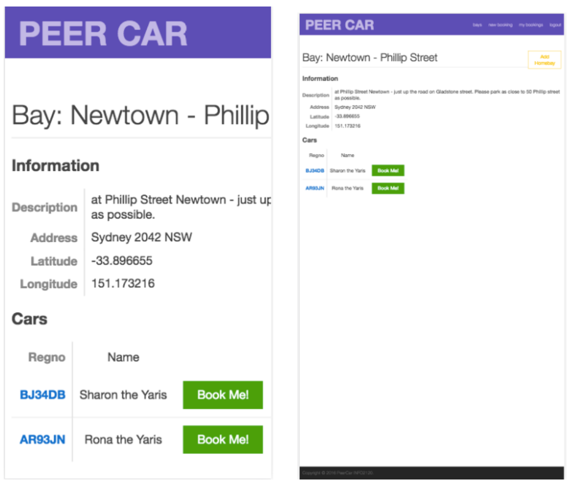
Data-User-Interaction Analysis and Design

Introduction:
The skeleton code provides a simplistic user interface that is very easy to use. The log in screen has good error trapping to ensure users are entering their correct details. Once logged in, a user can easily access any of the pages with the help of the nav-bar. This yet again is very simplistic, easy to find and easy to use. All of the information is displayed clearly with a minimalist style in mind. There is however, a few particular areas where the user interface could be greatly improved. Mobile is an extremely important aspect of business in this current age and it is essential that “Peer Car” has their service optimised for mobile devices. Along with this, users are always on the go. They need the ability to be able to find an available car that is closest to their current position. With this busy lifestyle in mind, an additional feature to extend a currently active booking would also be essential to add.

Usability for mobile:
Based on the current skeleton code, the service is not fit to be used on mobile. All of the aspects still work and it does get the job done. However it is very clear that the site has not been optimised for a mobile device. When the site launches the page is zoomed in so that it does not show all of the contents. However if a user zooms out, the text becomes incredibly difficult to see and the buttons almost impossible to press accurately. This would mean a user needs to constantly zoom in and out or pan back and forth across the screen. To add to this, the navbar stays in the same layout from desktop to mobile. This extends the width of the page to a much larger length than needed when the site is used on a mobile device.


PEER CAR

Bay: Newtown - Phillip

Information

Description	at Phillip Street Newtown - just up as possible.
Address	Sydney 2042 NSW
Latitude	-33.896655
Longitude	151.173216

Cars

Regno	Name	
BJ34DB	Sharon the Yaris	Book Me!
AR93JN	Rona the Yaris	Book Me!

PEER CAR bays new booking my bookings logout

Bay: Newtown - Phillip Street | Add Homebay

Information

Description	at Phillip Street Newtown - just up the road on Gladstone street. Please park as close to 50 Phillip street as possible.
Address	Sydney 2042 NSW
Latitude	-33.896655
Longitude	151.173216

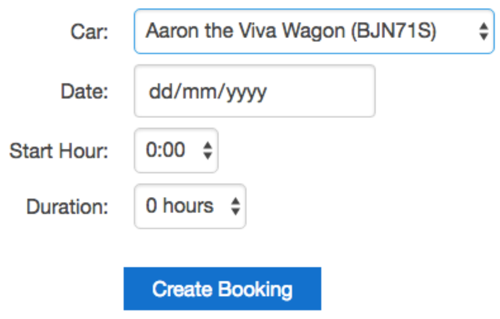
Cars

Regno	Name	
BJ34DB	Sharon the Yaris	Book Me!
AR93JN	Rona the Yaris	Book Me!

Website on iPhone 6 device

The Solution:
Making the website optimised for mobile is quite easy as long as the developer follows a few key principles. Firstly, the navigation bar should be collapsible. The use of a “hamburger menu” will allow each of the navigation tabs to be hidden, but upon touching the button they all appear. This saves a lot of space to account for the smaller screen size and lack of screen real estate. The next principle to follow is to make all of the images and buttons“responsive”. This can be achieved using a development system known as “bootstrap” quite easily. Buttons and images will now scale depending on the users screen size and this can account for any mobile device whether is be as large as an iPad or as small as an iPhone 4. All of the pages in the skeleton code have a lot of “white space” and so converting to a “mobile optimised” format would not be a big issue.

Available cars close to your location:
The current skeleton code would make things extremely difficult to find nearby cars if a user is on the go. At this current point in time they can perform a search to find a nearby car bay, but what if the cars are already booked? What if the user is in a new area and they do not know the neighbouring street names? Alternatively a user could go through each car, one by one, looking at its location and trying to match it to their own. This still has the issue of what cars are available and there are far too many cars for this to be a feasible task.



