PERSONAL PORTFOLIO FOR SPRINT 2

Alana Fitzgerald

Student Number: 8496846

Group 70

Github Link: https://github.com/rdugg4/IFB299-Group-70-S.K.R.A.M

Table of Contents

Artefact 1: UI Designs

General Description

The UI Designs are the initial static designs that guide the layout and formatting for each of the necessary pages throughout the web application. They determine aspects such as sizing, font, colour etc and serve as the guide for the developers to ensure they meet the client's expectations.

Use/Contribution

For simplicity, only one design has been included as an example however this applies to all designs.

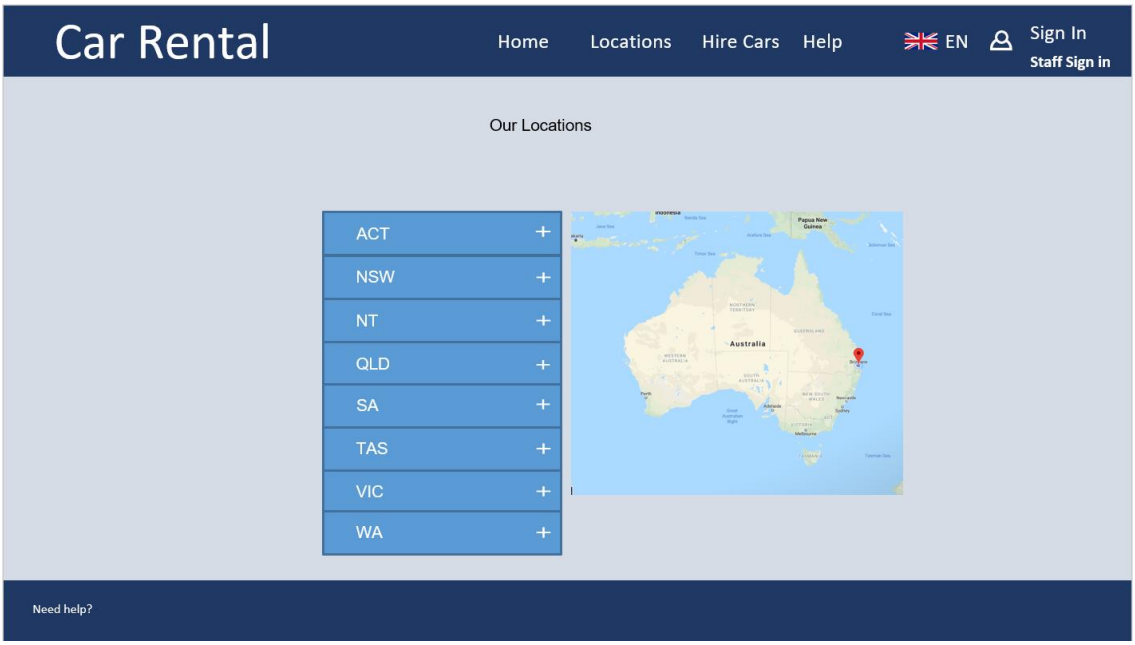

Figure 1 The initial design of the locations page

And then this is the webpage that was created based on the design:

