Sprint 2 – Personal Portfolio

Wenona Cramp-Church

Due Date: 26th October 2018

Tutor: Prakash Bhandari

Group Number: 92

GitHub: https://github.com/taylorbindon/IFB299.git

Declaration

By submitting this assignment, I am aware of the University rule that a student must not act in a manner which constitutes academic dishonesty as stated and explained in the QUT Manual of Policies and Procedures. I confirm that this work represents our efforts. I have viewed the final version and declare that it does not contain plagiarized material.

Full Name	Student No.	Signature
Wenona Cramp-Church	n9738436	W.C. Church

Table of Contents

Personal Portfolio

Artefact 1 – Study of Estimation Techniques

Artefact description

This artefact contains the techniques and analysis used to estimate the work breakdown, project estimation and deadlines of the project. Techniques used include a velocity re-calculation and time estimation.

Contribution to the project/how it was used

This analysis was used to ensure that the team adhered to the project goals and that the user stories required for Sprint 2 were delivered in the final product. Early in Sprint 1, it was clear that the team had underestimated how long it would take to connect the database. At the end of Sprint 1, the webpage had limited functionality. The group had completed specific tasks, but no user stories were 'fully' completed. Therefore, the group finalised stories 1-4 and 7 in Sprint 2. The adjusted velocity, and total hours taken to complete the tasks were calculated and submitted in the final release.

Screen capture of where it's used

Calculated Velocity

The team consists of five members. The sprint lasts for a 5-week period (including the mid-semester break). Estimated approximately 1.5 days per week per team member
= 5 (5 x 1.44)
= 36 days spent on the task.

Estimate that approximately 1/2 of each working day is spent doing this task
= 36 x (1/2)
= 18 actual days per sprint.

Assume one story point takes 1 day to complete.

Team velocity is 18 story points/sprint.

NT: if one story takes one day to complete, assume 1 day = 8 hours.

Time spent completing the task
= 1/2 of each working day
= 4 hours.

Therefore, it takes 4 hours to complete one user point.

Sprint 2 Velocity

Sprint two included user stories 1-7, worth 18 user points. The total time to complete Sprint 2

= number of hours to complete one user story point * number of user story points in Sprint 2

= 4 * 18

= 72 hours

Overall Sprint 2 took 72 hours to complete.

Sprint 2 Time Estimation

The hours spent completing each task were also calculated (as seen in the final Release and Sprint Plan).

Total Story Points: 18 Total Hours: 72

Current Velocity: 18 story points

Story 01: Employee Login

Task ID	Task Description	Estimate	Taken
T01	Create employee login	2	Completed in Sprint 1
T02	Verify login is valid with database	3	6
T03	Verify story is complete (acceptance test)	1	2
	Story Points: 2 Total Hours:	6	8

Story 02: Vehicle Recommendations

Task ID	Task Description	Estimate	Taken
T04	Create database	4	Completed in Sprint 1
T05	Create main page	2	Completed in Sprint 1
T06	Create employee dashboard	2	Completed in Sprint 1
T07	Create search text field	1	Completed in Sprint 1
T08	Return results on new page	1	6
T09	Create filters on results page	1	7
T10	Verify story is complete (acceptance test)	1	3
	Story Points: 4 Total Hours:	12	16

Story 03: Generate Reports

Task ID	Task Description		Estimate	Taken
T11	Create board member dashboard		3	Completed in Sprint 1
T12	Implement dashboard functionality (Reports, charts, tables)		4	7
T13	Return results on new page		2	3
T14	Create filters on results page		2	5
T15	Verify story is complete (acceptance test)		1	1
	Story Points: 4	Total Hours:	12	16

Story 04: Customer Vehicle Search

Task ID	Task Description		Estimate	Taken
T16	Create booking search on main page		2	4
T17	Create date, location, and vehicle preference fields		2	5
T18	Return results on new page		1	5
T19	Verify story is complete (acceptance test)		1	1
	Story Points: 4	Total Hours:	6	16

Story 07: Employee Customer Search

Task ID	Task Description		Estimate	Taken
T29	Create employee search page		1	5
T30	Return results on table in webpage		2	8
T31	Verify story if complete (acceptance test)		1	3
	Story Points: 4	Total Hours:	4	16

Artefact 2 – Written Test Cases

Artefact description

This artefact includes the written test cases document required by the Week 12 Workshop. “A Test Case is a set of actions executed to verify a particular feature or functionality of your software application” (Guru99, 2018).

Contribution to the project/how it was used

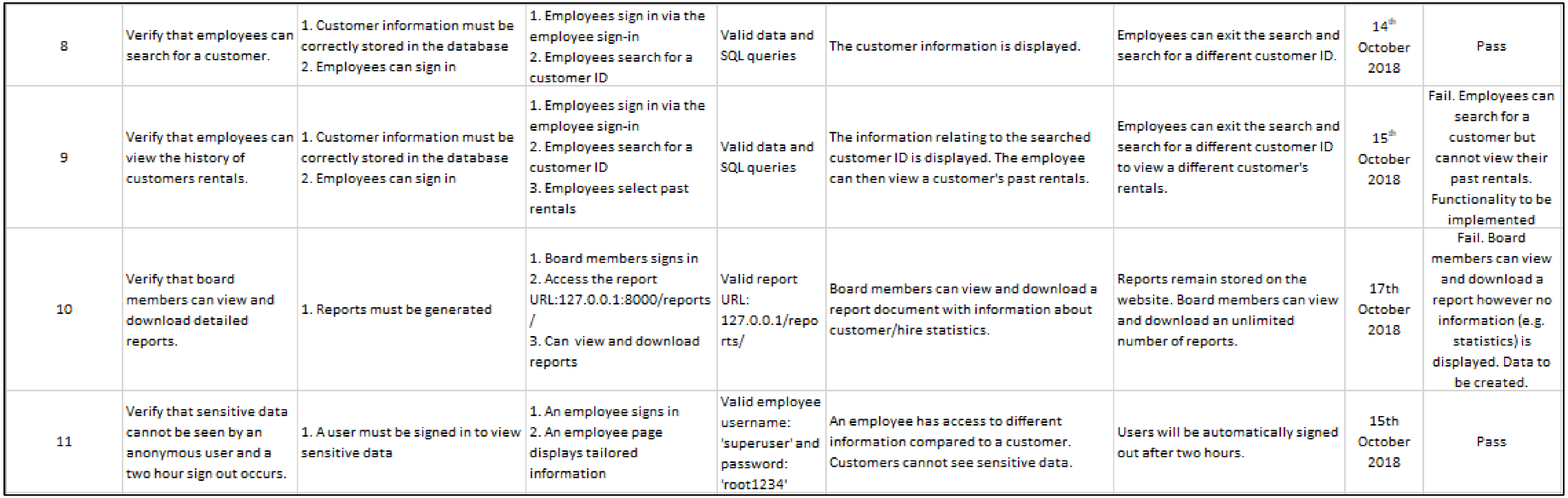
The testing of functionality allowed each team member to understand the progress of the project. This document was used to assist the client in understanding what functionality would be implemented in the final product. The success of each test case was used to improve the project estimation, prioritise and re-allocate tasks, and re-assess deadlines for specific user stories.

