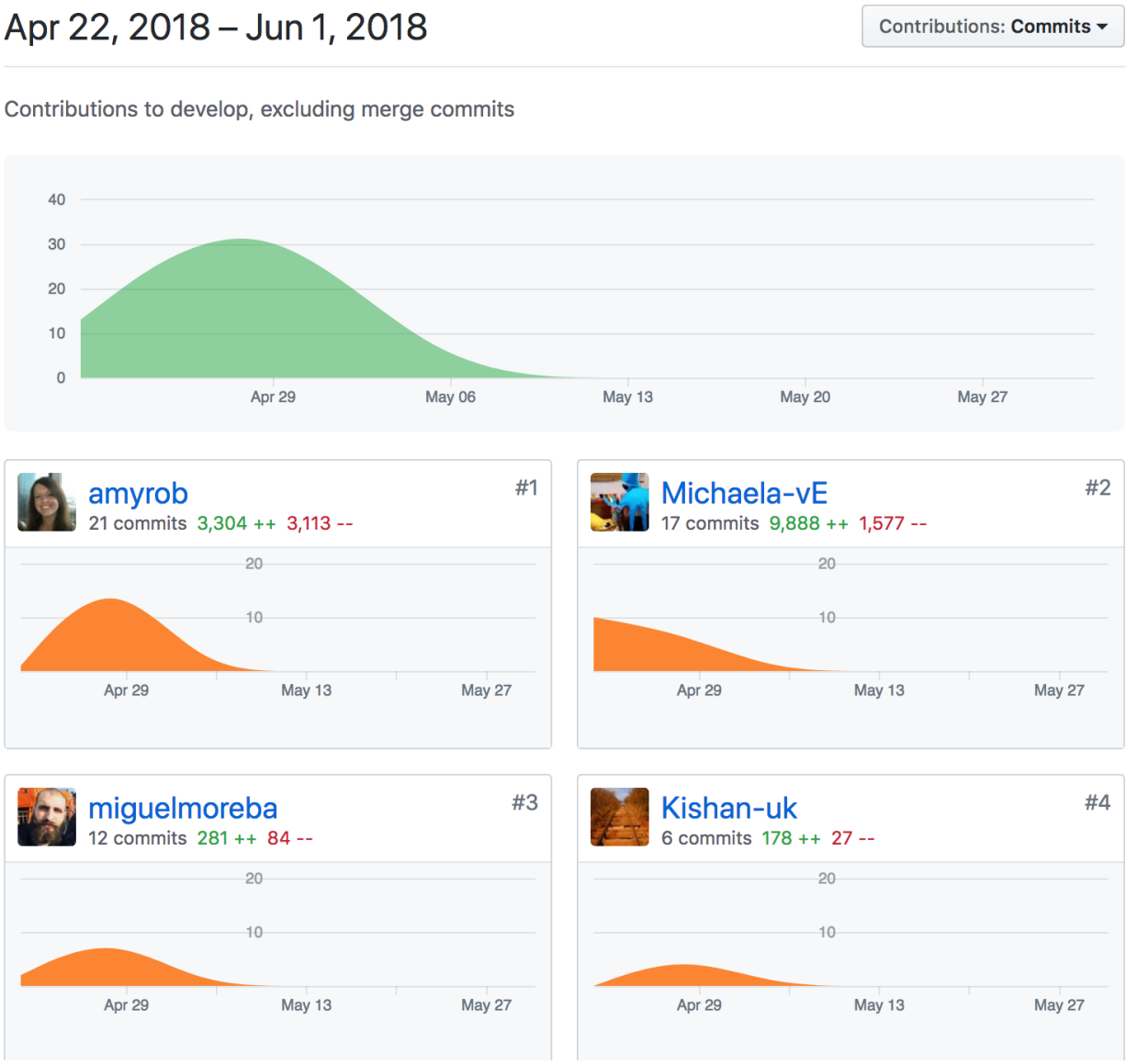
Project Unit

Kishan Bachoo
E19

P 1 Take a screenshot of the contibutor's page on Github from your group project to show the team you have worked with.



P 2 Take a screenshot of the project brief from your group project

Store Finder

Global pub giant Withoutaspoon, wants a pub finder that allows users to "Find your nearest store" for their website.

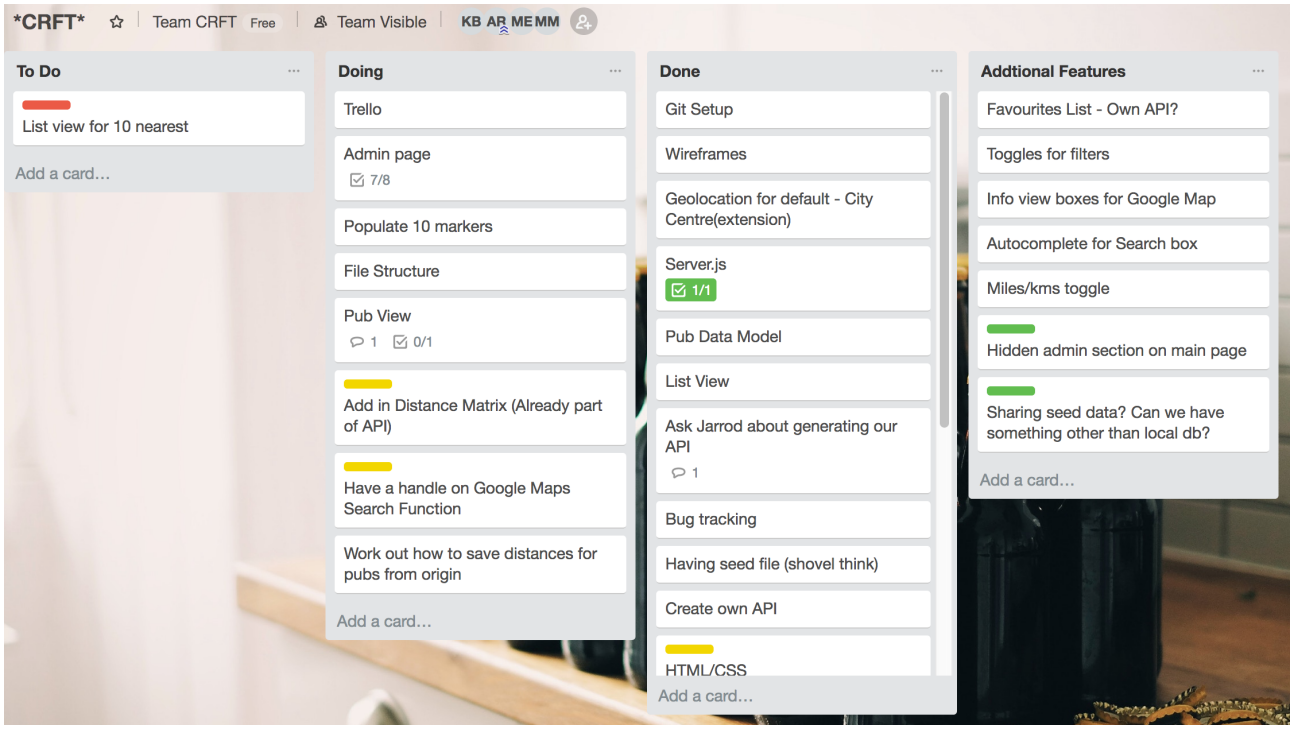
User should be able to enter their postcode or town or street name and get a list of the top ten nearest stores. Each store should have a distance, address, phone number, opening hours and a list of facilities and services (e.g. cash machine, car parking).