Car:	Aaron the Viva Wagon (BJN71S)
Date:	dd/mm/yyyy
Start Hour:	0:00
Duration:	0 hours

Skeleton “new booking” system

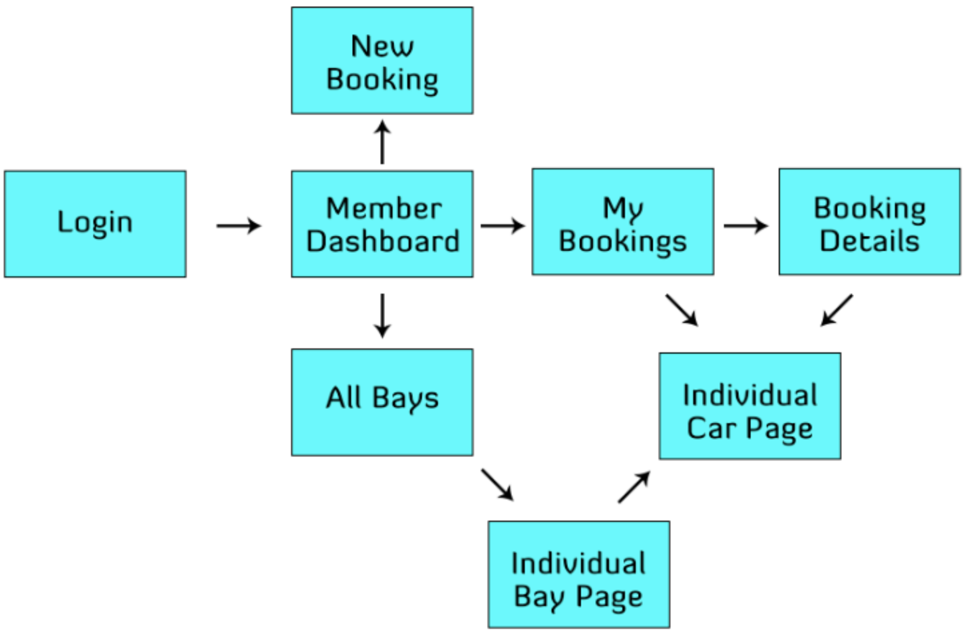
The Solution:
A more intuitive way of achieving this task would be through integrating the use of a Global Position System (GPS). Many users would be on their mobile phones when needing this feature and even for those who don’t, desktop computers still offer tracking services. By turning location to “on”, users would be able to see a list of cars that were available and within a small distance. Code would need to be added to determine if vehicles were available or not and extensive code would be required to determine the distance cars were away from the current user. The google maps API would be a great starting place to have a fluid layout of the area the person is in. This feature would be added to the “New booking” page in place of or assisting the current car selection tool.

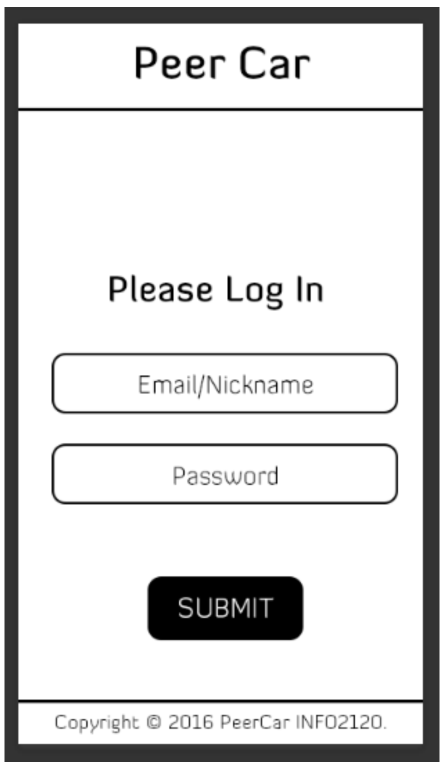
Extending an active booking:
The ability to extend a currently active booking would be an amazing feature for this service. There are a number of different scenarios where a person may be running late or urgently needing the vehicle for an extended period of time. It would be no issue to extend the current booking as long as it did not cause a clash with another booking. The current skeleton code would require the user creates an entirely new booking that starts on the hour. This is highly inefficient and could be extremely frustrating to customers who simply need a small extension in order to make it back on time or run a quick errand.

The Solution:
In order to implement a system into the current skeleton that allows a user to extend an active booking, some significant changes would need to be made. This new feature would be available from the “My Bookings” page where an active booking would show at the top, highlighted in some way. From here a user would be able to click the already existing “view details” button. It is this page that all of the details about the booking are shown, including the duration and start time. A user would be able to apply for an extended booking based on how much time they need. The system would compare the new booking end time with the start time for any other bookings of that same car on that same day. If there was not a clash with another booking then the application would be approved immediately. If there was a clash however, the latest end time possible, taking into consideration a small time buffer, would be displayed for when the user must return the car. The user can choose to accept or reject this proposed time.

Conclusion:
It is clearly evident that the skeleton code is a simplistic and easy to use version of the service. The proposed solutions to the issues with the current service will benefit “Peer Car” immensely.

