

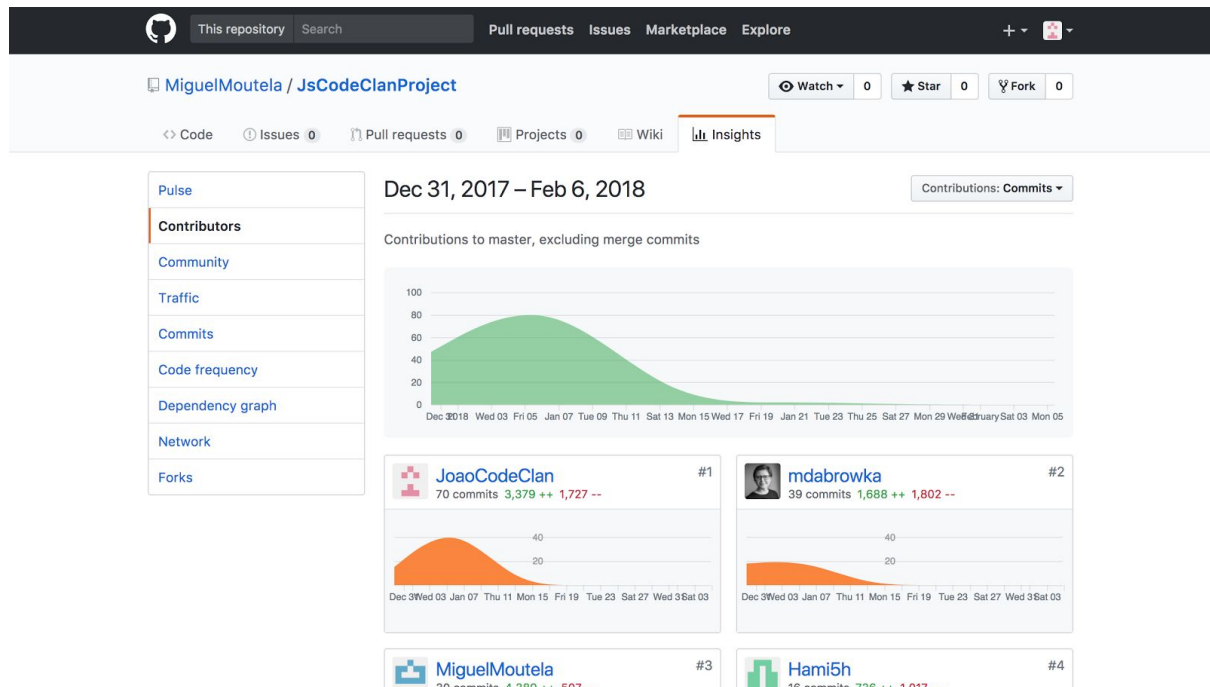
Evidence for Project Unit

Your name here: Joao Nequinha

Your Cohort: E16

Date here: 05.02.2018

P- 1 Github Contributors page



P- 2 Project Brief

Users looking to attend events around them or in a city of their choice are to be able to view relevant events in a table. Use an existing API or create a new API to display information about times, location and content of events.