Screen capture of where it’s used

See next page.

Written Test Cases

Serial Number	Test Case Description	Entry Crtieria (Precondition)	Steps	Test Data	Expected Result/Acceptance Criteria	Exit Criteria (post-condition)	Test Date	Test Pass/Fail
1	View the CRC homepage through a URL to access the site.	1. The Django server must be set up on the web machine 2. Web browsers should be installed on the test machine	1. Run the CRC project on the local instance of the Django server 2. Start any web browser 3. Enter the URL: in the address bar of the web server	A valid URL: 127.0.0.1:8000 /home/	The user can access the CRC homepage. The navigation bar that appears at the top is as follows: 1. CRC logo (homepage) 2. About 3. Contact 4. FAQ 5. SignIn 	Users should be able to navigate to different webpages.	1st October 2018	Pass
2	Verify that all tabs link to all web pages.	1. All pages must be created using HTML/CSS.	1. Enter the URL: in the address bar on the web server 2. Select any tab from the navigation bar	Valid webpages/URL s	A user can access each of the pages from the navigation bar. The information displayed on each page should change.	Users can select any page an unlimited number of times.	4th October 2018	Pass
3	Verify that tabs are underlined when the mouse hovers over the button.	1. All tabs must be created 2. All the necessary code to incorporate functionality is written	1. Access any page 2. Hover the mouse over a navigation heading 3. The heading becomes underlined	Valid HTML/CSS	A user sees an underline below the heading and is aware which page they will be taken to if they 'Click'.	When the user hovers away from the navigation heading, the underline is removed. If a user selects an underlined heading they will be taken to that webpage.	4th October 2018	Pass
4	Verify that the correct information is displayed on each page.	1. Textboxes/divs must be created for each webpage	1. A user accesses any webpage 2. Relevant information is displayed	Valid HTML/CSS and text	A user can view relevant information on each page which allows them to perform a task.	When a user accesses a different page, the information displayed will change.	1st October 2018	Pass
5	Verify that employees/ customers can log into their account.	1. The sign in page has been created for different role access 2. Example usernames and passwords have been created	1. Select the sign-in tab 2. Select either customer or employee 3. Enter the correct username and password 4. 'Click' the sign-in button	Valid employee username: 'superuser' and password: 'root1234'	Employees/customers can sign in using a created username and password.	Users can sign out of their account.	8th October 2018	Pass (customer username and posword to be created).
6	Verify that employees have specific role access.	1. There must be data for an employee/ customer to test 2. The webpage must have separate tabs created 3. Users can log in	1. A user accesses the sign-in page 2. The user then has specifc information displayed pertaining to their role	Valid HTML/CSS and text for different users	When a customer signs in they can view their previous rentals. When an employee signs in they can view customer information. When a board member signs in, they can view past reports.	Users can log out and log back in and unlimited number of times.	12th October 2018	Pass (data to be created).
7	Verify that customers can search for vehicles by specifying certain criteria (e.g. dates).	1. All searches must query correct information from the database.	1. Customer accesses the customer home page 2. Customer makes selections according to preferred date, location and pick up time	Valid data and SQL queries	The vehicles meeting those requirements are displayed.	The customer can arrange the vehicles displayed according to the headings (model etc). The customer can perform an unlimited number of searches.	15th October 2018	Pass



8	Verify that employees can search for a customer.	1. Customer information must be correctly stored in the database 2. Employees can sign in	1. Employees sign in via the employee sign-in 2. Employees search for a customer ID	Valid data and SQL queries	The customer information is displayed.	Employees can exit the search and search for a different customer ID.	14th October 2018	Pass
9	Verify that employees can view the history of customers rentals.	1. Customer information must be correctly stored in the database 2. Employees can sign in	1. Employees sign in via the employee sign-in 2. Employees search for a customer ID 3. Employees select past rentals	Valid data and SQL queries	The information relating to the searched customer ID is displayed. The employee can then view a customer's past rentals.	Employees can exit the search and search for a different customer ID to view a different customer's rentals.	15th October 2018	Fail. Employees can search for a customer but cannot view their past rentals. Functionality to be implemented
10	Verify that board members can view and download detailed reports.	1. Reports must be generated	1. Board members signs in 2. Access the report URL:127.0.0.1:8000/reports/ 3. Can view and download reports	Valid report URL: 127.0.0.1/reports/	Board members can view and download a report document with information about customer/hire statistics.	Reports remain stored on the website. Board members can view and download an unlimited number of reports.	17th October 2018	Fail. Board members can view and download a report however no information (e.g. statistics) is displayed. Data to be created.
11	Verify that sensitive data cannot be seen by an anonymous user and a two hour sign out occurs.	1. A user must be signed in to view sensitive data	1. An employee signs in 2. An employee page displays tailored information	Valid employee username: 'superuser' and password: 'root1234'	An employee has access to different information compared to a customer. Customers cannot see sensitive data.	Users will be automatically signed out after two hours.	15th October 2018	Pass

This has been committed to GitHub: Github Commit –

taylorbindon/IFB299/Final Release/Written-Test-Cases.xlsx

Artefact 3 – HTML Page: Fourth Iteration

Artefact description

One member of the team (Bri) created mock-up graphical user interfaces (GUI's) using InDesign. In total, 10 mock-ups were created to simulate the following web pages: About, Contact, Customer Home, Customers, Employee Home, FAQ, Reports, Sign In, Stores, and Vehicles. This artefact will contain the fourth HTML iteration for the Customer Home page.