Current Site Layout





Peer Car

Please Log In

Email/Nickname

Password

SUBMIT

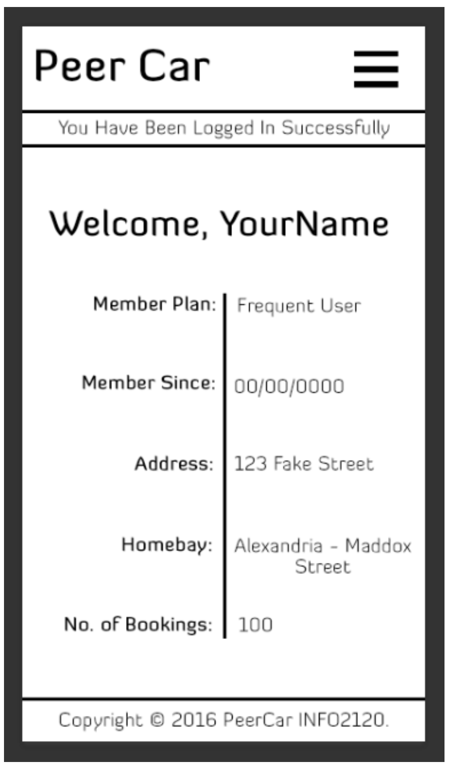
Copyright © 2016 PeerCar INFO2120.

Log In Page

In the wireframe for the login, the layout is very similar to the original. There is far less white space and the design has been streamlined for mobile.

User Dashboard

The user dashboard welcomes the user by name. It lists all of their key information. A message at the top tells the user they managed to log in successfully.



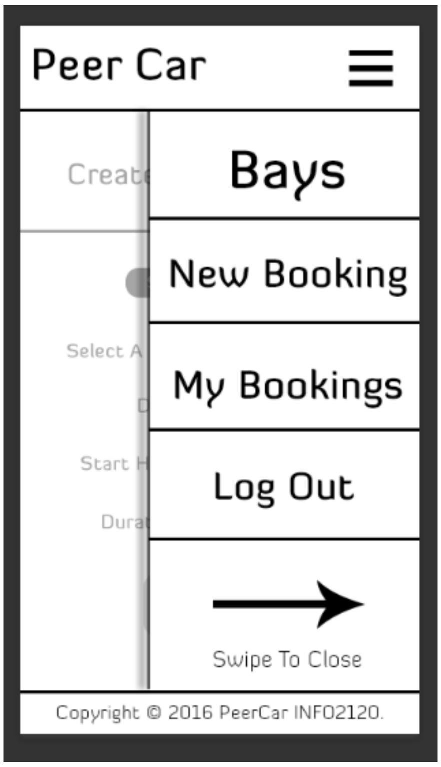
Peer Car

You Have Been Logged In Successfully

Welcome, YourName

Member Plan:	Frequent User
Member Since:	00/00/0000
Address:	123 Fake Street
Homebay:	Alexandria - Maddox Street
No. of Bookings:	100

Copyright © 2016 PeerCar INFO2120.

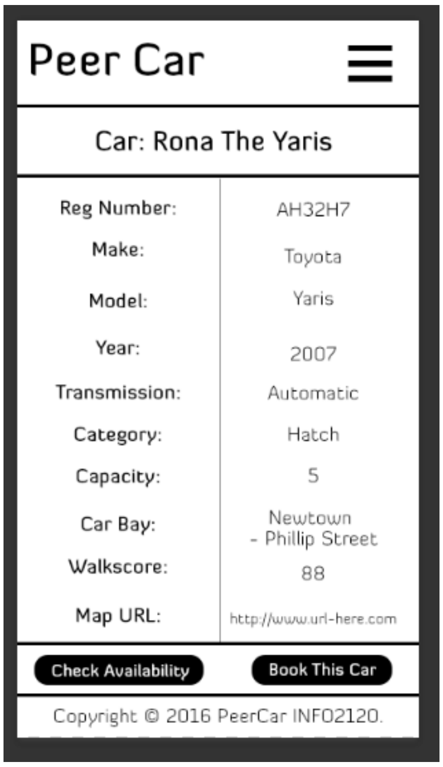


Hamburger Menu

A hamburger menu replaces the standard navbar. This saves a lot of space. With the tap of a button the regular menu elements slide out and are accessible to the user.

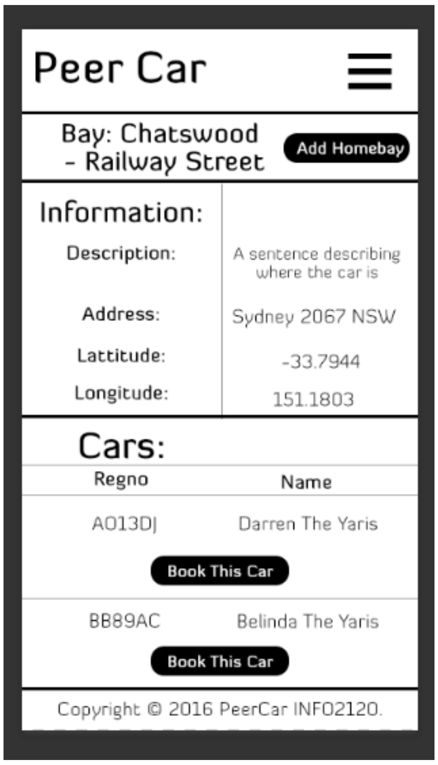
Individual Car

Each car has an individual page listing all of its details. From this page, access to a new feature is available. This feature shows the times that this car is available for that day.


