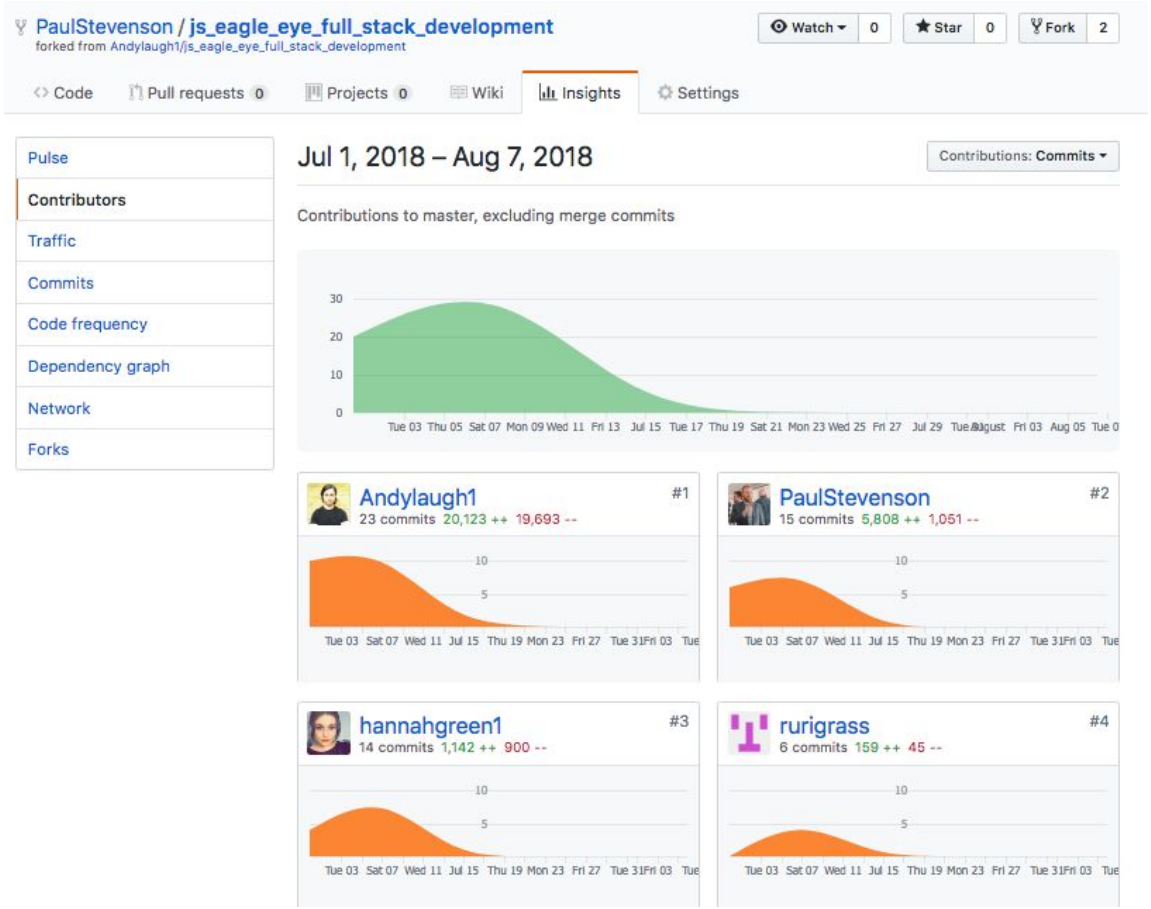
Project Unit
Paul McPhail Stevenson
Cohort E21