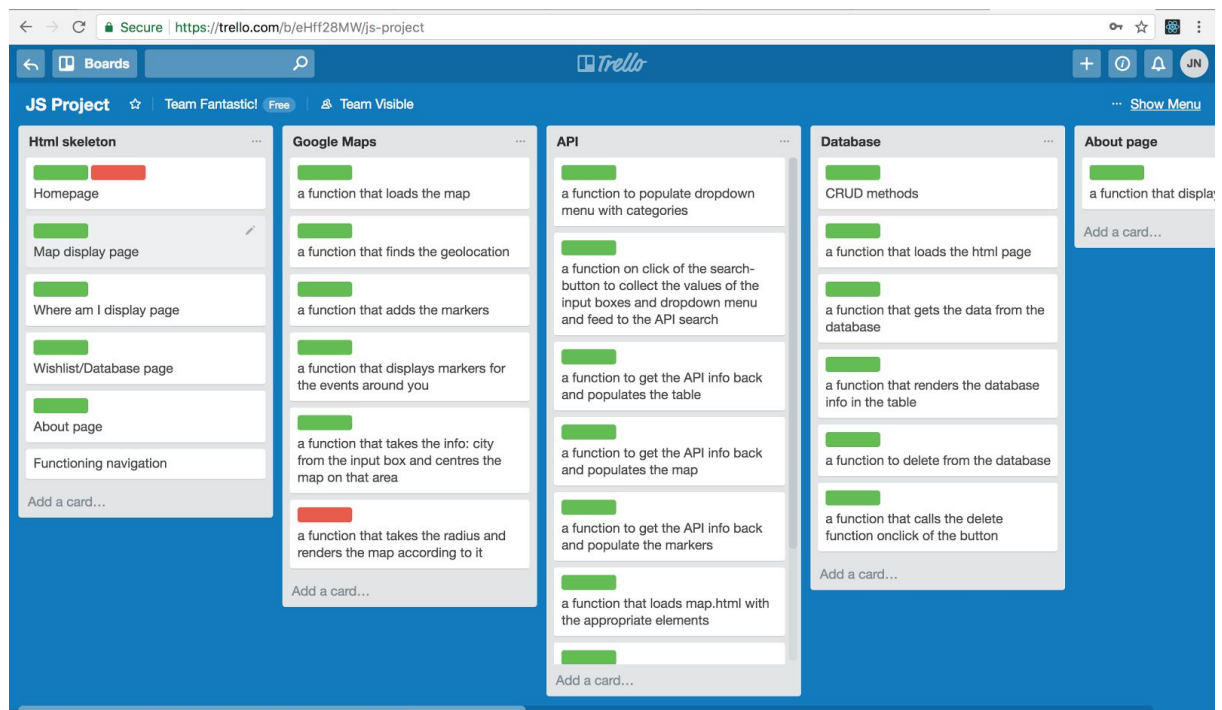
APIs used:

- Eventful API and Google Maps API

MVP user has to be able to:

- Display events on the map
- Display info about the events in a table and on pop-up windows
- Save chosen events to the database