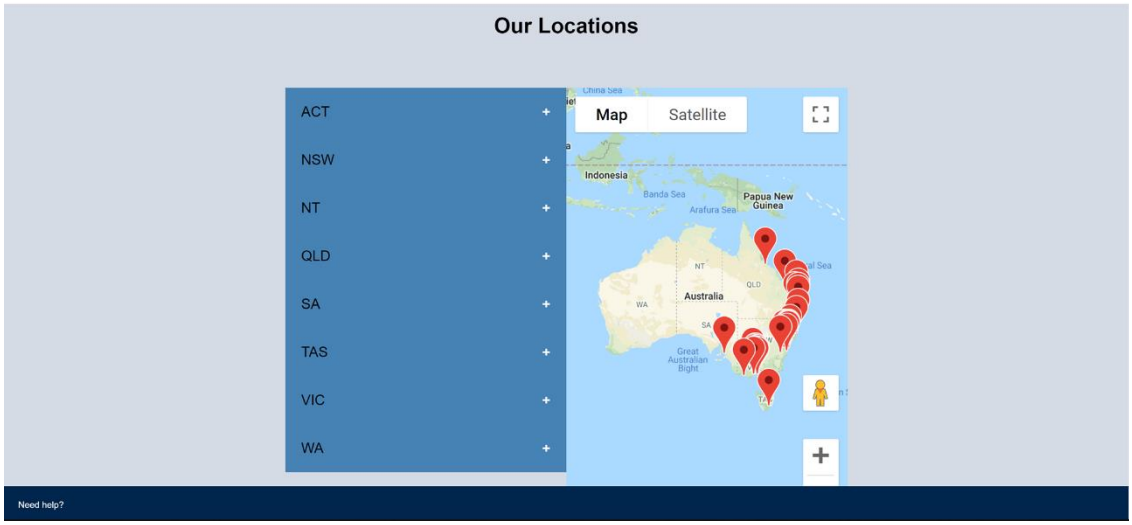

Figure 2 The final Locations page, with minor formatting changes to be consistent with the rest of the website.

Artefact 2: Create Dataset for Unit Testing

General Description

This artefact involves developing a dataset that will produce known outputs. By doing this, the results can be compared across a variety of outcomes to ensure that the system is generating the correct results.

Use/Contribution

This is a small sample of the dataset produced. Beneath this is the unit results, showing that the system passed the tests. This contributes to good coding/business practice through testing the product for accuracy and also for development to identify errors that need correcting.

```
order3 = Orders(createdate = 20181018,
    pickupdate = 20181020,
    pickupstore = store17,
    returndate = 20181103,
    returnstore = store17,
    droppedoff = 'FALSE',
    customerid = 11015,
    carid = 14838)
order3.save()

order4 = Orders(createdate = 20180101,
    pickupdate = 20180101,
    pickupstore = store3,
    returndate = 20180901,
    returnstore = store12,
    droppedoff = 'TRUE',
    customerid = 11051,
    carid = 14871)
order4.save()
```

```
order5 = Orders(createdate = 20180201,
    pickupdate = 20180214,
    pickupstore = store21,
    returndate = 20180217,
    returnstore = store22,
    droppedoff = 'TRUE',
    customerid = 11313,
    carid = 15127)
order5.save()

order6 = Orders(createdate = 20180217,
    pickupdate = 20180221,
    pickupstore = store40,
    returndate = 20180228,
    returnstore = store40,
    droppedoff = 'TRUE',
    customerid = 15199,
    carid = 14904)
order6.save()
```

Figure 3 Creating order codes to test if the code produces known outputs

```
.....................................................................................
----------------------------------------------------------------------
Ran 89 tests in 18.733s

OK
```

Figure 4 Tests proved to be successful. More detailed data found in SKRAM70/TestApp/Datasets

Artefact 3: General Coding

General Description

This artefact refers to the main HTML aspect of the website. The pages were built in order to serve as the front-end for all of the customer and staff interaction.

Use/Contribution

These pages have been used as the front end of the website. The CS students then worked on implementing all of this front end development with the back end to create queries, inputs and interact with the database etc.

```
<button class="accordion"> SA </button>
<div class="panel">
  <br>
  <p> 213 Valencia Place, <br>
    Cloverdale SA </p>
  <br>
  <p> 9111 Rose Ann Ave, <br>
    Findon SA </p>
  <br>
  <p> 6385 Mark Twain, <br>
    Perth SA </p>
  <br>
</div>

<button class="accordion"> TAS </button>
<div class="panel">
  <br>
  <p> 636 Vine Hill Way, <br>
    Hobart TAS 7005 </p>
  <br>
</div>

<button class="accordion"> VIC </button>
<div class="panel">
  <br>
  <p> 6465 Detroit Ave., <br>
    Bendigo, VIC  </p>
  <br>
  <p> 626 Bentley Street, <br>
    Cranbourne, VIC  </p>
  <br>
  <p> 5927 Rainbow Dr, <br>
    Geelong, VIC </p>
  <br>
  <p> 5167 Condor Place, <br>
    Melbourne, VIC  </p>
  <br>
  <p> 1873 Mt. Whitney Dr, <br>
    Melton, VIC  </p>
```

Figure 5 Creating the expanding buttons that show locations for each state

Refer to Figure 1 for the display result of this piece of code. Full code can be found on the AlanaLocations.html file on Github.