You'll have to build your own API to persist the store data and use an external api to find addresses from postcodes.

MVP

- Search by postcode or town or street name
- A list of stores including the details of the stores
- A map marking the stores
- Use Geolocation to allow users to "use current location" to find stores

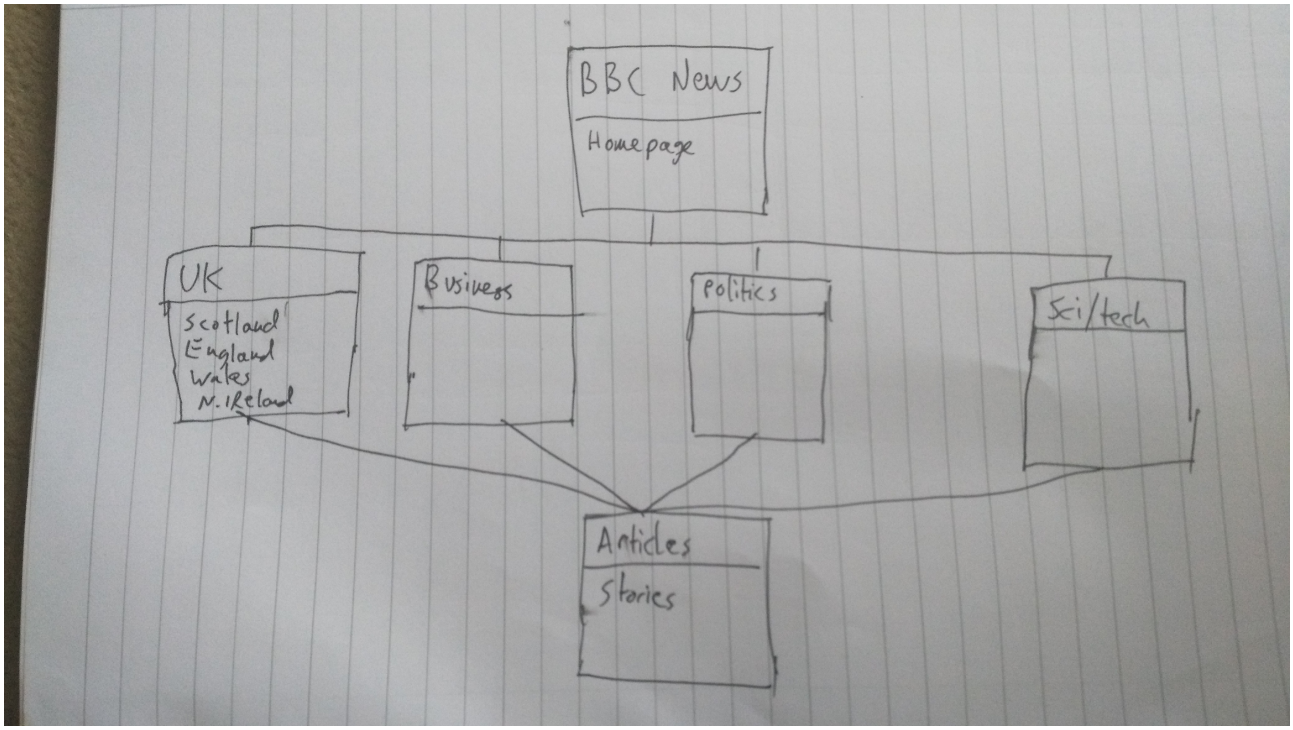
P 3 Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board



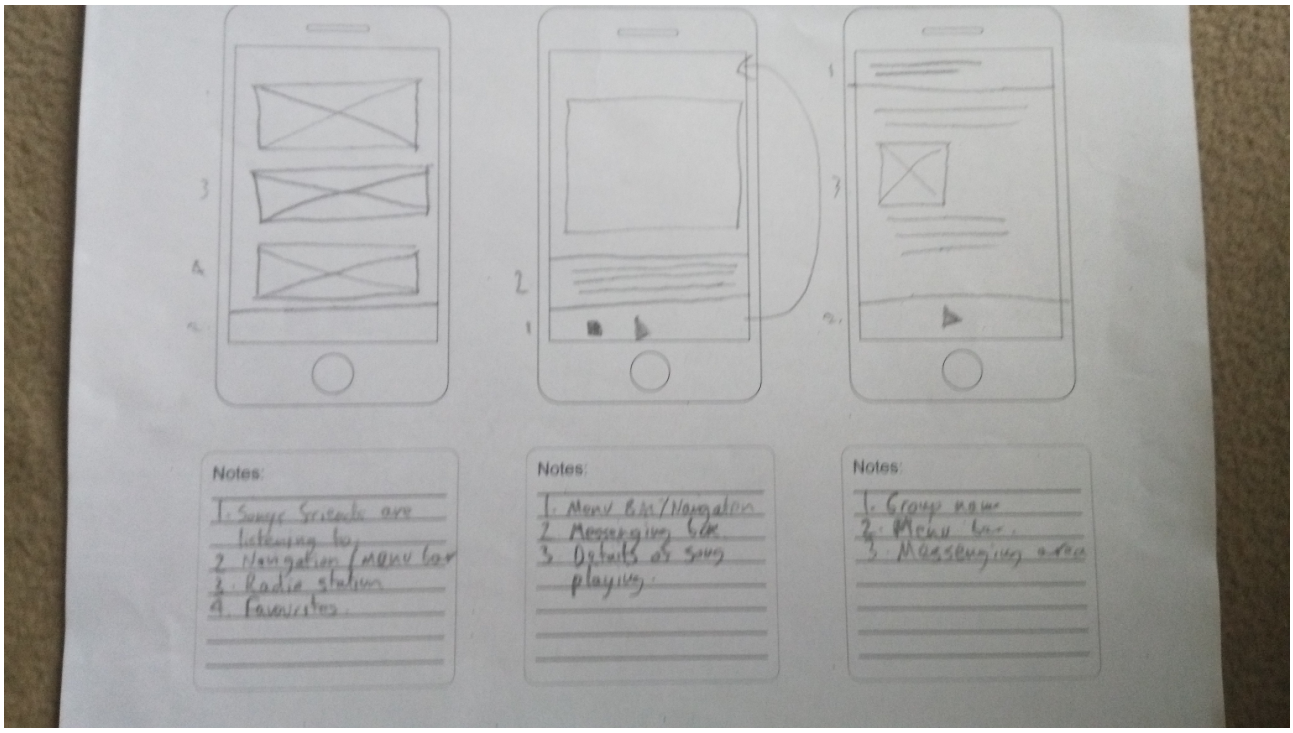
P 4 Write an acceptance criteria and test plan