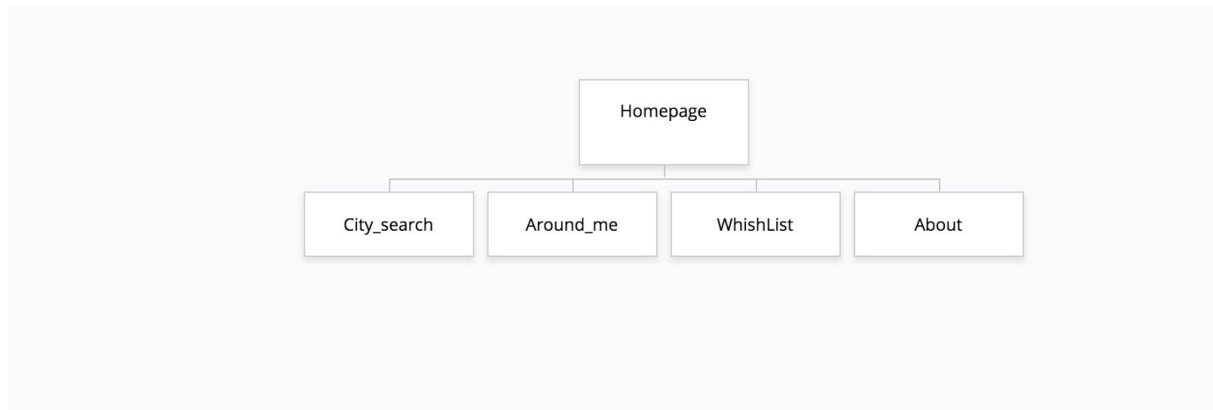
P-3 Use of Trello



P-4 Acceptance Criteria

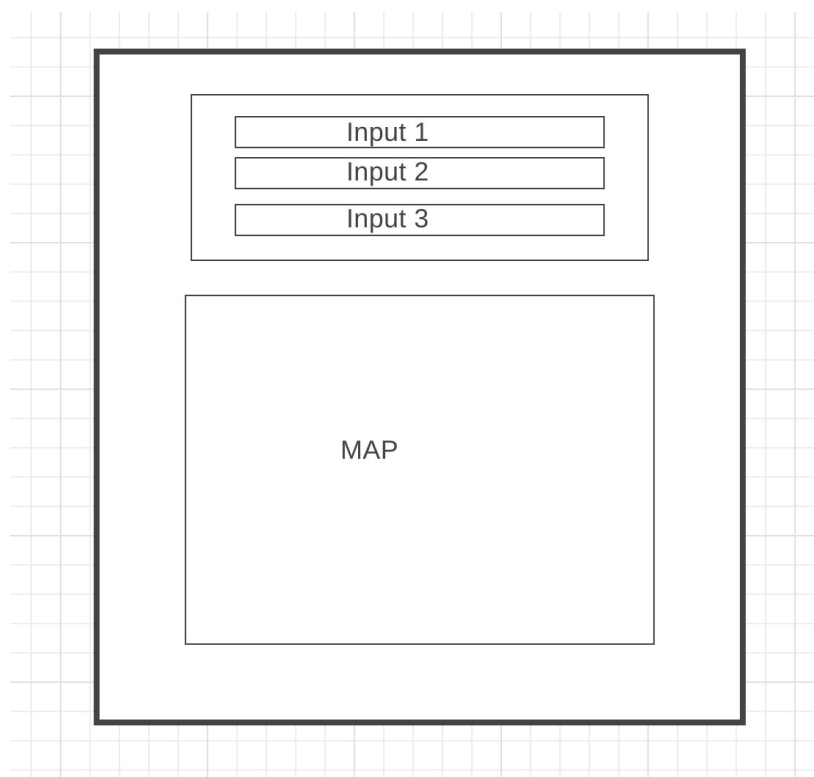
User need	Acceptance criteria	pass/fail
User is able to search events by city	Events are shown on map and table according to search criteria	PASS
User is able to get information about events	Infowindow shows on marker when clicked by user	PASS
User is able to add events to database	Event is added to the database when add button on table is pressed by user	PASS
User is able to delete events from database	Event is deleted from database when user presses delete button on table on database page	PASS