Peer Car

Car: Rona The Yaris

Reg Number:	AH32H7
Make:	Toyota
Model:	Yaris
Year:	2007
Transmission:	Automatic
Category:	Hatch
Capacity:	5
Car Bay:	Newtown - Phillip Street
Walkscore:	88
Map URL:	http://www.url-here.com


Peer Car

Bay: Chatswood - Railway Street

Information:

Description:	A sentence describing where the car is
Address:	Sydney 2067 NSW
Lattitude:	-33.7944
Longitude:	151.1803

Cars:

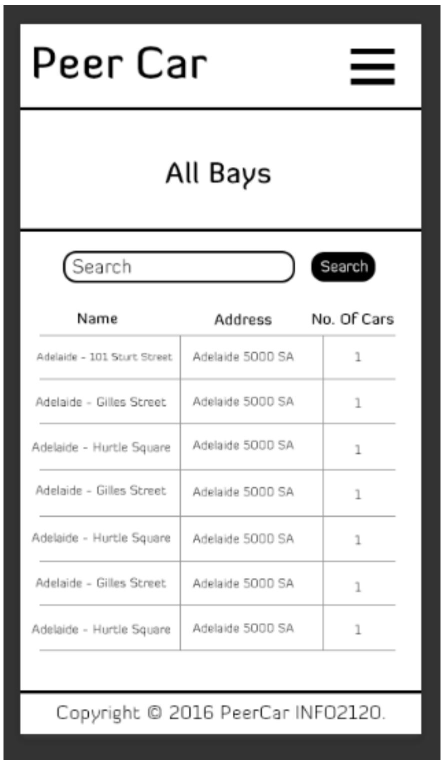
Regno	Name
AO13DJ	Darren The Yaris
BB89AC	Belinda The Yaris

Individual Bay

Each bay also has its own individual page. This provides key information about where the bay is located and also what cars are stationed at this bay. From here a user can book that specific car.

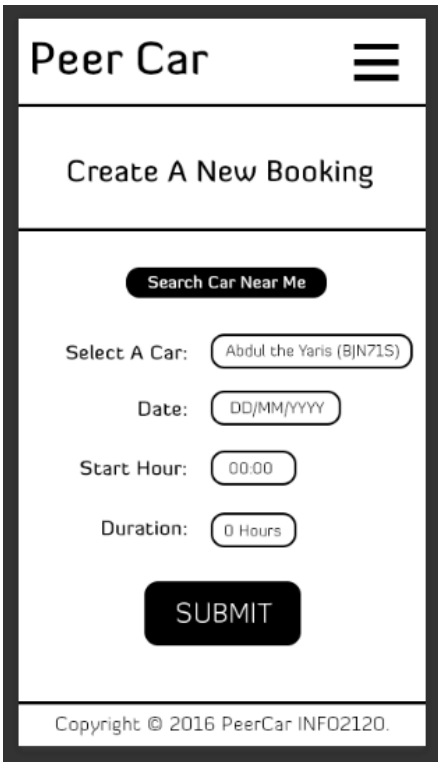
All Bays

A user is also able to view all of the existing bays. This can help them find a particular bay that they need. A search feature allows filtering the large amounts of information to be more efficient.


Peer Car

All Bays

Name	Address	No. Of Cars
Adelaide - 101 Sturt Street	Adelaide 5000 SA	1
Adelaide - Gilles Street	Adelaide 5000 SA	1
Adelaide - Hurtle Square	Adelaide 5000 SA	1
Adelaide - Gilles Street	Adelaide 5000 SA	1
Adelaide - Hurtle Square	Adelaide 5000 SA	1
Adelaide - Gilles Street	Adelaide 5000 SA	1
Adelaide - Hurtle Square	Adelaide 5000 SA	1



Peer Car

Create A New Booking

Search Car Near Me

Select A Car:	Abdul the Yaris (BJN71S)
Date:	DD/MM/YYYY
Start Hour:	00:00
Duration:	0 Hours

SUBMIT

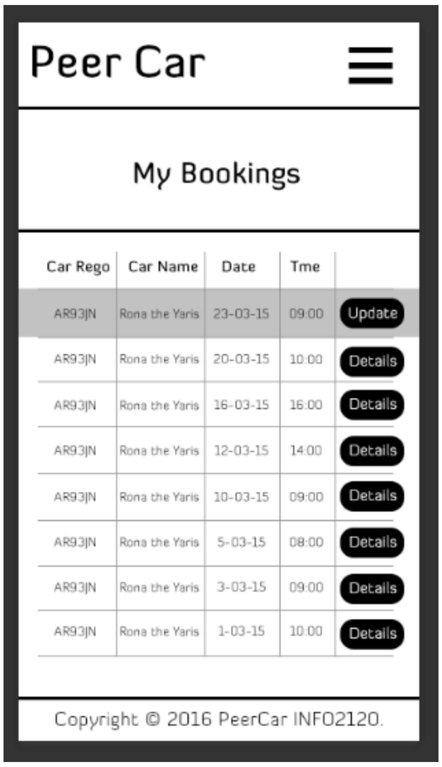
Copyright © 2016 PeerCar INFO2120.