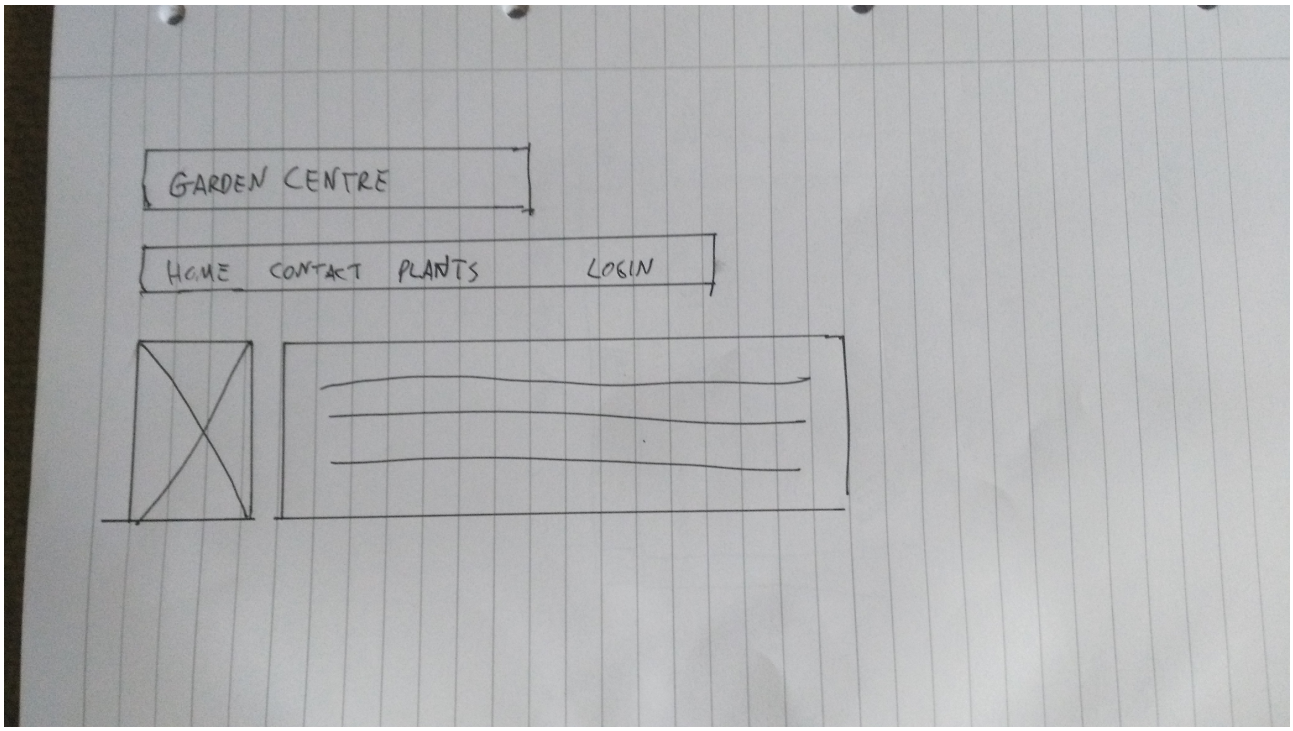
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user should be able to check move around the map	The user should be able to click and hold a button that allows them to pan around the map	PASS
The user should be able to select a pub and get details about it	The user should be able to click on a pub marker and bring up information about the pub in an infobox	PASS
The user should be able to get the 10 nearest pubs from their location	The map should display 10 markers of the nearest pubs from their location	PASS

