PROPER COFFEE

Report by Harry Foster

Contents

Contents

Idea Development

For my final Major Project I have decided to redevelop a concept I used for a previous project. The project I worked on before went under the working name of ‘Not another Starbucks’ an application which allows users to locate independent coffee shops.

I chose to continue this project as I saw so many opportunities to expand on what I created in the original application and wanted to take it further, using an idea I’ve previously had also allowed to me to gain a head start on the project meaning that I can take it much further. Not another Starbucks was a website which used HTML, CSS and JavaScript, I expanded the functionality using Javascript libraries such as the Leaflet.js library which is a mapping plug in.

I began this new project by looking back at what I managed to achieve in the previous, it helped to refresh my memory of the past project and also allowed me to look at it from a critical standpoint and see what I could do different and what I could add. Whilst I am using that project as inspiration this new one will be started from scratch as I don’t want to limit myself by using code intended for a previous application as I that would lead to not looking at different and better ways to implement features.

In the next pages I will briefly Summarise the development of Not another Starbucks.

NOT ANOTHER
STARBUCKS

Concepts

For Not another Starbucks I generated content showing how the app would be advertised and materials like the logo as I this gave me a better understanding of what the wireframes would need to look like. I discovered the statistic that in 2012 80% of the coffee sold by the UK's top 100 coffee retailers was from just 2 brands and thought this would be good to include.

One of the reasons that I identified that somebody wouldn't want to use the app is that they have an existing loyalty scheme with one of the coffee chains so feel that they might miss out by going to indie coffee shops. A united loyalty scheme which some of the indie coffee shops could opt in to could solve this problem.

I decided to use a maps API to display the coffee shops in a map view to help the user navigate to the shop. The logo was designed to be clean and simple yet explain the basic concept of the app at the same time. I created all of the graphics in the concepts to make sure they fit with the style of images I want within my app to have a consistent theme.

The key messages that I wanted to come through in my concepts are detailed on the following pages in the form of advertising materials.

BE LESS BORING

80% OF COFFEE SOLD IN THE UK IS
SOLD BY JUST 2 BRANDS*

TRY SOMETHING NEW

*80% OF THE COFFEE SOLD BY THE TOP 100 COFFEE RETAILERS IN THE UK WERE SOLD BY STARBUCKS AND COSTA

NOT ANOTHER
STARBUCKS

SO WHAT DOES IT DO?

NOT ANOTHER STARBUCKS LOCATES AND RECOMMENDS THE BEST INDIE COFFEE SHOPS NEAR YOUR LOCATION

THE BEST COFFEE

EACH SHOP HAS A RATING AND REVIEWS TO MAKE SURE YOU ALWAYS GET AMAZING COFFEE

HAVE A LOYALTY CARD?

SO DO WE

THE NOT ANOTHER STARBUCKS CARD
UNITES INDIE COFFEE SHOPS TO REWARD
YOUR LOYALTY AT SELECTED COFFEE SHOPS.

FIND YOUR WAY

DON'T GET LOST OVER A COFFEE WITH INTEGRATED MAPPING USING THE GOOGLE API YOU WILL ALWAYS KNOW THE WAY

SO WHEN YOU GO SOMEWHERE NEW

TRY SOMEWHERE NEW

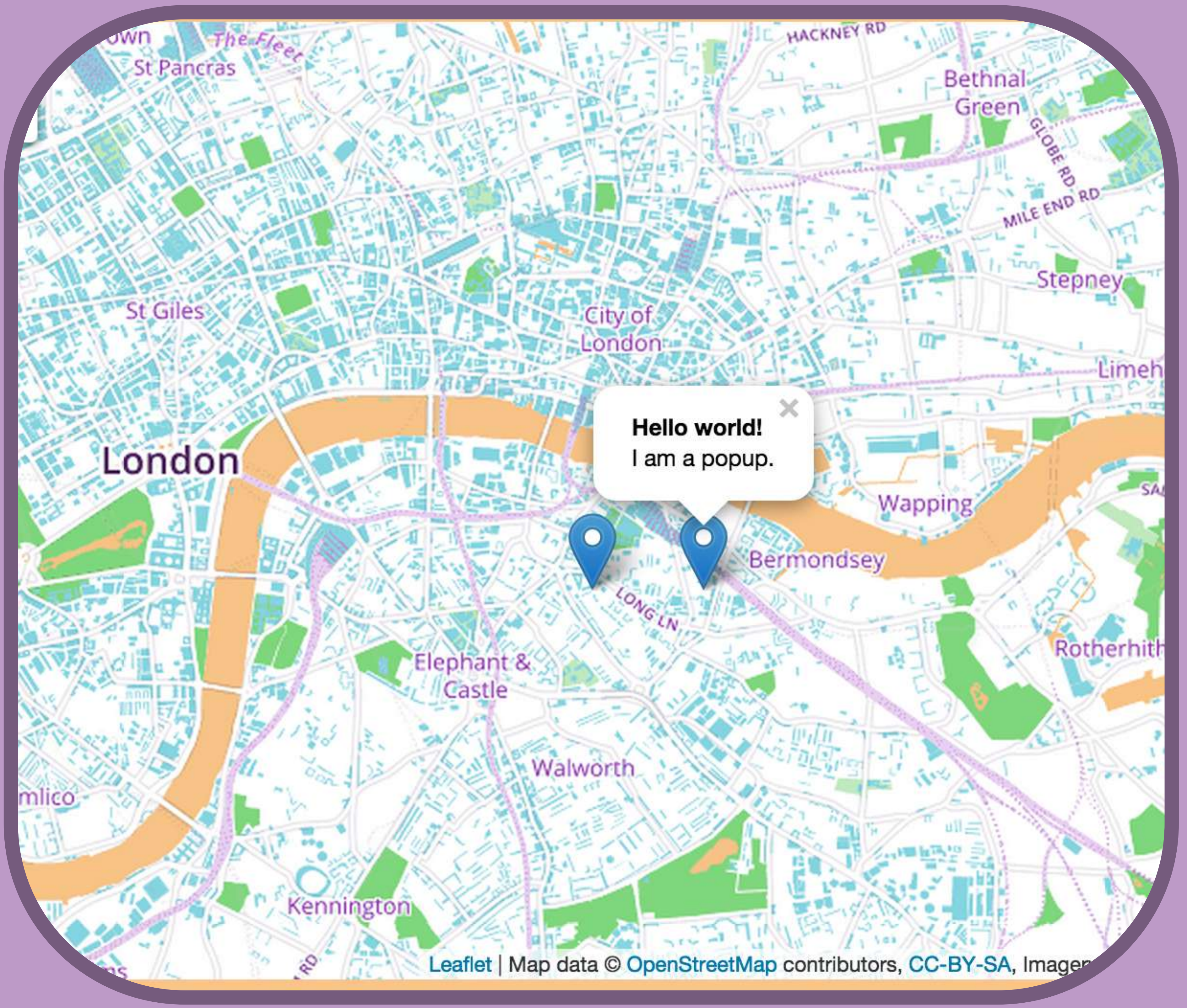
NOT ANOTHER
STARBUCKS

Development

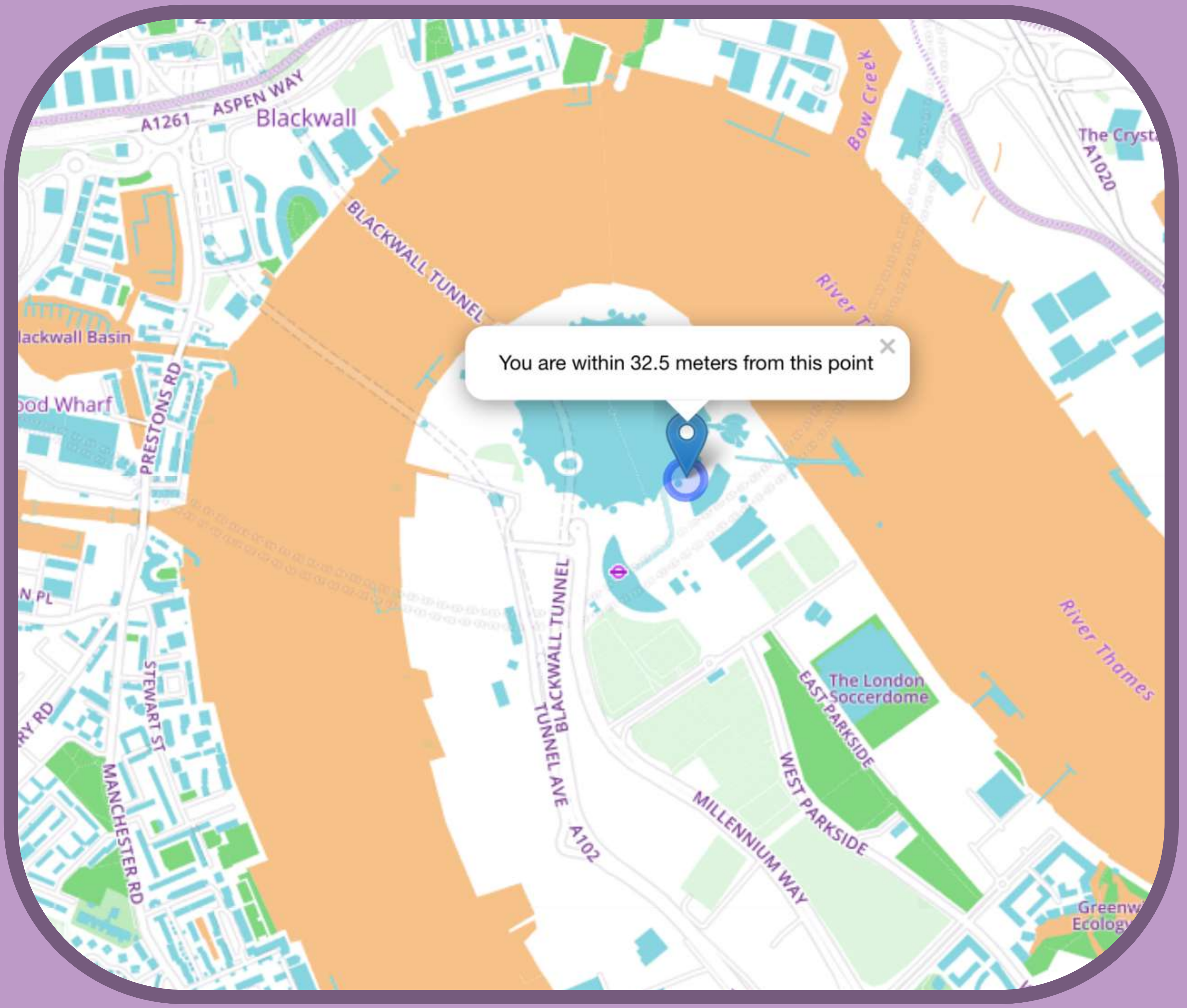
I used HTML5 Boilerplate to give a nicely structured blank canvas to work from for my app. It structures the folders ready to start developing which saves a lot of time. I had to edit it slightly to make it fit for my purpose but it saves a huge amount of time compared to starting from scratch.

One of the first things I wanted to get implemented was mapping as this is one of the key features of the app and take up most of the display, I used leaflet.js javascript library and Mapbox to implement a browser based map displaying a specific location with points on and pop up boxes and then used MapBox to create a design for the the map tiles and got some mixed results, it's natively responsive and has functionality with iOS and android for touch screen use.

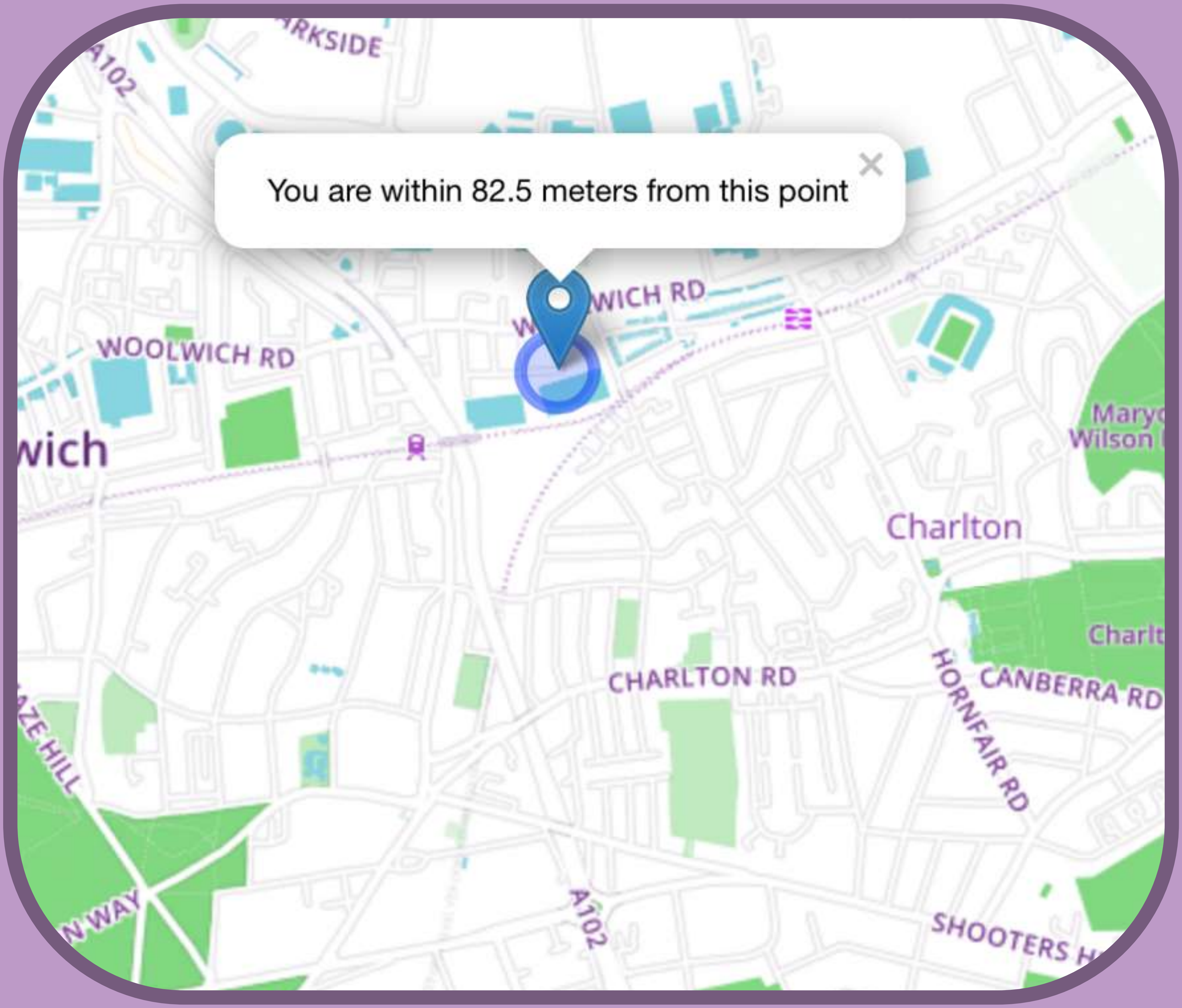
I then worked on making the map take up the entire browser window and loading your current location on opening, the map will then be centered around your location. I then added a circle which shows your current location with a popup message showing the accuracy of the location shown.



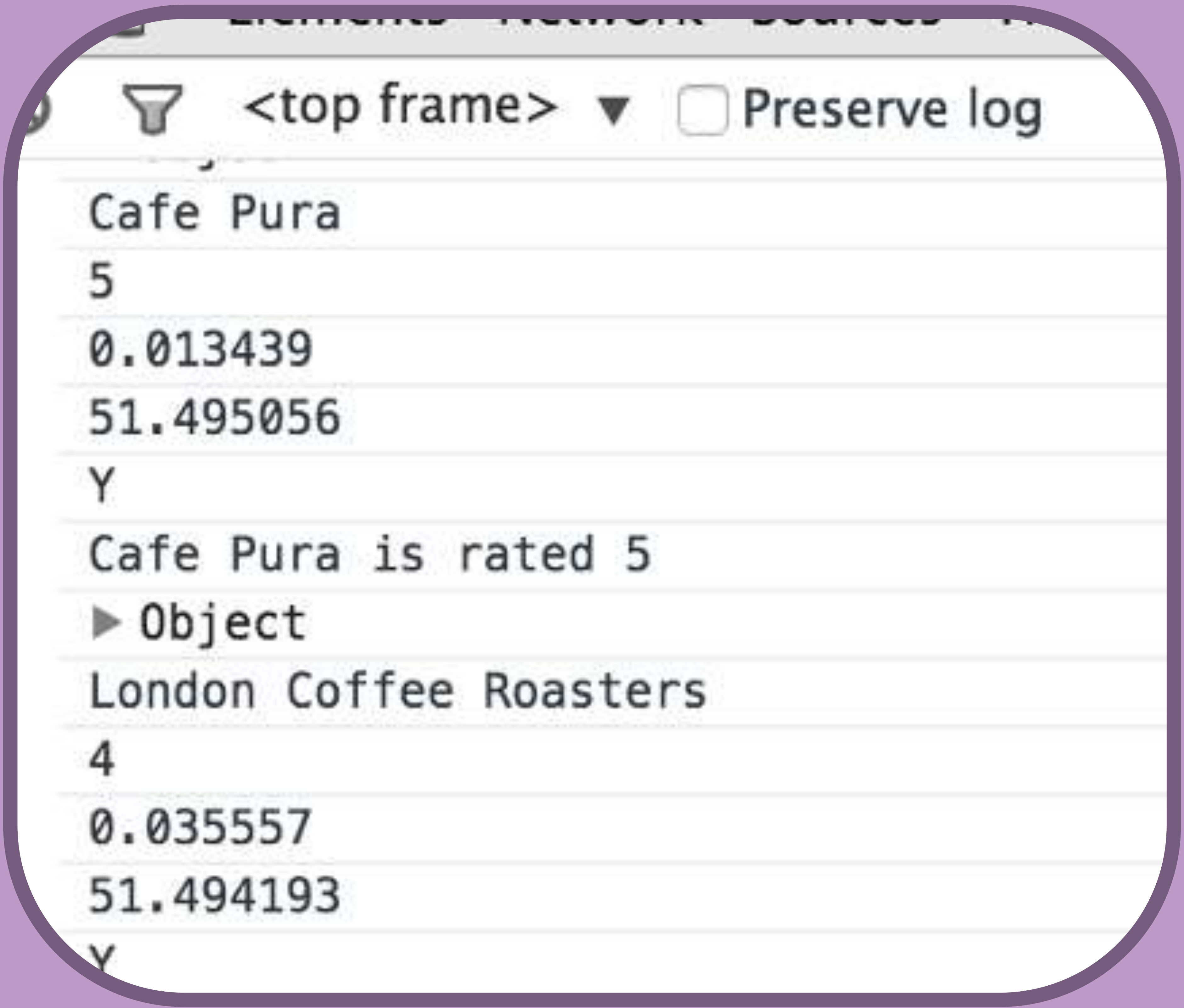
Hello world!
I am a popup.



You are within 32.5 meters from this point



You are within 82.5 meters from this point



```
Cafe Pura
5
0.013439
51.495056
Y
Cafe Pura is rated 5
▶ Object
London Coffee Roasters
4
0.035557
51.494193
Y
```

Development

I then begun working on using a spreadsheet of coffee shops and importing the data with JSON. I used the code I made previously as a basis and started adapting it. The fields I decided to use were coffee shop name, longitude, latitude, rating and availability of loyalty scheme. I encountered some problems but achieved a layer of markers of the coffee shops on the map which are imported from a spreadsheet.

The browser has access to the spreadsheet and imports the data from it as a JSON object I then get the browser to interpret fields like longitude and latitude and represent it on the map.

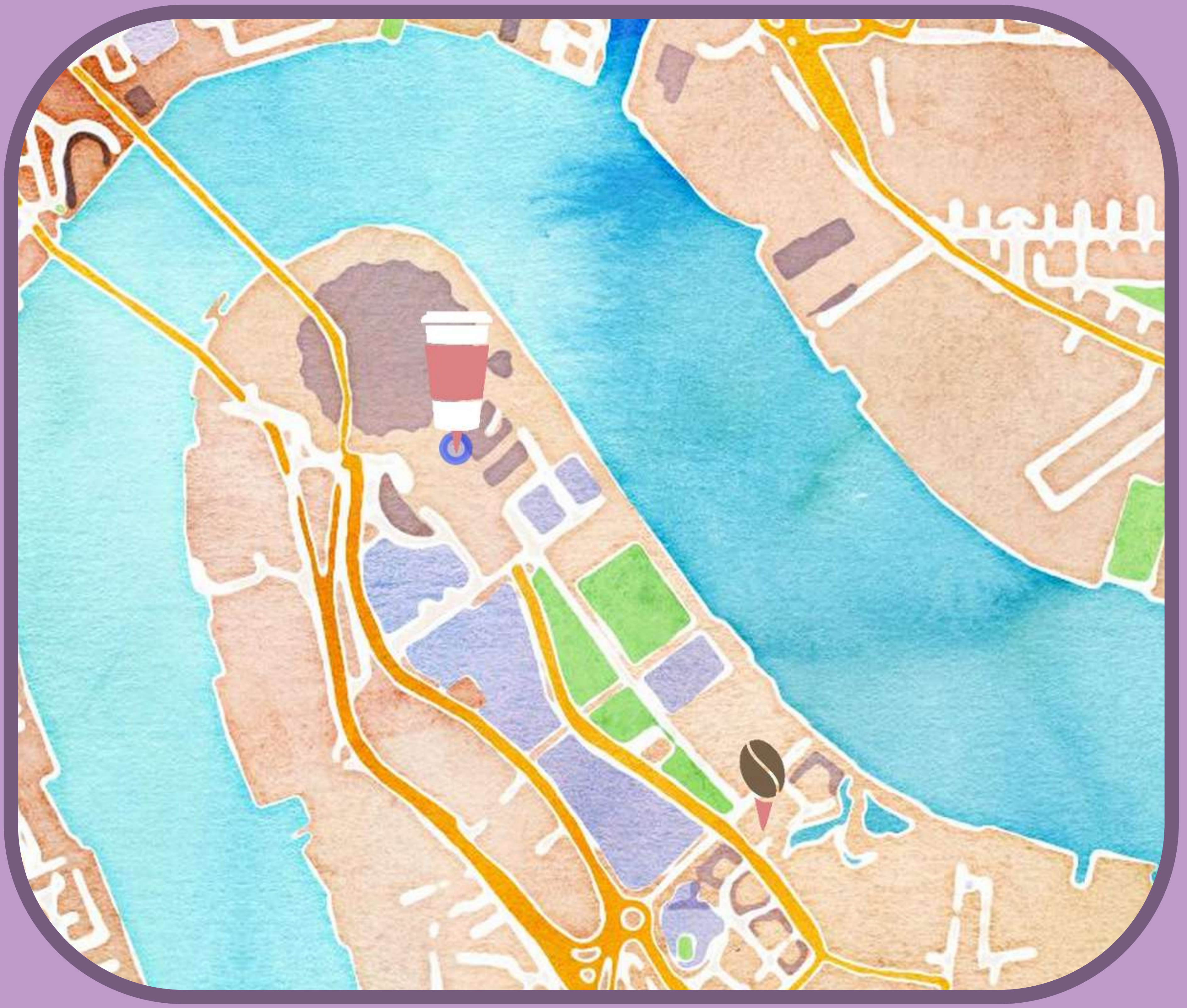
Initially the app outputs specific parts of the data object I imported from my spreadsheet using JSON to the console, such as the longitude and latitude. By managing to get the application to print this data to the console I can use it to place markers on the map in the locations of the coffee shops I want to show. This is all done using javascript.

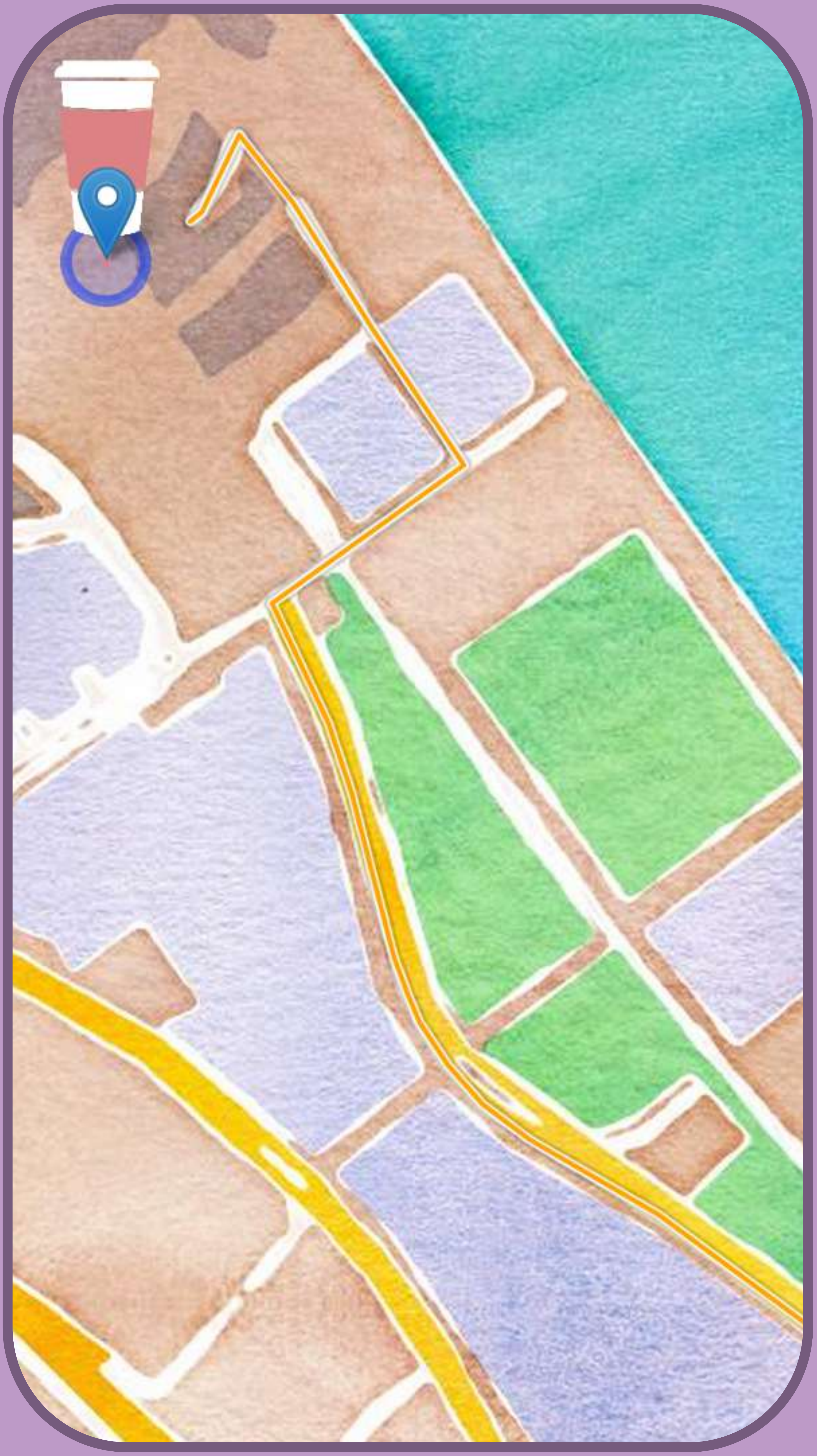
Development

Although I intended to move back to using a design I made using MapBox to change the map tiles for now until I had more time to spend on working on a really nice design I implemented a tile design called Watercolor which was made by Stamen; a website which has a selection of map tile designs that are compatible with Open Street Map (Which powers Leaflet). Stamen don't give you the option to customise the design which if you were actually using it to guide yourself to a location might be slightly lacking in detail.

Next I worked on customising the markers on the map to use ones that I designed, a coffee bean for the location of the coffee shops, these took some design refinement to look fitting for the design and achieved a fitting marker although it may require extra refinement later.

This part of the process was notable because it was the first time the application began to look like a map of coffee shops and makes the project project unique instead of just displaying a map.





Development

The next piece of functionality I added was the ability to use a routing plugin from Leaflet to help guide you to your location. This plug in is called Leaflet Routing Machine when you press one of the coffee shops I added a 'Get directions' link, this activates the javascript to route from your location to the chosen location, it makes 2 new points on top of the existing points to do this. It works well and also displays a list of directions although these could be slightly intrusive on mobile.

Then I created a list of next steps to implement:

- Make route line a different colour
- Populate the spreadsheet with ~50 independent coffee shops.
- Info popup: First line of address, postcode, Open right now, Free WiFi, link website.
- Text directions hidden by default, tap to reveal
- Splash screen
- Menu with list of coffee shops and mention of loyalty card (progressive enhancement)
- Upload it to
- Push all code to GitHub

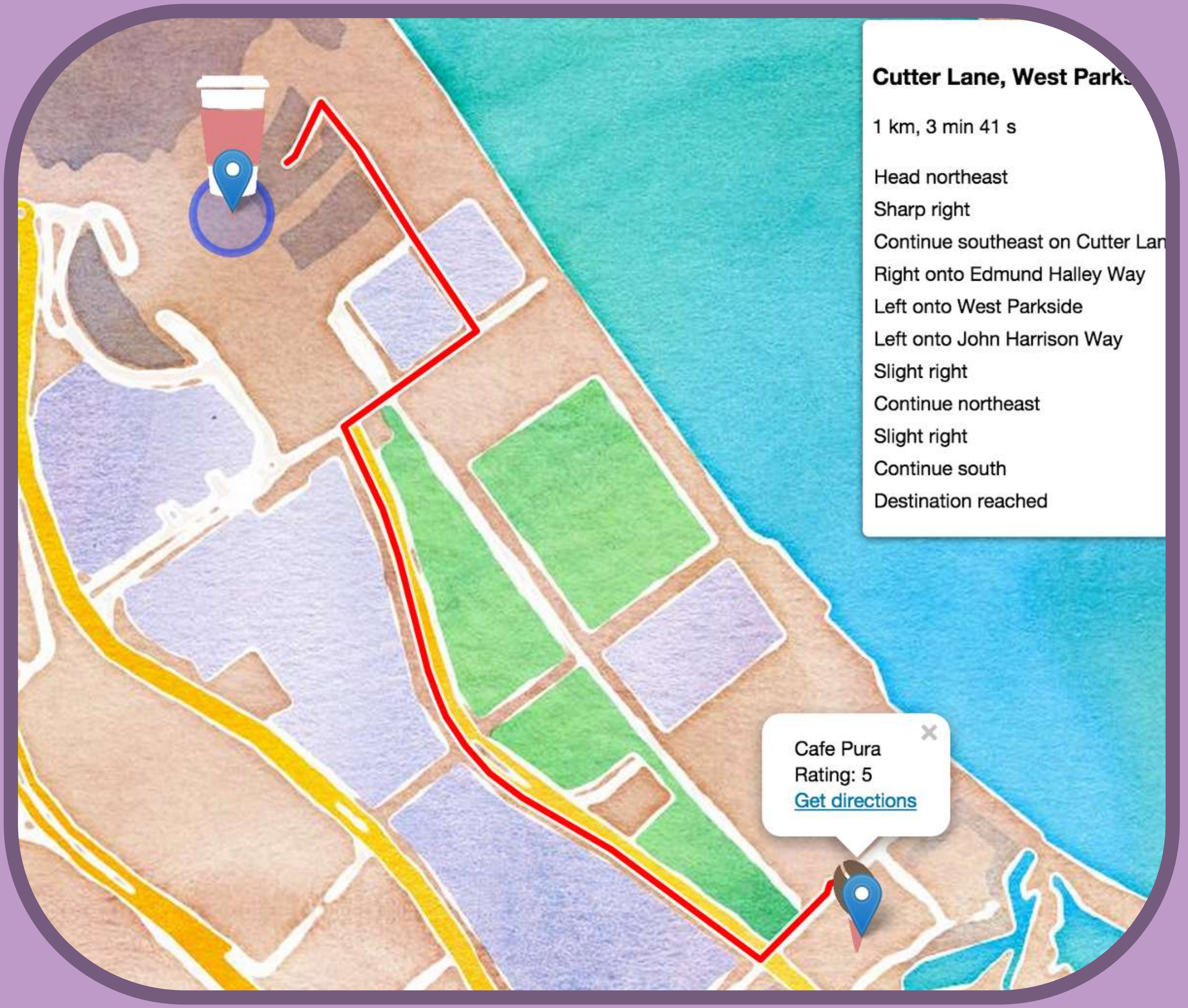
Development

I worked on changing the line colour to red, in doing so I also realised that part of the reason that the lines visibility wasn't great was because it wasn't thick enough so I changed the weight of the line as well.

I then began to bulk out my spreadsheet a little bit more, I added more fields so I can add some more data to the points, I included things like wifi availability and opening times as well as some blank columns for functionality I may add at a later point. The data I used was pulled from foursquare and google maps to double verify and gain as much data as possible.

Then I began incorporating a lot more of this data in the pop up boxes that appear next to each coffee shop as well as adding phone call links and website links as well as icons to represent these.

The opening times are static and don't change by day as I didn't take into account that shops tend to have different opening times on different days, in future I would incorporate JQuery to detect what day it is and then output opening times from that day but at the stage focusing on getting more functionality out of my app took a more important role.

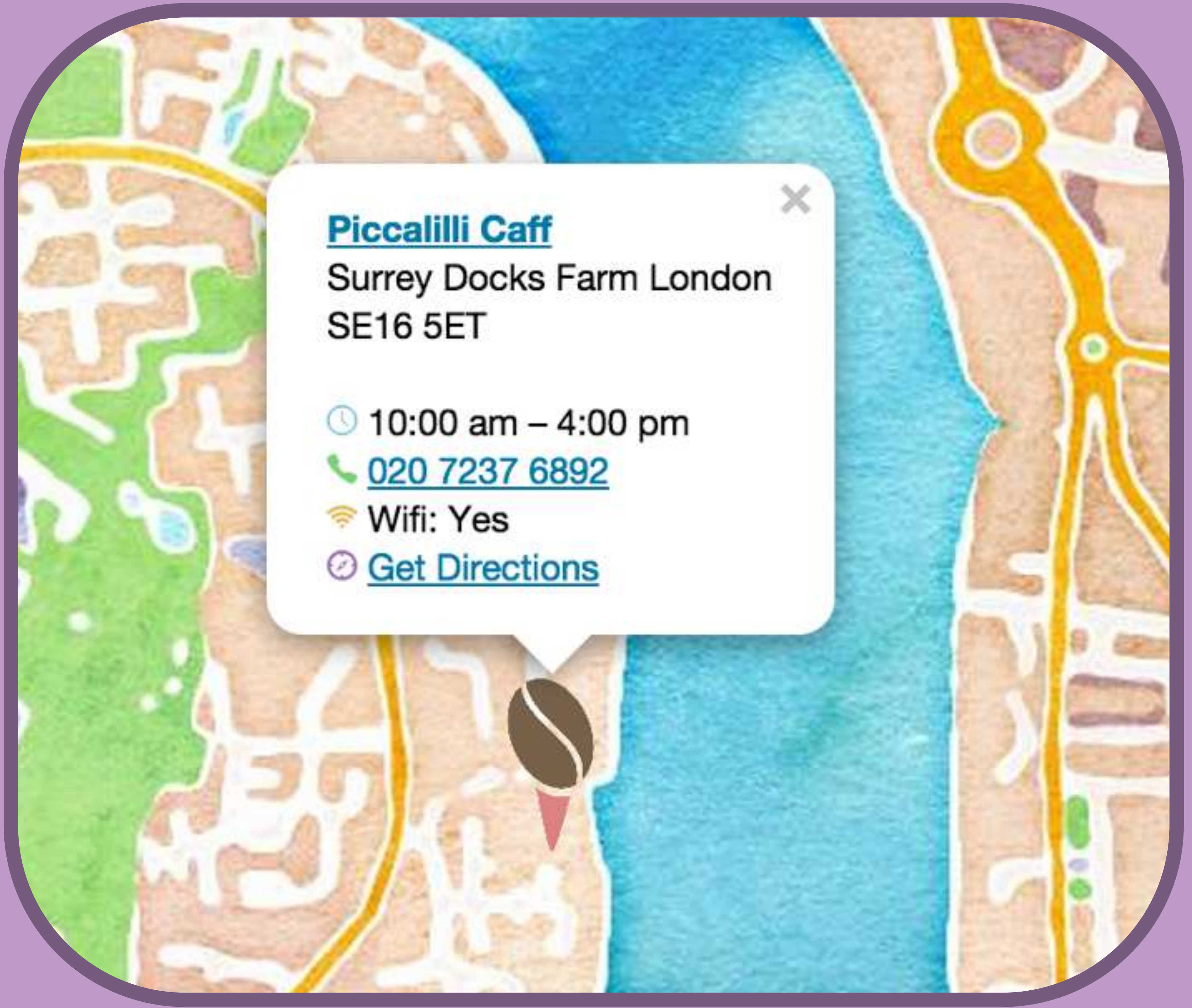


Cutter Lane, West Parks

1 km, 3 min 41 s

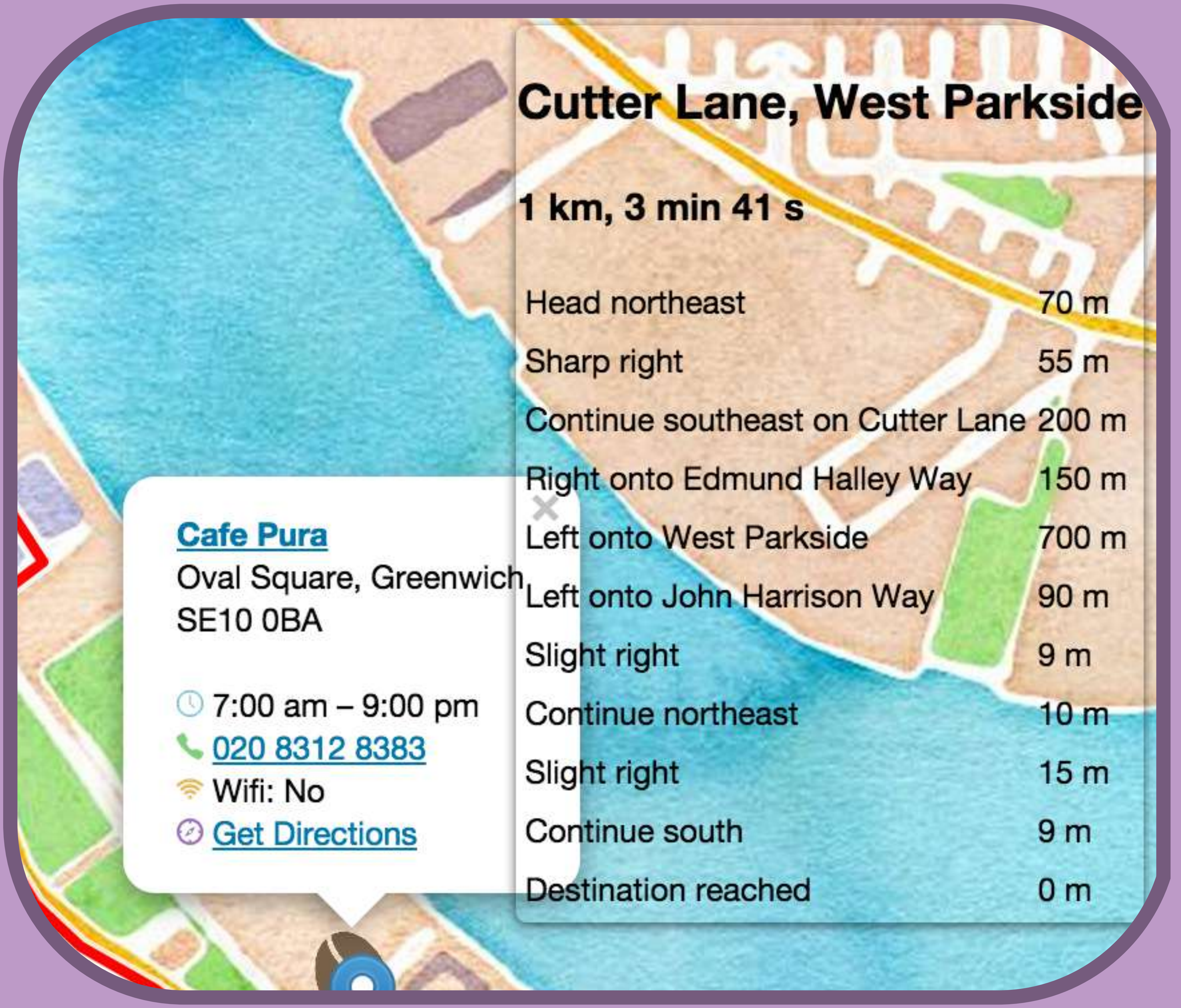
Head northeast
Sharp right
Continue southeast on Cutter Lan
Right onto Edmund Halley Way
Left onto West Parkside
Left onto John Harrison Way
Slight right
Continue northeast
Slight right
Continue south
Destination reached

Cafe Pura
Rating: 5
Get directions



Piccalilli Caff
Surrey Docks Farm London
SE16 5ET

10:00 am – 4:00 pm
020 7237 6892
Wifi: Yes
Get Directions



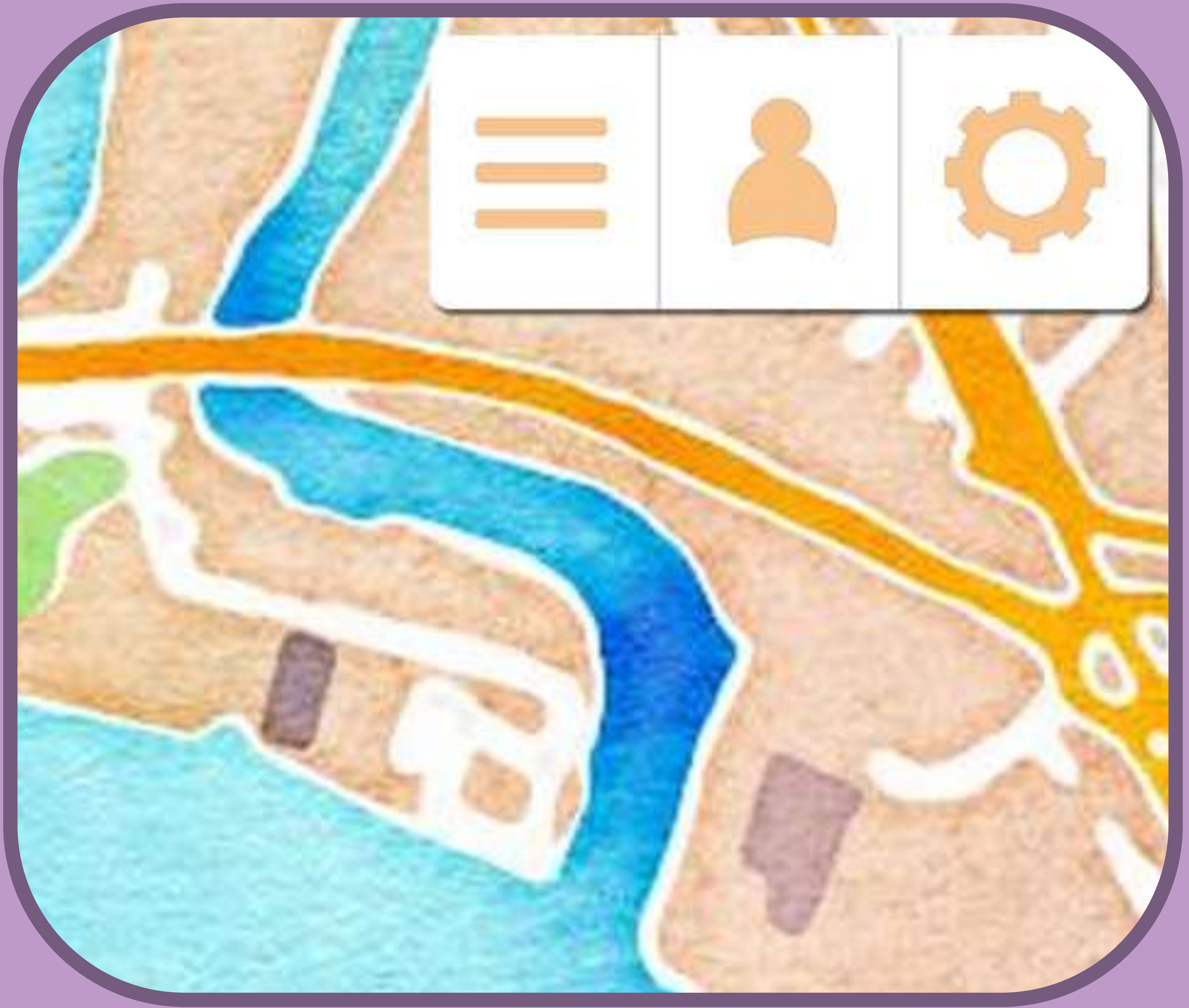
Cutter Lane, West Parkside

1 km, 3 min 41 s

Head northeast	70 m
Sharp right	55 m
Continue southeast on Cutter Lane	200 m
Right onto Edmund Halley Way	150 m
Left onto West Parkside	700 m
Left onto John Harrison Way	90 m
Slight right	9 m
Continue northeast	10 m
Slight right	15 m
Continue south	9 m
Destination reached	0 m

Cafe Pura
Oval Square, Greenwich
SE10 0BA

7:00 am – 9:00 pm
020 8312 8383
Wifi: No
Get Directions



Development

Removing the directions list box presented some difficulties. I thought I had removed it and wanted to focus on giving the user the ability to view it if they wanted but it was still showing up in all browsers other than chrome but with a transparent background. Although I feel that this is really important I didn't want to waste time dwelling on something I found extremely difficult to fix. However I did discover that it is referred to as itinerary in the code and will work on this at a later point.

The next bit I started working on was adding a small menu, I worked in photoshop to develop something that I thought would fit with the style already in use by my app, leaflet, the watercolour tiling from Stamen and the existing zoom buttons on the top left. After originally designing the icons in the menu to be black like the zoom buttons I didn't like this ascetic so decided to go with the main colour of the app and choose orange, then made these menu buttons stick to the right hand side but stay in line with the zoom buttons. The icons I chose represent:

List view - An option which will show you the coffee shops in a list.

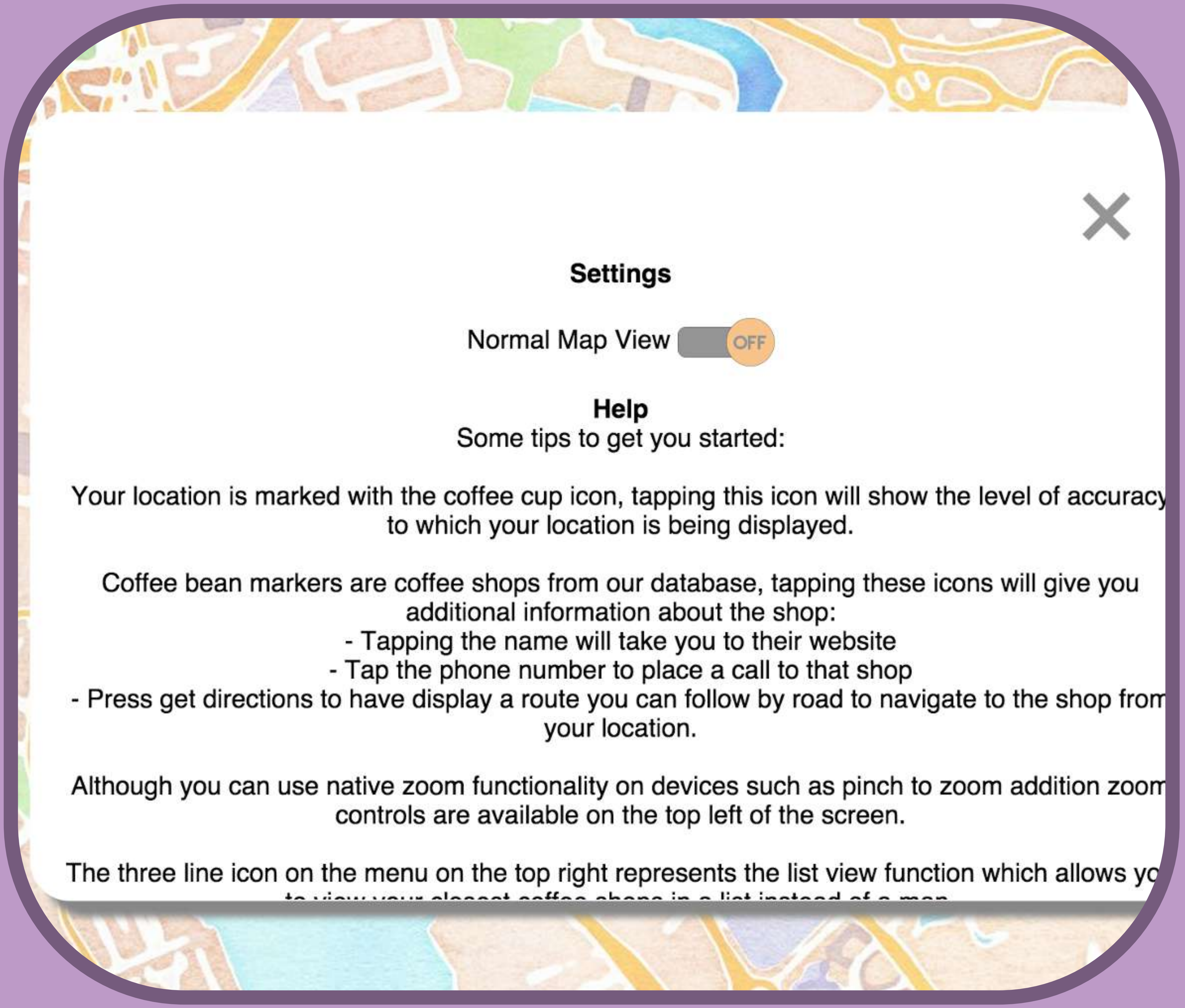
Profile - Your account which will have details such as your loyalty card and your email address.

Settings — Mainly accessibility settings that I have decided to include, although the styling of the map looks great it's not particularly clear to somebody who might have a visual impairment.

Development

I then needed to work on making this menu work, I decided that for the profile and setting menus I wanted a light-box style popup to appear over the screen so I made a large translucent div which overlays the application when the button is pressed using JS and then a smaller rounded corners div on top which drops a shadow to display the content in. I chose this style for both the profile and settings popup as I felt it suited the purpose but for the list view made it a full screen white. I added a grey cross which I tried to fit with the style that leaflet natively uses to give you the ability to close the popups although with the light box ones you can just click off to close but this isn't intuitive for all users.

The first menu I decided to make was the settings one, the only setting I could think to add so far was to switch the map design to a more traditional clearer map for people who may have difficulty viewing the one I've been using by default. I was going to load this using javascript but instead made a link to an identical page with a different set of map tiles (these map tiles are from MapBox) loaded up as this was simple and didn't require the index page to load up 2 sets of map tiles which could have increased my load time. I also included some basic tips on app functionality in this menu as it seemed the most appropriate place and I had space I could use left on the popup.


×

Settings

Normal Map View OFF

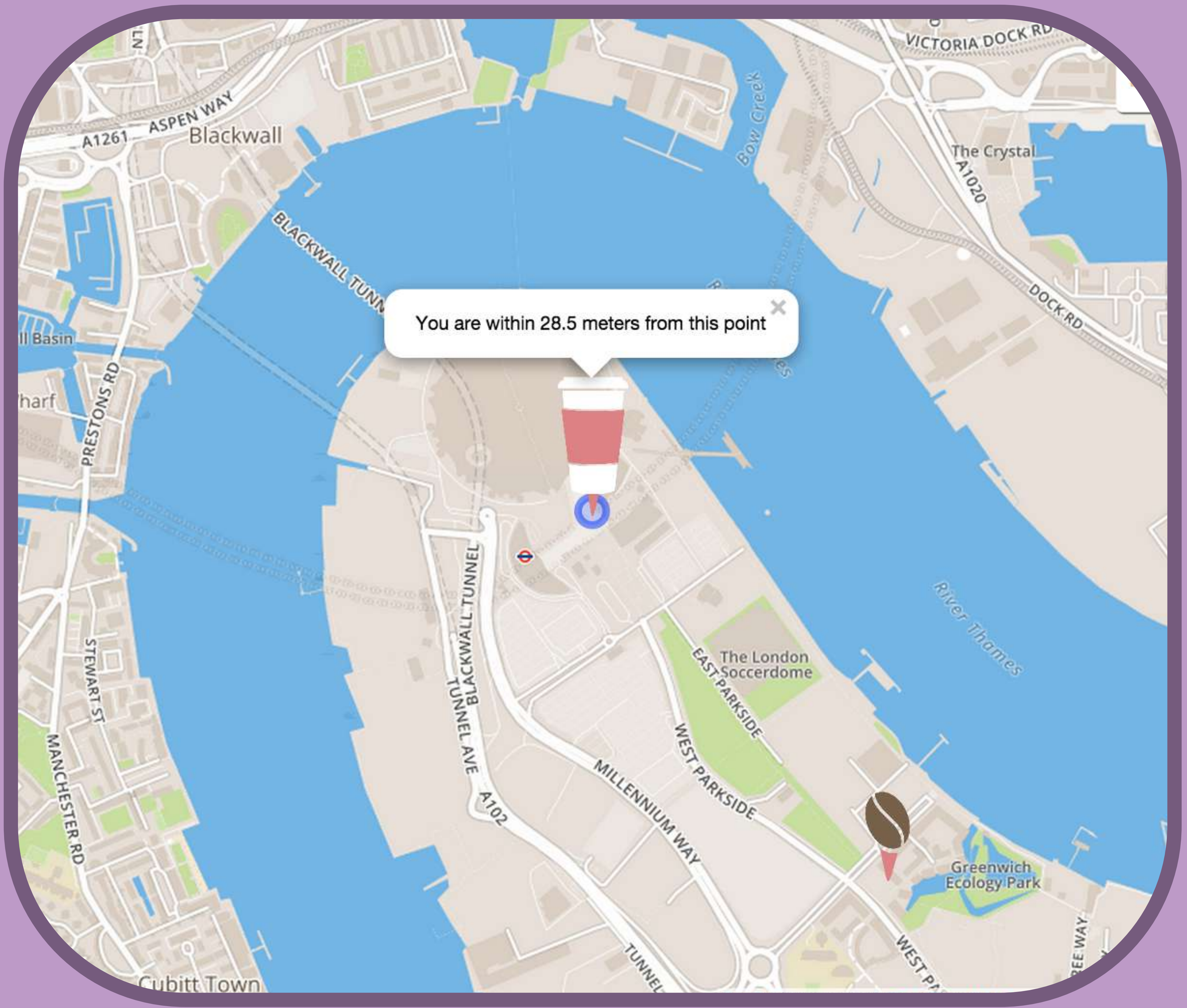
Help
Some tips to get you started:

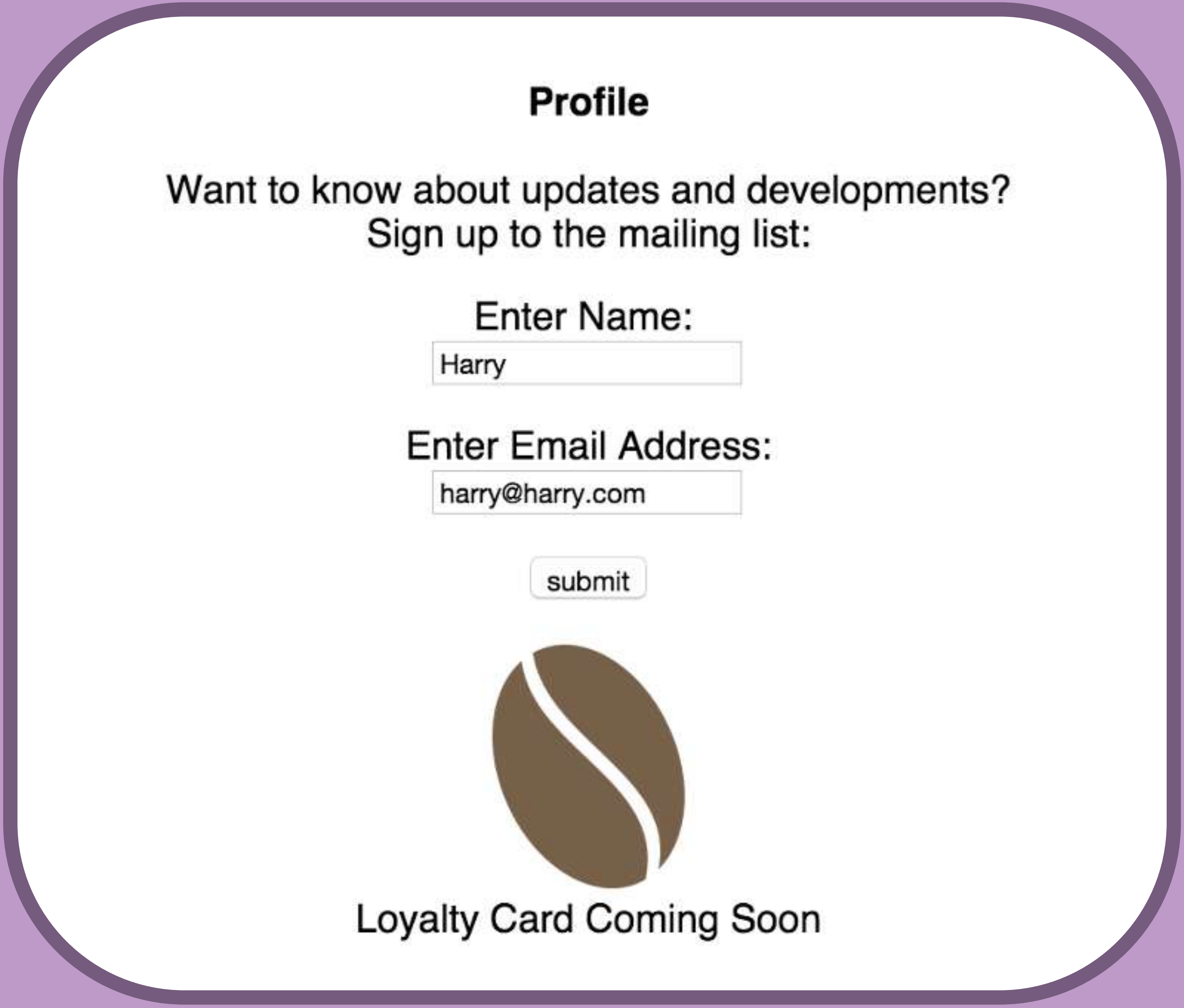
Your location is marked with the coffee cup icon, tapping this icon will show the level of accuracy to which your location is being displayed.

Coffee bean markers are coffee shops from our database, tapping these icons will give you additional information about the shop:
- Tapping the name will take you to their website
- Tap the phone number to place a call to that shop
- Press get directions to have display a route you can follow by road to navigate to the shop from your location.

Although you can use native zoom functionality on devices such as pinch to zoom addition zoom controls are available on the top left of the screen.

The three line icon on the menu on the top right represents the list view function which allows yo to view your closest coffee shops in a list instead of a map.


You are within 28.5 meters from this point



Profile

Want to know about updates and developments?
Sign up to the mailing list:

Enter Name:
Harry

Enter Email Address:
harry@harry.com

Loyalty Card Coming Soon

THANKS!
YOU'RE NOW
SIGNED UP

Development

I then began to work on the profile popup. This popup will eventually contain the loyalty card functionality that my app hopes to implement in future iterations but as doing this scheme will involve negotiations with coffee shops and creating a secure method of implementing this instead for the moment there is a message letting the user know that this will be available in the future. The mailing list is something I put into the app with help from html-form-guide, I know how to make a basic form but wanted to add fully working functionality so that the form sends an email containing the users name and email address to a specified email with validation points such as requiring both a name and email to be entered and the email to be valid so I used code templates to help out with that. Once you have submitted your email address the form sends you to a Thank You page for 3 seconds before looping back around to the page you were on before. This form now works and sends the resulting email to my personal email address.

Development

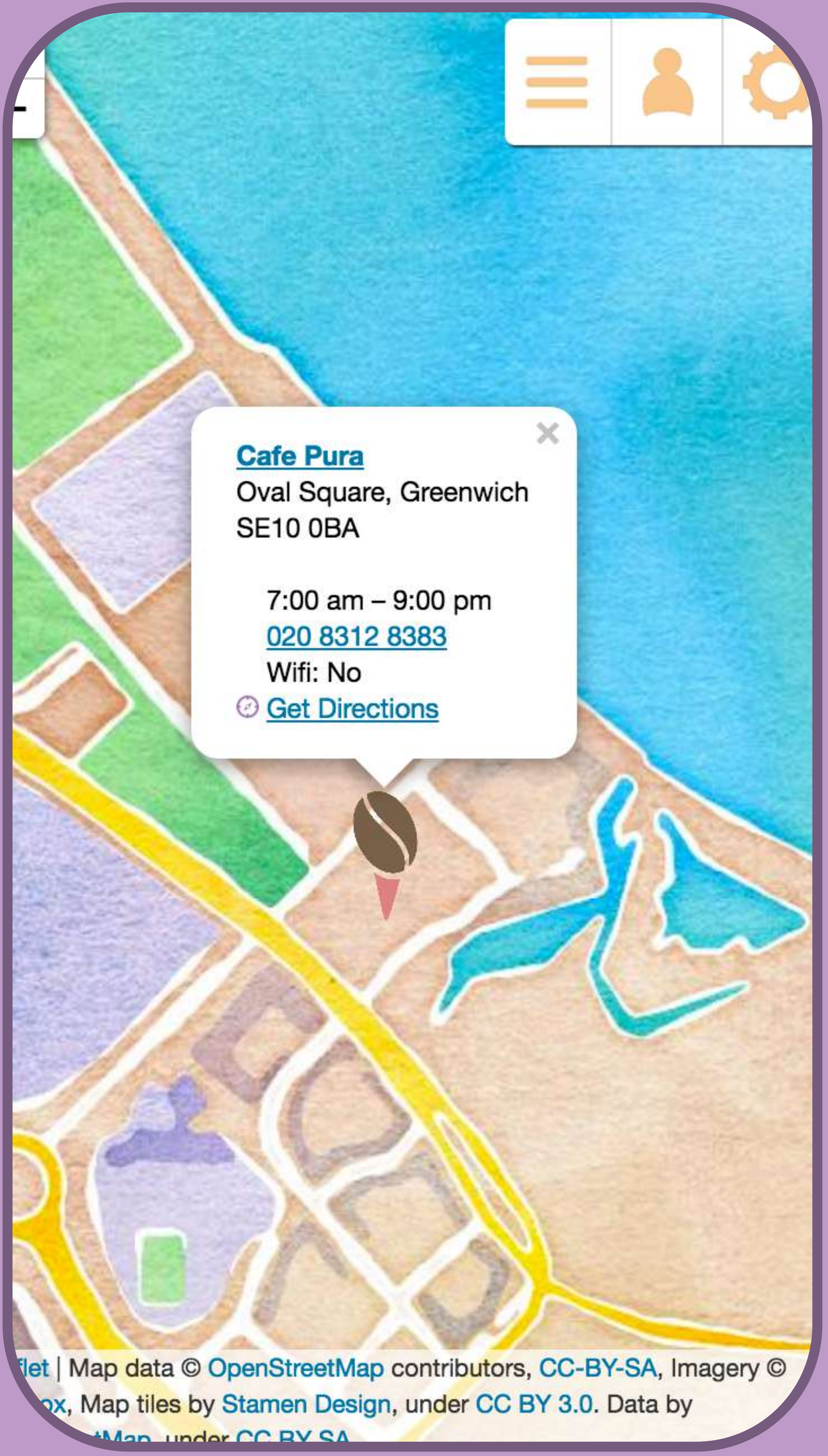
I managed to remove the itinerary by trialling “display:none” on various options in the css as although I seemed to be able to remove it in the javascript it still kept displaying. This made the directions function a whole load more useful as you can use it unobstructed on a mobile phone and gives you a much clearer view. What I then wanted to do was to add the option to view this list of directions if you would like so I added a box to appear when get directions is hit; this shows up in the top right underneath the menu. The style of the popup for this and the list view of coffee shops is the same.

The final things I needed to add were:

- Toggling list view of shops on and off (and sorting by distance).
- List view for directions
- Continuing to add as many shops as possible

I also worked on testing the site on mobile as I went along to ensure that all of the functionality works at least on iOS and that there’s no errors on mobile devices.

Screenshots of the final stages of development are on the following page.