New Booking

A user is able to create a new booking. From here a new feature was added where they can search for a car close by. By pressing the button a map is launched with nearby cars highlighted.

My Bookings

A user can see all of their past bookings. Any active bookings are highlighted and by pressing update they can apply for an extension on their time for the car.

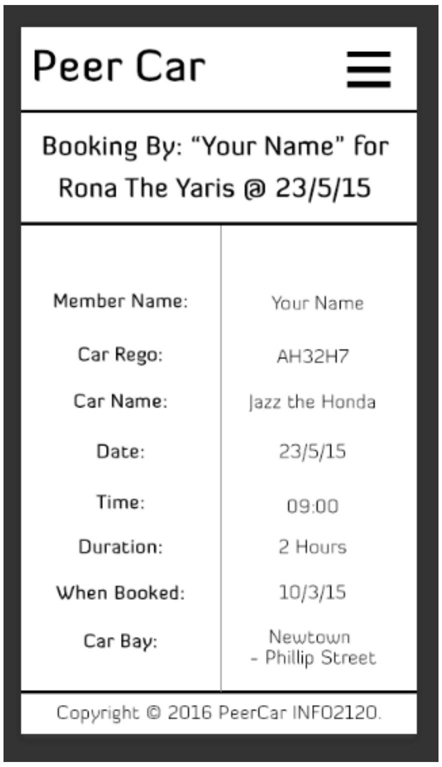


Peer Car

My Bookings

Car Rego	Car Name	Date	Tme	
AR93JN	Rona the Yaris	23-03-15	09:00	Update
AR93JN	Rona the Yaris	20-03-15	10:00	Details
AR93JN	Rona the Yaris	16-03-15	16:00	Details
AR93JN	Rona the Yaris	12-03-15	14:00	Details
AR93JN	Rona the Yaris	10-03-15	09:00	Details
AR93JN	Rona the Yaris	5-03-15	08:00	Details
AR93JN	Rona the Yaris	3-03-15	09:00	Details
AR93JN	Rona the Yaris	1-03-15	10:00	Details

Copyright © 2016 PeerCar INFO2120.



Peer Car

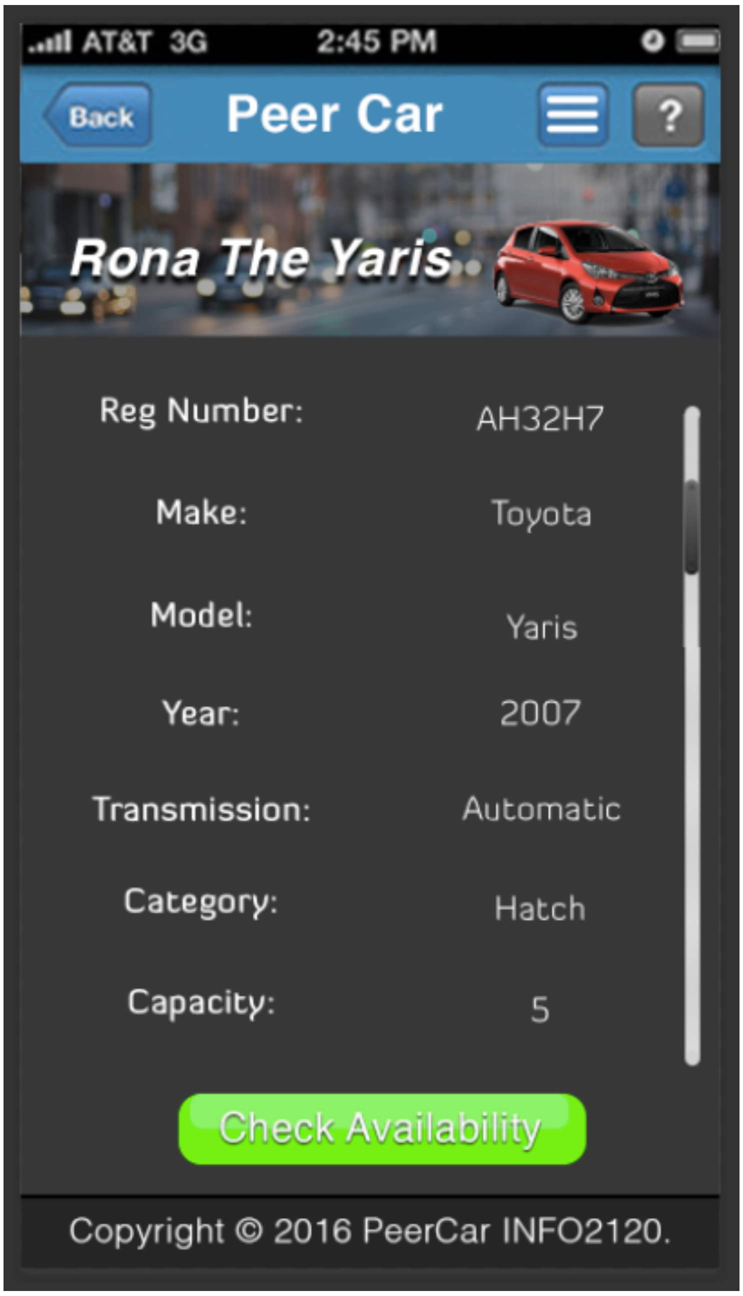
Booking By: "Your Name" for Rona The Yaris @ 23/5/15