P-5 User sitemap

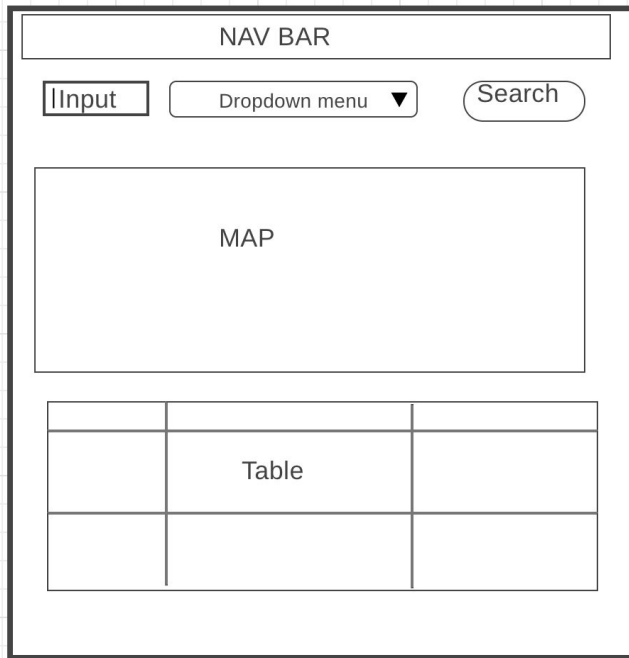


P-6 Wireframes designs

1-

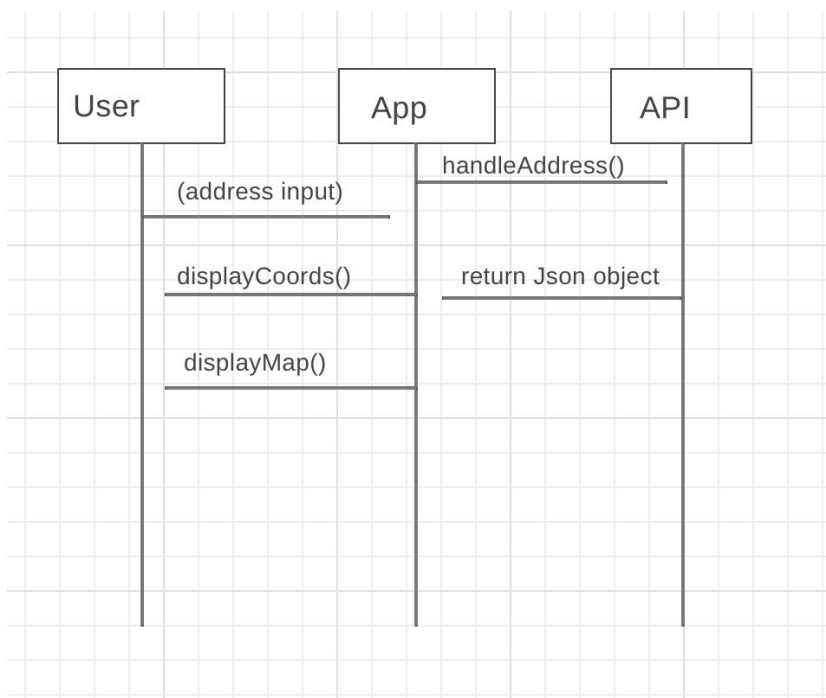


2-

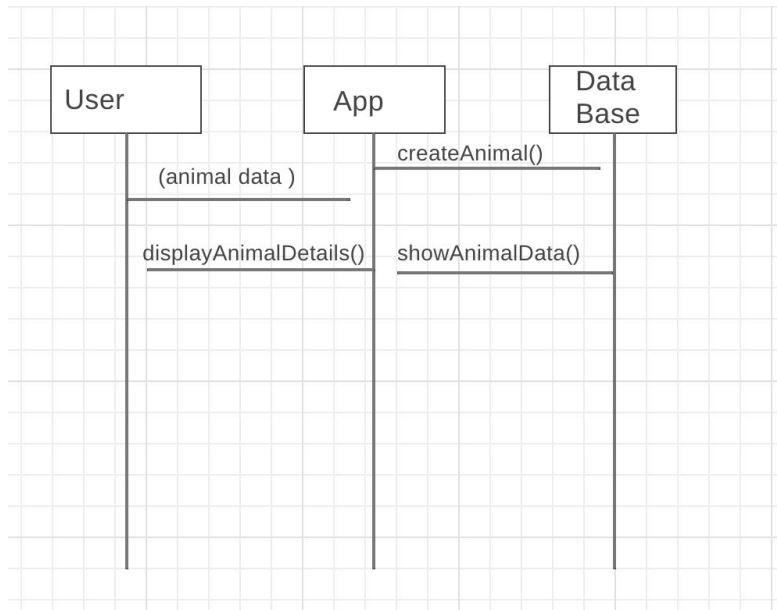


P-7 System interactions diagrams

1-

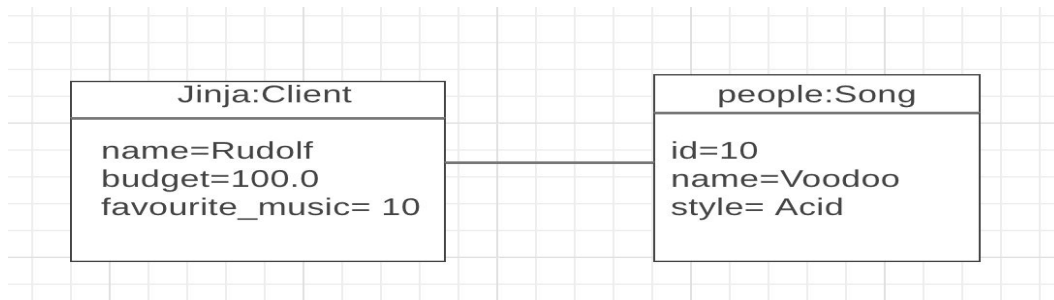


2-

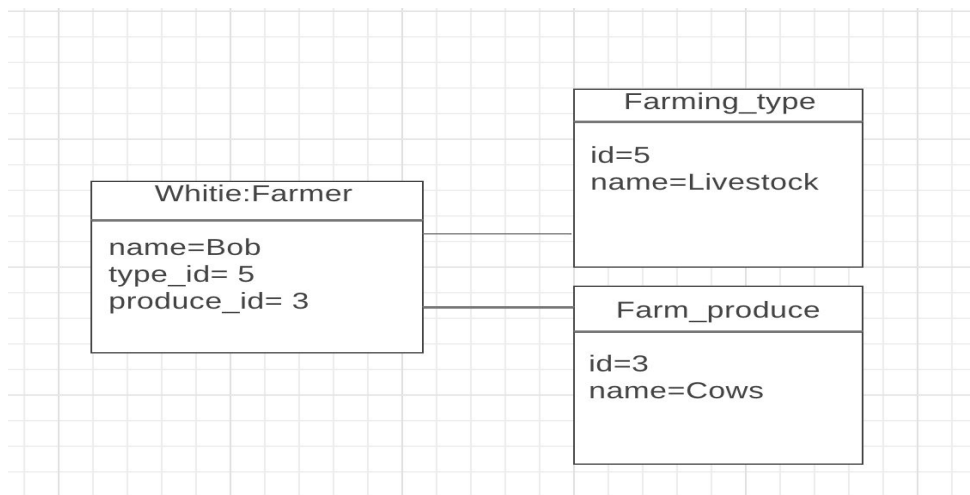


P-8 Two Object Diagrams

1-



2-



P- 9 -Choice of two algorithms

```
def spent()
  sql = "SELECT SUM(amount) FROM transactions INNER JOIN tags ON tags.id = transactions.tag_id
  WHERE tags.id = $1"
  values = [@id]
  results_array = SqlRunner.run(sql,values)
  return results_array.first['sum'].to_f
end

end
```