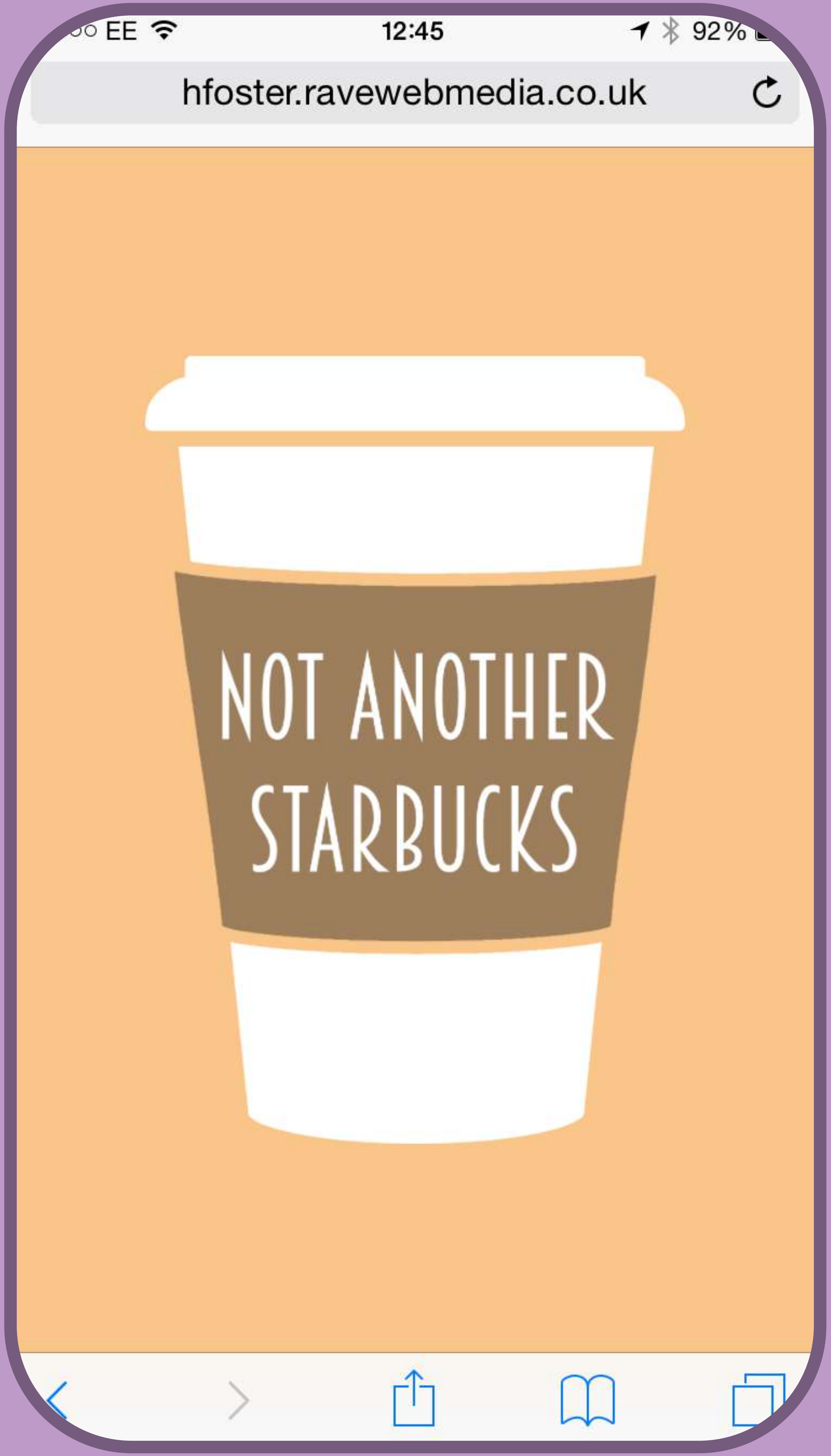


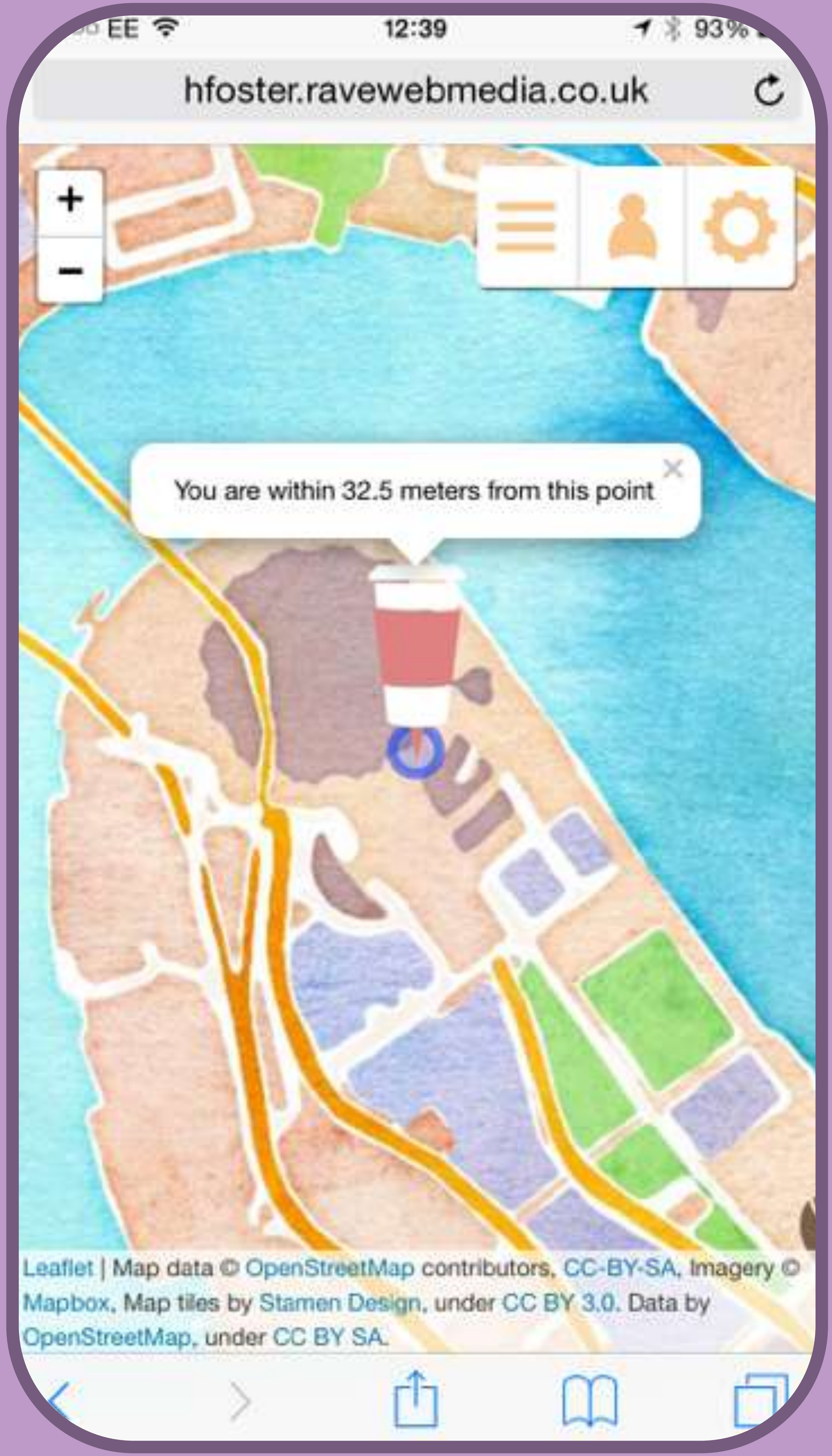
Cafe Pura
Oval Square, Greenwich
SE10 0BA

7:00 am – 9:00 pm
020 8312 8383
Wifi: No
Get Directions

flet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery ©
ox, Map tiles by Stamen Design, under CC BY 3.0. Data by
tMap, under CC BY SA

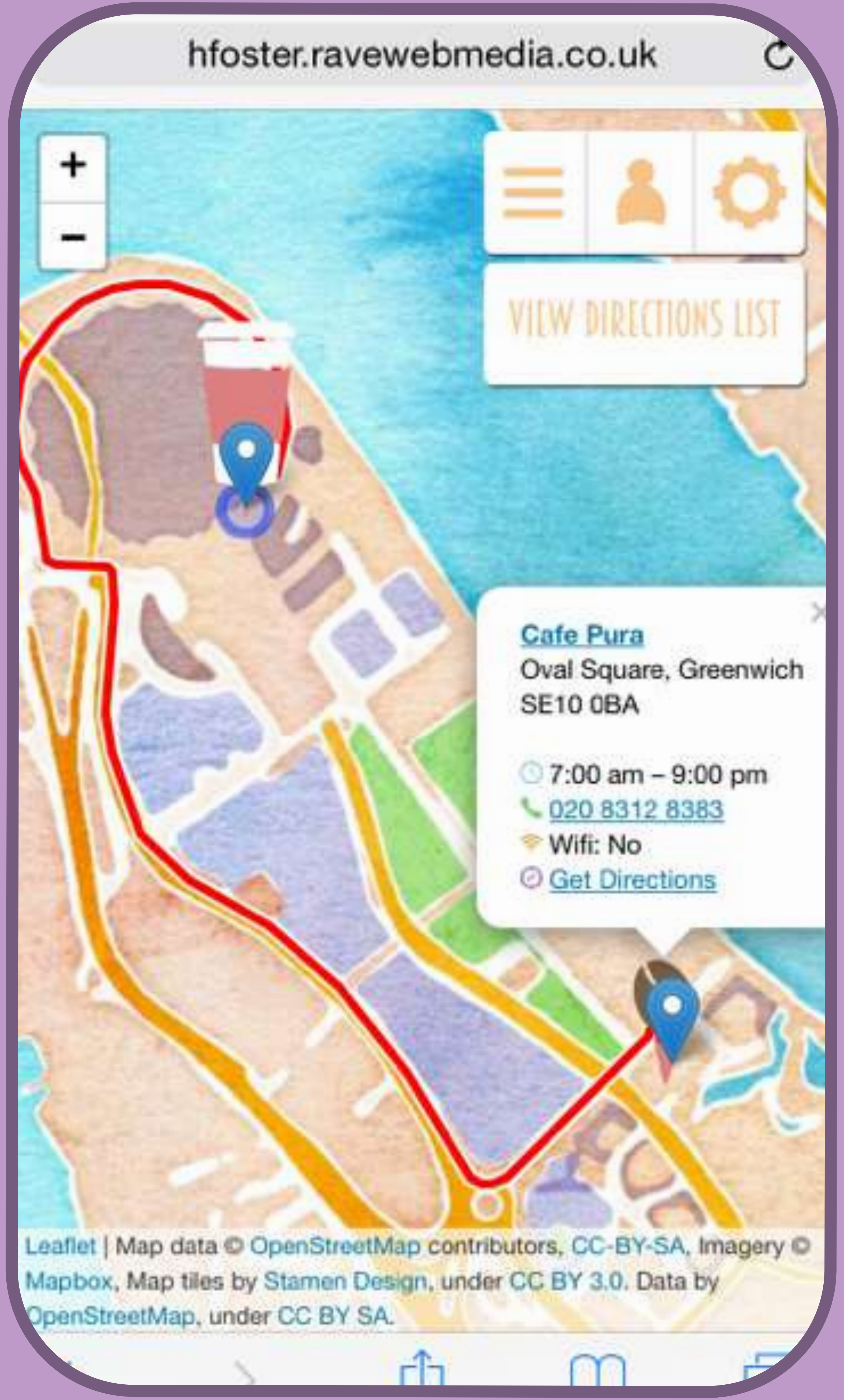
Mobile Screenshots





You are within 32.5 meters from this point

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox, Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

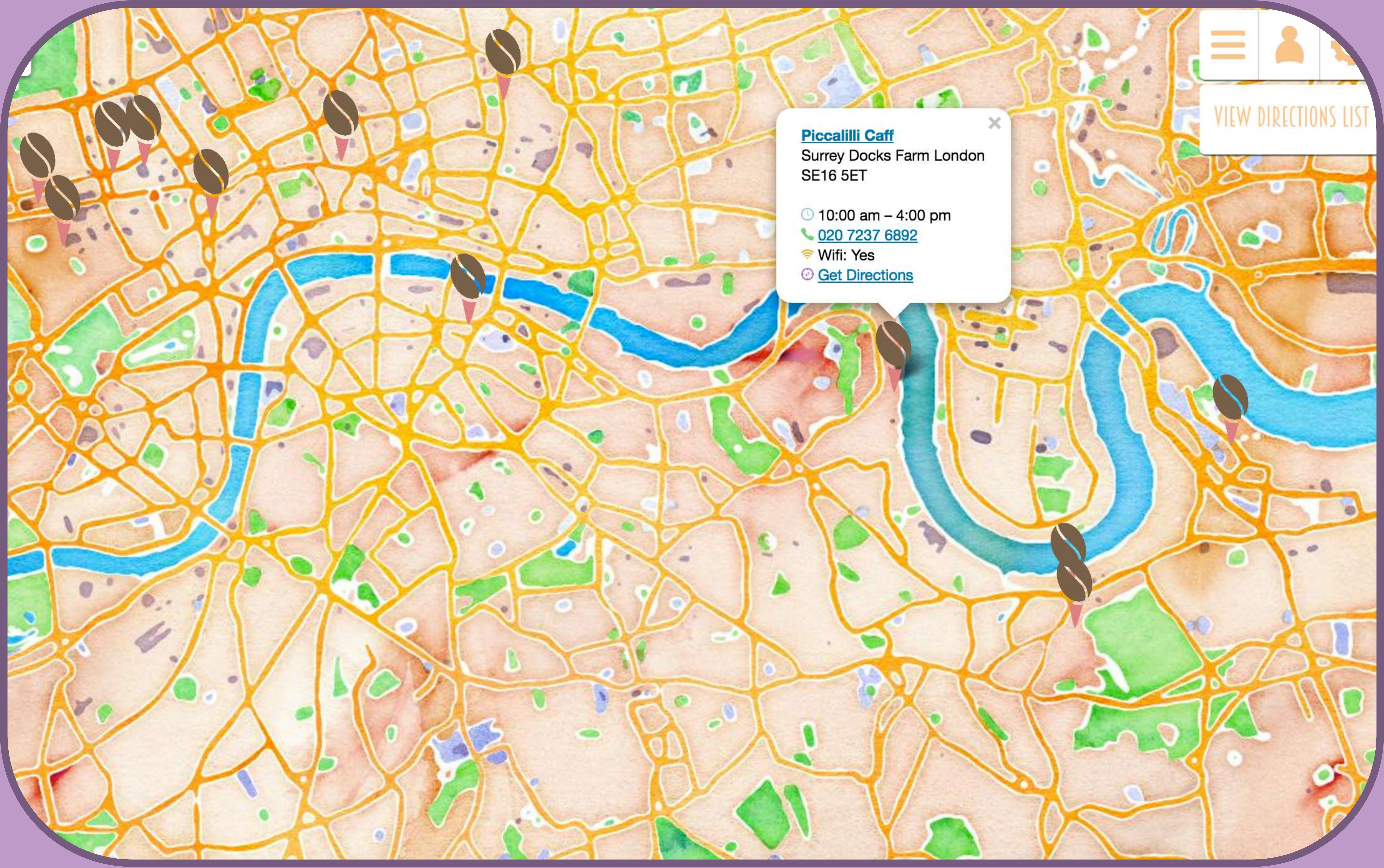


Cafe Pura
Oval Square, Greenwich
SE10 0BA

7:00 am – 9:00 pm
020 8312 8383
Wifi: No
Get Directions

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox, Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Desktop Screenshot



Piccalilli Caff
Surrey Docks Farm London
SE16 5ET

🕓 10:00 am – 4:00 pm
☎ 020 7237 6892
Wifi: Yes
Get Directions

Final Major Project

Initial Ideas

For my final major project I am going to begin the development process of this application again from scratch, I will learn from what I did the first time around however not make assumptions based on it.

For this project I am going to conduct focus groups and interviews with independent coffee retailers in order to design a product which is as attractive to them as possible, I am going to conduct multiple rounds of user testing and I intend to fully implement all aspects of the application.

I am going to set out some initial aims for my project which will be subject to change once I have conducted research as this may validate some of my aims, invalidate them or create new ones. My initial aims are:

- To create an application which allows users to locate independent coffee shops which are signed up to my service.

- Give users relevant information about the coffee shops they find such as menu, opening times, contact numbers etc.

- Include navigation features to ensure that users can easily find their way to their chosen coffee shop.

- Include a loyalty system which unites independent coffee shops so that a user has an incentive to continue to use the application and visit independent retailers instead of chain coffee shops.

- Be usable on a variety of mobile devices so that anybody with a smart phone can use the service and that a user isn't tied to a particular operating system.

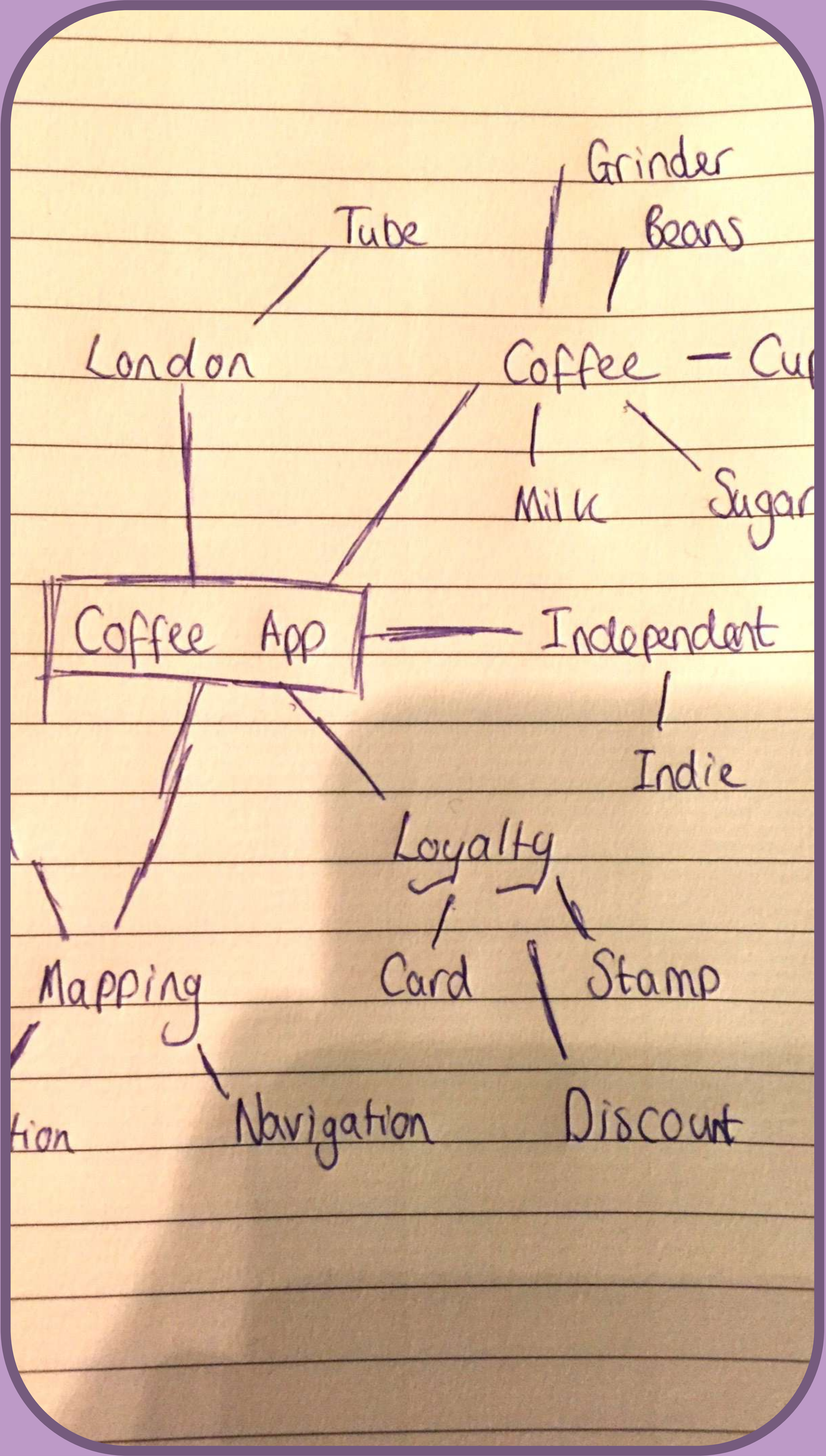
- Have an effective, intuitive yet innovative UI.

Initial Ideas

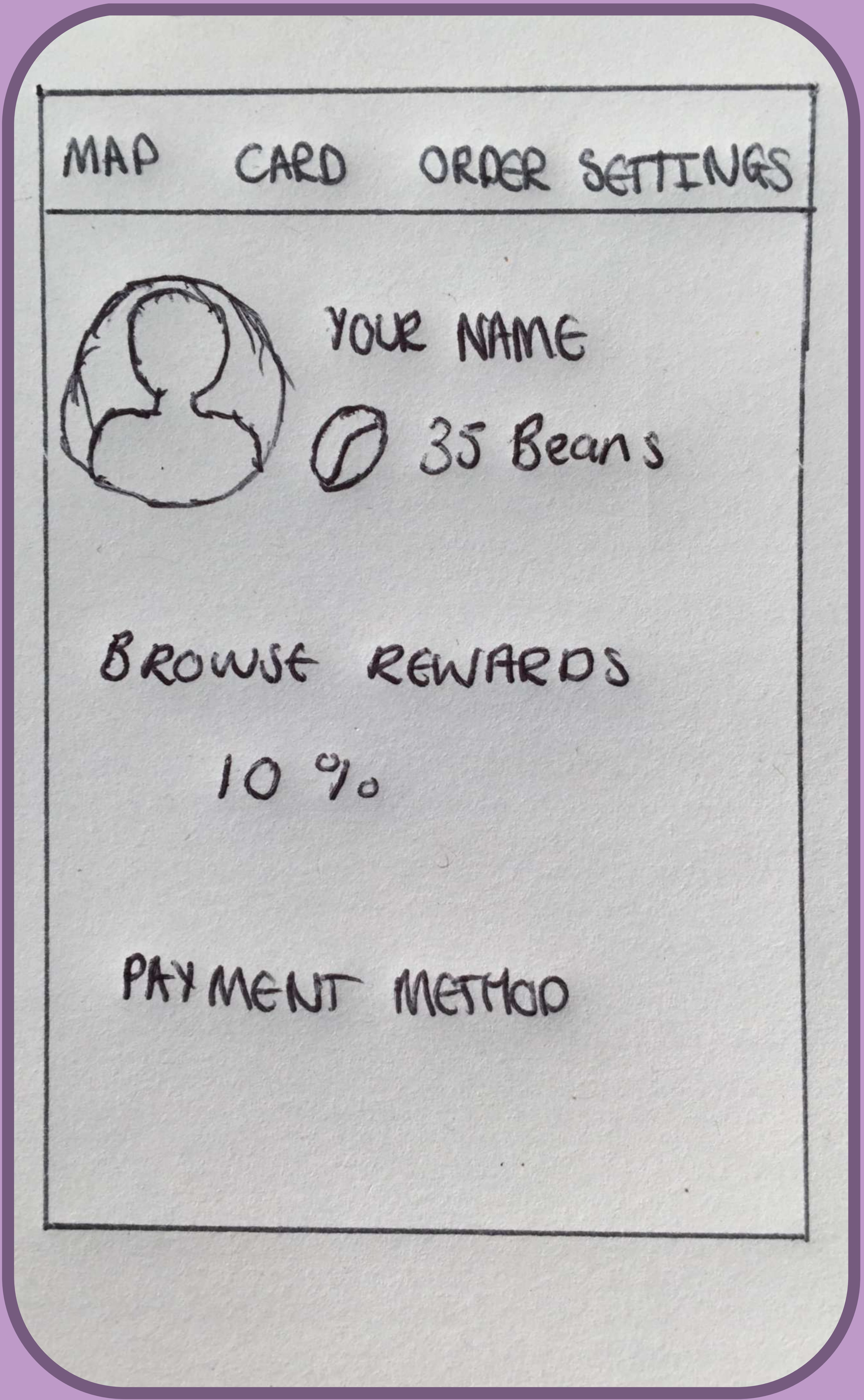
To come up with a working title I did a quick brain storm of words relating to what my application will aim to do. I have decided to use the working title of Indie Coffee as it's simple and informative.

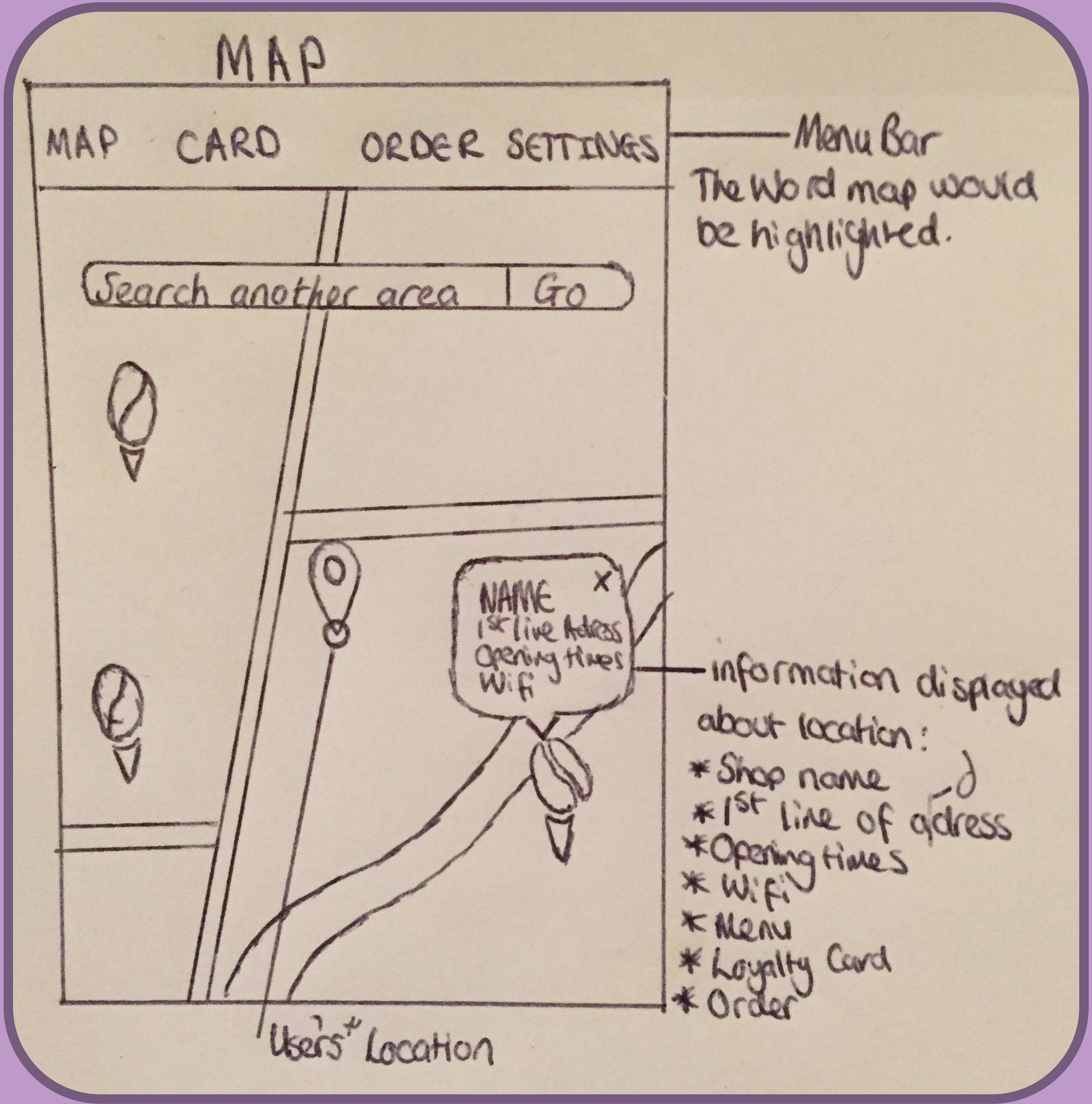
After spending some time thinking about how I can develop this application to expand the functionality beyond what I created with Not another Starbucks I have decided that I would like to add the ability for a user to order coffee using the application from one of the participating retailers. The ability to order through the application will be facilitated by a piece of hardware which participating retailers can get from me which has a different version of the application specifically created for retailers.

This retailers app will receive the orders that come in from users of the app so that the retailers can prepare the order so that when the customer arrives they can head straight to the collection point and pick it up. The retailer application is also their means of entering in information about their shop and products, whilst basic information is required from them before they are signed up and able to use the app the retailer app will allow them to create their menu online, set the prices and add pictures, set and edit their opening times and provide additional information.

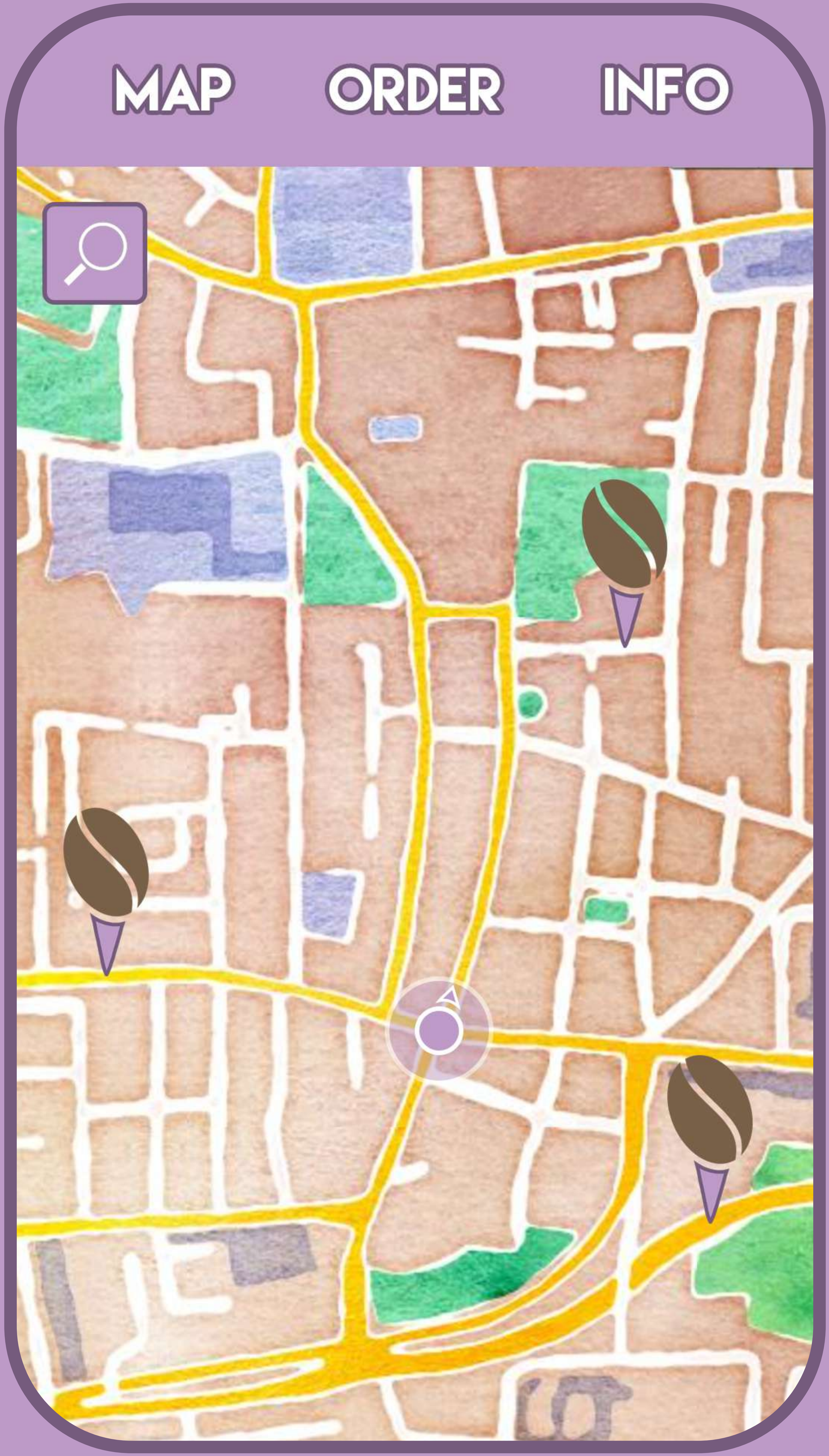


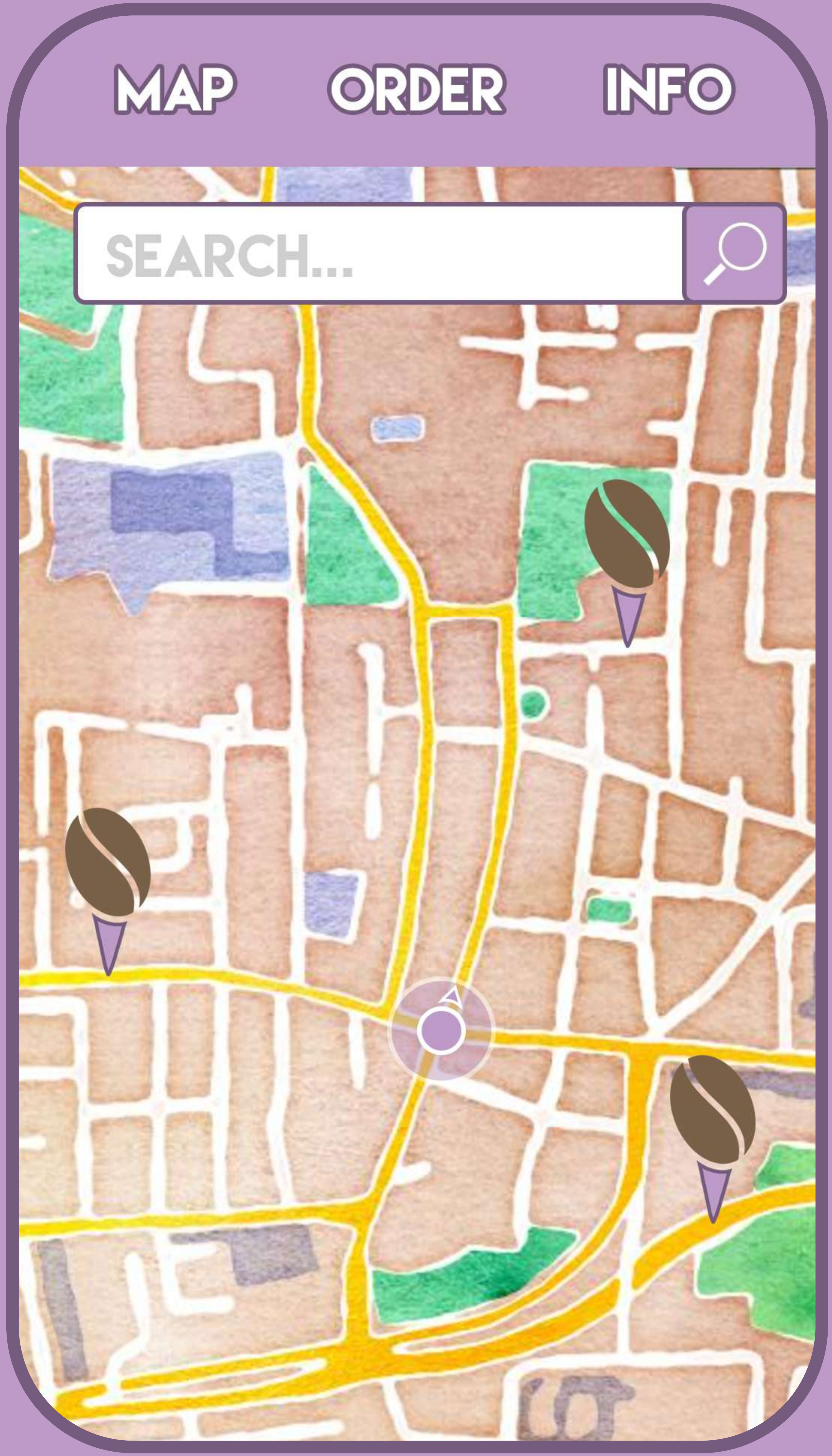
Paper Wireframes

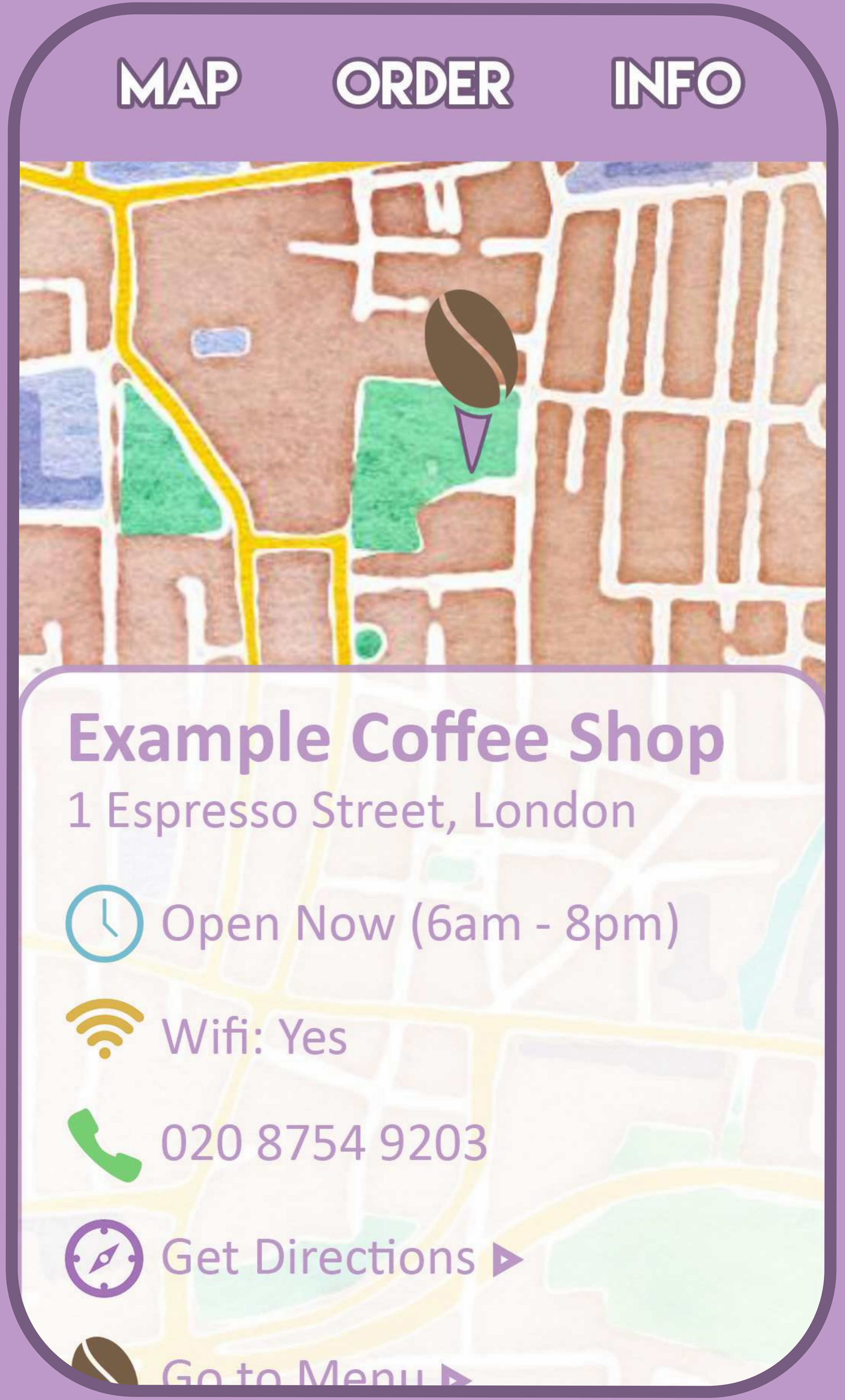




Photoshop Wireframes







Example Coffee Shop

1 Espresso Street, London

Open Now (6am - 8pm)

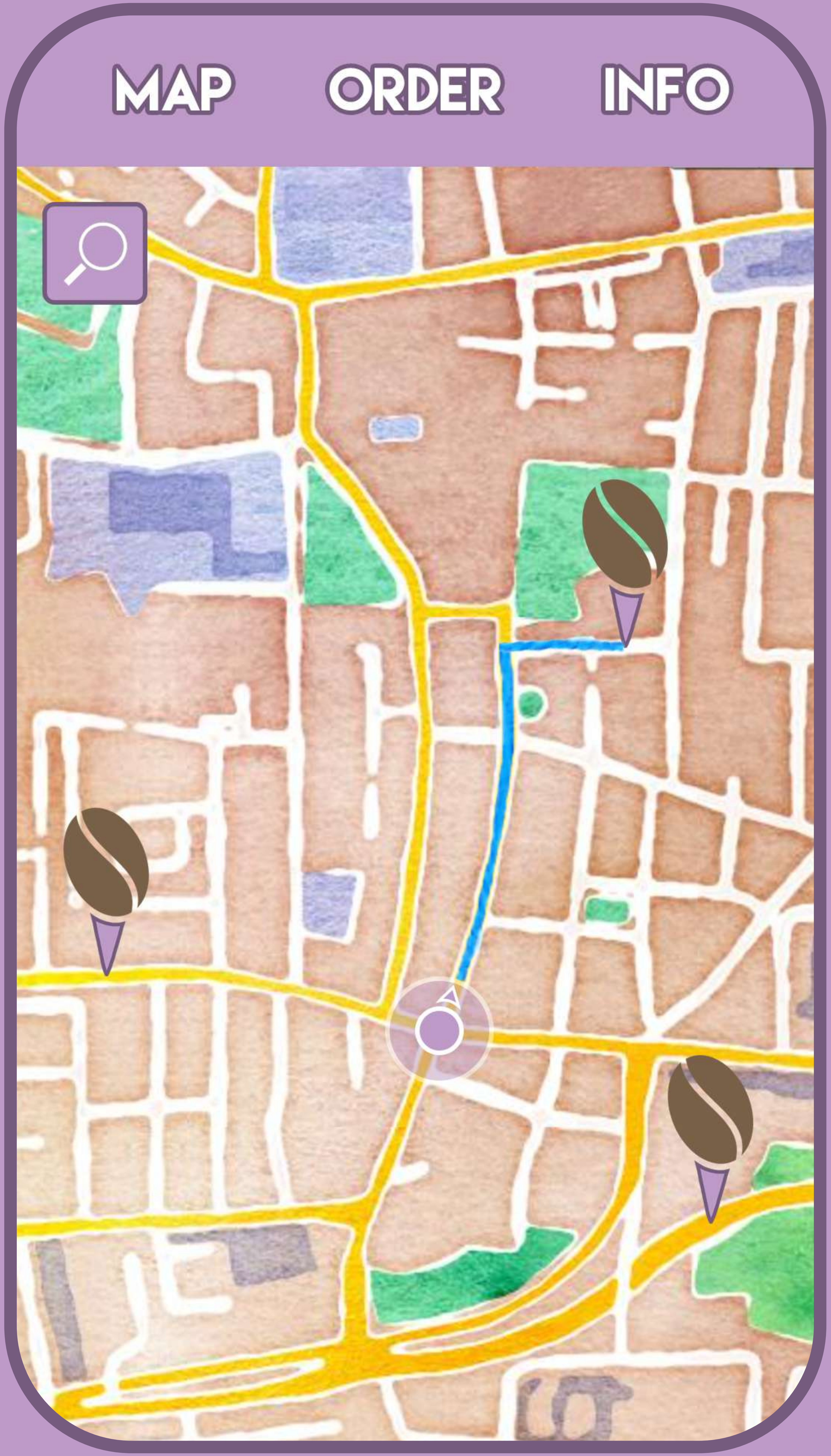
Wifi: Yes

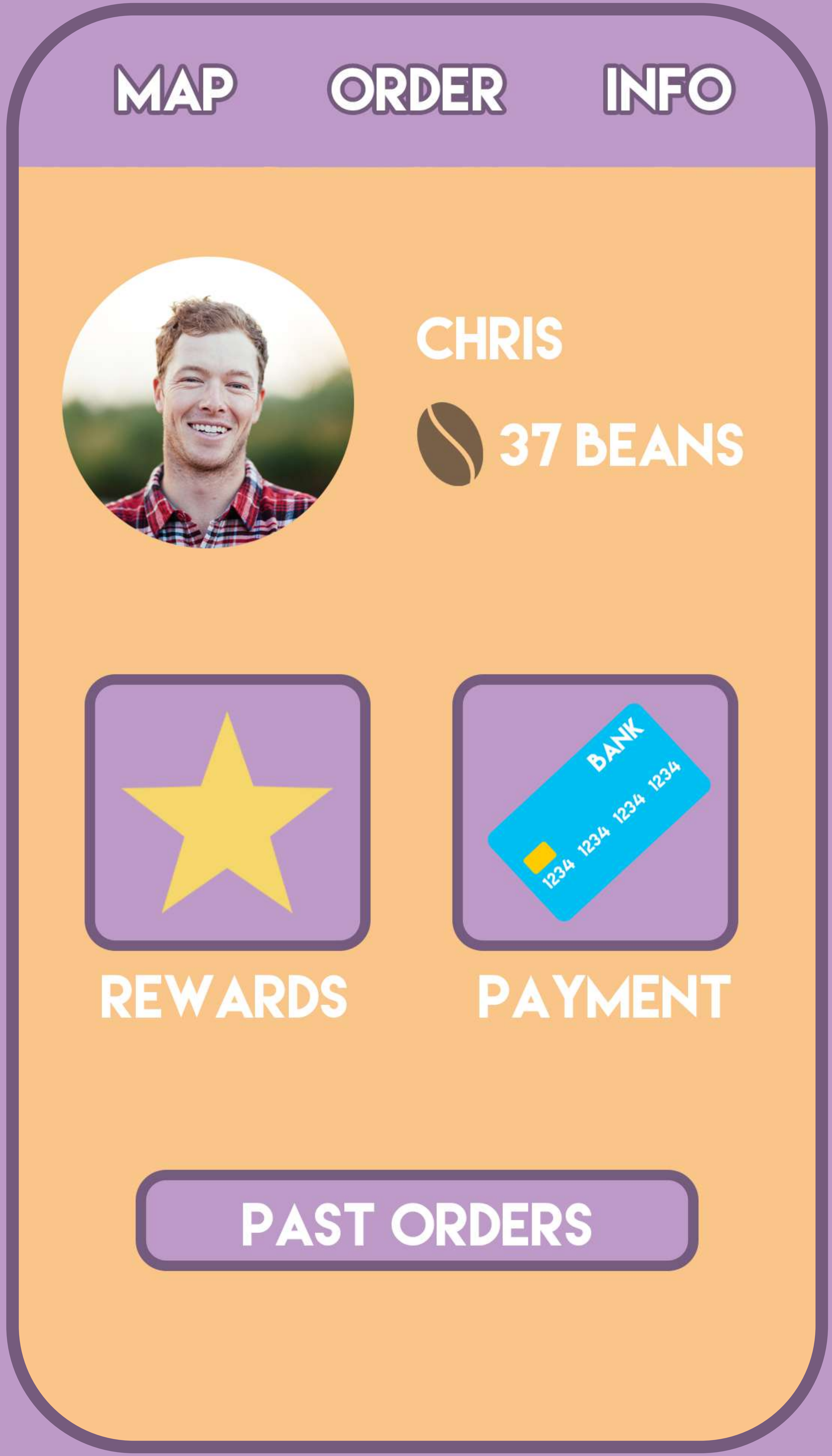
020 8754 9203

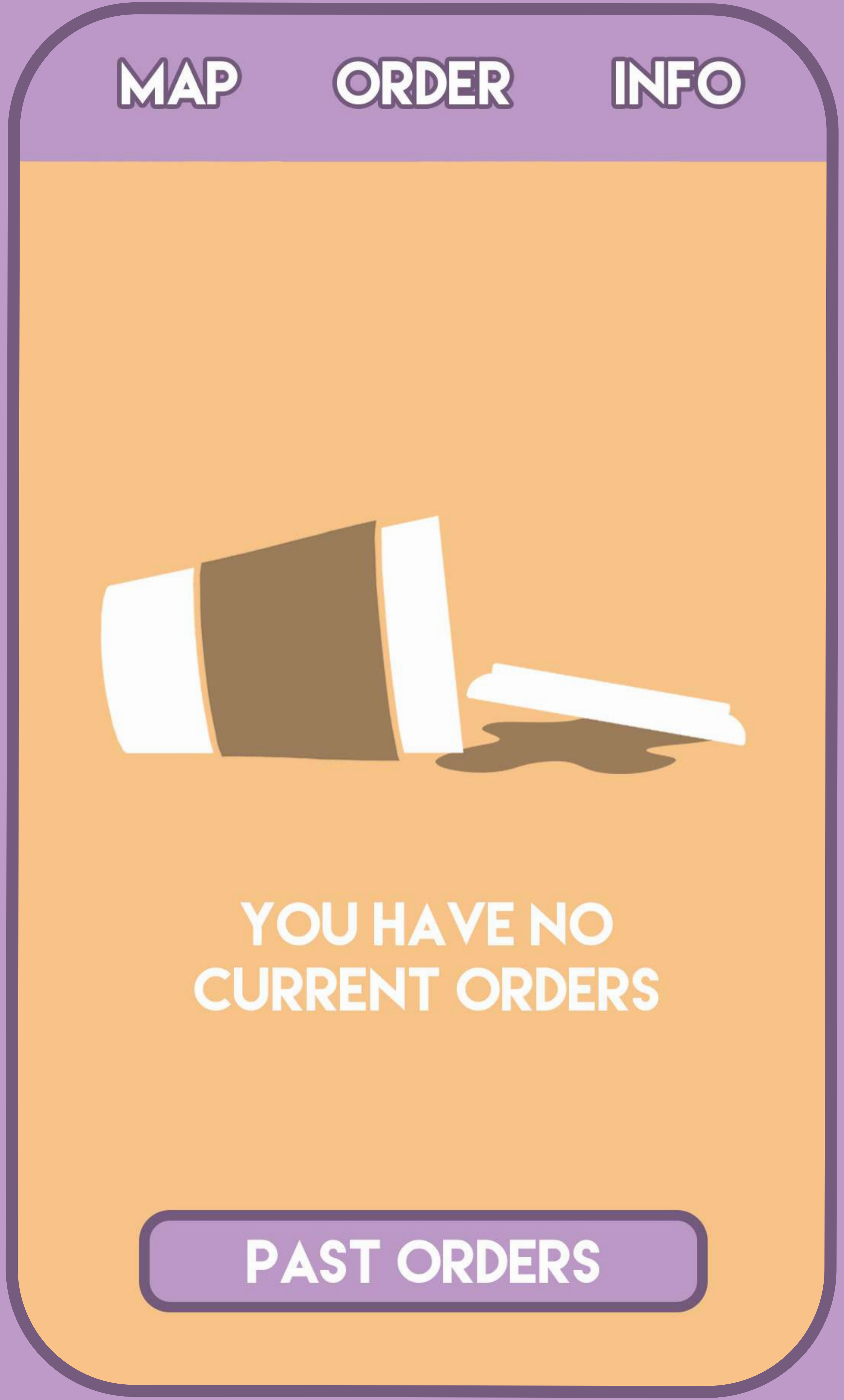
Get Directions ▶

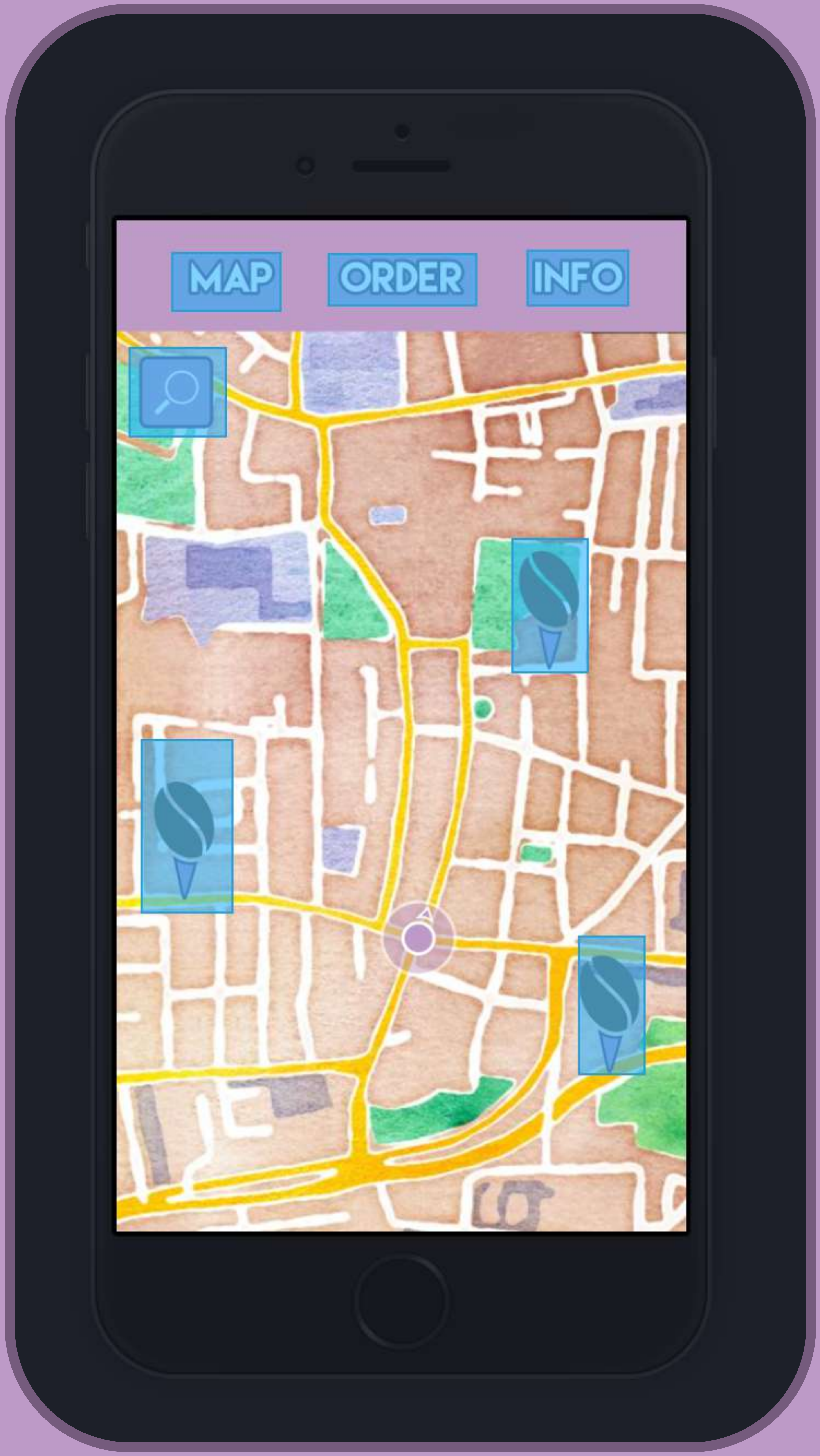
Go to Menu ▶

Photoshop Wireframes









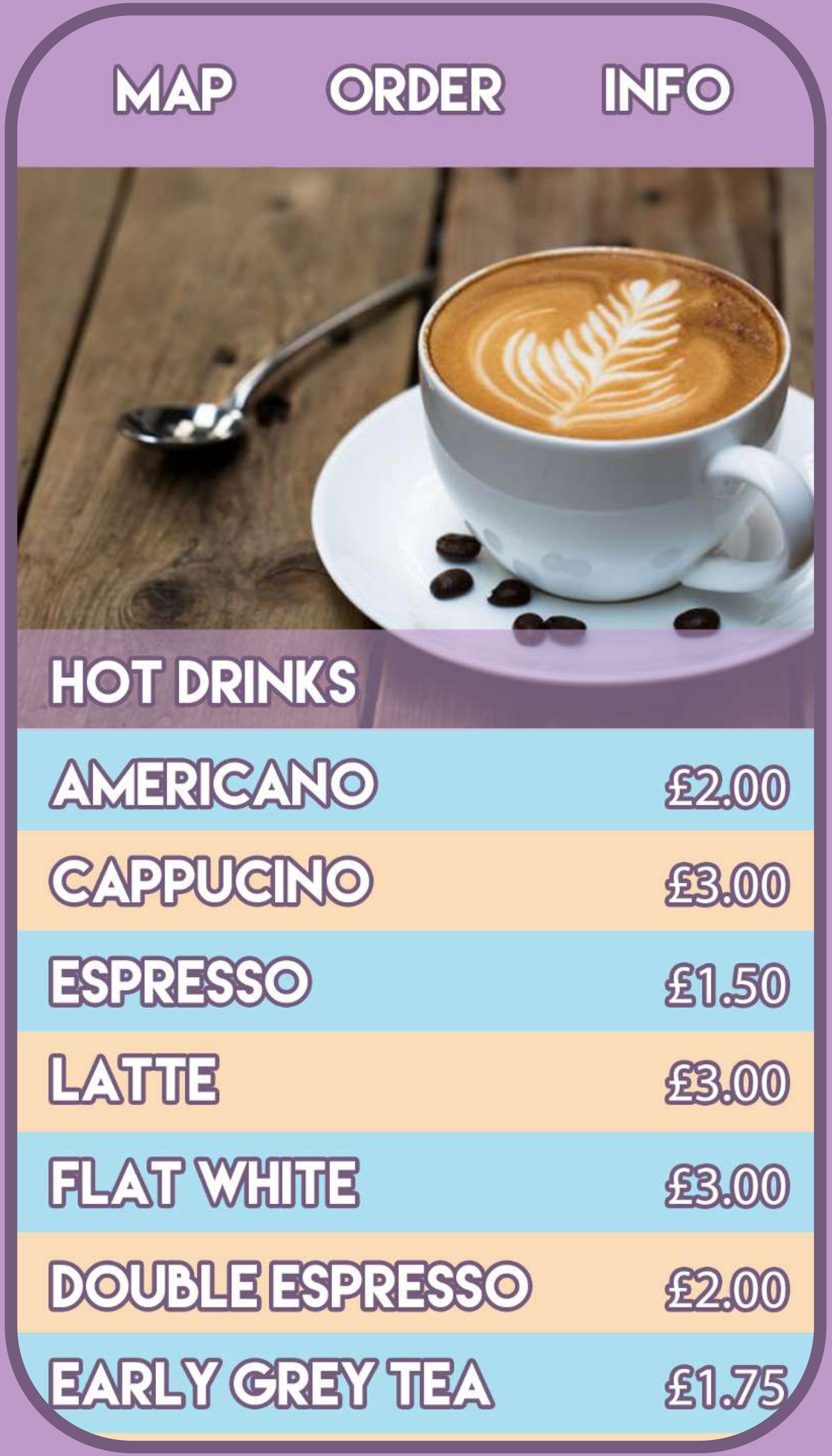
User Testing

Having these basic screenshots created meant I could make a basic wireframe map which I did in Invision to begin constructing how a user might interact with my app, it helped me get a better idea of what does work and what doesn't allow me to do user testing and see what other wireframes I might need to create.

I spent some time generating feedback on the UX of my application by getting users to work through my Invision application and provide me with feedback, I could also watch their interaction with the application which helped me to understand what a user would require. I discovered that users generally thought that the order tab at the top was to place an order rather than to view and order so i decided to change that grammatically to say orders instead. I also got some comments about the font i've used in my application being a bit harsh compared to other aspects of the design like the colour and the map. Users also didn't seem to know what they would type into the search bar I wanted them to type a location but some thought you'd be searching for a particular shop.

I expanded my Invision project to add addition wireframes based upon this initial round of user testing as I was finding the prototype I'd created didn't have quite enough depth to get useful feedback from user testing. I also decided to beginning the wire framing of the retailer app.

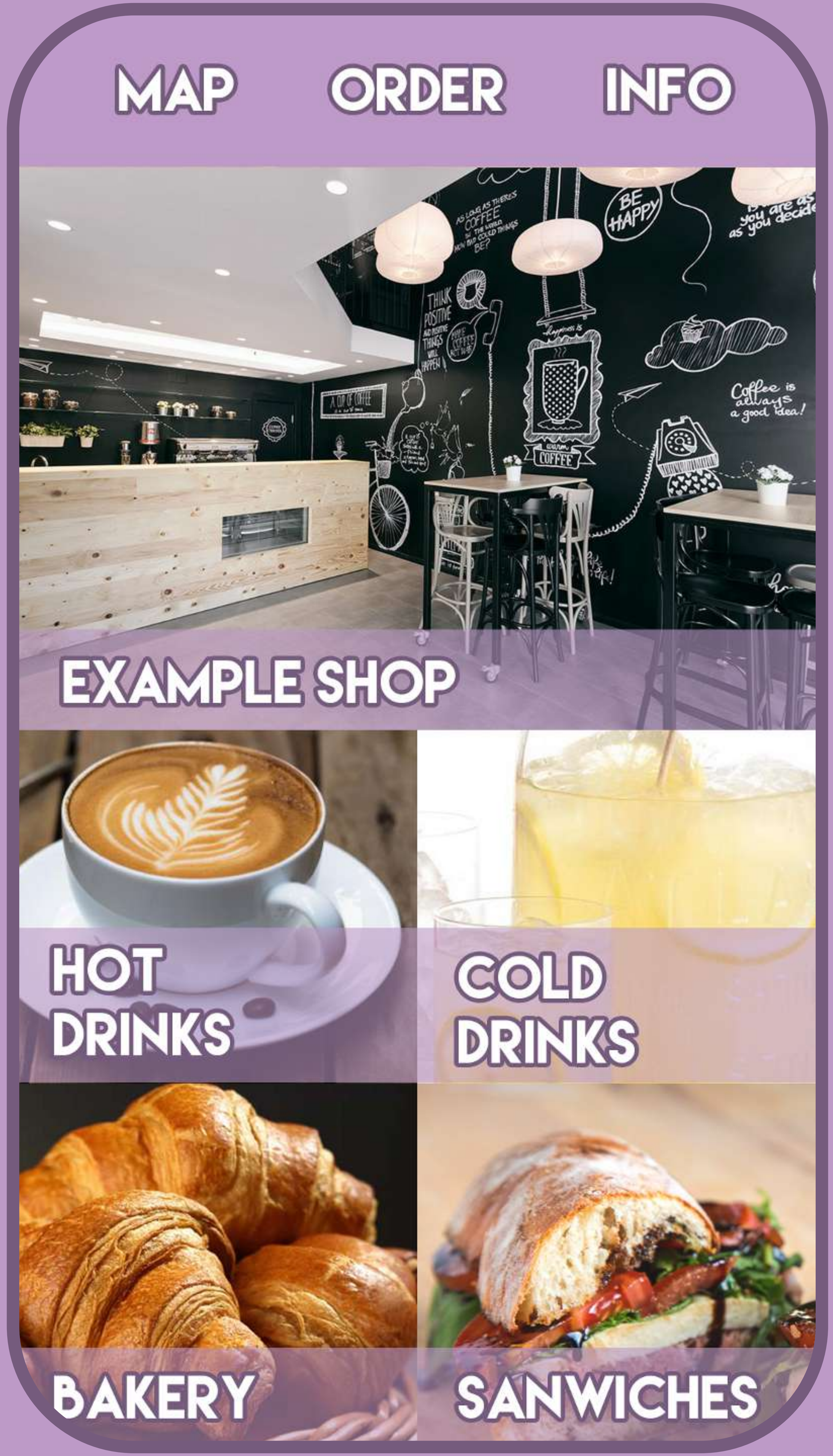
Photoshop Wireframes



MAP ORDER INFO

HOT DRINKS

AMERICANO	£2.00
CAPPUCINO	£3.00
ESPRESSO	£1.50
LATTE	£3.00
FLAT WHITE	£3.00
DOUBLE ESPRESSO	£2.00
EARLY GREY TEA	£1.75



MAP ORDER INFO

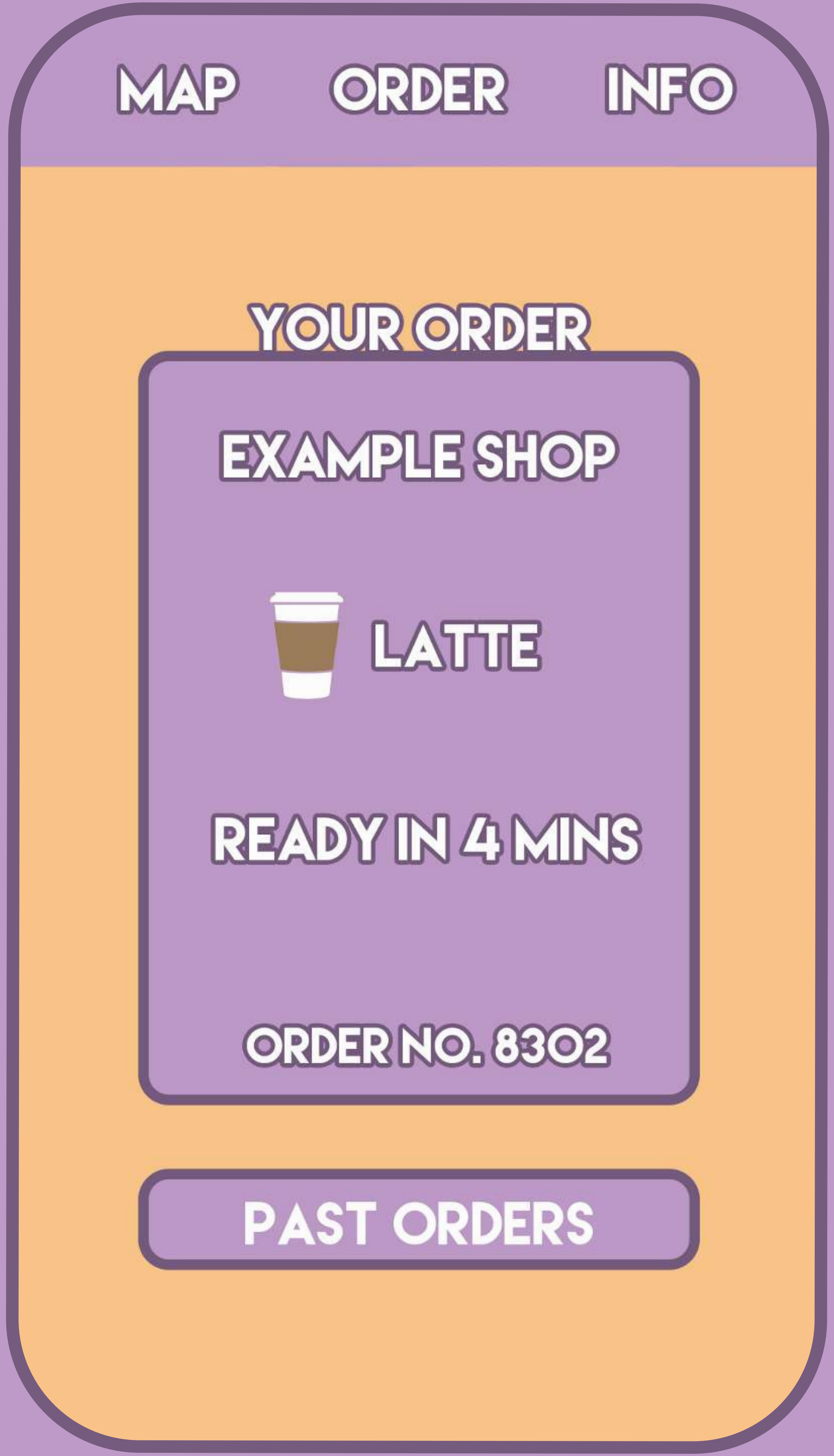
EXAMPLE SHOP

HOT DRINKS

COLD DRINKS

BAKERY

SANWICHES



MAP ORDER INFO

YOUR ORDER

EXAMPLE SHOP

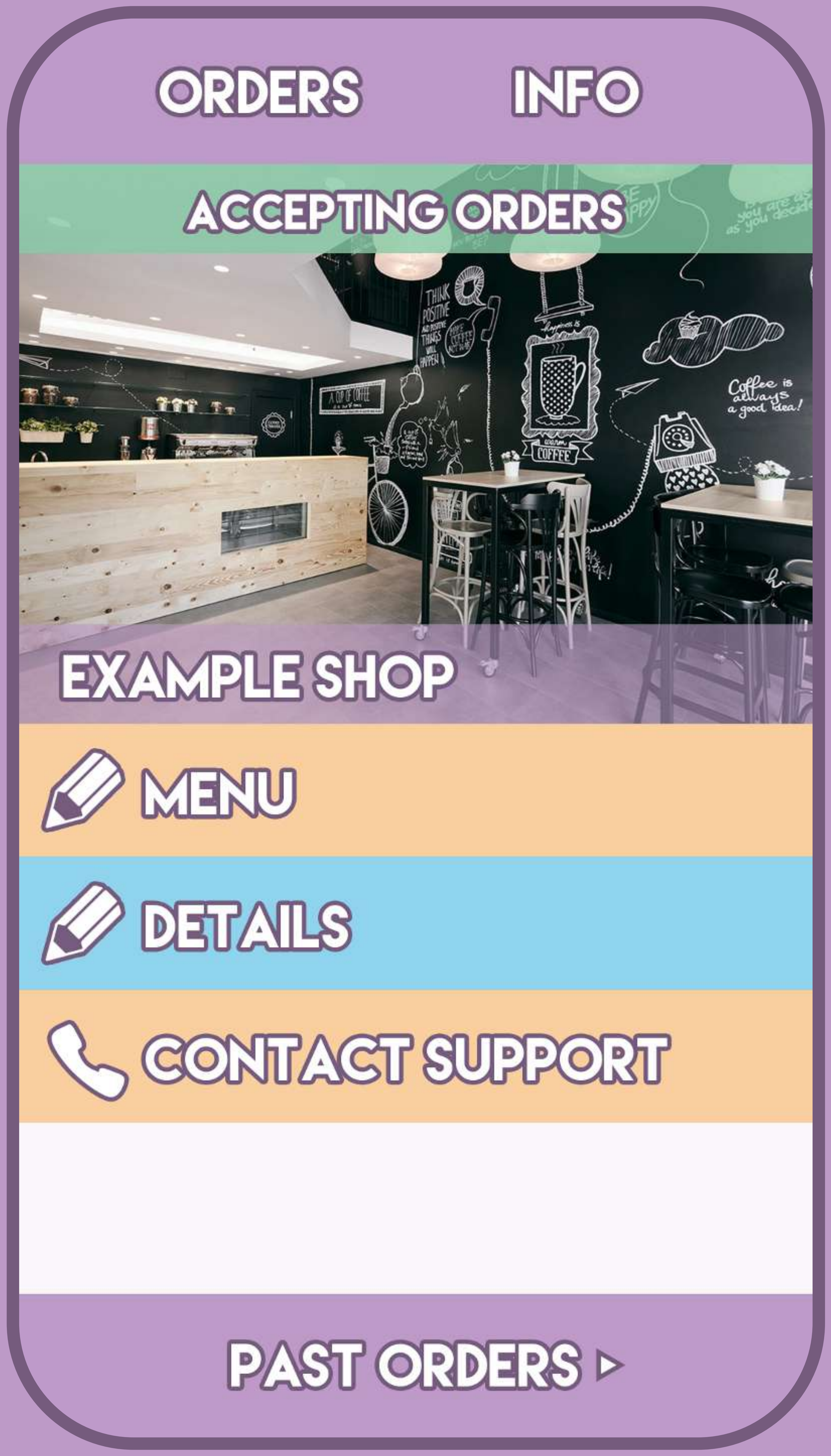
LATTE

READY IN 4 MINS

ORDER NO. 8302

PAST ORDERS

Photoshop Wireframes



ORDERS INFO

ACCEPTING ORDERS

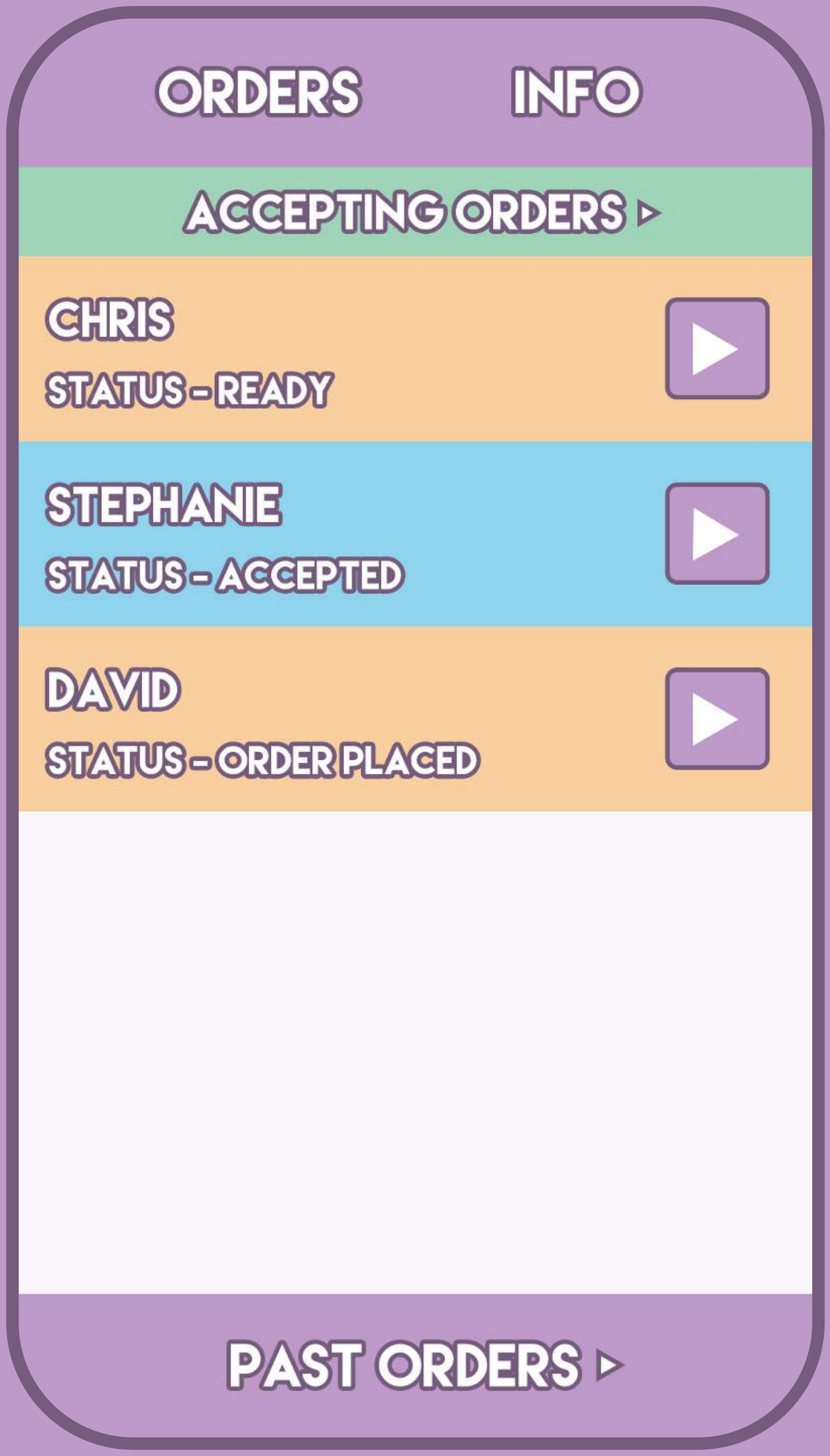
EXAMPLE SHOP

MENU

DETAILS

☎ CONTACT SUPPORT

PAST ORDERS ▸



ORDERS INFO

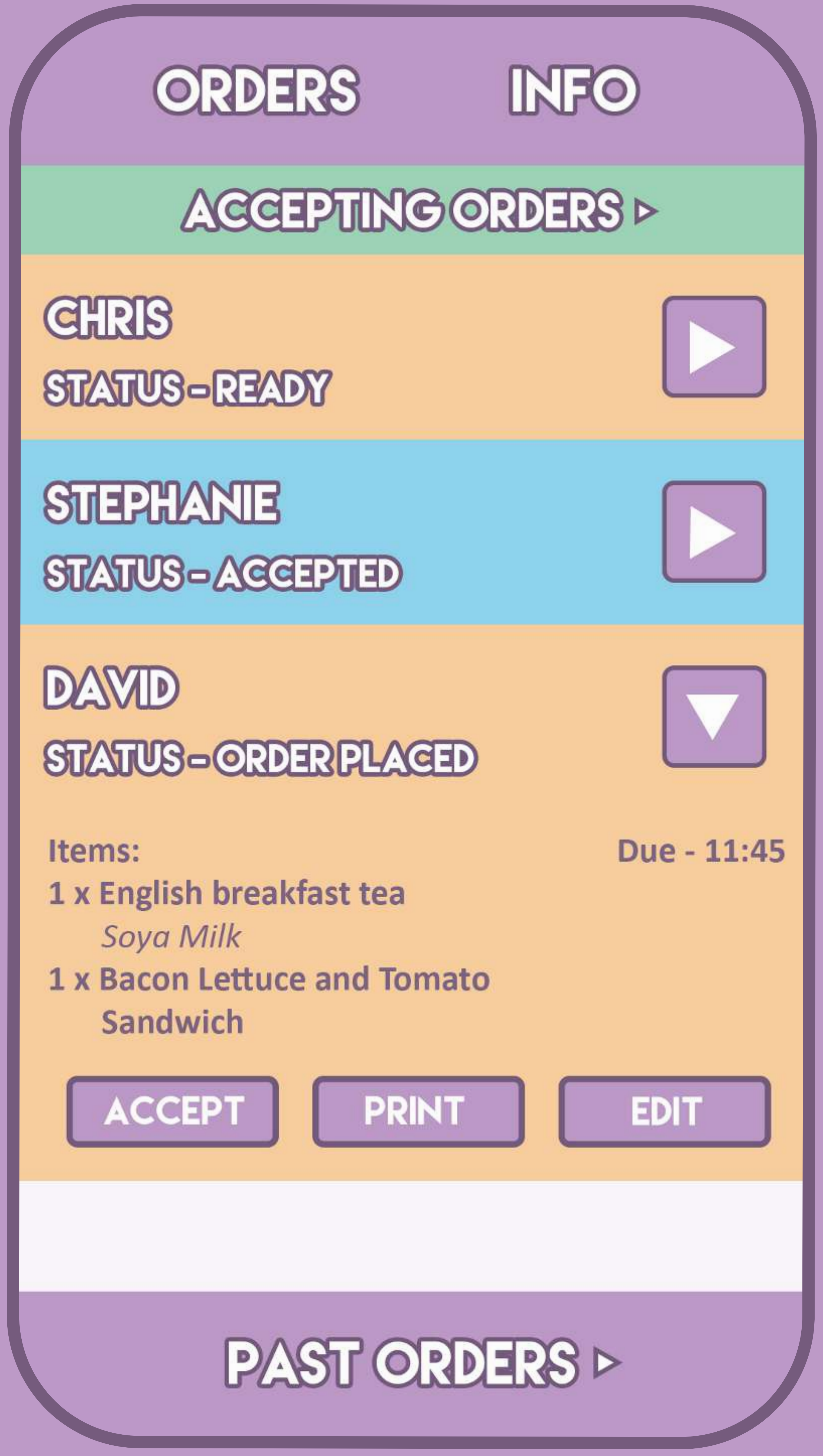
ACCEPTING ORDERS ▸

CHRIS
STATUS - READY ▶

STEPHANIE
STATUS - ACCEPTED ▶

DAVID
STATUS - ORDER PLACED ▶

PAST ORDERS ▸



ORDERS INFO

ACCEPTING ORDERS ▸

CHRIS
STATUS - READY ▶

STEPHANIE
STATUS - ACCEPTED ▶

DAVID
STATUS - ORDER PLACED ▼

Items: Due - 11:45
1 x English breakfast tea
Soya Milk
1 x Bacon Lettuce and Tomato
Sandwich

ACCEPT PRINT EDIT

PAST ORDERS ▸

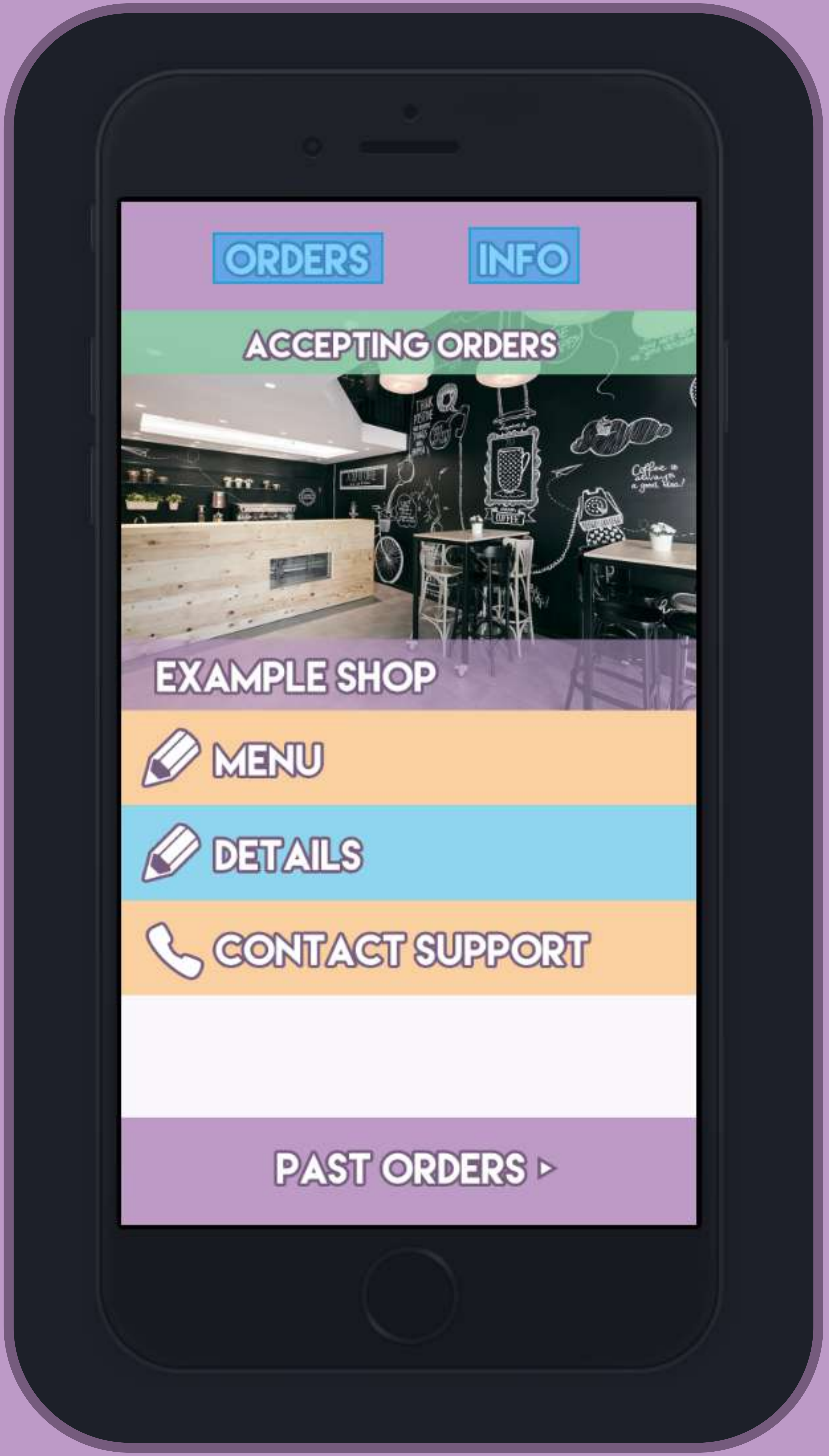
User Testing

In my second round of user testing I got some more detailed feedback as the user was able to interact with a more fully formed application.