Member Name:	Your Name
Car Rego:	AH32H7
Car Name:	Jazz the Honda
Date:	23/5/15
Time:	09:00
Duration:	2 Hours
When Booked:	10/3/15
Car Bay:	Newtown - Phillip Street

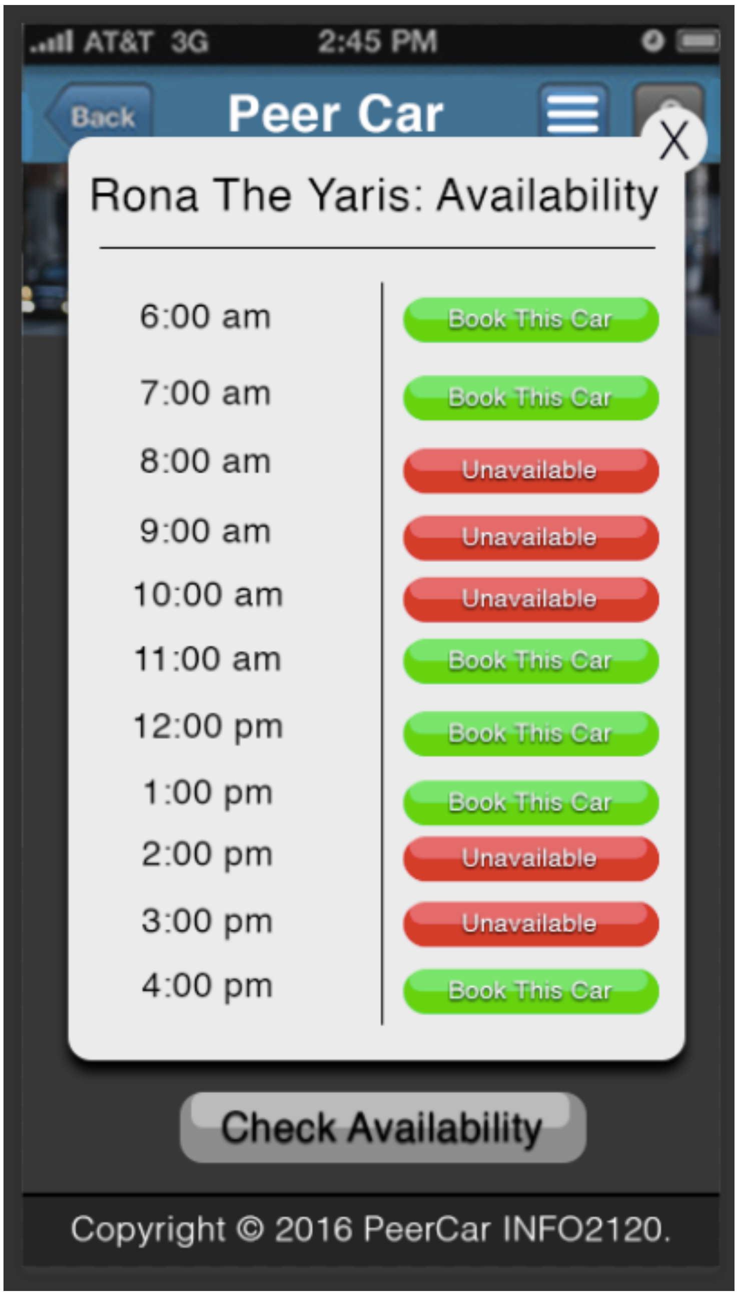
Copyright © 2016 PeerCar INFO2120.

My Bookings

Each booking has its own individual details. A user can look at this in order to see all of the key information. The design has been streamlined in order to be optimised for mobile.