- Search algorithm - for a budget managing app this algorithm was created to create a prepared statement that will query the database for the transactions with a specific tag id and generate a total of the property amount for those transactions. This was chosen to be able to provide the user with the total spending for a certain tag.

```
def delete()
  sql = " DELETE FROM merchants WHERE id= $1"
  values = [@id]
  SqlRunner.run(sql,values)
end
```

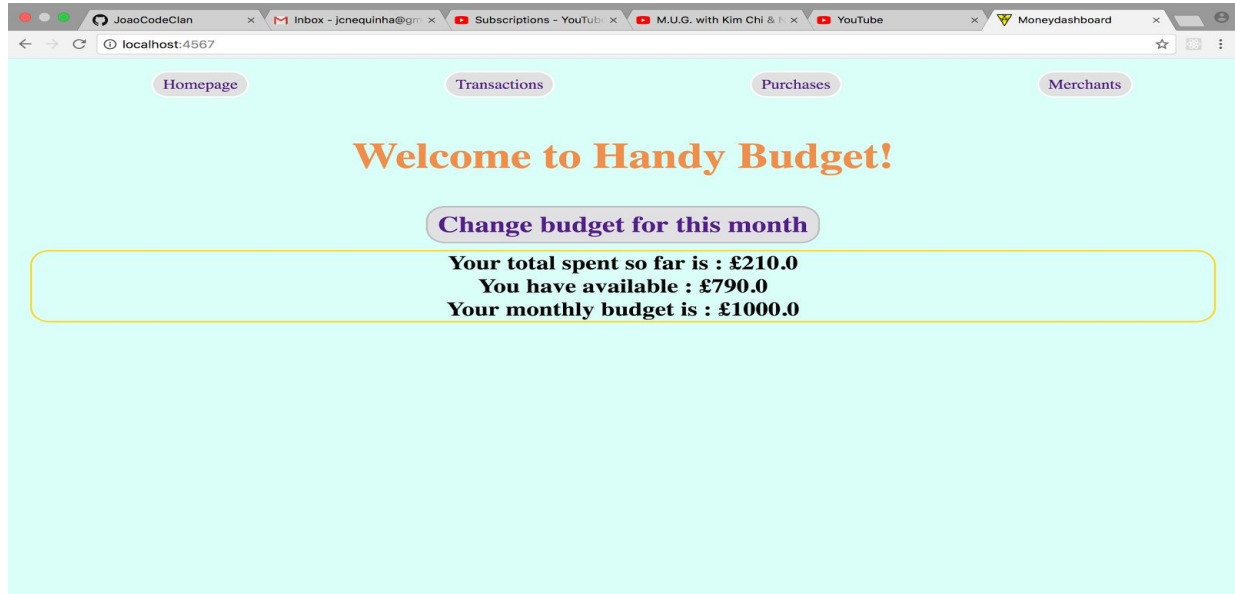
- Delete algorithm - for a budget managing app this algorithm creates a prepared statement that will query the database for a transaction with a specific id and will remove it from the database. This was chosen to be able to provide the user with the ability to delete transactions from the database.

