**The Driva App**

**Project Scope**

I am looking to develop a native iOS mobile app for cars. I have plans to extend this app to Android in the future. The app will be an iOS app for drivers and it will be launched for users in UK market. The app will be optimized for both iPhone and iPad use (iOS 9 and up).

The main features of the app are as follows:

1) Allows users to register a vehicle/s that belong to them.

2) Allows users to diagnose their vehicle, view and reset problem codes.

3) Allows users to search and find service garages or vehicle related services (recovery, mobile tire shops, servicing, etc…) based on feedback ratings or closest distance.

4) Allows users to book a service with a garage, and leave feedback and ratings once the service is complete.

5) Allows users to keep a track record of their vehicle’s service history record all under their account. Service history can also be transferred from user to user once a vehicle’s ownership has changed.

**UI of the Driva app**

The prototype of the app can be found here:  
<https://marvelapp.com/993d7h4/screen/46015134>

The prototype demonstrates the main functionality of the app. It may require a few edits (such as a few screen inclusions) once development has begun, but I would say that it is 95% complete. I have all the screens in PSD format ready.

**APIs required for the Driva app**

There are a few API’s that will be required for developing the app.

1. **Registering a vehicle**  
   <https://marvelapp.com/993d7h4/screen/46015144>  
   This allows the user to simply search up their vehicle using the vehicle registration number. Only UK vehicles can work here.  
     
   The API for this functionality can be obtained via the following website:  
   <https://www.motorspecs.co.uk/>

You will be required to sign up. There is a free test service available.

This API also retrieves the display picture of the vehicle as shown here:  
<https://marvelapp.com/993d7h4/screen/46015154>

1. **Diagnose a vehicle**  
   <https://marvelapp.com/993d7h4/screen/46015148>  
   Diagnosis allows the user to communicate with their vehicle’s dashboard computer. This is the most important part of the app and it is imperative that this is developed correctly. How it works is that the user will have a hardware connector that will connect to their vehicle. This hardware connector allows the vehicle’s dashboard computer to communicate with our app on the iPhone/iPad device (via a wireless Bluetooth connection). Troubleshooting issues with the vehicle can then be relayed to the user’s device.   
     
   The link to this API can be found here:  
   <https://console.bluemix.net/docs/services/HellaVentures/index.html#setting-up-car-diagnostic-api>  
   Again, this is an important part of the app. Please look up on this API and ensure that you are able to integrate it into the app.
2. **UK Postcode search**  
   <https://marvelapp.com/993d7h4/screen/46015172>  
   Allows the user to put in a UK postcode or location into the app that when searched will return a list of the nearest garages to that postcode or location.  
     
   Google has an API for UK addresses and postcodes.

**Premium Users**

Generally, the app will be free to use, but there will be some features that require the user to upgrade their account. These features include:

* Adding multiple vehicles onto an account
* Revealing and troubleshooting diagnostic issues found

In order to upgrade an account, I think the best idea is for the user to make payment on our site rather than through the app. This way, the fees for processing payments via the App Store (and eventually Play Store) are avoided. It can work by having a link in the app that directs the user to our website. Once payment has been completed on our website, the user’s account on the app will automatically be upgraded, unlocking all the features for the user.

**Bookings**

<https://marvelapp.com/993d7h4/screen/46015178>  
<https://marvelapp.com/993d7h4/screen/46015184>

The complete scenario of the booking process is not displayed on the prototype as it could be lengthy to display this way, so please read the explanation below so you can know the details of booking on the app.

Bookings are where the app user and the garage user can negotiate/agree on prices for a service and is also where the user can get updates on a booking (example, nearly complete, complete, started, etc…). I never included a “Messages” section because I wish to avoid a “conversational chat” format between the app user and garage user on the app. Only the most important stages of a service will be captured by the app.

For example, Chris contacts a garage called AutoFix to service his car. Chris uses the app to book a suitable date to bring his car in. AutoFix will see this request and will outline the service in detail (in this case servicing) and will either except the date or request a change of date. AutoFix will also provide a quotation for the service. If Chris accepts the time and quotation, he is primarily booked in and AutoFix expects his car on the agreed date. When the car gets to AutoFix on the agreed date, AutoFix will then check the car and will have the final opportunity to edit the service details, quotation and estimated completion date. If terms between Chris and AutoFix are agreed upon, Chris will leave his car with AutoFix and the servicing can begin. Chris will receive updates to the progress of his car (such as work undergoing, nearly complete, etc…). Once the servicing has been complete by AutoCar, Chris will be indicated of it and can make payment for the servicing as agreed and leave feedback for AutoFix.