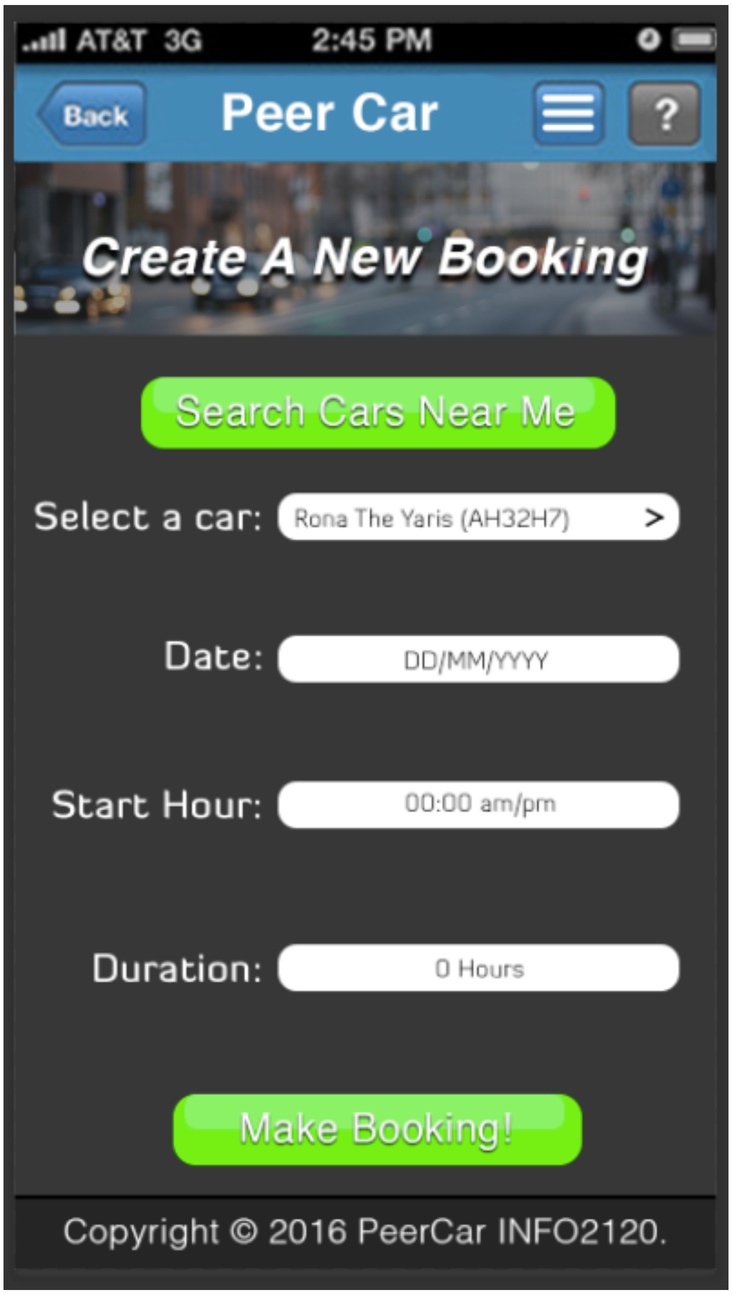
Reg Number:	AH32H7
Make:	Toyota
Model:	Yaris
Year:	2007
Transmission:	Automatic
Category:	Hatch
Capacity:	5



Rona The Yaris: Availability

6:00 am	Book This Car
7:00 am	Book This Car
8:00 am	Unavailable
9:00 am	Unavailable
10:00 am	Unavailable
11:00 am	Book This Car
12:00 pm	Book This Car
1:00 pm	Book This Car
2:00 pm	Unavailable
3:00 pm	Unavailable
4:00 pm	Book This Car

The mockups above depict what a real version of the individual car page would look like. When the user presses the green "check availability" button, a pop up window is launched. The button becomes greyed out to show it has been selected. The user can see what times the car is available for that day and they can make a direct booking on that particular car at that time.This would launch the "new booking" page and autofill the data for the start time, the date and the car they wish to use. Pressing the X in the top right corner can close this pop up and allow the user to keep checking out the details of the car. A small image in the top right shows what the car looks like to help give the user a graphical sense of the car they are booking. A scroll bar can be seen on the right to indicate to the user that there is hidden information lower down. This information is not super important however it is available to the user just in case they need it. This design heavily iterates upon the wireframe and also the original skeleton code. It is a sleek and simple design that is easy for the user to understand.