P 1 Github Contribution Page



Project Unit
Paul McPhail Stevenson
Cohort E21

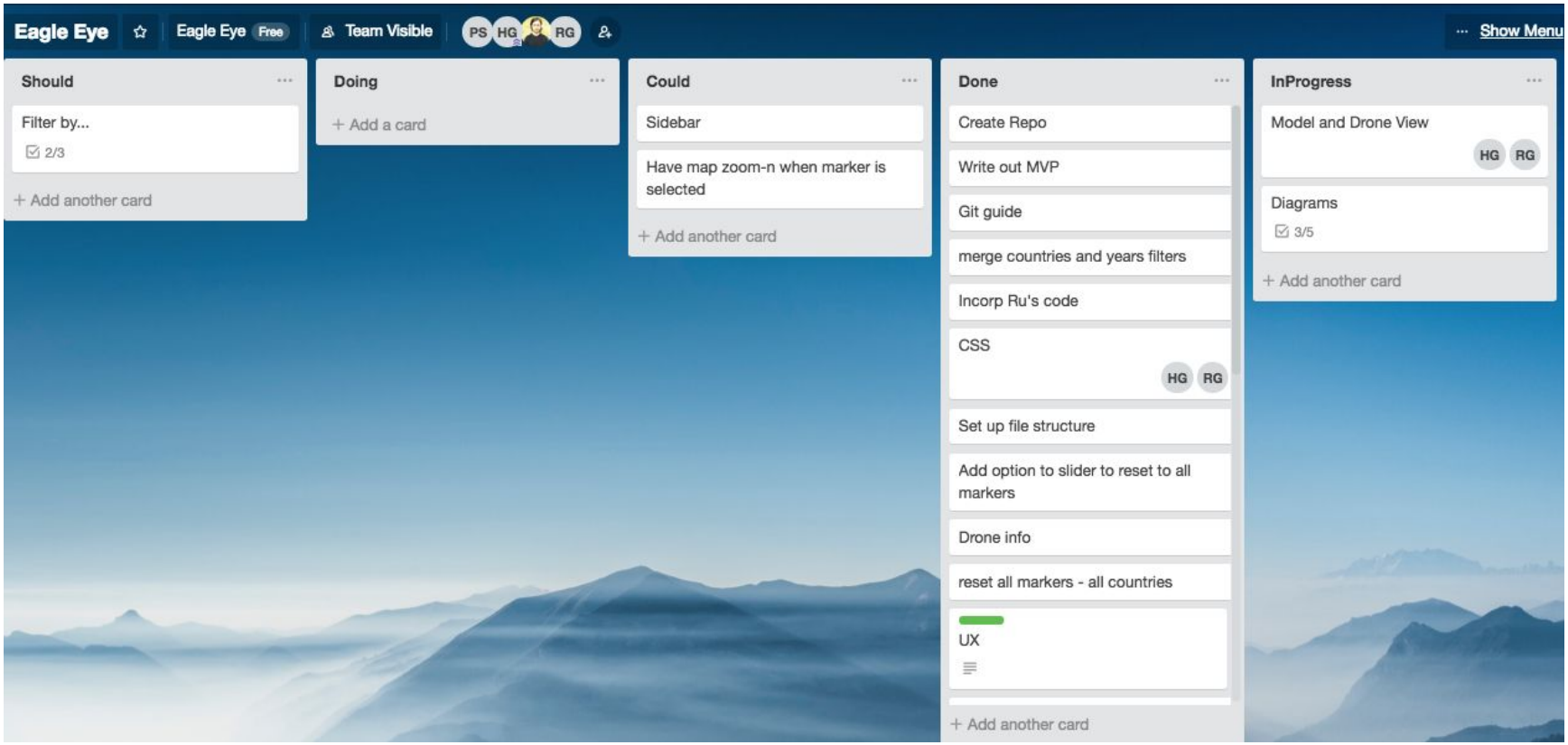
P 2 Project Brief

readme.txt

```
Educational App- Project Brief
The BBC are looking to improve their online offering of educational content by developing some interactive apps that display information
in a fun and interesting way. Your task is to make an MVP to put forward to them - this may only be for a small set of information, and
may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store
facts.

MVP- Designed by team
Build an app to allow users to view information regarding covert drone strikes since 2002.
Users should be able to click on map and find out the drone data from that area.

* Build Full Stack JavaScript application
   * NodeJS
   * Webpack
   * HTML
   * CSS
* Plan project using:
   * UX
   * Trello
* Behaviour Driven Development
* Make use of API's
   * https://api.dronestre.am/data
* Have some interactivity that enables a user to move through different sections of content

How to open and run this project:

From terminal run the following commands:
- Npm install
- Npm run Build
- Npm run server:dev
- mongod
- Open the project in editor
- http://localhost:3000/

Possible improvements in future or time permitting:
- Code refactor and clean up (movement of logic into models)
- Switch to REACT
- Add more chart features
```