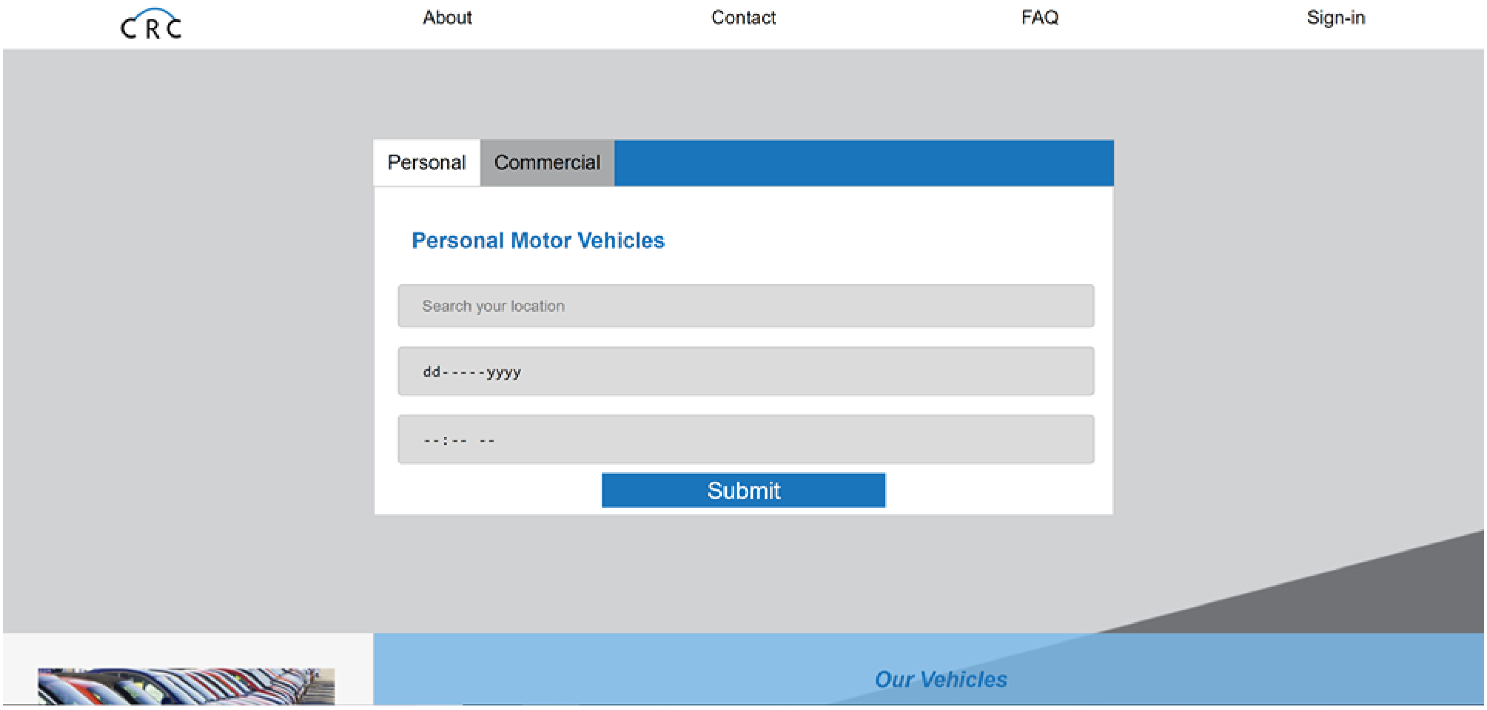
Contribution to the project/how it was used

The UI annotations were used when creating the HTML for each web page. A total of five HTML/CSS iterations were created for the Customer Home Page. This is because additional functionality was requested by the client at meetings and new tables displaying information were created. The CSS was sent to Kirsten for testing and to ensure that all CSS was connected properly to the webpage. Kirsten then uploaded the final HTML, CSS, and webpage to GitHub.

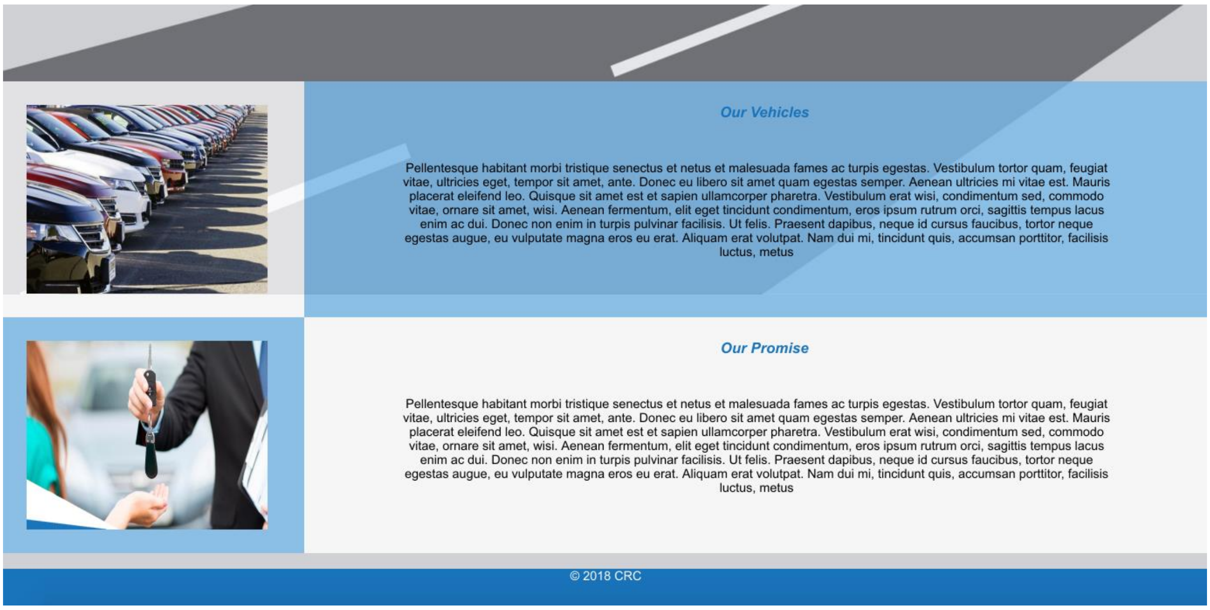
Screen capture of where it's used

The webpages can be accessed by customers and employees through the online website. The images below are part of the 4th iteration of the HTML page. This is because TJ implemented the HTML code to fit Python Django's requirements, creating a 5th iteration. Along with this, the overall design changed slightly due to how the team decided to implement the search functionality.

Fourth iteration: CSS and HTML



The above image is the top half of the customer home page.



