to that section of the application.

Please see the user manual for a high level overview of the application from a standard users perspective.

Section 2 - App Screen Features

2.1 - Welcome And Authentication Functionality

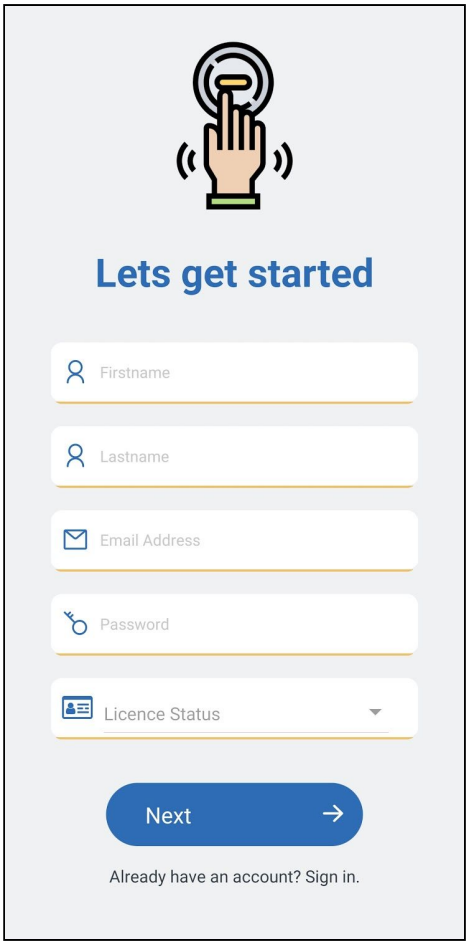


Welcome Screen

This is the opening screen the user will be greeted with. From here there are three paths, two triggered by the user and one triggered by the application itself.

The user can either move to login or sign up, or if the user has previously logged in and selected the ‘remember me’ option on the login screen then from here the user will automatically be directed to the homescreen.

Fig. 1 Welcome Screen screenshot

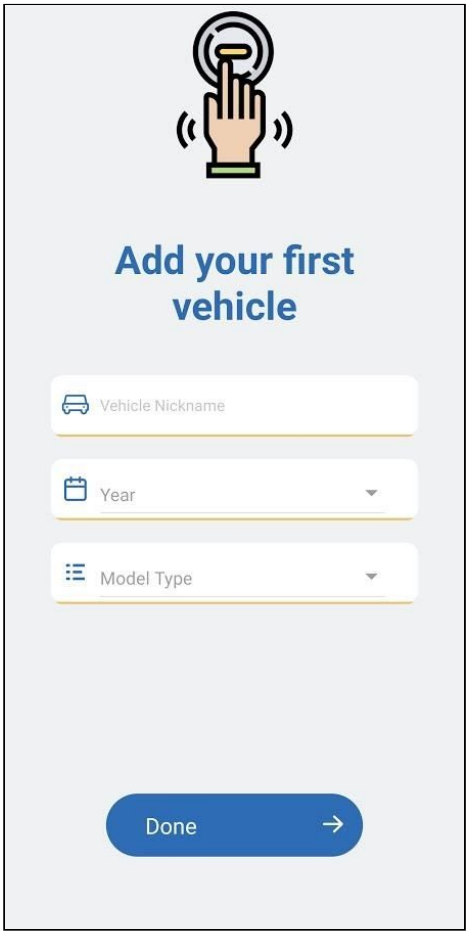


Create New User Screen

For a user to use the system they must be registered before they can log in. To register with the system the user enters their first and last name, their email address, a password (at least 8 characters long) and their current licence (provisional licence or full licence).

The form requires all fields to be filled in and valid before moving onto the next stage of registration.

Fig. 2 Create new user screenshot



Add Vehicle Screen

Once the user is registered with the system they will be directed to a screen where they are required to add their first vehicle.

All fields are required.

One form inputs are validated and successfully added to the database then the user is redirected to the Home screen.