Project Unit
Paul McPhail Stevenson
Cohort E21

P 3 Project Planning



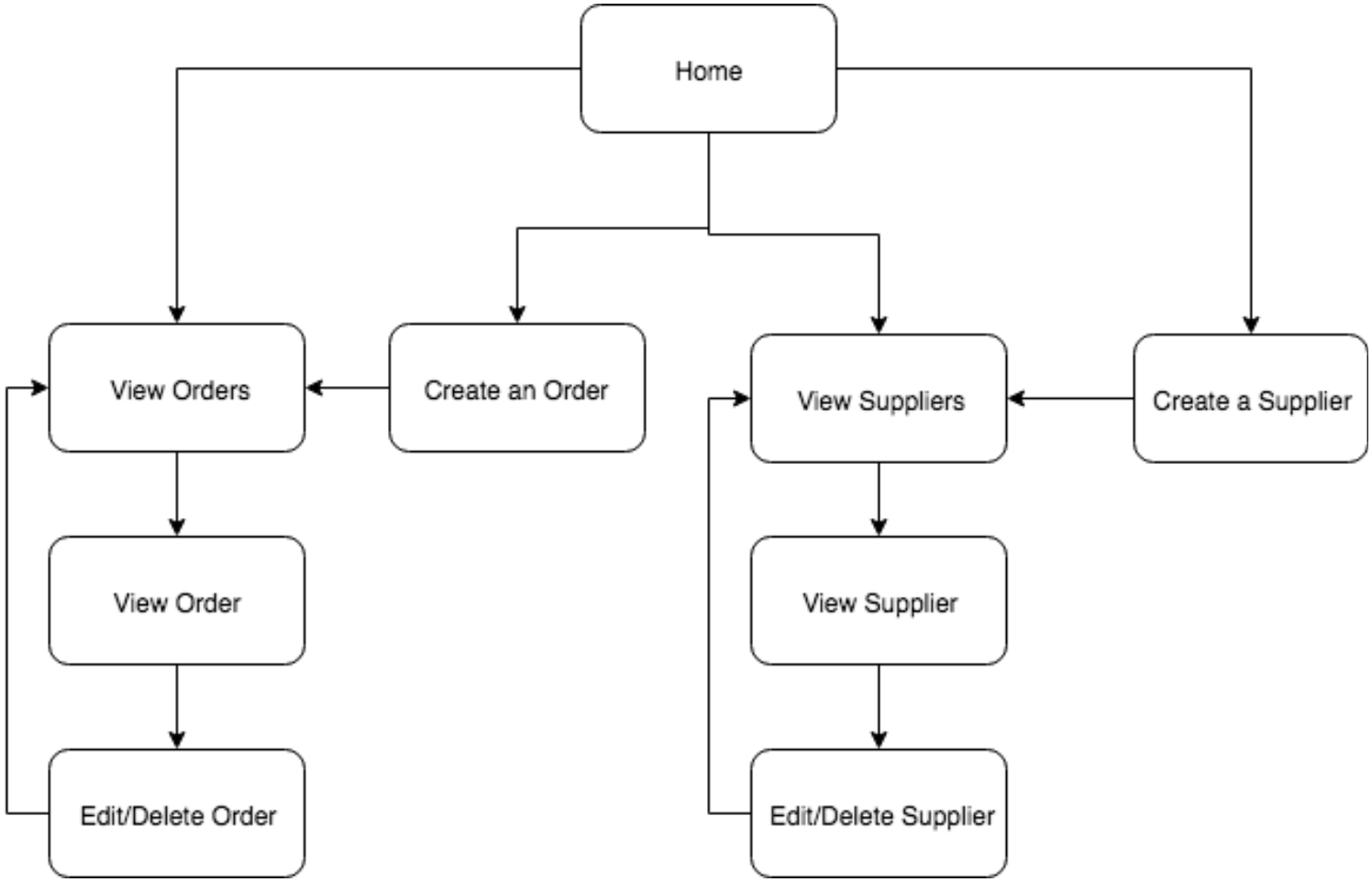
Project Unit
Paul McPhail Stevenson
Cohort E21