Our Vehicles

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

Our Promise

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

© 2018 CRC

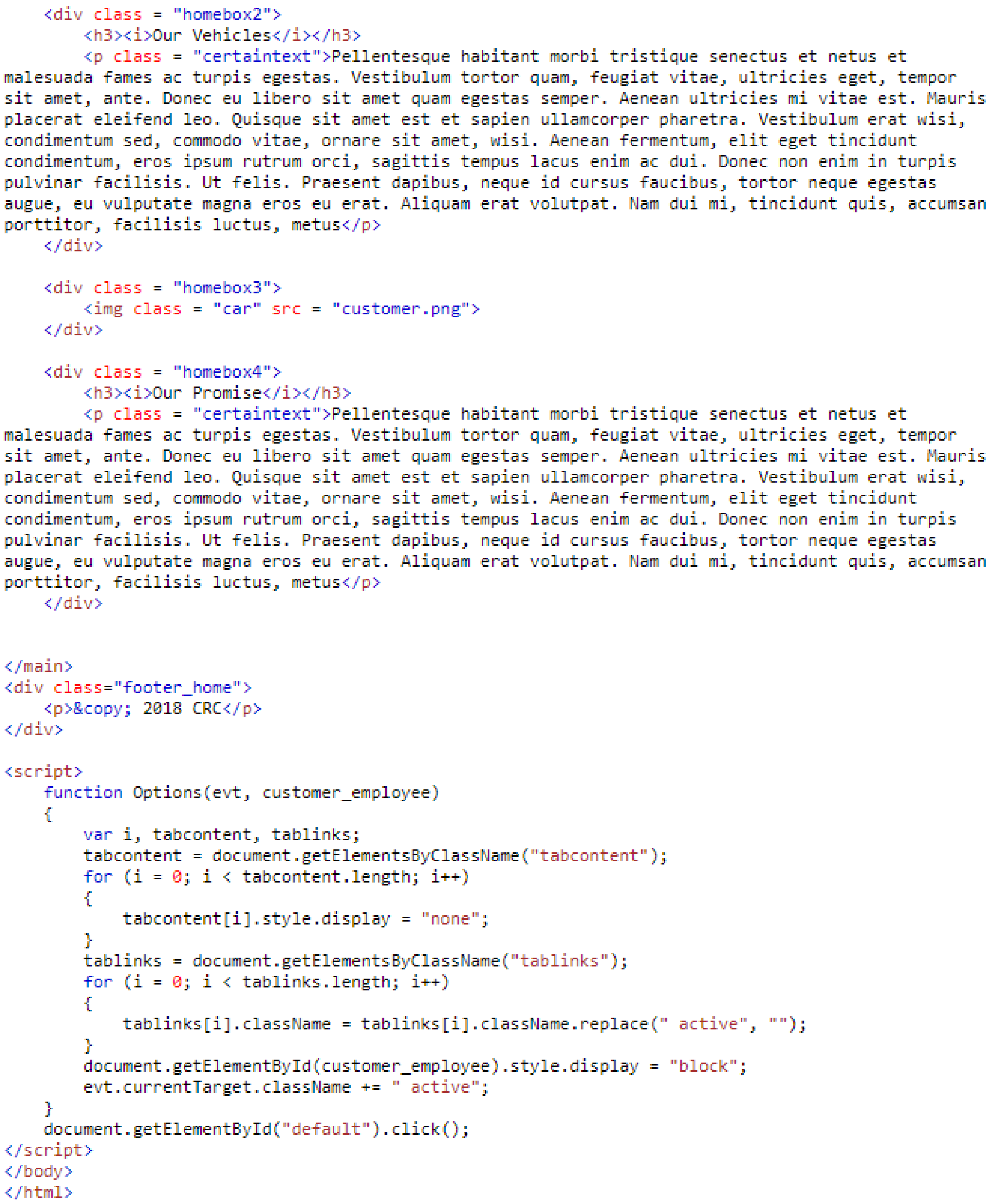
The above image is the bottom half of the customer home page (when the viewer scrolls down). The navigation bar will stay fixed at the top of the page.

The HTML and CSS for this artefact [fourth iteration] is shown below:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="stylesheet2.css">
</head>
<body>
    <nav>
        <ul>
            <li><a href =
"file:///C:/Users/20123/Documents/University/2018/Semester%202/IFB299%20IT%20Project%20Design%20and
%20Development/Assessment/Webpages/home_page.html"><img class = "logo" src = "logo.png"> </a></li>
            <li> <a href =
"file:///C:/Users/20123/Documents/University/2018/Semester%202/IFB299%20IT%20Project%20Design%20and
%20Development/Assessment/Webpages/about.html"> About </a></li>
            <li> <a href =
"file:///C:/Users/20123/Documents/University/2018/Semester%202/IFB299%20IT%20Project%20Design%20and
%20Development/Assessment/Webpages/contact.html"> Contact </a></li>
            <li> <a href =
"file:///C:/Users/20123/Documents/University/2018/Semester%202/IFB299%20IT%20Project%20Design%20and
%20Development/Assessment/Webpages/FAQ.html"> FAQ</a></li>
            <li> <a href =
"file:///C:/Users/20123/Documents/University/2018/Semester%202/IFB299%20IT%20Project%20Design%20and
%20Development/Assessment/Webpages/sign-in_final.html"> Sign-in </a></li>
        </ul>
    </nav>
<main class = "main_home">
    <div class = "tab">
        <button class = "tablinks" onclick="Options(event, 'Personal')" id="default"> Personal
</button>
        <button class = "tablinks" onclick="Options(event, 'Commercial')"> Commercial </button>
    </div>
    <div id = "Personal" class = "tabcontent">
        <h3>Personal Motor Vehicles</h3>
        <input class = "whole_length_form" type = "search" name = "Location_entered"
placeholder="Search your location">
        <input class = "half_length_form" type = "date" name = "date_entered" >
        <input class = "half_length_form" type = "time" name = "time_entered">
        <button class = "submit_search"> Submit </button>
    </div>
    <div id = "Commercial" class = "tabcontent">
        <h3>Commercial Motor Vehicles</h3>
        <input class = "whole_length_form" type = "search" name = "Location_entered"
placeholder="Search your location">
        <input class = "half_length_form" type = "date" name = "date_entered">
        <input class = "half_length_form" type = "time" name = "time_entered">
        <button class = "submit_search" type="submit"> Submit </button>

    </div>

    <div class = "homebox1">
        <img class ="car" src ="cars.png">
    </div>
```



```
    <div class = "homebox2">
        <h3><i>Our Vehicles</i></h3>
        <p class = "certaintext">Pellentesque habitant morbi tristique senectus et netus et 
malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor 
sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris 
placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, 
condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt 
condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis 
pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas 
augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan 
porttitor, facilisis luctus, metus</p>
    </div>

    <div class = "homebox3">
        <img class = "car" src = "customer.png">
    </div>

    <div class = "homebox4">
        <h3><i>Our Promise</i></h3>
        <p class = "certaintext">Pellentesque habitant morbi tristique senectus et netus et 
malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor 
sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris 
placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, 
condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt 
condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis 
pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas 
augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan 
porttitor, facilisis luctus, metus</p>
    </div>