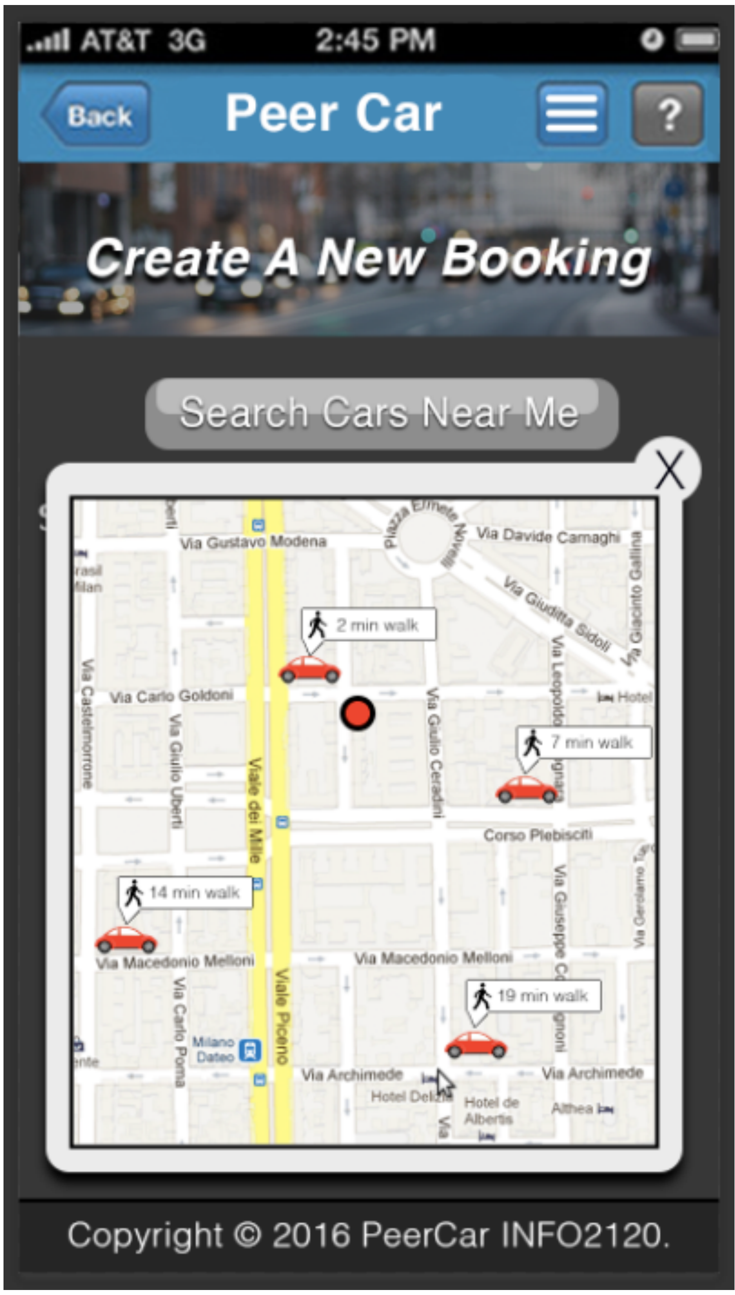


Select a car: Rona The Yaris (AH32H7) >

Date: DD/MM/YYYY

Start Hour: 00:00 am/pm

Duration: 0 Hours



The mockups above depict what a real version of the “create new booking” page would look like. This is in mobile format as many users will be on the go when they need to make a booking. A user is able to “Search Cars Near Me” by pressing the green button on the top left. This button becomes greyed out to show that it has been clicked and launched a map showing the users location and all of the nearby cars. A tiny pop up shows how far away each car is using Google Maps API. A user can then select which car they want to use and this directs them to the individual car page shown on the previous page. From here they can see the times available and make a booking that suits them. When they return to the form the date, start hour and car will all have been autofilled based on their earlier selections. They simply enter a duration and press “Make Booking!”.

The Effected Code:

In creating and implementing the aforementioned features and changes, there would be a large amount of code effected. The majority of the code changes would be cosmetic however there are a few key changes that would specifically relate to the database. The changes are in regards to the “car availability checker” and also the “car near me” function.

The code effected for the “car availability checker” is mostly SQL read statements. When this function is launched the code needs to select the start time of all the bookings related to a particular car. From here a graphical system will show “unavailable” on all of the start time hours acquired from the read. The code will then show “available” on all of the hours that were not returned in the read. All of the information required to make this feature work is already in the database. An example of the required read statement would look like this:

```
SELECT bookingid
FROM booking JOIN member ON (madeby = memberno)
WHERE (car=%s OR email = %s) AND (endtime > %s AND starttime < %s)
```

The parameters being passed in are the endtime of the current booking and a new time which is adds the duration. In this way it also checks that the user does not have a booking which will overlap.

The next part of the code effected would be in relation the to “car near me” function. This car takes a users location and searches for cars that are nearby. Yet again read statements would achieve this task and all of the necessary data is already available within the database. The code would need to acquire the attitude and longitude attributes of all of the car bays. Within each bay is a number of cars. All the bays within a nearby radius of the person, perhaps a 20 minute walk will be listed. The code would need to read the location of the car bay. Using google maps it can then determine the distance between the car bay and the user. If the bay is within the 20 minute radius it will show up on the map. For this reason the latitude and longitude attributes are extremely important.

It is clearly evident that some SQL statements would need to be created in order to successfully implement these features. It is extremely important to note however, that no additional information is required in order to make these features functional. Simply by reading the current data and comparing it, all of the required information can be determined.