Whilst I thought having interchanging colours on list items was visually appealing users found it confusing as it made it seem that the 2 button types were differentiated for a reason and that the 2 buttons would act in different ways.

Users also had queries as to how the application would handle having multiple orders placed on the orders page and it wasn't immediately obvious that this is where they should look to see the status of their order once it has been placed, I decided that due to this once a user has an active order a tab button should appear at the bottom which links them to view their current order and a confirmation screen telling the user how to view their now active order would help to tackle this.

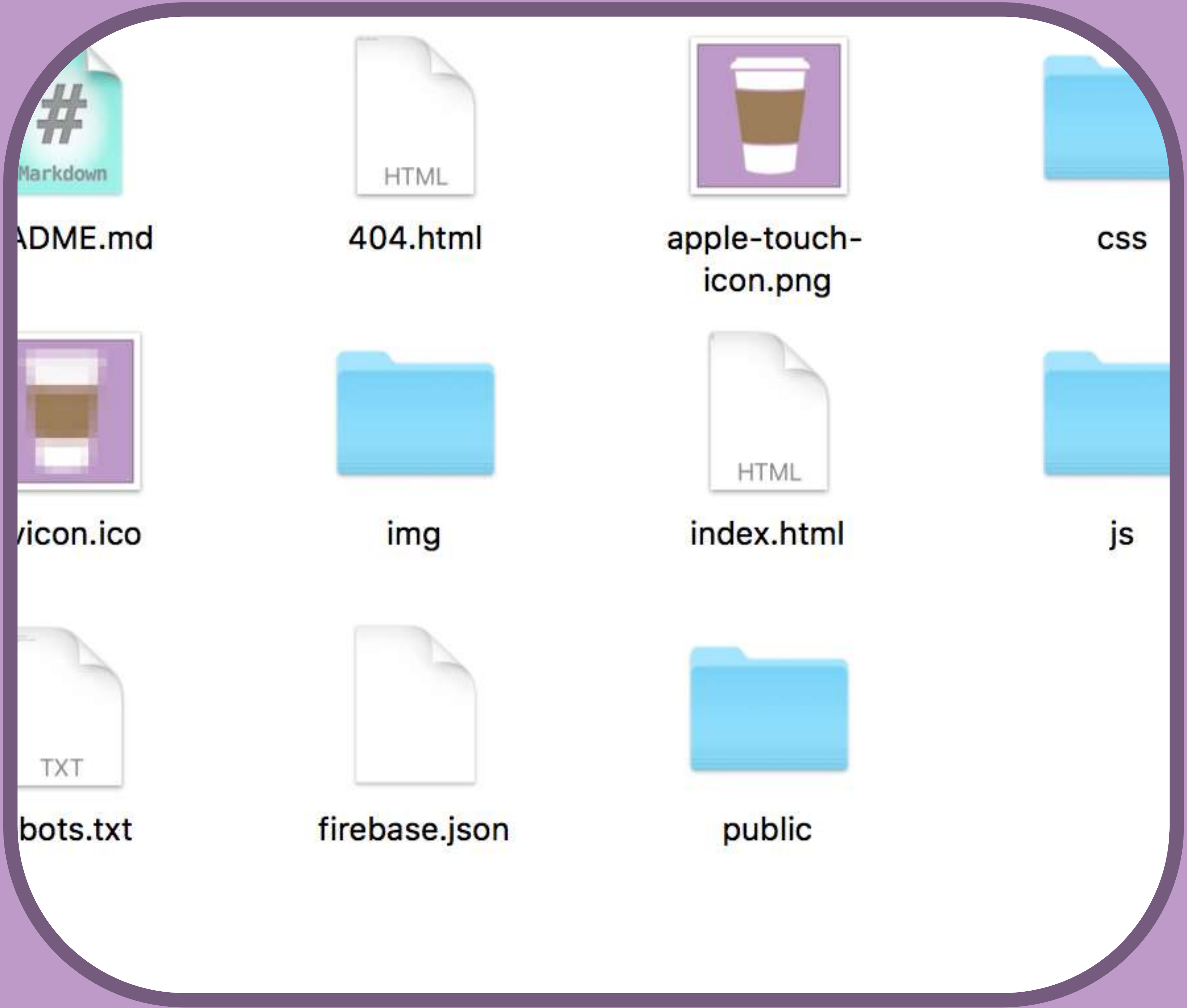
Users also had questions about elements like the loyalty system and were curious as to how this would work, it led me to realise that there are multiple possible ways in which a loyalty system could work and it will be important to try multiple ideas and see which fares best with consumers and retailers.

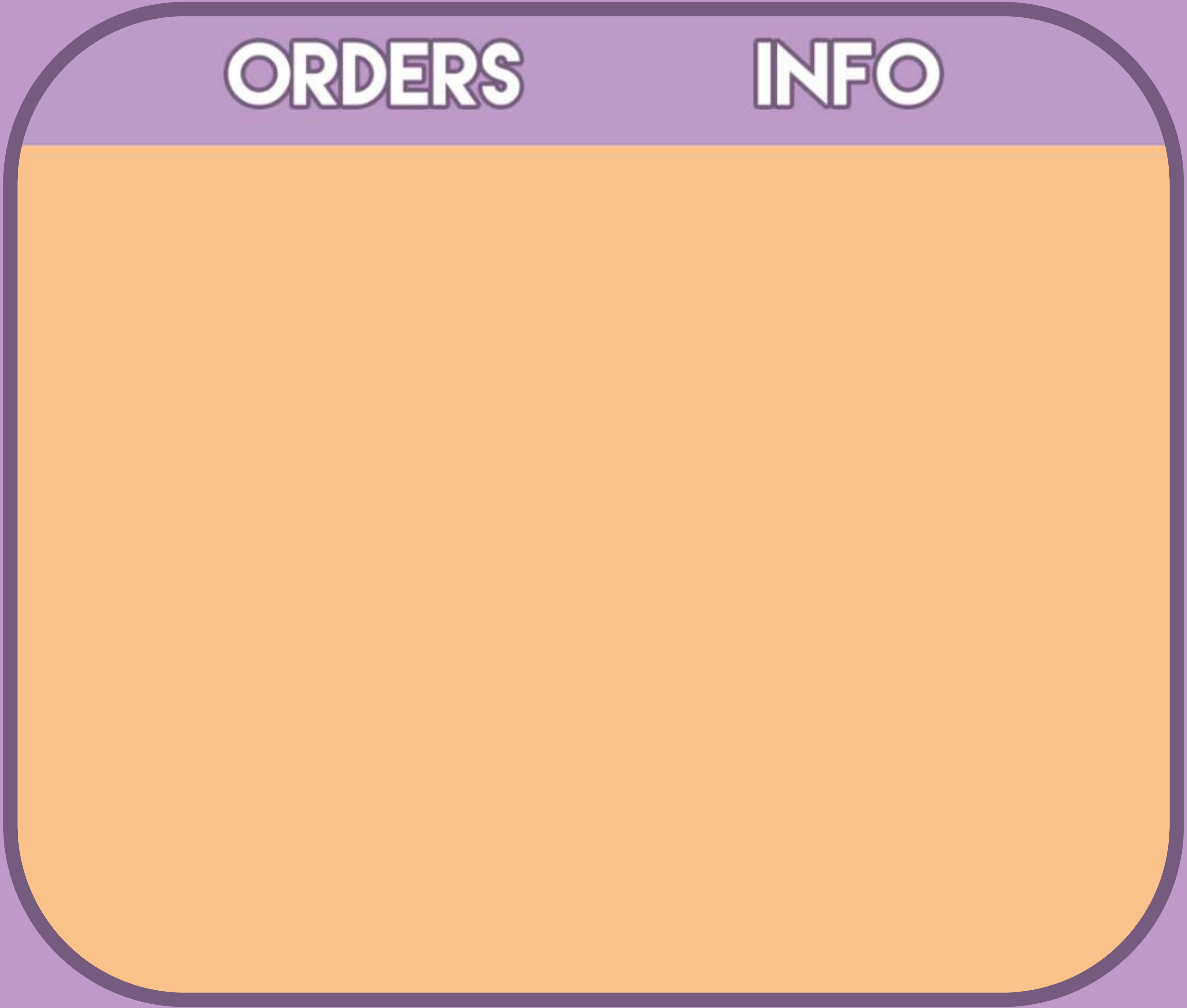


Proper Coffee

At this point I decided to change the name of the application, after thinking about it I decided to use the name ‘Proper Coffee’. Proper Coffee implies quality and concisely lets the user know it’s genre.

Starting Development





To start my development off I made a folder for my applications and within that folder created one for each of my apps, the consumer and the retailer side. Then I set up a version control manager, after some advice I decided to go with bitbucket as I could make my project private whilst I was doing my initial development and source tree for committing to this repo. I used HTML5 Boilerplate to get some basic file structure and edited this down to the files that I would need, replaced some of the default images like favicons and changed some of the naming to suit me. After I had created my own boilerplate I copied that over for the retailer application so I had 2 folders which started in the same way

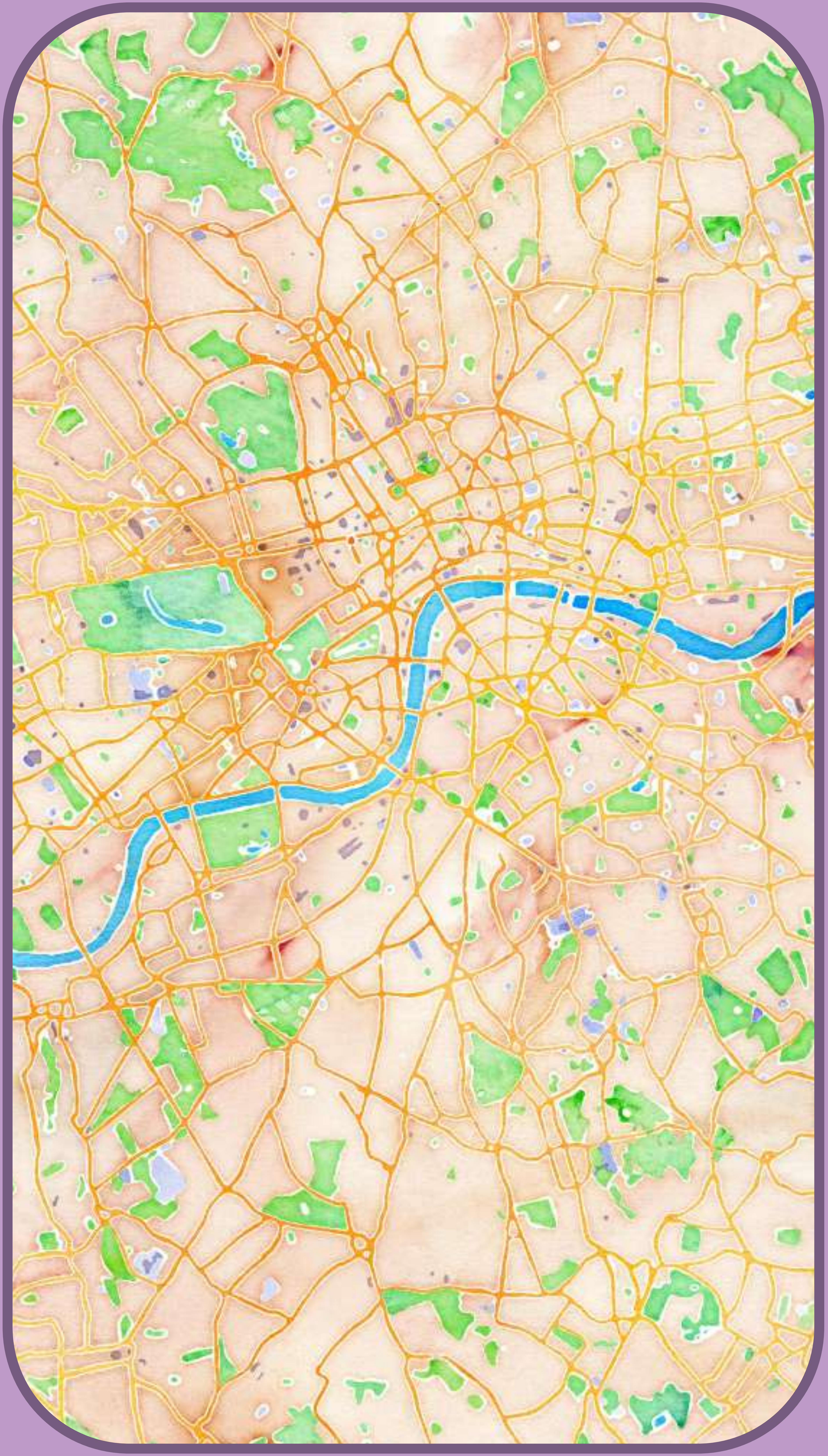
I then started to use HTML, CSS and Javascript to create a basic structure for the application. I decided to start with creating just the menu bar and use javascript to allow the user to click and switch between the different tabs and used different coloured backgrounds on the divs that appear to represent the different pages so that I could see that it is working.

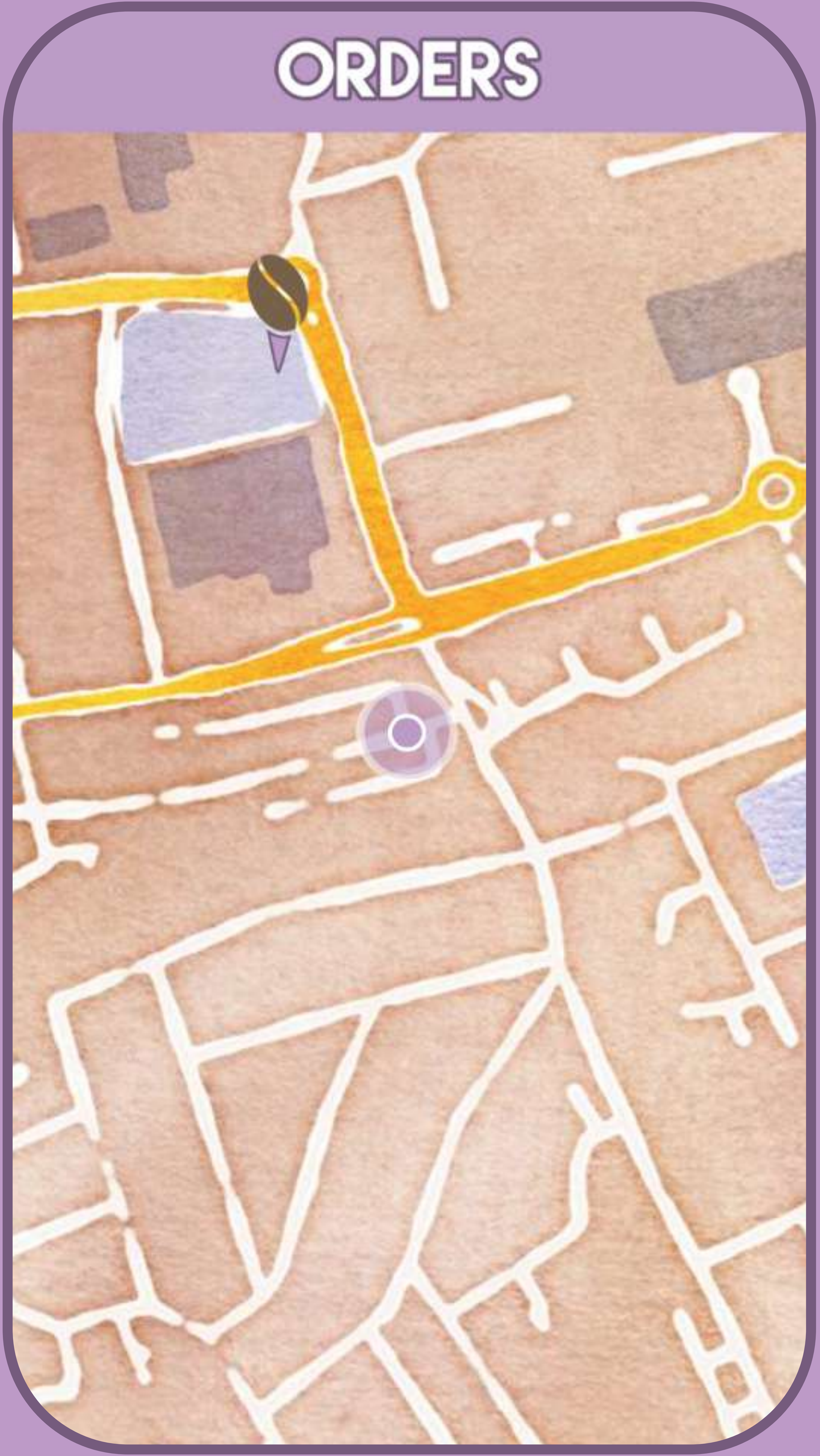
Starting Development

I used the same colours and font as I did in the wireframes initially but this is subject to change based on the results of further user testing. I also ensured that everything in the application is responsive so that it will fit to whatever sized screen is being used and items within the application appear in an appropriate position.

This is as far as I needed to take the retailer app at this point as I'm yet to set up any kind of system to pull information in. With the consumer app I also decided to implement the map tab to show the users current location.

When I created 'Not Another Starbucks' I used some open source mapping software called Leaflet, leaflet is a Javascript library which creates mobile friendly maps in the browser, the data for these maps is provided by Open Street Map and I used map tiles created by a customised map tile styling company called Stamen. Leaflet is a really well documented library which makes it easier for me to develop as I can refer to this documentation to find out the capabilities and achieve my goals so I decided to implement mapping into the consumer app in the same way.





Starting Development

I added the map into the consumer app in a div which appears lower than other elements in the application so that items like the menu bar can overlay the map. The users location is represented by a purple circle like in the wireframes and a translucent wider circle around this represents the radius of the accuracy of the location data it has been given by the device accessing the page.

When implementing the mapping I referred to the Leaflet documentation and to my old project ‘Not another Starbucks’ but after implementing the same way I did previously I noticed that the console was logging an error that it could not find the map tiles I had specified in a local folder, however it shouldn’t have been looking for these map tiles locally anyway as I don’t have the map tiling software stored locally within the folder so I changed the code so that this no longer happens. I then added a marker on to the map, a marker is one of the items that leaflet allows you to add to a map to represent a location, the marker I added is at a location close to where I was developing from so that I could interact with it for testing purposes however it will need to be removed in future as the markers on the map will instead be pulled from a database, I styled the marker to look the same as it does in my wireframes.

Starting Development

The next step I decided to take with the consumer application was to implement a changed view when the marker is pressed.

In the wireframes the marker becomes centered and the map zooms in and a pop up info box overlays the map to give additional information about the marker that has been clicked on. Using the Leaflet library I created a pan and a zoom action when the users clicks on the marker to centre it in the view, Leaflet is particularly good at doing this as it can automatically generate an animated transition between your current view and the one it is changing to, this makes the change in view less jarring for the user.

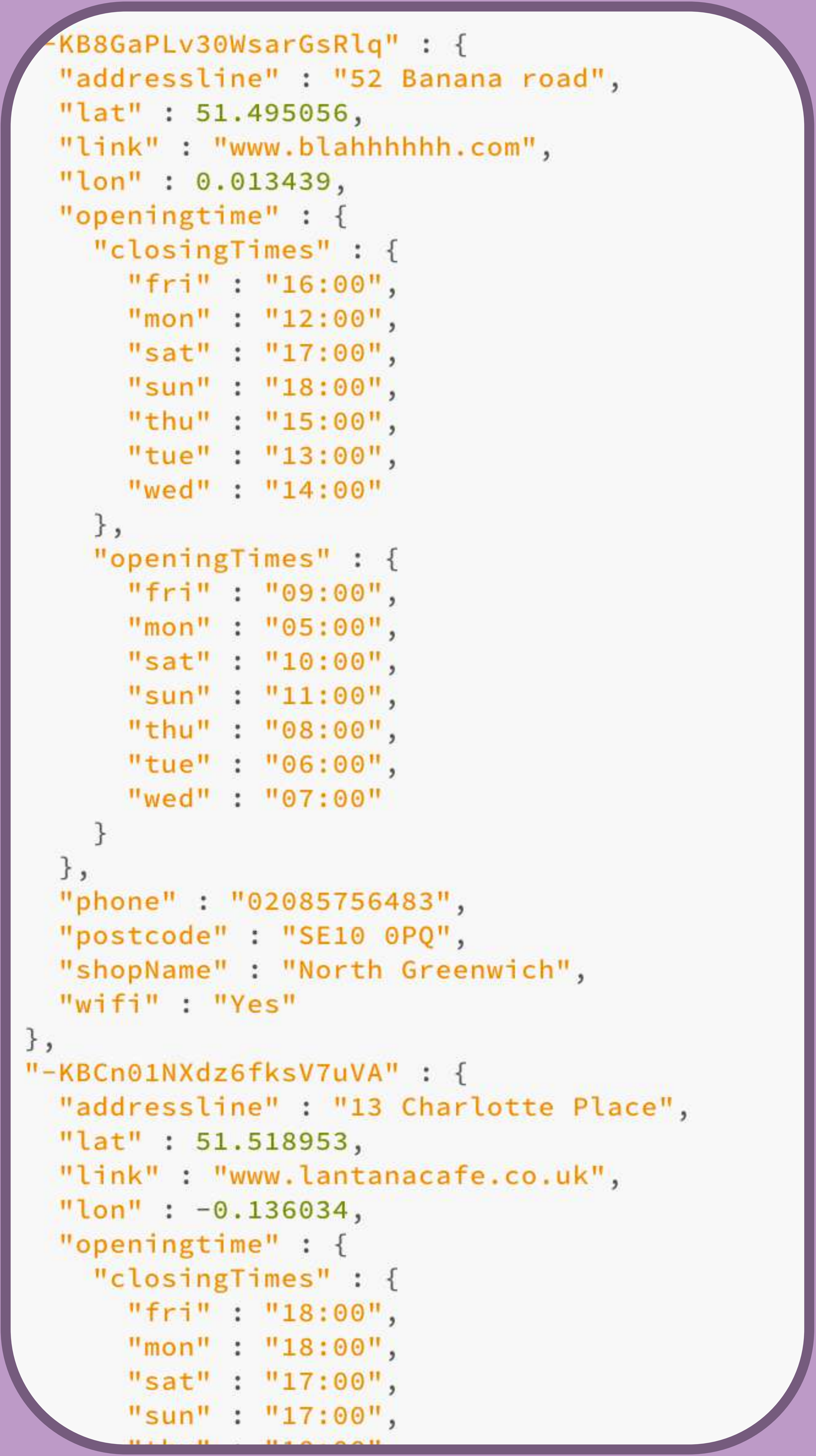
I then worked on making the pop up box appear where the additional information would be shown, I did this by creating a div that appears overlaying the map and using CSS to style it to look like it did in the wireframes, I had some issues doing this as when you add a border to a div it applies around the outside of the div, this means that with a div that currently stretches across the width of the browser the border makes the div bigger than the browsers view however I researched online and found a solution using the "box-sizing" element in CSS.

Once I'd created the div I hid it so that it only appears when the user clicks on the marker and then created an invisible div over the rest of the screen which the box doesn't appear on so that when the users clicks outside of the popup the popup disappears.

The zooming and panning I'm doing using leaflet centers the marker in the middle of the page rather than the space actually left in view once the tab has appeared, this can be fixed when I load in actual data from a database by slightly modifying the point which the view is being centered to by creating a new latitude variable which centers the view slightly lower than the marker, as I know the zoom level as I change it when the marker is pressed I should be able to do just one function which changes the latitude it centres to which will work across all markers.

Once the tab is closed I have set it so that the zoom goes back to where the user had it previously. This is my base structure for the application at this point, next I will focus on the creation of backend elements such as database creation so that the application can be populated with data.

Database



```
-KB8GaPLv30WsarGsRlq" : {
  "addressline" : "52 Banana road",
  "lat" : 51.495056,
  "link" : "www.blahhhhhh.com",
  "lon" : 0.013439,
  "openingtime" : {
    "closingTimes" : {
      "fri" : "16:00",
      "mon" : "12:00",
      "sat" : "17:00",
      "sun" : "18:00",
      "thu" : "15:00",
      "tue" : "13:00",
      "wed" : "14:00"
    },
    "openingTimes" : {
      "fri" : "09:00",
      "mon" : "05:00",
      "sat" : "10:00",
      "sun" : "11:00",
      "thu" : "08:00",
      "tue" : "06:00",
      "wed" : "07:00"
    }
  },
  "phone" : "02085756483",
  "postcode" : "SE10 0PQ",
  "shopName" : "North Greenwich",
  "wifi" : "Yes"
},
"-KBCn01NXdz6fksV7uVA" : {
  "addressline" : "13 Charlotte Place",
  "lat" : 51.518953,
  "link" : "www.lantanacafe.co.uk",
  "lon" : -0.136034,
  "openingtime" : {
    "closingTimes" : {
      "fri" : "18:00",
      "mon" : "18:00",
      "sat" : "17:00",
      "sun" : "17:00",
```

My applications both need to pull information from a database and write information to it as well, so that I don't have to reinvent the wheel and create my own database software I have investigated different services that are available to assist me with doing this. Parse was a real legitimate option, it has all kinds of features such as push notifications to iOS and android etc which made it a really good option for me to look into however just before I was going to begin looking at how I would implement parse into my application the owners of the service have decided to discontinue it, whilst they will be supporting existing applications that use Parse for a short while it's not a long term solution.

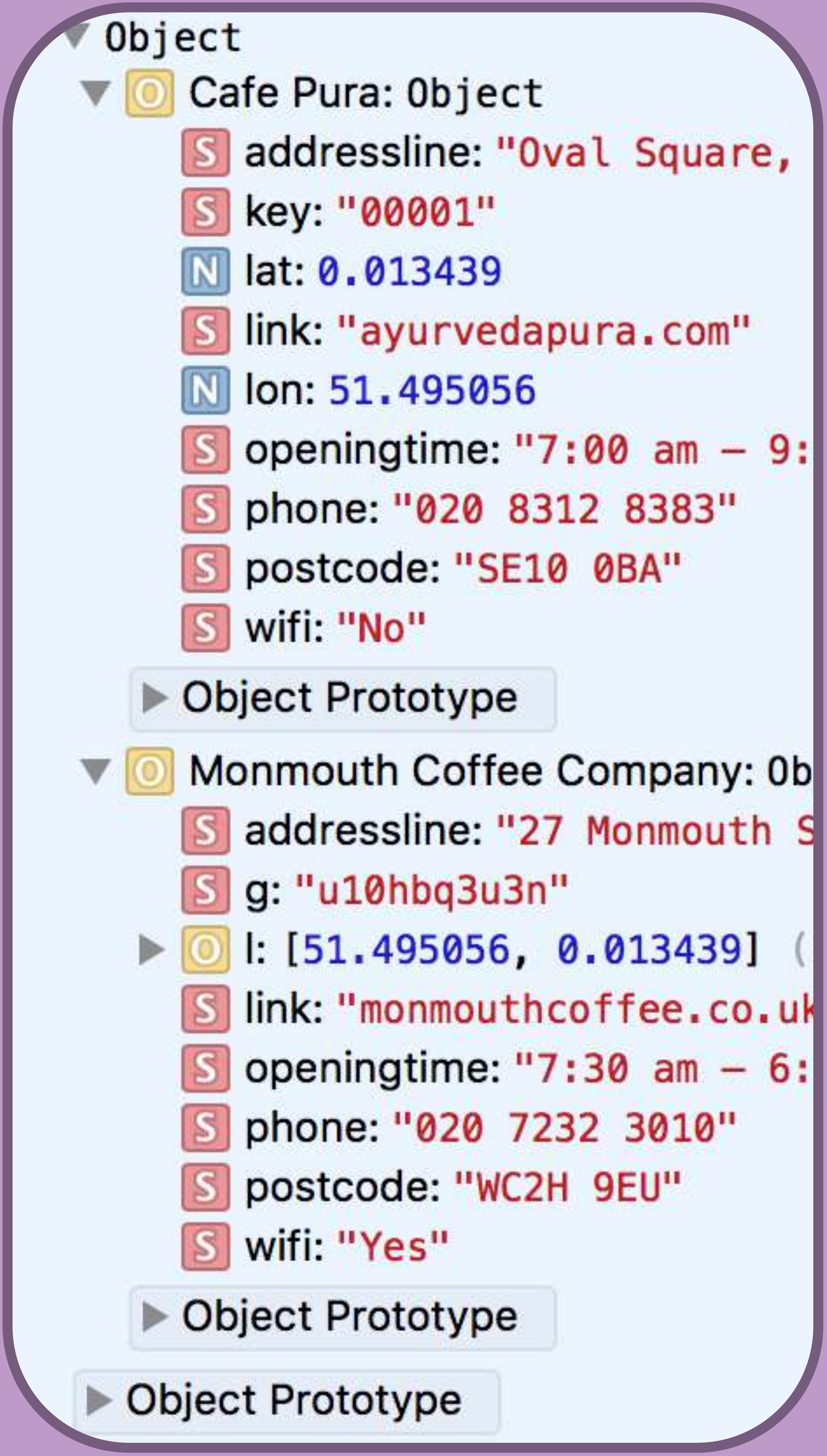
It seems the most popular alternative to Parse at the moment is Firebase which I was recommended to look into. Firebase is owned by Google and allows you to create a JSON based database with security and allows remote reading and writing of this database as well as more complex features like user authentication etc. I followed the basic tutorial they have on their website and decided to try to work on creating a Firebase database and see whether I could use it for my application. To use firebase I had to add some extra JS libraries to my application and I also created a basic JSON database file from 2 of the entries I had in my 'Not Another Starbucks' database (Which was powered by Google sheets, works for my basic application back then but isn't ideal for the more fully function Proper Coffee) and uploaded this to my firebase.

Database

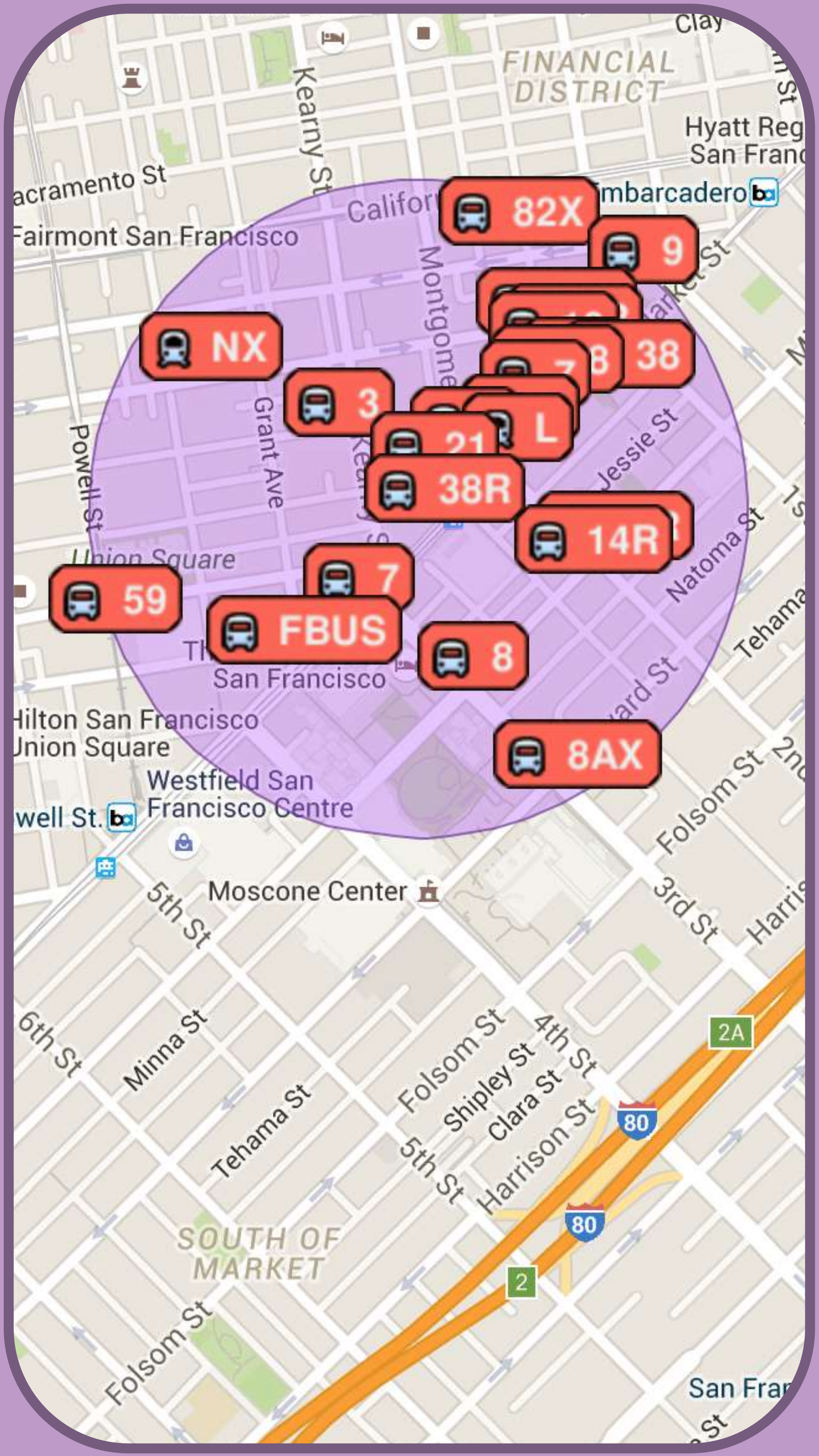
I'm going to need to use firebase for multiple things, I'm going to need it to store the locations of my coffee shops and additional information about them so that they can viewed on a map, I'm going to need it to store an order once a user places it, have that order pushed to the retailer application so that they can make it, have the retailer app be able to modify the orders status so that the user knows it's ready and use firebase for saving users log in details and additional information.

As I require so much from Firebase and it's a new tool to me I'm beginning with what I view as one of the simpler parts and that is the coffee shop database. After I had implemented a small database into my Firebase I began looking at how I can view that data from my application, I had already added the Firebase library to my code so I could start using Firebase commands in my javascript, I accessed the database by referencing it in my Javascript and then ran a test by getting it to console log the information that I had entered into the database and get it to return information to me in the console.

I also experimented with using other Firebase commands such as changing the data etc but that was just to test how it worked.



```
▼ Object
  ▼ O Cafe Pura: Object
      S addressline: "Oval Square,
      S key: "00001"
      N lat: 0.013439
      S link: "ayurvedapura.com"
      N lon: 51.495056
      S openingtime: "7:00 am – 9:
      S phone: "020 8312 8383"
      S postcode: "SE10 0BA"
      S wifi: "No"
    ▶ Object Prototype
  ▼ O Monmouth Coffee Company: Ob
      S addressline: "27 Monmouth S
      S g: "u10hbq3u3n"
    ▶ O l: [51.495056, 0.013439] (
      S link: "monmouthcoffee.co.uk
      S openingtime: "7:30 am – 6:
      S phone: "020 7232 3010"
      S postcode: "WC2H 9EU"
      S wifi: "Yes"
    ▶ Object Prototype
  ▶ Object Prototype
```



Geofire

Then I began to look at an extension for Firebase called GeoFire, GeoFire is an additional library that contains resources specifically for applications that use geographic data like I do with the locations of my shops. I viewed some examples and began to see how Geofire works, Geofire is really cool because it allows you to get data from it's Geofire database within specific areas of the map, for example I can load all of the shops that are within a 1 mile radius of the users location where in the past with 'Not another Starbucks' because I was using just a small dataset I had it so it just loaded the entire database.

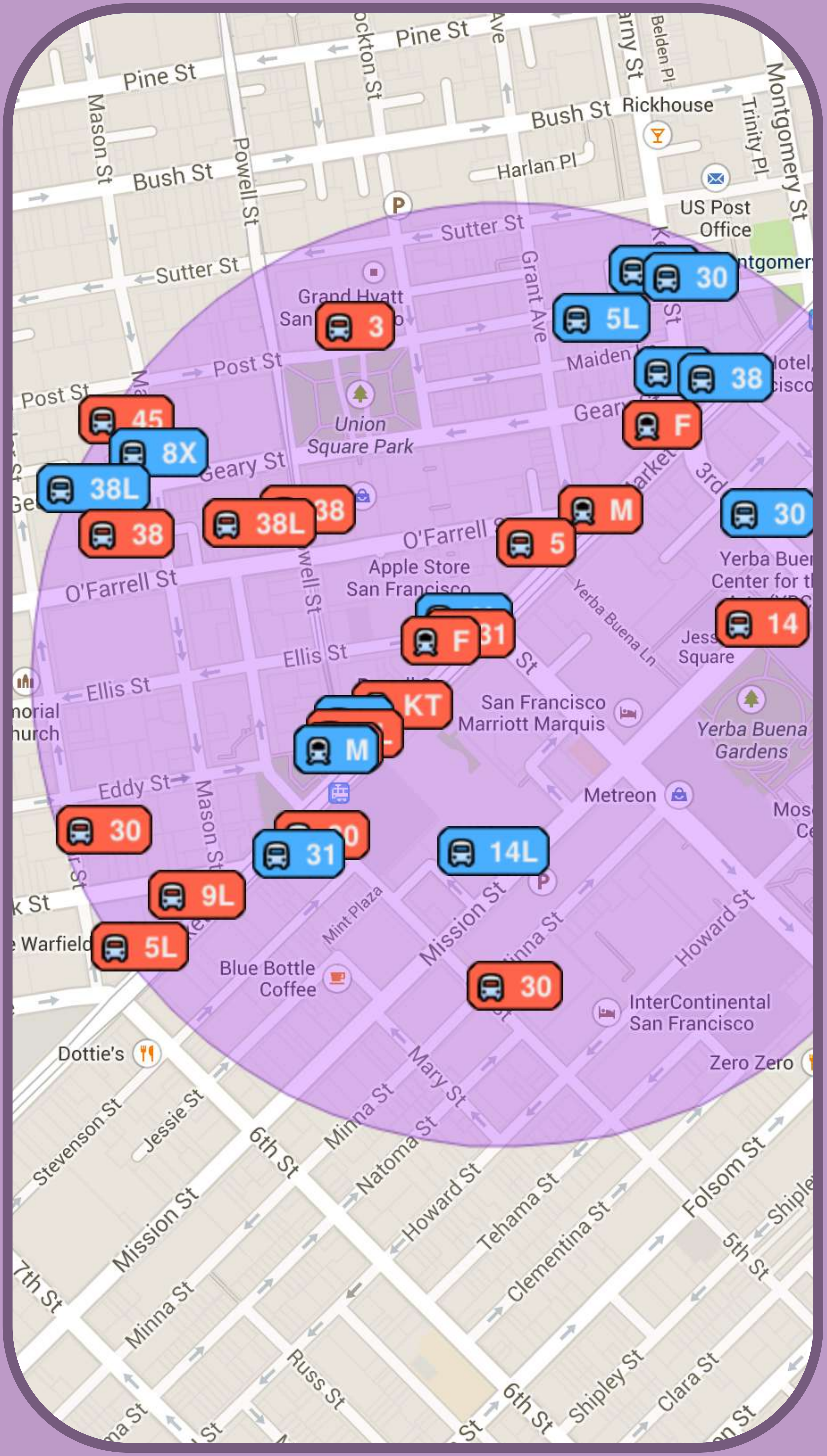
Loading an entire database every time a user accesses the app is bad idea as a long term strategy as it will affect load time and also use more of the servers bandwidth so implementing a GeoFire query where only the shops within a certain radius of the user or where the user is viewing on a map seems like a sensible option.

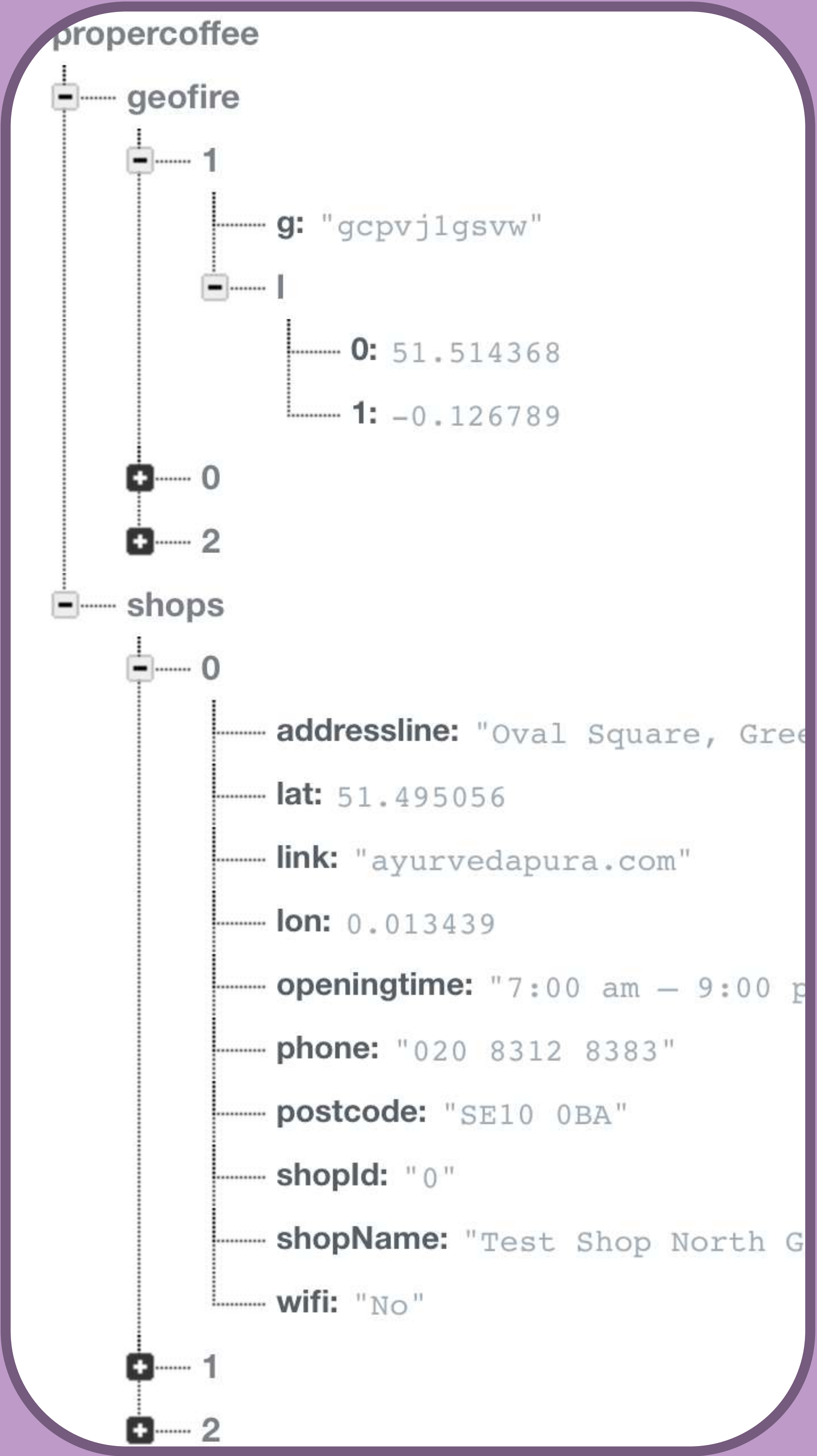
Geofire

One of the other interesting things with Geofire is that the data updates live so that you can see if something's location changes, in the example all of the buses are actually moving live on the app. Geofire seems to have to create it's own separate database within Firebase as it takes the longitude and latitude that I provide it with and gives it additional data in this database which seems to be the key to allowing you to find only locations within a specific area, if I try to get Geofire to make changes to my existing database it disregards information which is important for me like the shops phone number as it is only concerned with the geographic information.

I experimented with the functions available with Geofire just to see how it worked by adding information to the GeoFire datase, reading that information back and creating queries which take the users location and find out which objects in the databse are within a given radius.

What I need Geofire to do though is to read my existing Firebase database and take the information from it and then add it to it's own Geofire database within Firebase and then link the 2 entries in the 2 database together to display the appropriate shops. The example above with the buses seems to do this somehow as it takes information from a database about buses and then creates it's own Geofire database to show only the nearby buses. I will investigate how this is done in order to get it set up on my application.





```
propercoffee
  geofire
    1
      g: "gcpvj1gsvw"
      l
        0: 51.514368
        1: -0.126789
    0
    2
  shops
    0
      addressline: "Oval Square, Gree
      lat: 51.495056
      link: "ayurvedapura.com"
      lon: 0.013439
      openingtime: "7:00 am – 9:00 p
      phone: "020 8312 8383"
      postcode: "SE10 0BA"
      shopId: "0"
      shopName: "Test Shop North G
      wifi: "No"
    1
    2
```

Geofire

I've created code which gets GeoFire to scan my original Firebase database and create a new GeoFire database, it takes the lon and lat that I entered but it also generates unique codes for example "gcpvj1gsvw", I think this code represents a specific location on the world map in one string rather than 2 thus allowing Geofire to tell what's close to a given point without having to plot a point on a map to know.

I have implemented my code so that once the browser has told my Javascript the user's location it creates a Query which finds out which coffee shops are within 30km of the user and then adds them to the map and specifically only loads the data for these entries, this will massively help loading time if the database was to become really large. I still have to implement things like removing ones when the user changes view and loading them as currently it uses the users location rather than their view of the map so that is part of the next steps however that part should be easy now I have it all working.

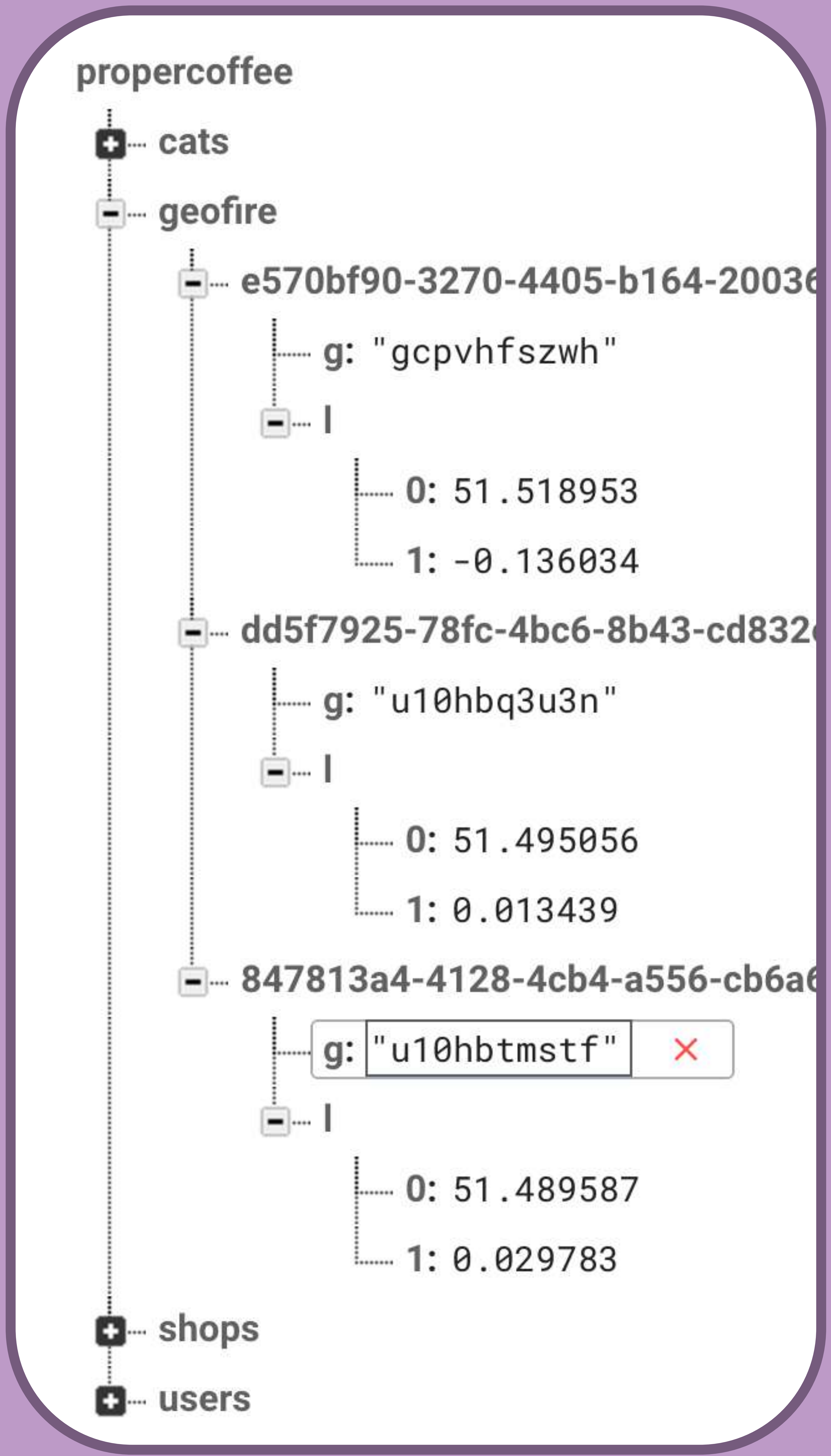
Implementing this for a user panning around and looking at a different area will have to be implemented based on the users current central map view so I will implement this next.

Geofire

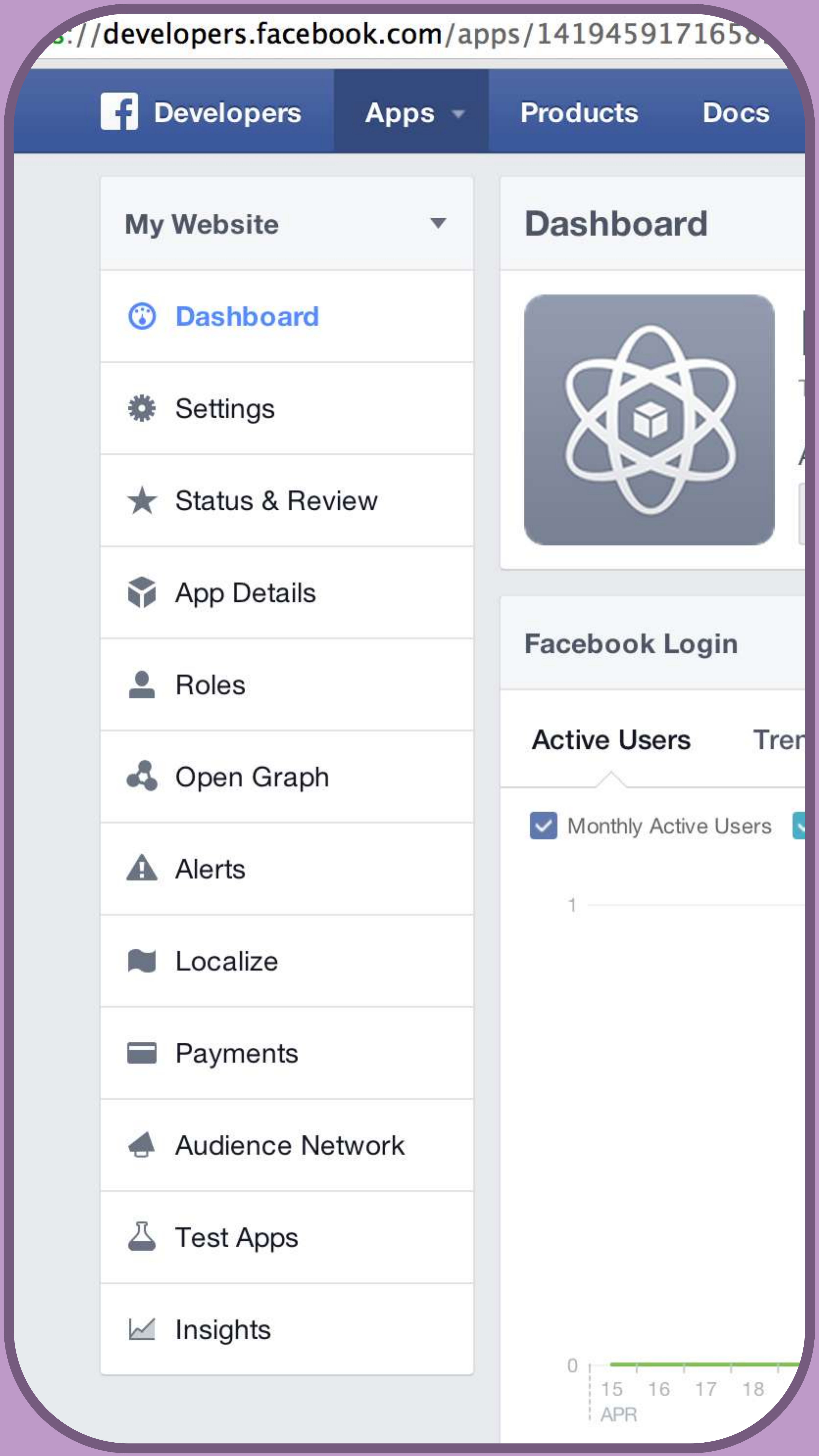
To make the required changes to the way that Geofire works I used leaflet to detect the users movement when they pan around of the map, whenever this happens it causes leaflet to update and find the current centre position of the map as the user sees it, it then passes the longitude and latitude it finds back into the GeoQuery and loads any additional shops which would now be showing, currently the radius it loads from the user is still at 30km as my database hasn't become sufficiently bulky that i'd need to load a smaller amount.

I then faced a big realisation with the way I had been building my application so far. I've been experimenting with how you set new entries to the database with Firebase and how I can then get Geofire to automate an entry from firebase into the Geofire database however having this included in the user part of the application ultimately doesn't make sense; having the users browser load up and check my whole database makes it pointless loading up only the data from nearby as the users browser is still downloading the whole database anyway.

My solution for tackling this is to create a whole new separate application, Proper Coffee Admin.



```
propercoffee
 ⊞ cats
 ⊟ geofire
    ⊟ e570bf90-3270-4405-b164-20036
         g: "gcpvhfszwh"
       ⊟ l
            0: 51.518953
            1: -0.136034
    ⊟ dd5f7925-78fc-4bc6-8b43-cd832
         g: "u10hbq3u3n"
       ⊟ l
            0: 51.495056
            1: 0.013439
    ⊟ 847813a4-4128-4cb4-a556-cb6a6
         g: "u10hbtmstf"  ✕
       ⊟ l
            0: 51.489587
            1: 0.029783
 ⊞ shops
 ⊞ users
```



Proper Coffee Admin

Proper Coffee Admin is an application which will have the ability to be able to add entries to my coffee shop list and then automate it so they are added to geoFire. To start this off first I had to restrict the way I had been doing it so far by adding some rules to my firebase, I allow all users to be able to read from my firebase shop list and my Geofire but not to be able to write this can be done on the Firebase dashboard and requires just an addition to a JSON list. Then I had a look at the different means I can have of logging in to my application using firebase, as well as adding my own email and password database Firebase also gives you the option to add Facebook, Twitter and other public platforms as your log in provider.

I decided that I would secure my new app proper coffee admin with my Facebook account so that I would have to log in with my Facebook account to make changes, this made more sense to me as I will only need very few admin accounts to begin with at least and it secures my data against something that's unrelated to my firebase database. To do this first you have to open a Facebook developer account and create an application, I created one called proper coffee admin and followed some steps from firebase to get the initial set up done as well as adding data from Facebook back into my firebase and adding the domain on which the Proper Coffee Admin account would be hosted.

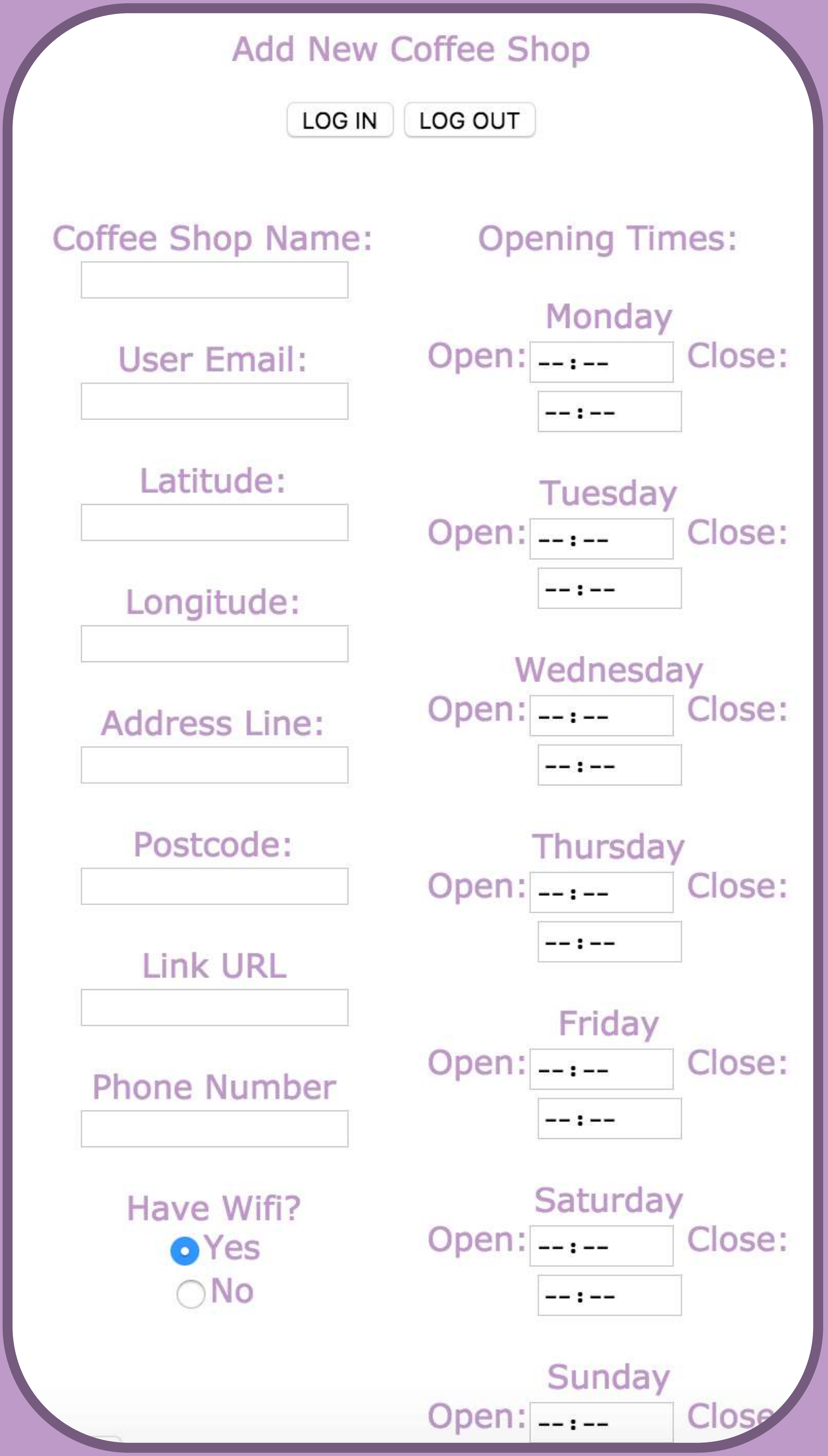
Proper Coffee Admin

I hit a lot of snagging points when trying to implement logging in with Facebook as it only works when you test it on a domain which meant uploading any changes before I could test them as well as a frustrating period where I couldn't work out why my local copy wouldn't work.

I also created a form to add the data for each coffee shop into my database, this form enters the data a little different from my previous database entries, I scrapped shop id as the unique key as firebase saves them uniquely when you submit anyway and I added opening and closing times for every day of the week but collapsed it inside it's own child of the object

Once I had the Facebook login working I also added log in and log out buttons and set a session length. If the user isn't logged in they will not see the form and the page will appear blank.

When the user hits log in a popup redirect to Facebook to log in with my application appears in a window



Add New Coffee Shop

Coffee Shop Name:

User Email:

Latitude:

Longitude:

Address Line:

Postcode:

Link URL

Phone Number

Have Wifi?
● Yes
○ No

Opening Times:

Monday
Open: --:-- Close: --:--

Tuesday
Open: --:-- Close: --:--

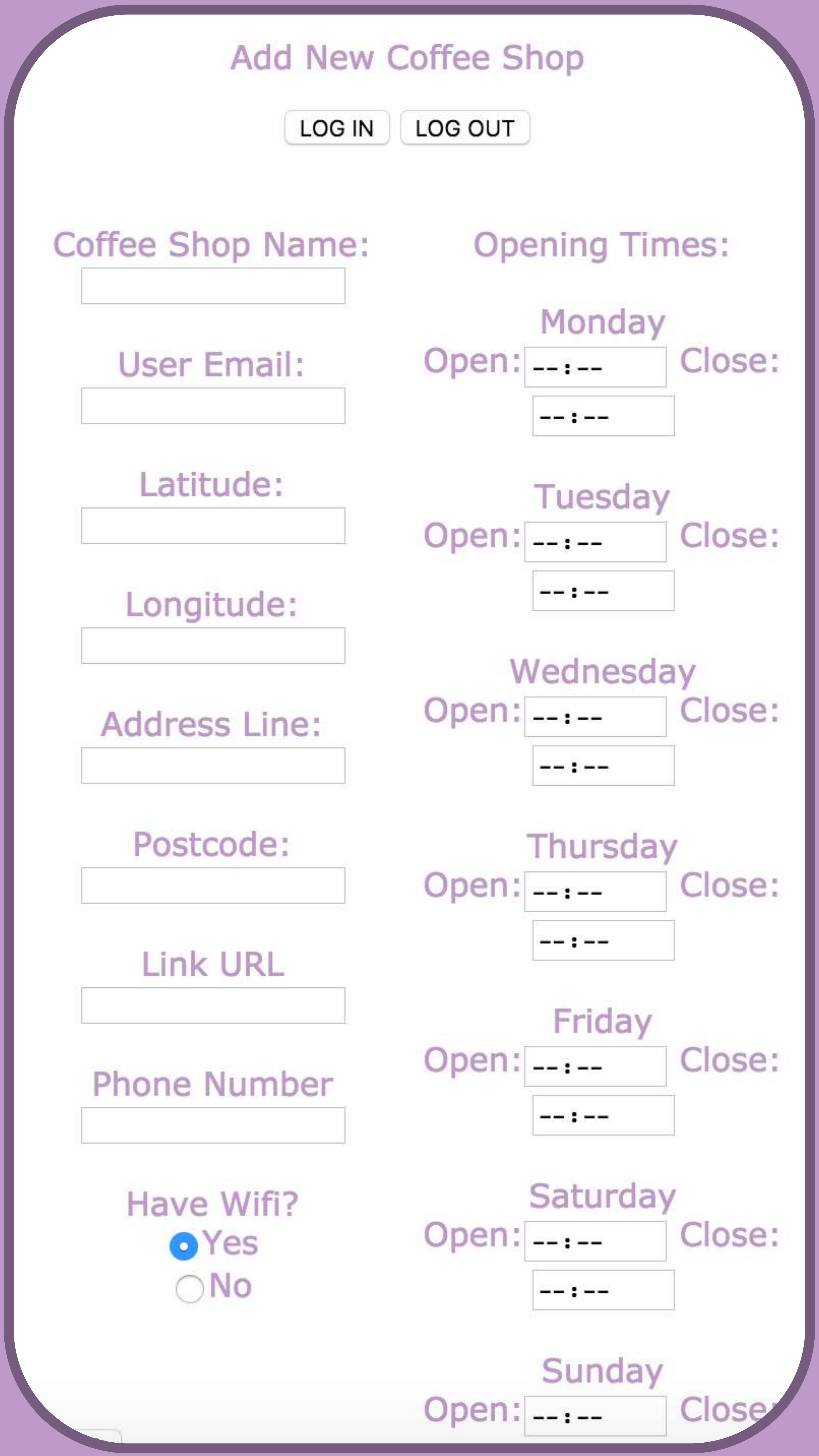
Wednesday
Open: --:-- Close: --:--

Thursday
Open: --:-- Close: --:--

Friday
Open: --:-- Close: --:--

Saturday
Open: --:-- Close: --:--

Sunday
Open: --:-- Close



Add New Coffee Shop

LOG IN LOG OUT

Coffee Shop Name:

User Email:

Latitude:

Longitude:

Address Line:

Postcode:

Link URL

Phone Number

Have Wifi?
● Yes
○ No

Opening Times:

Monday
Open: --:-- Close: --:--

Tuesday
Open: --:-- Close: --:--

Wednesday
Open: --:-- Close: --:--

Thursday
Open: --:-- Close: --:--

Friday
Open: --:-- Close: --:--

Saturday
Open: --:-- Close: --:--

Sunday
Open: --:-- Close

Proper Coffee Admin

As my account is already trusting this Facebook application Facebook sends an authorisation with a ton of different data back at my firebase, the only one I'm interested in is the user id, the user id is unique to me and when Facebook send it inside an auth my firebase rules can then read that the userid is mine and allow me write access with my browser session, that session lasts for one hour unless I log out before. At the moment because of the way the pop up works it doesn't work on mobile so I may switch it to a browser redirect as this seems to be the way other web apps do it.