Fig 3. Add vehicle screenshot

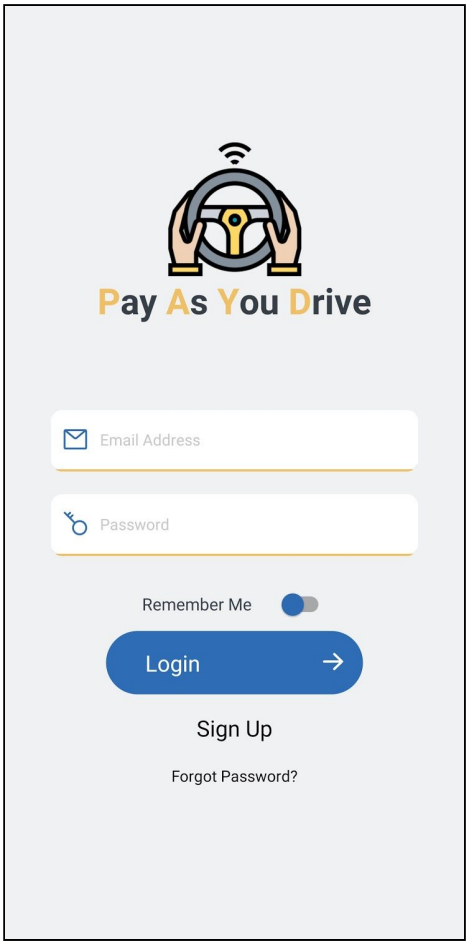


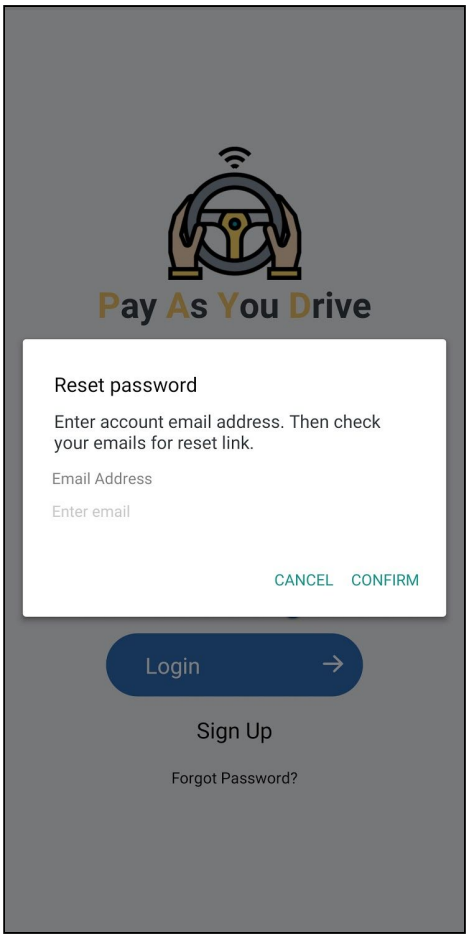
Fig. 4 Login screen screenshot

Login Screen

To authenticate the user the login screen takes in the users email address and password, and validates this information with Firebases authentication capabilities by comparing the inputs with previously registered users within the database.

A user can tell the application to ‘Remember Me’ so that the next time this user opens the application they will be automatically directed from the Welcome screen to the main Home screen.

If the user has forgotten their password they will see a prompt shown in the next screenshot below.



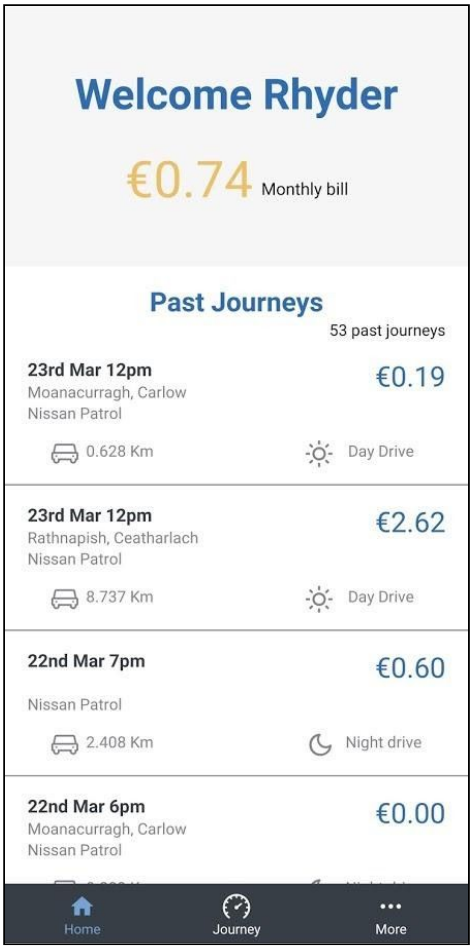
Reset Password Prompt

If the user has forgotten their password they can click the ‘Forgot password?’ button where they will be prompted to enter their email.

An email will be sent to the email address (if valid) that is associated with the users registered account.

Fig. 5 Reset password prompt

2.2 - View Bill & Past Journeys Functionality



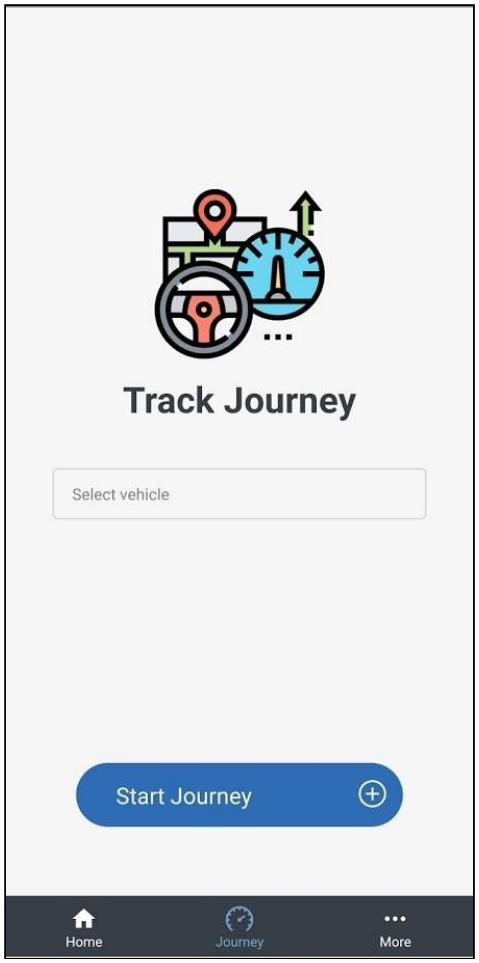
Home Screen

Once authenticated the user will land on the homescreen.

The home screen is made up of two sections, the upper section displays the bill for the current month, and the lower section shows a scrollable list of the users past journeys (ordered by date).

Fig. 6 Home screen

2.3 - Track Journey Functionality



Start Journey Screen

This screen allows the user to firstly select which vehicle they want to assign to the journey, and then start the journey.

A vehicle must be selected before a journey can begin, and an alert will be triggered if the user does not make this selection.

Starting the journey will trigger changes to the screen (not a new activity) as shown in the next screenshot.

Fig. 7 Start Journey Screenshot

Beginning Journey Countdown

This screen displays for 3 seconds as an interim before the journey starts. For convenience and usability the user is able to cancel the journey before it starts, in cases where a journey was started by accident.

The countdown timer indicates how long (in seconds) before the journey starts, counting down from 3 to 0.