**Addition Information on the UI**  
  
<https://marvelapp.com/993d7h4/screen/46015178>  
The garage profile must display accepted payments such as Cash, Credit Cards, BACS, etc… in the garage profile. This can be shown as normal text at the end. Please keep this in mind during developing the backend of the garage user.

<https://marvelapp.com/993d7h4/screen/46015191>

<https://marvelapp.com/993d7h4/screen/46015192>

The “History” and “Reviews” display the wrong display pictures. They are supposed to be Garages and there are no usernames with Garages. I can get them edited if required.

<https://marvelapp.com/993d7h4/screen/46015154>

The arrows to the left and right of the BMW display picture are only for premium users who have registered with more than one car on the app.

<https://marvelapp.com/993d7h4/screen/46015157>

The icons symbolize a particular service type. From the “AutoCar” example from left to right: “Electrical”, “Tyres”, “Repair” and “Parts Replacement”. The second icon from the “Hayes Engineer” example symbolizes “Body Repair”.

**The Backend**

There are two users that require backend development. Those users are the admin and the garage. I have decided to develop each backend using a themeforest template called OSCAR:   
  
<https://themeforest.net/item/oscar-the-ultimate-bootstrap-admin-template/20324506?s_rank=3>

The layout, design and functionality of each backend will be based solely on the OSCAR theme. Of course the logo, colour theme and perhaps the font style will change to accustom the appearance of the app.

I have created some basic wireframes, which demonstrate some functionalities required for each user:  
  
Admin  
<https://www.dropbox.com/s/ibz8hmgxk6d1lhf/Driva%20Web%20Admin.zip?dl=0>  
  
Garage  
<https://www.dropbox.com/s/obpguu4q3vr5zp2/Driva%20Garage%20User.zip?dl=0>

These wireframes should be used for reference only. They only demonstrate some functionality available for each user. They are not 100% accurate.

For example, the garage user wireframes may appear as though the garage user can advertise more than one ad, which is not the case. The garage user can only advertise their (registered with us) garage only.

Another example, bar chart statistics will not appear above each page (as it does on the wireframes) because that is not the layout of the OSCAR theme.

**Details of the wireframes**

Here, I will explain some details from the wireframes that may not be clear to understand just by viewing them. It is important to understand these details (and maybe make suggestions) because it involves developing functionality that is not stated in the wireframes.

**Admin user – Garage’s page**

The admin must approve of all garages that wish to sign up to the Driva network. This page will be where the admin takes actions to approve of garages and their profiles.

How it will work is that when a garage registers with us, a physical letter will be sent out to the address entered as the garage name. That letter will contain a verification code that they must enter into their account in order to verify their address and account. Once that has been verified and their profile details have been approved by the admin here, their garage profile will appear live on our network.

* **Approve**: The garage profile will be live and advertised
* **Revision**: The admin requests revision made to the garage profile in order to make it live. There will be rules that must be met in order for a profile to be live. If these rules are not met, the admin can request revision (with an option of a custom message sent from the admin to the garage’s registered email address).
* **Suspend**: The garage profile will be suspended and removed from being live. (The option to reverse the suspension decision will be present)
* **Message**: This is where the admin can message the main user of the garage.

N.B. The personal name of the registered user representing the garage is missing from here. If the admin wishes to send a message to a garage, the personal name whoever registered the garage (like a director of a company) will be used.

**Admin user - Ads Page**

This page and the garage page should join into one page. Being as they are both to do with garage profiles.

N.B. The designer of the wireframes wrongly thought that they are both different entities.

On this page, you will also see that the admin can search ads (which we should rename as “Garage Profiles”) based on their name or ad number. As a default, the ads (“Garage Profiles”) list will be sorted by the time the ad was uploaded. That way, the admin can easily prioritize with the ads (“Garage Profiles”) that were uploaded first. There should also be a way that the admin can filter the list by only showing the “Pending” ads (“Garage Profiles”) and also only showing the “Approved” ads (“Garage Profiles”).

**Admin user – Dashboard page**

The admin can view statistics of the network overall. On the theme, the dashboard is where all the statistics are, so we will proceed with these rather than what the wireframes display.

These stats can be displayed by day, week, month, and year. They include:

* Top garage locations
* Total amount of approved garages
* Total amount of bookings
* Total amount of ads
* Total amount of ads earning
* Total amount of users
* Total amount of premium users
* Total amount of premium user’s earning
* Server performance

**Admin user – Users**

This page is supposed to display normal users and premium membership users of the app. Premium users are the users who have paid to unlock all the features of the app.    
  