Because I changed the way data is stored inside the database when I created my admin app there were things I had to change with the consumer app. The consumer app simply printed a string I had entered for the opening times before but that didn't allow for different opening times of different days. To implement this I used jQuery to detect what day of the week the browser is telling it is (as I want to know the users local day not the day of a set area) which returns a number, 0 for sunday, 1 for monday etc. I then created an if function when the user presses on one of the markers shown on the map, it detects what day of the week it is and then fetches the opening time and the closing time and displays this formatted in the popup info box. Support for days when the shop isn't open will be implemented later.

Contacting Retailers

So after spending time my CSS and getting the repos in a general good order I decided to buy the domain. I bought the domain proper.coffee however have not yet bought any hosting to go with it so currently it's going to redirect to a placeholder site that I have on my ravewebmedia.co.uk webspace however i did try to use digital ocean as the webhost and spent a lot of time trying to figure it out but it seemed beyond what I needed at this point for the project.

I then created a template email to send to some coffee retailers in London to start populating my database the template is as follows:

"Hi,I'm Harry and I'm developing a London based Coffee Shop finding service called 'Proper Coffee', the results displayed by Proper Coffee will not include Starbucks, Costa, Cafe Nero or any major coffee shop chain and instead focuses on smaller independent businesses. This is for my final major project at university but hopefully would go live at a later point.

The eventual goal would be to have an application where a user can discover a coffee shop like yours, place an order and pay online and then you as the retailer get a notification on your version of the app to start making the order. The users can also accumulate loyalty points that can be used at any participating shop.

I would love to include your shop on my prototype application, if you could reply confirming you'd be ok with this that would be great. I will need the following details about your shop as you want them to be shown:

Shop Name, 1st line of address, Postcode, Website, Phone number, Do you have wifi, Opening times.

I look forward to your reply,
Harry Foster."

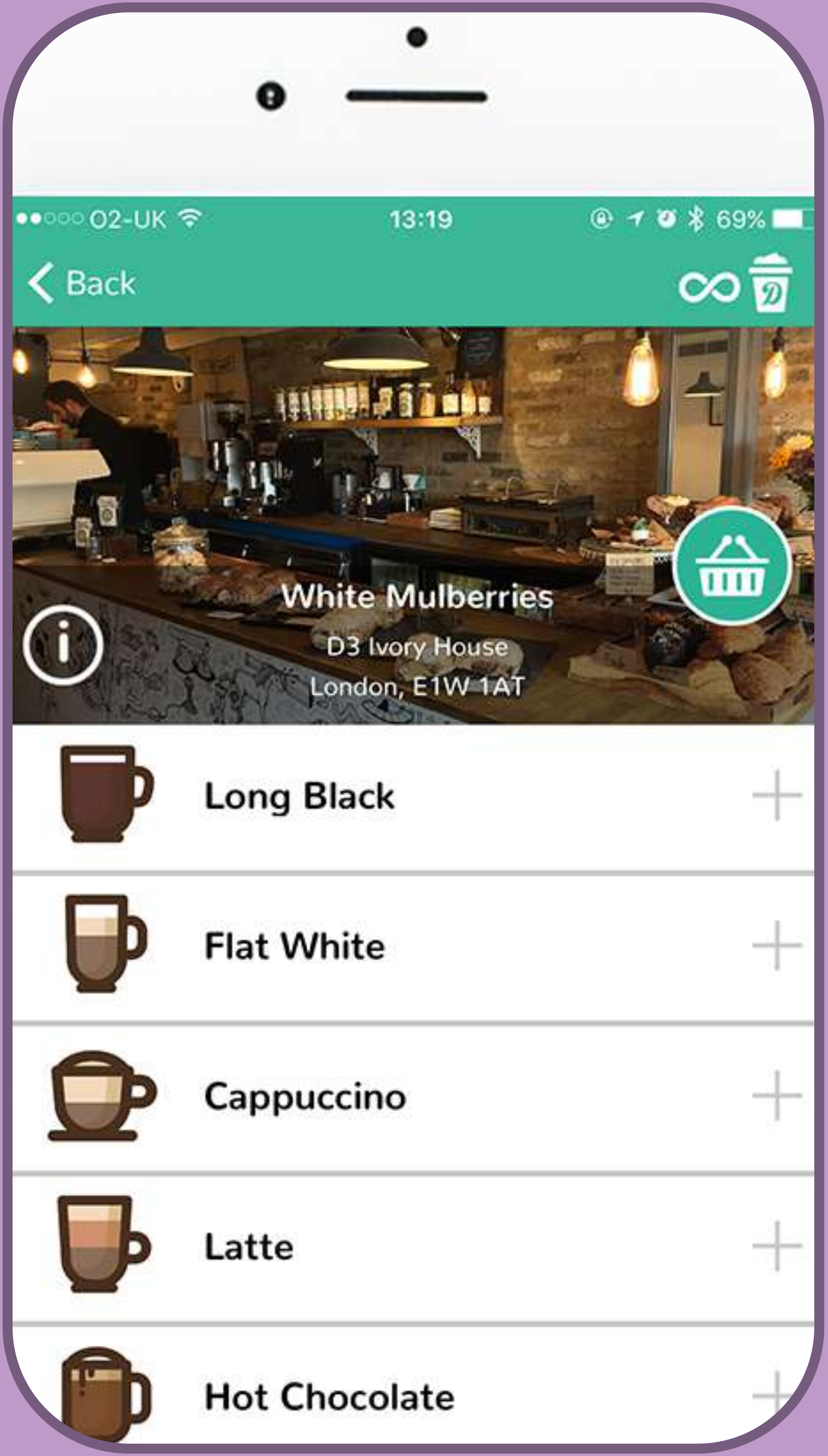
I then found a list published in the London Evening Standard last year detailing their list of the best independent coffee shops in London, I began going through the list, finding their websites and adding them to a BCC and managed to compile over 50 email addresses this will give me a start on beginning to populate my database.

Retailer Responses

I had a variety of responses from the retailers I contacted by email,it was the first opportunity I got of receiving feedback on the idea from a retailer perspective.

Results were mixed but largely positive most retailers that responded were willing to be featured on the application, they had concerns that there are currently other applications trying to do a similar thing. I was already aware of competitors and feel that Proper Coffee has a USP compared to these applications in terms of the ordering system and loyalty system, other applications either seem to allow you to buy a pre-paid batch of coffees at once which can be redeemed at participating retailers to save money 'Drip' for example, while other simply display retailers on a map like 'Not another Starbucks' did. Buying a batch of pre-paid coffees is an interesting concept however it relies heavily on the user making a substantial jump to buying a batch and has no automated ordering system but operates by having the user show the retailer a code within the application.

The data collected from those that responded was useful as it highlighted flaws with the way I currently enter retailer information such as variances in opening times and I made appropriate changes to the admin application.



Angular

I’ve now bought the domain http://proper.coffee and bought hosting from firebase and deployed my prototype so it’s online, it doesn’t need to be online and isn’t intended for use right now however it being online helps with testing as some features like Facebook authentication and asking for the users current location aren’t available offline and blocked by certain browsers so i’d spend time trying to troubleshoot a problem which wasn’t really there.

At this point i’ve decided to add Angular.js to my project, I’ve not used angular in the past so i’m taking some time to try to work out how and what it does, it is well known for being particularly good at working with HTML to make it more dynamic which is useful when developing an application as it keeps it from becoming bulky by doing things in the back end. Angular also has a specific firebase plug in which is helpful and is owned by Google.

It may take some time to implement on my application but it will be worth it as it will take longer later on and should save me time. I went through the guide on how to use firebase and angular together on the firebase website which was useful however I think I need to research how to set up an app in the angular way first before continuing development.

Angular

In order to further investigate the benefits of using Angular within my application I followed a tutorial on angularjs.org.

The angularjs.org website is the official website of Angular by Google and has an introduction to angular tutorial in which you develop a web app to view some android devices. The tutorial highlights several benefits of using Angular as a framework for a web application, angular uses controllers to dynamically display the content of a web application. Angular does this by making the HTML dynamic instead of static, generally when you create a HTML page although you can make it interactive with JavaScript essentially you ask the HTML page to load up a script you have written in JavaScript however the HTML will simply load this entire script and apply listeners to elements on the page, this makes sense to do if an application only consists of 1 page view however I want mine to consist of multiple, that means that the script I load in my HTML would have to contain all of the code that is needed to control all of the different page views, this would result in a large Javascript file which will impact on loading times and how bulky the page is. Angular does things differently to a static page, it allows you to load specific javascript files when you're doing different things with the page, this means that content you need to be loaded only loads when it's needed, it also reduces the amount of code that needs to be written by adding in functions like repeats and splits the javascript up into multiple files, which makes it easier for me as a developer to see the structure of my application as it expands, instead of having an ever expanding javascript file it allows me to break it up into multiple files for each view.

Implementing it won't be easy however as I didn't start creating my application in this way, I need to massively restructure my application to work in a way where I have set views that I change using angular instead of showing and hiding different divs etc. I need to work out whether with the time I have remaining it is worth spending the time doing it now and risking not having enough time to add all of the functionality I'd like to by changing the structure and using angular which I'm not familiar with to control the application now or add more functionality for now but that would in turn make it much more difficult and time consuming to change to angular as a method of control later.

Angular

So after trying to learn how Angular works I created a branch to begin restructuring my application to build in angular support and get it to use angular to become a single page application. This process was initially fairly simple, it was important for me to find an appropriate tutorial to be able to learn what to do which took some time but I found a great article about how to create SPAs with angular however what I found is that essentially unless you want a javascript script to be executed at the same time as it loads you need to execute that script in a controller, the config element tells angular which html document to load into the home as a template and then the controller tells it what information to pass in like a script. I separated out my HTML map elements into separate HTML files which get loaded in and moved the JS into the controller, however when this happens the Javascript loads quicker than the HTML file meaning that the script cannot find the map container it is supposed to fit into and crashes. I tried many method of trying to solve this, I added code forcing the JS not to be run until the page was loaded however for some reason this code made the map load more than once cause the map to give an already initialised error. The other option I had was to add the map div where the javascript executes the code to the index seeing as the map happens on the index so it'd be ok for that code to be loaded in the first place and doesn't really need to be separated onto separate files and then have the rest of additional content to be loaded with Angular.

It seems the app could be changed fully into angular which would be great for really large apps, however having a loading splash page and ensuring that I don't have too many pages on my application might be preferable, plus at that point no further page loads would happen on presses so at least all of the information would be loaded. Also one of the big benefits of using angular is having a URL structure that the user can understand that even thought they're on a single page application the URL changes when they're in different parts of the app to reflect that however as my app is going to be a full screen application the user wouldn't have access to the URL bar so they wouldn't actually benefit from this.

Angular

So I made a decision about the use of Angular in my project and have now discontinued Angular support and usage in Proper Coffee, this was motivated by multiple reasons, whilst I can see the benefits of angular on my application whilst restructuring I came across many obstacles and Angular clashed with much of the script i'd already written, in fact there is a specific version of leaflet designed to run with Angular. After taking a step back and assessing the situation I decided that because I would almost have to begin the app from scratch to implement Angular especially as it's a language I'm not that familiar with and it basically forms the backbone of the application rather than just a couple of elements in it I think i'd have to work with it from scratch to gain a good understanding.

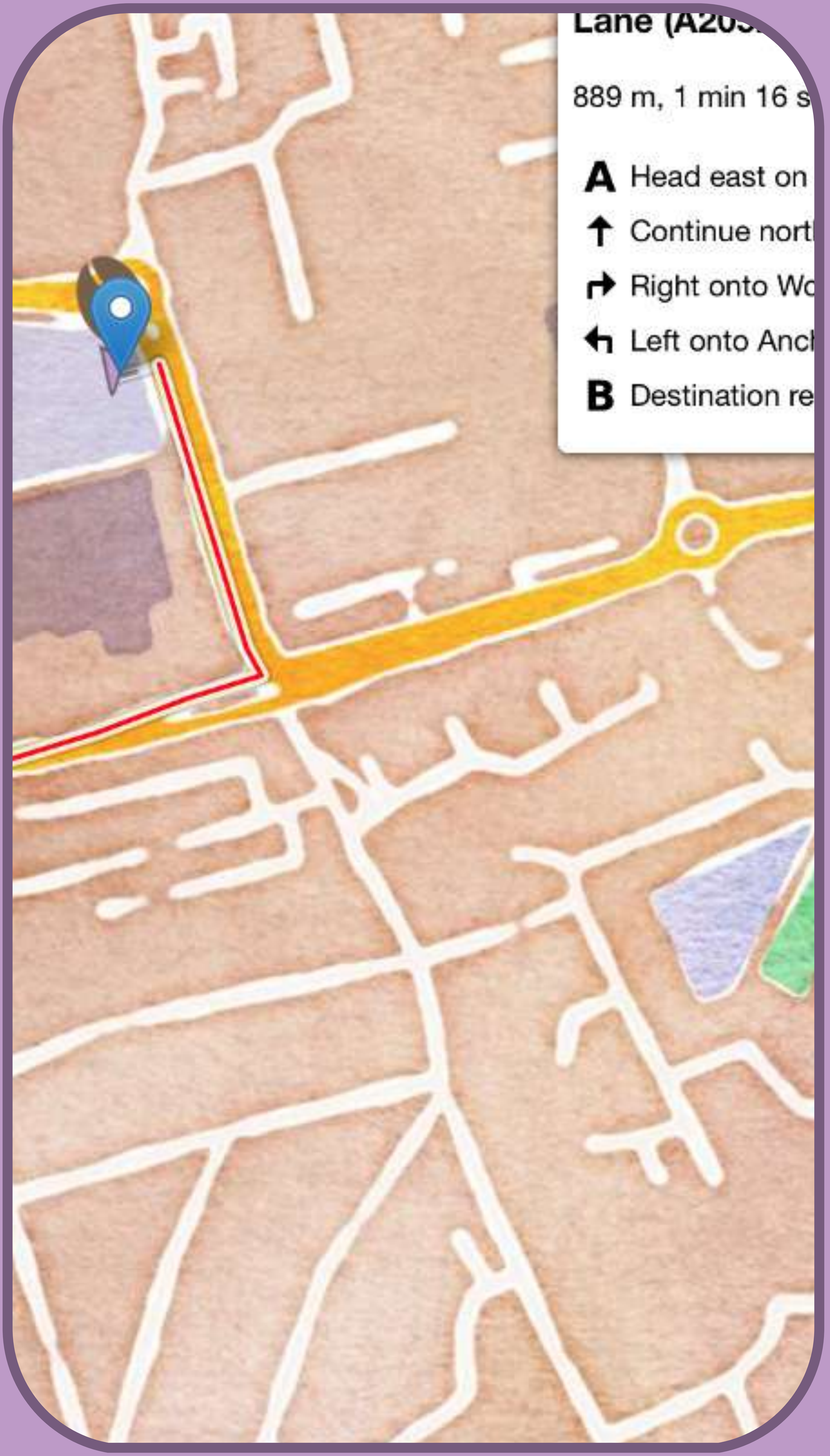
My application is really only going to consist of a few pages but dynamically display different content by pulling it from firebase and using javascript to change pages, I decided that the loading time wouldn't be impacted too heavily and that using a splash loading screen whilst all the content loads should be a adequate, i'd also be looking to port the application to native to run on iOS and/or Android so a lot of the content can be stored locally on those version to reduce loading times.

Angular support could be added later when I have more time to spare and am not prioritising application completeness and working with somebody who has a background in Angular.

Routing Machine

I decided to add the routing ability into my application now just to test how it will work and what I can change in leaflet routing machine as i used it before and remembered it being difficult.

By default Leaflet routing machine adds an Itinerary of where your route will take you, my app is generally going to be for people going fairly short distances so it probably isn't relevant to have that on, especially on mobile as it takes up so much space, the other issue with it is that it puts 2 markers on the map in order to draw a line between the 2, as my map will already have markers on this isn't ideal, I could style them to fit in with my application but that would still have multiple markers. The default line is also red so I want to style the line to fit in and referred back to my wireframes to ensure that the style I implement the routing matches the design that I'm aim for, this can be tricky as I'm dealing with a 3rd party javascript library but using it to fit around my needs so changing the way it works in order to fit my needs isn't always straight forward.



Lane (A20

889 m, 1 min 16 s

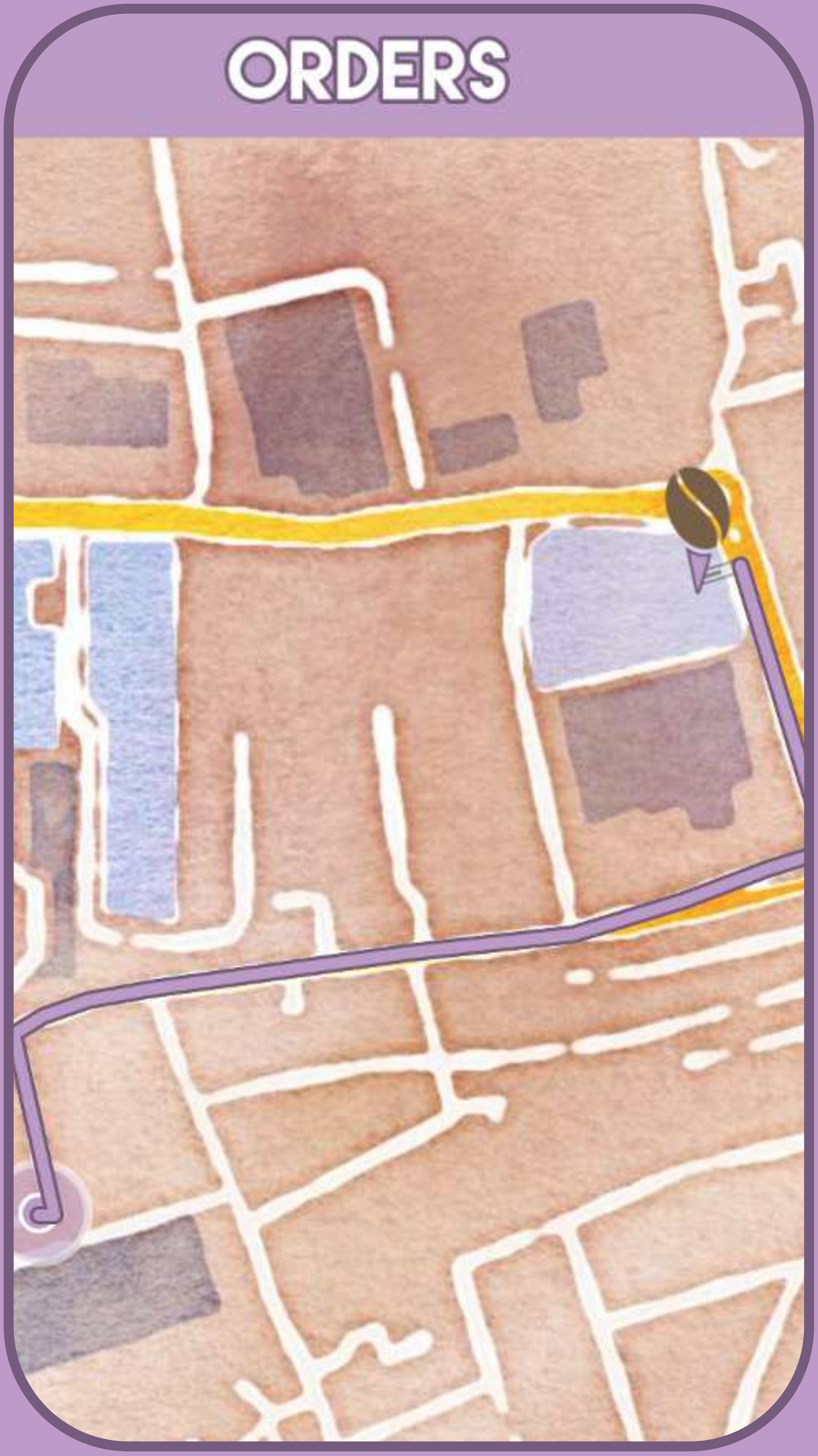
A Head east on

↑ Continue nort

↱ Right onto Wo

↰ Left onto Anc

B Destination re



Routing Machine

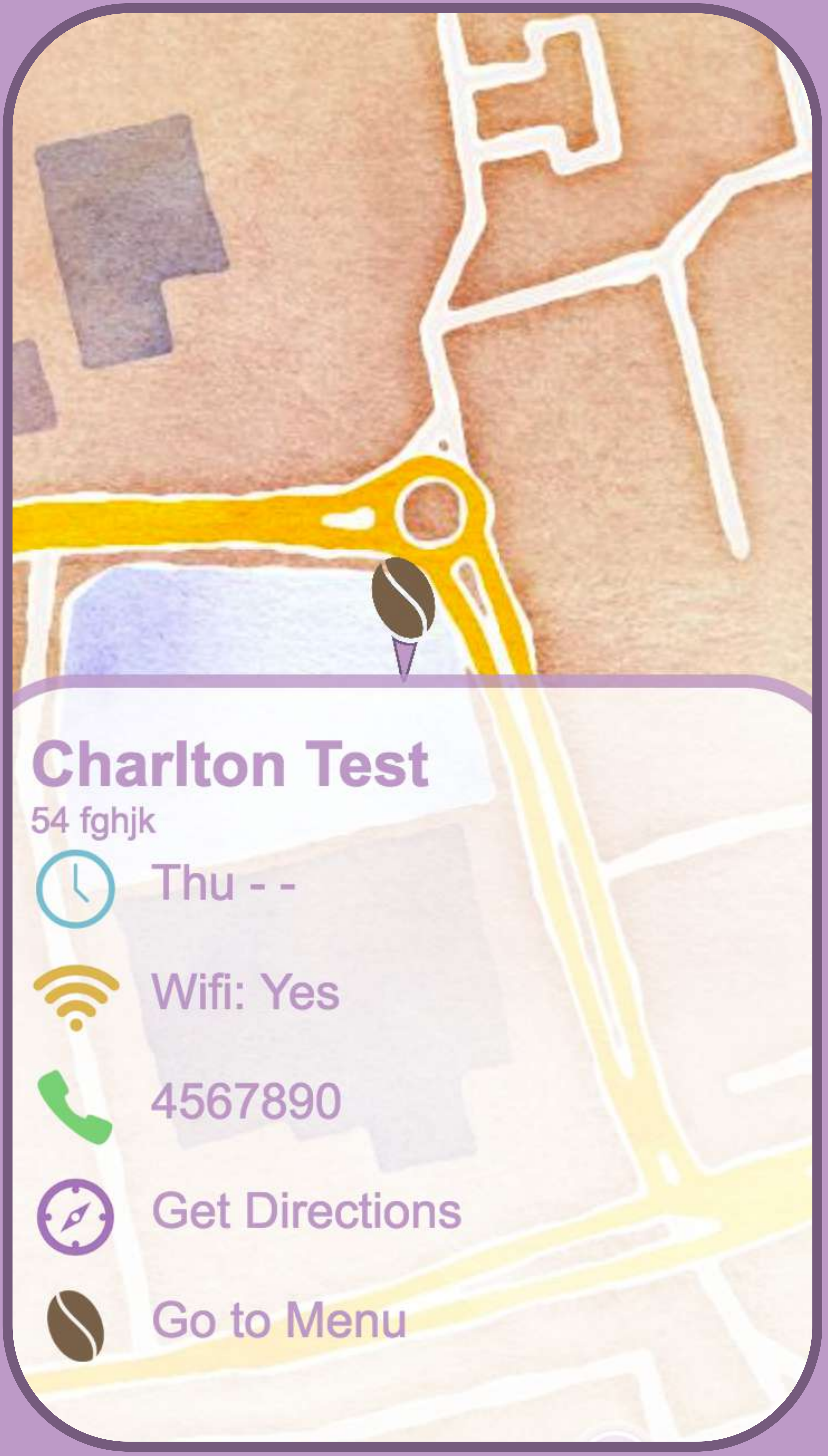
After reading a lot of the documentation and issues flagged by others I came to realise that whilst leaflet routing machine is pretty powerful the documentation is fragmented online with chunks missing and often confusing. I managed to set the itinerary to be hidden which seemed to be working at first however i realise once I tried routing to another coffee shop it was simply the contents of the itinerary that was hidden but the small box shows up, from just 1 routing the small box was hidden by the menu bar but when I ran another it stacks underneath making it visible. Initially I tried to replace the images used for the markers with transparent images which worked visually however I came to realise that they then stopped whatever was behind this transparent area to be unclickable so found some code to make the function create marker return null which seems to work at the moment. I followed the steps outlined in the API to change line colour but it seemed to clash with the JS file of leaflet routing machine which was strange and I eventually found another instance of somebody else doing it which worked.

So it works but I still needs work on moving the line to have a Z-Index lower than the marker that shows the users position, and clearing the path once you're done with it is as currently it'll just keep displaying all paths at the same time on the same map.

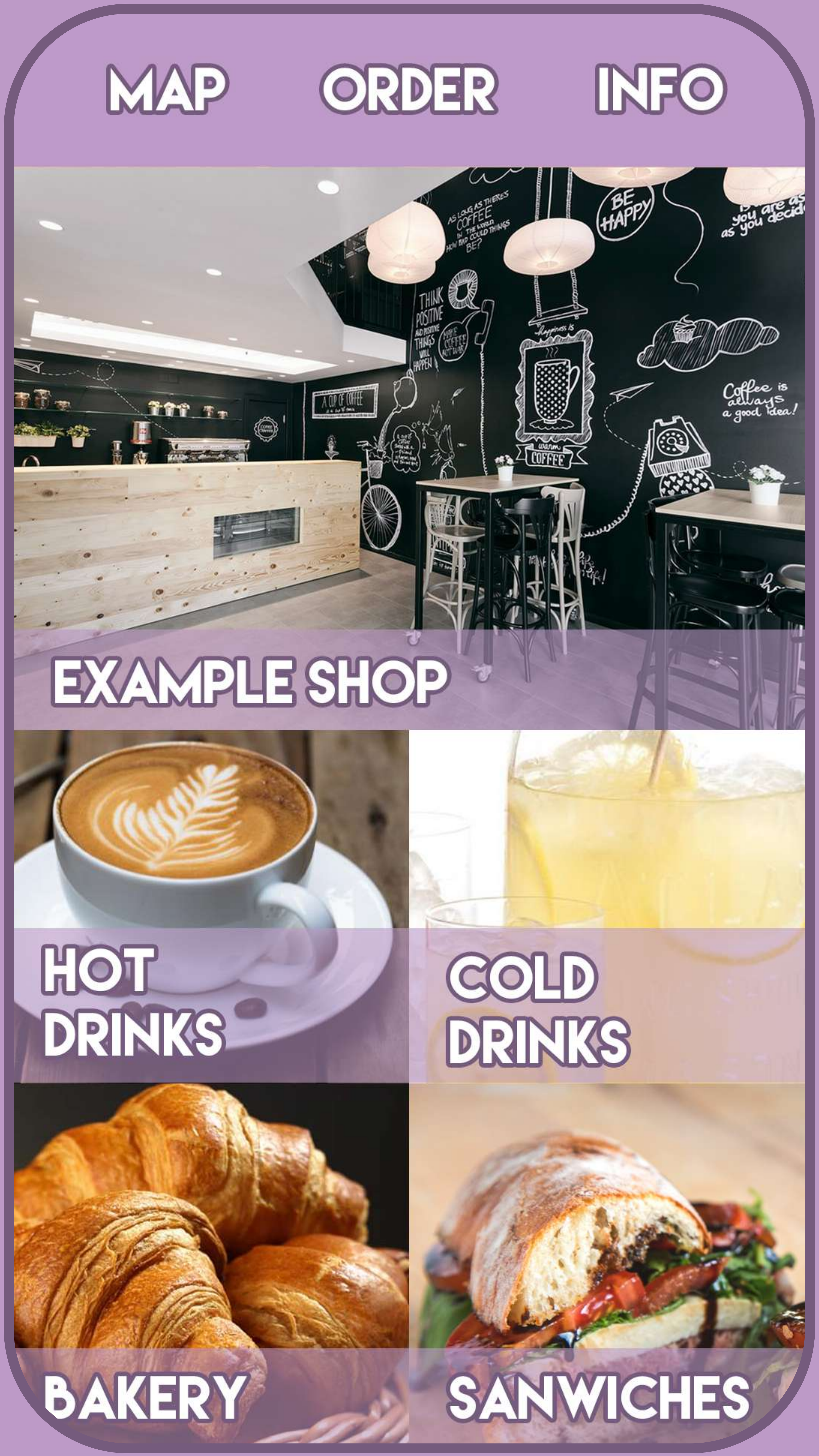
Menu Creation

To take this project forward I've decided to start working on the part where the user will be viewing the menu of a particular shop, the reason behind this is that once I have a template in place I can finally begin working on beginning the main work of the retailer application, the retailer would be adding the information that is being displayed in the menu part using the retailer application.

I worked with my wireframes of what the menu system should look like, I created a div in the index where the menu is going to be much like I did with the other page elements, the first thing I began working on was linking the popups view menu button up to this page and pushing the shop name information to this page to show that it's showing the relevant menu even though I'm yet to add a menu to the database for the shop as the retailer app doesn't yet have the structure to create the menu and they will be the ones who create it.



Charlton Test
54 fghjk
Thu - -
Wifi: Yes
4567890
Get Directions
Go to Menu



Menu Creation

I then got some dummy images and began implementing the category buttons using HTML and CSS, which took a lot of working and tweaking, at the moment the images used are images stored locally as I'm treating this menu like a template that eventually firebase is going to populate once I've made the part of the retailer app to input this information, I've imposed some restrictions upon the retailed in that I've decided the image of the shop should be at least 500 x 250 px and approximately that size or ratio otherwise cropping will occur, this is to prevent ugly image stretching and keep the app looking slick, I also decided that in the first Alpha build of Proper Coffee I will use set images for categories but that could be opened up later here.

A big difference between the wireframe and the page that I created is I've decided to add a back button from user feedback, the map button in the menu bar would take the user back but this may not be immediately obvious to some and also there are deeper menu pages than this so it allows you navigate back to the previous menu page without exiting all the way back out of the menu.

Menu Creation

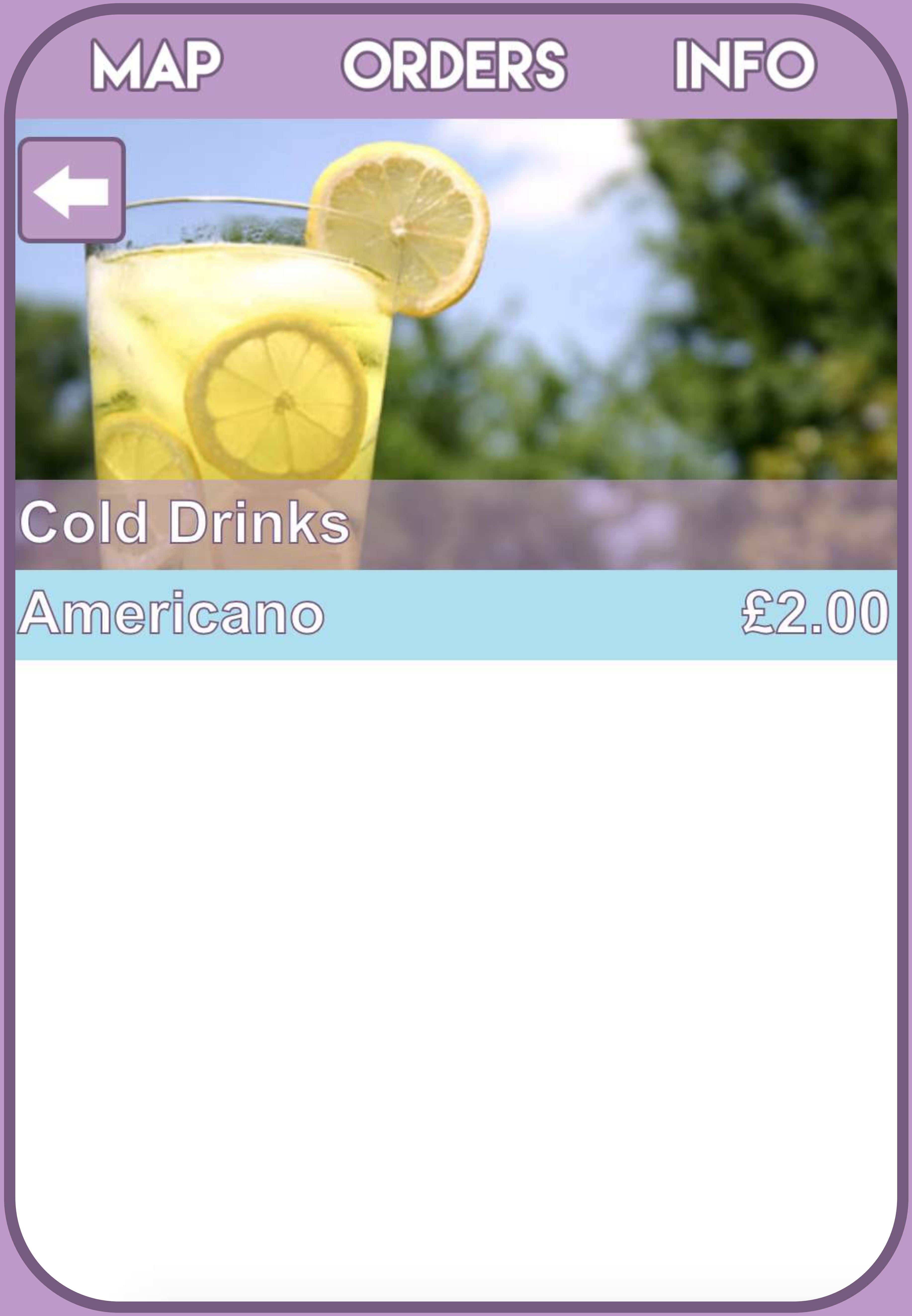
I began constructing the next menu page in a separate div on the index page, and I began to worry about quite how bulky all of my code was becoming, it was starting to look huge, so I backtracked and decided to come at it a different way, using Javascript to change, remove and add some of the elements from the first menu page allows me to have less HTML by using what I already had in place

Approaching it this way is a little more challenging because changing these elements mean dealing with the limitations of the styling I already have in place on these elements which is fine as it doesn't limit what I can do with these elements in anyway however where it gets tricky is if the user wants to go back a page, information like the image used on the first menu page is set by the user clicking on the marker, this is a totally separate function to the ones used by the menu which makes getting this image a little trickier but can be done, however it's not something I've implemented right now, the reason behind that is I just want to focus on adding functionality and then work on tweaking later so currently the back button which will be used for navigating the menu will just exit to the map.

The second page of the menu is what happens when you click on one of the categories, I change the image of the coffee shop to one that fits the category and the text to the name of the category, everything else is then hidden and only the new HTML added gets displayed. I only want to just put the 1 item in because ultimately all I'm putting in is dummy information and once it's actually pulling information from firebase it needs to find out how many items there are to display so adding more items to the menu is only creating stuff that I will later have to deconstruct.

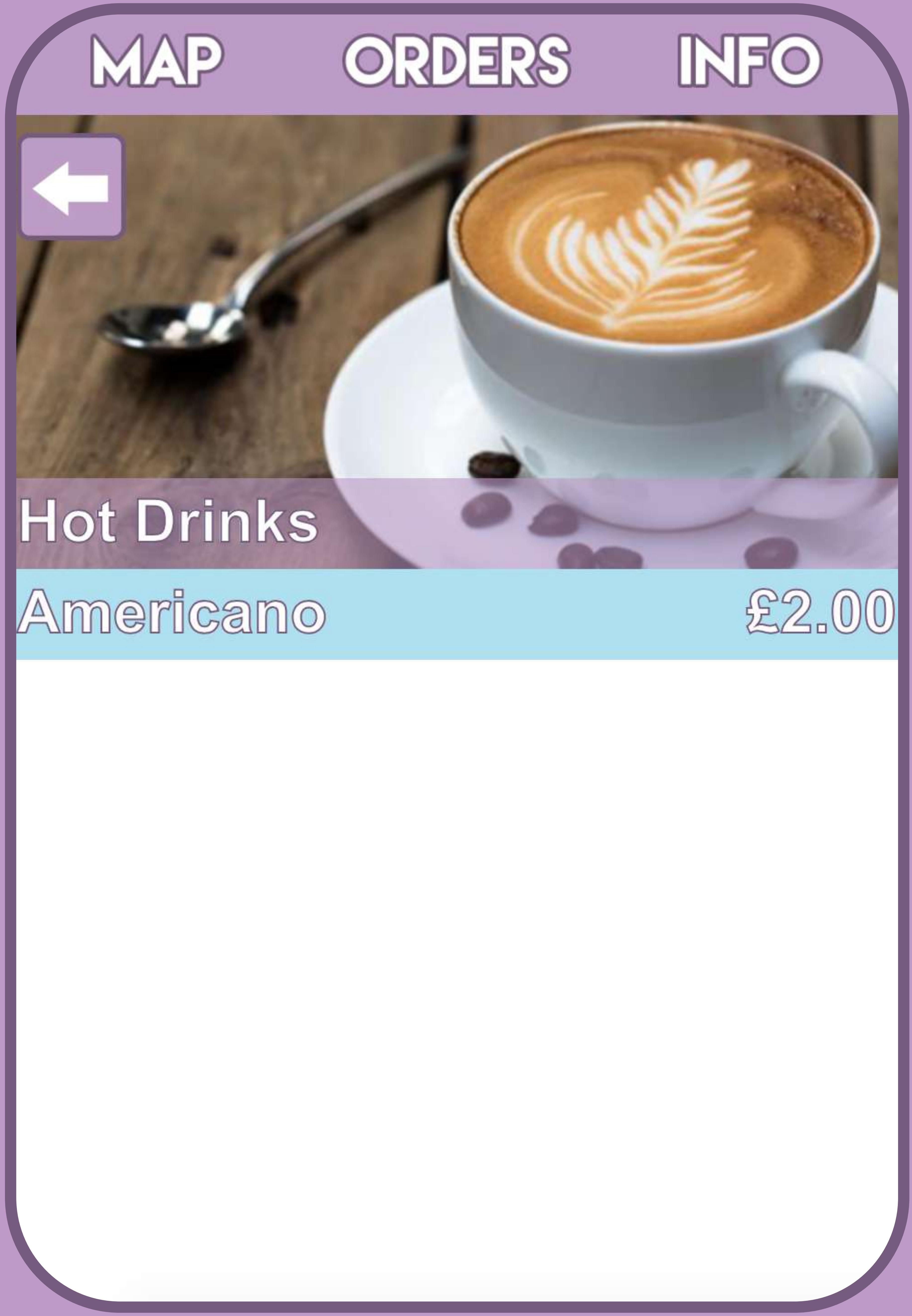
The other thing that I've been doing that is different from the wireframes is the font, the one I used in the wireframes is by no means the font I wanted to have on the final app as it looks a little harsh but it's just placeholder for now however one limitation that i've found so far is that in the wireframes I outline all of the text around the outside, CSS font strokes only allows you to center the stroke meaning it eats into the letter itself and I don't feel looks as effective.

Menu Creation



Cold Drinks

Americano £2.00



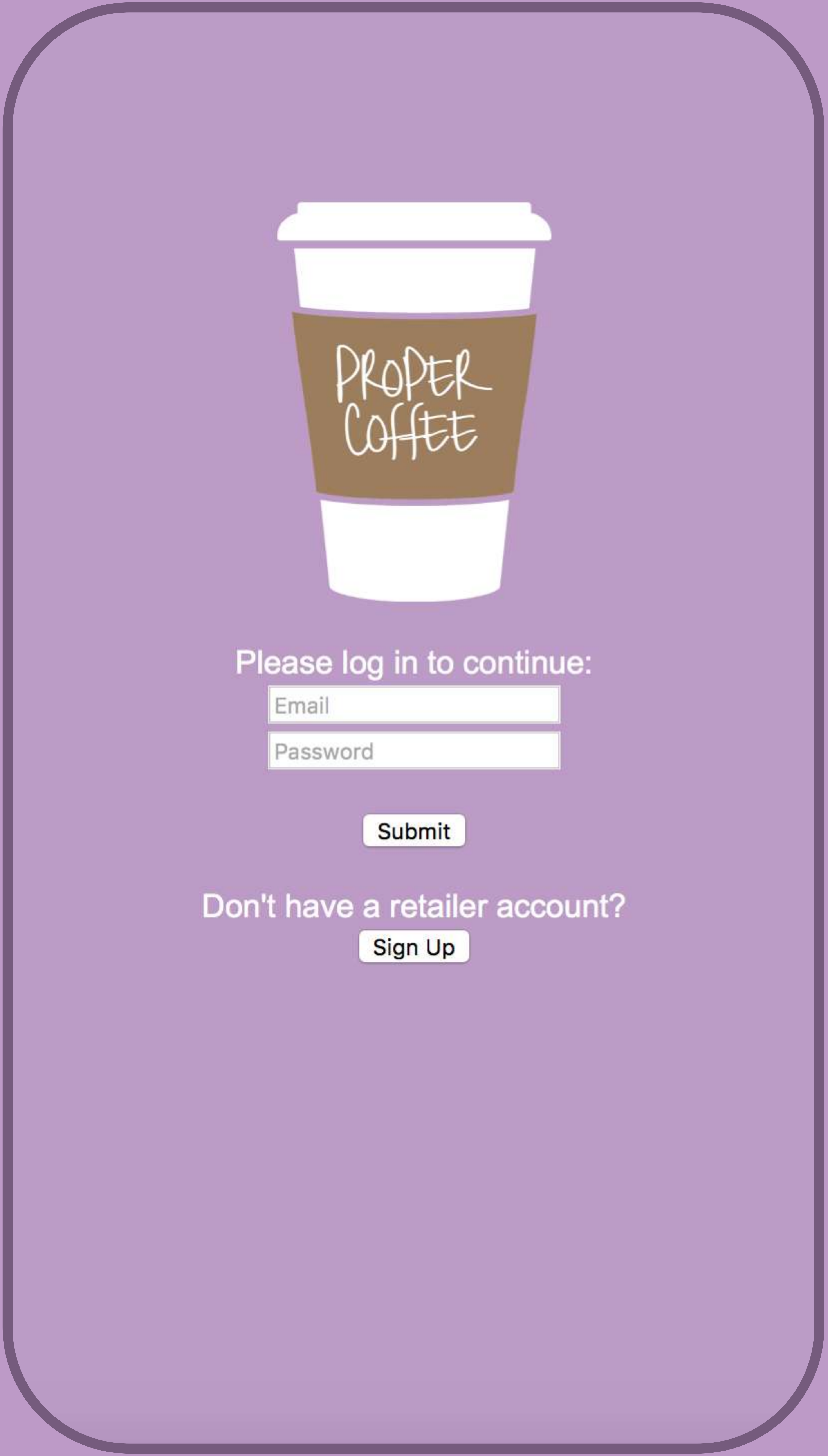
Hot Drinks

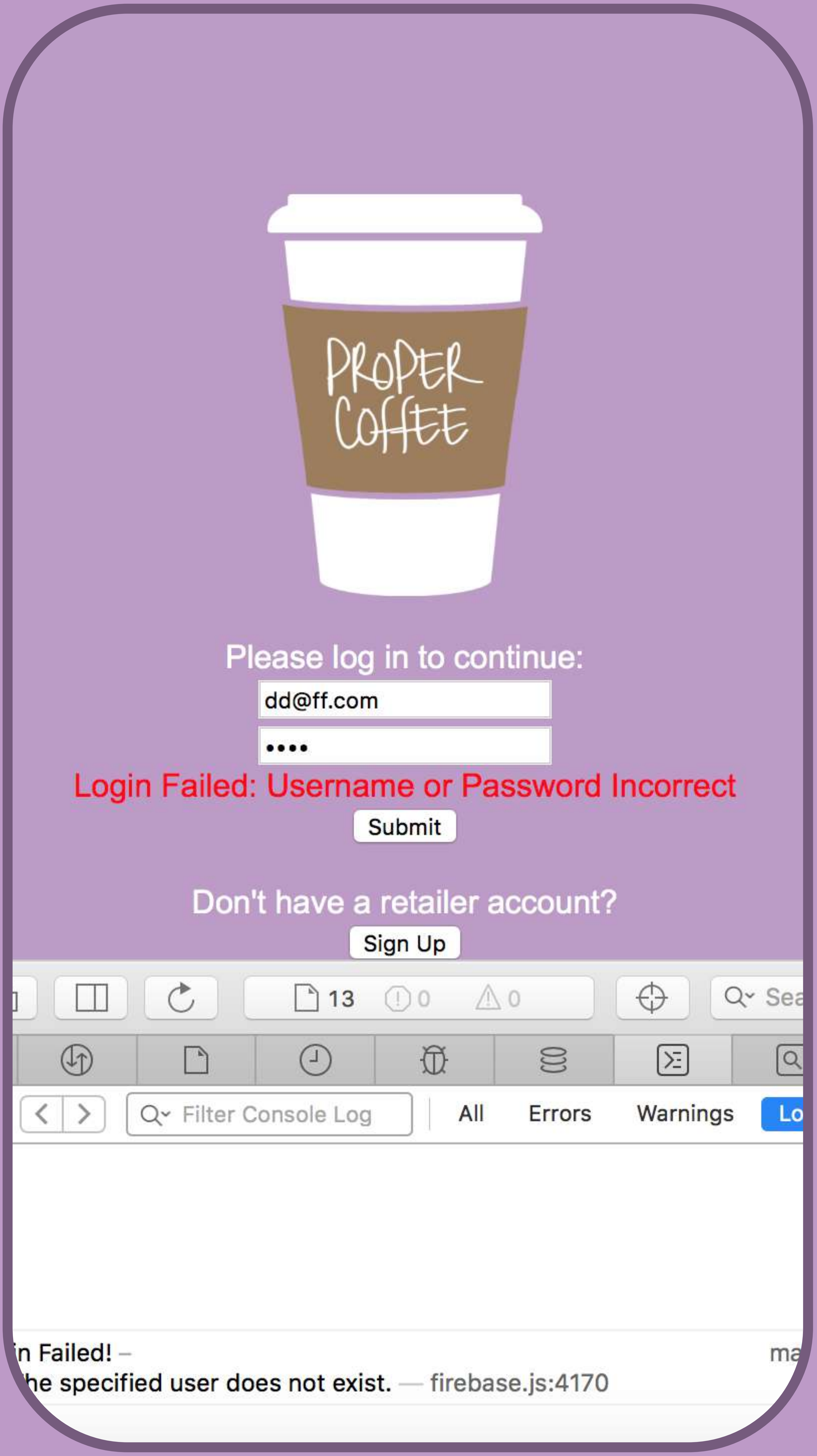
Americano £2.00

Retailer App

So now I have a template to put information about the coffee shops in I need to start adding support for my retailer application so that retailers can generate content, submit it to firebase and then populate my template. I already have my Retailer application boiler plate in place but I came to realise I don't want any person to just be able to come in and view my retailer application so I've now created a basic log in page to go at the start.

So now I'd made this basic log in page I had to make it actually work, I used my experience working with my admin application for this which allowed me to make fairly quick work of putting the form all together properly, then I began looking at the firebase tutorial to creating user accounts with an email address and a password, When you press the sign up button below the main form you get presented with a register page, it asks for a users email address and password and then submits this to firebase, providing that this is all done correctly firebase should accept it in which case you will loop back around to the log in screen ready to log in or you will receive an error telling you to try again.





Please log in to continue:

dd@ff.com

••••

Login Failed: Username or Password Incorrect

Submit

Don't have a retailer account?

Sign Up

```
n Failed! –                                                  ma
he specified user does not exist. — firebase.js:4170
```

Retailer App

A user can then use the form on the log in page to enter their username and password to get past this login screen and access the application behind it, there is an error message if the user makes a mistake.

So the log in system works, email and password is good because firebase has support for things like resetting users password when they forget them and etc so that’s part of the back end that I don’t have to worry about.

The next part to work on with this is partly coding but it’s also thinking about how exactly I’m going to make this work. I created proper coffee admin in the understanding that I’m going to be submitting the basic information for new shops myself or having someone my end do it rather than the retailer but then at some point the retailer assumes responsibility for their entry in firebase through the proper coffee retailer application, working out how I hand over from one to the other is important.

Retailer App

The email address, password and the User ID that firebase creates are in no way actually linked to a specific shop yet, it’s simply an account with no permission to do anything, what I need to do is to give specific account access to specific shop data. I don’t want to pass over the job I‘m doing with admin over to retailers as it’s open to potential abuse of anyone just making a shop but it means extra work for me as it means creating a user friendly way of finding things like longitude and latitude and submitting them and I want to make sure that the shops added are really great and not any cafe selling instant coffee. Right now once somebody has signed up I can then add permission manually for them to have access to specific data and edit their own it’s just how I’m going to do this on a wider scale that needs to be considered.

I created the proper coffee admin application to submit data for individual coffee shops to my firebase and it works by submitting the data to a branch in my database called shops, this currently works with no big issues. When I came to develop the retailer application I created a method for retailers to fill out a registration form and a way for them to log in to my firebase however that log in isn’t associated with a particular shop in any way, the challenge I’ve been facing is linking these profiles to an existing shop and doing it in a way which keeps new retailers signing up a low maintenance process for myself.

There are 2 important things that need to be done with a retailer account to make it accessible, I need to link the account to the correct shop so that when they sign in with an email/username and password the application knows which shop they’re trying to access and then displaying relevant data and I need to then give them security clearance to be able to read and write data for their shops entry.

I began to think that I could add an option for a new user signing up to choose the relevant coffee shop by pulling the full list of shops in and having them choose their shop which would create a link between their account with but this would require a review process and no instant access which didn’t feel like the most sensible move.

Retailer Accounts

Firebase allows for users to edit their own data by adding rules to the security which allows data stored within a unique 'uid' and can only be written to if a user with the same user id 'uid' is currently authenticated by being logged in, this allows for a low maintenance way of letting users be able to edit their own data when logged in but not others by adding just one rule. This led me to realise that the way I'm currently storing my shops data only makes sense if I'm the only person who needs to have access to it. If I want a user to have access to shop data this data should be assigned to their account in the first place, if the shops are in fact actually stored as user accounts then as soon as a user is logged in with that account I can make the link between their account and the shop data they need access to automatically, it would also allow me to create rules similar to the one above so that they can read and write their data but not anybody else's.

Although it felt slightly like tearing apart my own application I had to make fundamental changes to the way that I stored data, and the application which does this storing of data; Proper Coffee Admin. If when I create a new entry for a shop I create it as a user account rather than a data entry I can hand it over for the retailer to use from that point on, all i'd have to do is set a username/email and password for them myself when I create the account potentially by using a randomised password so i can inform the retailer without it having the potential to be compromised. I'd then have a new database which has all of the shop entries stored as unique user ids. Then once a user logs into the retailer application, they're given a prompt to edit their email/username and password and they're ready to go and I can start pulling in the appropriate data for them to use. This is going to require me to do some massive structural changes, to the way not only that the firebase stores the data but subsequently the way Proper Coffee Admin submits data, the way that the consumer app access the shop data and ensuring that when a retailer makes changes to their account that the GeoFire database is also updated without giving them access to my full GeoFire database. Progress on this could be fairly slow as i'm going to have to experiment with different ways of implementing this.

Updating Admin Form

I made some changes to the way that proper coffee admin works, as detailed before I needed to create a user account at the same time as entering this data.

How Proper coffee admin now works is the same but now there is also a email and password field, this email address should be the one of the retailer however if not a fake one e.g. coffeeshop@proper.coffee can be put in and the password from a random password generator. After I submit the form the app creates a user account, the user account then logs itself in, once logged in the user account creates an entry in the shops data base using it's own unique userid which firebase just created as the name of the entry, the security I have in place upon this firebase allows this as the user is allowed to read and write to an entry inside the shops category as long as the name of the entry is exactly the same as their user ID and that they are currently logged on with authentication from Firebase so they can add this information but not change the rest of the firebase at all, then just as before the longitude and latitude are extracted and a new Geofire entry is also added with the same unique user ID, the security works for this in the same way.

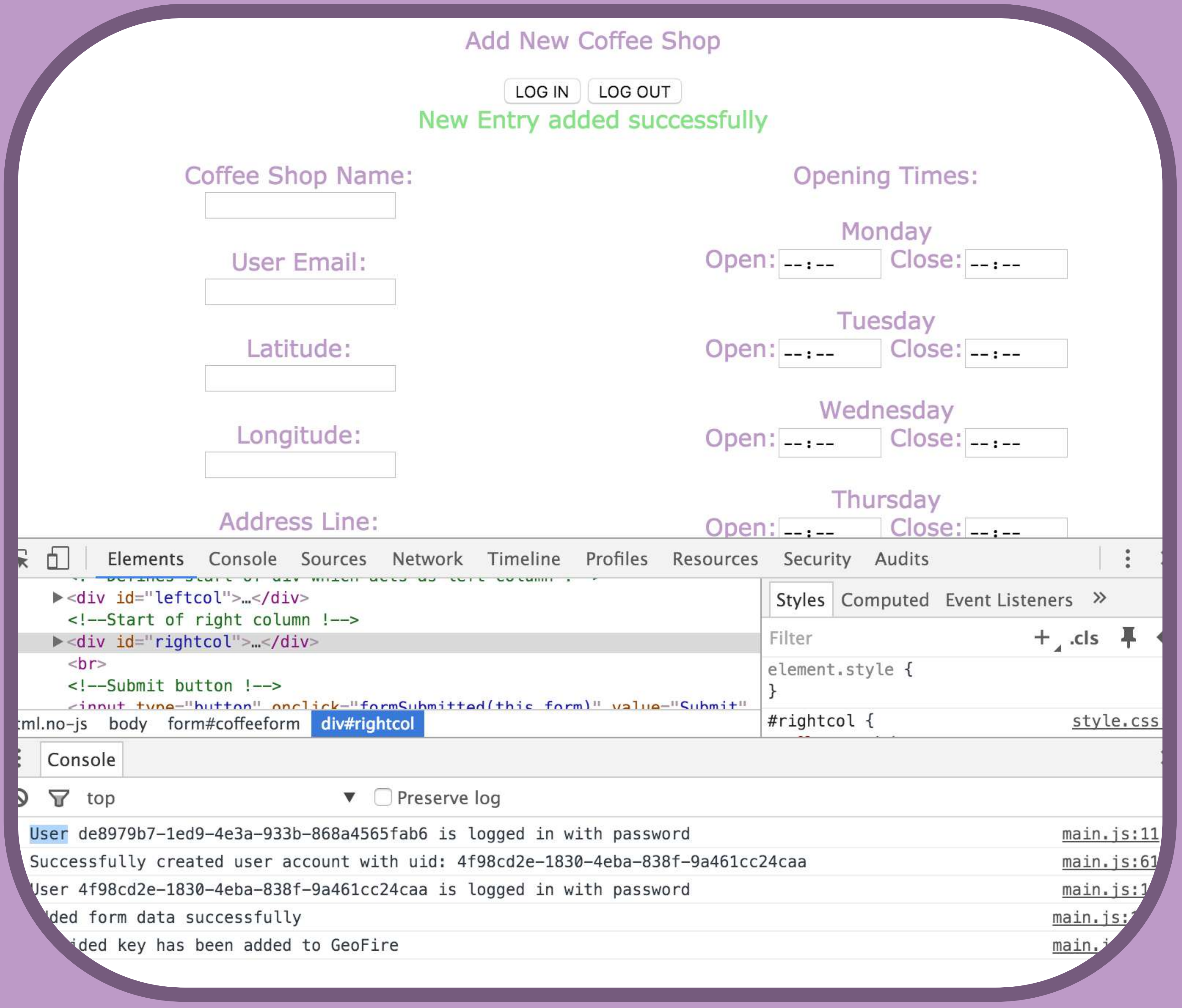
Proper coffee admin also now gives more detailed responses if an error occurs it displays it as text on the web page incase one day it isn't me entering it so you don't need to look in the console to find out what's gone wrong. I tested all of this in parts of the database the main application doesn't currently access and created new subcategories called retailers and retailer Geofire however once I'd found out that it was completely working it was no problem to get the data to be pushed to the old database that I was using so I don't need to change any of the other applications and the consumer app will just continue working.

Updating Admin Form

Next I had to delete the content of the current database as those entries didn't have the user name etc created alongside them, getting it back to how it was before is just a simple case of data entry and I exported what i had before ready to re-enter it however I feel like I may end up in the same situation again so for now am only entering the data for 3 shops for testing purposes and will added the full data set later when I'm more certain I don't need to make further changes to the form.

So now I can work on the retailer application as once the user has logged in I have a method for pulling up the relevant shop information.

I also added another change to the way Admin submits, it now adds some extra data to the database, one entry called first log in which will allow the retailer app to detect whether the user is logging in for the first time, if so it prompts them to change their password to their own choice and one called menu blank, this is so the consumer app can tell whether there is a menu available for a specific shop so it can adapt the UI and options appropriately.



Add New Coffee Shop

New Entry added successfully

Coffee Shop Name:

User Email:

Latitude:

Longitude:

Address Line:

Opening Times:

Monday
Open: --:-- Close: --:--

Tuesday
Open: --:-- Close: --:--

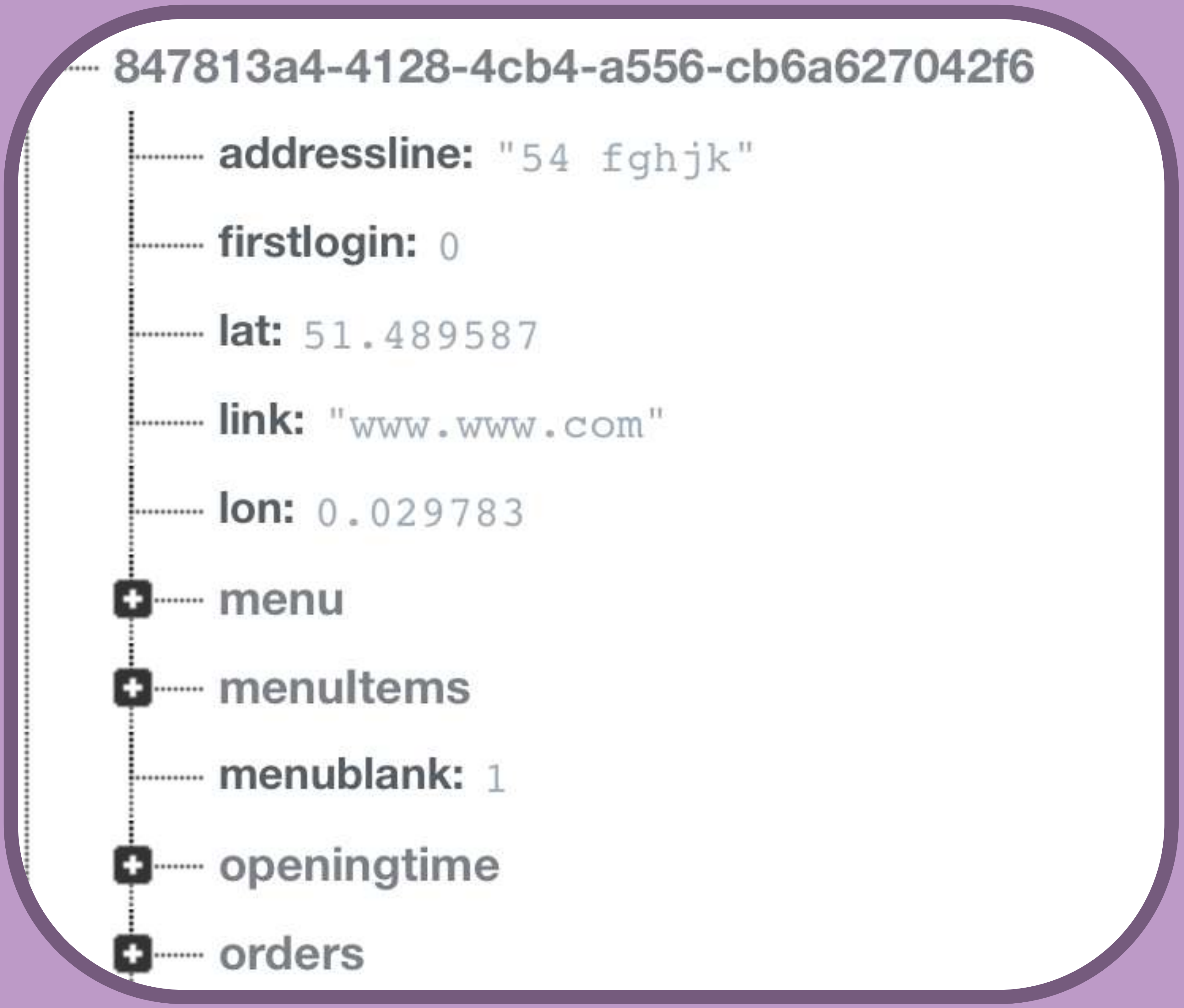
Wednesday
Open: --:-- Close: --:--

Thursday
Open: --:-- Close: --:--

```
▶<div id="leftcol">…</div>
 <!--Start of right column !-->
▶<div id="rightcol">…</div>
 <br>
 <!--Submit button !-->
 <input type="button" onclick="formSubmitted(this.form)" value="Submit"
```

```
element.style {
}
#rightcol {                                   style.css
```

```
User de8979b7-1ed9-4e3a-933b-868a4565fab6 is logged in with password            main.js:11
Successfully created user account with uid: 4f98cd2e-1830-4eba-838f-9a461cc24caa main.js:61
User 4f98cd2e-1830-4eba-838f-9a461cc24caa is logged in with password            main.js:1
dded form data successfully                                                     main.js:
ided key has been added to GeoFire                                              main.j
```



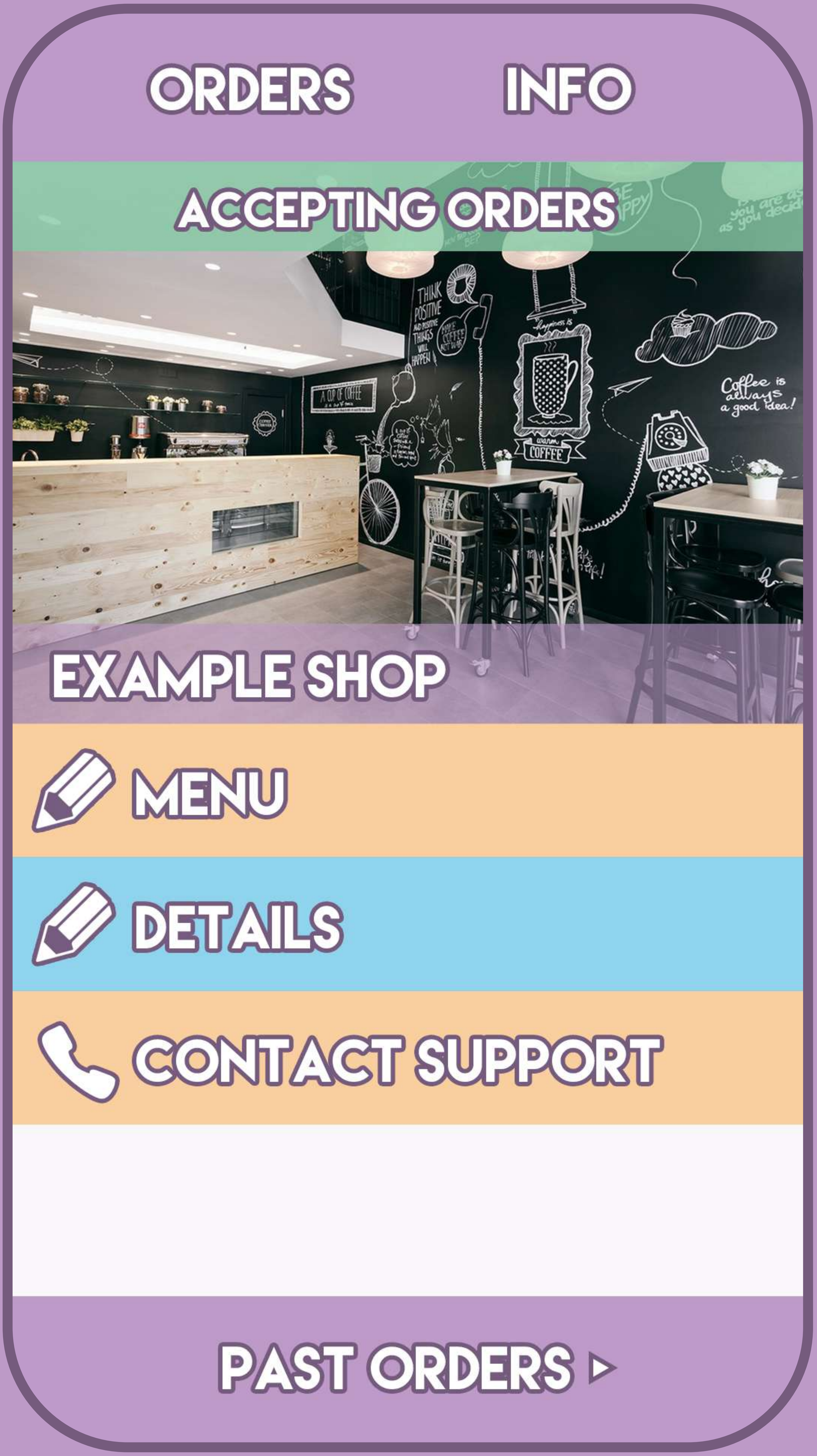
```
847813a4-4128-4cb4-a556-cb6a627042f6
  addressline: "54 fghjk"
  firstlogin: 0
  lat: 51.489587
  link: "www.www.com"
  lon: 0.029783
+ menu
+ menuItems
  menublank: 1
+ openingtime
+ orders
```

Retailer App

Now that I have a new structure for how retailer accounts work I began to add support for the retailer application to be able to log into these accounts and pull in relevant information, ensuring that the set up process is going to be obvious and clear to a retailer is important so once the application is active I intend to send them a first time log in email to the email address they have provided for me as the one they would like to use to log in, this is email will contain a password they can use the to activate their account which will be randomly generated.

Having this randomly generated password as their permanent password isn't ideal as I'm emailing them a plain text version of this password insecurely this is where the admin app setting a new shops "first login" status in fire base to 1 comes in useful, the application accepts the temporary password at first and logs the user in however instead of giving them access to the application straight away it shows them a page that most users will only see once which prompts them to change their password, it automatically uses the data provided when the user logs in to remember information like the unique user id and email address and asks the user to enter their existing password and then a new password twice (once for verification that they've entered it correctly) whilst I can assist a retailer with resetting their password later I don't actually have access to the password they have chosen as this is stored securely by Firebase outside of my database, the user then has a password for their account which is easier for them to remember and not one that has been transmitted or stored insecurely.

The app then sets first login to 0 so the user isn't prompted to do this every time. This process was a little complex as I've used the Firebase onauth function in my app which runs specific script when a user is logged in and that bypassed the password change page so I had to turn on and off the listeners for onauth at certain parts of my script to have it function correctly.



Retailer App

I then started working on the info tab of the retailer application which are based on the wireframe I created for this page, I could reuse some code I used for the consumer app menu system as it is supposed to mirror what the user sees but allow the retailer to make edits which slightly sped up the process. The retailer application checks the unique user id of the email and password and because of the new way in which the admin application submits data it also uses this unique user id as the shop identifier which allow the retailer to pull in the information of the correct shop and using fire base rules I added security so that if somebody took my app offline and hacked it they still only have write access to the shop data of the shop matching the unique user ID and password that requires the user to be logged in with authentication making somebody unable to access it this way.

Now that the retailer has write access to the relevant shop securely I could starting implementing ways for them to make changes, the first item I wanted to allow them to edit was the coffee shop profile image as I felt this would be one of the harder ones to implement as I haven't tried to submit/change this type of data before and because currently it is the only field of the shop that displays data which isn't actually accurate for that shop as it's just an image file that I've uploaded alongside other website content.

Retailer App

I discovered that there is no native way for Firebase to store image files within my database, Firebase has support for adding strings and numbers etc in my database as JSON but not images currently and after researching this I found multiple potential ways around this. One way of doing it would be a buy some cloud storage from a third party such as Amazon which I investigated, essentially a user would upload an image file to a different server which I has assigned them security and permission to be able to do and using information such as the users unique user ID as a file name to store them make code which defines the path and pulls in the relevant image file, this method makes sense however would take a while to implement as essentially is have to set up a new service and security for this and research a way of submitting these images in a different way to how firebase gets data added to it, however in the long term this would be a big benefit as it would allow me to make the application scalable if a lot of image files were being submitted.