Artefact 4: Acceptance Criteria Checks

General Description

As Scrum Master, during our meetings, the minutes were recorded for each week and then categorised into three categories (see image below). This information was used to keep the team on track and document and issues or necessary changes.

Use/Contribution

Whilst the results are not technically tangible, evidence that this documentation helped is in the submissions. For example, by noting down the tutor's instructions in the to do section, there is a clear documented outline to guide the team in developing the presentation. Another more obvious example is in the 'things to change'. Whilst the original code has been altered, changes can be noted in Github. All these notes were implemented in the tasks for each following week so many of the refinements on the application have been supplemented by this documentation.

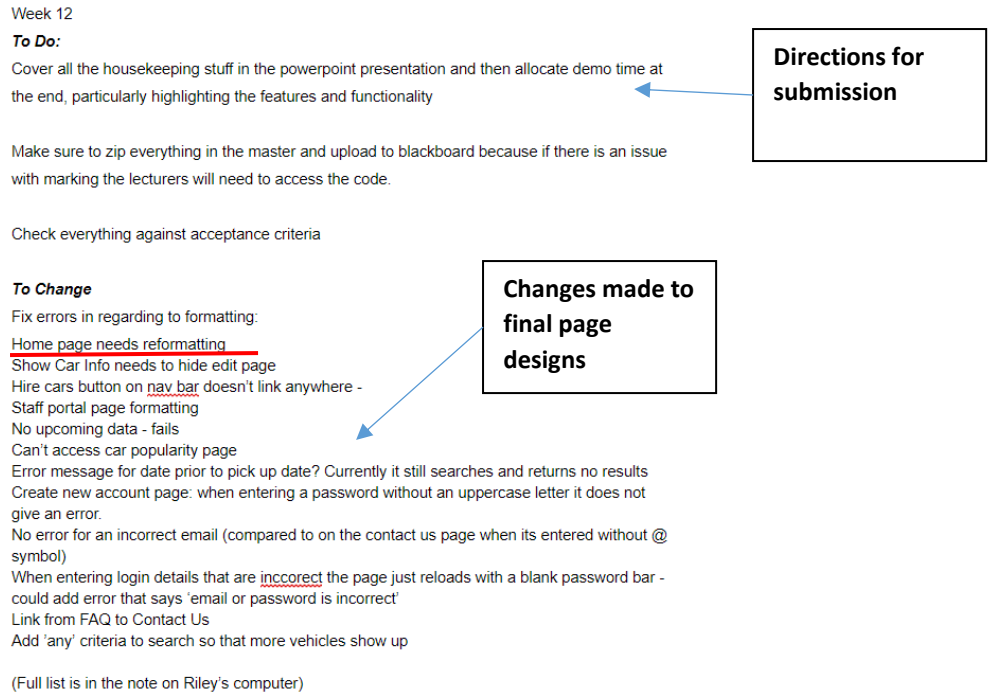

Figure 5 Notes taken during the meeting

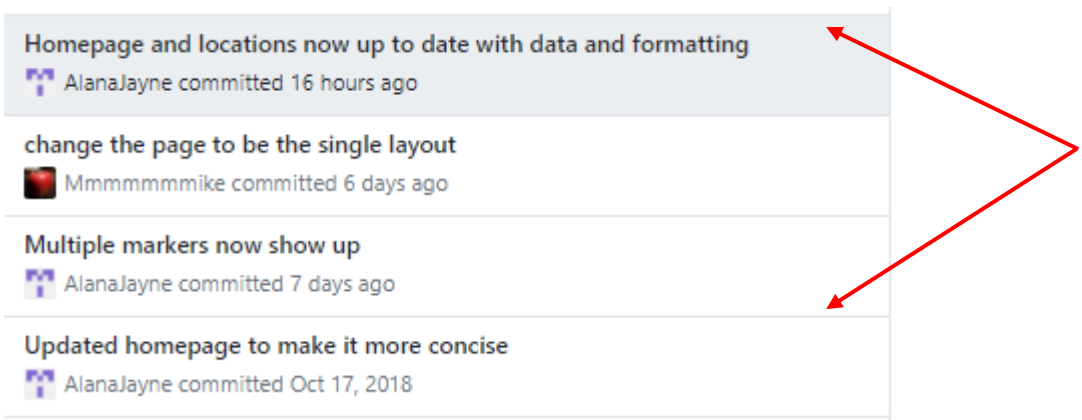

Figure 6 Results of notes being made into tasks

Artefact 5: Google API and Javascript Coding

General Description

This piece of code uses Javascript, which slightly more advanced than the previous html and css in terms of skills. An API key was created so that Google Maps could be used on the page, it is located just before call back in the script src. The code also uses lists, loops and creates a function, which again required a little more development than the standard html that was done previously.

```
<script>
// This is the function that implements the google map and adds markers to it