Fig. 8 Beginning Journey countdown screenshot

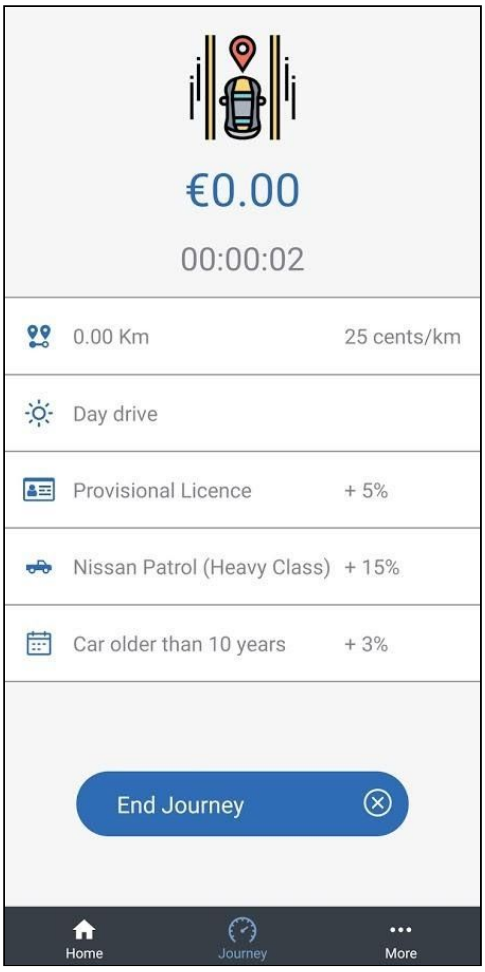


Fig. 9 Ongoing Journey Screen

Ongoing Journey Screen

Once a journey has started the user can view a summary for the duration of the journey.

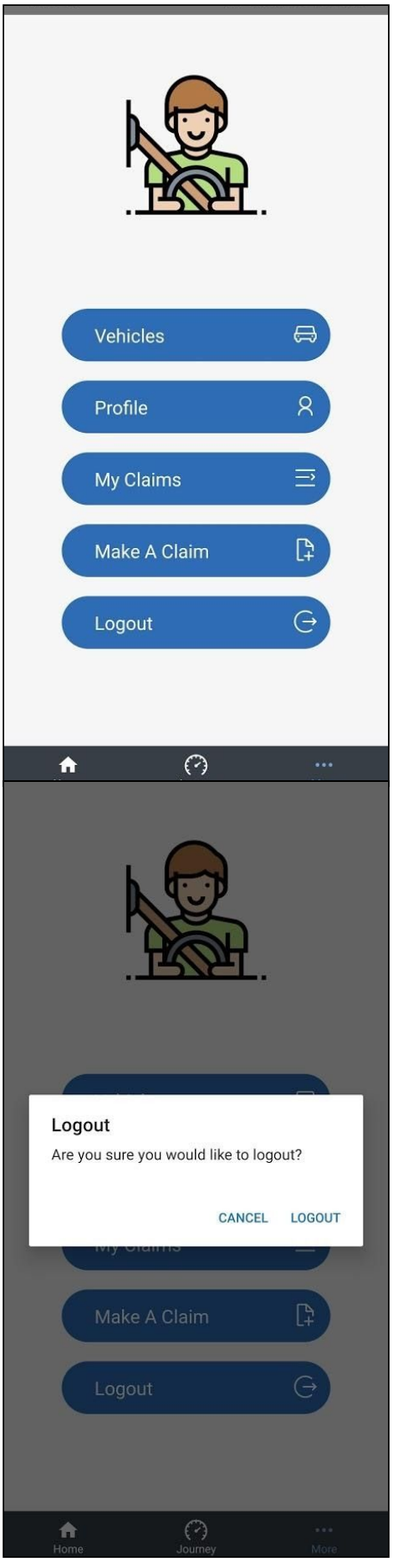
This screen is designed with minimal animations and changing figures so as to be of no distraction to the driver whilst driving.

At the top the user can see the cost of the ongoing journey and it's duration. In the middle is a breakdown of this cost into:

- Mileage
- Night or Day drive
- Licence status addition
- Vehicle class addition
- Year addition

The user can then end the journey and will return to the track journey screen above.

2.4 - Extras Menu



More Screen

This screen is found on the far right of the bottom navigation bar.

From here the user can access extra necessities:

- Vehicles
- Profile
- My Claims
- Make A Claim
- Logout

The user will be directed to the screen of their choosing.

If ‘logout’ is selected the the user is prompted to confirm if they would like to logout (shown in below screenshot).

Fig. 10 More screenshot

Logout Confirmation Prompt

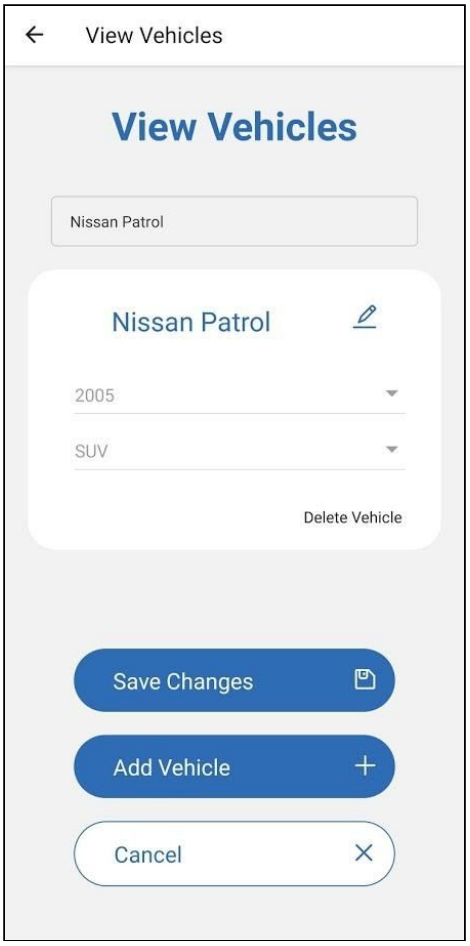
This prompt is displayed once the use selects the ‘Logout’ button.