P 5 Create a user sitemap

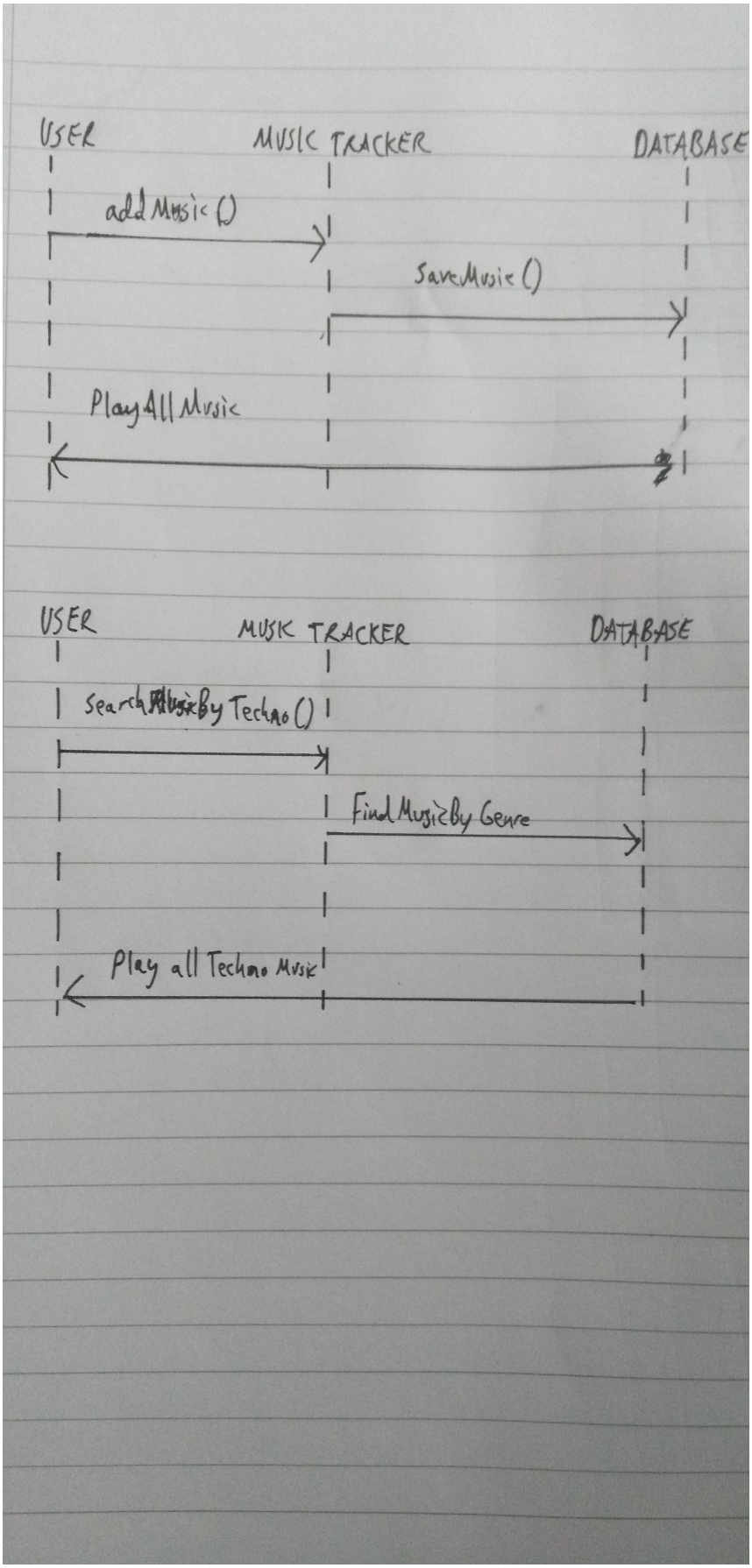


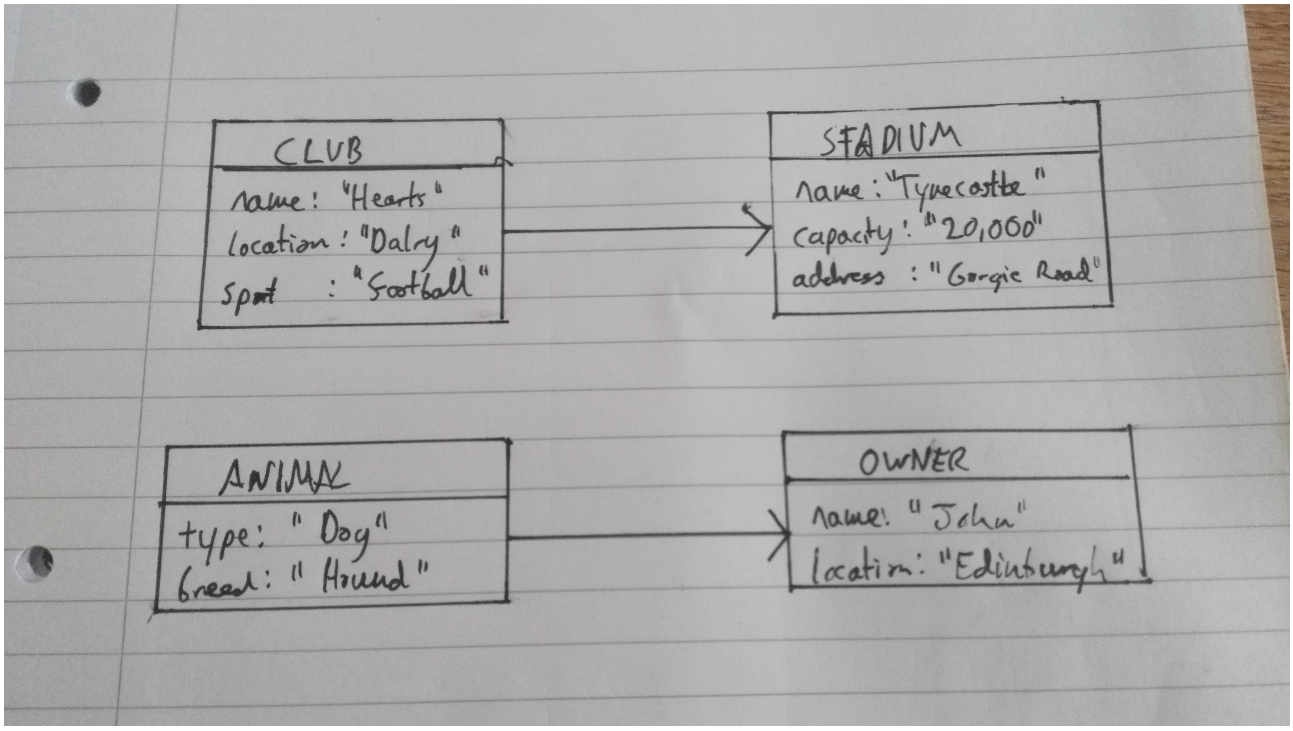
P 6 Produce two wireframe designs



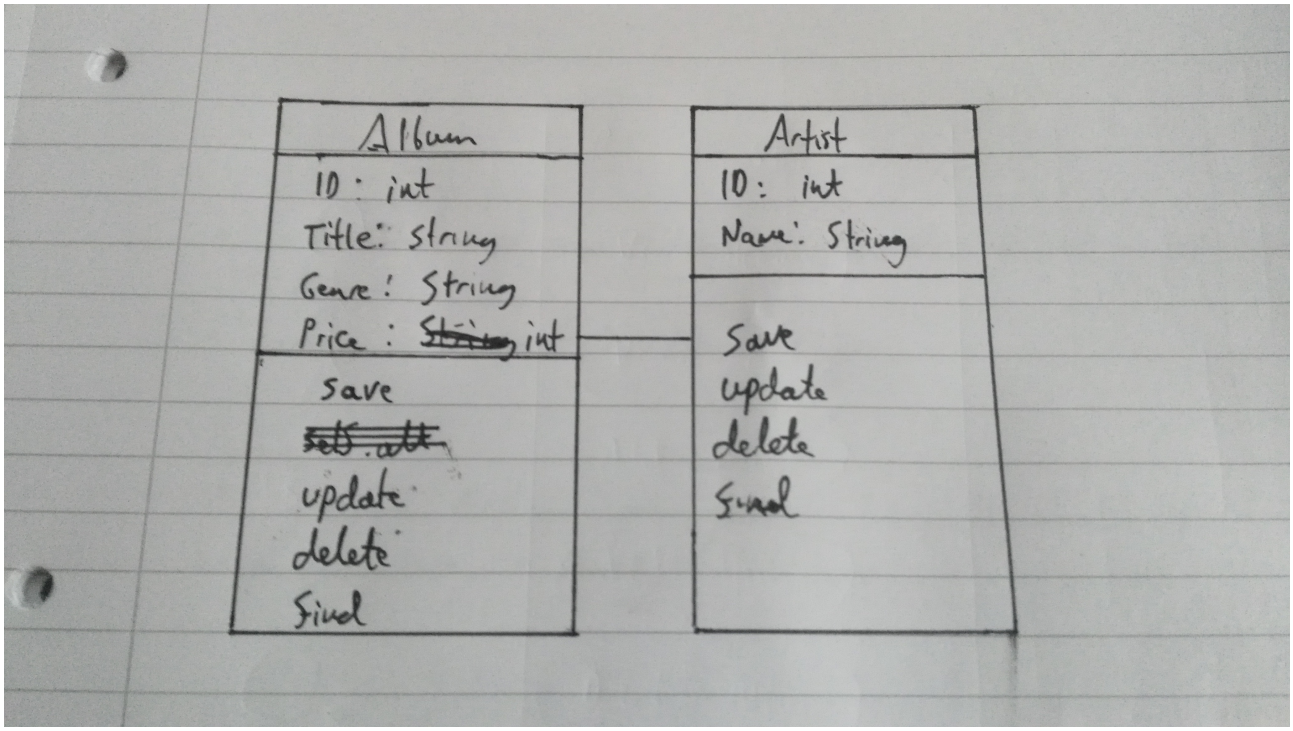


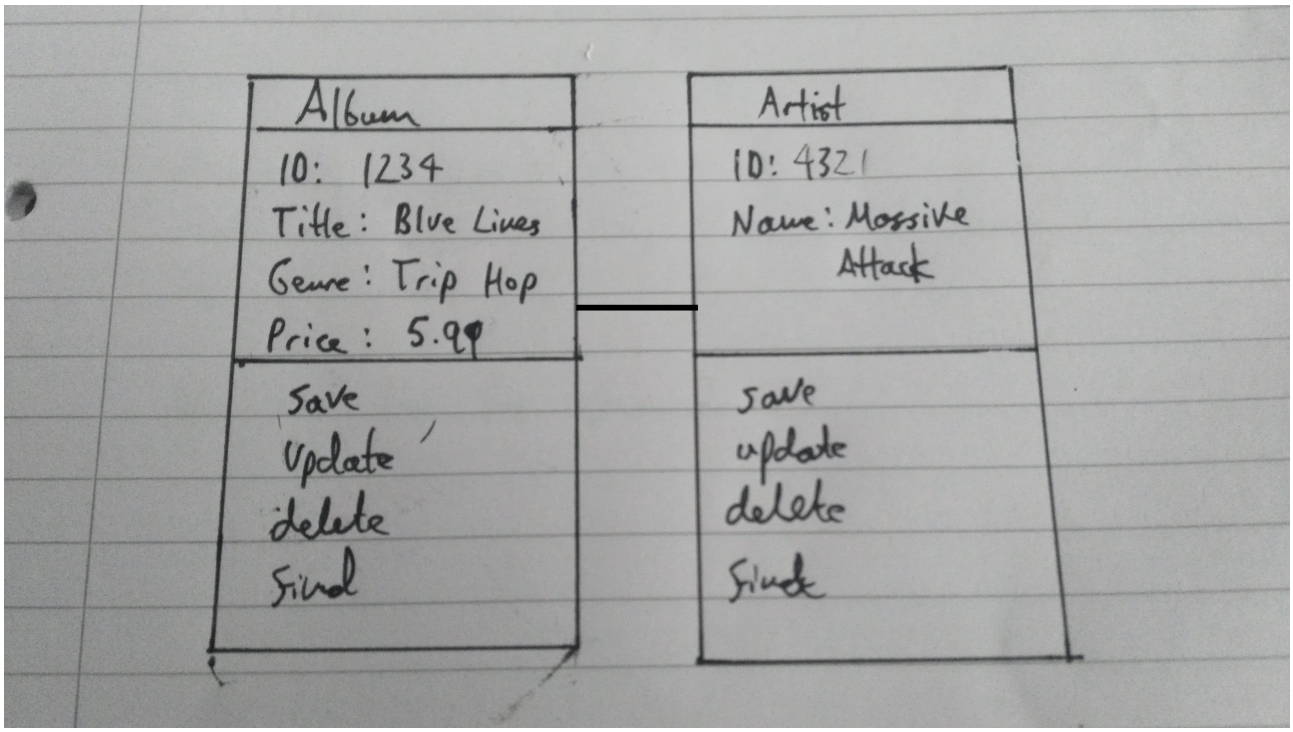
P 7 Produce two system interaction diagrams (sequence and/or collaboration diagrams)





P 8 Produce two object diagrams





P 9 Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms

```
public void selectPaymentType(String paymentMethod) {
    for(PaymentType paymentType: paymentList  ){
        if(paymentType.getPaymentMethod() == paymentMethod){
            payment = paymentType;
        }
    }

}
```

I chose this algorithm as it was the simplest way of selecting a payment method.