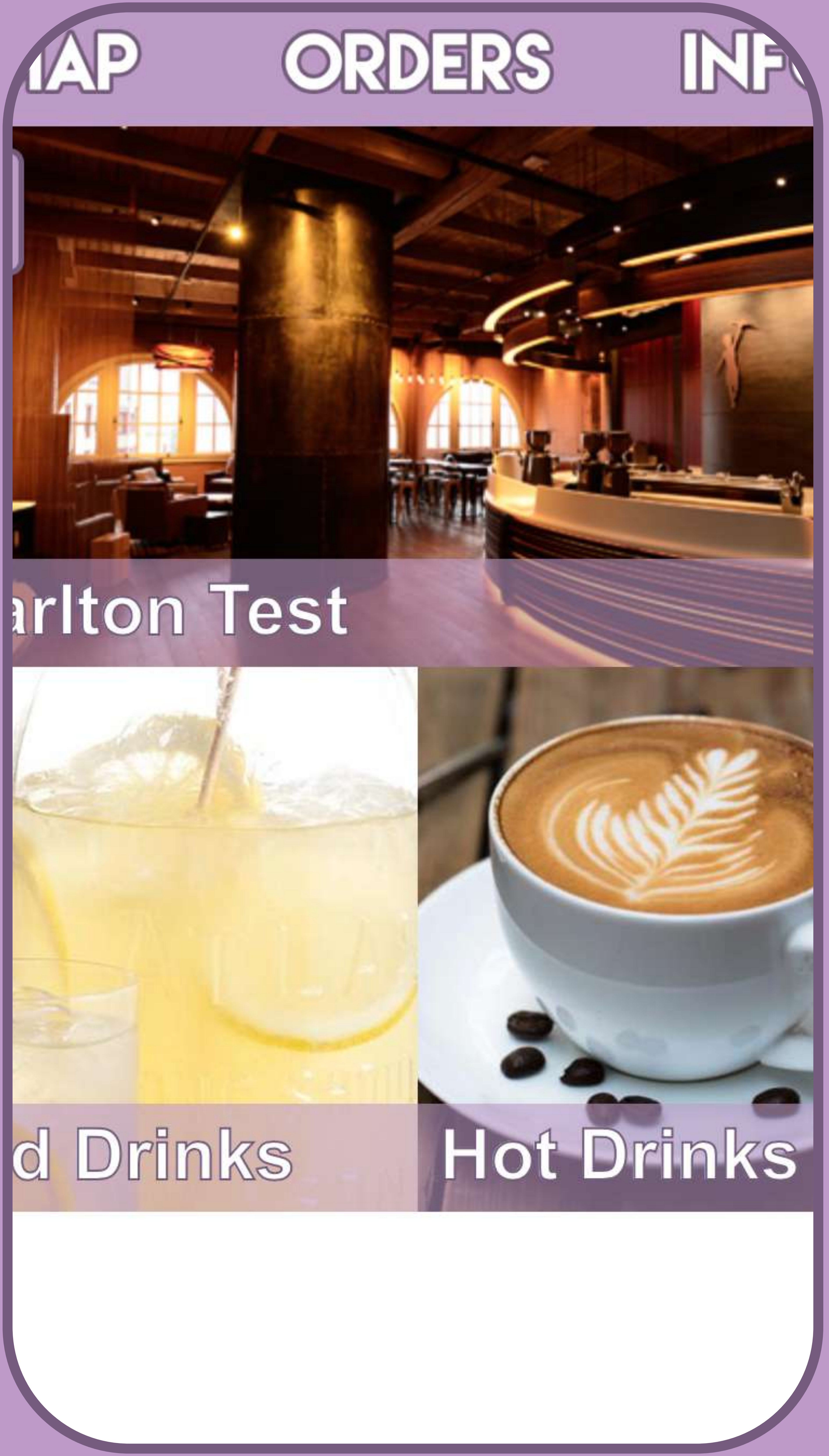
Another way I found of achieving what I need to is to store the images in firebase but not store them as images which is possible, viewing the image locally and using a JavaScript file reader to translate the image to base64 allows me to be able to store the image as a string in firebase. Using this image storing method wouldn't work for image files over 10mb and may not really be scalable as it would take up a lot of space in my database if I used this method for thousands of images but for getting this functionality working as quickly as possible it seems like the most appropriate and easy way of achieving it as I don't need it to be that scalable yet. This method of writing an image to firebase isn't really well documented as its not really good form as its not built for it but I found a git called Firepano which does it, Firepano is cool because it has a structure so that you see the image in place once it's been selected and displays a loading spinner until it's been uploaded making it easy for a user to understand however all of this was designed around a specific application use so the coding and styling had to be changed, it also has a bunch of other complicated stuff that needed editing out like it automatically changing my URL and it generating unique hash numbers to store the images under.

Retailer App

Creating the retailer apps menu creating panel was harder and more time consumer than I anticipated, I figured once i'd got the images out of the way the rest would be less challenging however that didn't end up being the case.

After refining the code from the image uploader more and merging it with my existing script I moved on to adding more functions. In order to have the retailer be able to edit their menu I wanted to recreate the consumer side but with edit buttons etc so they can see the same view as the consumer whilst they edit it this is where the first issue came in, whilst I had managed to make the app look like everything was correct on the consumer side that was not the case, the way it represented the menu categories were bad CSS which didn't stack or do anything as I wanted it to so the first thing was to work on that and ensure it stacked correctly.

Eventually I improved the css to make everything more controlled, it also meant that I could give each item an over-riding id. I cloned this over into the retailer app which left it with no back button and retained the edit image button temporarily. I decided to add an additional category when viewing the retailer application, an add category icon, however after implementing it I realise that that made sense for the initial usage of the app but not for editing later so I changed it to an edit icon, I don't think I like the way it looks alongside the other categories but I can refine the style at a later point.



Retailer App

I then had to make some changes to the admin application, in order for the retailer app to be able to detect which categories the retailer is using they needed values within firebase, so I set it so that when a new shop is created it sets the 6 categories in the menu object and set all of their value to 0.

I then started working on the script to power it, I first used firebase in the same way I normally would to get the values of what was currently in the database for the menu part, then I had to work out what to do with them, initially I made all of the different categories in the HTML and had the javascript display and hide them, this worked however I began to realise that once I added editing function on top I will then have to add to each HTML element again and it would soon become bulky. I then created plus and minus icons to be overlaid on top of the categories when the retailer goes into edit mode, overlaying these proved tricky as my previous stacking system was already so confusing and precarious but eventually I managed to get them all into the correct place and then hid them all. I also began to realise that I had left myself no scope for adding categories at a later point and that I'm be stuck with the 6 I've thought of right at this time, that was the point that I realise I needed to think of another way to do this and restructure the method I was using.

I decided to create a separate database in firebase called categories, inside that database were the categories stored in the same name type that they are in each of the shops and their value is a string which is how they would be written to the user, the javascript can now access and read this categories file so the retailer application is aware of all possible categories yet doesn't change the data in that shops object. I then removed all but one of the HTML categories, I created a script which onauth when the user logs in and we know which shop to go to checks the categories database available to everyone, it then writes HTML to the app in a cycle using one script to write the HTML category block for each of the 6 categories, it then checks which categories have a value of 1 in their shops database, displays them and then hide all the rest.

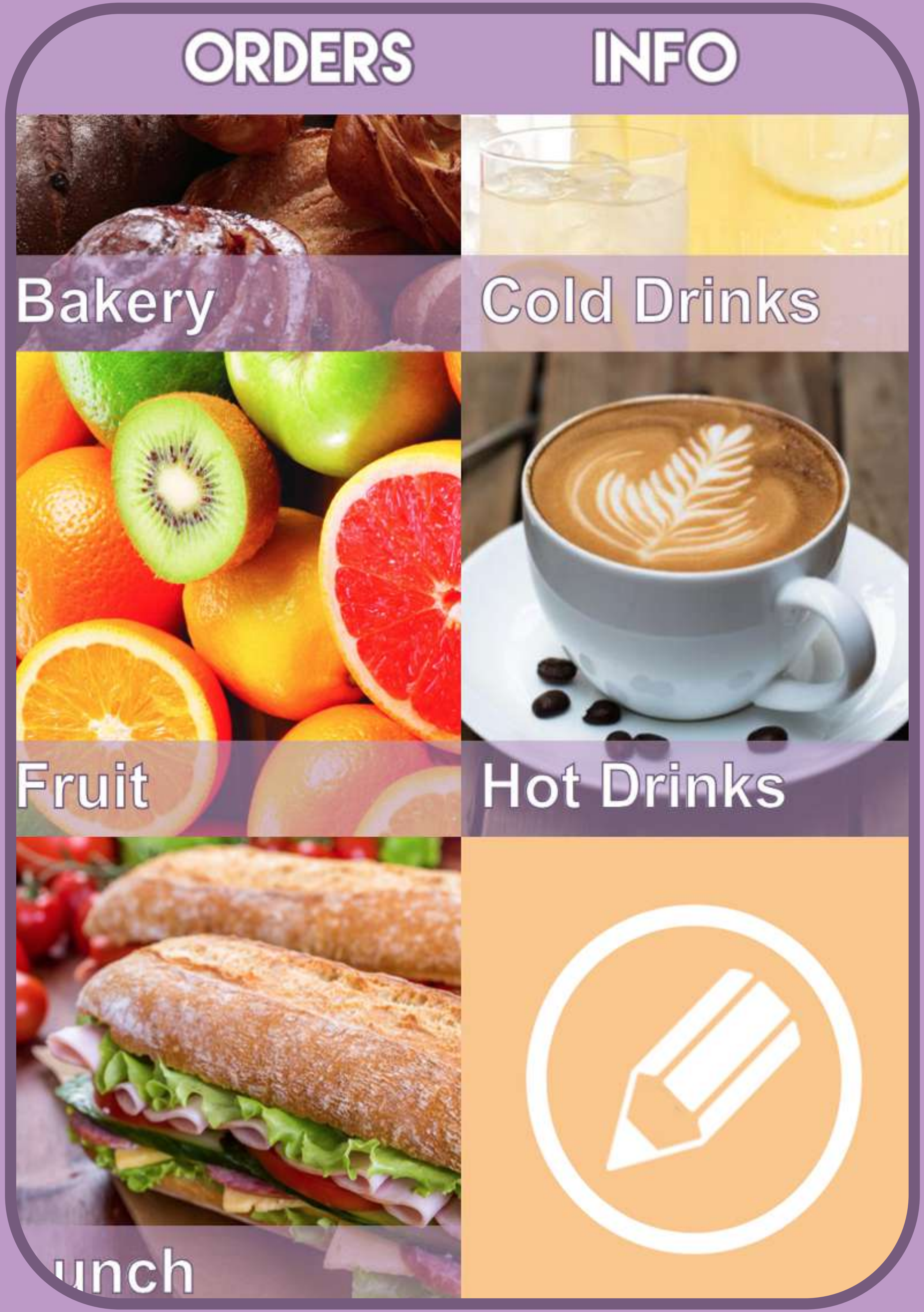
Retailer App

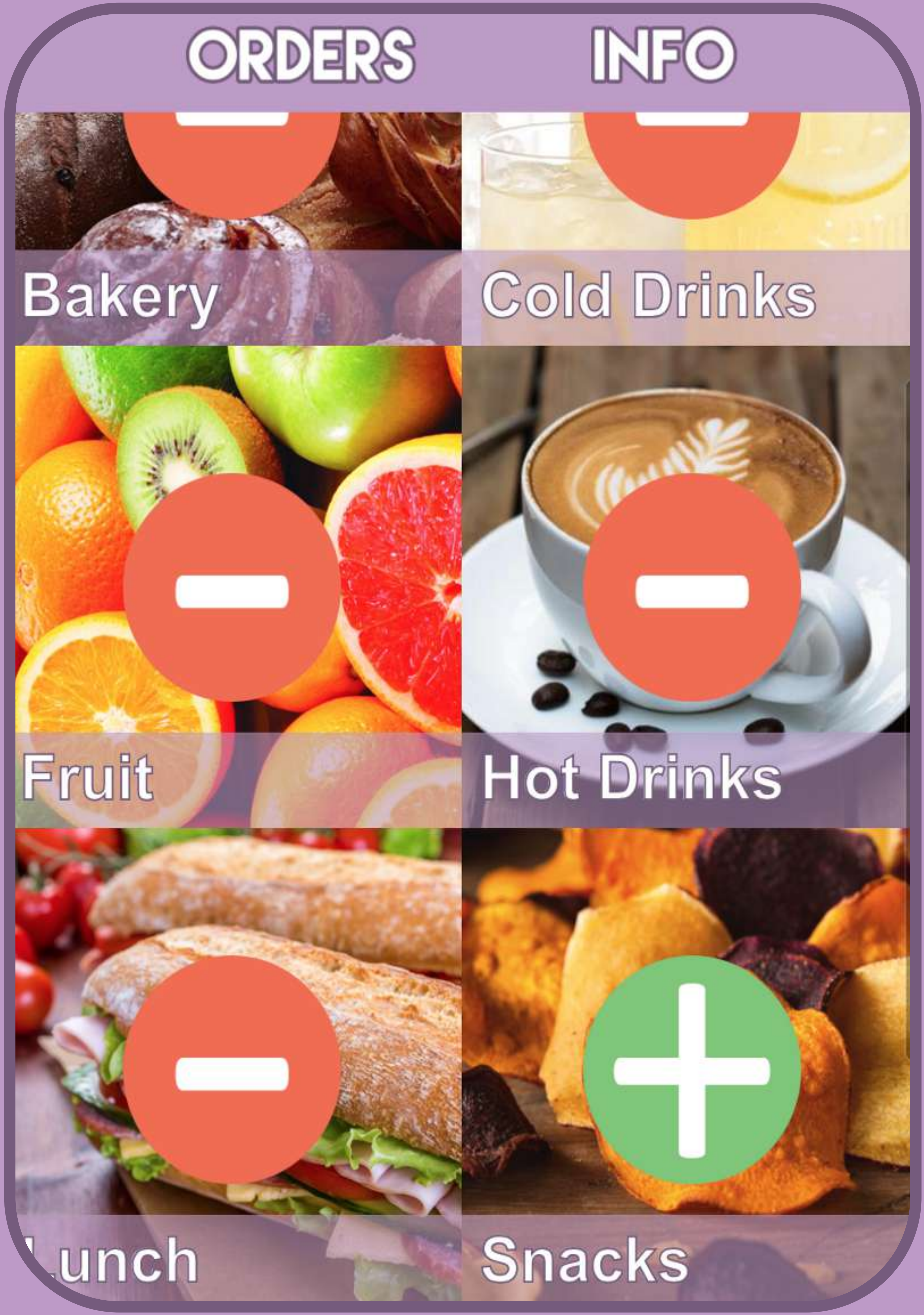
When the user then clicks on the edit button the script shows all of the possible categories available, it detects again which ones the shop currently has and displays the minus icon on top of those and the ones it doesn't with a plus on. If the user then clicks on one of these icons it will write what the user does back to firebase and either add or takeaway the subject from the shop. This took a lot of work and refining, I came up against a huge number of bugs, there is still a minor one where on occasion everything disappears but I think I know that this is being caused by the script trying to act before it has managed to write to firebase which is why it's intermittent.

I had an issue where whenever you added or took away a shop it loaded another 6 categories and another 6 categories looping the same ones over and over because of listeners on the database that are supposed to be checking for changes but because I was using onauth was constructing multiple times, I also had to add new images for the categories and the backgrounds once you have added them. The name of the functions for what happens whenever you press one of the categories is generated by the javascript writing the HTML, and as such they don't attach the database to the category in anyway which means at the moment that all 6 categories not only have their own functions but 3 functions each, one when pressed, one when added and one when taken away. There has to be a better way to do this I just haven't worked it out yet.

I added the back button and removed the edit button from the categories page and created the on click events for each of them so it goes to their specific page (all the same page but javascript changes elements to make it look unique) and then began to get the consumer app to represent this. The consumer app was less work but because of different naming structures in both applications and the different ways they work it took a while to translate it all over however it now works for the first time so changes made on the retailer app are represented on the consumer side too.

Retailer App





Retailer App

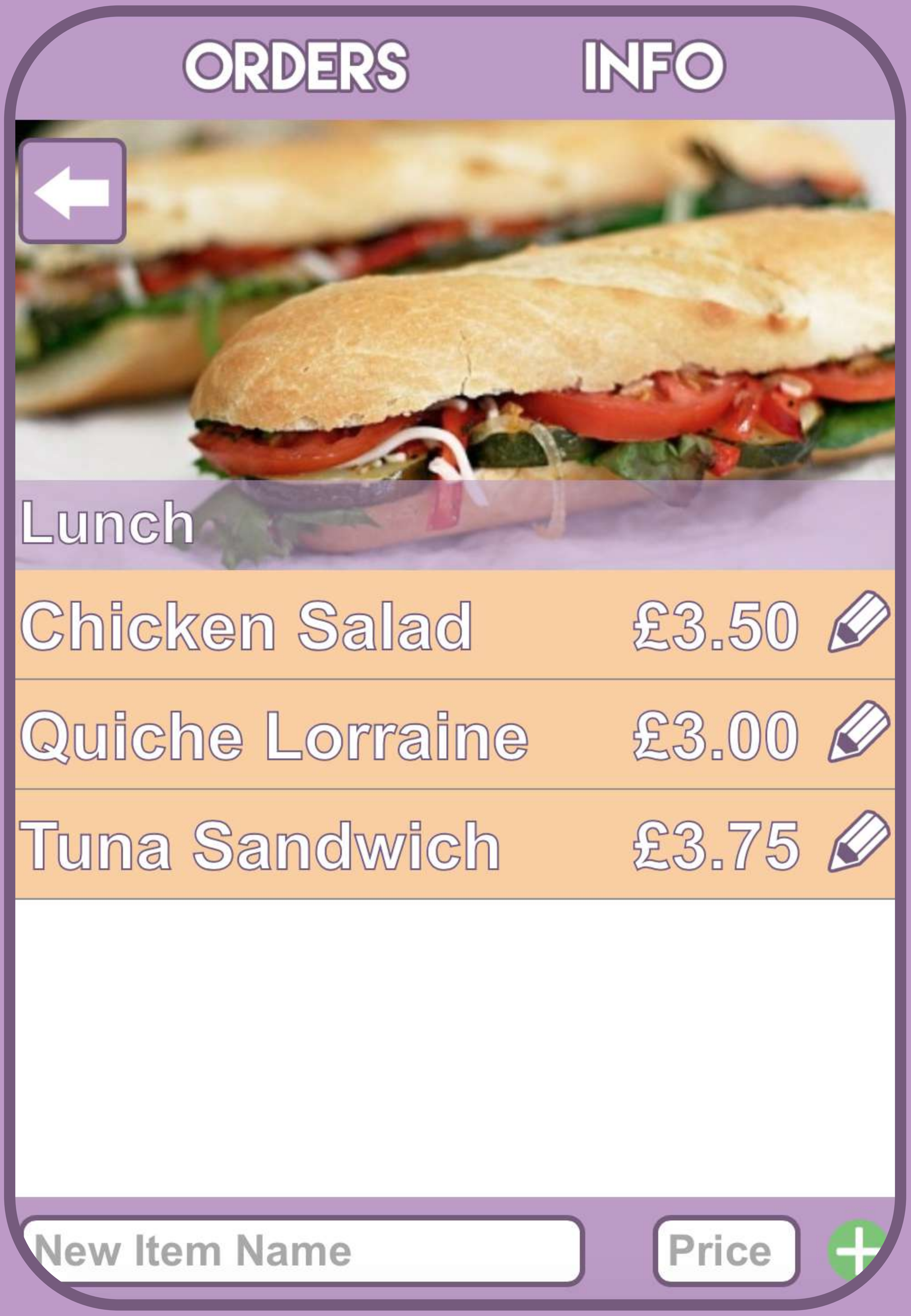
I added a menu item functionality to the retailer application, I tried a new way of creating the javascript function this time, when the retailer pressed on one of the categories to get to the category page to add items to that category a global variable which contains the current category is set, this means that a global variable holds the value of what type of data I need to pull from firebase. Using this global variable method made it significantly easier to make sure I'm pulling in the right data without setting everything and then unsetting the non-relevant info.

When the retailer views the page they see it like to consumer would but have an edit icon next to each menu item on the right hand side which budges over the price. There is also a short form at the bottom of the category page to add additional menu items. I then went about implementing the same structure in the consumer application, I do it in a similar way to setting up the categories so it was easier than expected. I still need to make the edit function actually work and I want to add a child_added listener so I can add the menu item to the page live as the retailer creates a new one currently you can't see it until the list is refreshed.

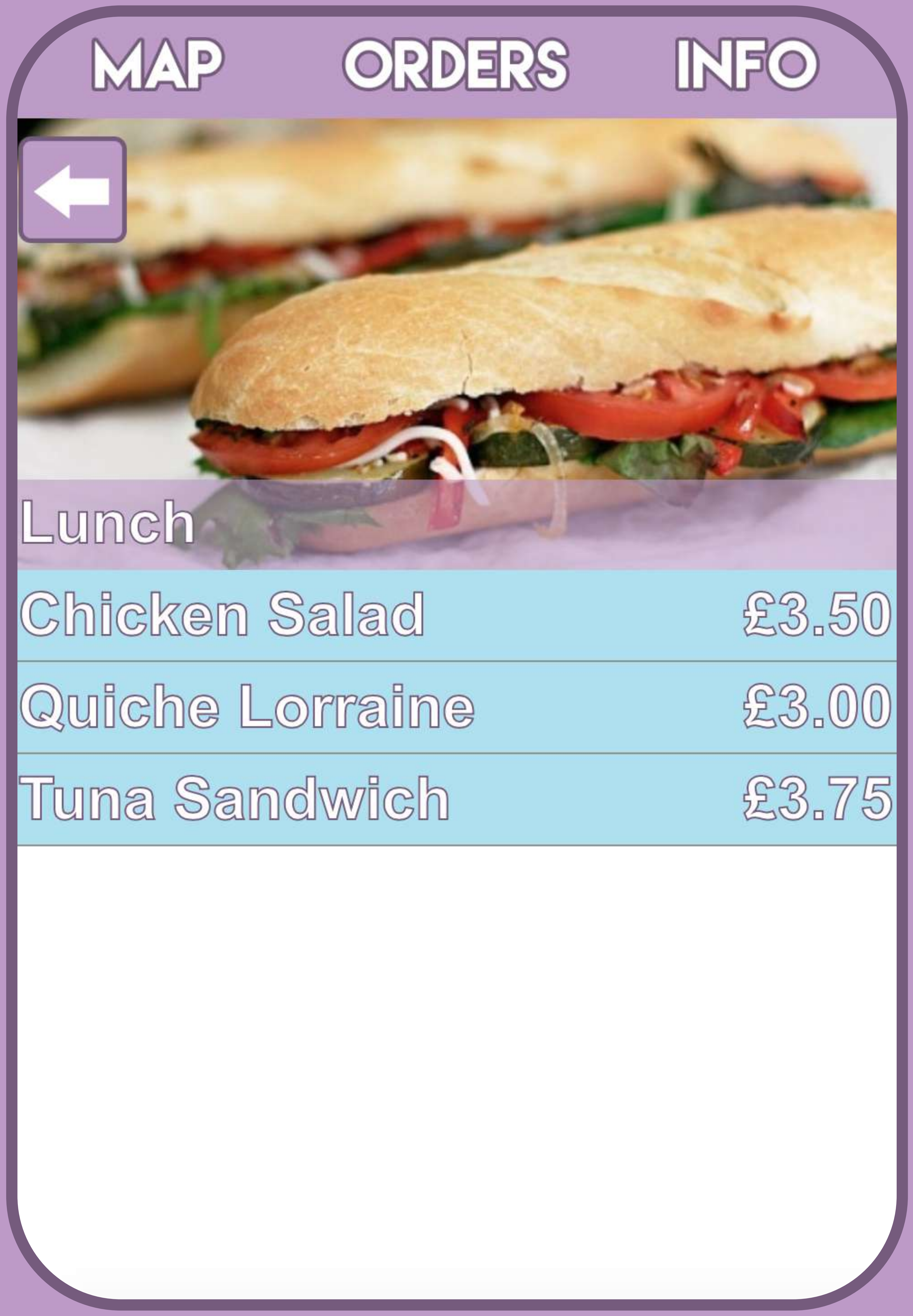
I really want to get the ordering functionality working, that is the core functionality of what I want my app to do and vital to the project so adding all of this functionality as quickly as possible without worrying to much about refinements is important. I added a few 'patches' to the existing app to fix bugs, for example there is an error with the category adder when a user first logs in and is pre authenticated which I couldn't find an obvious way to fix, to tackle this for now after logging in the application is refreshed, obviously not an ideal scenario but it's a working temporary fix so I can move on and get stuff done.

Once I complete the retailers menu editor to a level where they can fully construct their menu and make changes to it I will have the beginning of the structure I require to lead towards ordering being implemented.

Retailer App

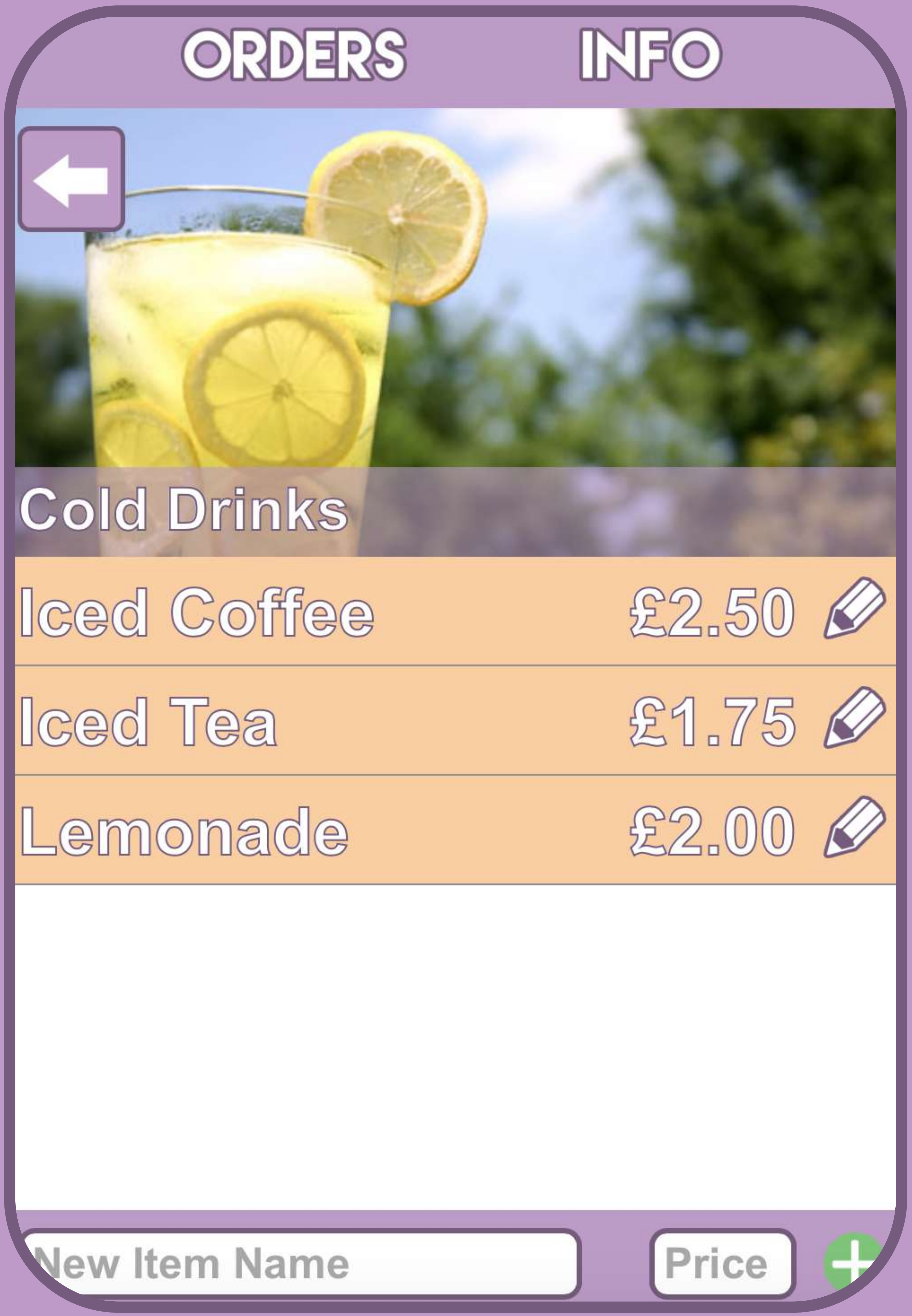


Chicken Salad	£3.50	✎
Quiche Lorraine	£3.00	✎
Tuna Sandwich	£3.75	✎



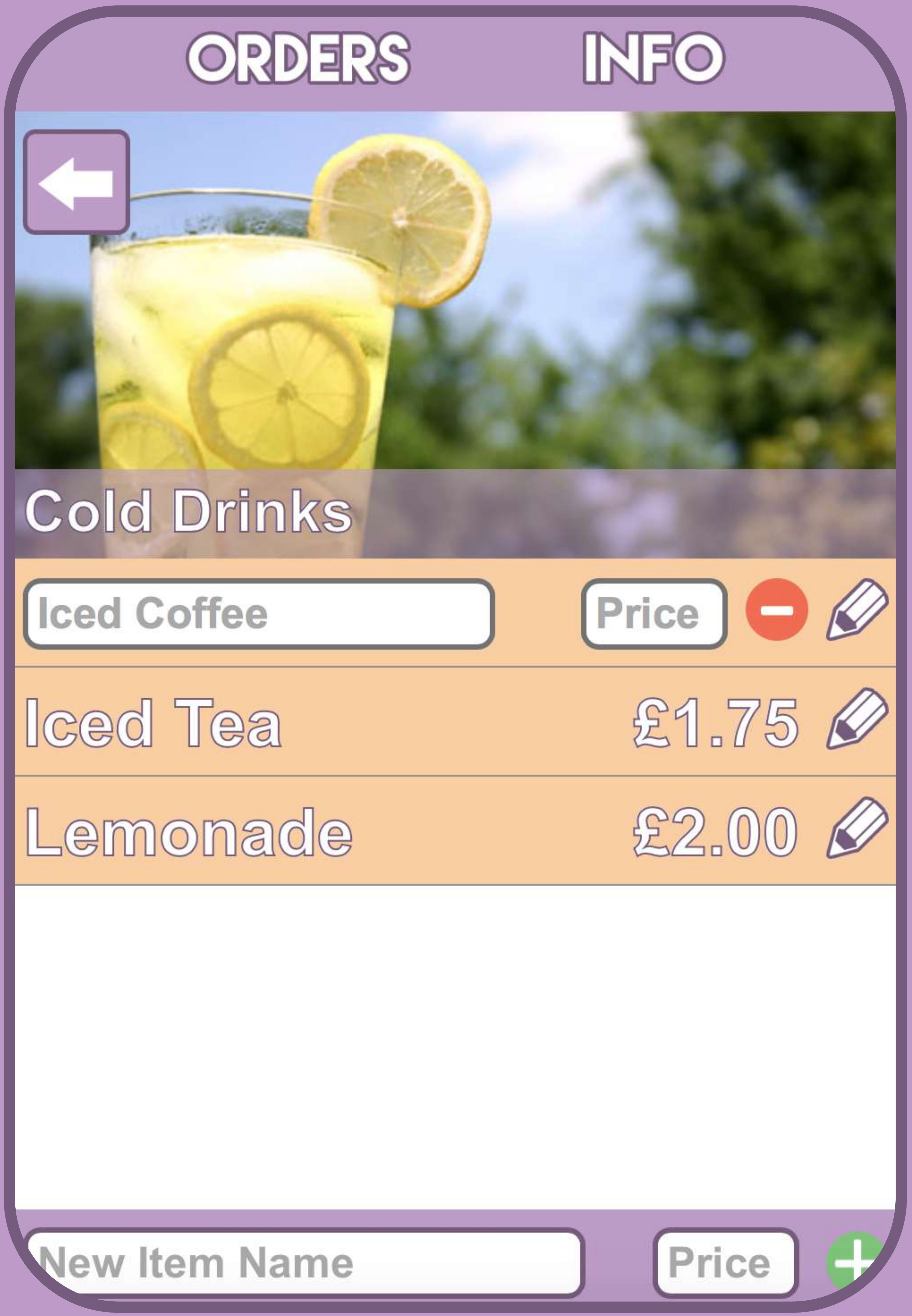
Chicken Salad	£3.50
Quiche Lorraine	£3.00
Tuna Sandwich	£3.75

Retailer App



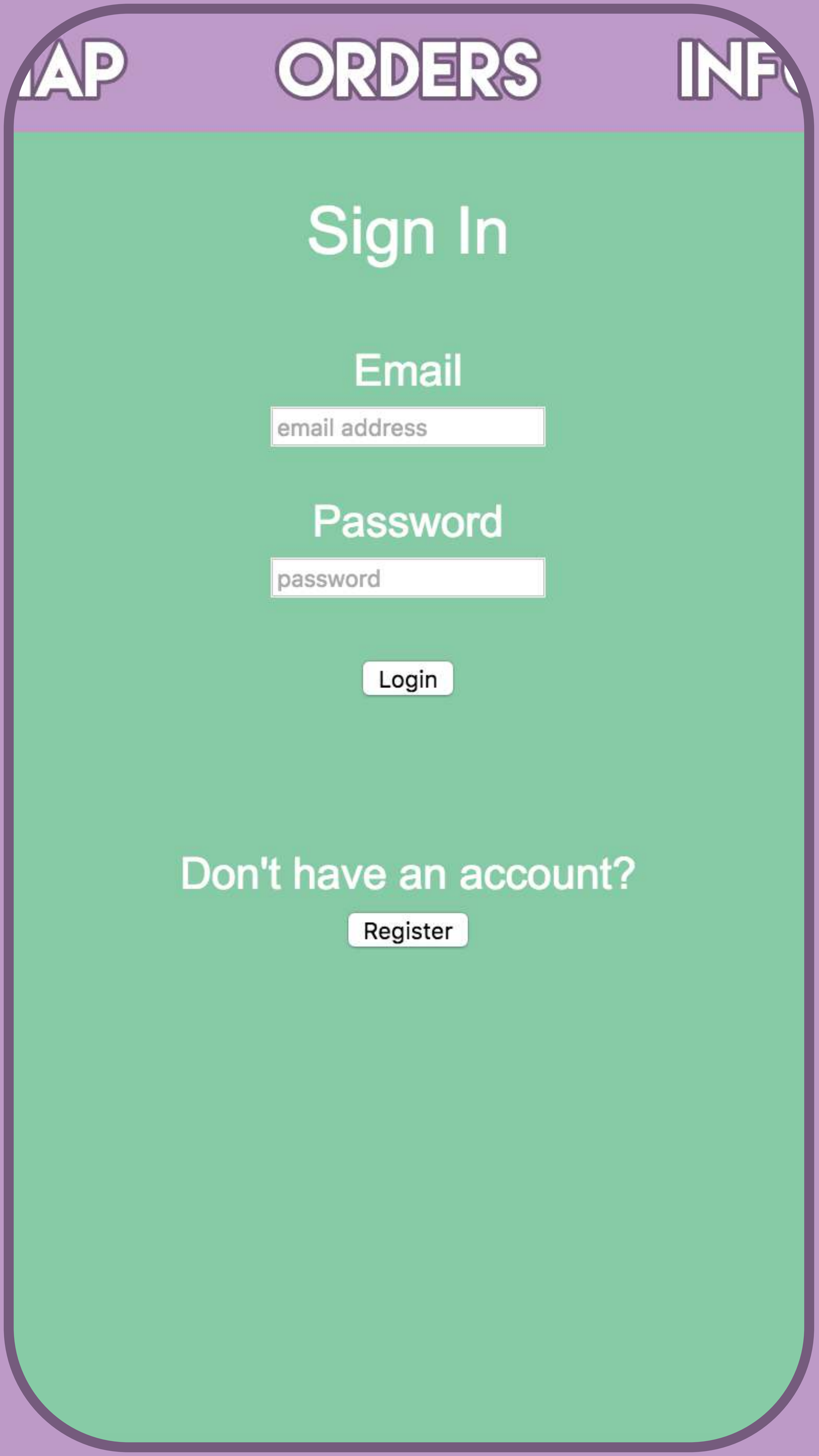
Cold Drinks

Iced Coffee	£2.50
Iced Tea	£1.75
Lemonade	£2.00



Cold Drinks

Iced Coffee	Price
Iced Tea	£1.75
Lemonade	£2.00



Sign In

Email

email address

Password

password

Don't have an account?

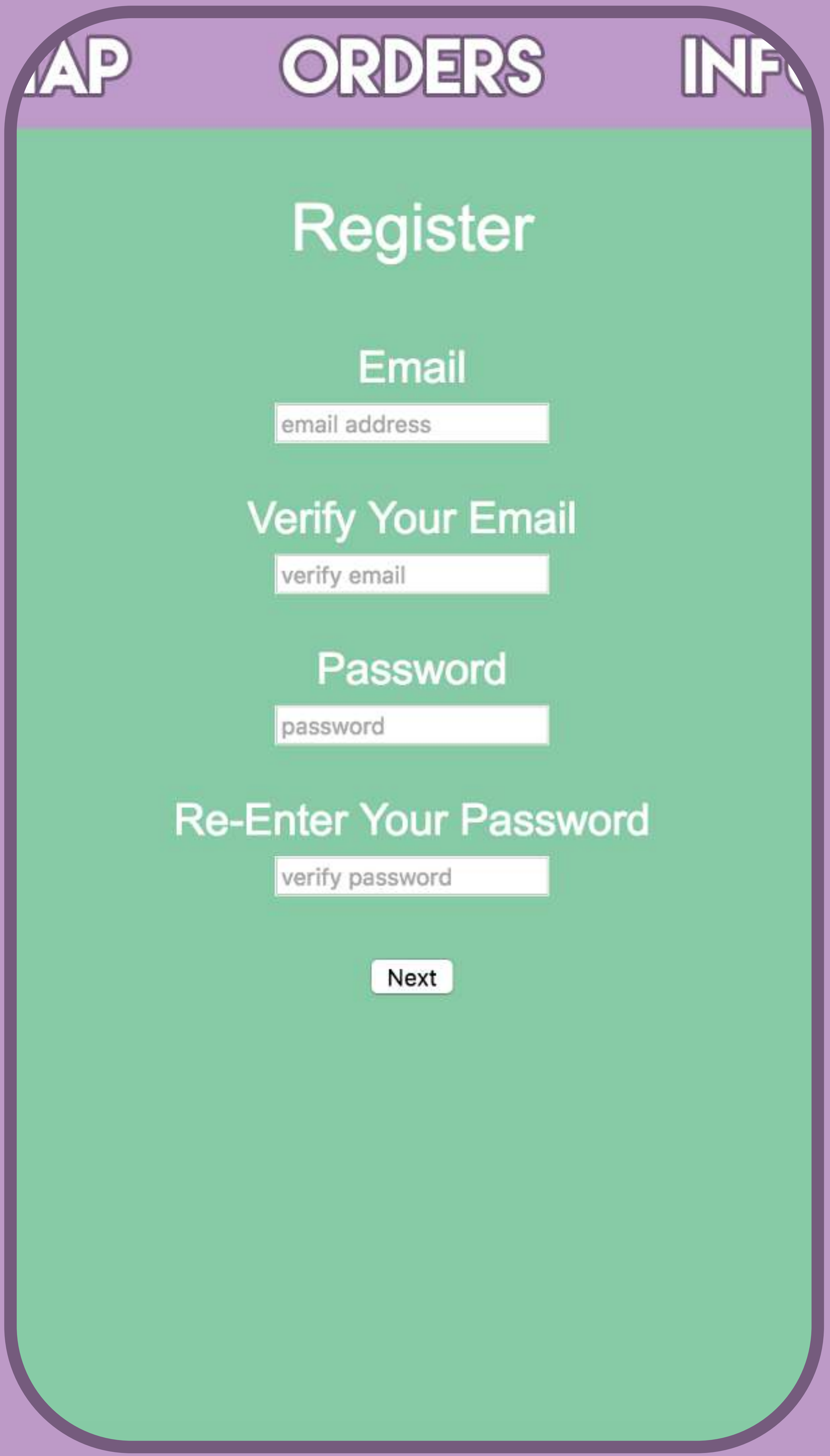
Ordering

I began working on the core functionality available to the consumer, placing an order, the first thing that I came up against when I decided to do this is that I realised I had no way of storing that order against the current user as currently there is no way for a user to log in, sign up or anything of that type, I decided that I would populate the info tab with a really basic log in and registration form so that I can get that user id, I don't intend this to look like a finished product in any shape or form however to be able to test ordering in a way which will eventually be able to be configured for use by every single user I need to create this to make it realistic otherwise I'd be creating an ordering method which I couldn't actually roll out without beginning again with it.

The form I created looks basic in terms of styling however it works in a way which would be usable later on when I come back to adding more detail to the info tab, the user is first presented with a basic log in form, if they don't have a user account already they can press the register button below.

Once the user clicks the register button they're presented with a different form asking them for both their email and a password and to verify each of these, this is to ensure that the user has entered them correctly, if they don't match the user will received prompt text telling them what they have done incorrectly and how they can go about continuing.

Ordering



Register

Email

email address

Verify Your Email

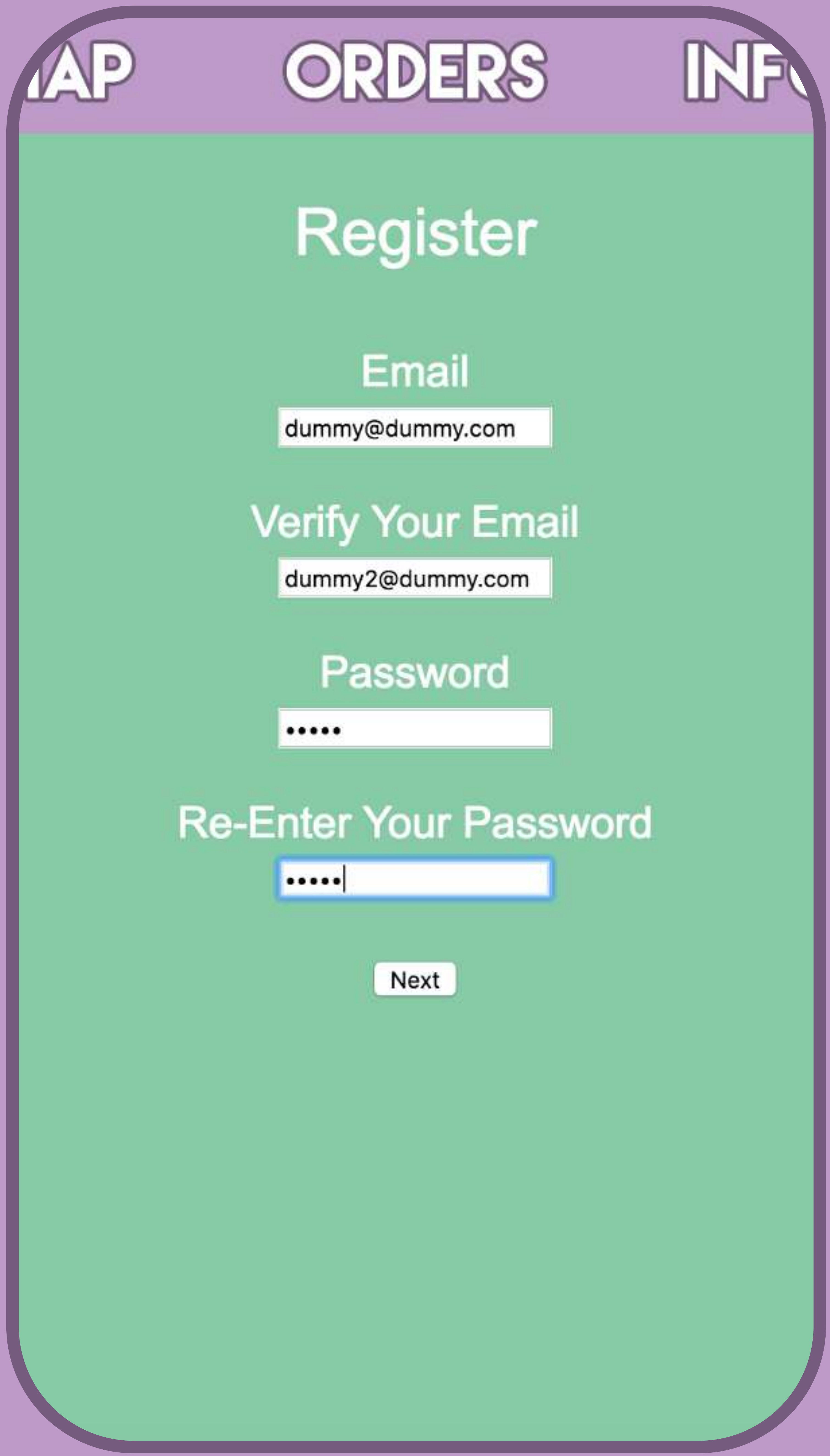
verify email

Password

password

Re-Enter Your Password

verify password



Register

Email

dummy@dummy.com

Verify Your Email

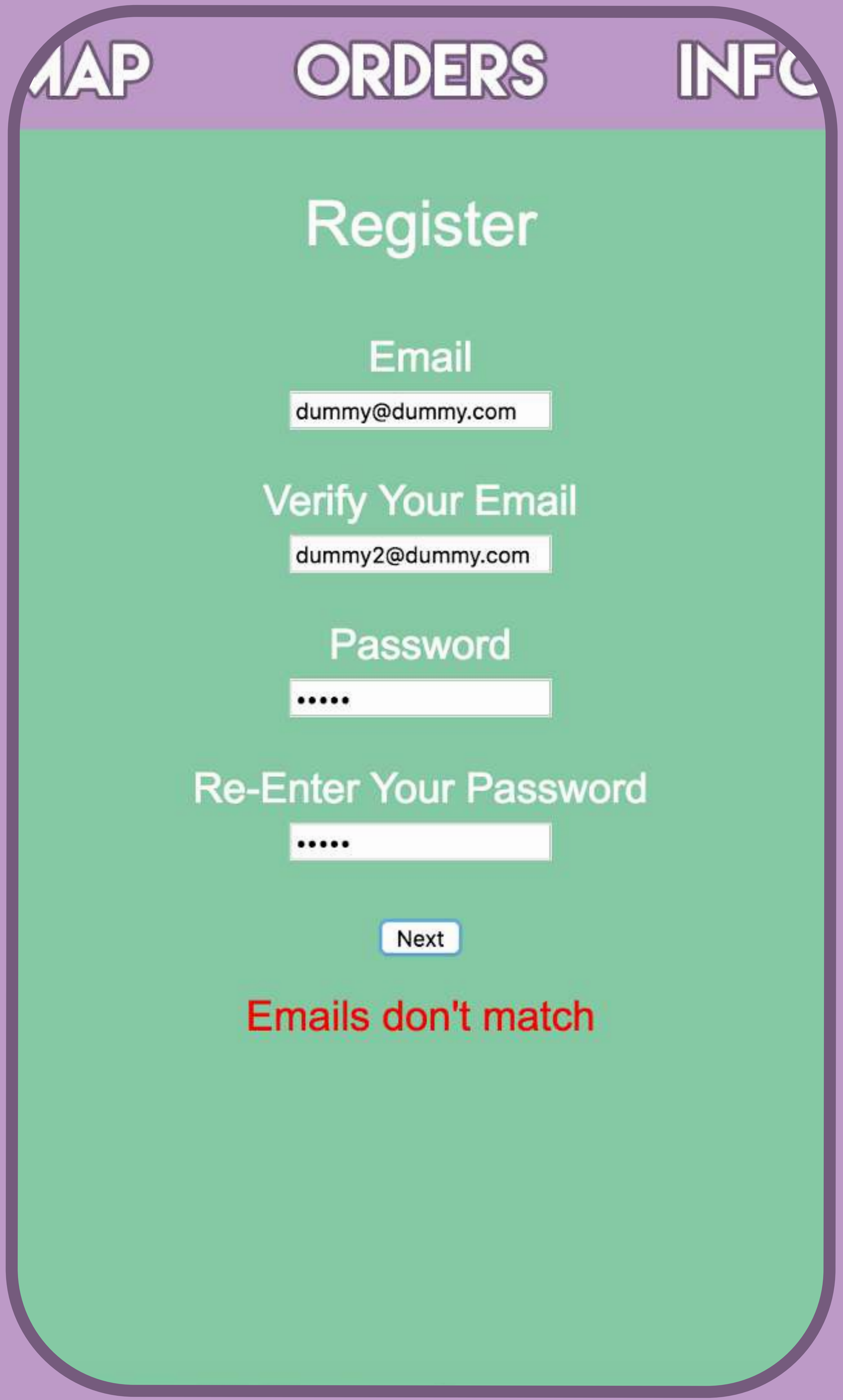
dummy2@dummy.com

Password

•••••

Re-Enter Your Password

•••••



Register

Email

dummy@dummy.com

Verify Your Email

dummy2@dummy.com

Password

•••••

Re-Enter Your Password

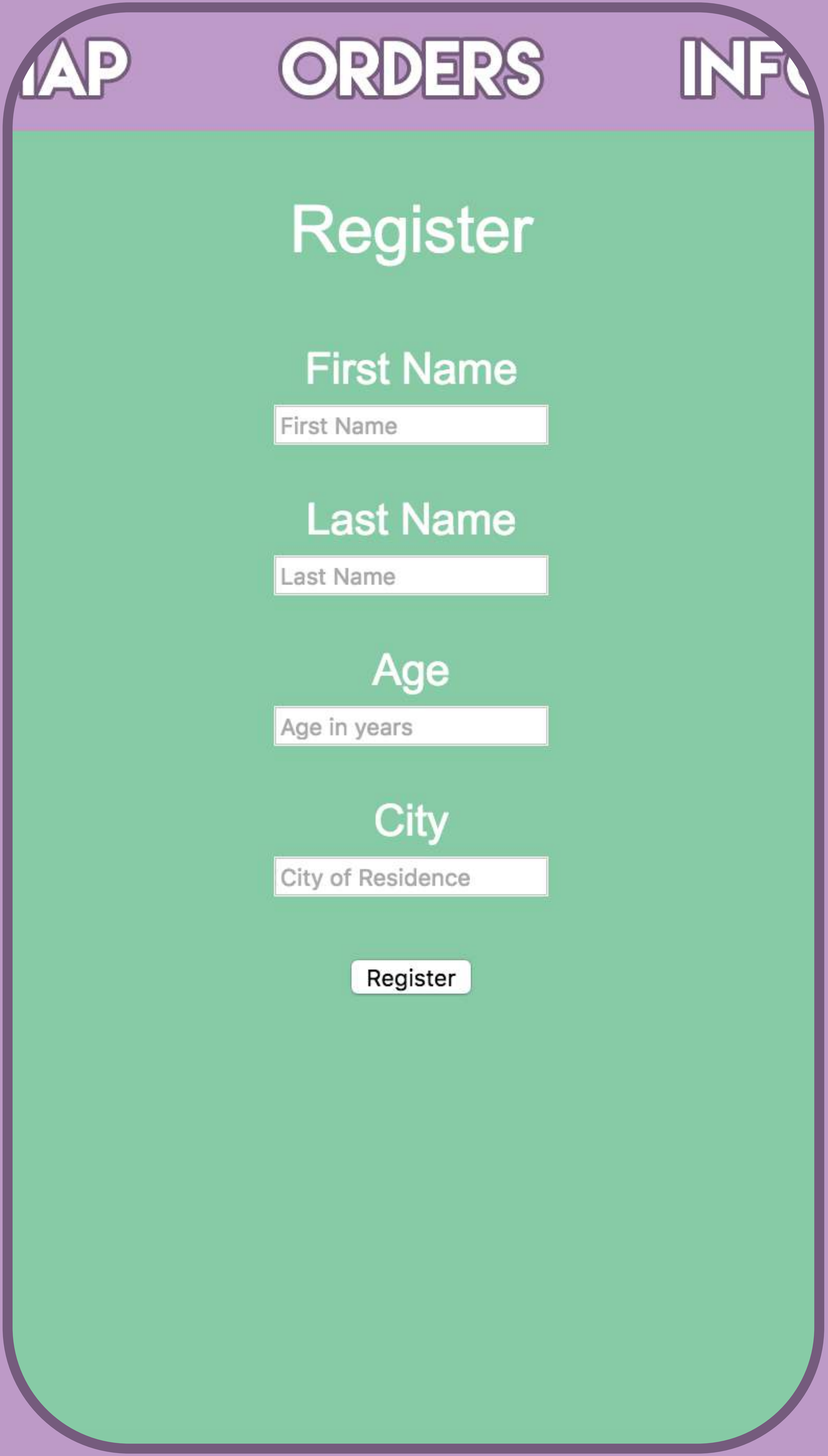
•••••

Emails don't match

Ordering

Once the user completes this part of the form sufficiently the next part of the form is shown, although it looks like at this point by pressing the next button the user is submitting the form in actual fact it just displays the second part of the form as long as both emails and passwords match so that all the data can be submitted as one, the form then asks for some additional information which at the moment is a not-thought-through selection of options as I just wanted the data to test submitting it to the users data not to store the actual information I'm going to eventually need as I don't want to spend too much time on it without sufficient user testing.

Upon form completion the page then changes to simply display 'user logged in', this is also the case if the user visits the page but is still authenticated from the previous session or if they just go through the log in form as they already have an account, this is to give me an indicator that this account is authenticated and ready to place an order.



Register

First Name

First Name

Last Name

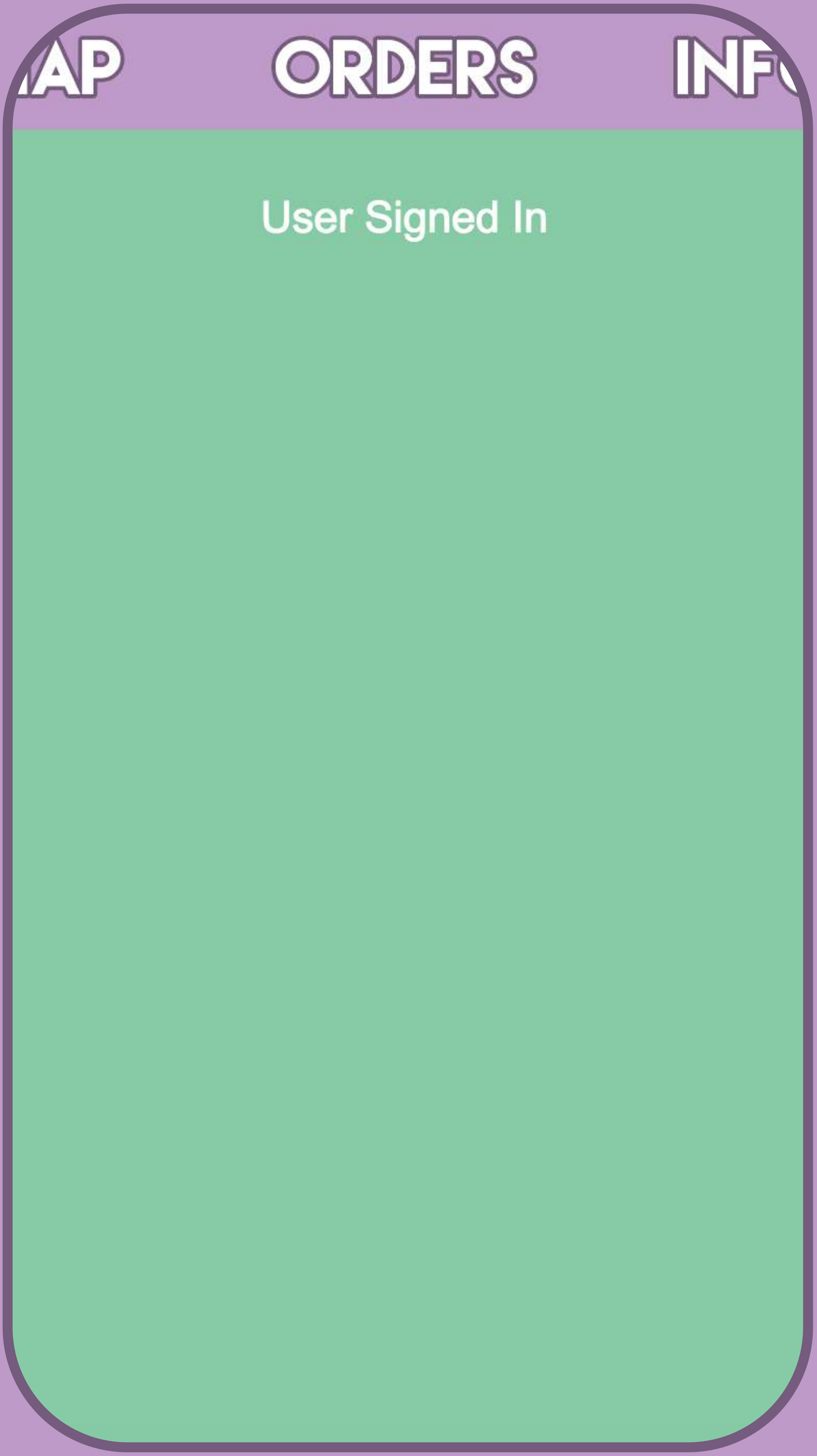
Last Name

Age

Age in years

City

City of Residence



User Signed In

Ordering

The way this form submits is similar to how the retailer accounts work, the email and password are set not to my firebase database as that is insecure but using the email authentication service available through firebase, they are also assigned a unique user id and I created a new database within my firebase which stores the user IDs with the users additional information in it like age, location etc this means that once a user logs in securely with their email and password which I never see and is handled completely by firebase I can make a link between that account and the users additional info just by using the unique user ID as that matches both.

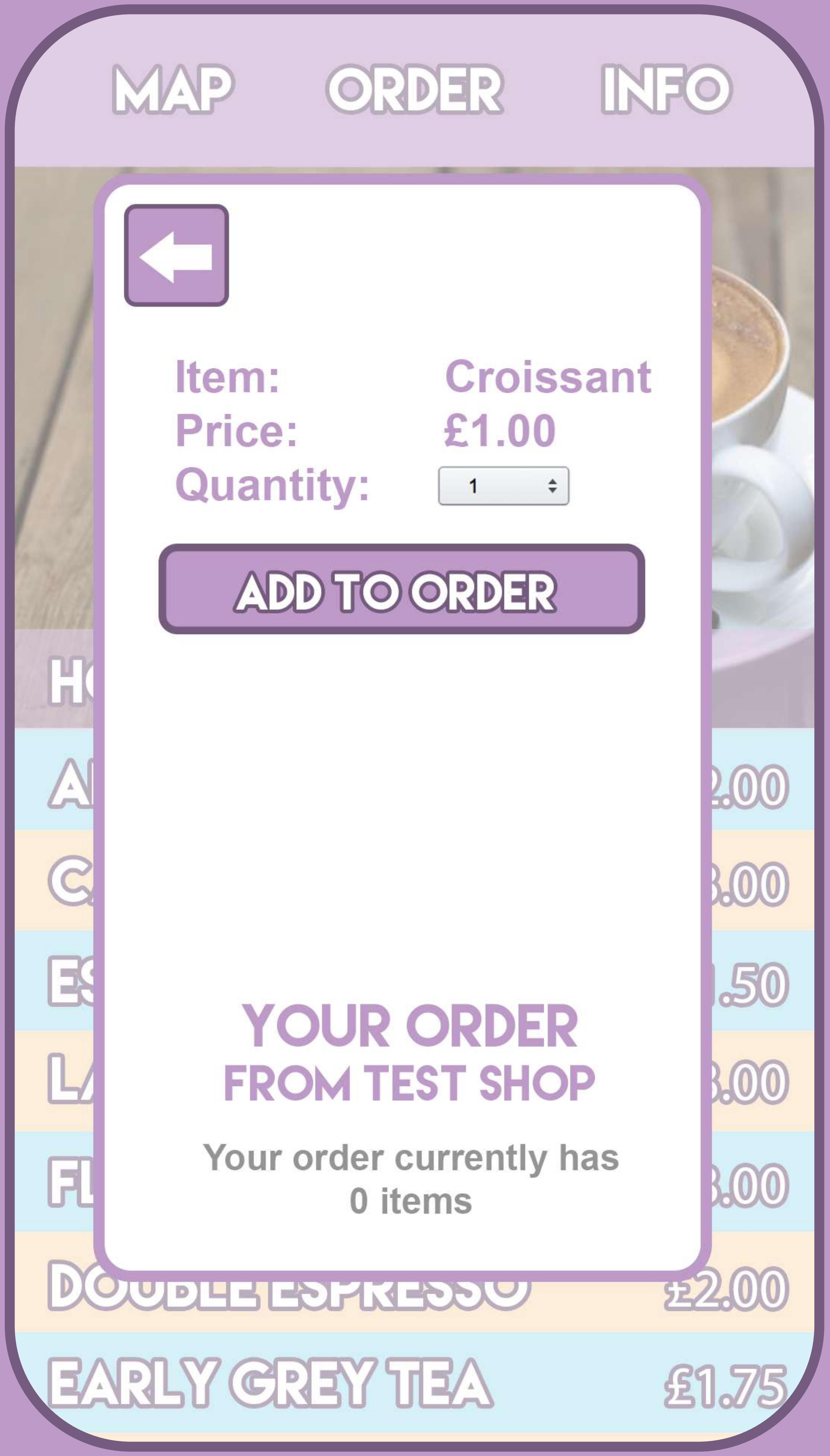
Next i will move on to allowing this user to place an order by them constructing an order in their firebase entry ready to be submitted.

Now that the user has an account I could begin to work out how they're going to go about placing orders, I created a basic wireframe of how I wanted this system to work so I knew what I was going to be working towards.

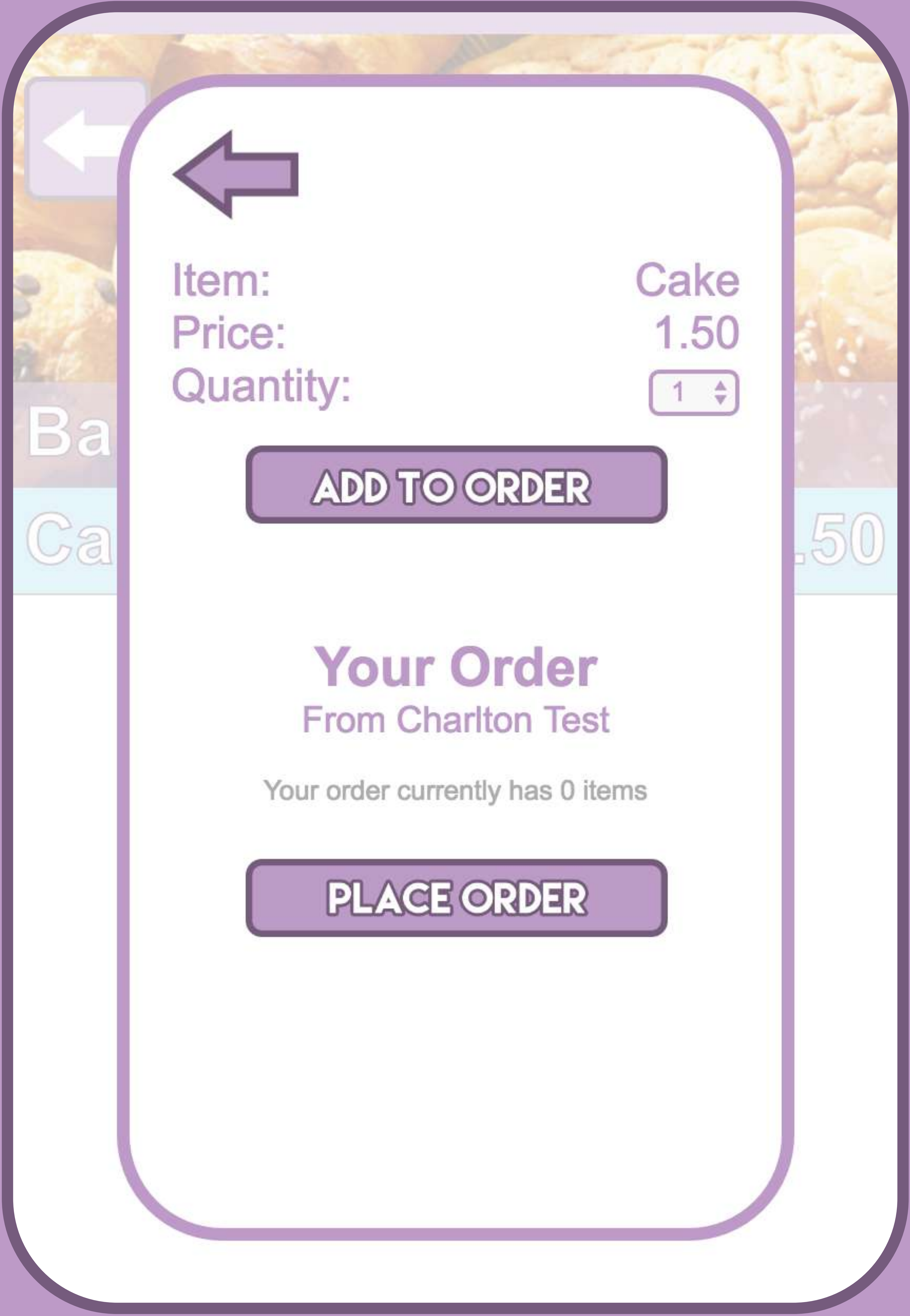
Ordering

This is what I want to happen once a user selects an item from the menu list, at the top it displays the item they have just pressed, how much it costs and then they have the option to choose how many they would like to order, below if the user currently has a partially built order I want it to be displayed so they can see how their order is building or if they have nothing to inform them of that. once they have built an order I want them to be able to hit a place order button at the very bottom of that box to submit it to the retailer. Additionally once a user has a current order in place I'm going to have a current order button appear at the bottom of the screen much like the purple add item to order bar at the bottom of the retailer app so that they can jump to their order once they have one active to place it.

Once the user clicks the item in the menu the javascript displays a the purple bordered box, a transparent overlay and a back button no matter which item is pressed, then the javascript writes to a template of the contents area inside that purple bordered box, it checks which item the user has pressed and then creates a form for it, the form visually only consists of the ability to change the amount of items the user is ordering, then the javascript checks the users unique user id's entry in firebase, it checks whether under their 'orders' they have a 'current order' if so it reads this current order and displays it below in the your order area and displays which shop it's from, if there is no current order it simply displays that the user currently has no items in their order from that shop.



Retailer App



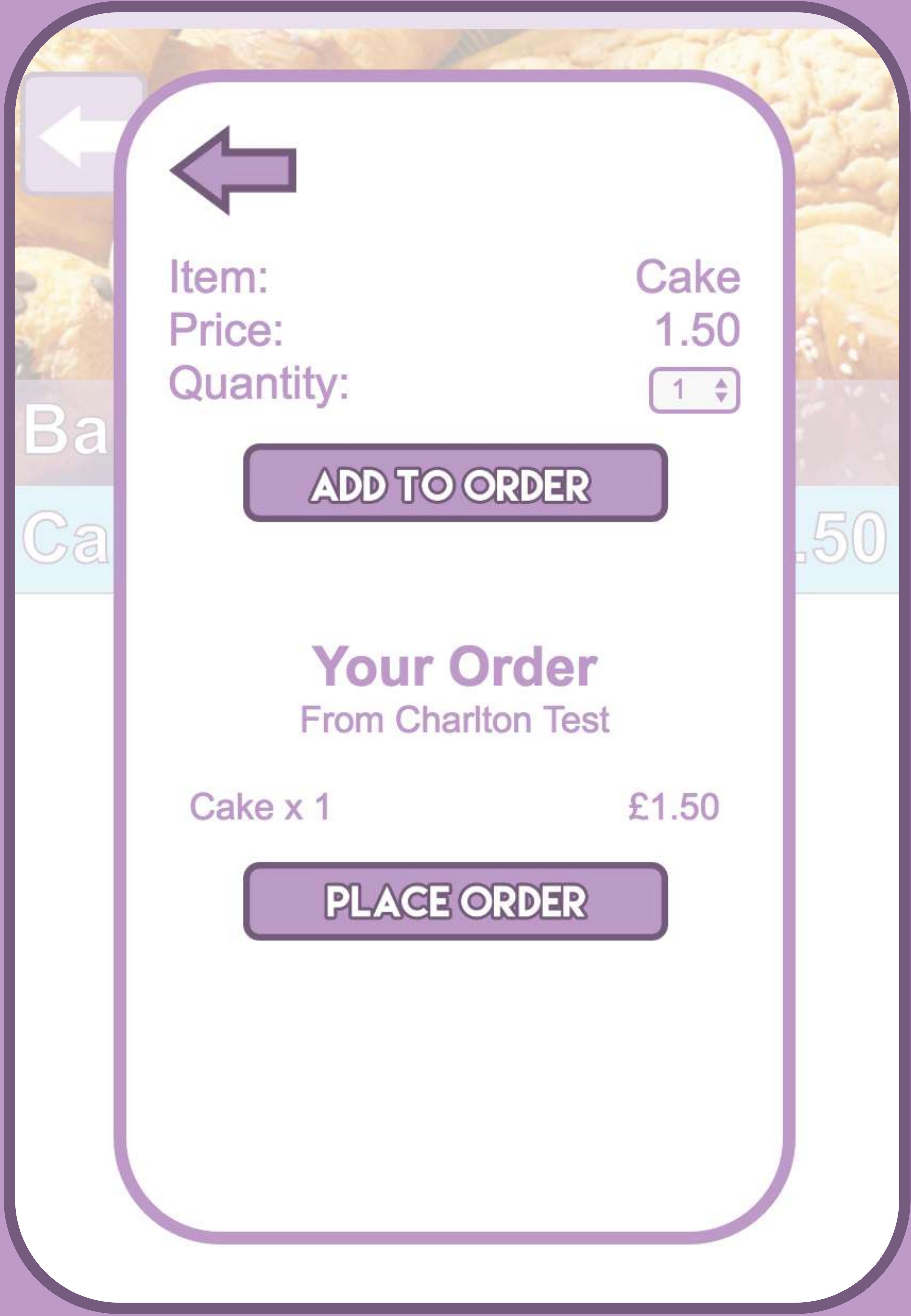
Item: Cake
Price: 1.50
Quantity: 1

ADD TO ORDER

Your Order
From Charlton Test

Your order currently has 0 items

PLACE ORDER



Item: Cake
Price: 1.50
Quantity: 1

ADD TO ORDER

Your Order
From Charlton Test

Cake x 1 £1.50

PLACE ORDER

Firebase Restructuring

In order to make my app work smoothly I’ve had to take a few steps back and change the way in which some information is stored, I stored all of the shops initially under unique IDs, then I changed this to user IDs to be able to link these to user accounts, with a lot of information that I’ve been storing such as shop menu items, users current orders, I’ve been using IDs based on what the users entered such as ‘Cake’ this makes it more difficult for me to treat them as an object with a unique ID as it may not be unique and doesn’t allow for editing of the name as that is also the ID so if the user changes the name, it would change the ID creating duplicates, essentially they way I did it worked temporarily but it was a matter of time before bad form caught up with me.