If they confirm logout then they will be directed back to the welcome screen.

This event triggers the firebase authentication to logout the user and the remember me async storage to forget the user.

Fig. 11 Logout prompt

2.5 - CRUD Vehicles Functionality



View/Edit Vehicles

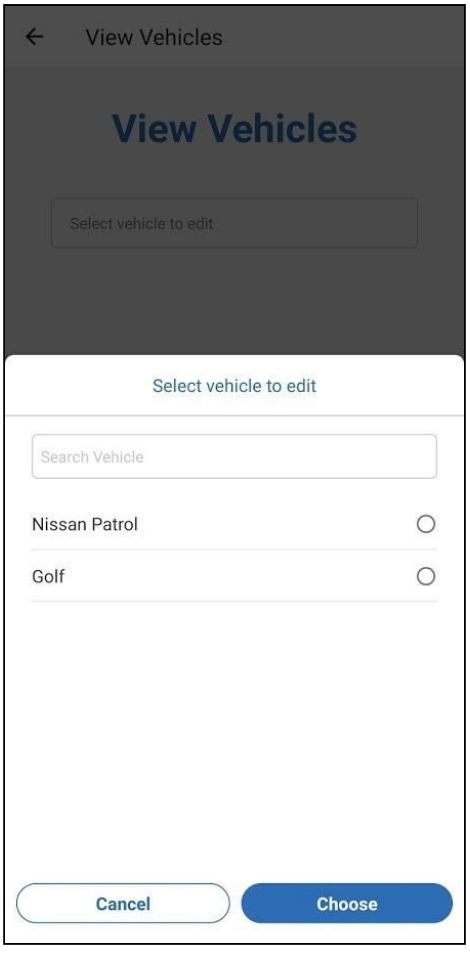
On this screen the user can select a vehicle from the popup shown below.

Once a vehicle is selected the user can view or edit the details of the vehicle, and upon submission the new data within the database will be updated.

The user can also choose to delete the vehicle. If this is selected the user is prompted for confirmation to make sure this is what the user wants.

From here the user can choose to add a vehicle and they will be directed to the Add Vehicle screen shown below.

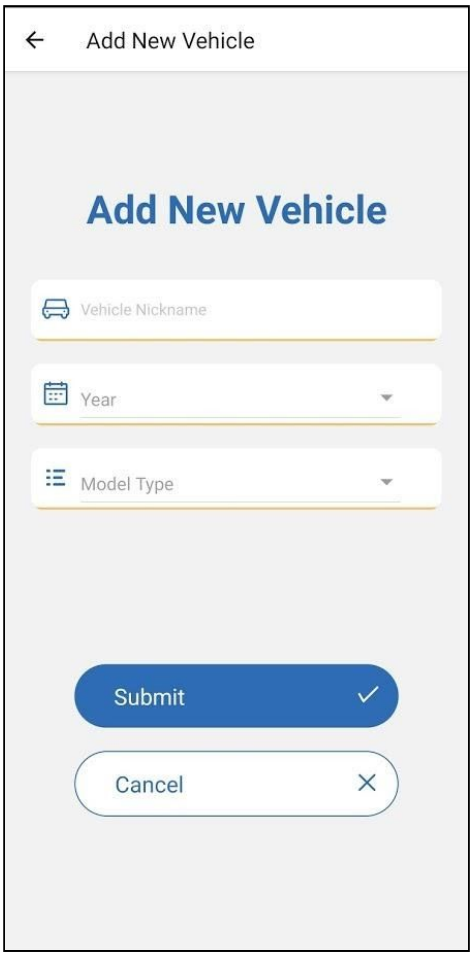
Fig. 12 View/Edit Vehicles screenshot



Select Vehicle Popup

This popup shows a searchable list of vehicles, where only one vehicle can be chosen.

Fig. 13 Select vehicle popup screenshot



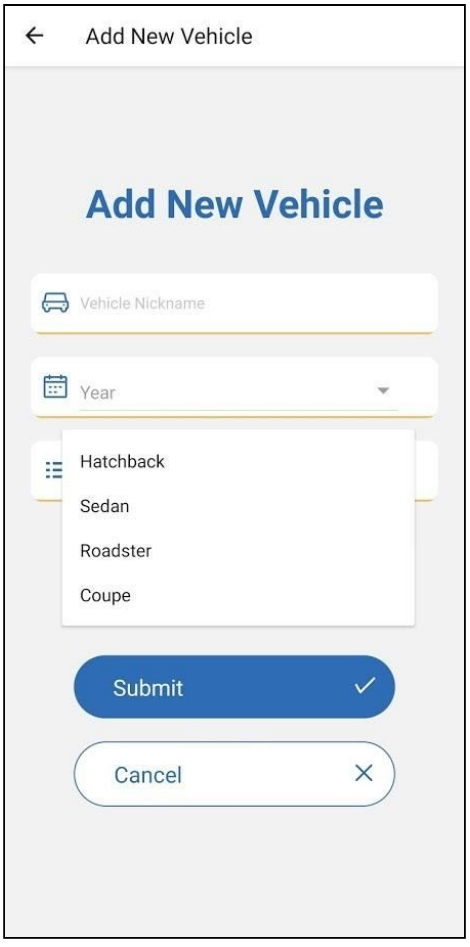
Add New Vehicle Screen

The user can add a new vehicle to their Vehicle list here. The list of years is taken from from the current year and going back 50 years.

The dropdown showing the model type data is shown in the next screenshot.

All fields are required.