</main>
<div class="footer_home">
    <p>&copy; 2018 CRC</p>
</div>

<script>
    function Options(evt, customer_employee)
    {
        var i, tabcontent, tablinks;
        tabcontent = document.getElementsByClassName("tabcontent");
        for (i = 0; i < tabcontent.length; i++)
        {
            tabcontent[i].style.display = "none";
        }
        tablinks = document.getElementsByClassName("tablinks");
        for (i = 0; i < tablinks.length; i++)
        {
            tablinks[i].className = tablinks[i].className.replace(" active", "");
        }
        document.getElementById(customer_employee).style.display = "block";
        evt.currentTarget.className += " active";
    }
    document.getElementById("default").click();
</script>
</body>
</html>
```

Please note: please refer to the read me document (.txt file) in the GitHub for all references used in the website.

This has been committed to GitHub:

Github Commit –

taylorbindon/IFB299/Documents/Webpages/home_page.html

Artefact 4 – Query

Artefact description

The code outlined below demonstrates the process of search functionality. The way this is done is the search query is collected ('get') in the views.py file. Then it checks if this query is valid by comparing it to the data in the database. If the code is valid it will return the database entries that match the query and send this data to the forms.py file where it separates this data into a table which will be displayed on the webpage.

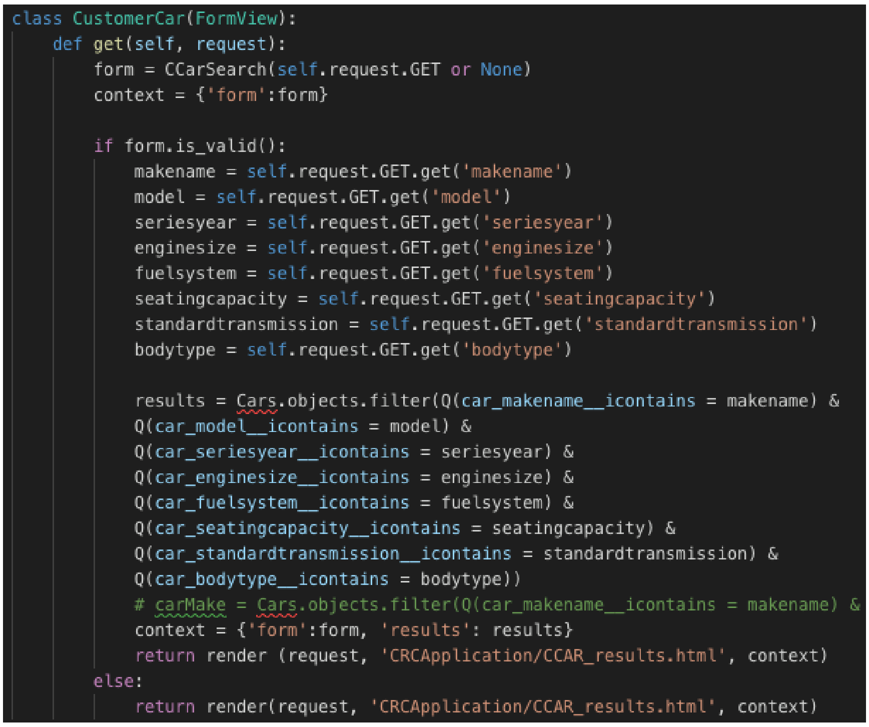
Contribution to the project/how it was used

This artefact was used to provide functionality to the webpage and meet CRC's requirements. This code enables a customer to search for a vehicle based on: make, model, year, engine size, fuel system, seating capacity, transmission and body type. If the search matches the data in the database, the relevant 'matching' vehicles will be presented in a table. This artefact was shared between myself and Kirsten.

Screen capture of where it's used

The code portions below are in separate python files in GitHub.

Code 1 filename: views.py

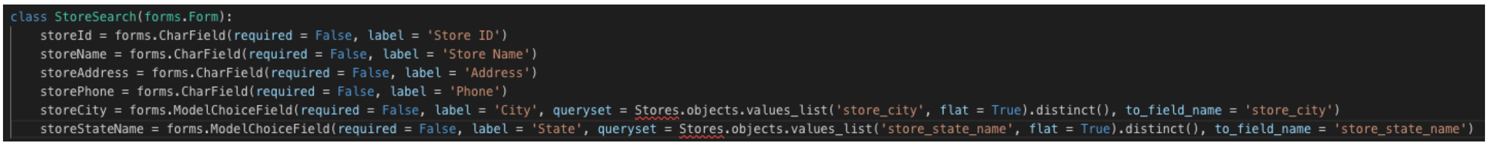


```python
class CustomerCar(FormView):
    def get(self, request):
        form = CCarSearch(self.request.GET or None)
        context = {'form':form}

        if form.is_valid():
            makename = self.request.GET.get('makename')
            model = self.request.GET.get('model')
            seriesyear = self.request.GET.get('seriesyear')
            enginesize = self.request.GET.get('enginesize')
            fuelsystem = self.request.GET.get('fuelsystem')
            seatingcapacity = self.request.GET.get('seatingcapacity')
            standardtransmission = self.request.GET.get('standardtransmission')
            bodytype = self.request.GET.get('bodytype')

            results = Cars.objects.filter(Q(car_makename__icontains = makename) &
            Q(car_model__icontains = model) &
            Q(car_seriesyear__icontains = seriesyear) &
            Q(car_enginesize__icontains = enginesize) &
            Q(car_fuelsystem__icontains = fuelsystem) &
            Q(car_seatingcapacity__icontains = seatingcapacity) &
            Q(car_standardtransmission__icontains = standardtransmission) &
            Q(car_bodytype__icontains = bodytype))
            # carMake = Cars.objects.filter(Q(car_makename__icontains = makename) &
            context = {'form':form, 'results': results}
            return render (request, 'CRCApplication/CCAR_results.html', context)
        else:
            return render(request, 'CRCApplication/CCAR_results.html', context)
```

Code 2 filename: forms.py