P - 10 Pseudocode

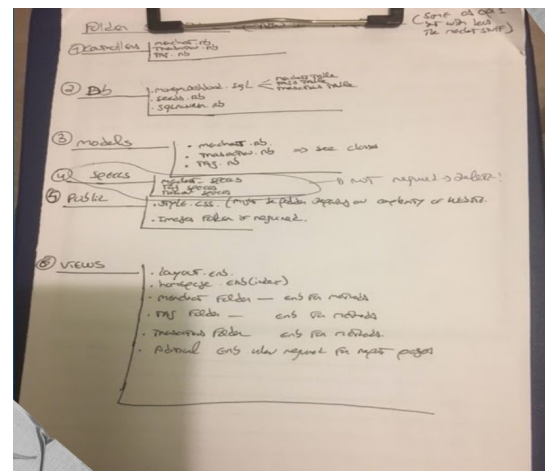
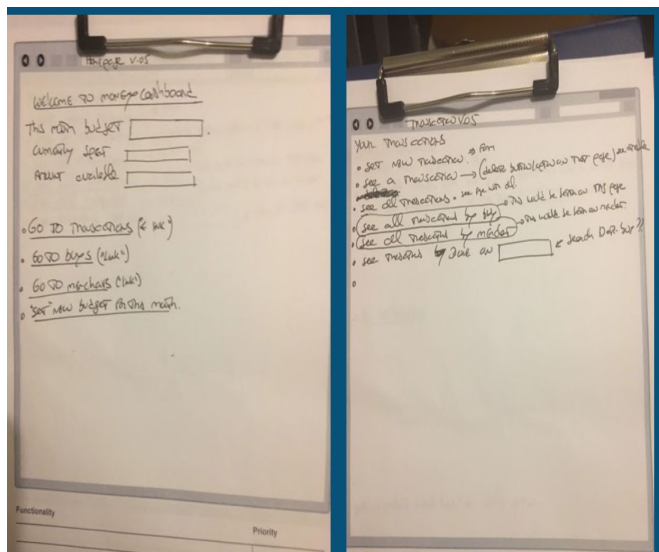
```
//public void add patron to a table function (takes a patron and a table as arguments ){
//checks if table.length is below table capacity property
//if check is positive -takes the patron and adds it to the table ( table is an array)
//if check is positive updates the restaurant total_patrons property by +1
//if check is negative returns a " table is full" message
//}
```

P - 11 Github link to one of your projects

https://github.com/JoaoCodeClan/joao_first_project



P - 12 Screenshot of your planning and the different stages of development to show changes.



