**PROJECT DESCRIPTION OVERVIEW**

Our project is a web-platform marketplace for users to trade vehicles with other users. The project is to build a website (using something like Ruby on Rails) with a back-end database sufficient for users to search for exactly what they want and – through personally contacting them - organize a swap and transferal of registration with each other.

Ideally, this would be a suitable alternative to listing their cars on GumTree or CarSales and simply noting that they are “interested in swaps for *x, y* and *z*” – in their description. Instead, they would create their listing using elements from CarSales (ticking boxes and detailing modifications and other minor details) and attaching their preferred makes and models – with the ability to search **by** these details.

The outcome of this project would be a completely new platform tailored to motoring enthusiasts, who don’t have the opportunity to quickly and *easily* find a suitable swap for their cars.

The motivations for this project are simple – there is no platform in the market which specifically facilitates swaps of motor vehicles. If you were to look around any of the multiple online car communities (FaceBook pages i.e BoosTed, Melbourne Honda Club) almost EVERY second car sale advert mentions an interest in swaps – be it as an added alternative to a set sale price or **only** interested in swaps. Current IT trends show that online marketplaces are getting narrower in scope and their targeted demographics – serving a niche and local userbase. Building a project like this, shows a future employer that one is capable of building a back-end/front-end website and code in Ruby on Rails which is an employability-boosting skill.

There are few similar products available – the closest is GumTree. Only in creating listing can you ever specify that you would like a swap, in which the sale price is replaced with the text “Swap/Trade” – further necessitating a specification of what one would like in the description, which requires being on the listing’s page to begin with. The main point of difference with the proposed ‘CarTrade’ is that it is allowing the users to search for exactly what they want, returning only the vehicles that match potential swaps with what they currently own – and leaving all of the finalizations up to the users contacting each other.

**DETAILED DESCRITION**

**Aims**

Provide a robust platform for people to swap their cars.

We plan to achieve this by creating an online marketplace where users can sign up and create listings where they will accurately describe the characteristics of their cars with an intuitive template. Moving on, a user will then be able to search for a car that they in-turn wish to swap for – where a list of vehicles will be returned, looking for the initial user’s car for swap and offering that same initial user’s choice of car. Because of the scale of this project, the aim’s goals are broken down into functionality that will be implemented along the way.

Set up the project’s code base.

The first goal is to implement the main architecture for the website. In the case of this website project we are planning to use Ruby on Rails – so using that to set up the back-end characteristics (views/models/controllers) is incredibly simple, utilizing scripts that get the project up and running for you. This is obviously the first integral hump of the project, as it cannot exist without it.

Implement user database and pertaining functions.

The project will store data in two main databases – one for the users/sessions and another for the listings themselves. As most marketplaces nowadays all require accounts in order to store details and safely manage communication – it becomes a given that so would ours, not only because it is now common practice but because it also ensures a secure connection for users to communicate with one another.

Create and manage a database for storing listings.

The key feature of this marketplace is obviously the ability to create a listing and view the listings of other users. This is done through creating a database and implementing the ability (in the form of - well forms) for the user to input their cars details and any pertaining image files, in the end creating a record in a huge listing database. This is the first half of our main goal of a complete product, the next being the ability to search.

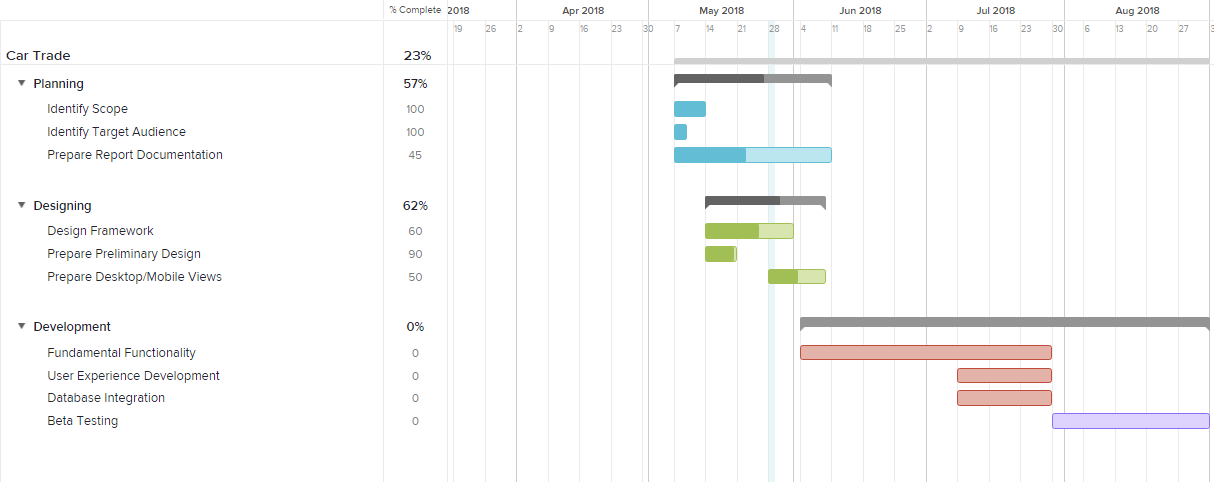
Implement listing search function.