```python
class StoreSearch(forms.Form):
    storeId = forms.CharField(required = False, label = 'Store ID')
    storeName = forms.CharField(required = False, label = 'Store Name')
    storeAddress = forms.CharField(required = False, label = 'Address')
    storePhone = forms.CharField(required = False, label = 'Phone')
    storeCity = forms.ModelChoiceField(required = False, label = 'City', queryset = Stores.objects.values_list('store_city', flat = True).distinct(), to_field_name = 'store_city')
    storeStateName = forms.ModelChoiceField(required = False, label = 'State', queryset = Stores.objects.values_list('store_state_name', flat = True).distinct(), to_field_name = 'store_state_name')
```

Artefact 5 – Client meeting with developer

Artefact description

This artefact contains the information recorded from meetings with the client (TJ).

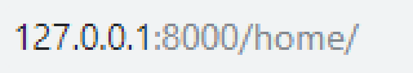
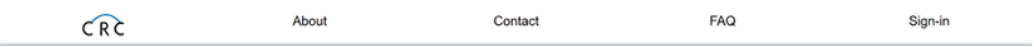
Contribution to the project/how it was used

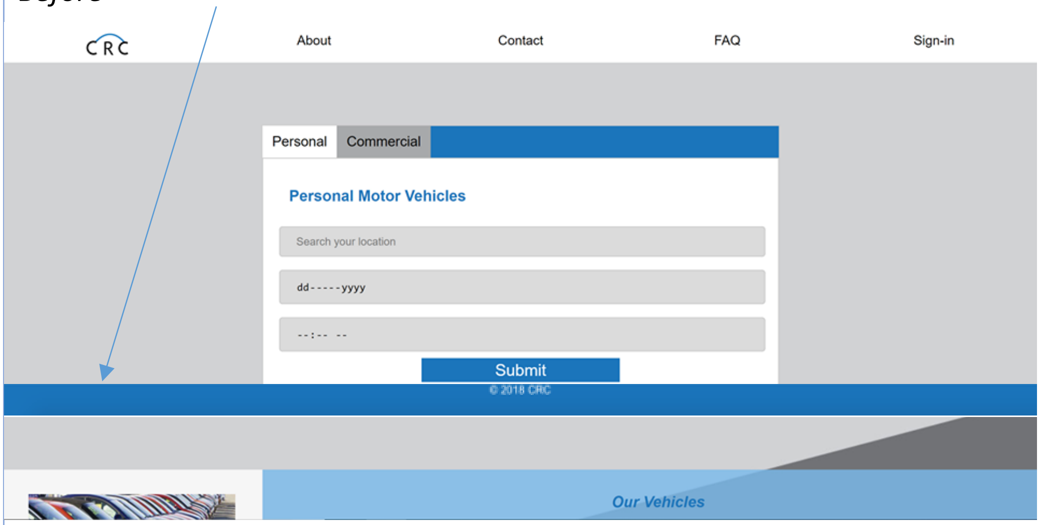
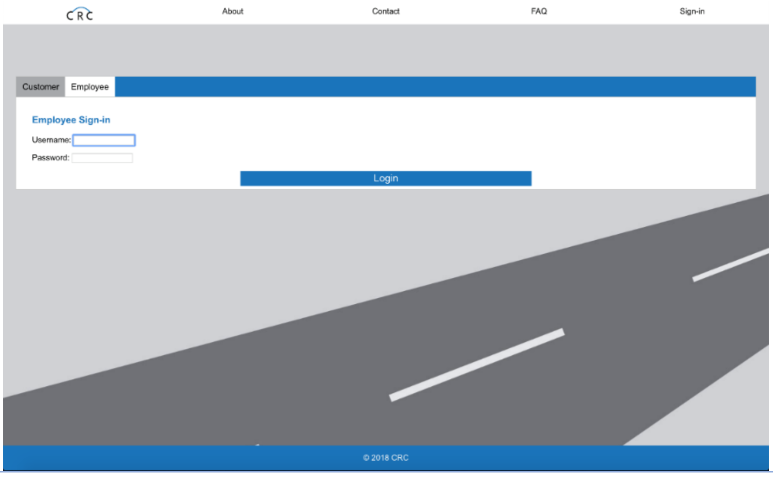
Meetings with the client have been used to gain clarification and communicate the progress of Sprint 2. This information was used to improve and finalise the design of the webpage and communicate the current functionalities of the database. For example, the client communicated which user stories were a priority, whilst I communicated the current state of the test cases (as per Artefact 2 – Written Test Cases). The team will implement the feedback provided in Sprint 2. E.g. it was decided that only the user stories 1-4 and 7 will be implemented in this project.

Screen capture of where it's used

Below is the table produced that details the discussion points/requirements from each meeting.

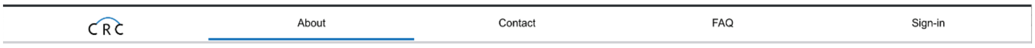
Please note: the table begins at meeting 5 as meetings 1-4 were conducted in Sprint 1.

<table>
<tr><th>Meeting #</th><th>Date</th><th>Time</th><th>Discussion Points/Requirements</th></tr>
<tr><td rowspan="2"><b>5</b></td><td>2nd October 2018</td><td>2pm – 4pm</td><td>▪ Client viewed the CRC homepage through a valid URL: 127.0.0.1:8000/home/
▪ All the navigation tabs appeared on homepage but were not linked
▪ Each page is created with the valid information, but pages are not linked
▪ Client is satisfied with the overall look of the site but requests more functionality by the next meeting (e.g. linked pages, log in)
▪ Client advised that footer needs to be altered</td></tr>