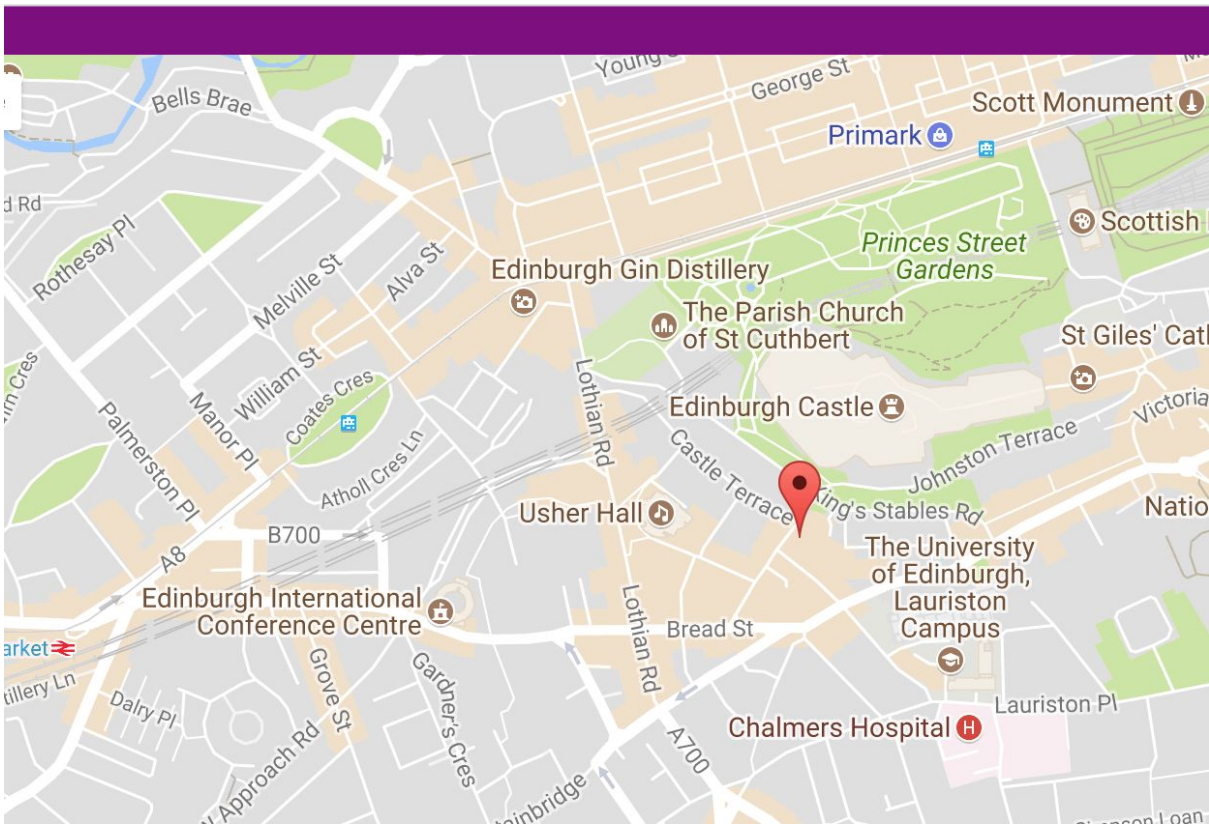
P - 13 User input

Codeclan

Latitude

Longitude

Codeclan
55.9469623
-3.2019583000000001



P - 14 Interaction with data persistence

Homepage Transactions Purchases Merchants

Create your new transaction below:

Amount Spent Date of transaction Select a merchant Select a type of purchase Record your transaction

Homepage Transactions Purchases Merchants

Your transaction has been created

All transactions

P - 15 Show the correct output of results and feedback to the user:

User looks up total amount spent by month and gets page with the total and forecast for monthly average and annual average based on that month.

Homepage Transactions Purchases

Your transactions

Add a new transaction
See all your transactions
See how much you've spent by month

Homepage Transactions Purchases Merchants

Choose the month and year of your transactions:

Month of transaction Year of transaction Search for your transactions

Homepage Transactions Purchases Merchants

Your total expenses for February in 2018 is:

30.0

Total annual expected expenses : 360.0

Average monthly expenses : 30.0

P-16 Show an API being used within your program:

Code that uses API :

```
AddressFormView.prototype.getAddress = function () {  
  
    const address = document.querySelector('#address-input').value;  
  
    const geocoder = new google.maps.Geocoder();  
    geocoder.geocode({'address': address}, function(results, status) {  
        if (status === 'OK') {  
            const result = results[0].geometry.location;  
            const lat = result.lat();  
            const lng = result.lng();  
  
            const latOutput = document.querySelector('#lat-output');  
            latOutput.value= lat;  
            const lngOutput = document.querySelector('#lng-output');  
            lngOutput.value = lng;  
  
            const mapContainer = document.querySelector('#google-map');  
            const newLocation = {  
  
                lat: lat,  
                lng: lng  
            }  
            const map = new MapView(mapContainer, newLocation, 15);  
            map.addMarker(newLocation);  
  
        }else{alert("Address not found.Please check if address is correct.");}  
    })  
};
```

```
const MapView = function(container, coords, zoom) {  
  this.map = new google.maps.Map(container, {  
    center: coords,  
    zoom: zoom  
  });  
}
```

```
MapView.prototype.addMarker= function (coords) {  
  const marker = new google.maps.Marker({  
    position: coords,  
    map: this.map,  
    animation: google.maps.Animation.DROP  
  });  
  return marker;  
}
```