Fig. 14 Add New Vehicle Screenshot



Model Type Dropdown

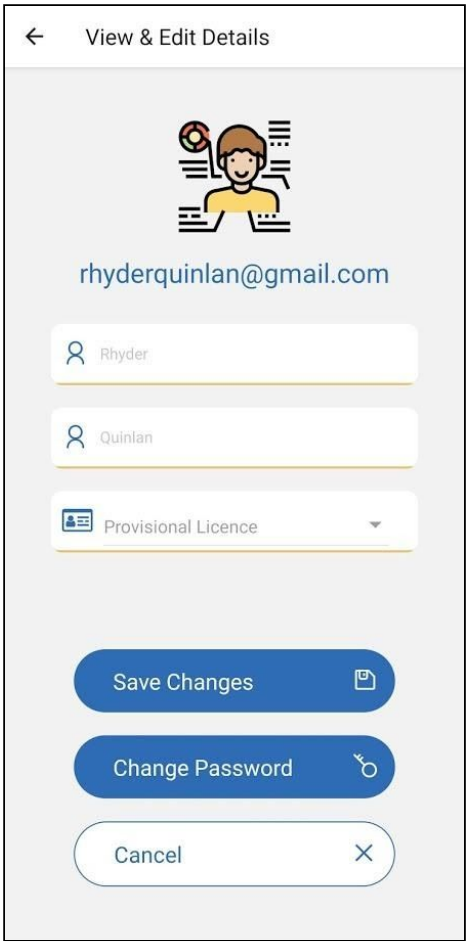
This dropdown shows a list of predefined car model types:

- Hatchback
- Sedan
- Roadster
- Coupe
- SUV
- Pickup
- Minivan

List is scrollable so this screenshot does not display all the options.

Fig. 15 Model type dropdown screenshot

2.6 - Update User Details Functionality



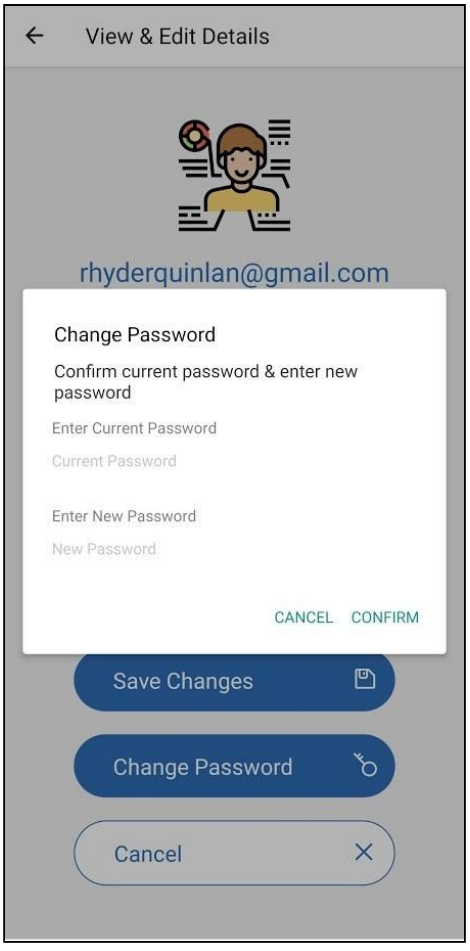
Update User Details Screen

On this screen the user can update their profile information.

On submission the users authentication details and profile details will be updated.

If the user chooses to change their password they will see a security prompt as described in the next screenshot.

Fig. 16 Update User details screenshot



Change Password Prompt

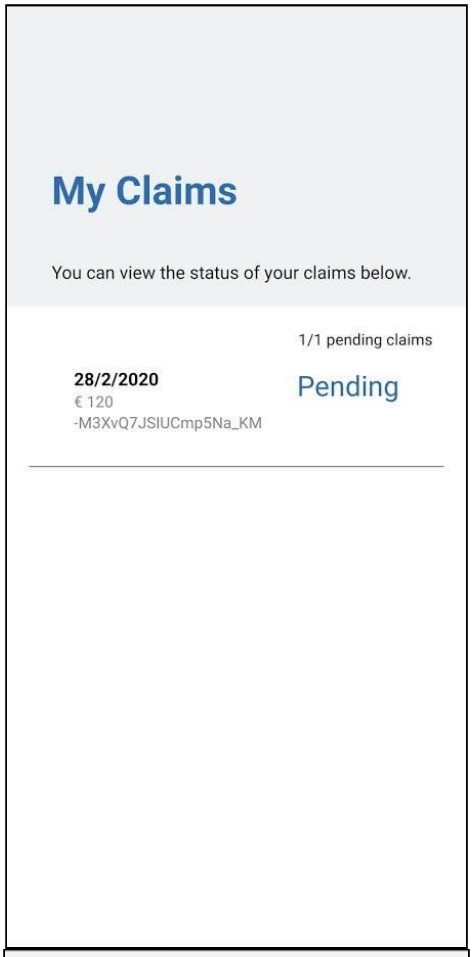
This prompt is triggered when the user wants to change their password.

The user is asked to enter their current password, and their new password.

The current password is cross-referenced with the Firebase authentication information and if correct the password is changed. If it is incorrect an error will display on the screen alerting the user the action was unsuccessful. If the action is successful the user will receive an on-screen prompt to alert them of this.