The second half – and arguable most exciting part – of the whole process is being able to actually view all the listings with shared swap interests. This would mean creating a function in which the user’s search terms are queried on the main listing database and returns all results pertaining to those terms. This is obviously a **major** goal in the project, and is expected to be a part of the finished product.

Give users the power to communicate **through** the marketplace.

The last detail of our project is to implement a communication protocol via email, similarly to how GumTree has their own interface for messaging between users – but **also** sends copies of the messages to the users. This is a final goal and would be considered by the team as the finishing touch.

**Plans and Progress**

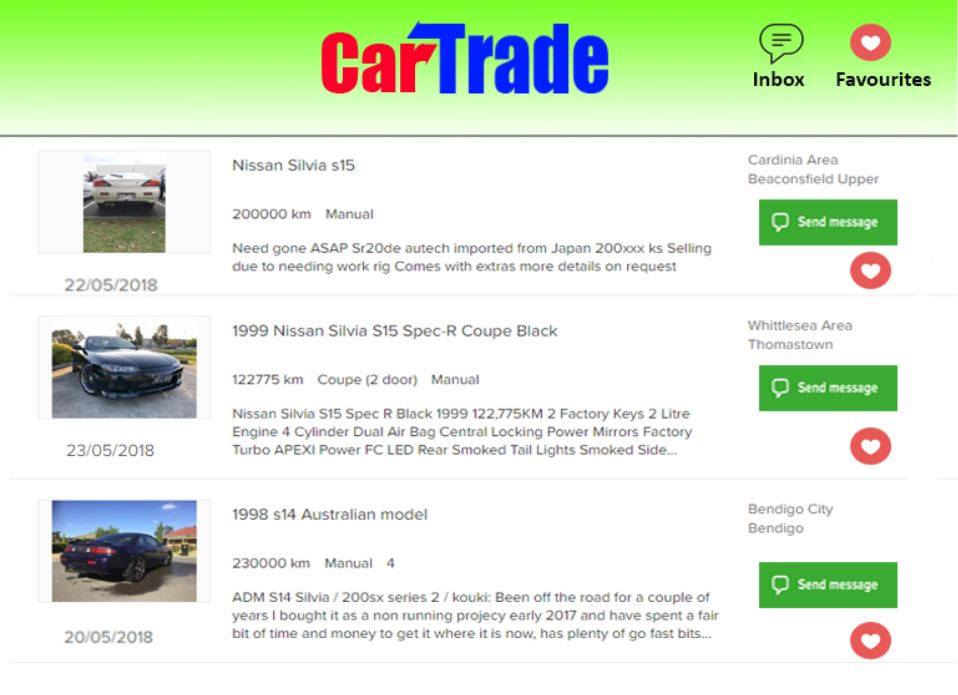


With the timeline of our project we were theoretically looking to start the development next week, intersecting the first week with the completion of our report.  As you can see, the report was a little behind, however we were somewhat ahead of schedule with designs - drawing up a rough idea of how our project will present itself. Currently, we are finishing our report and getting it ready for submission, while development has taken a halt – thanks to exams.

We have however, finished mock-ups for a couple of different views:

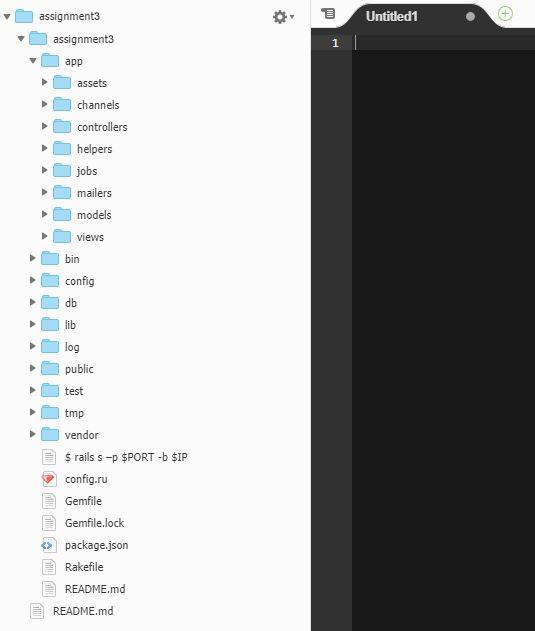


As you can see this first one is a rough idea of how a logged-in user would create their listing - taking key details such as the make and model as well as the transmission and number of kilometers the car has done along with a few photos. Ideally, the bottom half of the page would only reveal itself when the first forms are filled in, drop-down box selections are made and at least one photo is uploaded.