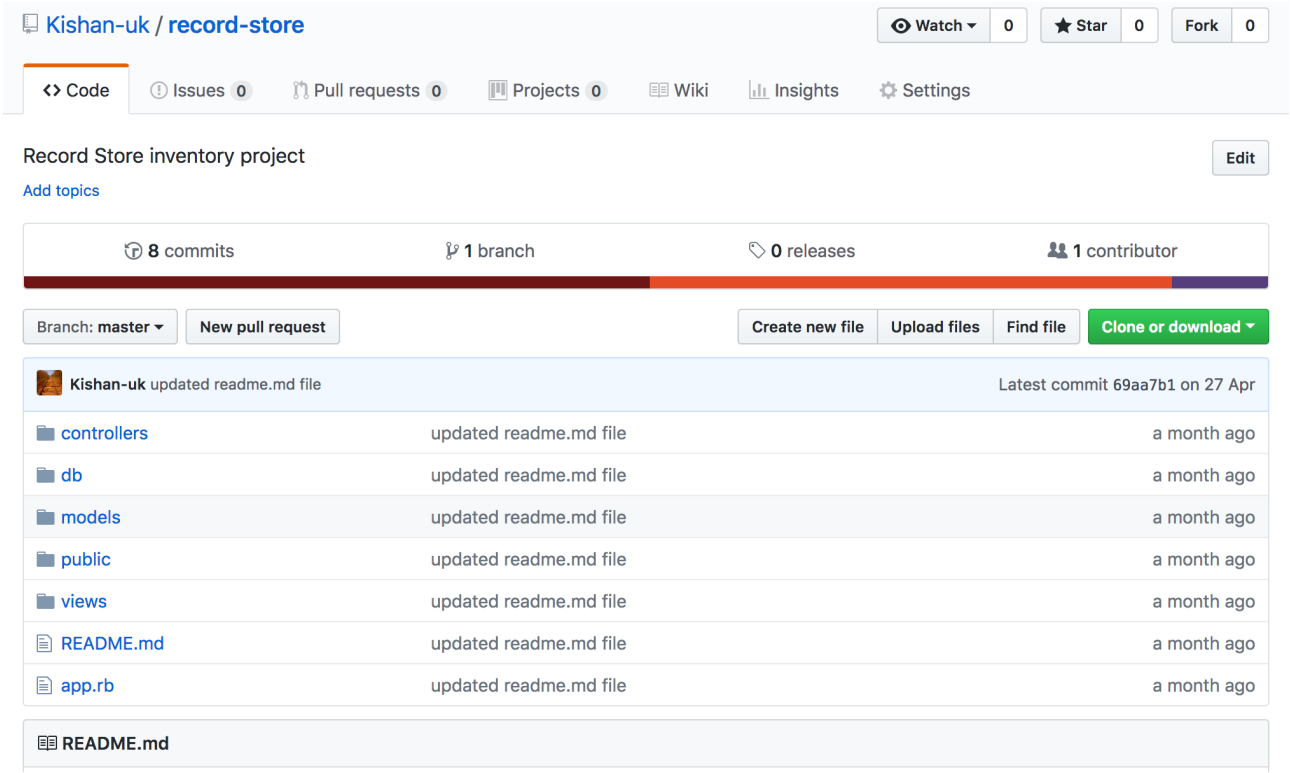
```java
public void makeSaleToCustomer(Product product, Customer customer, int orderItemQuantity) {

    if (products.contains(product)) {
        if (product.getQuantity() > 0) {
            customer.buyProduct(product.getPrice(),orderItemQuantity);
            purchasedProducts.add(product);
            sales = sales + (product.getPrice() * orderItemQuantity);
            int newQuantity = product.getQuantity() - orderItemQuantity ;
            product.setQuantity(newQuantity);
        }

        if (product.getQuantity() == 0) {
            products.remove(product);
        }
    }
}
```

I chose the algorithm as it demonstrated the robustness of working out the sales price only when there are items that have been added to the product quantity.

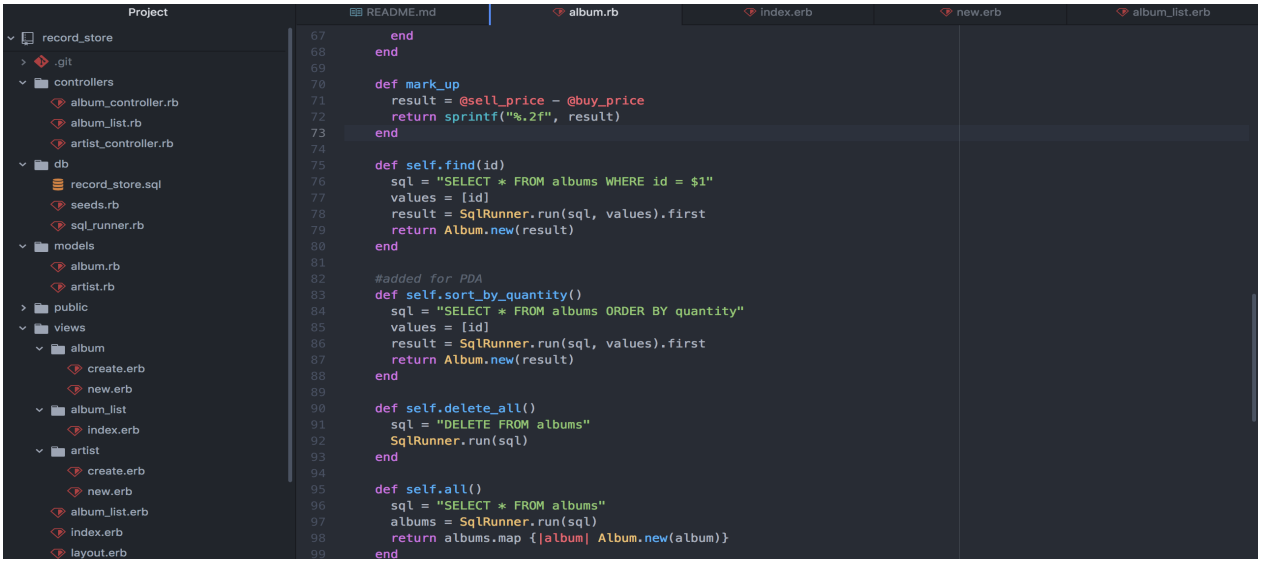
P 10 Take a screenshot of an example of pseudocode for a function

```
1  // Create function that allows users to display all Albums
2  // Select all albums from the database
3  // Return albums via a Map
4  // End function
```