function myMap() {
   var locations = [
     ['Wollongong', -34.425135, 150.894603, 4],
     ['Sydney', -33.918416, 151.225770, 5],
     ['Melbourne', -37.831699, 144.964333, 3],
     ['Brisbane', -27.477376, 153.028169, 2],
     ['Adelaide', -34.924787, 138.595701, 1]
   ];

// This part loads the google map and sets the default location
   var map = new google.maps.Map(document.getElementById('map'), {
      zoom: 10,
      center: new google.maps.LatLng(-27.47, 153.02),
    });

   var infowindow = new google.maps.InfoWindow();

   var marker, i;

// This pulls from the list of locations above to know where to place the markers
   for (i = 0; i < locations.length; i++) {
     marker = new google.maps.Marker({
       position: new google.maps.LatLng(locations[i][1], locations[i][2]),
       map: map
     });

     google.maps.event.addListener(marker, 'click', (function(marker, i) {
       return function() {
         infowindow.setContent(locations[i][0]);
         infowindow.open(map, marker);
       }
     })(marker, i));
   }
 }
 </script>

         <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAVArWOV0RwGEHyQPfSOFJiUsgs5E5Q_yw&callback=myMap">
         </script>
      </div>
    </div>
```

Figure 6 Script that loads the Google Maps interface, along with markers for each location

Use/Contribution

This has been implemented on the final locations page. As demonstrated in the screen capture, the map successfully loads and the markers are visible in the correct locations.

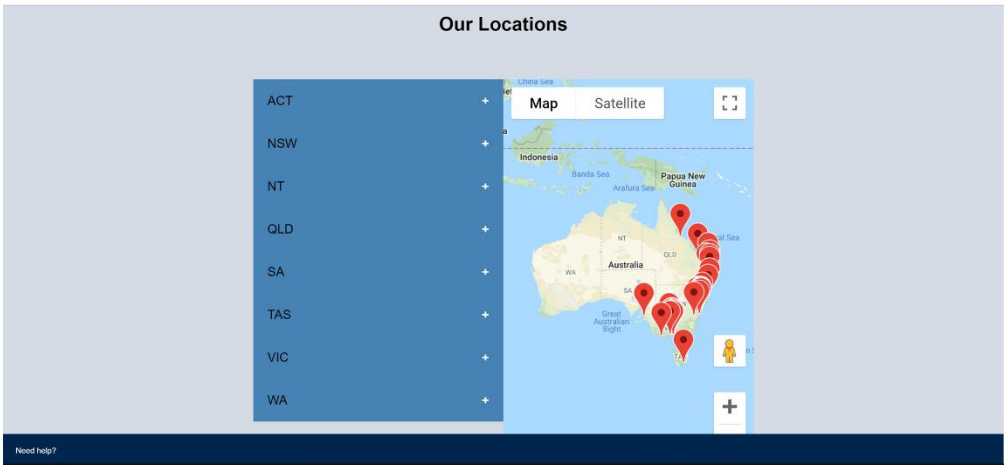

Figure 7 Locations page again, showing the markers over the map