I began working with the retailer application and blanked off any menu items I had held on my test shop, I then changed the way in which the form submits the information, instead of ‘setting’ the information at a path which I define by pulling what the user has entered in I ‘push’ the information as an object with a random unique ID generated by firebase as an object which contains what the user has entered as attributes instead of:

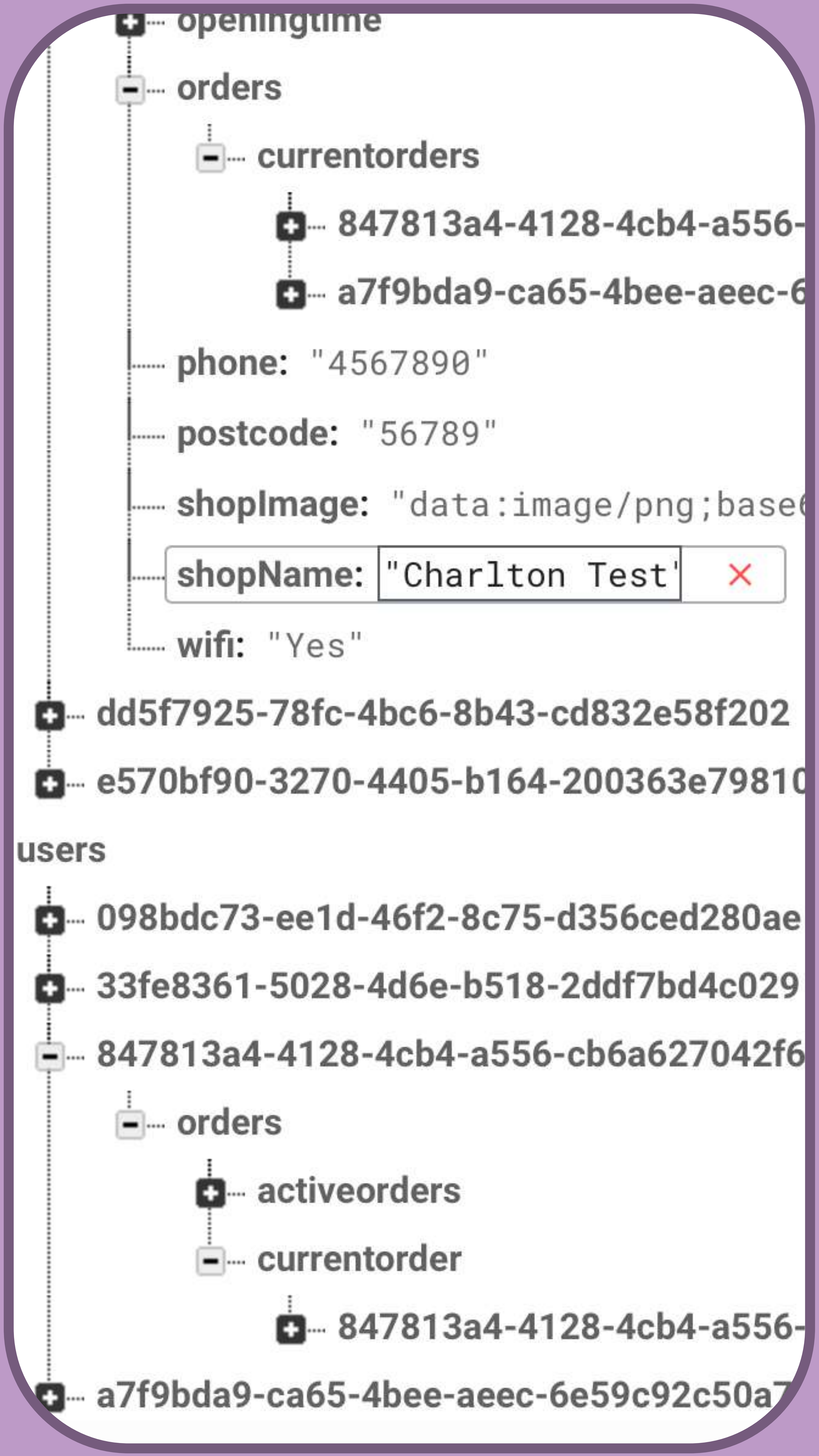
Cake { Price: 2.00 }

I do:

HJFA-8741 { Name: Cake, Price: 2.00 }

I also had to change the way the retailer app deletes and edits these items. This allows me to do grunt work using the user ID behind the scenes without the user seeing to pull the relevant info in and give me better structure and allows me scope for making changes in future. After changing the way the retailer menu editing works i had to then change the way the consumer app displays this information as it now thinks these unique IDs are the names of the items which is incorrect, so I had to change all the javascript for the way that is set. Then I changed the way a current order is placed, initially I was going to create all orders whether current or not as the same with an attribute of whether the order is current or not but then I realised that would mean the app would have to check all of the orders to find out which of the orders is current.

Firebase Restructuring



```
  openingtime
  orders
     currentorders
        847813a4-4128-4cb4-a556-
        a7f9bda9-ca65-4bee-aeec-6
  phone: "4567890"
  postcode: "56789"
  shopImage: "data:image/png;base6
  shopName: "Charlton Test'  ✕
  wifi: "Yes"
dd5f7925-78fc-4bc6-8b43-cd832e58f202
e570bf90-3270-4405-b164-200363e79810
users
dd 098bdc73-ee1d-46f2-8c75-d356ced280ae
```

```
users
098bdc73-ee1d-46f2-8c75-d356ced280ae
33fe8361-5028-4d6e-b518-2ddf7bd4c029
847813a4-4128-4cb4-a556-cb6a627042f6
  orders
     activeorders
     currentorder
        847813a4-4128-4cb4-a556-
a7f9bda9-ca65-4bee-aeec-6e59c92c50a7
```

I decided to retain a structure where the user has a current order attribute and an orders attribute, when the order is in the current orders attribute each item in the order is stored as an object under a unique ID and within that it stores the item name, item ID as its stored in the shops firebase entry and the quantity the user has chosen, once the user places this order and it is in the past all of the items get packaged within another unique ID and moved over to the orders object and stored there. This is still not an ideal way of doing this as it doesn't account for a user having multiple current orders from different retailers so I need to account for that as well.

The app now saves multiple current orders in current orders, each one is under the shops unique user ID so depending on which shop you choose it knows which current order is to be pulled in, once this is submitted then the order will be moved to past past orders and attributes changed.

Ordering

Now that I have changed the structure of the firebase appropriately I can now begin working on the ordering process, whilst a user can now place an order the retailer app in no way currently acknowledges this submission.

The first part was to work on the consumer app to make the place order button actually do something, I had to allow access to the user at a specific path within the retailers firebase entry, I did this by allowing the to write only to a path within the retailer app inside orders and then inside their own unique user Id so they aren't able to change the order of anybody else. Now that they have permissions to write I made it so when the user presses the submit order button it creates a copy of their current order in the shops firebase entry, it does this inside a path orders>current orders> their user Id. The way in which the order is written to firebase uses their push method which means that the order is saved under a unique Id within their Id to allow for multiple current orders from 1 user in the shop, now that the order has been placed by the consumer it should no longer be in their current orders areas within their firebase entry.

I created a function that performs the following: once the user order has been submitted to the shops firebase and successfully written the app then reads back exactly what it has written in order to find out the unique Id that firebase has saved it under, once it has this information it creates a copy of it and writes it to a new area in users' firebase entry under orders>active orders and then removes the entry from current orders.

The reason I did it this way is that it gives the order a unique Id which is mirrored in both the retailer and consumer data so later when the retailer makes changes to the order or the user wants to do something with that order the same Id can change data in either the shops firebase entry or the users firebase entry.

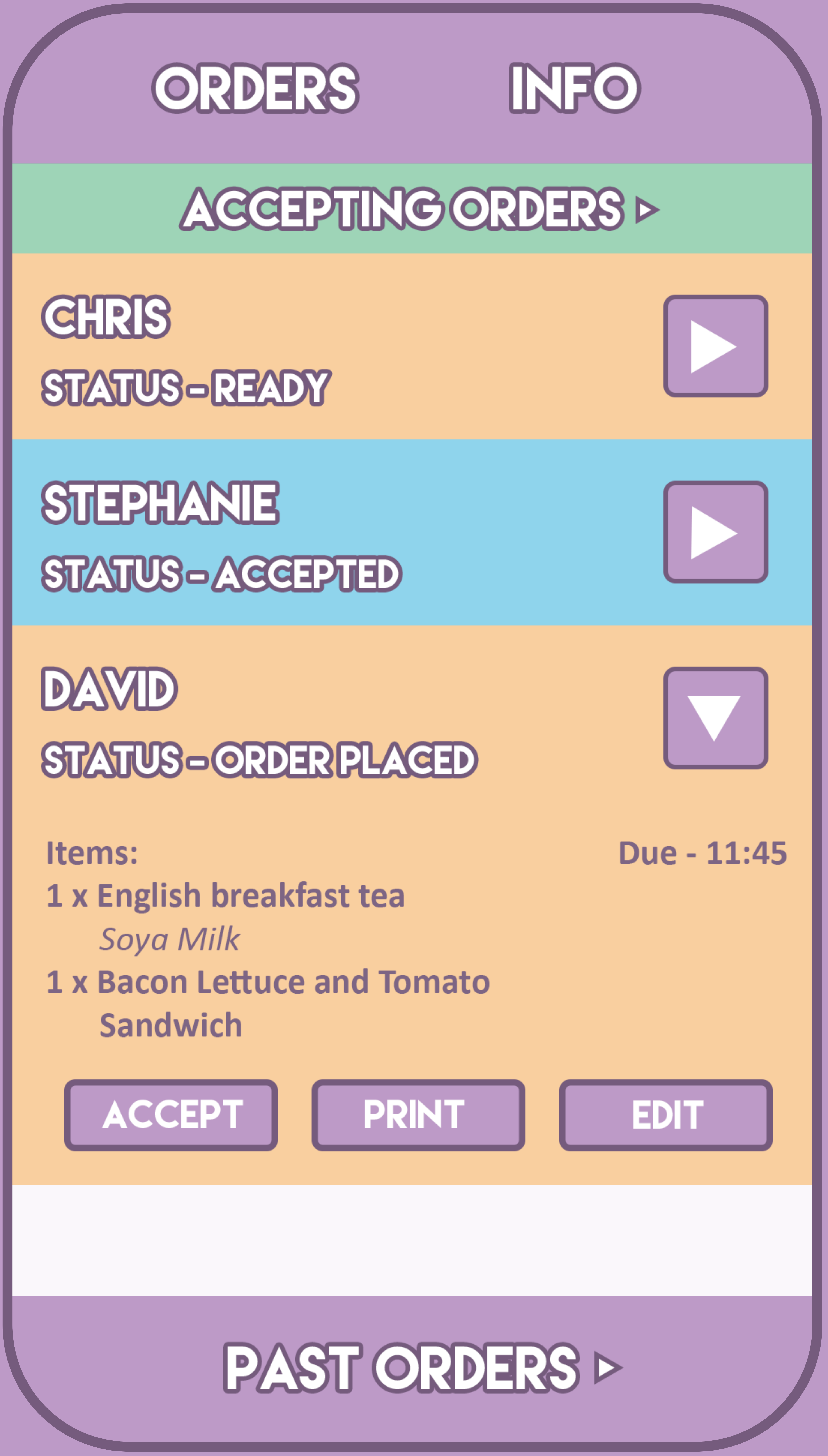
Ordering

Now that the consumer application has written the data I need to work on getting the retailer app to access and display this data, I referred back to the wireframes I created for the retailer application and looked at the orders page.

I created some assets from the photoshop file such as the accept, print and edit buttons and began to work on the static HTML, this is a method i've used throughout creating this application I first create a static template of an entry as I want it to be shown so i can get the HTML and CSS ready for when it is being written to by the javascript, whilst working on this I changed some elements of my design as I didn't like the way the order displayed in the details below the order and eventually came up with a template.

Using the rounded box makes a kind of separation of the actual order that I like also the direction of the drop down arrow has changed to make more sense with design standards that users would be used to.

These design changes are shown on the next page.



Items:

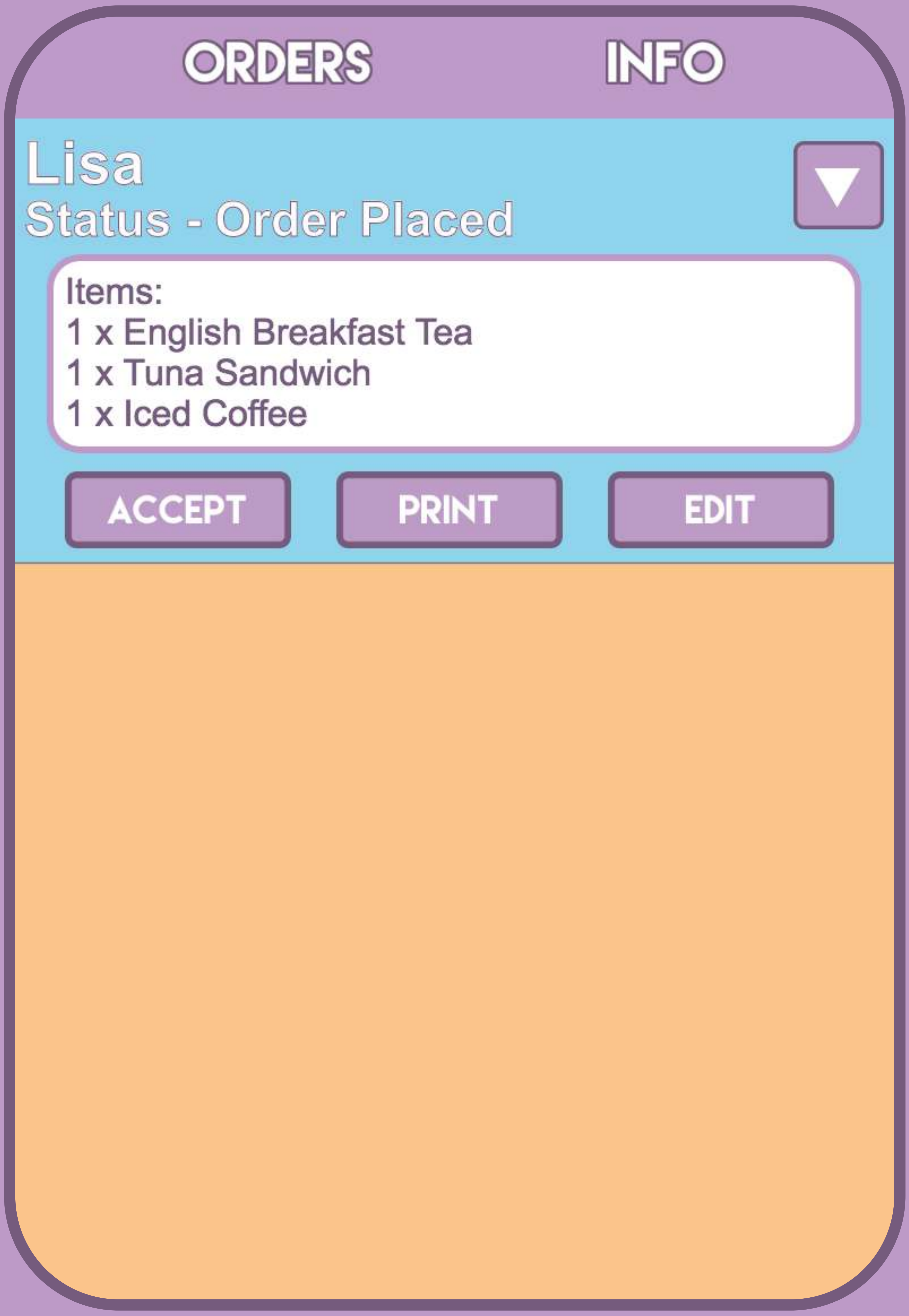
Due - 11:45

1 x English breakfast tea

Soya Milk

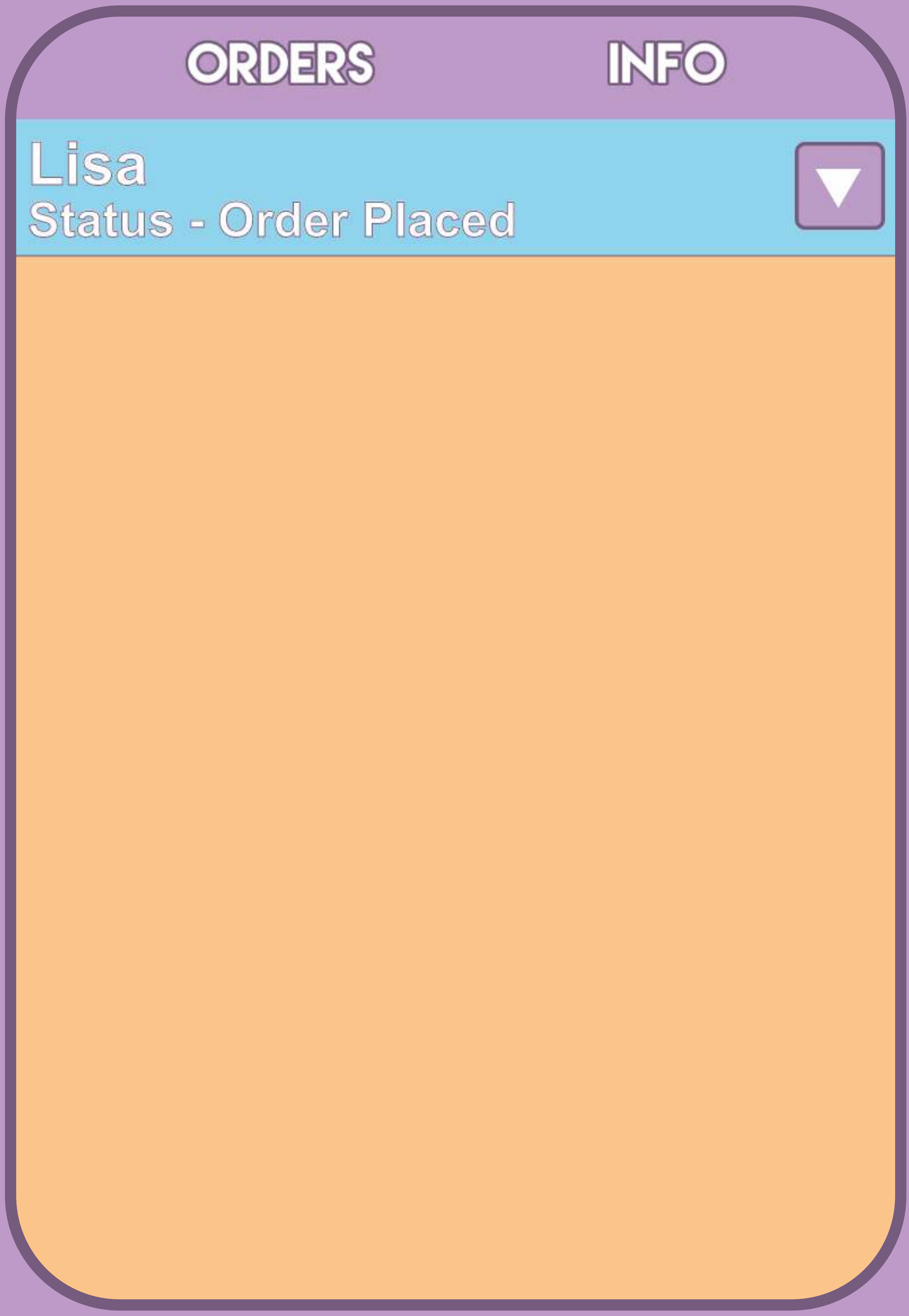
1 x Bacon Lettuce and Tomato Sandwich

Ordering

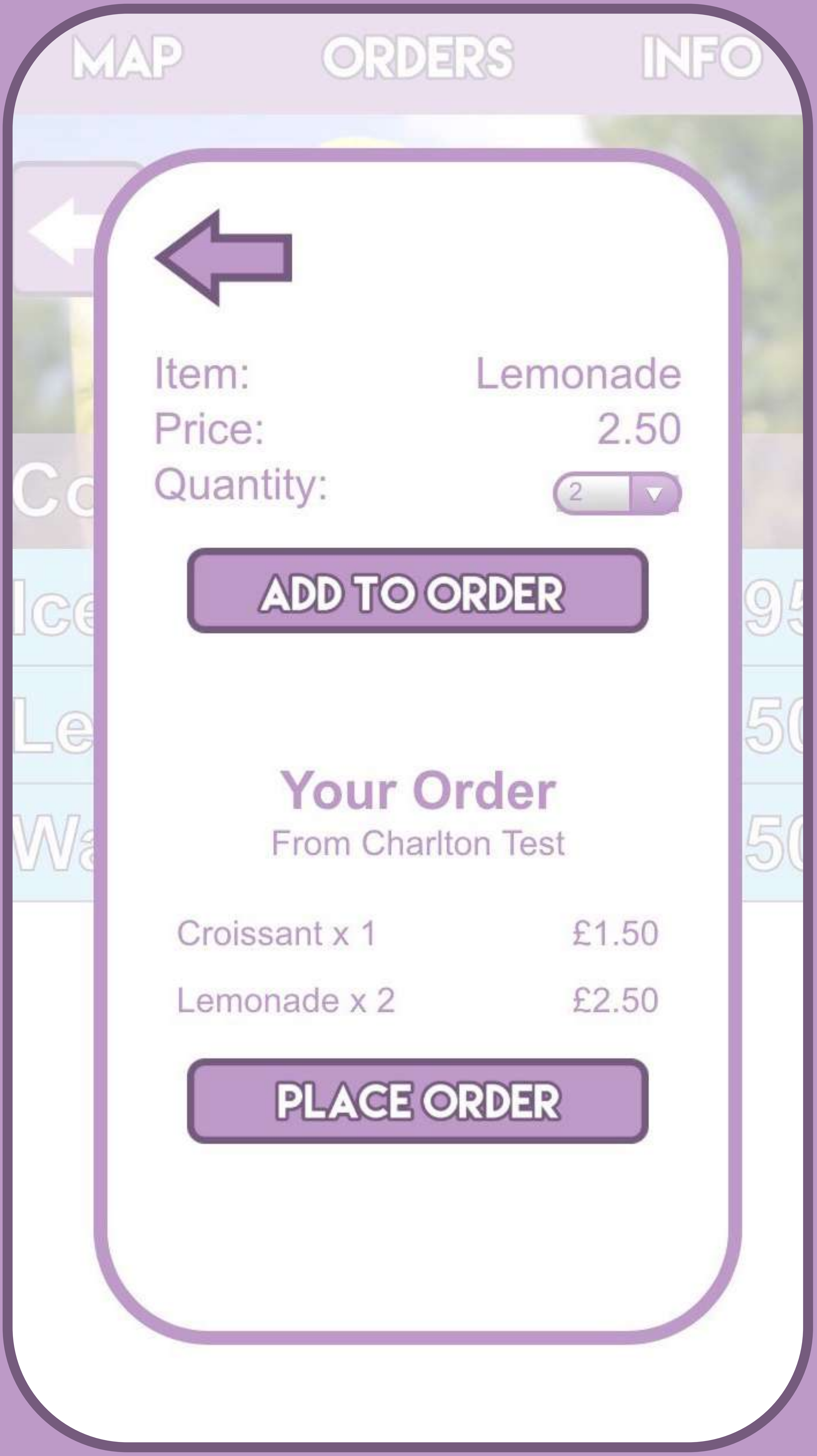


Lisa
Status - Order Placed

Items:
1 x English Breakfast Tea
1 x Tuna Sandwich
1 x Iced Coffee



Lisa
Status - Order Placed



Item: Lemonade
Price: 2.50
Quantity: 2

Your Order
From Charlton Test

Croissant x 1	£1.50
Lemonade x 2	£2.50

Ordering

Once I had the template in place I could start accessing the data in firebase and writing the orders with data actually in the firebase. I pulled in everything that I considered could be relevant, all the items currently on the order, info about the user who submitted it such as their name and used jQuery to do each functions, essentially the javascript reads the basic information about the order, how many there are and who made them and then creates an order entry div on the page, then the javascript dives further into this information and looks at each order and the items they contain, it then changes another div which is inside the div that was created further up the script and writes info about each item.

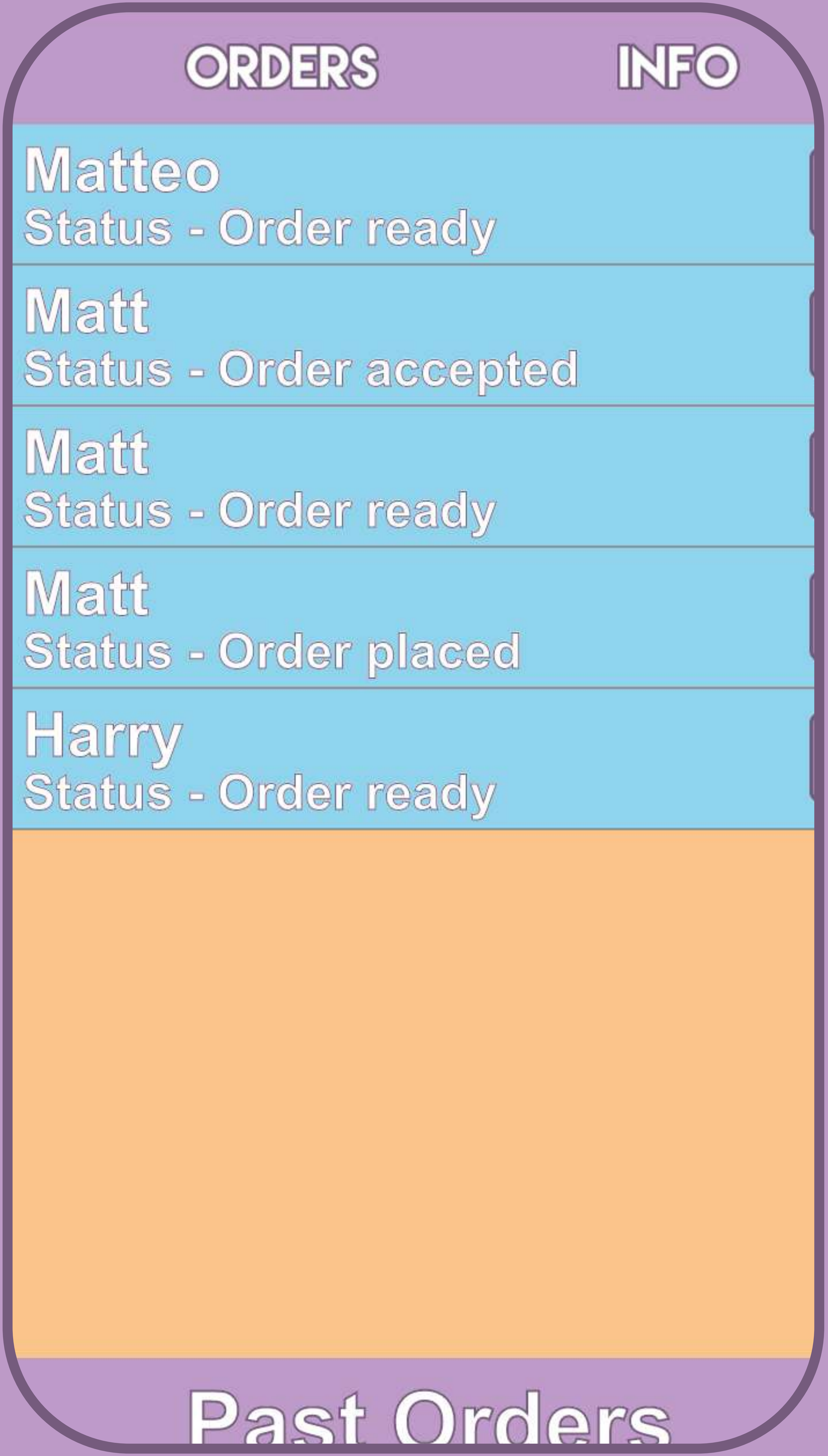
This resulted in the retailer app being able to successfully display the orders that are currently on the app but it didn't account for orders being placed live. I used a firebase function to listen to the relevant shops orders by listening for a "child_added" event, essentially a new order, then once it detects a new order to clear the orders currently in place and get them again updating the order list, this created a live system so as soon as an order is placed by the consumer it is displayed by the retailer.

Next i need to refine this process, allow users to edit their order and remove items etc as well as adding functionality for the retailer to move the order on and change the status.

Ordering

It was important for the retailer to also be able to make changes to the order placed by the consumer so they can update the status of the order, I achieved this with the ordering buttons below the order in the retailer app view and added a new element to the order in the database called 'status' which the retailer can update, they can accept the order changing the status from placed to accepted and then change it again to ready.

Once this was in place I decided that I need to work on how to show this updated status change in the consumer application. To do this i needed to refine my ordering dialog in the consumer app, currently the ordering pop up is buggy and has annoying overflows because of the way that I created the CSS for it, essentially the part at the top of the popup that appears when you select an item that you want to add to your order is a fixed size as it is 1 item, price and quantity and an add to order button, but the area below that which shows the current order could be a variety of sizes depending on how many items are in that order, to make it work properly I need to add maximum height limits to that box and make it scroll when the order is too large, so have now changed the styling, it is going to need some more work to make it work for all use cases as it is still buggy, once this part is done I can work on functions like editing the order and displaying it in the orders tab to get the consumer app to an almost complete minimum viable product state.



Matteo
Status - Order ready

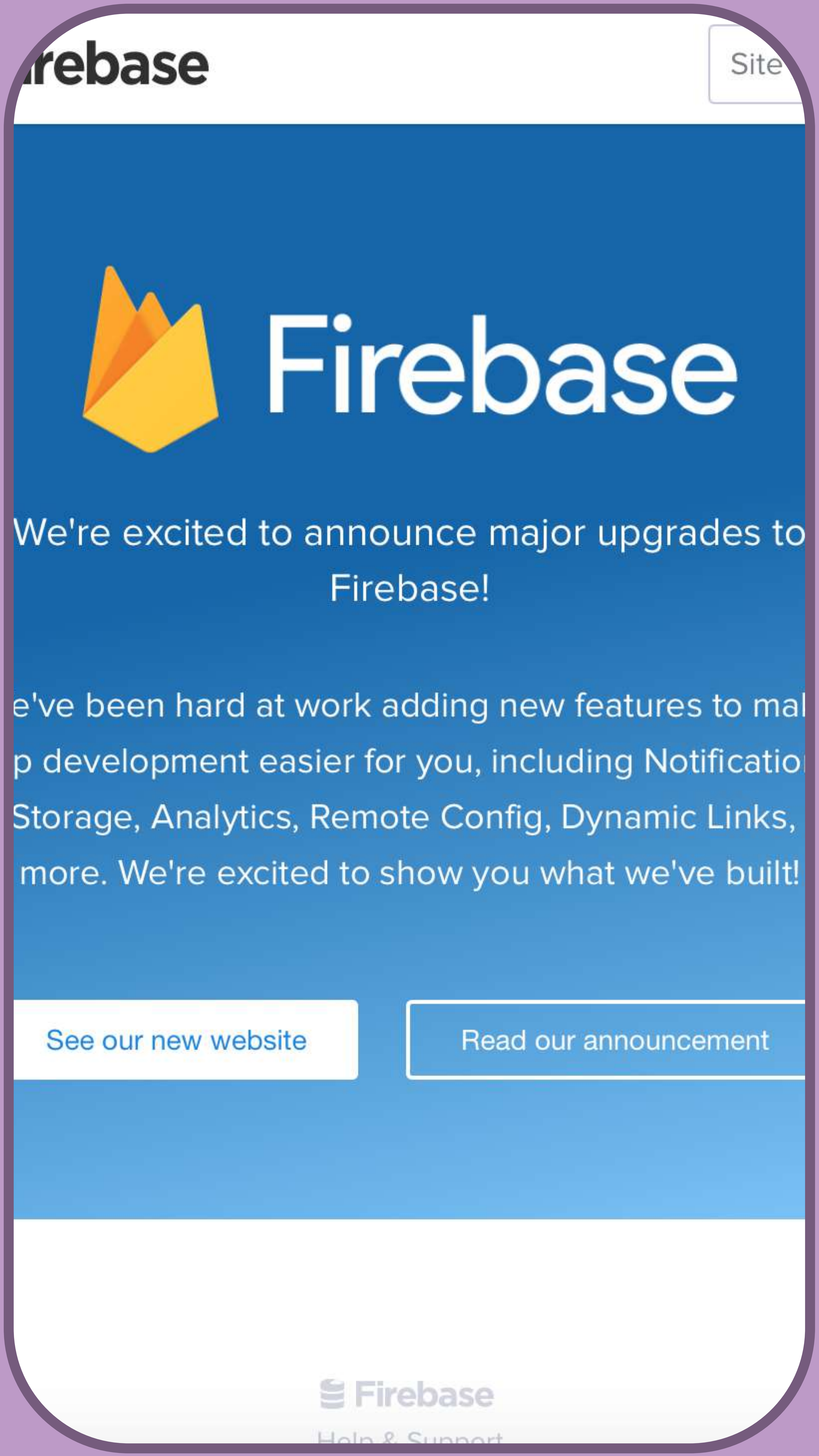
Matt
Status - Order accepted

Matt
Status - Order ready

Matt
Status - Order placed

Harry
Status - Order ready

Firebase Update



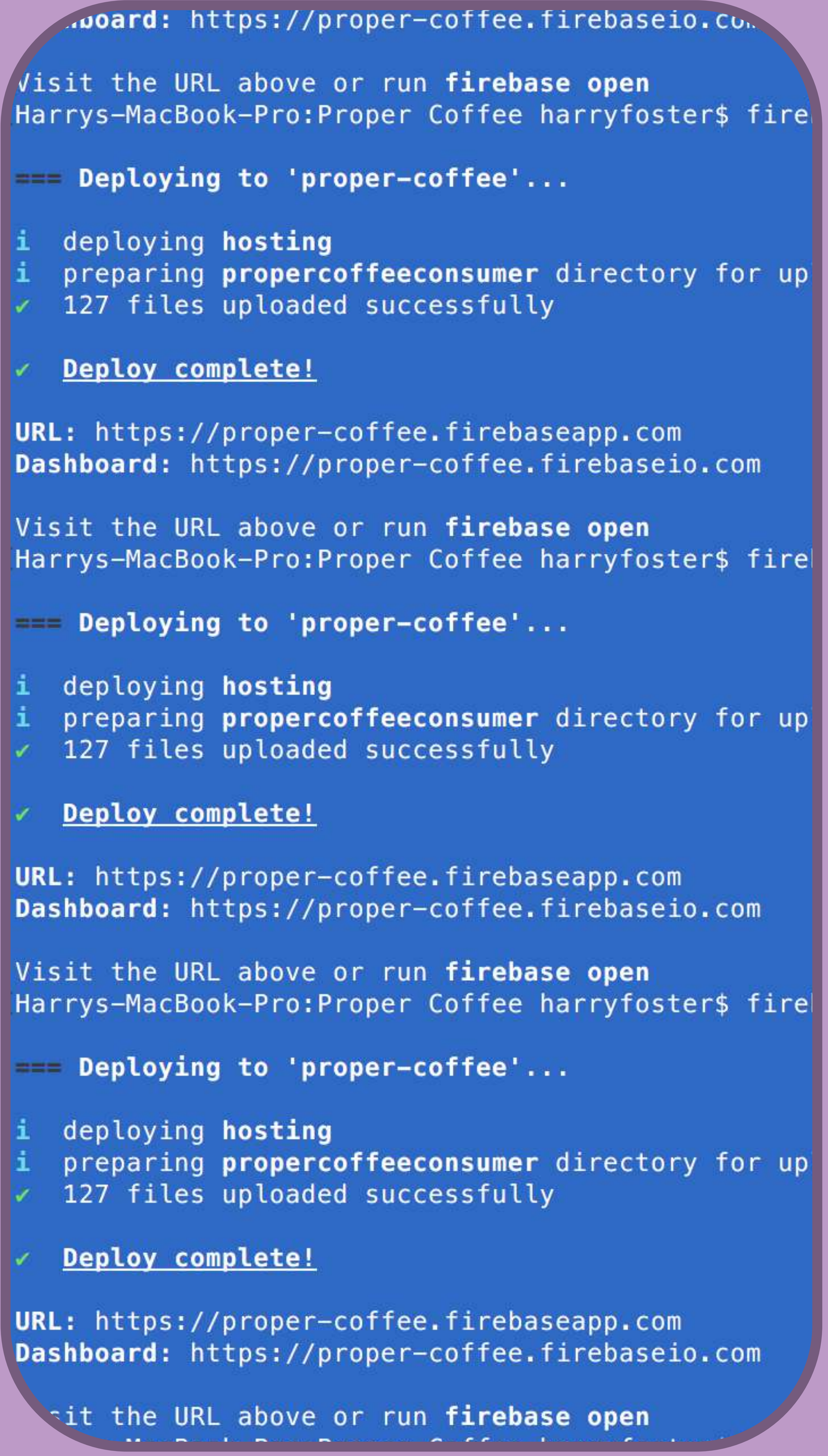
We're excited to announce major upgrades to Firebase!

e've been hard at work adding new features to mal p development easier for you, including Notificatio Storage, Analytics, Remote Config, Dynamic Links, more. We're excited to show you what we've built!

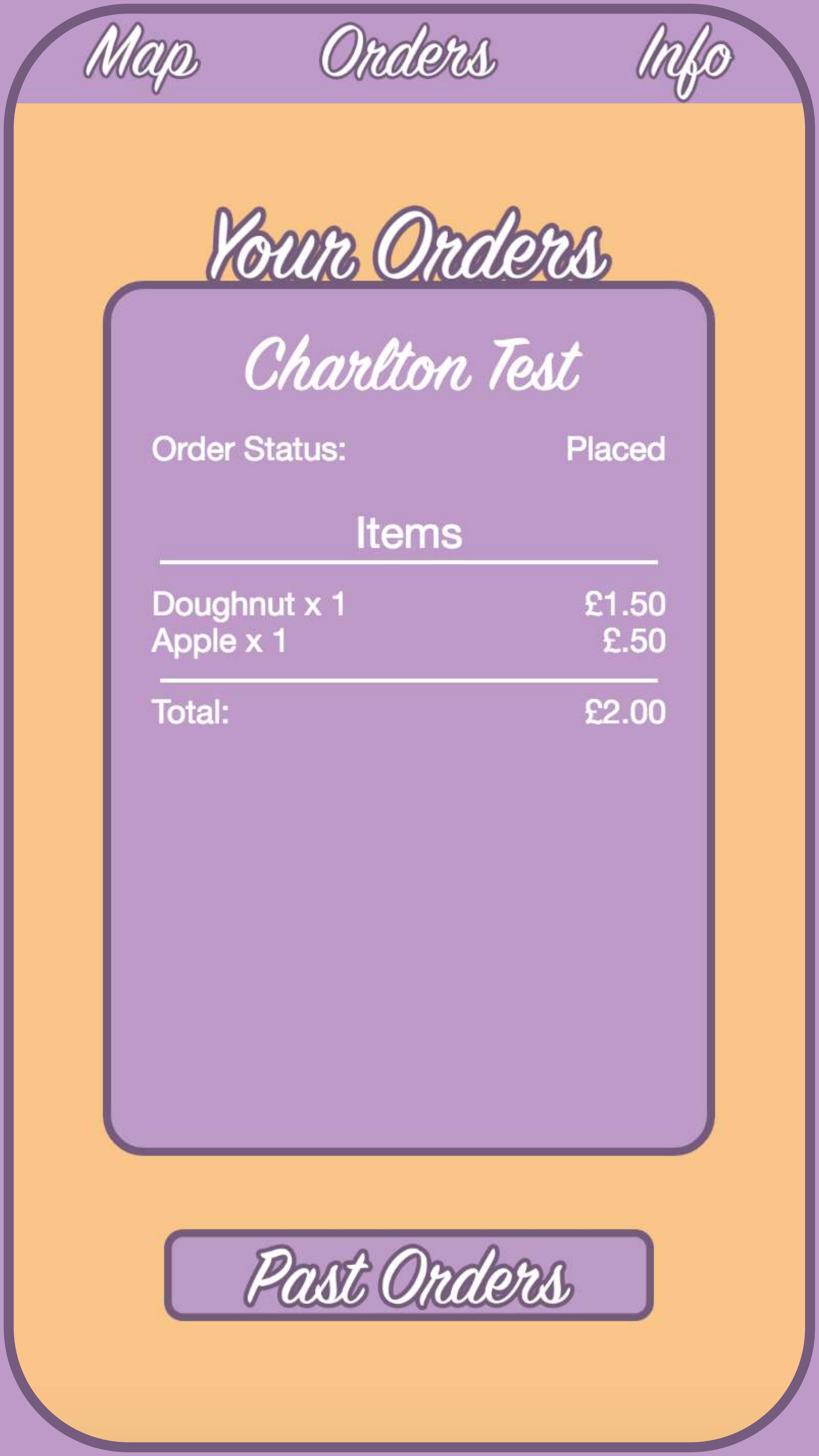
Recently Firebase who are owned by Google have released an update to Firebase, features such as custom URL redirect which I use have now been made free in the new version of Firebase so I decided to convert my project to a new Google Firebase project. The new Firebase is similar but has added functionality like support for file uploads so issues I experienced with being able to upload images are now possible. The new firebase requires some changes to the code so I followed the steps to do this, it was fairly difficult as it broke most of the page and I had to change the way in which the app refers to firebase as most of the functions in the firebase library had changed, once I began to work on it for a while most functions had begun to return to normal however I discovered that firebase 3.X does not yet support Geofire. Firebase 3.X has a lot of benefits, issues I had in the past with Firebase not being able to store images have changed and Firebase now has native support for this however Geofire is an important function within my app as it makes the application scalable, if I didn't use Geofire it the application would have to load the entire database of shops to be useable which could take an extremely long time, because of this I decided I had to roll my application back to firebase 2.X to continue working on it for now until I can get Geofire supported.

Firebase Update

Rolling back to Firebase 2.X should be fairly simple however you cannot move your application back from the new Firebase to the old so I had to create a new project with the old Firebase website, this meant that I had to enter data again as even if I imported the one that I have stored on the newer version of Firebase the unique user IDs wouldn't be the same so I wouldn't be able to authenticate retailers or users with the email addresses that they were using in the past as the unique user ID is generated when you sign up and is random so I can't create an account with the same ID. Facebook authentication was easier to copy over, the Facebook user ID doesn't change so i just had to go to my Facebook app and authorise the new firebase project and that was working. The security rules I'd been using previously were also easy to transfer, as they don't apply to a specific user ID so I could just copy the rules over in JSON format. Firebase tools is the command line tools that you install on your computer to manage Firebase hosting, I experienced issues rolling these tools back in particular. Firebase tools are install using NPM and nodejs, it eventually turned out that the $PATH my nodejs was set up to use was incorrect and Firebase Tools themselves were not at fault once I'd fixed the $PATH to install nodes in the correct place I rolled back to the older version of Firebase Tools without much issue and could then continue working on the project, Firebase 2.X is going to be supported until later this year so continuing to use the older version should suffice until I can get Geofire support.



```
board: https://proper-coffee.firebaseio.co

Visit the URL above or run firebase open
Harrys-MacBook-Pro:Proper Coffee harryfoster$ fire

=== Deploying to 'proper-coffee'...

i  deploying hosting
i  preparing propercoffeeconsumer directory for up
✓  127 files uploaded successfully

✓  Deploy complete!

URL: https://proper-coffee.firebaseapp.com
Dashboard: https://proper-coffee.firebaseio.com

Visit the URL above or run firebase open
Harrys-MacBook-Pro:Proper Coffee harryfoster$ fire

=== Deploying to 'proper-coffee'...

i  deploying hosting
i  preparing propercoffeeconsumer directory for up
✓  127 files uploaded successfully

✓  Deploy complete!

URL: https://proper-coffee.firebaseapp.com
Dashboard: https://proper-coffee.firebaseio.com

Visit the URL above or run firebase open
Harrys-MacBook-Pro:Proper Coffee harryfoster$ fire

=== Deploying to 'proper-coffee'...

i  deploying hosting
i  preparing propercoffeeconsumer directory for up
✓  127 files uploaded successfully

✓  Deploy complete!

URL: https://proper-coffee.firebaseapp.com
Dashboard: https://proper-coffee.firebaseio.com

sit the URL above or run firebase open
```



Your Orders

Charlton Test

Order Status: Placed

Items

Doughnut x 1	£1.50
Apple x 1	£.50
Total:	£2.00

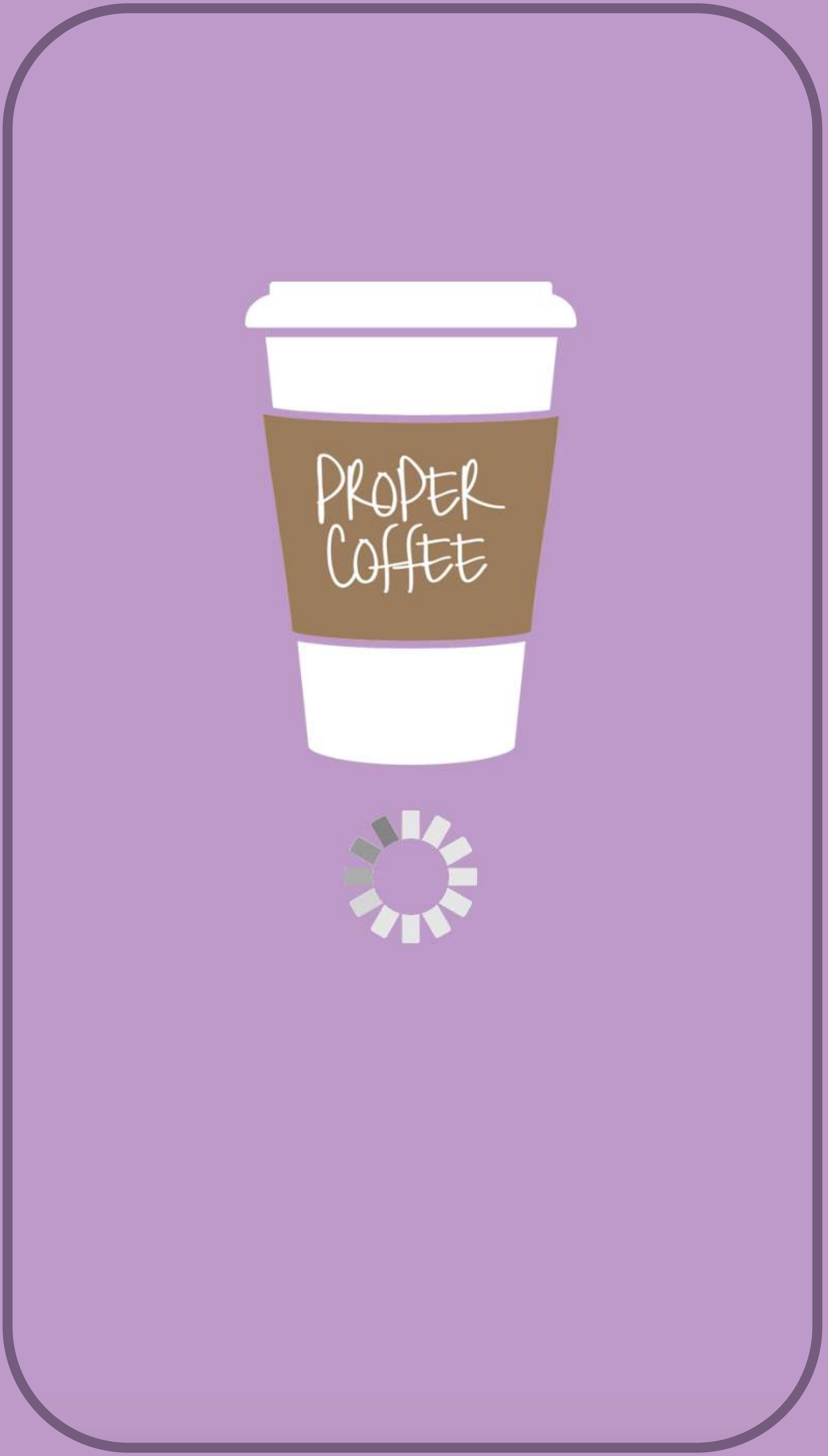
Orders Page

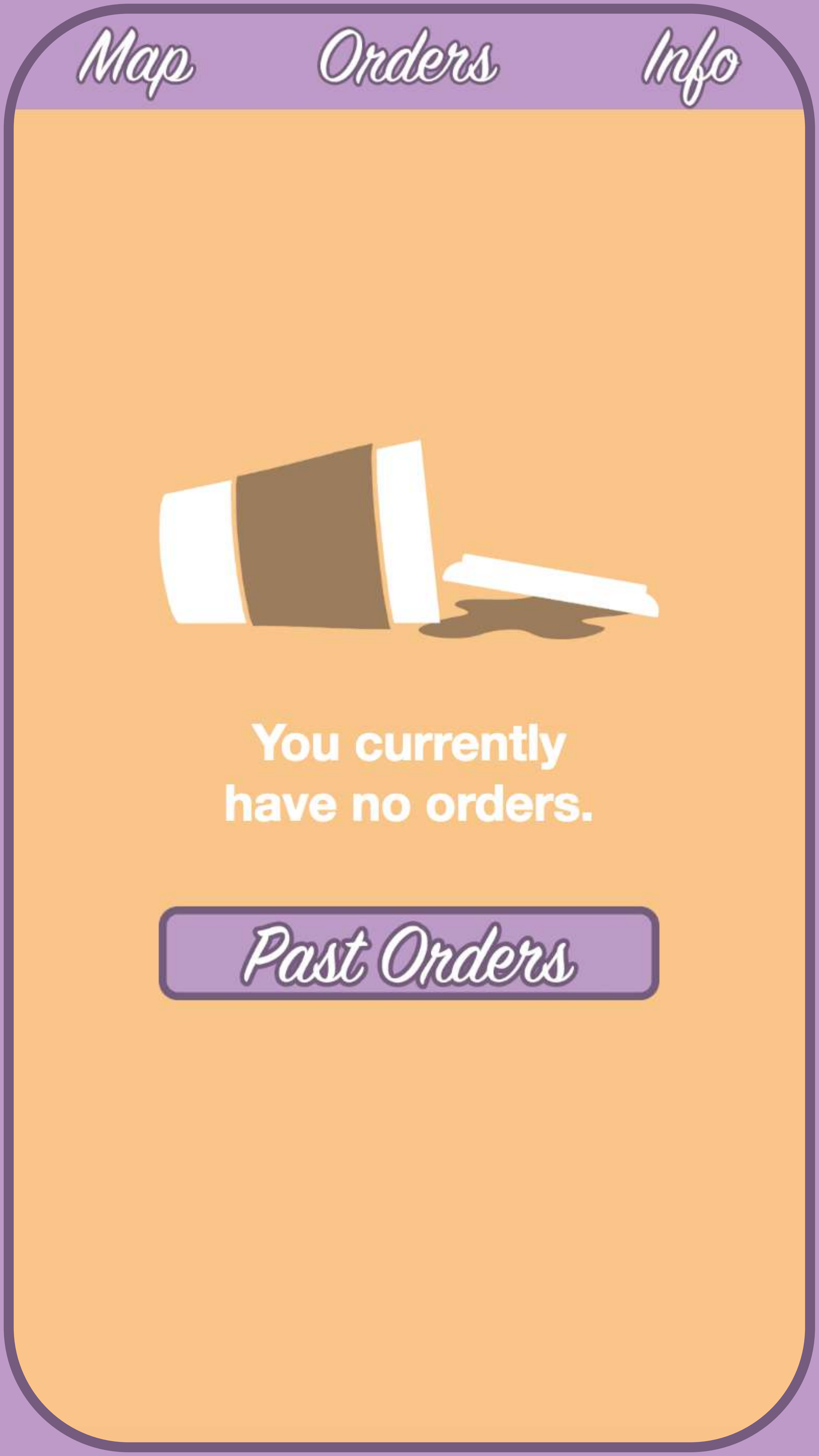
Now that a user can place an order on the consumer app I want them to be able to see this order on the orders page, to do that I followed my wireframes I'd created in the past as a basic template but made some tweaks as there were things I wanted to show that I hadn't included when i created the wireframes. The orders page checks when the webpage is loaded whether the user has any active orders and it also refreshes this when a user places an order. The orders page finally adds an extra dimension to the application and makes use of the page tabs on the menu bar.

As it's unlikely I'm going to be able to implement the info page fully to represent the users profile and loyalty scheme I decided to put placeholder content when the user is signed in rather than just the text user signed in, this is really basic as it's simply an image file of the info page wireframe but it's good for representing what the application is going to be able to do once it does become implemented.

Polishing Functions

I want my application to have a polished native application feel to it however for a variety of reasons it feels like the web application that it is. To make the application feel more slick I am going to add a variety of additional design changes, CSS animations rather than showing and hiding elements will make the app feel smoother, I began implementing this by adding in a slide in of the popup bar that appears when you press a shop, instead of just appearing it is going to slide up from the bottom of the page. I'm also ensuring that font is unified across the application making everything feel more cohesive. Another way In which I can make the application feel more native is not letting the user see elements loading, I can do this by adding in loading and splash screens, I've implemented a full screen splash screen which loads before the user can see the website so that the map is loaded with at least 1 marker before the user sees it. Continuing to implement these design changes is going to make a significant improvement to my ability to showcase the application.





Styling & Bug Fixes

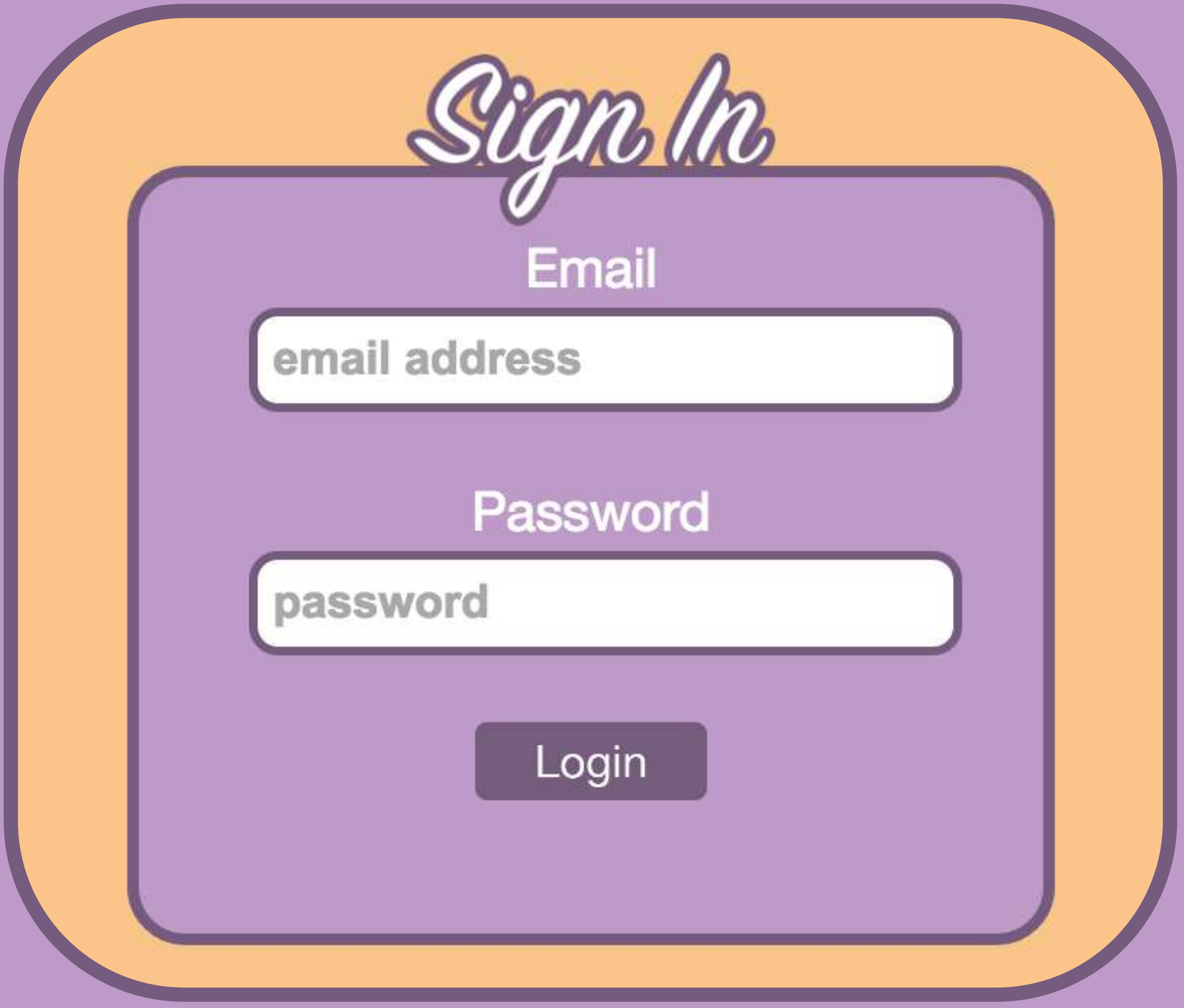
I've continued to develop the way my app looks, work on adding a little more functionality and fixing some of the more minor bugs that are apparent. An issue I'd been seeing occur quite a lot was the duplication of menu items and the categories that are displayed when you view the menu, that wasn't great and was fairly frustrating as it seemed to occur sometimes and not others. Eventually I managed to work out that the trigger I had for clearing the menu only happens when you go to the menu and then go back but the menu is created when the user clicks on the marker so if for some reason they decide not to open the menu, or get directions instead etc the duplication occurs, I added an if function that detects whether the menu is present when you exit the popup and clears it if it is fixing this bug.

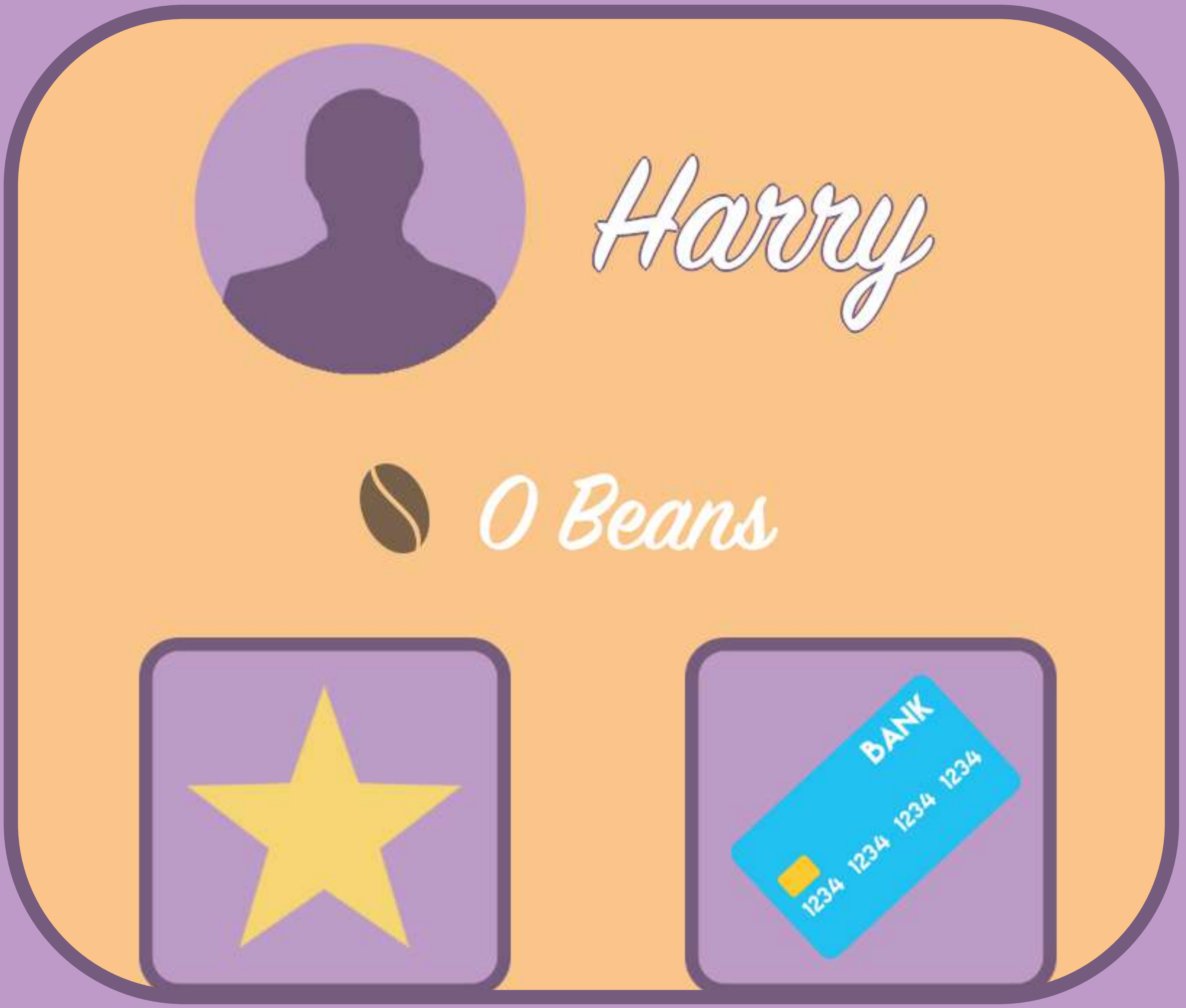
I then added a display for if no order is currently present on the order page so that it wasn't blank and indicated this status to the user, I had a wireframe of this so it was pretty quick to do.

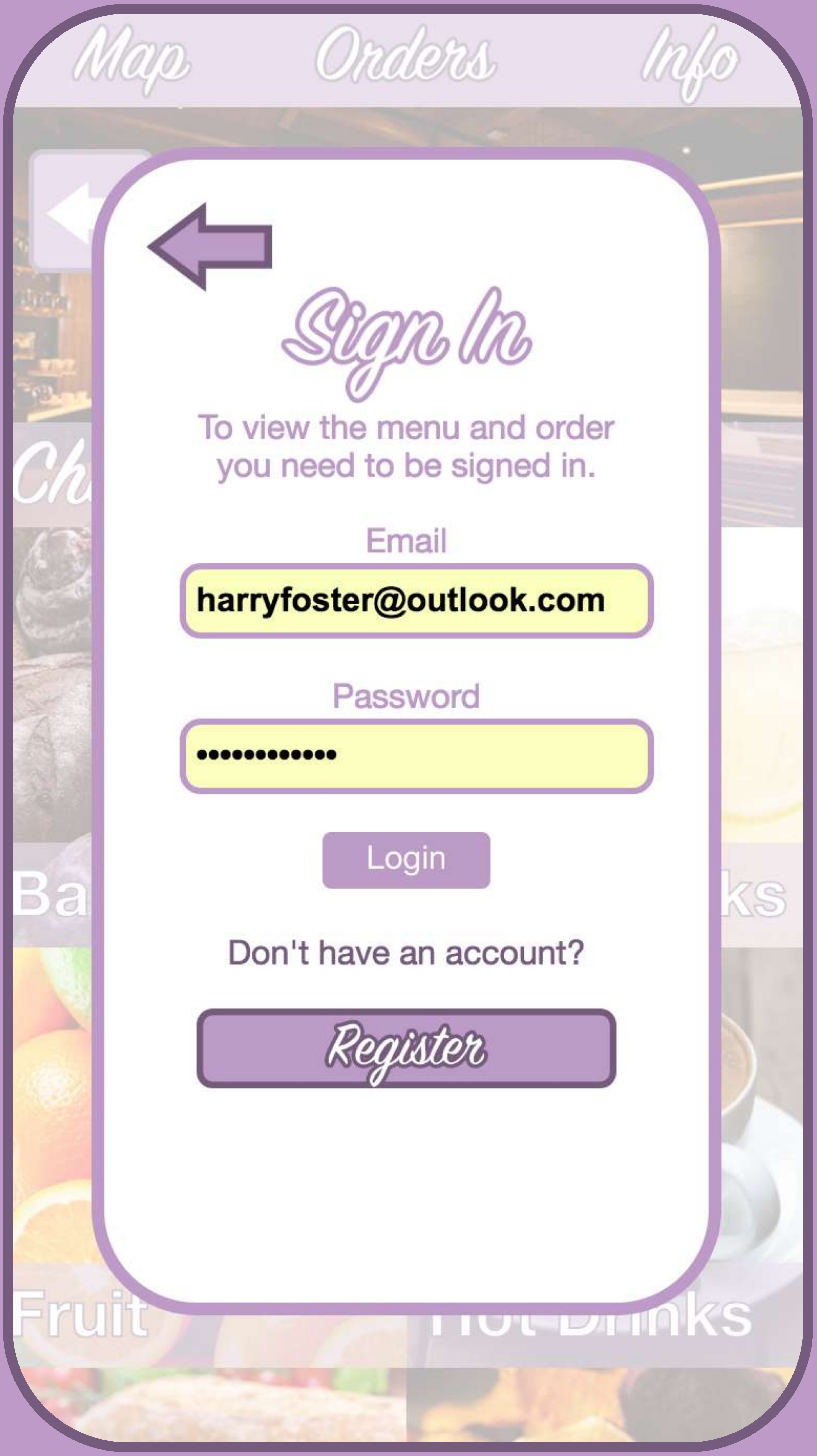
Styling & Bug Fixes

I also decided to implement some styling on the info page, I wanted the user log in and registration form to have styling which reflects the styling throughout the rest of the application and I based the look on the orders page and the existing content on the login form.

I then wanted to have the info page display a little more than a screenshot of the wireframe, I deconstructed elements from the wireframe and separated images I would need out into assets, I also created a blank profile image placeholder as I don’t want uploading of profile pictures just yet as this would take time and use the base64 implementation I’ve used before when firebase 3.X would have full support for images. The info page now takes the users first name and displays it on the info page and there is a log out button rather than the past orders button I had on the wireframe as I felt this made more sense.







To view the menu and order you need to be signed in.

Email

harryfoster@outlook.com

Password

•••••••••••••

Styling & Bug Fixes

I also wanted to ensure that a user with an account but isn’t logged in or a user without an account understand why the menu isn’t working when they try to place an order, In order to do that I created an overlay of the menu explaining to the user that they need to log in before they can view and order from the menu or register for an account.

This makes the app flow more as a lot of the time whilst user testing people didn’t understand why they couldn’t order and i’d have to sign them in, I always intended to implement this but it was just low on the priorities.

Styling & Bug Fixes

From user testing the application with these new additions I have discovered a few things.

The order page is where people go to try to place an order sometimes if they haven't pressed on one of the shop pop-ups yet, telling them that they have no current orders is important but showing them how to place an order would also help, having something that's only shown once to a registered user so they understand the flow of the app.

If a user clicks an item with a quantity of x1 more than once they expect the quantity to increase, implementing this could be worthwhile.

The maths on the current order popup isn't always accurate if the user is ordering items in different quantities and seems to multiply every item by the last quantity added, bug fixes are needed for this.

People try and click every button, even after I tell them the profile page doesn't function yet, placeholder content for the rewards and the payment parts could be useful, I could put placeholder content in stating it as a demo version of the app therefore those functions serve no purpose when I implement them to give a better experience at the degree show.

Sometimes my orders page repeats the price on multiple lines, flooding the page and crashing the app, bug fix needed.

The page transitions sliding in from the right to the left are jarring because the page they were viewing disappears instantly so the wipe in takes place over white which doesn't feel like a smooth transition.

Fixing these bugs are what I'm going to focus on next, I also think the retailer app needs a little more work, things like the log out button and styling changes are yet to be done and whilst it is less important to have these changes on the retailer app than the consumer app it is something I'd like to add if there's time.

Finishing Ordering System

I want my application to be in a condition for the deadline that no matter what the user does they can't end up in a situation where it looks like the application isn't performing as it should be. In order to address this I began working on the ordering page again, currently once a user has placed one order the application can handle that fine however if they place another order from the same shop it adds it onto the end of the order and duplicates the pricing and if a user places an order from 2 different shops the orders clash resulting in some of the content from just one order being displayed like the shop name. To tackle this issue I decided to change the way the ordering page works so that a user can have multiple current orders running side by side and can view the separate orders by navigating between them. To tackle this issue I had to change the way in which the application reads the orders from firebase, instead of directly reading what is there and just showing it on the page I decided that the application will have a look at all of the orders that are currently active for the user and then push these orders into an array. The orders page checks the array and looks at which order is in position 0 in that array and shows just this order first by default. This fixed the issues with clashing orders and by pushing them into an array and also adding in the ID of the shop that the order was from it stopped the orders from clashing as each order has a unique identifier yet I can still find out which shop it is ordered from by accessing the shop ID in side the array. Then I needed to work on creating a way for a user to navigate between orders as currently they can only see the order at position 0 within the array. I decided to have some text at the bottom of the ordering box which tells the user they are viewing order "1 of of 1" or "1 of 2" etc and then have navigation arrows either side of this text to go to the previous order or the next one, this will work in a cycle so if they get to the highest order in the array when they press the next button they will go to the lowest and I achieved this using if functions.