Fig. 17 Change password prompt screenshot

2.7 - CRUD Insurance Claim



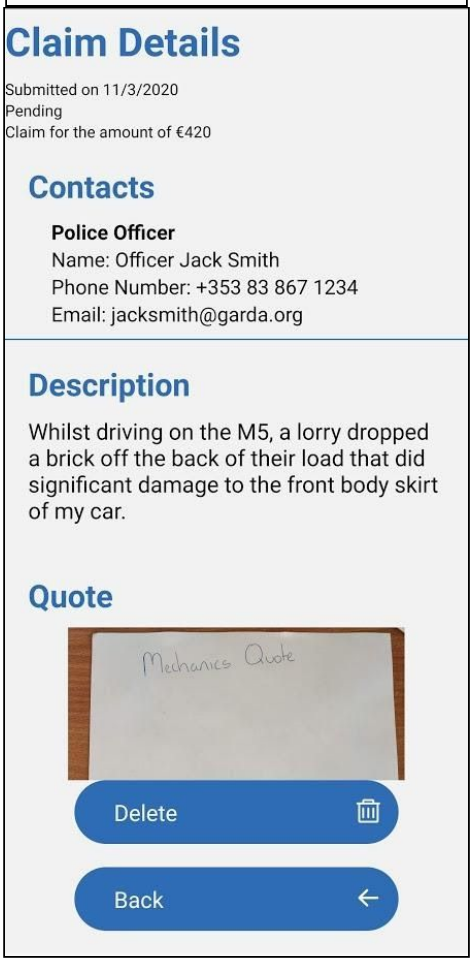
My Claims Screen

This screen shows a list of claims the user has made in a scrollable list.

Each claim in the list can be selected and the user will be directed to the View Claim screen shown below.

A claims status is either:
Pending - Claim is waiting for review by admin.
Rejected - An admin has rejected the request.
Accepted - An admin has accepted the request.

Fig. 18 My Claims Screen screenshot



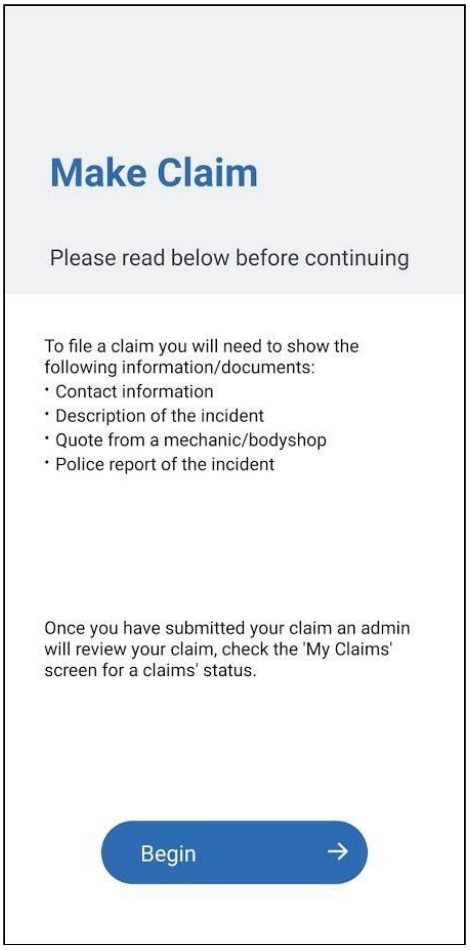
View Claim Screen

On this screen the user can view the details of a submitted claim, but make no further changes to the details that are there.
A claim can only be deleted if the claim is still pending and has not been reviewed by the Admin, and so the delete button would not be visible on a claim that has been reviewed.
The details of a claim are in a scrollable view, so all contents of the claim are not visible in this screenshot. The details shown are:

- Contacts
- Description
- Picture of Mechanic/Body shop quote
- Picture of police report

Fig. 19 View Claim Screen

2.8 - Make A Claim Functionality



Make A Claim Information Page

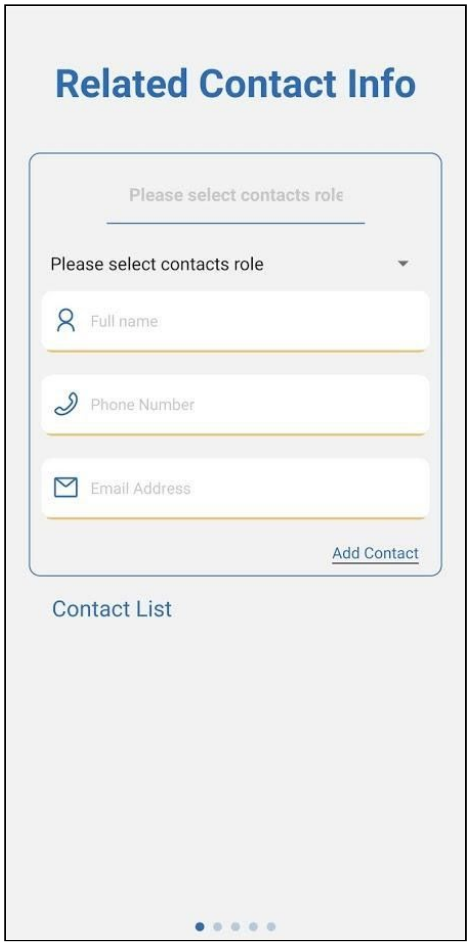
From the Extras Menu the user can select Make a Claim whereby they will be directed here.

This screen displays useful information to know before beginning the process of making a claim, as well as what happens after a claim is submitted.

After the user chooses to begin they will be directed through the process described in the screenshots found on the next page.

Fig. 20 Make a claim information page

The following screenshots show the process a user goes through to make an insurance claim. To boost the users experience with this process the screens are on a ‘carousel’, this means the user can swipe left or right on the screen to bring up the next page. The user's position on the ‘carousel’ is shown by the page indicator at the bottom of the screen.