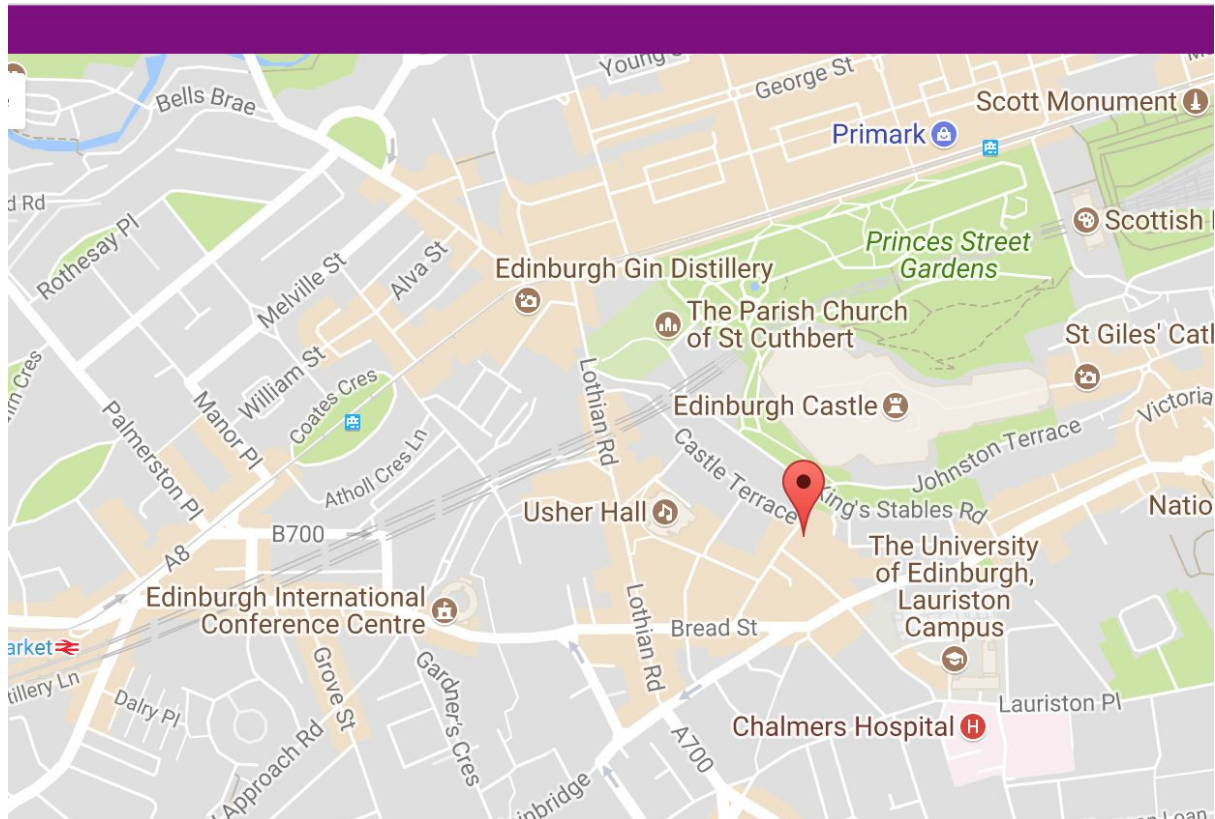
```
module.exports = MapView;
```

API being used by program whilst running:

Codeclan

55.9469623

-3.2019583000000001



P - 17 Bug tracking report showing the errors diagnosed and corrected.

Map is centered around the user location on Around me pager	Failed	Corrected code using geolocation from Google Maps and passing it to the create map function	Passed
User is able to see the location of events on map (location marked by pins)	Failed	Corrected createmarker function so that it would return the marker	Passed
User is able to see all events on table	Failed	Added pagination functionality to table	Passed
User is able to use navigation bar	Failed	Corrected code in order to redirect to correct pages	Passed
Search button is loaded when navigating between pages	Failed	Changed the property display of the html elements in order to correctly update when navigating between pages	Passed

P -18 Demonstrate testing in your program :

Example of test code:

```
import ...

public class PatronTest {

    Patron patron;
    Menu menu;
    Dish dish1;
    Dish dish2;
    Ingredient ingredient1;
    Ingredient ingredient2;

    @Before

    public void before(){

        patron= new Patron();
        menu = new Menu();
        dish1= new Dish("soup",3.0);
        dish2 = new Dish( "drumsticks", 4.5);
        ingredient1 = new Ingredient("Passatta", 3.0, 5);
        ingredient2 = new Ingredient("egg", 3.0, 5);

    }

    @Test

    public void canGetBillValueAndStartsAtZero(){
        double actual= patron.getBill();
        assertEquals(0,actual, 0.01 );
    }

    @Test

    public void canSetBill(){
        patron.setBill(20.0);
        double actual = patron.getBill();
        assertEquals(20. actual. 0.01);
    }
}
```

Test code failing to pass:

@Test

```
public void canGetBillValueAndStartsAtZero(){
    double actual= patron.getBill();
    assertEquals( expected: 1,actual, delta: 0.01 );
}
```

@Test



Test code once errors have been corrected:

@Test

```
public void canGetBillValueAndStartsAtZero(){  
    double actual= patron.getBill();  
    assertEquals( expected: 0,actual, delta: 0.01 );  
}
```

The test code passing :