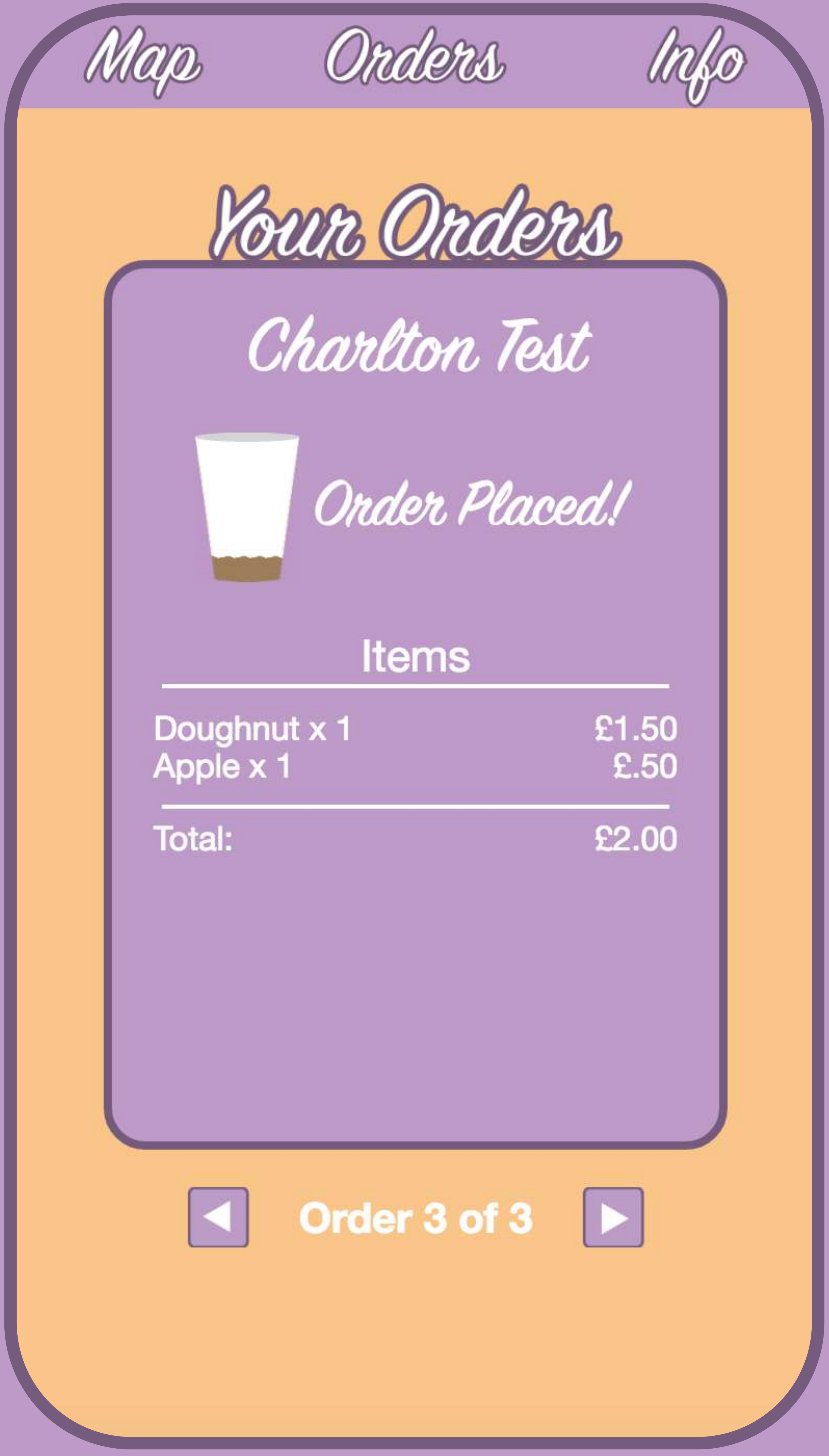
The trickiest part about implementing this way ensuring the application knew which order it was currently showing in order to go up or down to the next one, javascript marks elements in an array starting a 0 but a user would probably find it much it much easier to understand if the orders began at 1 so I have to translate and add 1 to each of the numbers in the javascript array when I'm displaying them in the UI.

Finishing Ordering System

It is actually fairly unlikely that a user would have multiple orders running at the same time but it is possible and as currently there is no way to edit or cancel an order and it's unlikely I'll have time to implement these before the deadline it seemed a sensible way of getting around the fact the multiple orders would cause the app to look extremely buggy and will work much better for demonstrative purposes. Another element that I thought was important to change on the order page is the order status, the status of the order is probably the most important piece of information on that page so it needed to be emphasised. I decided that implementing a graphical progress element to display the status alongside text would be informative, draw the attention of the user and be visually pleasing so I began working on some short animated images of a coffee cup being filled up to represent this information. All the images needed to represent the different stages of the order are loaded when the page loads to ensure smooth display of the relevant images and when the status of the order is changed in firebase a listener in the javascript updates the source of the image displayed on the order page. The image is an animated gif which starts at the ending position of the previous image so it seems to the user as if the cup suddenly starts filling up more. The order once marked as done by the retailer app previously had it's status simply changed to done and this was reflected by the consumer app however I wanted to change that so the order disappeared. Now when the done button is pressed in the retailer application it creates the order again in both the users part of firebase and it's own in a subdirectory of orders called old orders and deletes the active order, I then changed the javascript within both applications to be able to stop displaying an order once it was removed.

In recent updates I have removed the previous orders button from the orders page, whist this would be a feature in future version of the application to be able to review old orders in the old orders part of firebase and place orders again based on past orders it is simply too large a task at this point in the projects lifecycle and I think it would be more important to have a more polished application that have this feature supported.

Finishing Ordering System



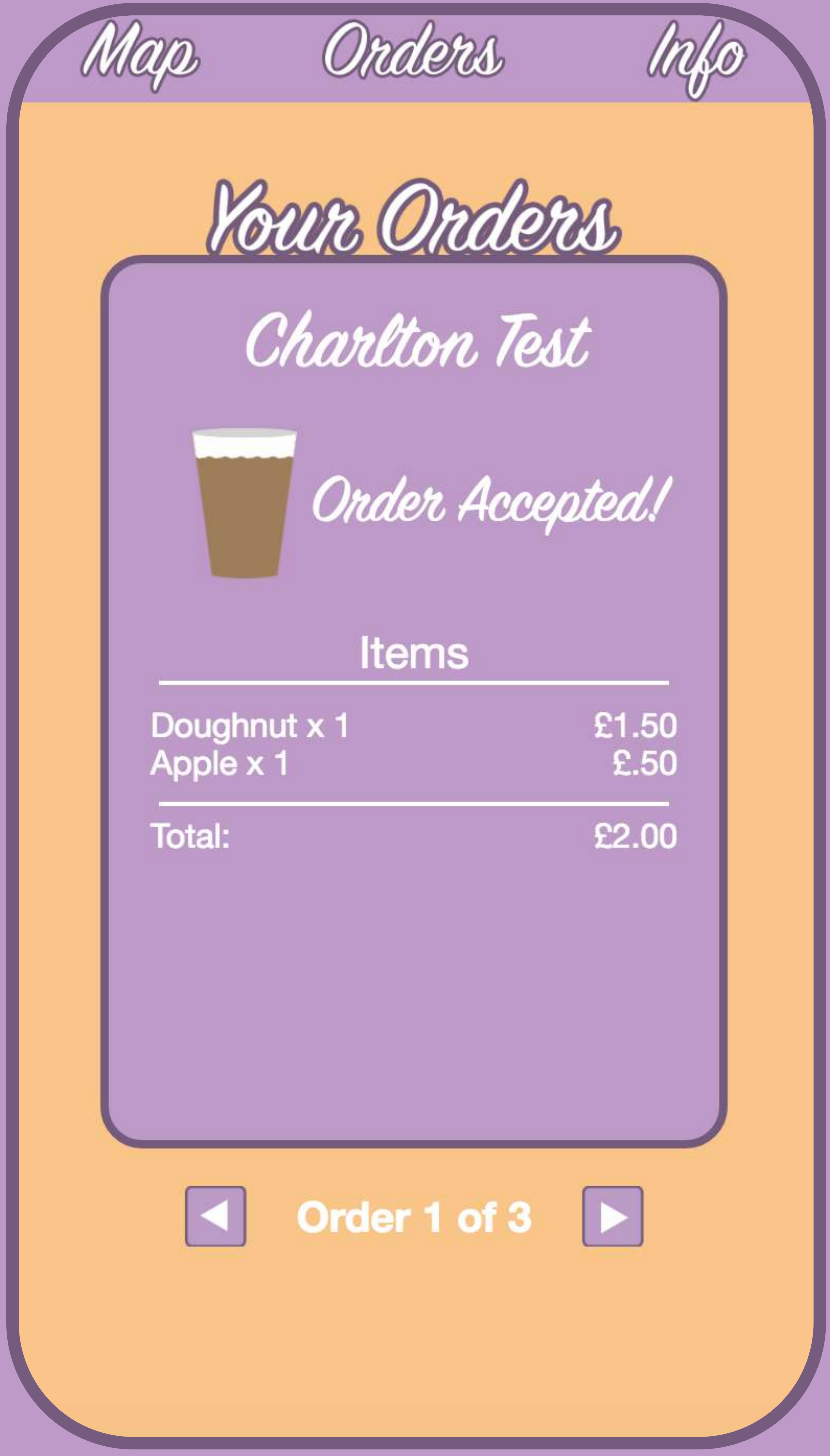
Your Orders

Charlton Test

Order Placed!

Items	
Doughnut x 1	£1.50
Apple x 1	£.50
Total:	£2.00

Order 3 of 3



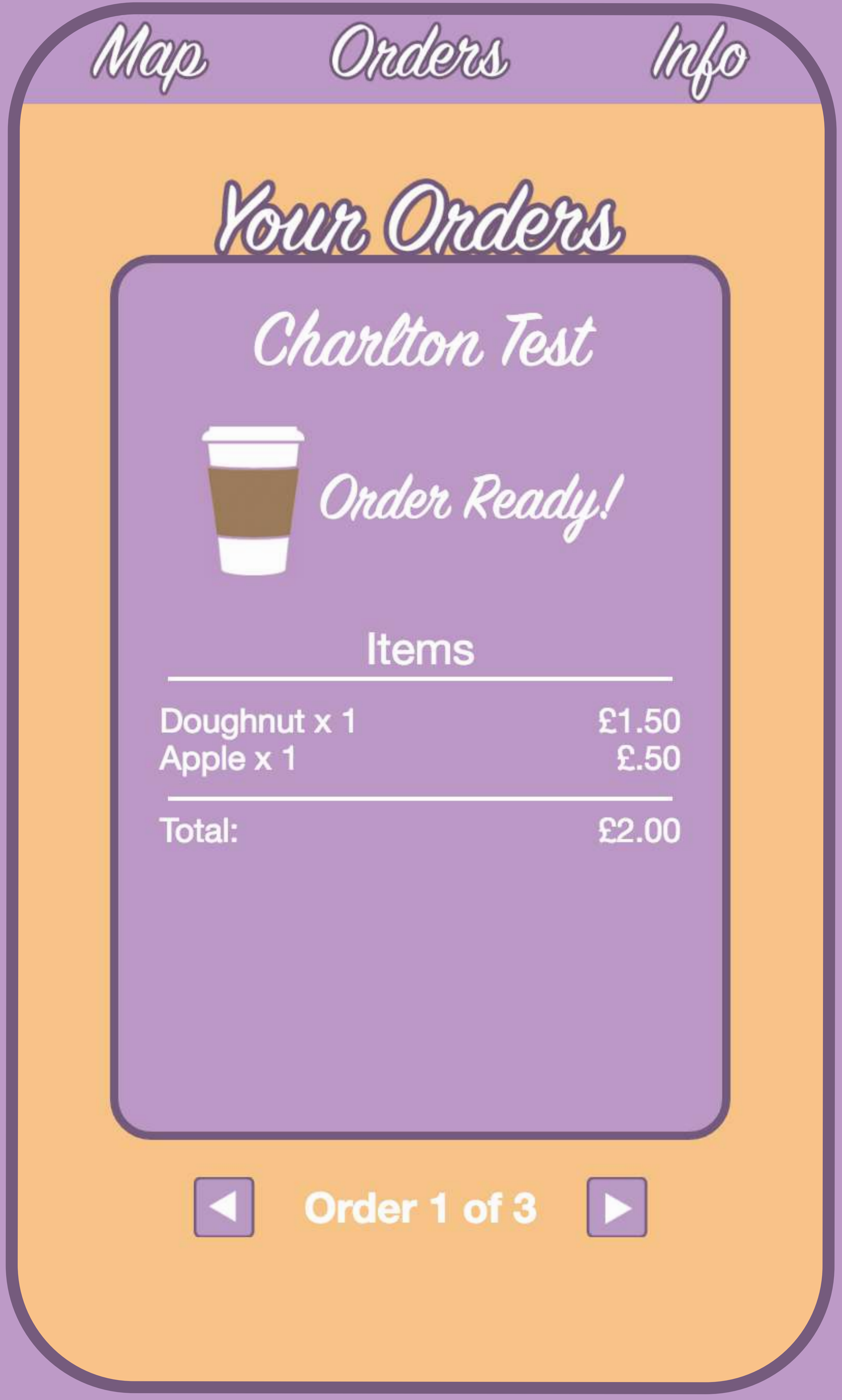
Your Orders

Charlton Test

Order Accepted!

Items	
Doughnut x 1	£1.50
Apple x 1	£.50
Total:	£2.00

Order 1 of 3



Your Orders

Charlton Test

Order Ready!

Items	
Doughnut x 1	£1.50
Apple x 1	£.50
Total:	£2.00

Order 1 of 3

Finishing Touches

To get my application ready for presenting for the summative I wanted to smooth off any rough edges, ensure that any button that is pressed does something even if they actual function isn't active, there should at least be a placeholder message, make sure that the application cannot be broken by a user and that it looks visually appealing.

After deciding to move the prompt when a user isn't logged in and they're try to view the menu to the point where they are trying to order for UX purposes I realised I had some issues with the application, depending on the order of page presses sometimes the user can be presented with a blank page. There were additional bugs of this type:

Bad maths — application couldn't add up correctly with a current order if the user has products of multiple quantities, this required rewriting the way the app deals with the quality.

The registration page couldn't be accessed from some pages due to the animation that had been placed on it, I needed to add some animation in places and hide the not logged in pages in other areas.

The marker which marks your location on the map is a circle not a marker meaning it doesn't scale if you zoom, I changed it to a circle marker but also had to investigate how to add styling in the leaflet documentation, turns out you can't add it whilst creating it you need to apply a style set to it.

The retailer app had buttons which did nothing like adding shop details I decided to comment these out as they would be a useful feature but simply not doable for the deadline.

If a shop doesn't have a menu or shop image it causes the console to log an error and a user can pull up a blank menu, I changed the way that the markers are created so that on the user clicking them an if function displays no menu available if there isn't one.

Finishing Touches

Retailer application doesn't know what to do if a shop doesn't have a header image, e.g. it's a new shop I had to create an if function the place an instruction image in instead to tell the user to upload one and to not break the styling.

The order page of the retailer application is whats displayed by default but looks very bare if there are no current orders, I decided to implement a placeholder no orders image just like on the consumer application to let the user know this.

The retailer app had no log out button and didn't know how to handle a log out, fixing this was simple as it was implemented in the same way as the consumer app.

The font used in a lot of areas of the application is disjointed, some arial, some helvetica etc, I changed all to use helvetica or operating system default (roboto if being used on an android phone), text outline in some areas of the app was garish because of the way that CSS text outlines always center on the text, in order to change this I had to rethink the styling and go for a more plain white font in the menu system and change the background colour of the menu containers and opacity of overlays to ensure it stood out.

Some form elements didn't appear correctly on webkit based mobile browsers as they overrode the styling, I ensured that the application displays correctly on these devices by turning some webkit styling off and using images and text which interact with the form via javascript instead.

All of the buttons in the retailer application needed changing in the retail application to match the consumer app.

Removed the past orders button in the retailer application as this function isn't going to be available with the first iteration of the application.

Added alerts where a retailer trys to print or edit and order to let them know that no printer is connected and that the order isn't available for editing.

Finishing Touches

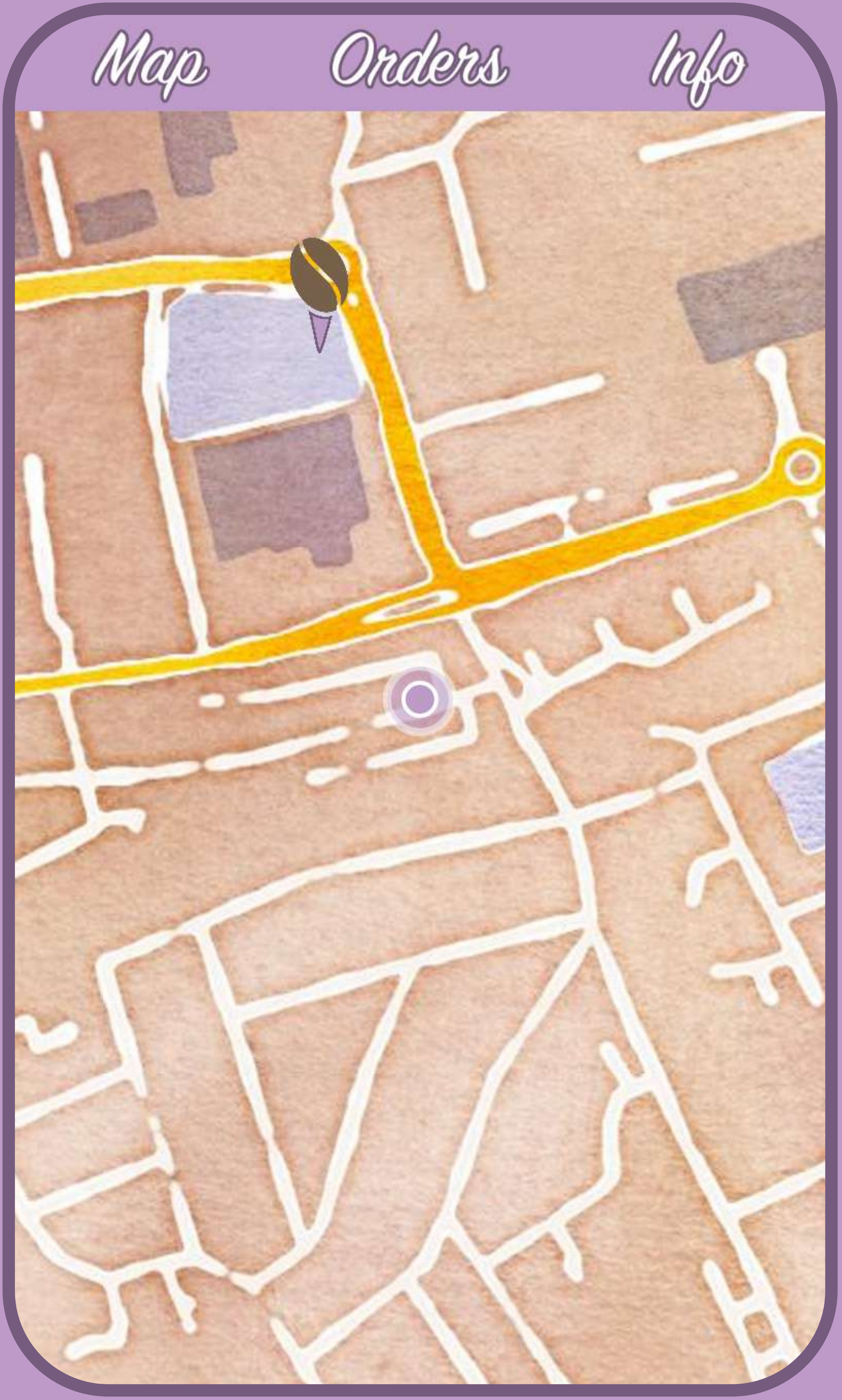
Fixed some positioning elements which made the application hard to interact with when using a smartphone keyboard.

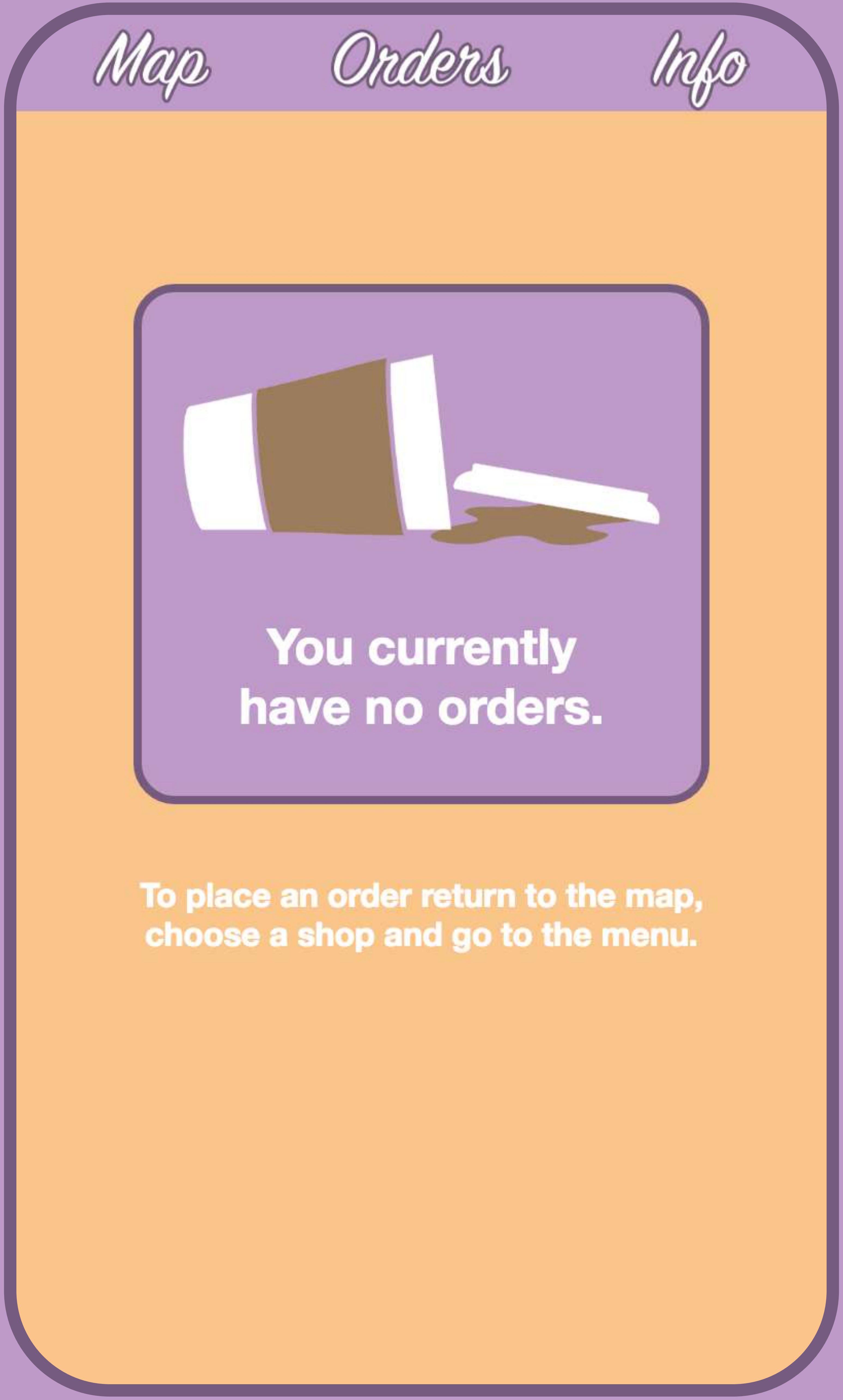
The only part that now needs to be done is final testing and adding data to the database to show the variety of shops on there.

The application is now ready to be demonstrated as a concept and should work smoothly in all aspects which it can currently do, I feel I set myself a difficult task in trying the achieve this application however I think i also went far past the Minimum viable product that it could have been.

In the future I would love to continue to work on this application to integrate the payment system, loyalty system, firebase 3.X and past orders. If I got to this point I feel that I could began beta testing with selected shop partner and a small public release which could lead to a Proper Coffee becoming a legitimate business.

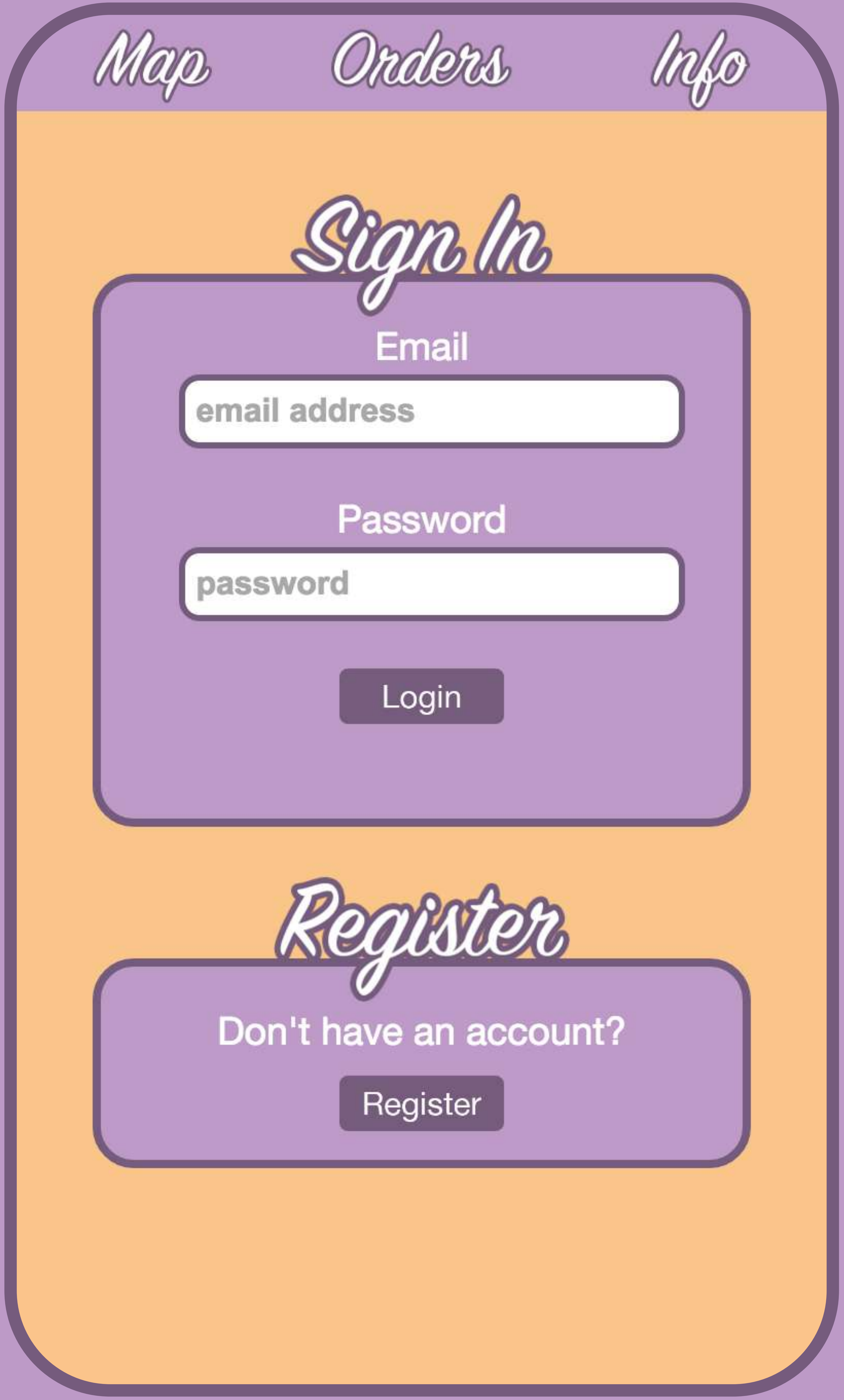
Final Screenshots





You currently
have no orders.

To place an order return to the map,
choose a shop and go to the menu.



Sign In

Email

email address

Password

password

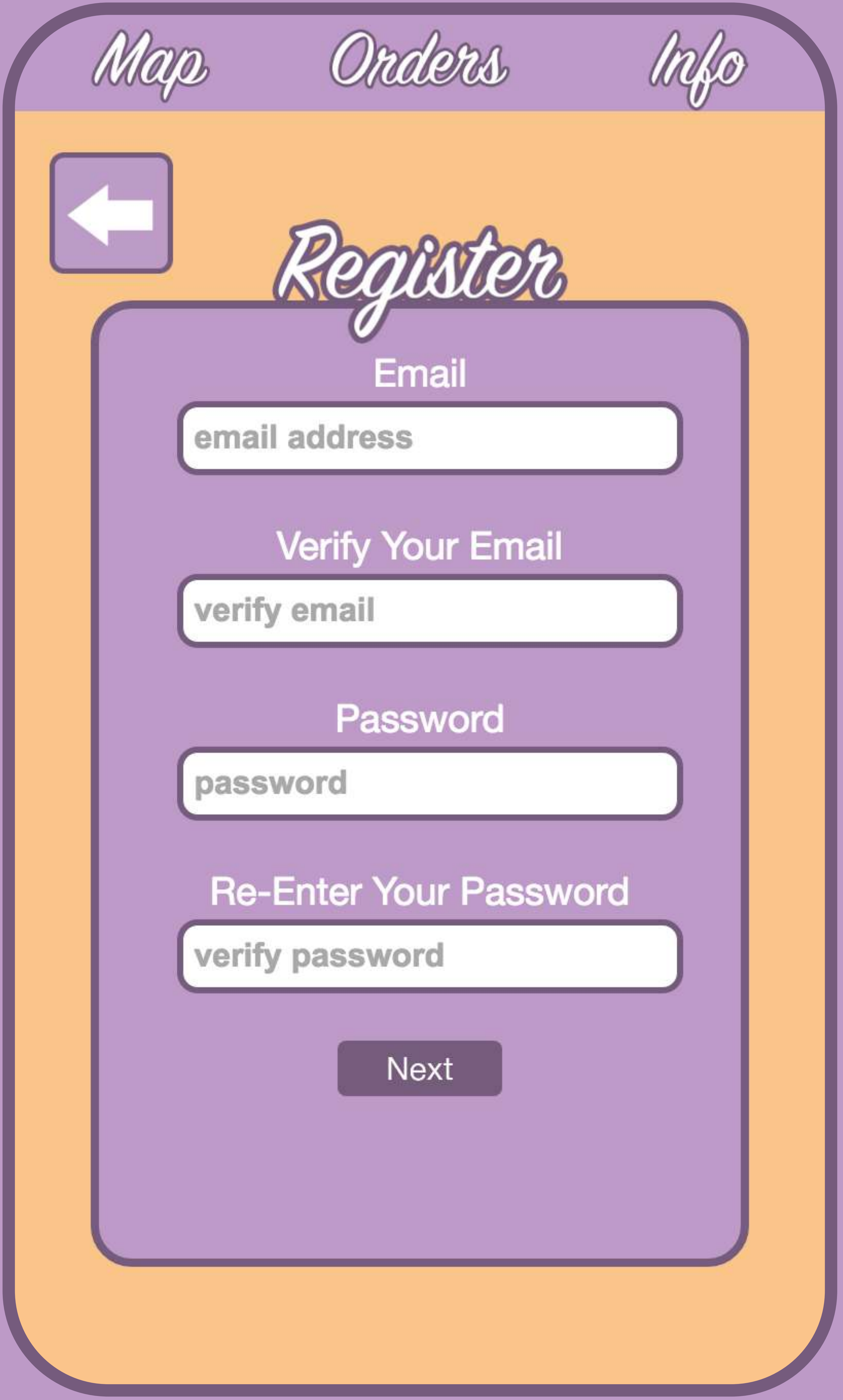
Login

Register

Don't have an account?

Register

Final Screenshots



Register

Email

email address

Verify Your Email

verify email

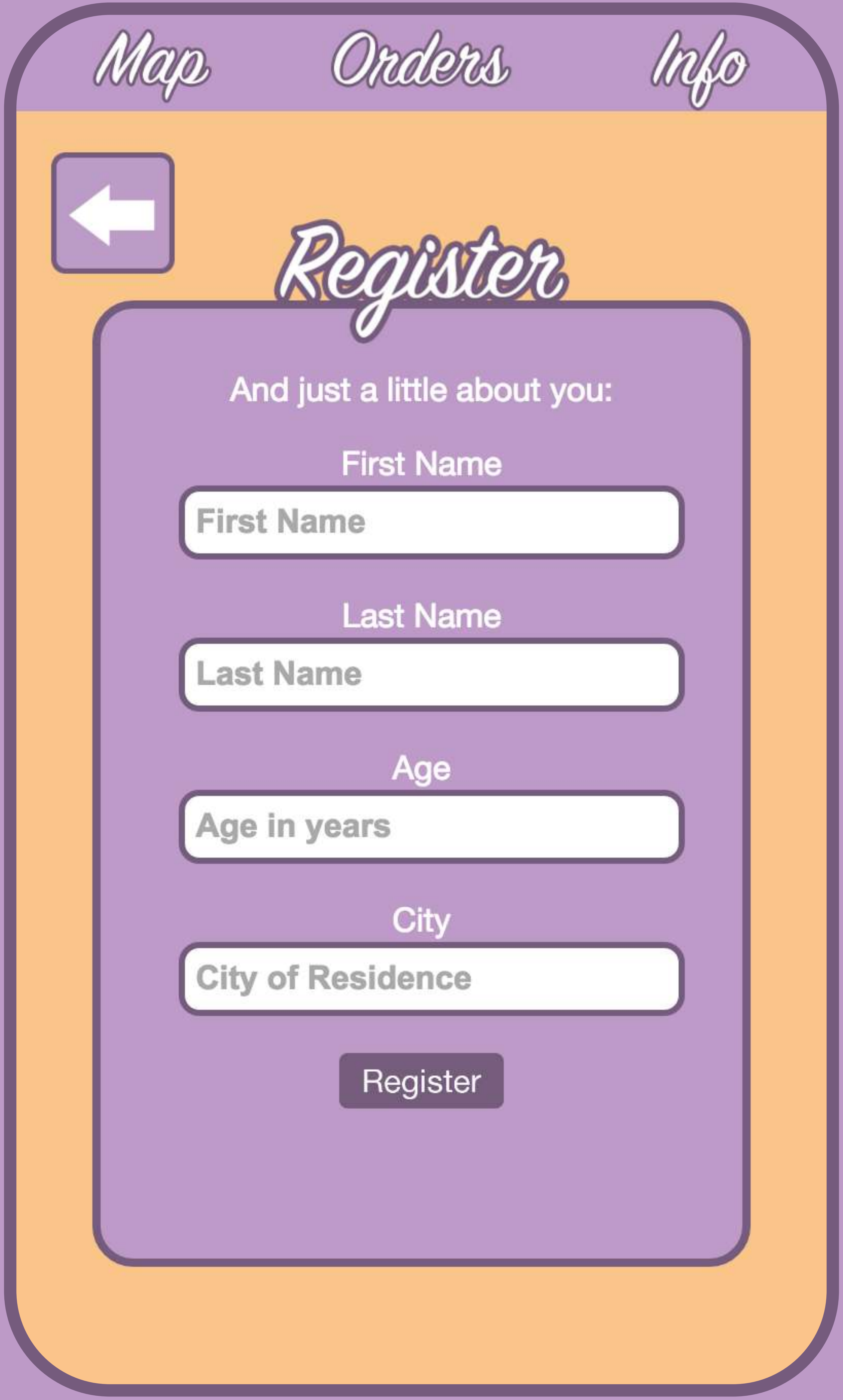
Password

password

Re-Enter Your Password

verify password

Next



Register

And just a little about you:

First Name

First Name

Last Name

Last Name

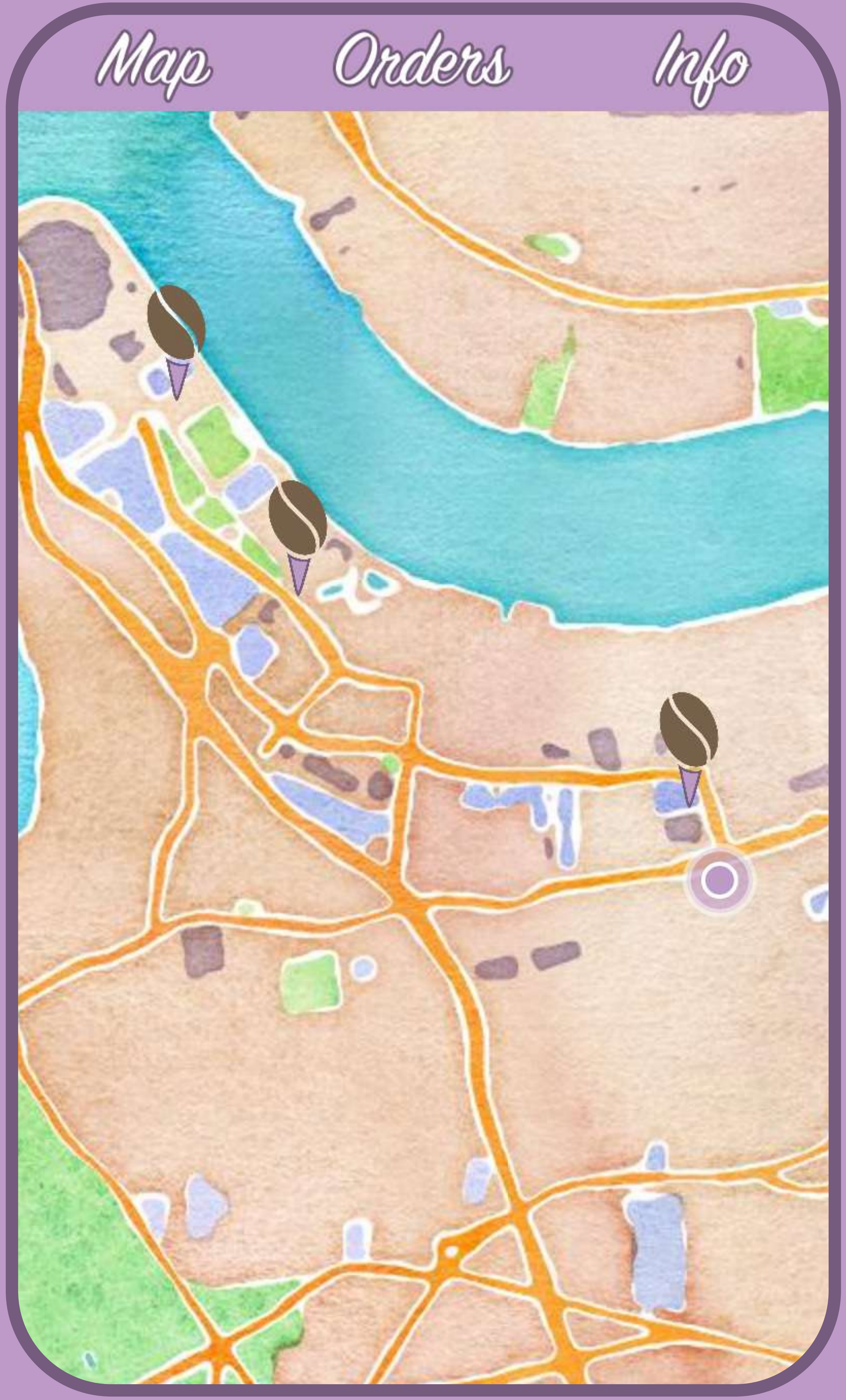
Age

Age in years

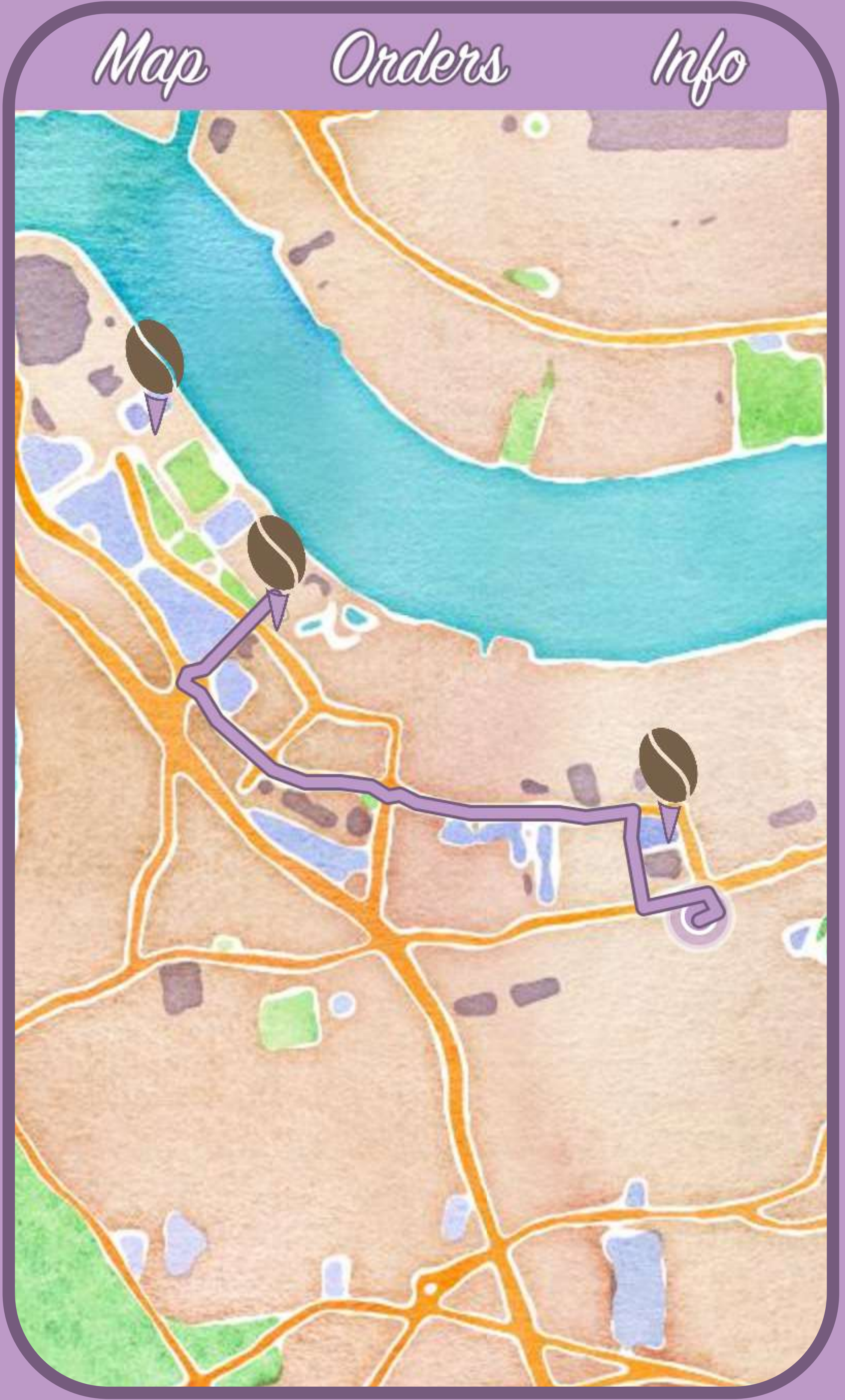
City

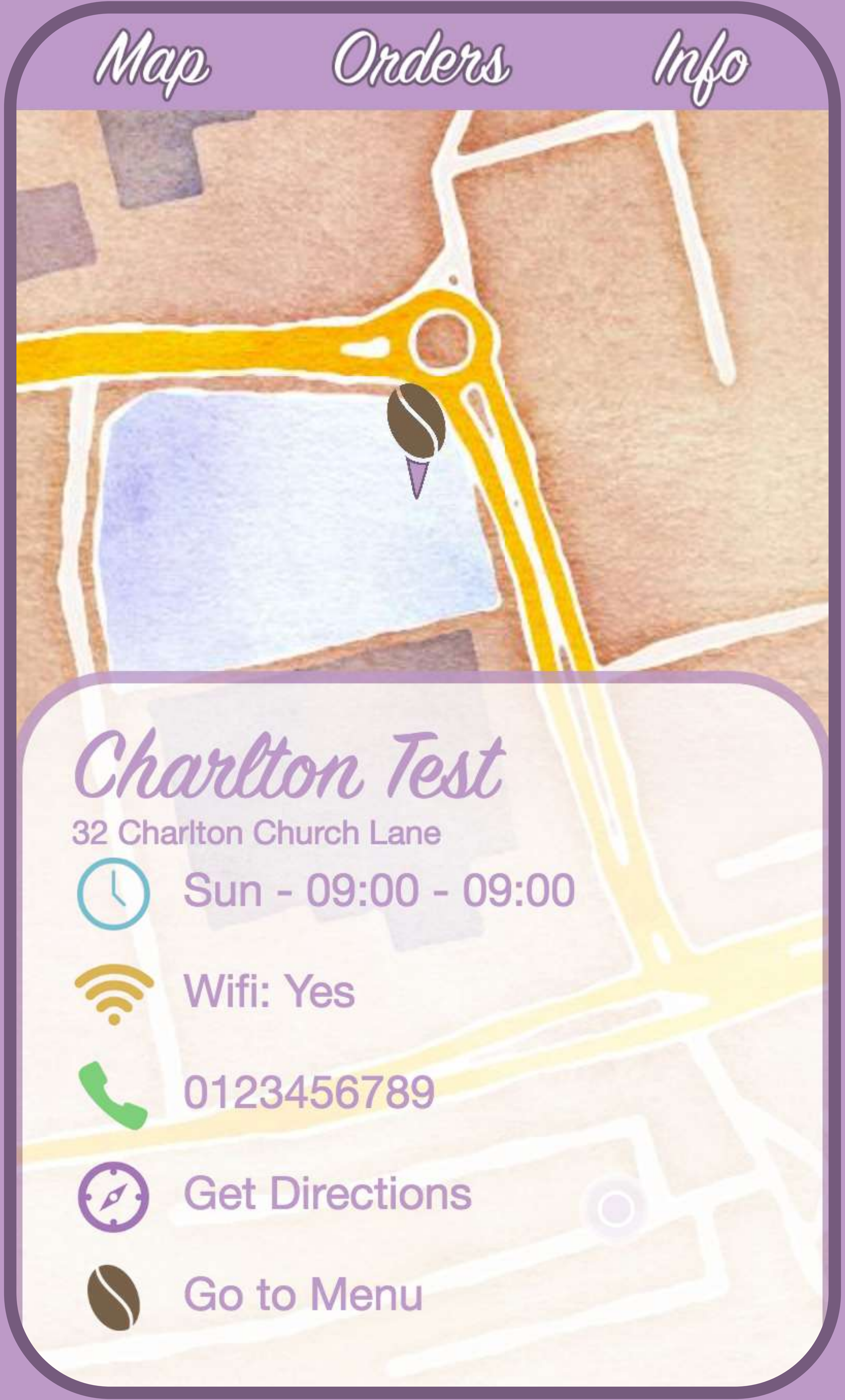
City of Residence

Register



Final Screenshots





Charlton Test

32 Charlton Church Lane

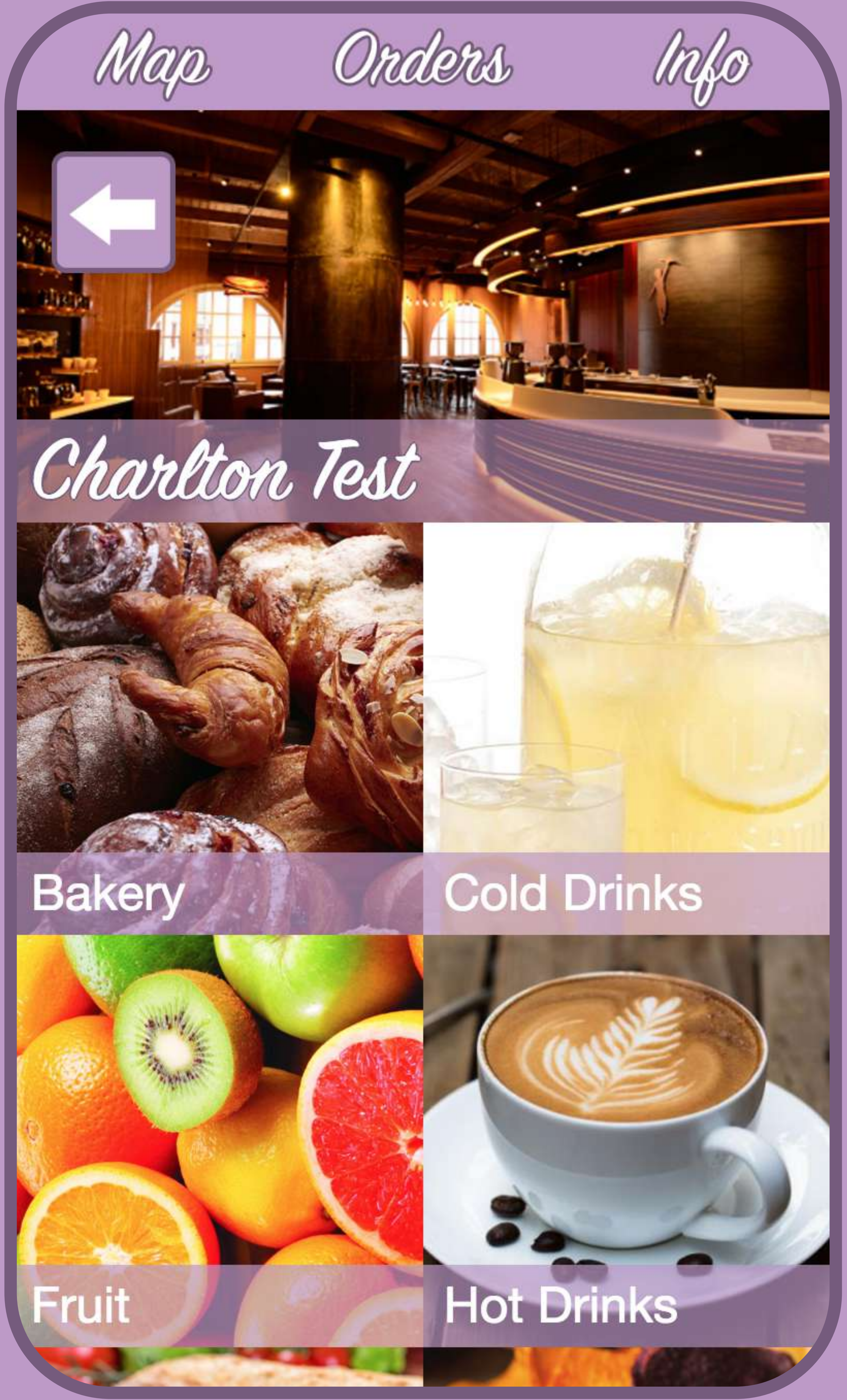
Sun - 09:00 - 09:00

Wifi: Yes

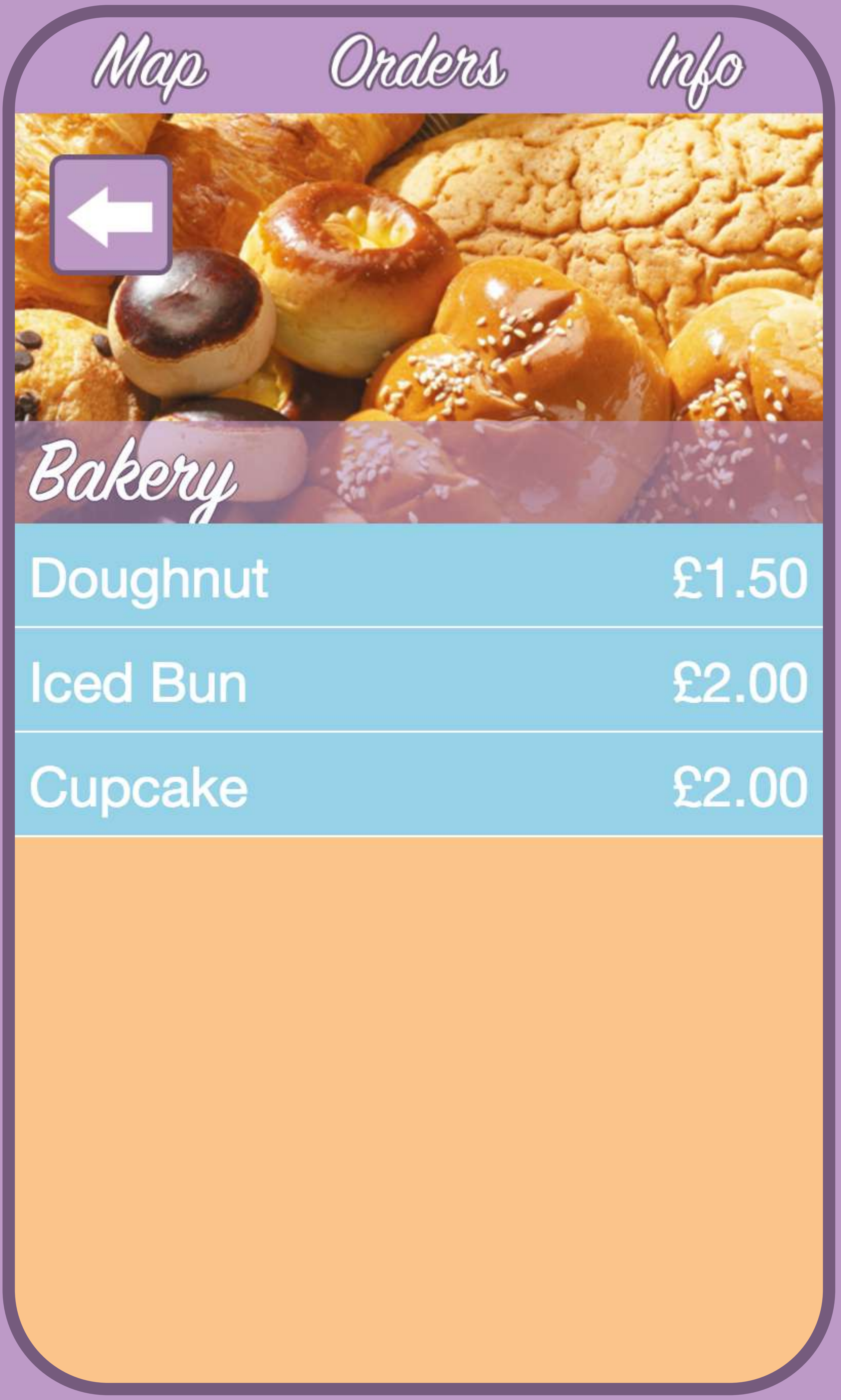
0123456789

Get Directions

Go to Menu

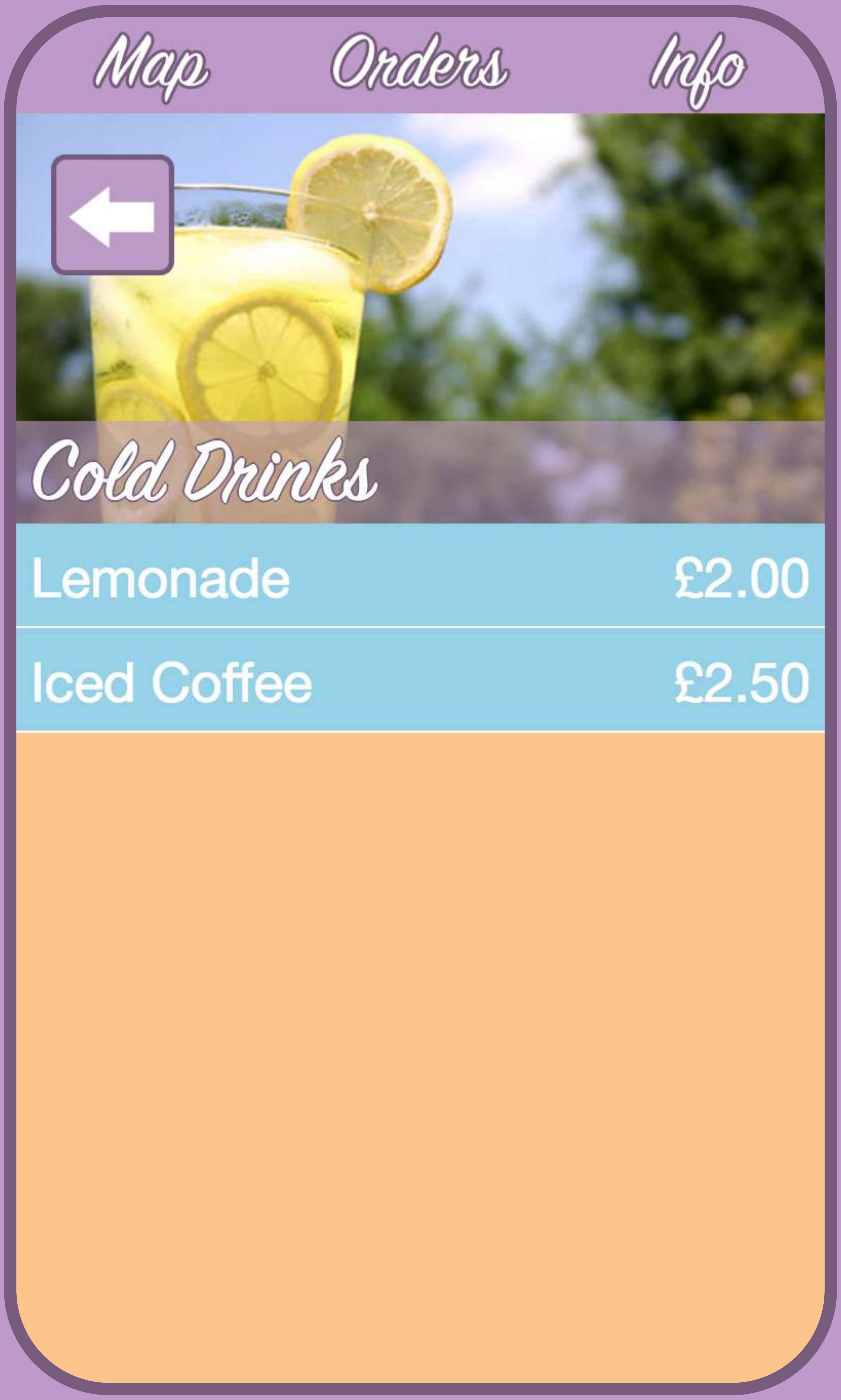


Final Screenshots



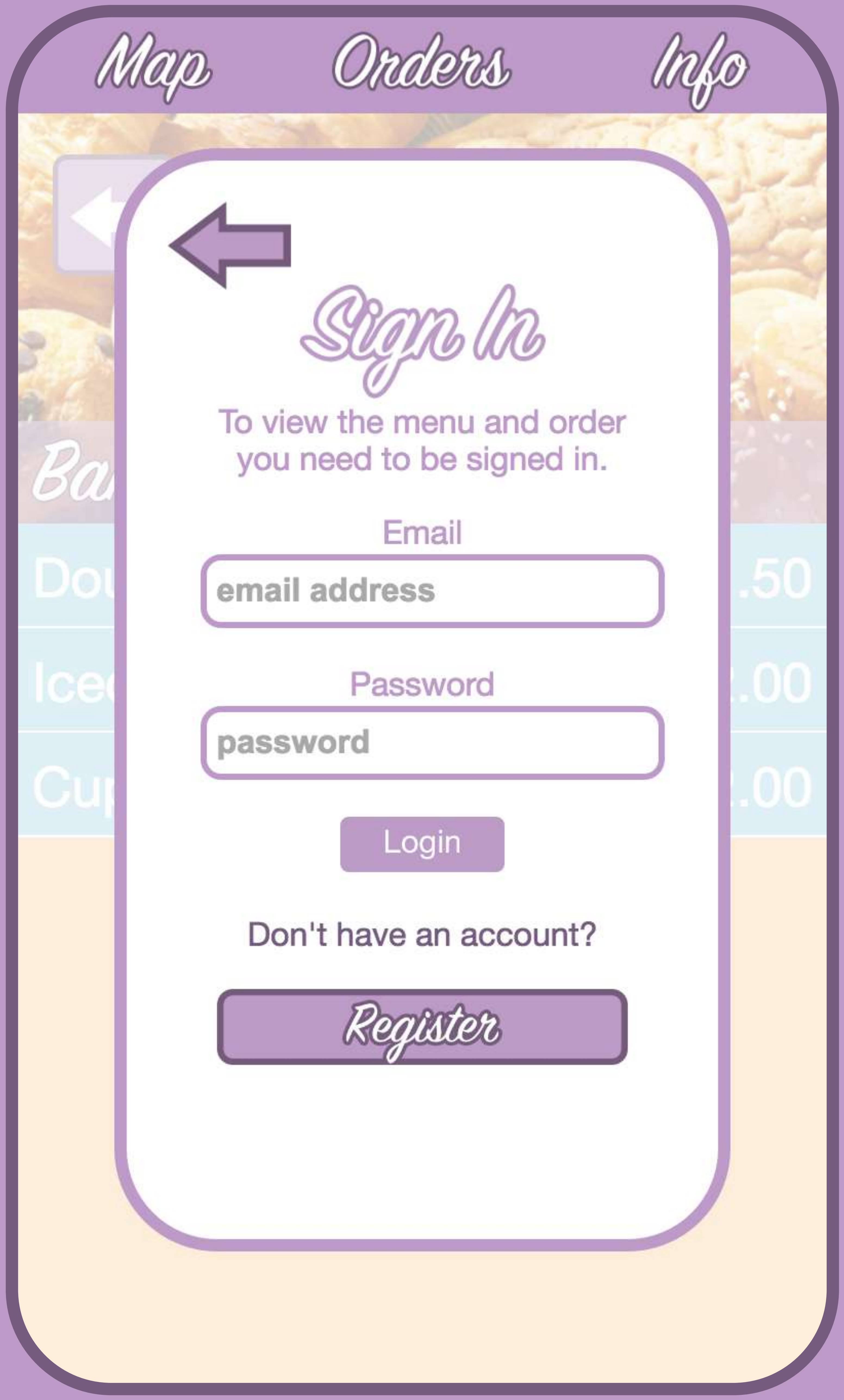
Bakery

Doughnut	£1.50
Iced Bun	£2.00
Cupcake	£2.00



Cold Drinks

Lemonade	£2.00
Iced Coffee	£2.50



Sign In

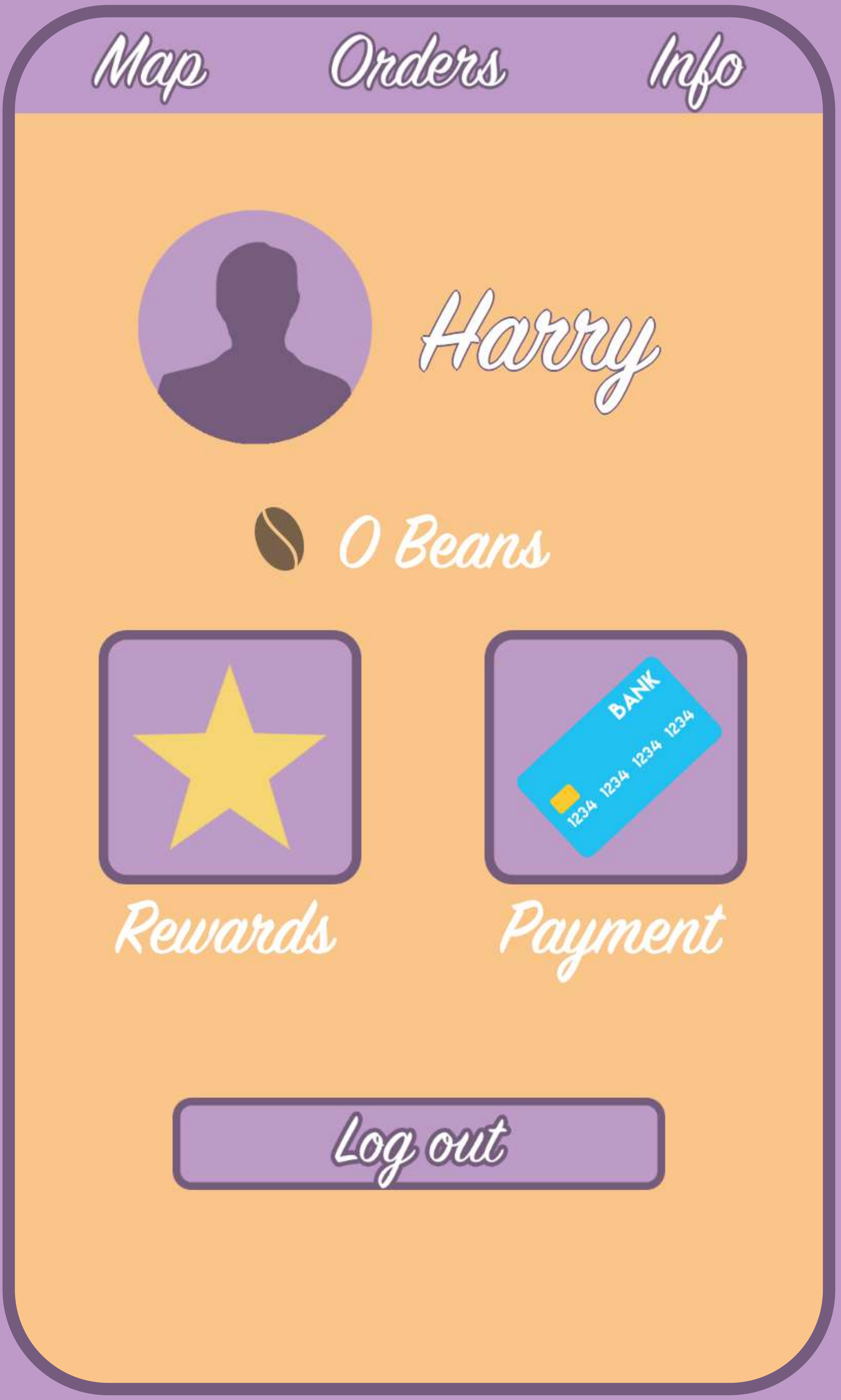
To view the menu and order you need to be signed in.

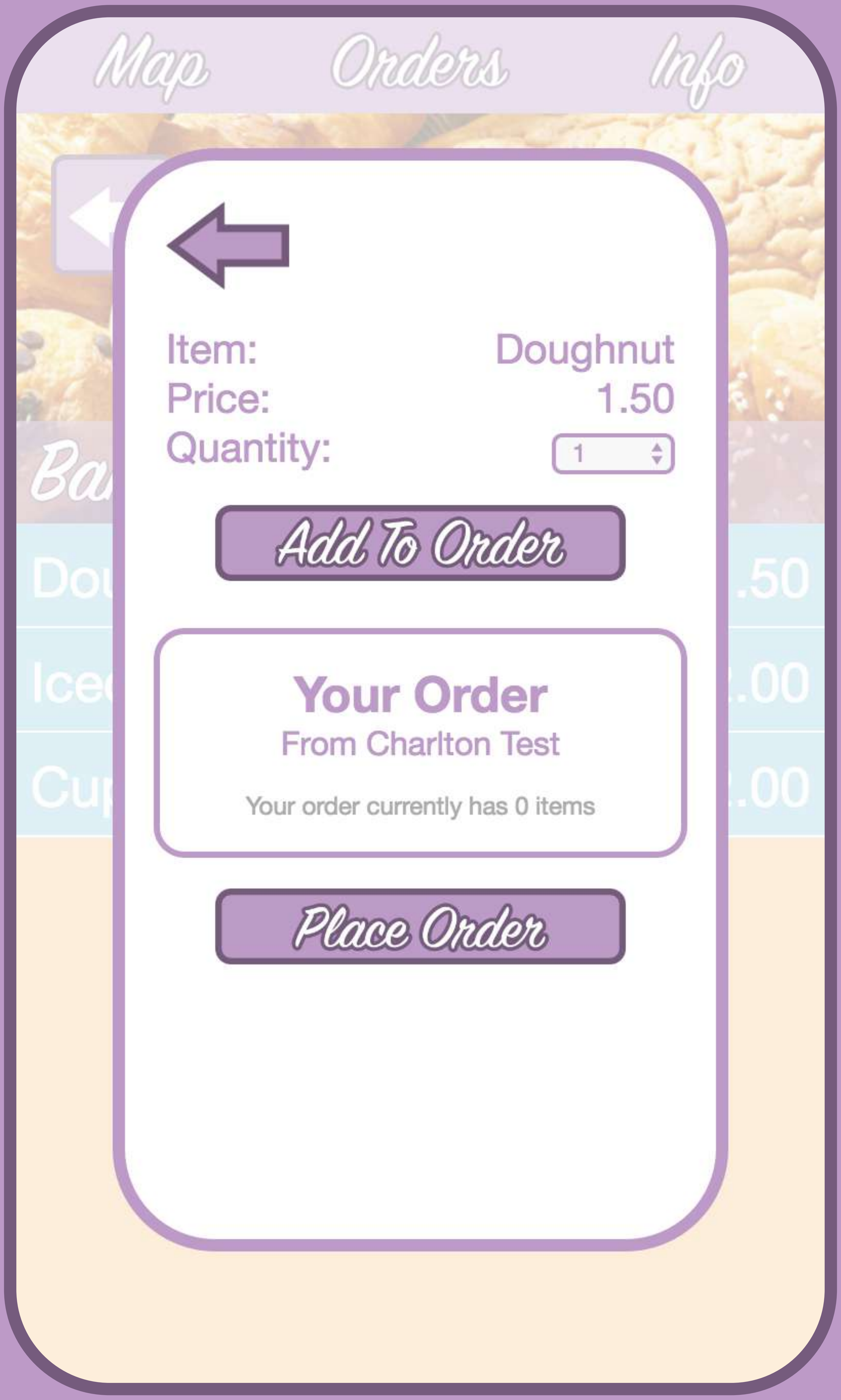
Email: email address

Password: password

Don't have an account?