Here is a simple mock-up for what searching for swaps would look like. In the listing mock-up you would see that our example user is looking for a Nissan Silvia (with several constraints) - and this is displaying all Nissan Silvias available, looking for a swap with the initial user’s listing’s Make, Model and other details that the returned listings’ users were looking for as well.

Thus far, only the background architecture is in place for the website itself – using several Ruby on Rails scripts that are already in place. As you can see, this is our current file structure, which is typically self-explanatory but in this case it serves as significant evidence of an involved process and some kind of progress following the initial idea and the conception of its design:



**Roles**

Marcel – *Information Architect*

* The information architect structures the way content is organized across the site, and plans the navigation/front-end experience.

Dhrumil *– UI Designer*

* The UI Designer plans and implements the visual graphics used throughout the website.

Omar – *Web Programmer*

Joseph – *Web Programmer*

* Both Web Programmers are in charge of implementing and configuring the back-end first and foremost, as well as implement the designs and plans brought by both UI Designer and Information Architect.

After the early stages of the development cycle, both UI Designer and Information Architect blend into a more hands-on programming role and instead aid the other two who were initially the only programmers on the team.

**Scope and Limits**

Scope

|  |  |
| --- | --- |
| INCLUDED | NOT INCLUDED |
| Create an account/log-in and view responses to listing. | Set a price or offer cash in exchange for cars. |
| Create a new listing. | See user’s locations or phone numbers. |
| View listings with mutual swaps. |  |
| Contact users with active listings. |  |
| View related listings. |  |

The key limitation in our project is the time aspect. Unfortunately, a little more than 16 weeks is necessary when it comes to building a project as robust and ambitious as our own – for our web developers to learn Ruby on Rails and effectively create a functioning back-end database could potentially take months. Apart from that, all the technology and knowledge required to build this kind of product is out there and *easily* accessible.

At the time of writing this report, it seems that very little in the way of actual development will be completed – if anything. Thus, we’re focusing on making sure the concept comes across as a feasible idea and the process itself is incredibly easy to jump into and get started.

**Tools and Technologies**

The main software we are using for this project is Cloud9 and GitHub. Cloud9 is an IDE that supports Ruby on Rails **and** live collaboration – which is very useful for a group of beginners to start this project and help each other out as you go. GitHub as well will be used to keep tabs on our progress and backups of the multiple progressive builds that we finish.

**Timeframe**

|  |  |
| --- | --- |
| **Week1** | **EVERYBODY: Discuss which project we’ll do** |
| **Week2** | **EVERYBODY: Identify Requirements** |
| **Week3** | **EVERYBODY: Identify roles we’ll take and smaller details** |
| **Week4** | **EVERYBODY: Translate discussions and identified needs into report sections** |
| **Week5** | **Dhrumil/Marcel: Design Interface/Mock-ups**  **Omar/Joseph: Start Presentation** |
| **Week6** | **EVERYBODY: Complete and compile sections for presentation** |
| **Week7** | **Marcel: Design Database**  **Dhrumil: Design Interface**  **Omar/Joseph: Build architecture** |
| **Week8** | **Marcel/Omar/Joseph: Develop User/Session portion of code**  **Dhrumil: Design forms and layout** |
| **Week9** | **EVERYBODY: Continue user/session handling** |
| **Week10** | **EVERYBODY: Continue user/session handling** |
| **Week11** | **EVERYBODY: Finish user/session handling** |
| **Week12** | **Marcel: Create/Manage listing database**  **Omar/Dhrumil/Joseph: Work on listing forms** |
| **Week13** | **Marcel: Finish/Manage listing database**  **Omar/Dhrumil/Joseph: Continue listing section** |
| **Week14** | **EVERYBODY: Finish listing functionality** |
| **Week15** | **EVERYBODY: Start search listing functionality** |
| **Week16** | **EVERYBODY: Continue search listing functionality** |

**Risks**

The main risk associated with our chosen project idea is that we will run out of time. As previously stated, this is an ambitious project for a group of beginner developers and it is unlikely that we will follow our timeframe closely – expecting that mostly the early processes will take much longer than just a couple of weeks. Another risk is the choice of language may be a little too complex and we as a group may have our hands full when it comes to learning Ruby on Rails.

**Group processes and communications**

We will be using Slack to communicate between one another separately *and* as a whole group. It accommodates for people to speak in small groups - such as when certain weeks require that 3 people work together or we work in pairs. Apart from our one class together, time permits that we can meet and discuss our project during other classes and time between them.