Pg. 1 - Add Relevant Contact Information

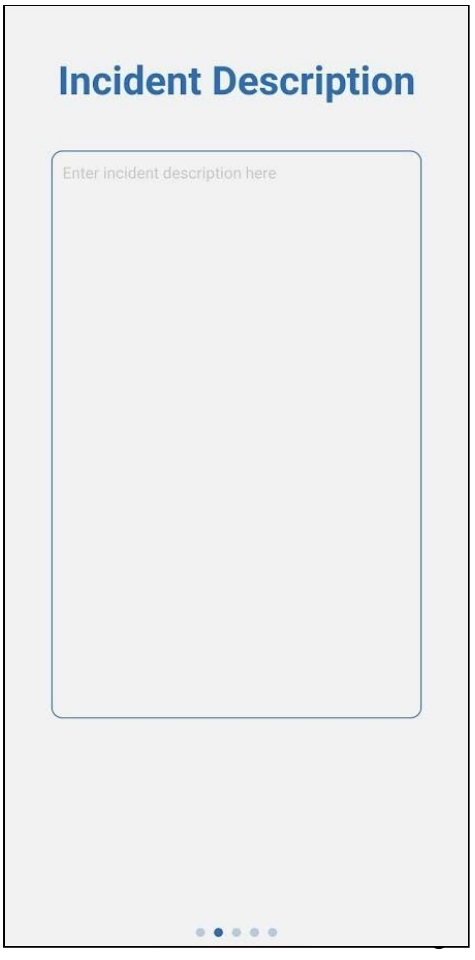
This is the first page of the carousel where the user is asked to add any contacts that may be helpful when the claim is being reviewed.

The contacts roles can either be a witness to the incident, or a responding police officer.

The user can add as many contacts as they see fit, and then grow their contacts list shown at the lower section of the screen.

Fig. 21 add contact info screenshot

Swipe right to left



Pg. 2 - Incident Description

This screen lets the user enter a paragraph explaining their claim, and what happened to cause the incident.

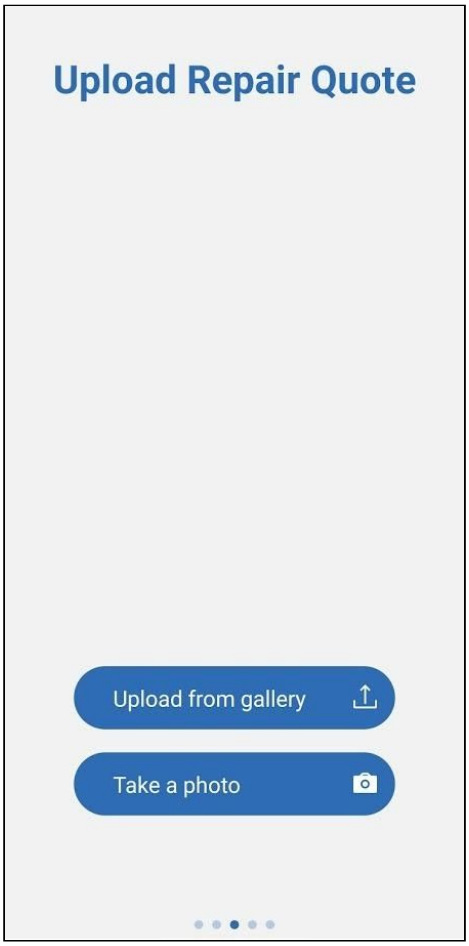
Fig. 22 Incident description screenshot

Pg. 3 & 4 - Upload Images

The next two screens require the user to upload an image either chosen from their phone storage (“Upload from Gallery”) or take a photo with their phone camera (“Take a Photo”).

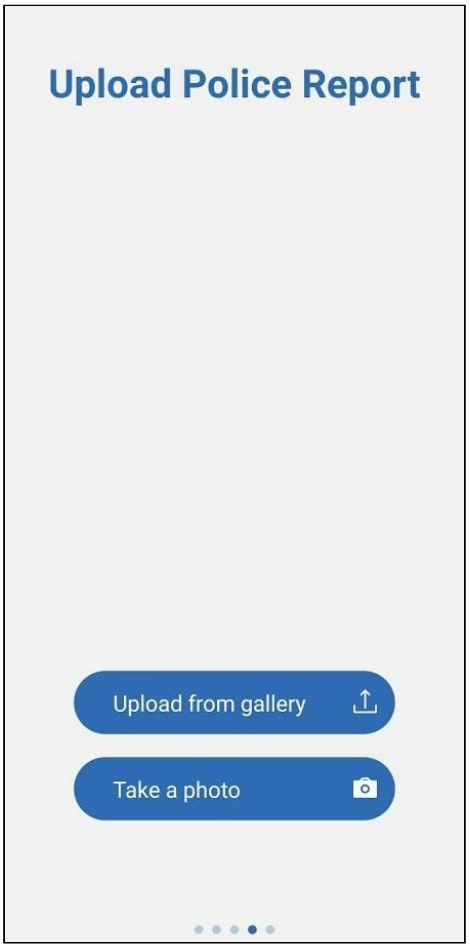
If ‘Upload from Gallery’ is chosen the user will either be directed to Camera Roll on iOS or to Android Files on Android, where they can select a stored image (.jpeg or .png) to upload. If ‘Take a photo’ is selected then the phone's camera application will open, once the picture is taken the user can select ‘yes’ or ‘no’ to confirm.

Once a picture is chosen it will be displayed in the center of the screen below the heading.