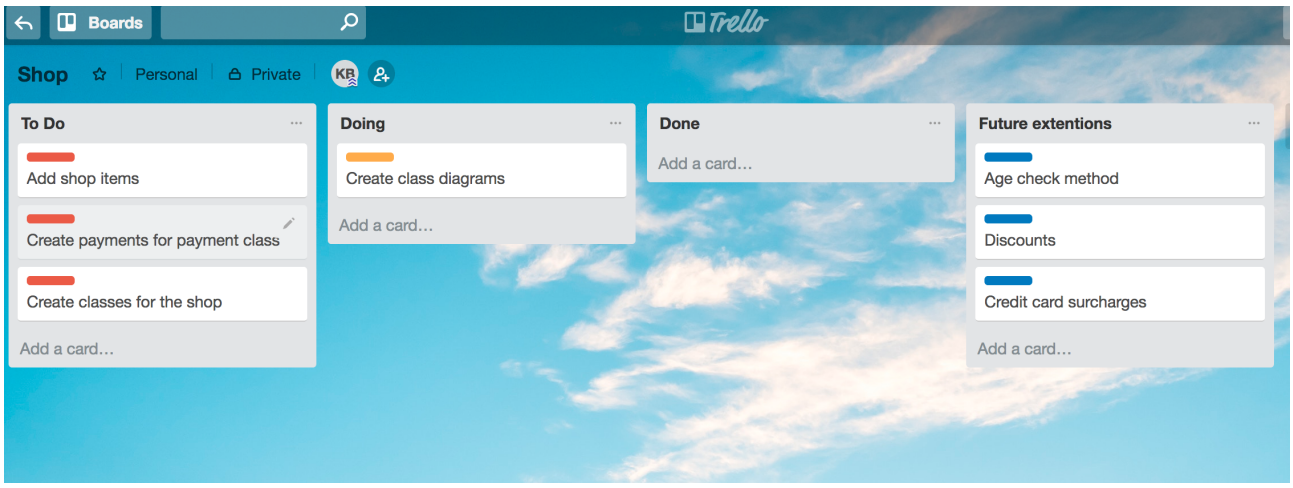
P 11 Take a screenst hot of one of your projects where you have worked alone and attach a Github link

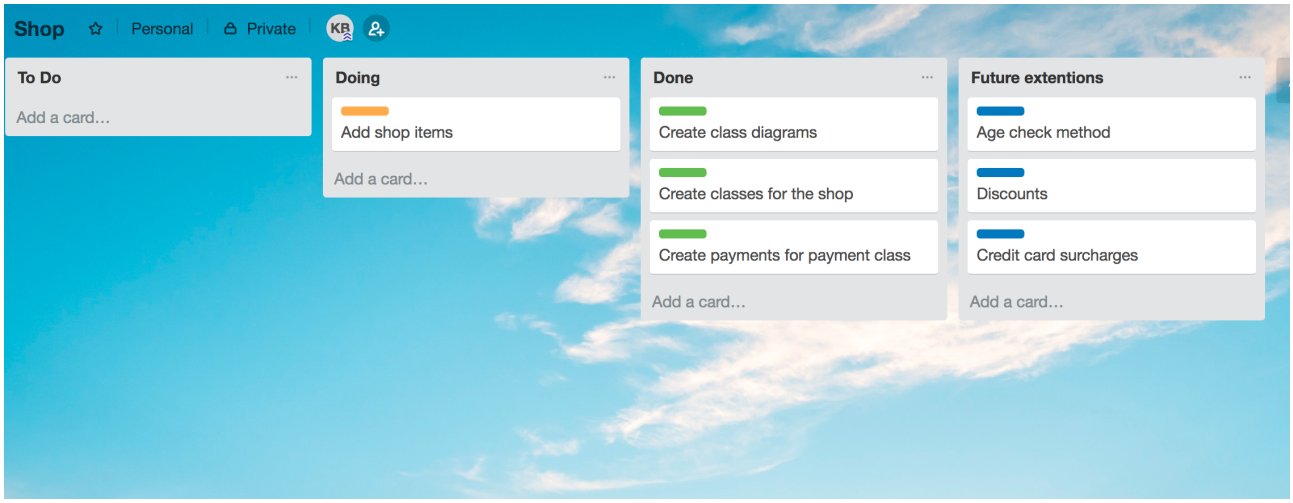


https://github.com/Kishan-uk/record-store



P 12 Take screenshots or photos of your planning and the different stages of development to show changes.





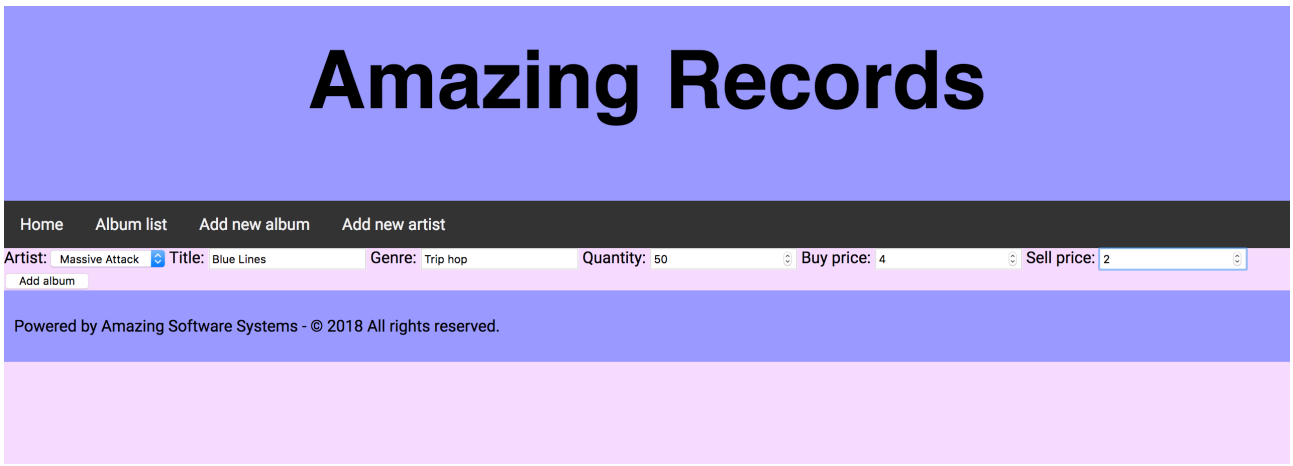
P 13 Show user input being processed according to design requirements. Take a screenshot of:

- Data being inputted into your program

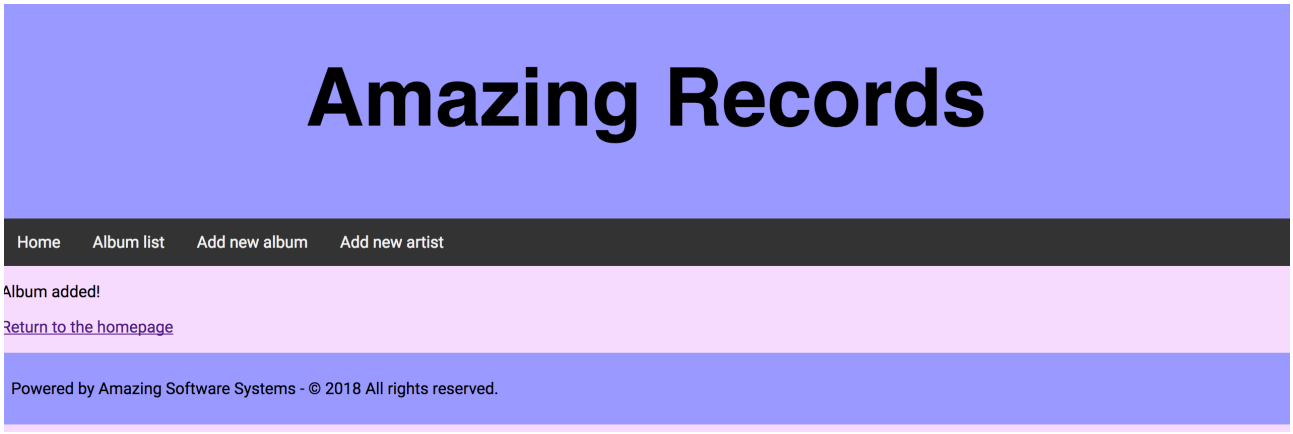
- The user input being saved

P14 Show an interaction with data persistence. Take a screenshot of:

- Data being inputted into your program

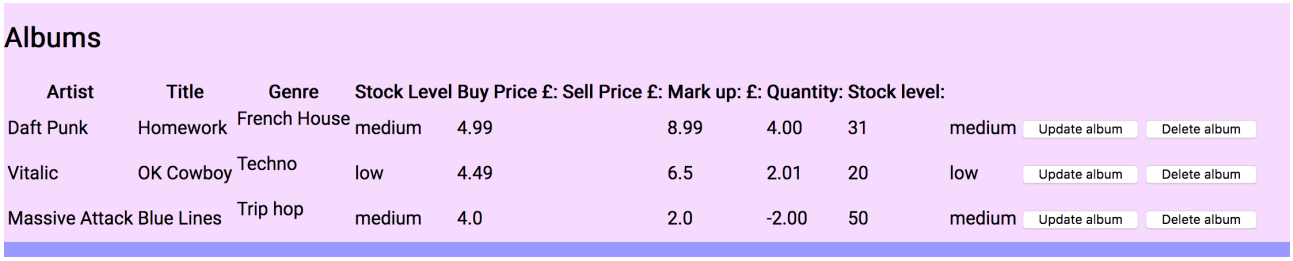


- Confirmation of the data being saved



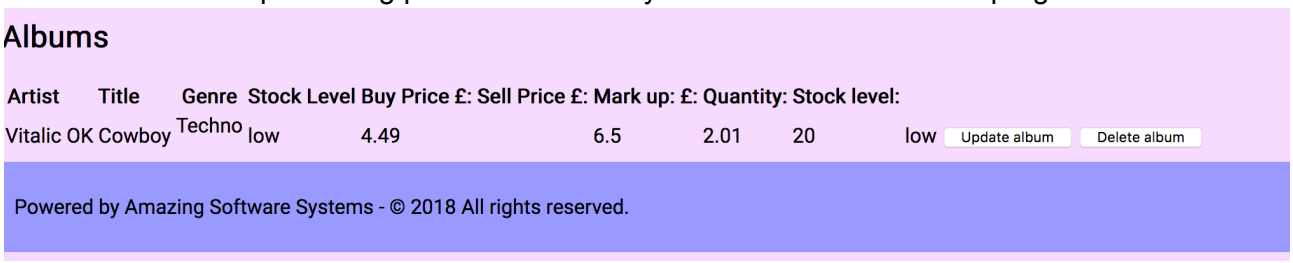
P 15 Show the correct output of results and feedback to user. Take a screenshot of:

- The user requesting information or an action to be performed



(albums to be deleted through the *delete album* button)

- The user request beng processed correctly and demonstrated in the program



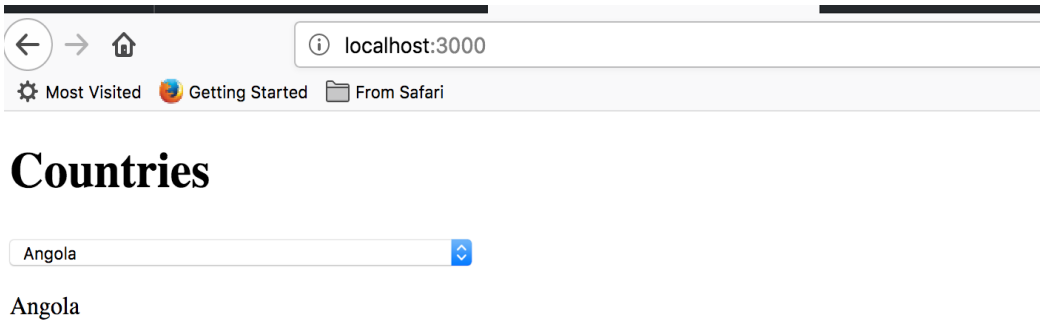
P 16 Show an API being used within your program. Take a screenshot of:

- The code that uses or implements the API

```
const Request = require('../helpers/request.js');

const CountryData = function() {
  this.url = 'https://restcountries.eu/rest/v2/all';
  this.data = null;
}

CountryData.prototype.getData = function (onComplete) {
  const request = new Request(this.url);
  request.get((data) => {
    console.log(data);
    this.data = data;
    onComplete(data);
  })
};

module.exports = CountryData;
```

```
const CountryView = function(selectElement, countryContainer) {
  this.selectElement = selectElement;
  this.countryContainer = countryContainer;
}

CountryView.prototype.renderSelect = function (countryData) {
  countryData.forEach((country, index) => {
    const countryOption = document.createElement('option');
    countryOption.textContent = country.name;
    countryOption.value = index;
    this.selectElement.appendChild(countryOption);
  });
};

CountryView.prototype.renderDetail = function (country) {
  const countryName = document.createElement('p');
  countryName.textContent = country.name;
  this.countryContainer.appendChild(countryName);
}

module.exports = CountryView;
```

- This API being used by the program whilst running



P 17 Produce a bug tracking report

User must be able to connect to DB	Failed	Placement of server listen within Mongo client. No root url being passed in app.js server();	Passed
User must be able to see map	Failed	Reran build	Passed

P 18 Demonstrate testing in your program. Take screenshots of:

- Example of test code

```
@Test
public void makeSaleToCustomerTest() {
    shop.makeSaleToCustomer(product, customer, product.getQuantity());
    assertEquals( expected: 3, product.getQuantity());
    assertEquals( expected: 107.96, shop.getShopSales(),  delta: 0.01);
    assertEquals( expected: 192.04, customer.getBalanceFromPaymentType(),  delta: 0.01);

}
```

- The test code failing to pass

- Example of the test code once errors have been corrected

- The test code passing