Final Screenshots

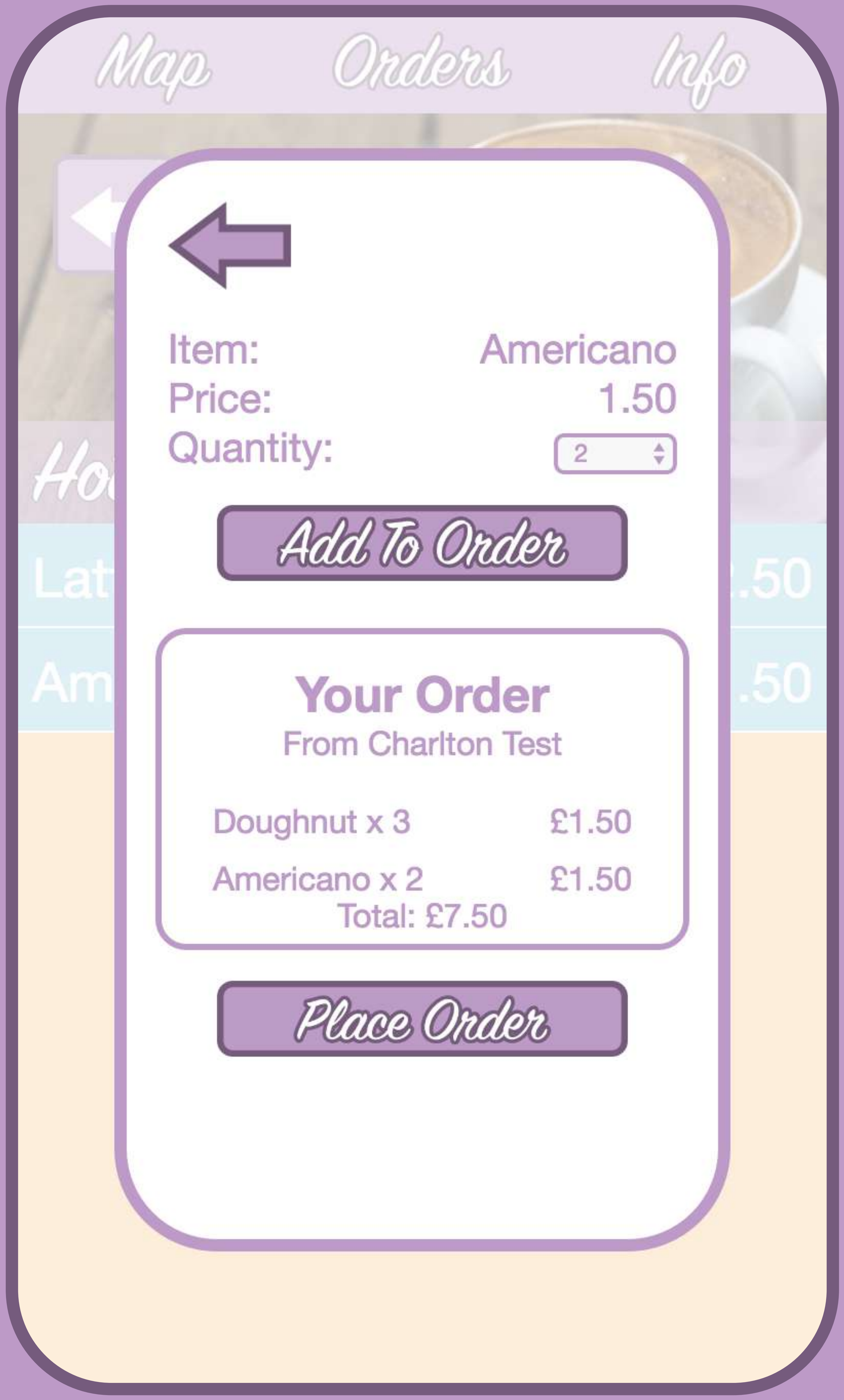




Item: Doughnut
Price: 1.50
Quantity: 1

Your Order
From Charlton Test

Your order currently has 0 items



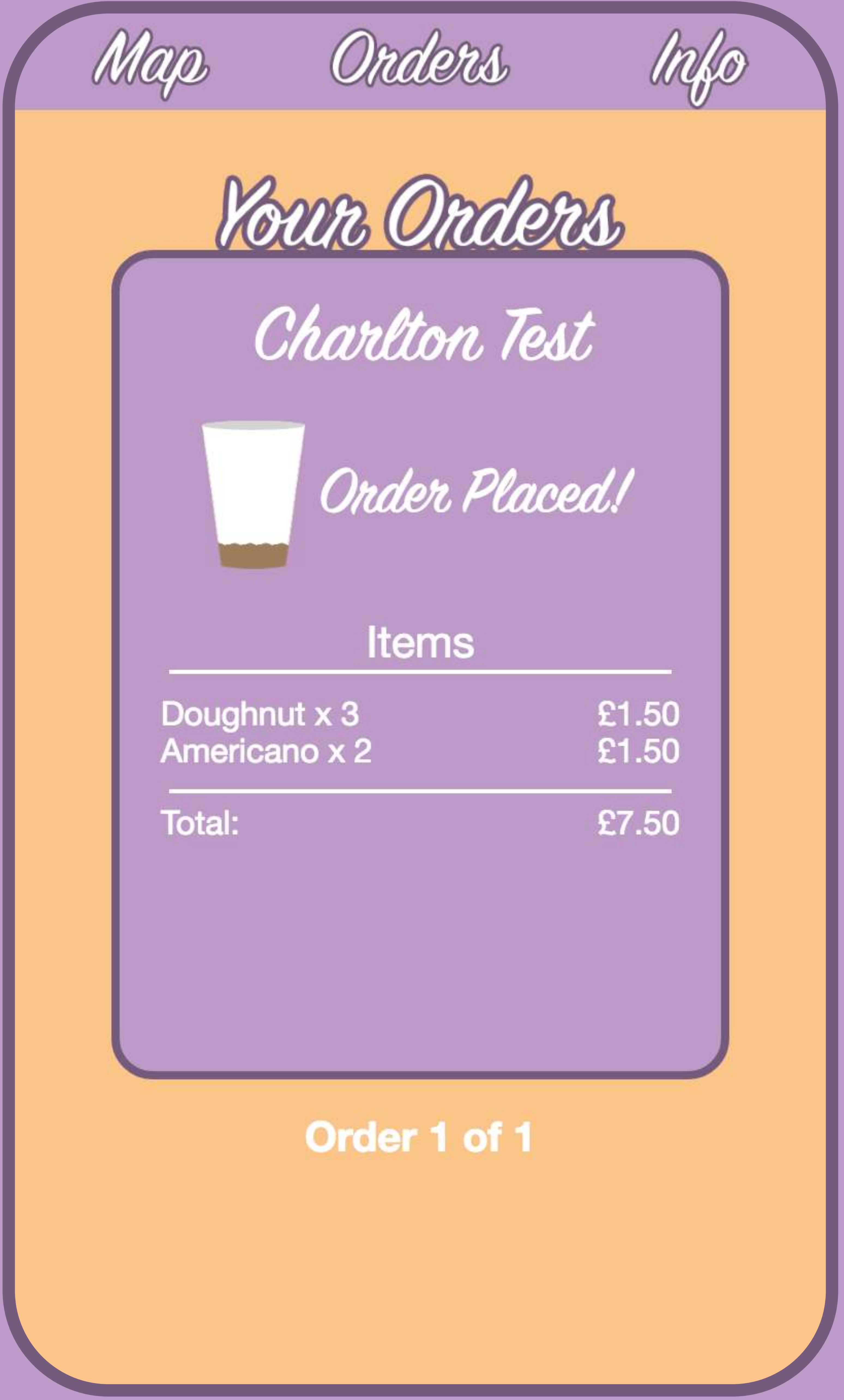
Item: Americano
Price: 1.50
Quantity: 2

Your Order
From Charlton Test

Doughnut x 3	£1.50
Americano x 2	£1.50

Total: £7.50

Final Screenshots



Your Orders

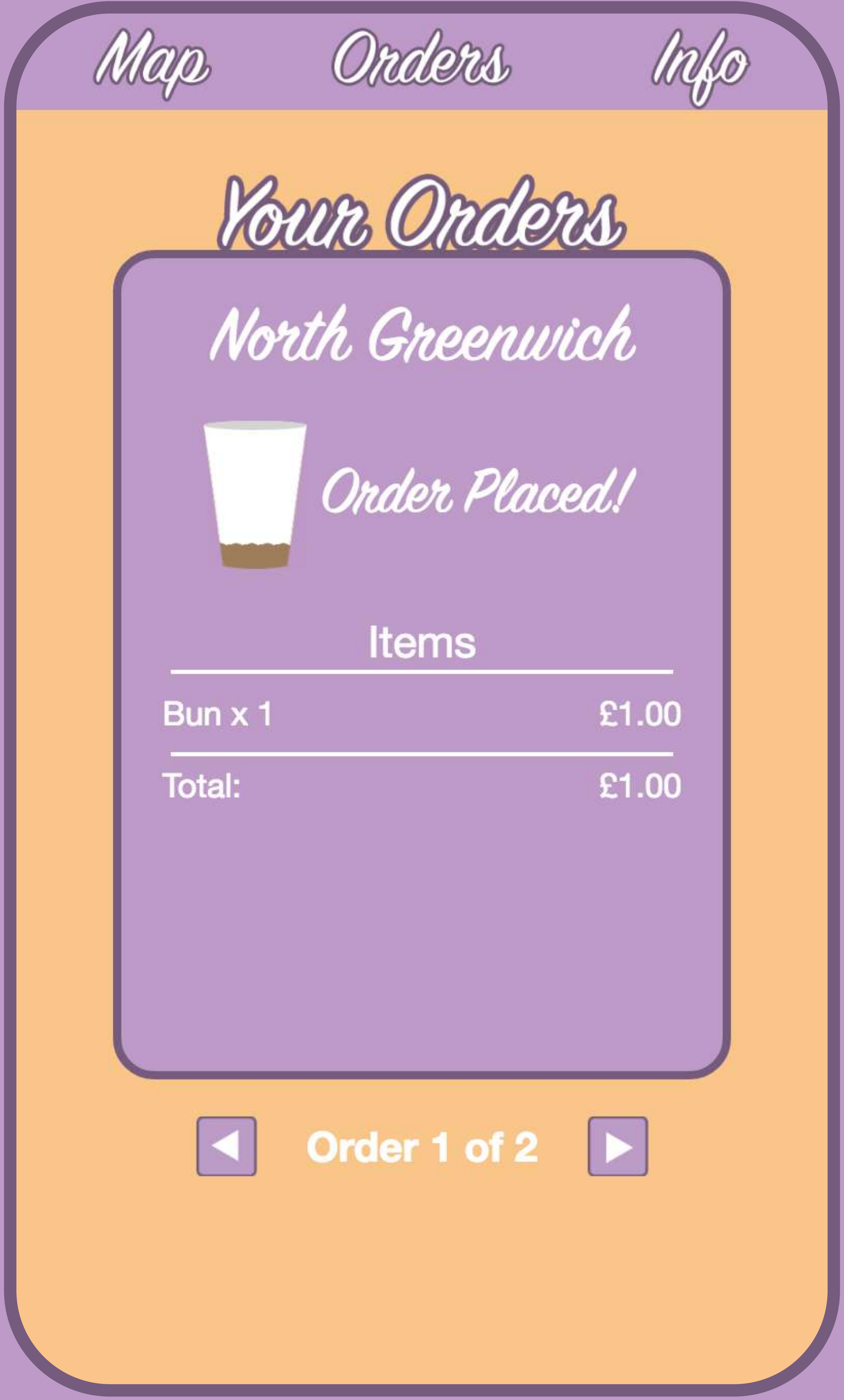
Charlton Test

Order Placed!

Items

Doughnut x 3	£1.50
Americano x 2	£1.50
Total:	£7.50

Order 1 of 1



Your Orders

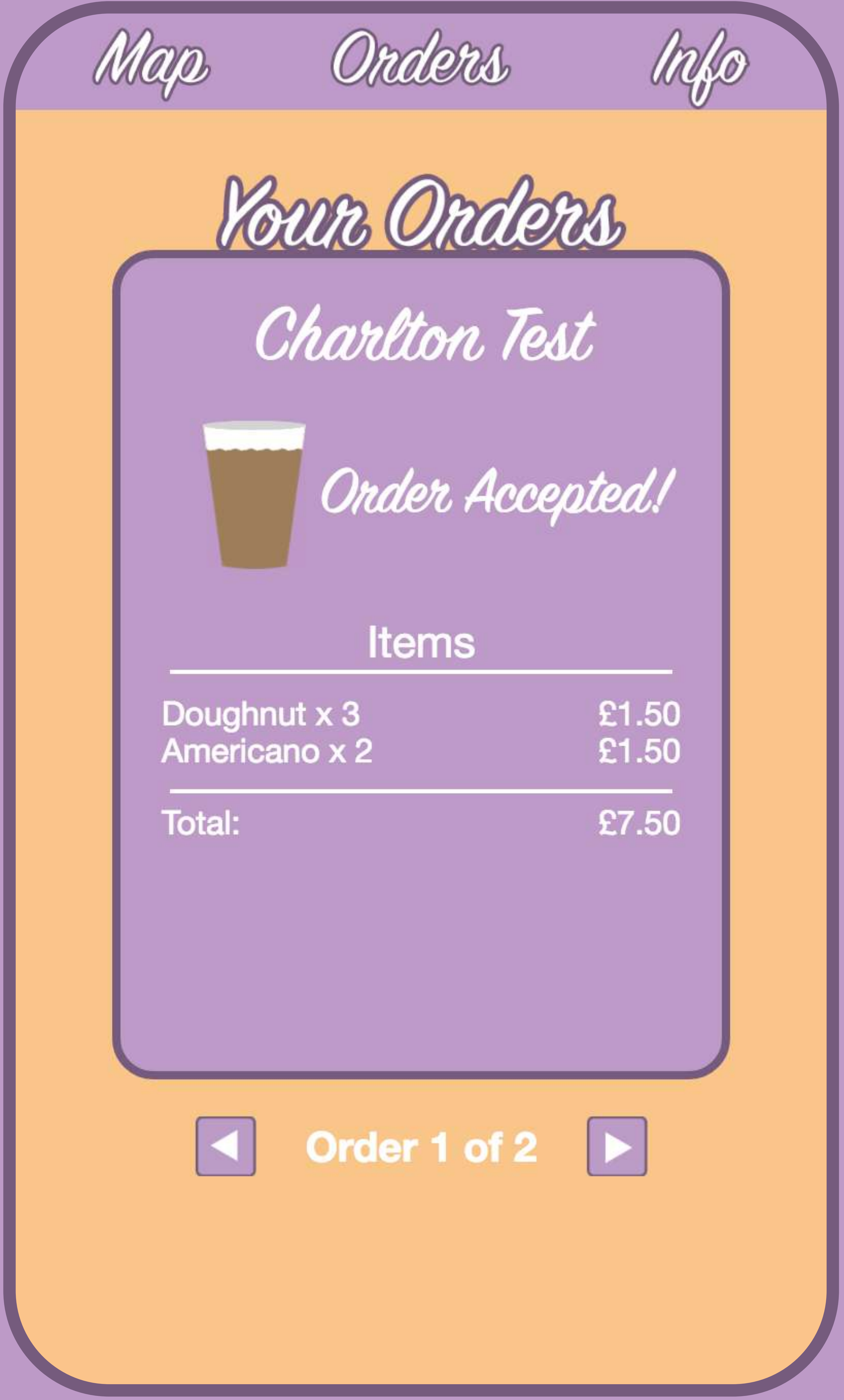
North Greenwich

Order Placed!

Items

Bun x 1	£1.00
Total:	£1.00

◀ Order 1 of 2 ▶



Your Orders

Charlton Test

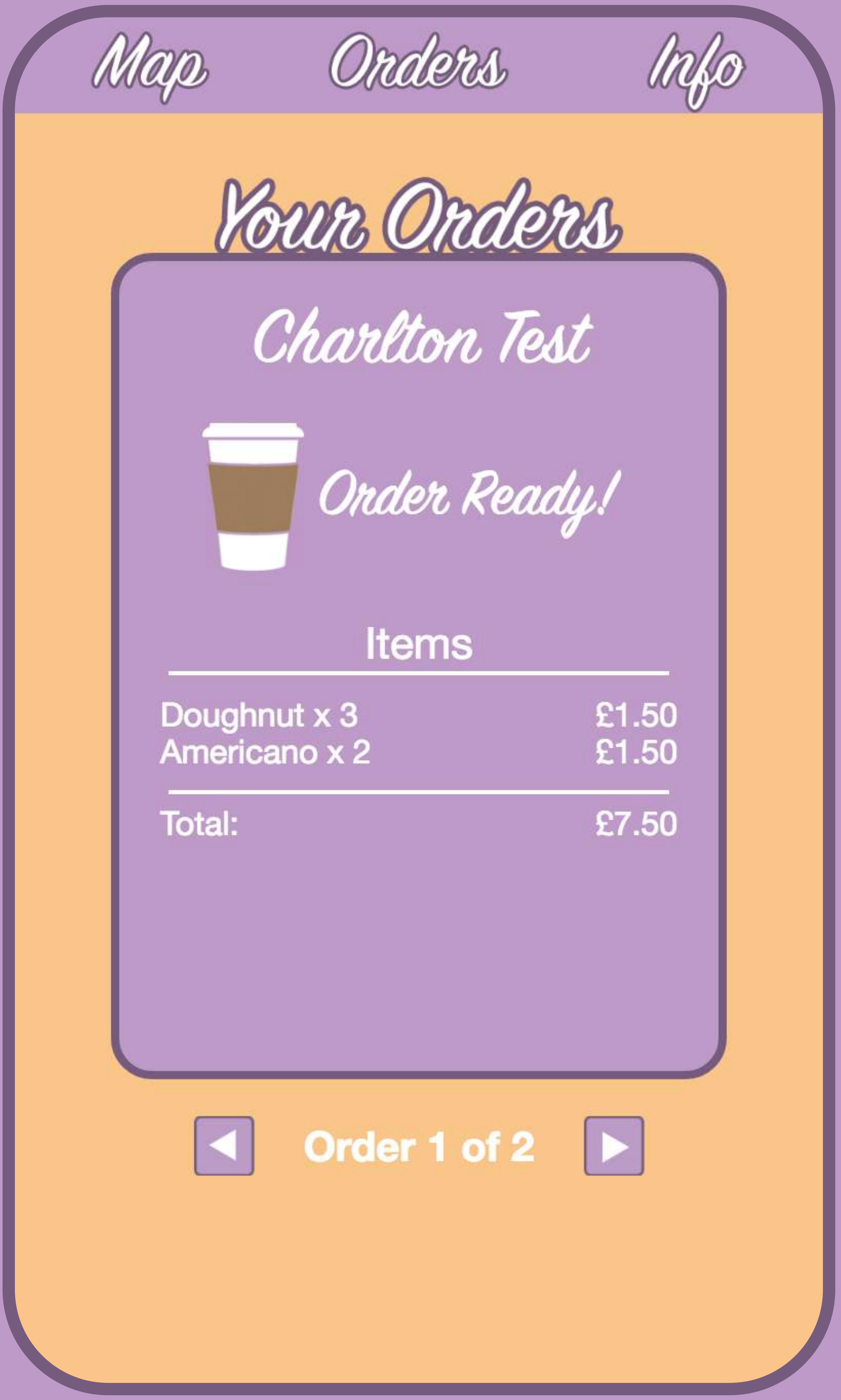
Order Accepted!

Items

Doughnut x 3	£1.50
Americano x 2	£1.50
Total:	£7.50

◀ Order 1 of 2 ▶

Final Screenshots



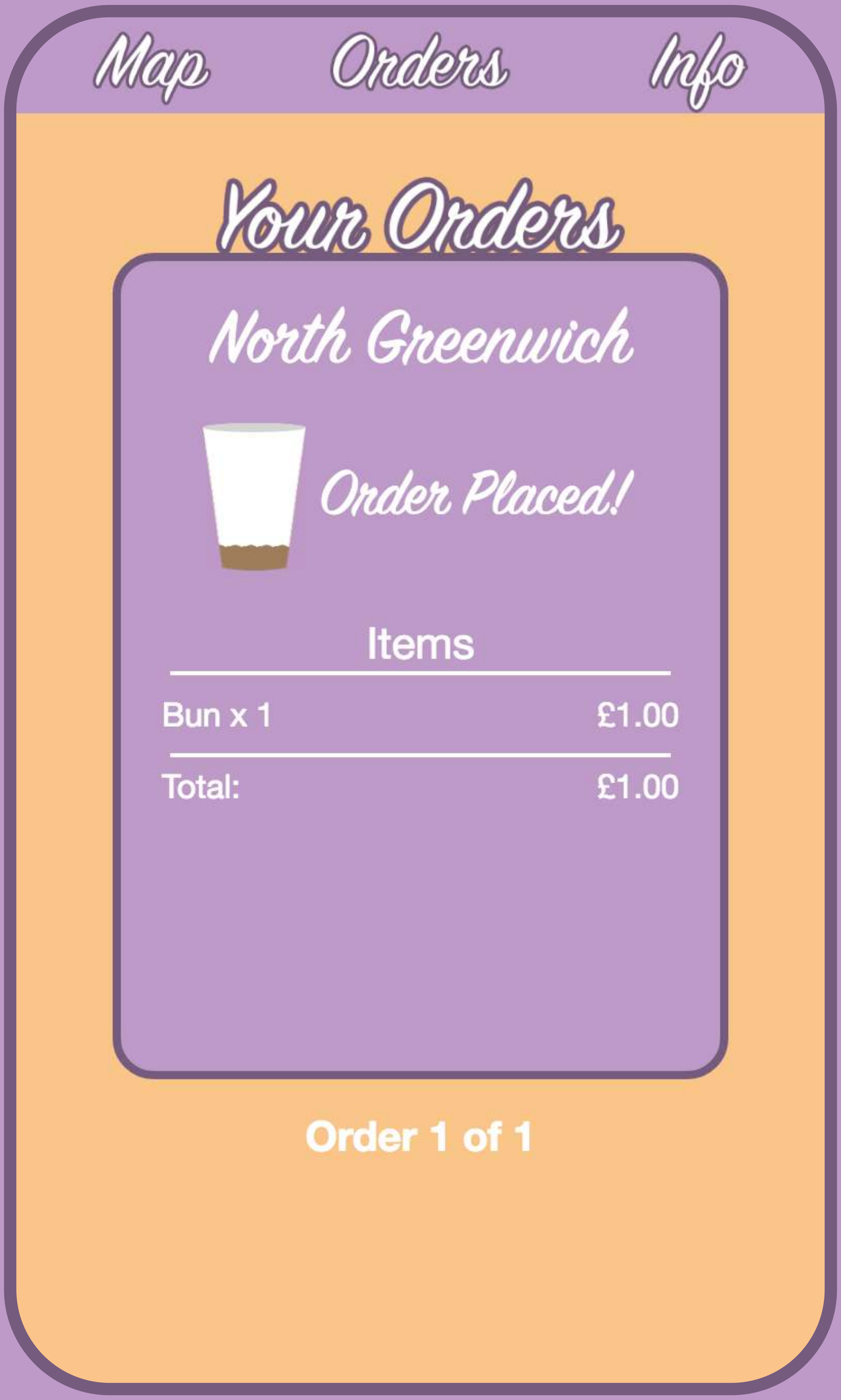
Your Orders

Charlton Test

Order Ready!

Items

Doughnut x 3	£1.50
Americano x 2	£1.50
Total:	£7.50



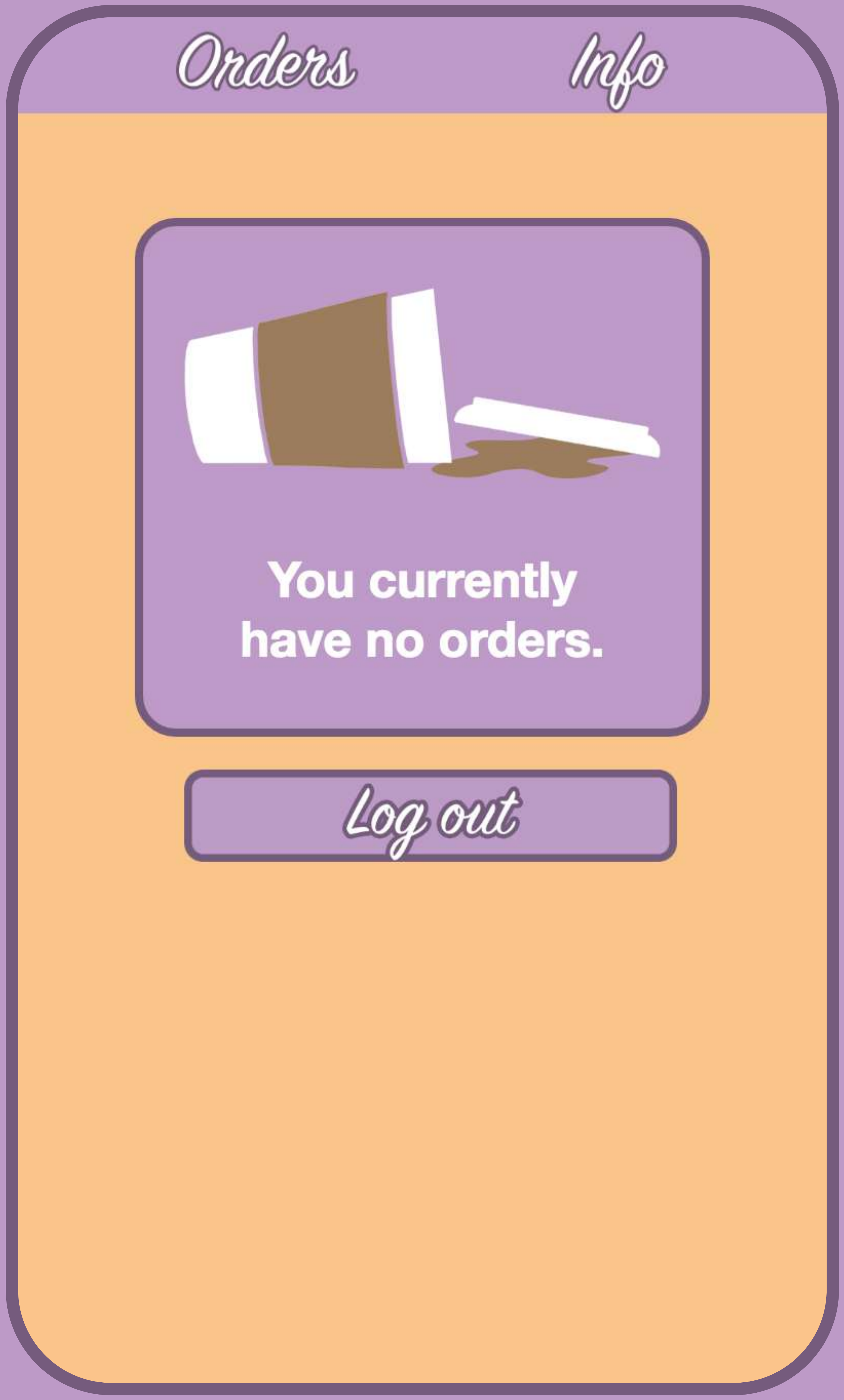
Your Orders

North Greenwich

Order Placed!

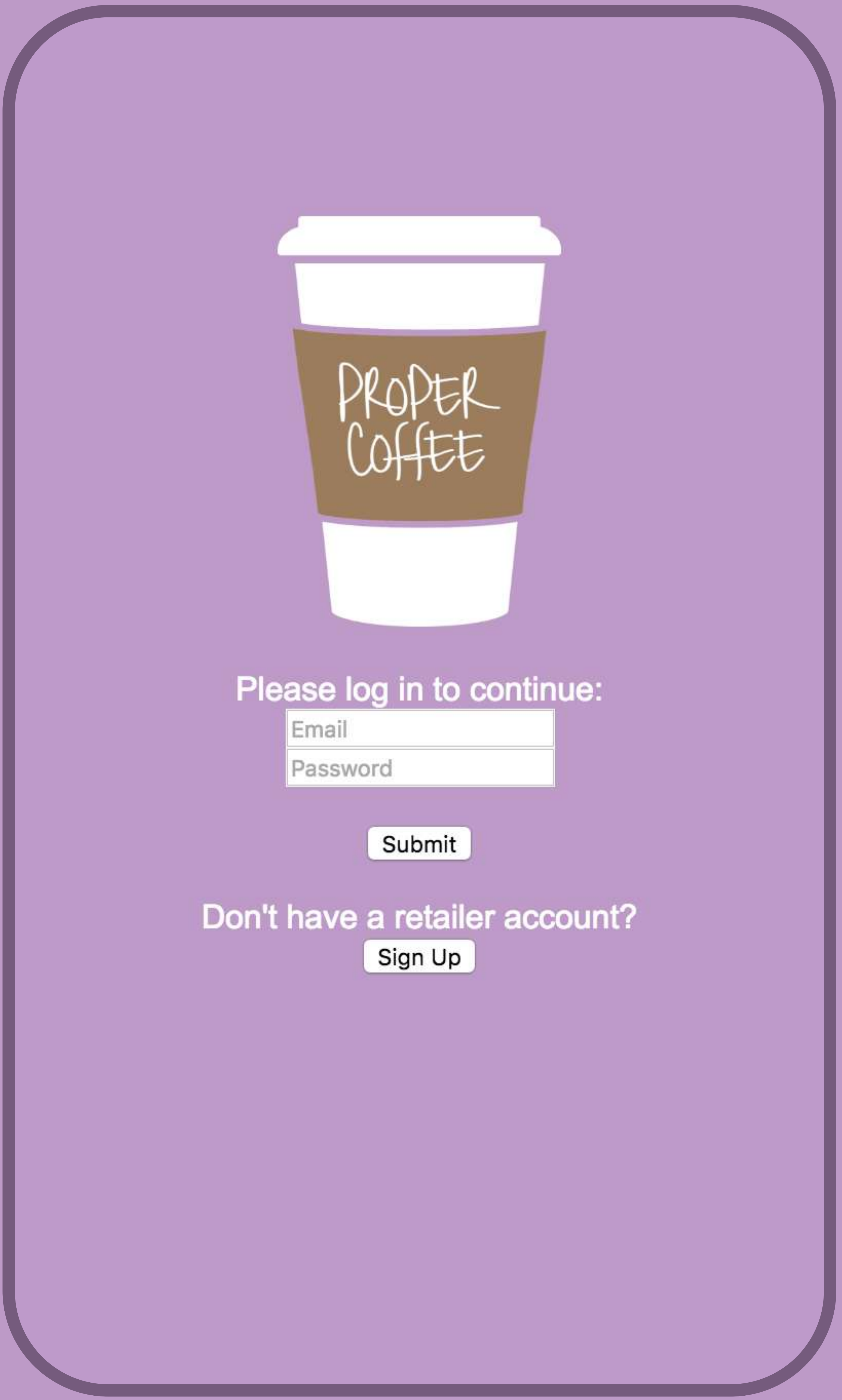
Items

Bun x 1	£1.00
Total:	£1.00



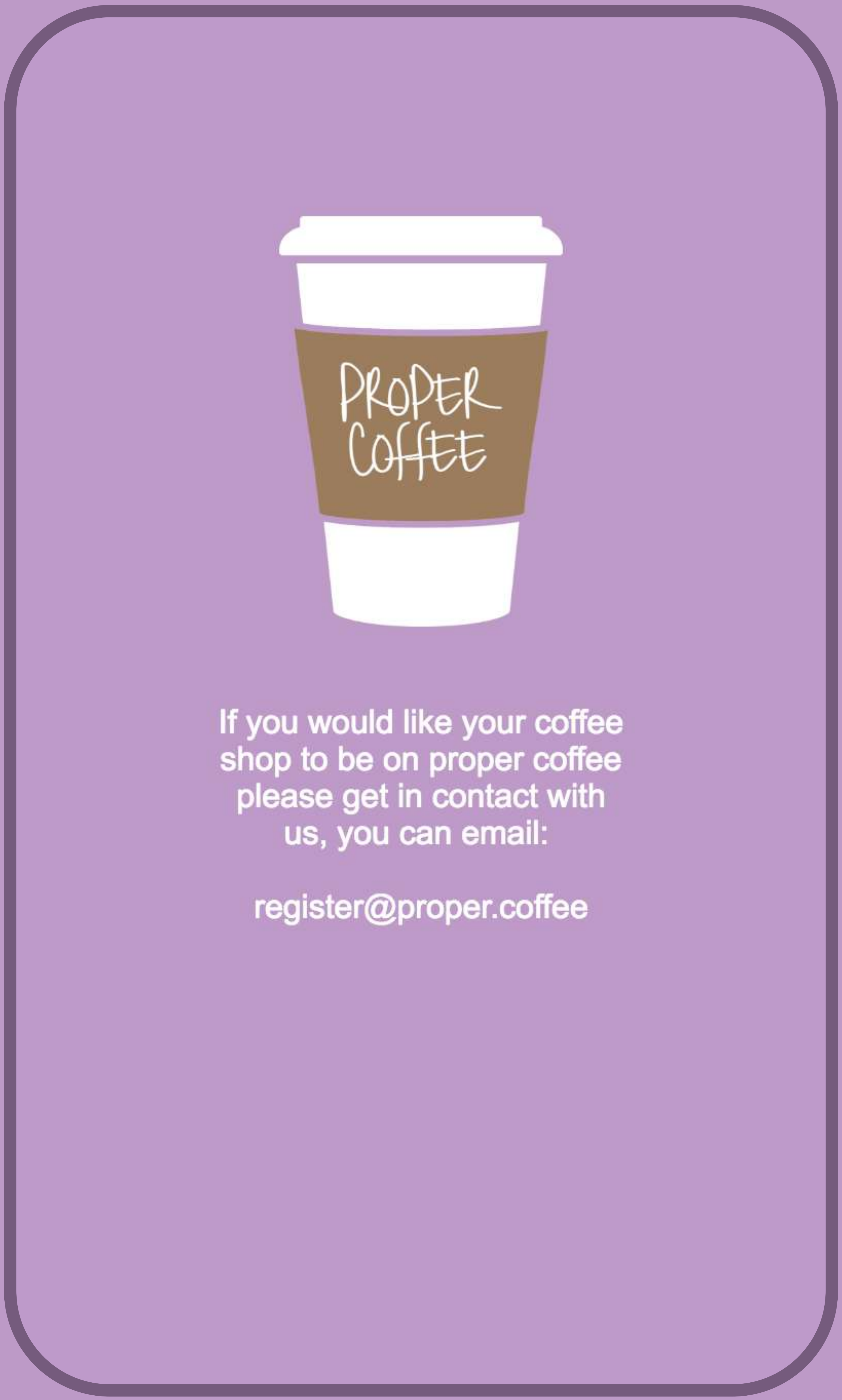
You currently have no orders.

Final Screenshots



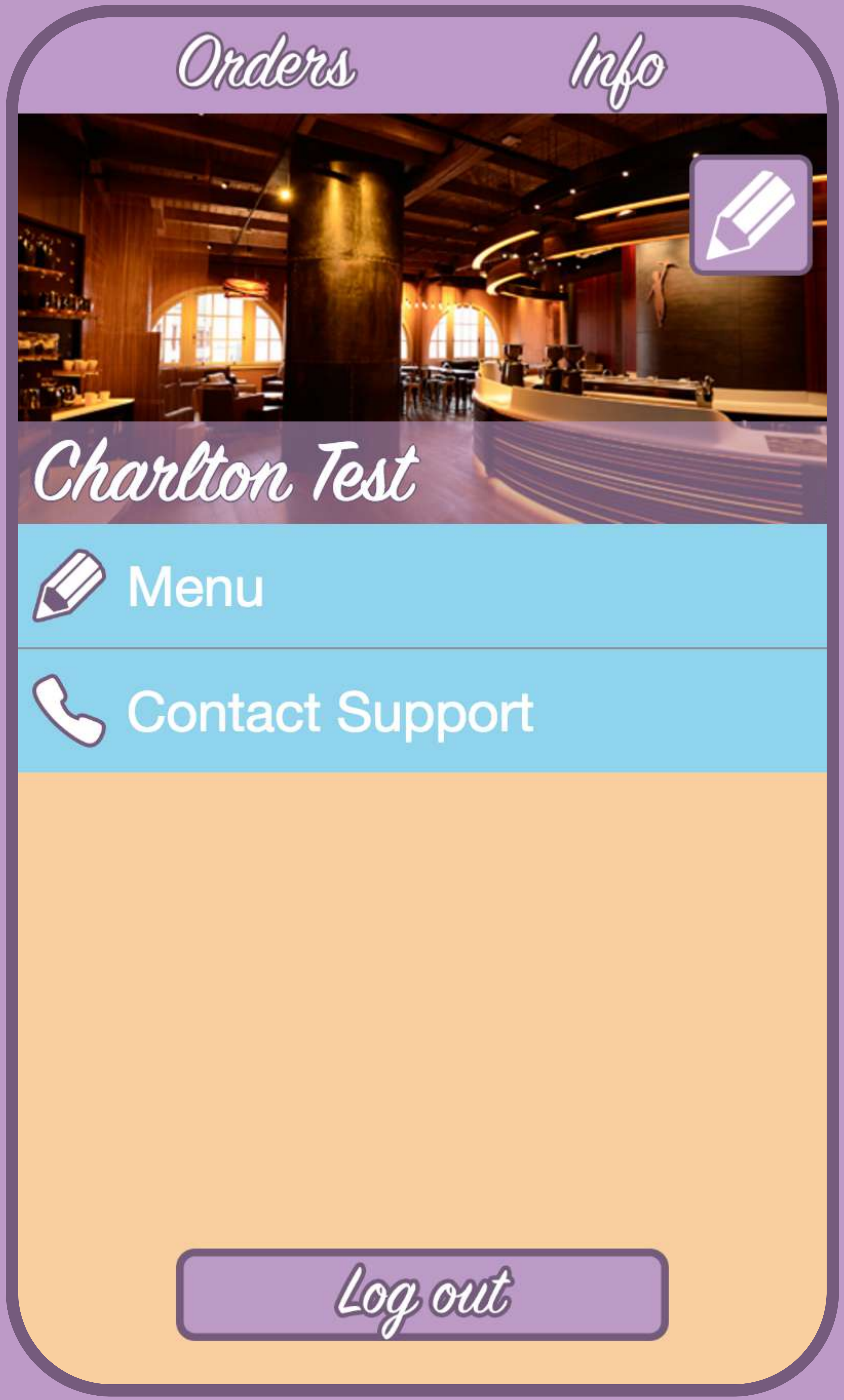
Please log in to continue:

Don't have a retailer account?

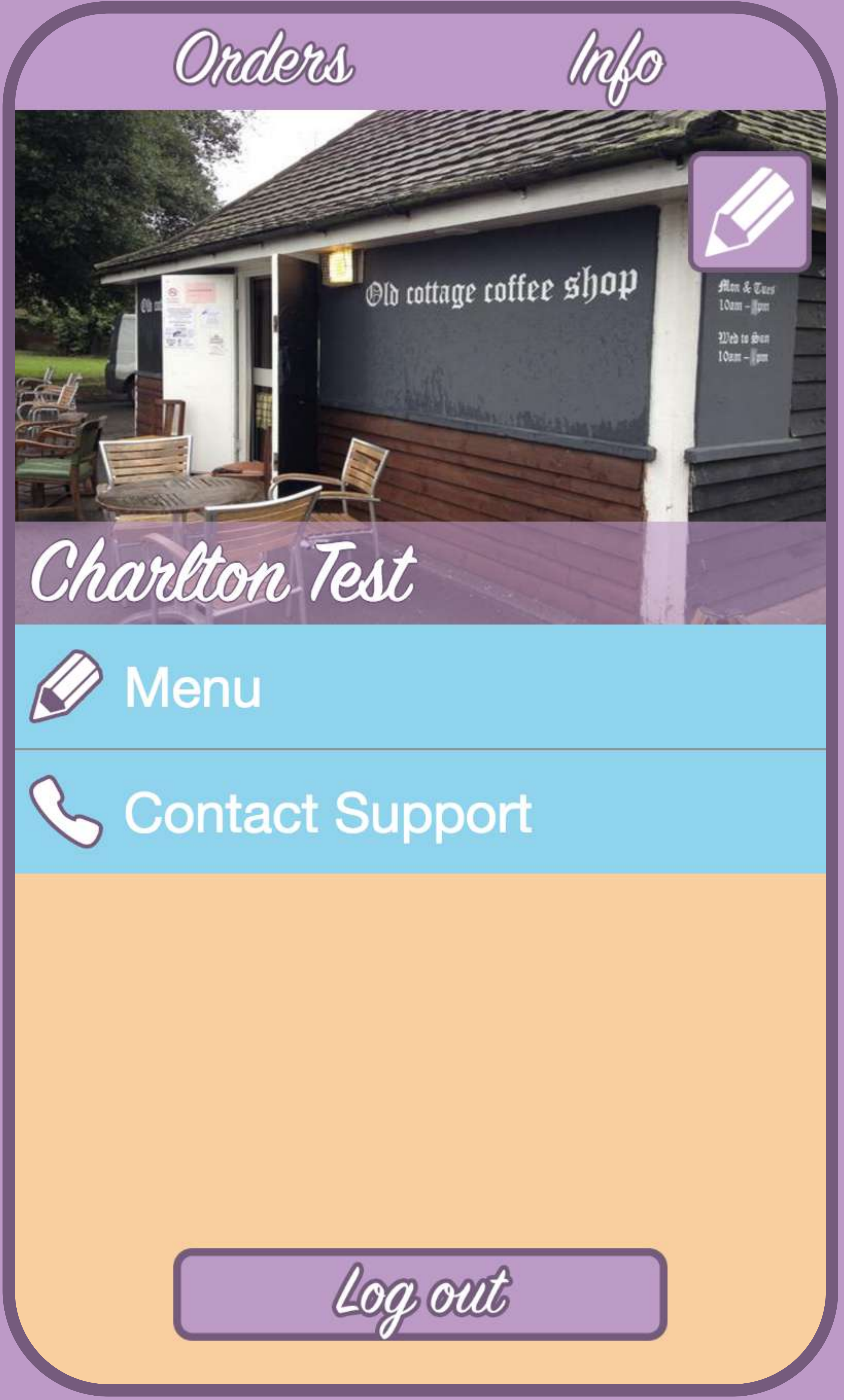


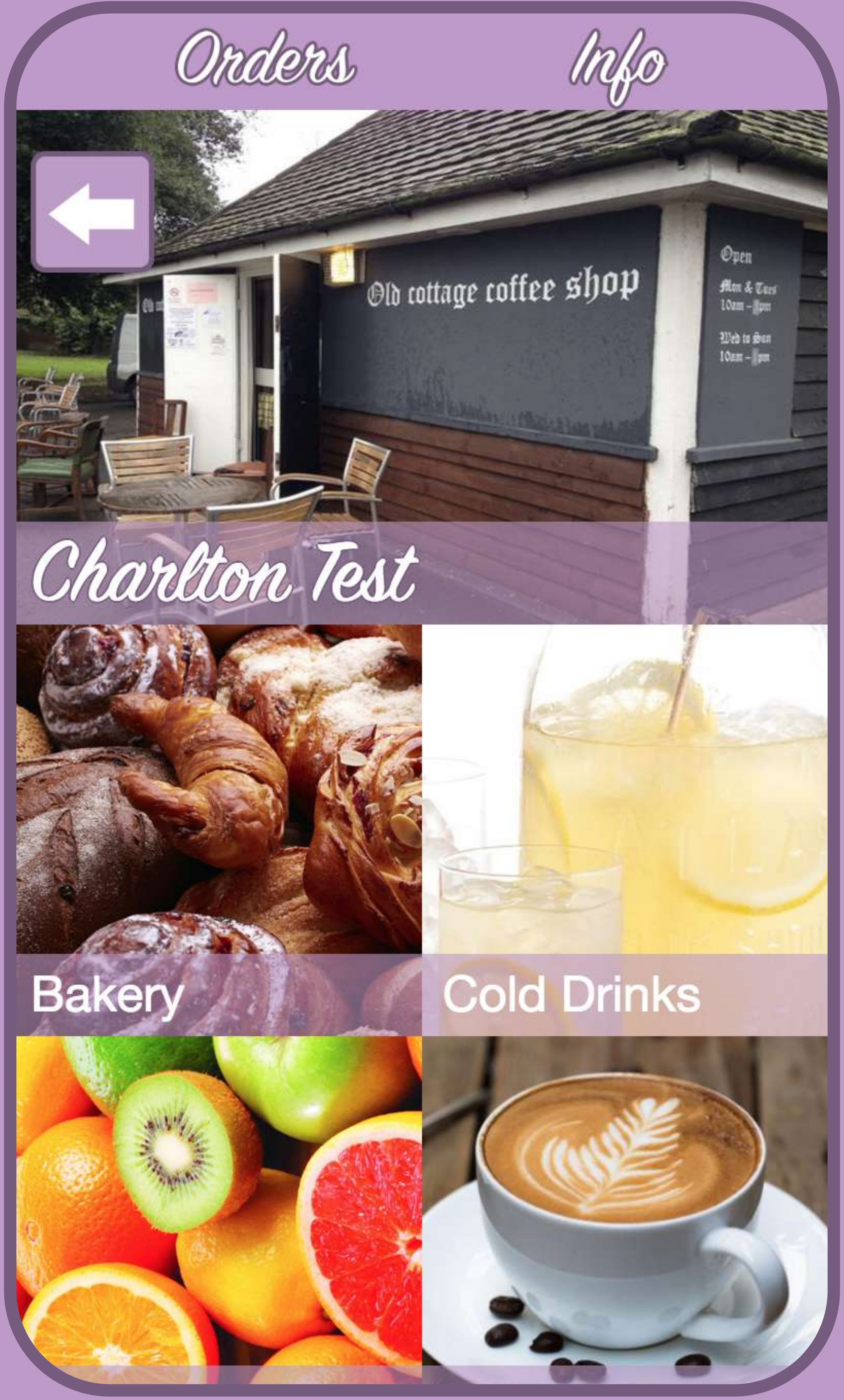
If you would like your coffee shop to be on proper coffee please get in contact with us, you can email:

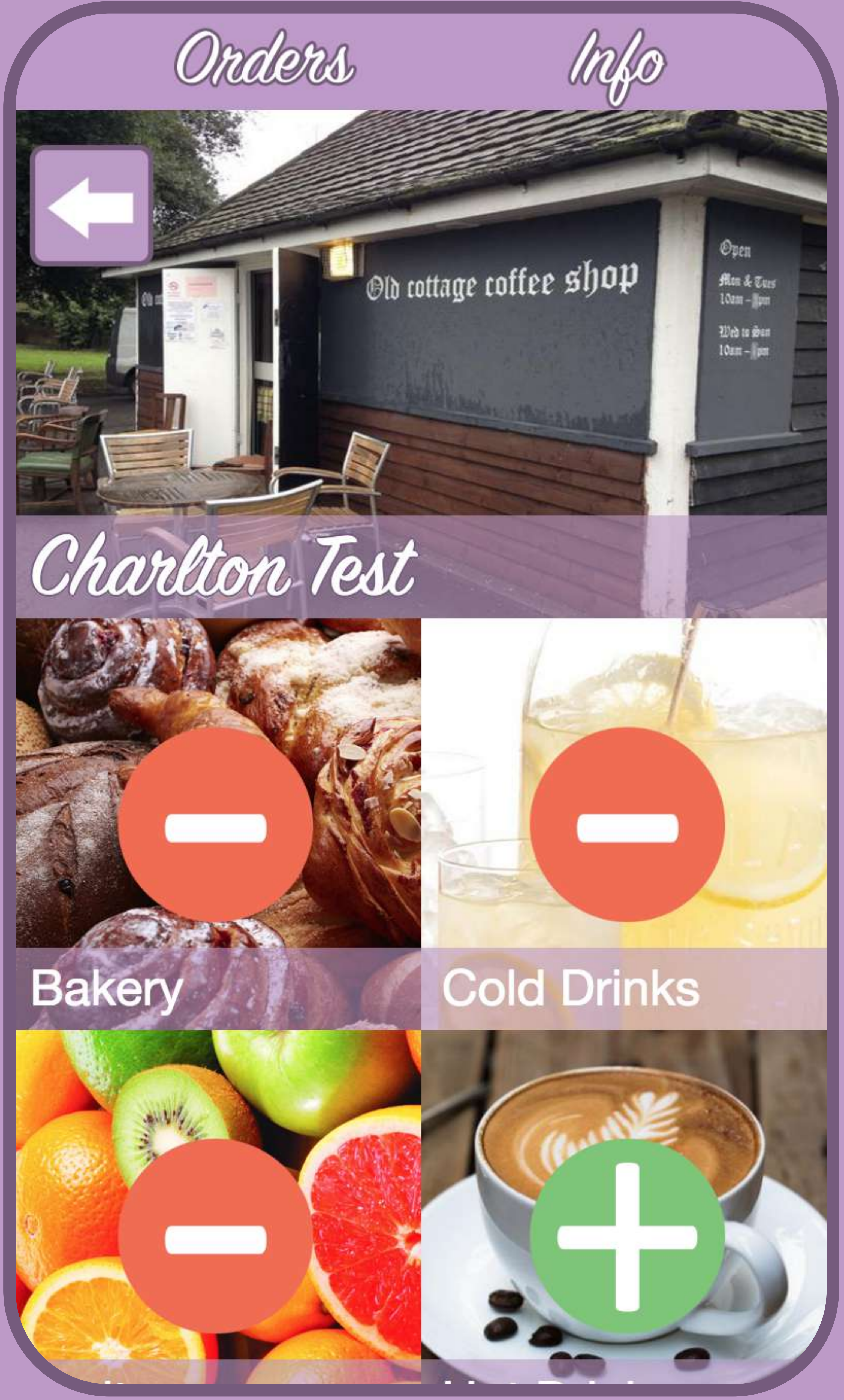
register@proper.coffee



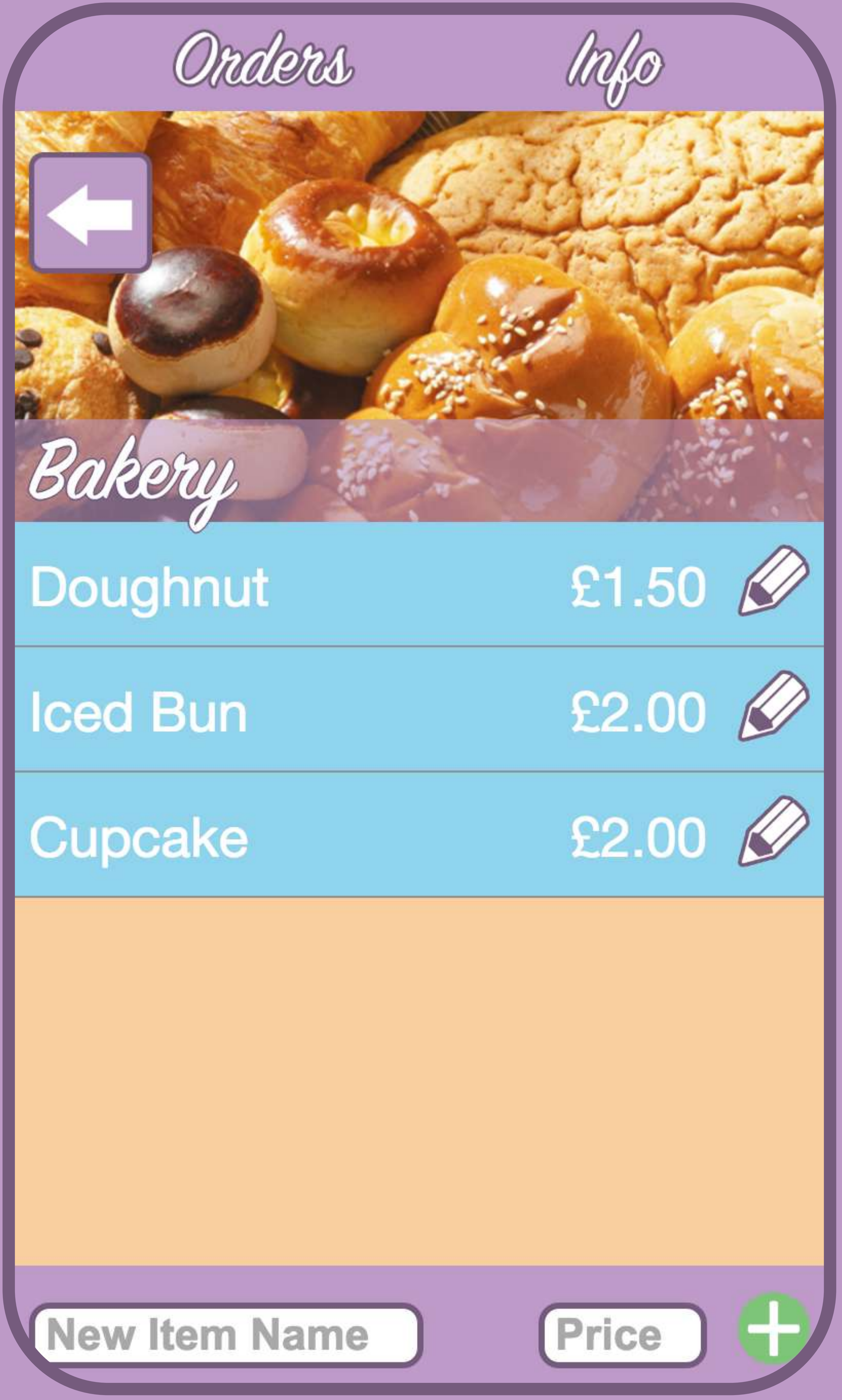
Final Screenshots







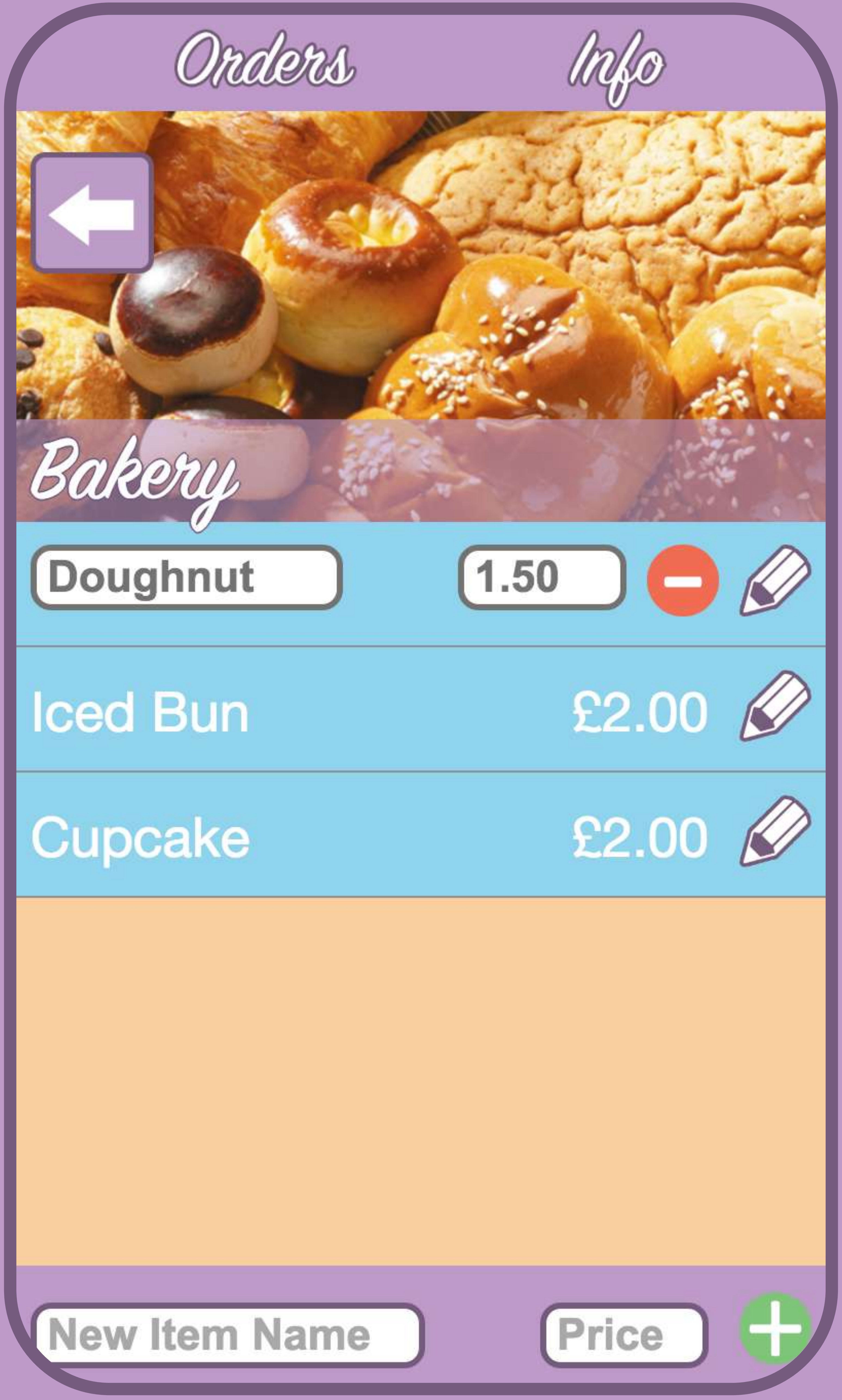
Final Screenshots



Doughnut £1.50

Iced Bun £2.00

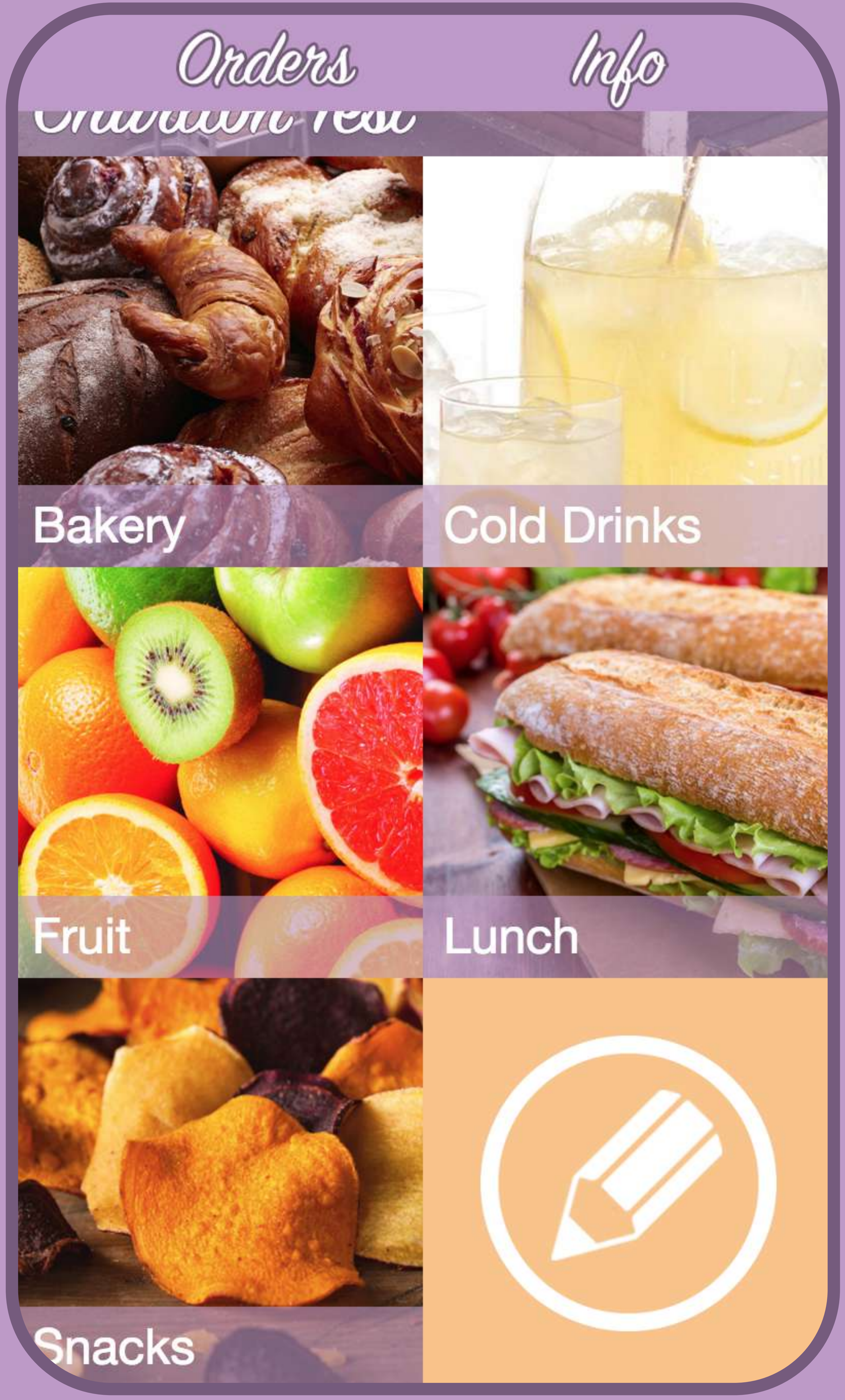
Cupcake £2.00



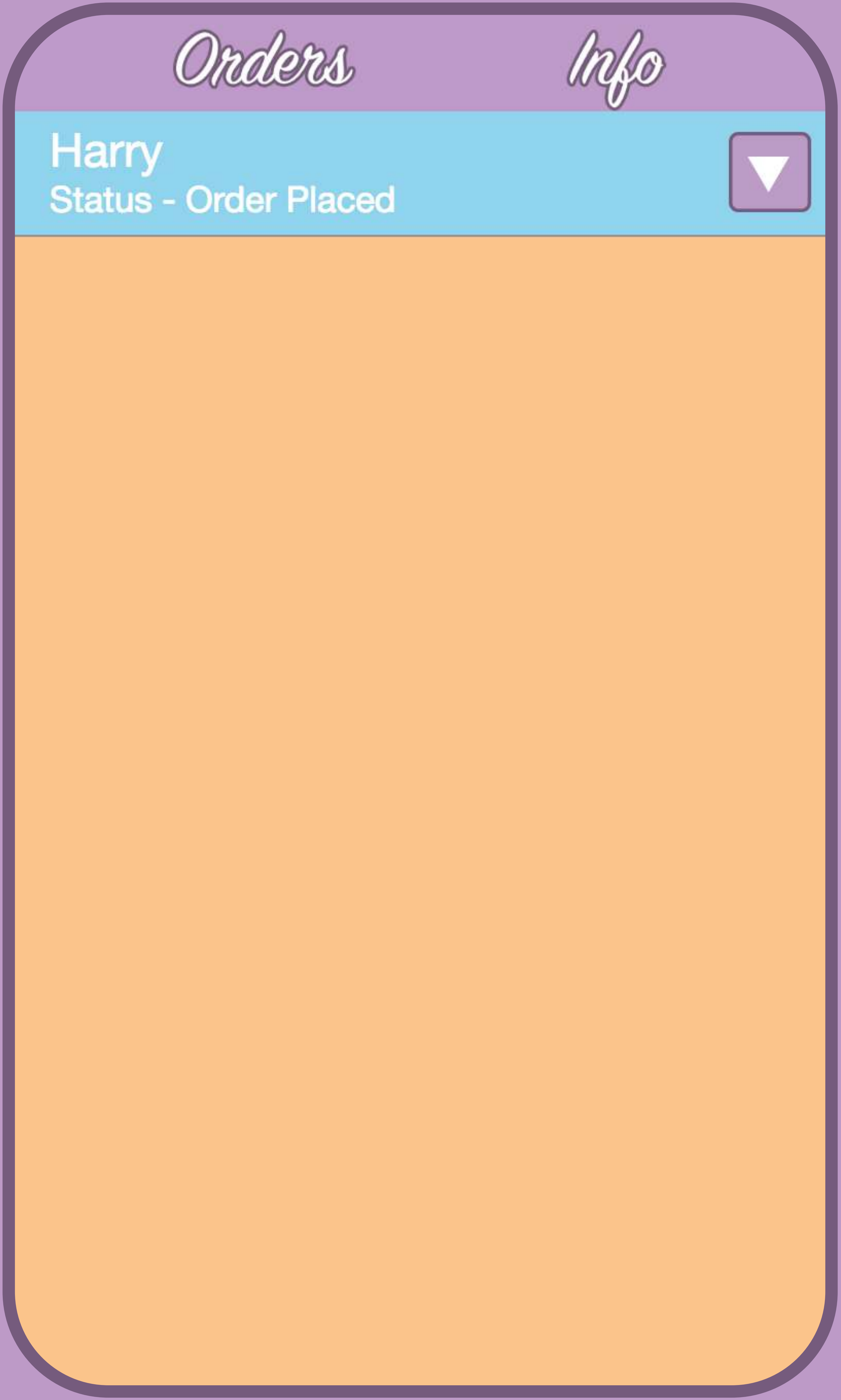
Doughnut 1.50

Iced Bun £2.00

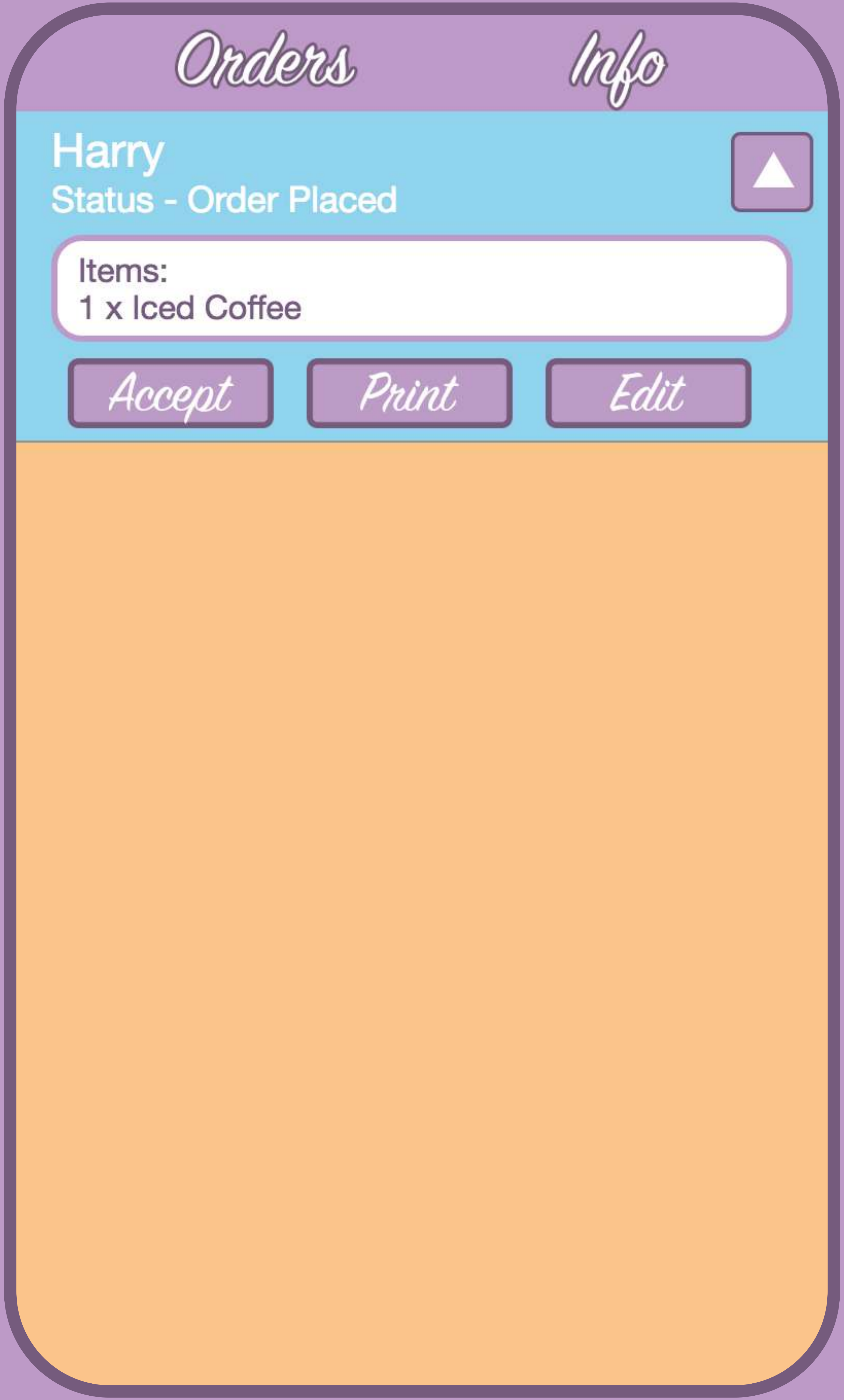
Cupcake £2.00



Final Screenshots

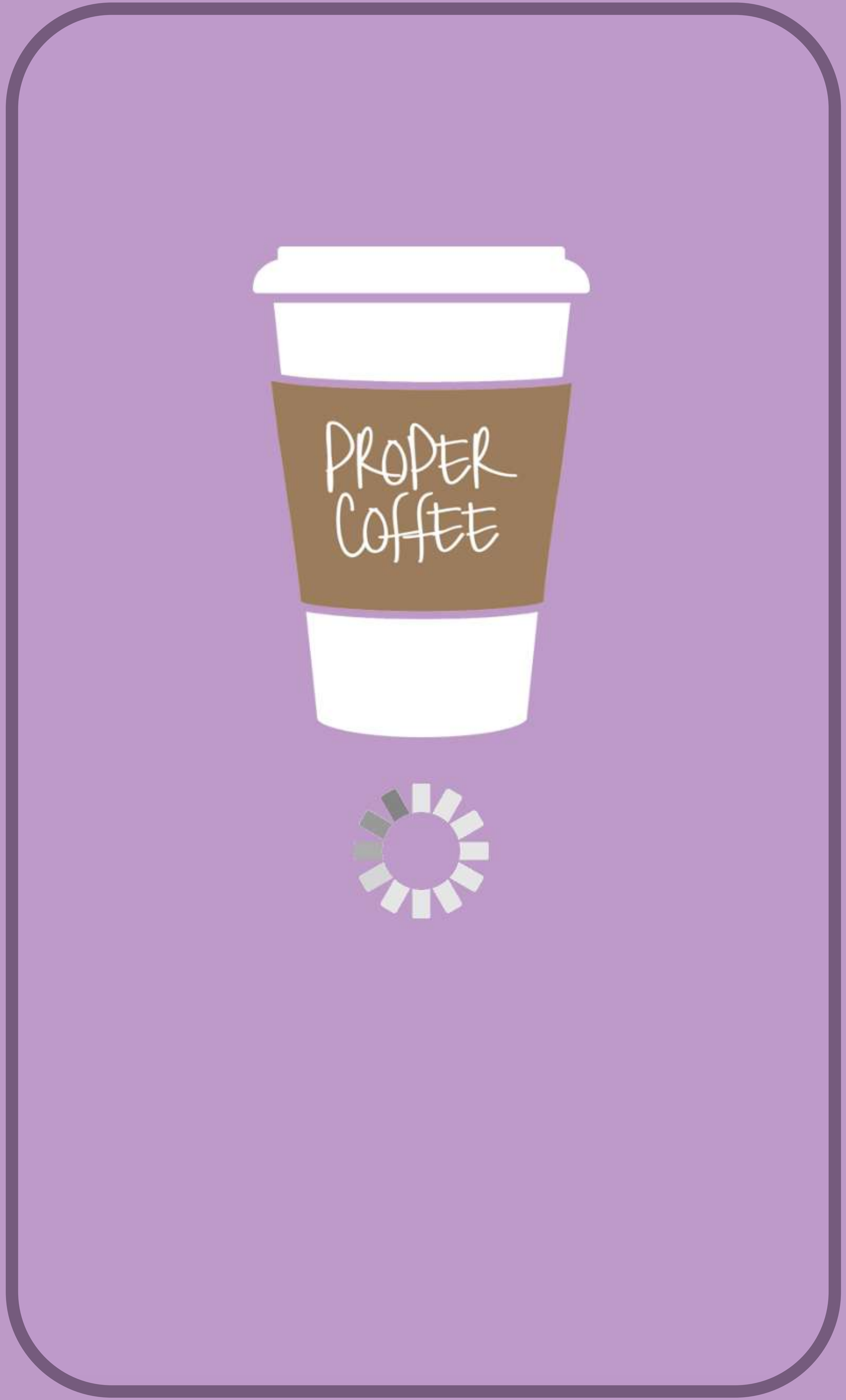


Harry
Status - Order Placed



Harry
Status - Order Placed

Items:
1 x Iced Coffee


PROPER
COFFEE