<tr><td colspan="3"><b>Evidence</b>
Homepage URL:
127.0.0.1:8000/home/
Navigation bar:
 </td></tr>
</table>

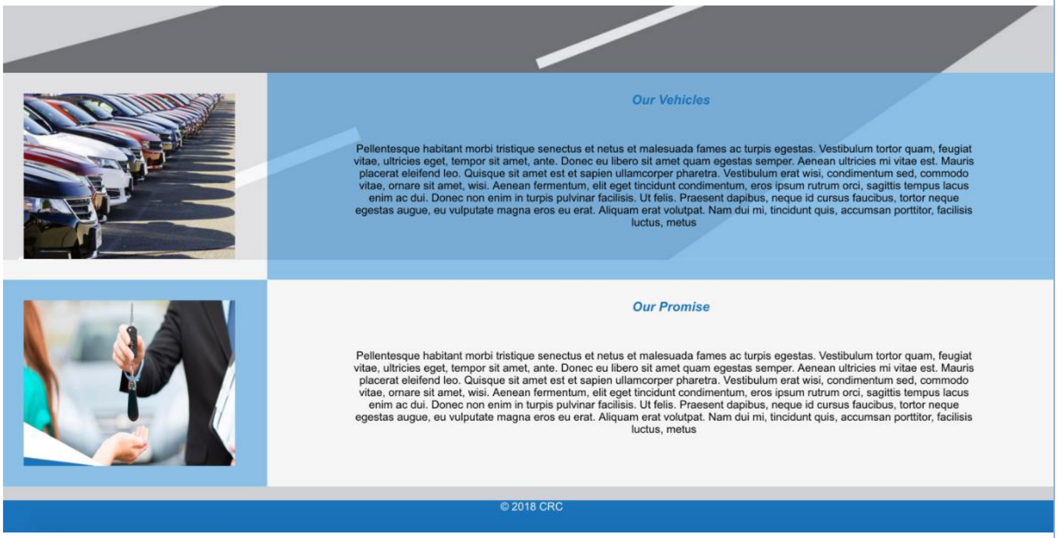
	Footer: *Before* Footer appeared in centre of page  CRC About Contact FAQ Sign-in Personal Commercial Personal Motor Vehicles Search your location dd-----yyyy --:-- -- Submit © 2018 CRC Our Vehicles		
6	12th October 2018	8am – 9:30am	▪ Employees and customers can sign in through the same page ▪ Client was provided with the current superuser username and password ▪ All pages are linked ▪ Navigation heading is underlined when user hovers over a tab ▪ Client was shown how to create and alter the username and password when the website is finalised ▪ Client satisfied that employees have specific role access and functionality ▪ The final design alterations approved (e.g. footer, table displays)
	Evidence Sign in:  CRC About Contact FAQ Sign-in Customer Employee Employee Sign-in Username: Password: Login © 2018 CRC		

Superuser username: superuser
Password: root1234

Navigation heading (underlined):



Footer (final):



Our Vehicles

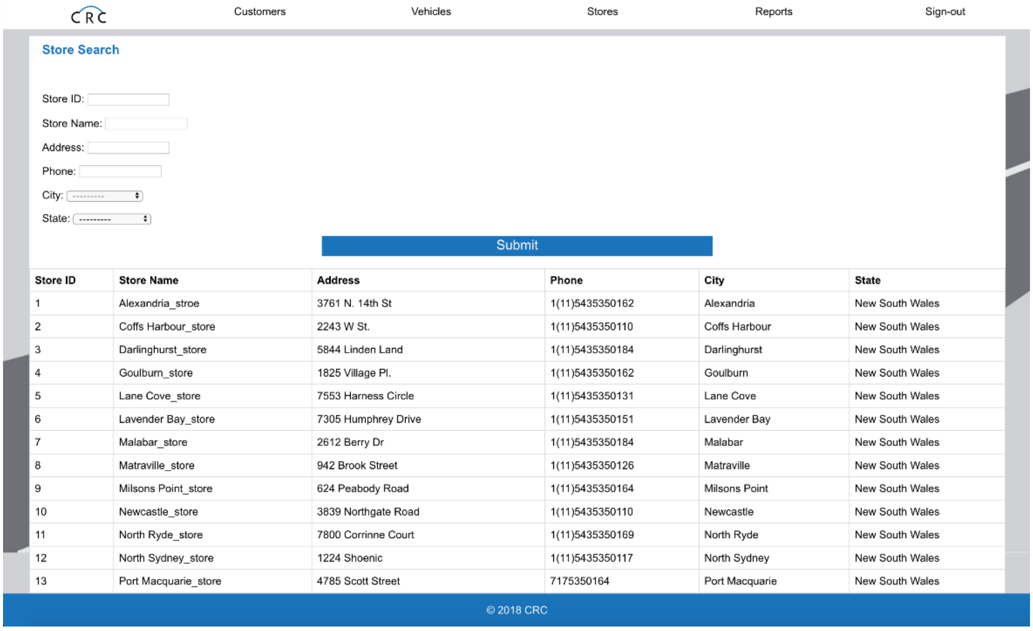
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

Our Promise

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

© 2018 CRC

Table display:



Store Search

Store ID:
Store Name:
Address:
Phone:
City: ---------
State: ---------

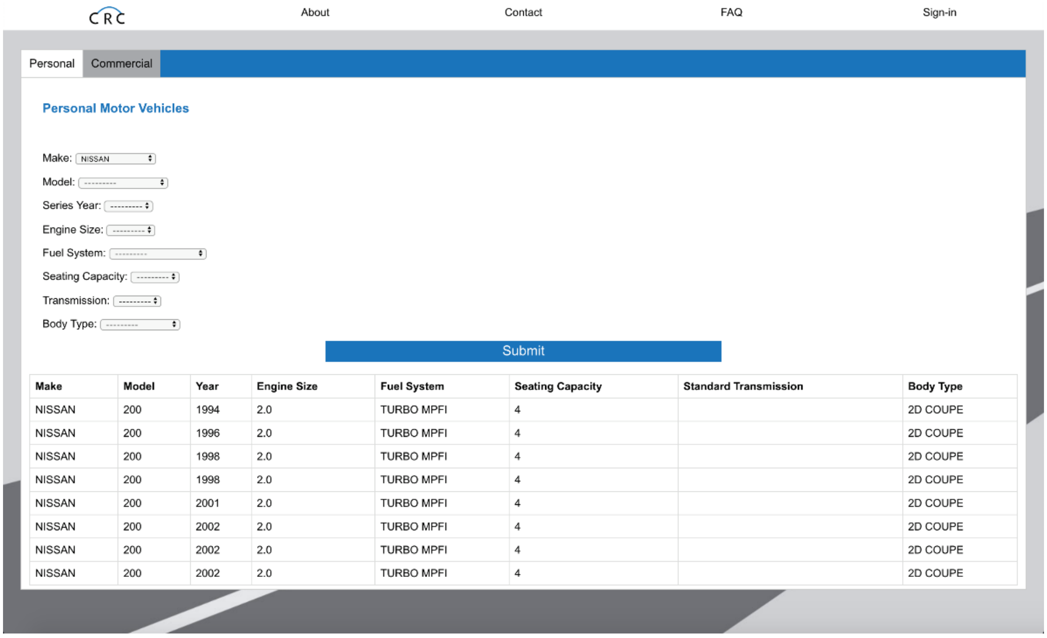
Store ID	Store Name	Address	Phone	City	State
1	Alexandria_stroe	3761 N. 14th St	1(11)5435350162	Alexandria	New South Wales
2	Coffs Harbour_store	2243 W St.	1(11)5435350110	Coffs Harbour	New South Wales
3	Darlinghurst_store	5844 Linden Land	1(11)5435350184	Darlinghurst	New South Wales
4	Goulburn_store	1825 Village Pl.	1(11)5435350162	Goulburn	New South Wales
5	Lane Cove_store	7553 Harness Circle	1(11)5435350131	Lane Cove	New South Wales
6	Lavender Bay_store	7305 Humphrey Drive	1(11)5435350151	Lavender Bay	New South Wales
7	Malabar_store	2612 Berry Dr	1(11)5435350184	Malabar	New South Wales
8	Matraville_store	942 Brook Street	1(11)5435350126	Matraville	New South Wales
9	Milsons Point_store	624 Peabody Road	1(11)5435350164	Milsons Point	New South Wales
10	Newcastle_store	3839 Northgate Road	1(11)5435350110	Newcastle	New South Wales
11	North Ryde_store	7800 Corrinne Court	1(11)5435350169	North Ryde	New South Wales
12	North Sydney_store	1224 Shoenic	1(11)5435350117	North Sydney	New South Wales
13	Port Macquarie_store	4785 Scott Street	7175350164	Port Macquarie	New South Wales