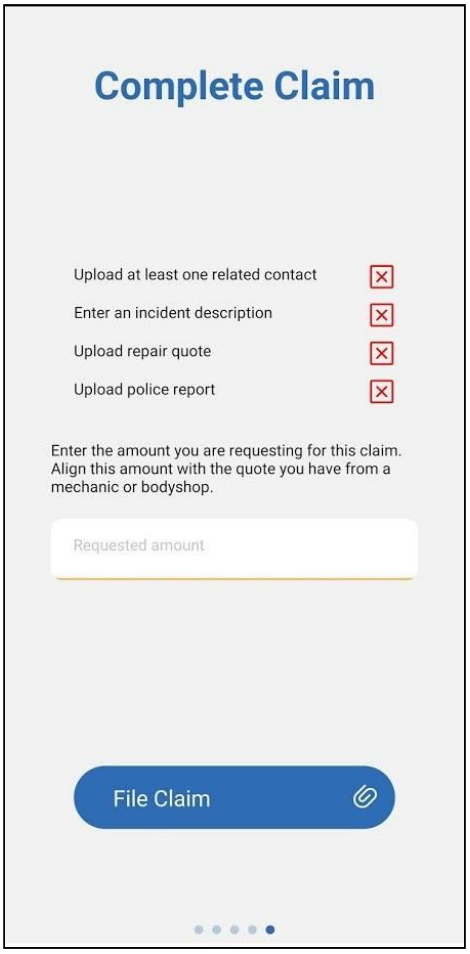
Pg. 3

Fig. 23 Upload repair quote



Pg. 4

Fig. 24 Upload Police Report



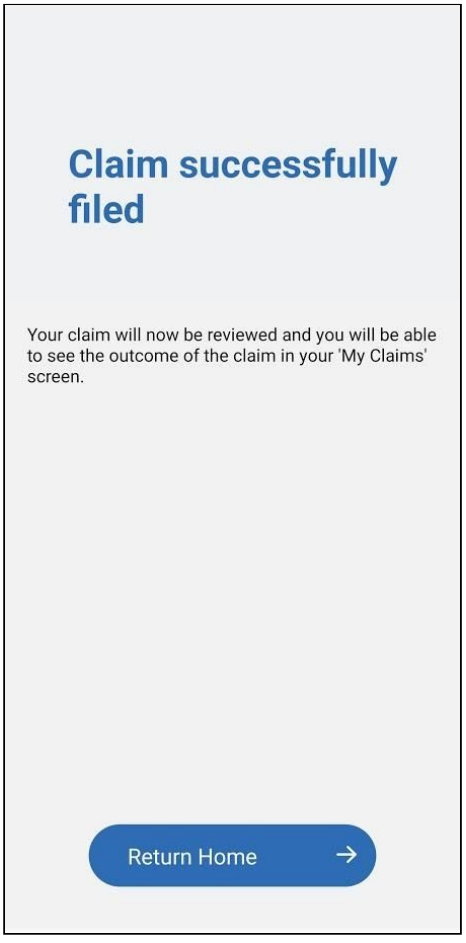
Pg. 5 - Claim check list and amount

The final page of the 'carousel' is a checklist, whereby the user can see the validity of the information they have entered, and the remaining tasks to achieve. Once an item is considered complete the red cross will change to a green tick.

The user is also asked for the amount they are claiming for.

Once the user chooses to file the claim, the system will validate if all required information has been entered, and if so proceed to the Claim success screen shown below.

Fig. 25 Claim checklist and amount screenshot



Claim Confirmation Screen

If the claim is successfully filed the user will be directed to this screen.

It is simply acknowledgement that the user has filed the claim and it is now pending review.

From here the user is directed back to the Home screen.

Fig. 26 Claim confirmation screenshot

Section 3 - Track Journey

At the start of the journey an event listener runs the following code:

```
navigator.geolocation.watchPosition((position) => {
    //journey code here
    //position {object Object}
})
```

This function runs every time a change in Latitude and Longitude occurs.

The important information need from *position* is:

Object {
"coords": Object {
"accuracy": 347.6700134277344,
"altitude": 0,
"heading": 0,
"latitude": 52.8279772,
"longitude": -6.9349665,
"speed": 0,
},
"mocked": false,
"timestamp": 1573302050037,
}

With this array updating every time there is a change in position, the following functions are built.:

3.1 - Track Distance

```
//return distance in km
calcDistance(start, end){
    //distance = difference in longitude and latitude
    //using haversine algorithm
    const distance = haversine(start, end, {unit: 'km'}) || 0
    return distance
}
```

The calcDistance takes in 2 variables, *start* and *end. Start* represents an array of the new/current latitude and longitude, and *end* represents an array of the previous latitude and longitude.

The calculation uses haversine's formula for calculating the distance between the coordinates. The algorithm works as follows:

$$\operatorname{hav}(\Theta) = \operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)$$

(Wikipedia, 2020)

Where:

- φ_1, φ_2: latitude of point 1 and latitude of point 2 (in radians),
- λ_1, λ_2: longitude of point 1 and longitude of point 2 (in radians).

3.2 - Track Speed

Although the position array has a speed element, it was very inaccurate and design reverted back to calculating speed as distance over time.

```
//return spped in km/hr
calcSpeed(prevTimestamp, newTimestamp, distance) {
    //get delta time in milliseconds
    const unix_time = new Date(newTimestamp).getTime() - new Date(prevTimestamp).getTime()

    var speed = 0
    if(distance != 0){
        //speed = distance / time
        speed = distance / ((((unix_time)/1000)/60)/60) //convert to km/hr
    }

    return speed
}
```

calcSpeed takes in the time at which the last location was recorded, the time at which the new location was recorded, and the distance travelled (calculated by the formula above).

The difference between the two timestamps is then calculated (in milliseconds). To convert the unix time to hours it must be divided by /1000 (milliseconds), /60 (seconds), and then /60 (minutes).

Then using speed = distance/time the speed can be calculated.

3.3 - Track Acceleration

```
//return acceleration in m/s^2
calcAcceleration(v_0, v, t){
    //acceleration = v/seconds - v_0/seconds / time elapsed
    return ((v/3.6) - (v_0/3.6))/t
}
```

Acceleration is the change in velocity over a period of time. To work this out:

- *V_0* represents the initial velocity.
- *V* represents the final velocity.
- *T* represents the change in time

Velocity measurements are in km/hr originally so must be converted to m/s^2 by dividing V_0 and V by 3.6.

The formula is:

$$\overline{a} = \frac{v - v_0}{t} = \frac{\Delta v}{\Delta t}$$

(Wikipedia, 2020)

Section 4 - Bibliography

Wikipedia, 2020. *Haversine Formula*. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/wiki/Haversine_formula> [Accessed 20 April 2020].

Wikipedia, 2020. *Acceleration*. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/wiki/Acceleration> [Accessed 20 April 2020].