N.B. This page is nothing to do with Garage users.

Here, the admin has the ability to:

* **Message**: The admin can message a premium user
* **Suspend**: This will suspend a user (The option to reverse the suspension will be present)

Like my explanation on the “Ads Page”, the admin will have the ability to search users (based on their UID), view normal, premium and both users in one list.

**Admin user – Messages**

This page is not clear in the wireframes. This page is where the admin can view messages/details about services exchanged between users and garage users. There should be a way that will allow the admin to search up a conversation between the two parties in the event of an issue (like a dispute).

N.B. The Driva app does not handle any payment for services between a user and garage. By dispute, I refer to mostly feedback disputes.

**Admin user – Settings**

This page is where the admin can: 

* Update the email login address
* Update password
* Update the admin display pic
* Update the terms and conditions (for User and Garage users)
* Please suggest other Settings options

**Admin user – Reports \*\***

This page is not present in the wireframes but I think should be present in the backend. It will be where the admin will deal with reports about users, garage profiles and feedback disputes.

**Admin user – Admin Enquiries \*\***

This page is not present in the wireframes but I think should be present in the backend. You may advice that it joins with the “Report” page.

The page will be where the admin will handle normal enquiries from users and garage users.

NOTE: Please discuss your thoughts and make suggestions

**Garage user – Profile and Ads**

N.B. The profile of the garage is the ad. A garage cannot have multiple ads (which the wireframe suggests). I think it would be better to join them into one.

How it works is that the garage user will update his garage profile details. Once all the details have been filled in, he will be able to preview the profile and then submit it. The status of the profile will be pending until either approved or rejected for revision by the admin team. If approved, the garage user will be able to view daily or weekly statistics of the profile (such as views) and promote the garage profile. Promoting the garage profile will allow the profile to be featured at the top of the list for the location (referring to a district, and not a town) in which the garage is located.

Garage profile ads that have been edited in any way must be resubmitted and approved by admin first before going live. If a profile ad has been previously approved and not been edited, the garage user can simply pay for promotion for the profile ad to automatically be promoted for the duration paid for.

The garage user can also suspend the garage profile from the network. The garage user can also delete his account (which is an option that would maybe be better in Settings).

**Garage user – Bookings**

The garage user can view all active/current bookings, negotiate enquiries and agree on new bookings + estimate price (not shown in the wireframes), and can view all the history of all bookings (again not shown in the wireframes).

**Garage user – Settings**

This page is where the garage user can:

* Update the email login address
* Update password
* Update display pic
* Please suggest other Settings options

**Modules**

Login

* Save login credentials (for Garage User)
* Reset password module

Email

* Automated “click on link” for verification purposes
* Threads created and sent to email when enquiry is sent to admin (much like how Fiverr customer service works). Threads are closed by admin.

Payment

* Automated payment service (PayPal)
* Users that wish to pay for the premium service must do so on our website (and not on the app to avoid fees). Once the user confirms login credentials, he can then pay for the premium service. Once payment is approved, the user will become premium automatically on the app.

**The Front End**

The Driva website will need a front end website and I would like a simple template such as:

<http://templates.aucreative.co/snac/>

This type of theme introduces users to the app and tries to sell the app to the user. The one page effect of the theme allows information to be concise (which I would like to keep). However, I think there should be links in this one page style that allows the user to navigate to other sections of the website that will not have this one page style (for example, Registering (garage user), Login (garage user), Payments (upgrade for app user), Informative pages, etc…).

**What you will need to begin**

1. A Freematics OBD-II emulator will be required for testing.  
   <https://freematics.com/store/index.php?route=product/product&product_id=71&search=freematics&page=2>

I will provide you this if need be.

1. A bluetooth 4.0 version OBD-II scan dongle hardware that will connect to vehicles.  
   <https://www.ebay.co.uk/itm/Viecar-4-0-Bluetooth-v4-0-OBD2-Car-Diagnostics-Scanner-For-Apple-iOS-Android-PC/263336819319?epid=630663454&hash=item3d5018fa77:g:jAwAAOSwy~BaN88W>  
     
   I will provide you this if need be.
2. Research and understanding of all APIs listed.

Existing competitors

There are a few existing apps that provide similar functionalities to that proposed in this project specification. They provide the important “Diagnostic” functionality (which is one of the selling points of this app). The overall quality (end product) of the Driva app must be on par with these apps listed:

1. <https://automatic.com/>
2. <https://caristaapp.com/>
3. <https://www.obdautodoctor.com/>