© 2018 CRC

7	22nd October 2018	10am – 12pm	▪ Customers can search for vehicles and results are displayed ▪ Customers can make an unlimited number of searches ▪ Employees can search for a customer and view a customer's history. ▪ Employees can make an unlimited number of searches. ▪ The board can view/download reports ▪ The client was shown how to create and enter data in the database ▪ Sensitive data cannot be viewed by non-employees ▪ Automatic sign out after two-hours is functional ▪ Viewed and discussed the progress of the webpage design and database structure ▪ Discussed and explain the: written test cases and sample queries ▪ Client is assured that all required functionality (all user stories 1-4 and 7) will be implemented by the due date

Evidence

Customer vehicle search:

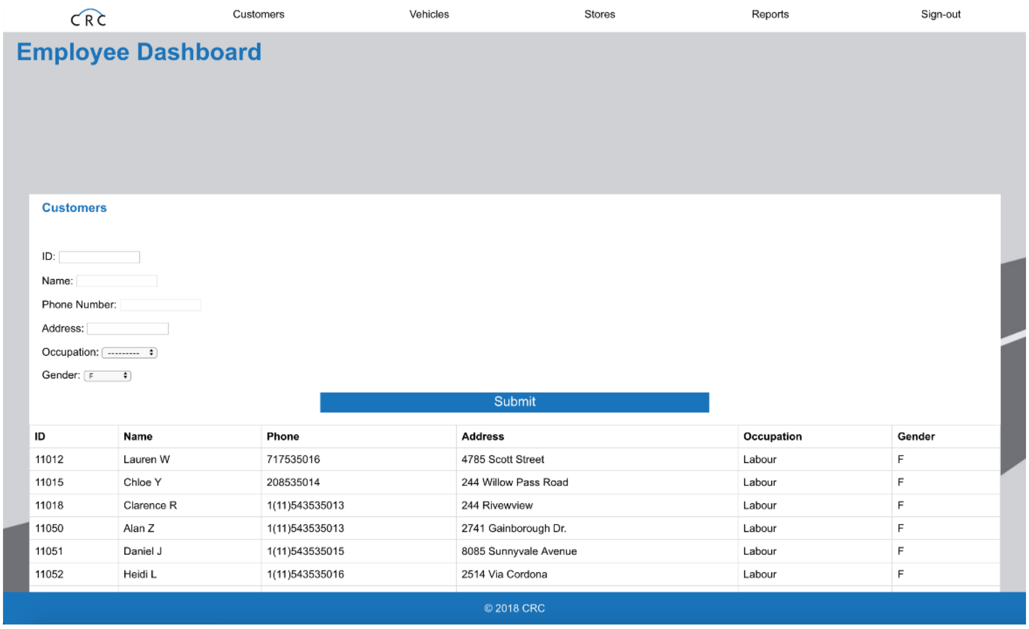


Personal Motor Vehicles

Make: NISSAN
Model: ---------
Series Year: ---------
Engine Size: ---------
Fuel System: ---------
Seating Capacity: ---------
Transmission: ---------
Body Type: ---------

Make	Model	Year	Engine Size	Fuel System	Seating Capacity	Standard Transmission	Body Type
NISSAN	200	1994	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	1996	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	1998	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	1998	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	2001	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	2002	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	2002	2.0	TURBO MPFI	4		2D COUPE
NISSAN	200	2002	2.0	TURBO MPFI	4		2D COUPE

Employee searching for customer:



Employee Dashboard

Customers

ID:

Name:

Phone Number:

Address:

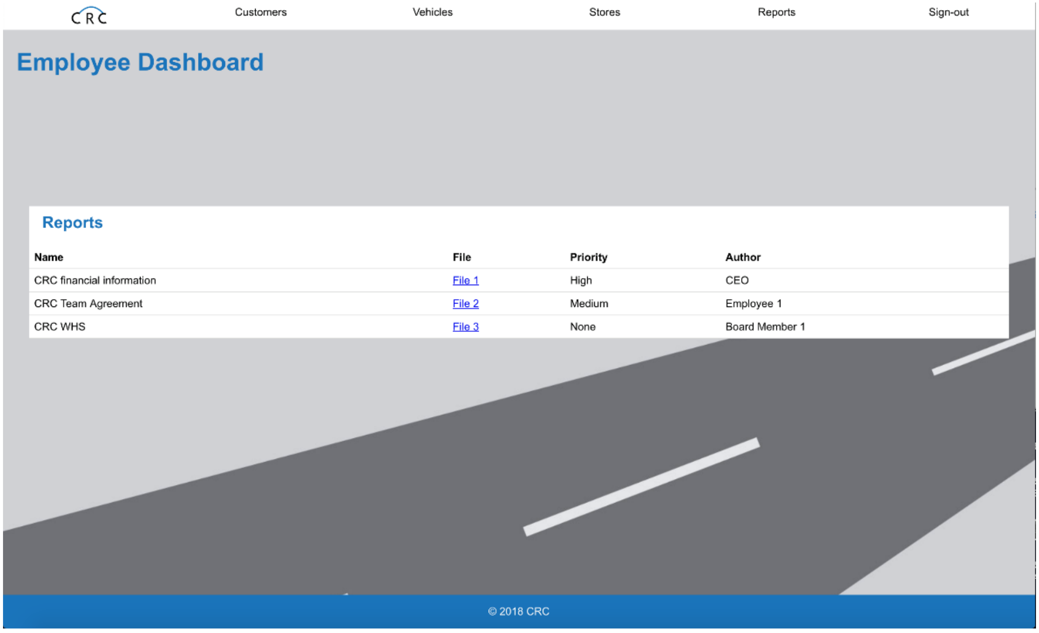
Occupation: ---------

Gender: F

ID	Name	Phone	Address	Occupation	Gender
11012	Lauren W	717535016	4785 Scott Street	Labour	F
11015	Chloe Y	208535014	244 Willow Pass Road	Labour	F
11018	Clarence R	1(11)543535013	244 Rivewview	Labour	F
11050	Alan Z	1(11)543535013	2741 Gainborough Dr.	Labour	F
11051	Daniel J	1(11)543535015	8085 Sunnyvale Avenue	Labour	F
11052	Heidi L	1(11)543535016	2514 Via Cordona	Labour	F

© 2018 CRC

View/download reports:



Employee Dashboard

Reports

Name	File	Priority	Author
CRC financial information	File 1	High	CEO
CRC Team Agreement	File 2	Medium	Employee 1
CRC WHS	File 3	None	Board Member 1

© 2018 CRC

References

Guru99. (2018). How to Write Test Cases: Sample Template with Examples. Retrieved from https://www.guru99.com/test-case.html