P 4 Acceptance Criteria and Test Plan

Acceptance Criteria	Expected Result	Pass
A user should be able to open and view the app	Webpage successfully loads with the map rendering the all the markers. Content should populate in the appropriate containers.	Pass
A user should be able to select a marker on the map, changing to the content displayed.	Clicking the marker populates the the ‘Strike Info’ container, and ‘Conflict History’ container specific to the data attached to the marker .	Pass
A user should be able to choose a specific country from the dropdown menu.	The map should zoom into the chosen country.	Pass
A user should be able to use the slider showing the chronological order of drone strikes.	The markers should render on the map in chronological order using the Date Key provided by the API using the slider.	Pass
A user should be able to reset the map to the default view	Clicking the reset button should recentre the map, and re-populate the markers.	Pass

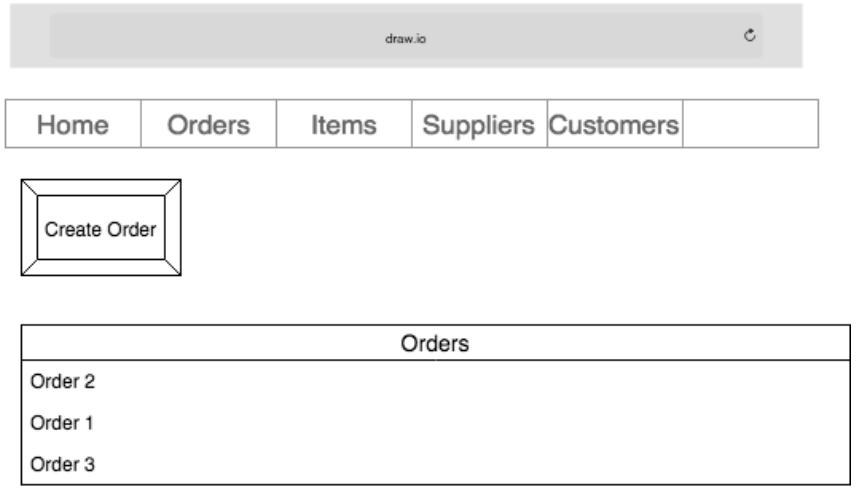
Project Unit
Paul McPhail Stevenson
Cohort E21

P 5 User Site Map



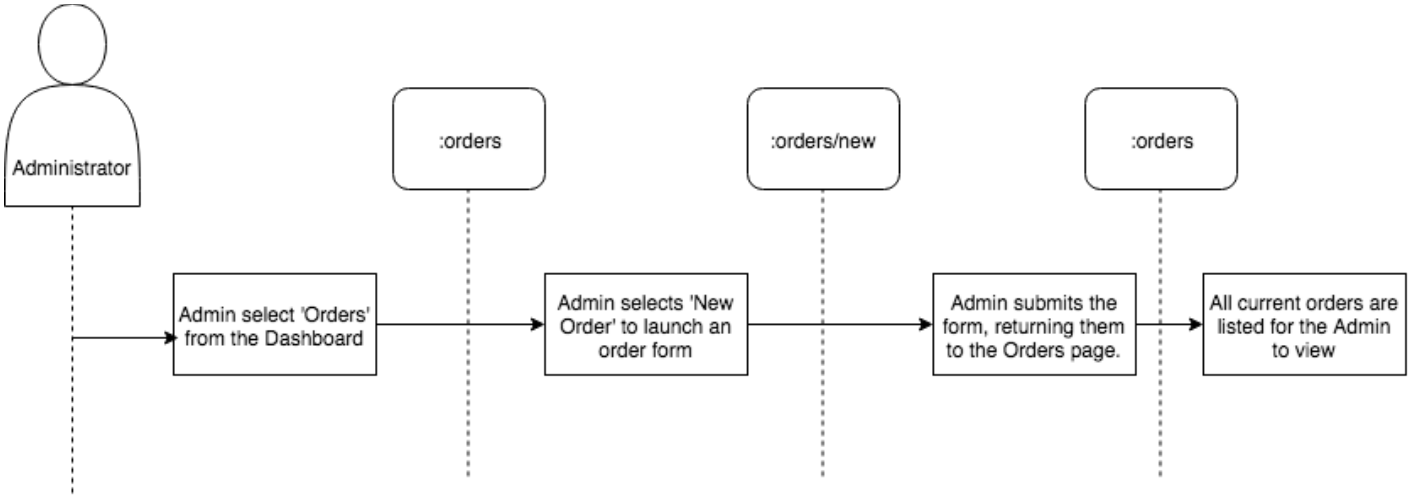
Project Unit
Paul McPhail Stevenson
Cohort E21

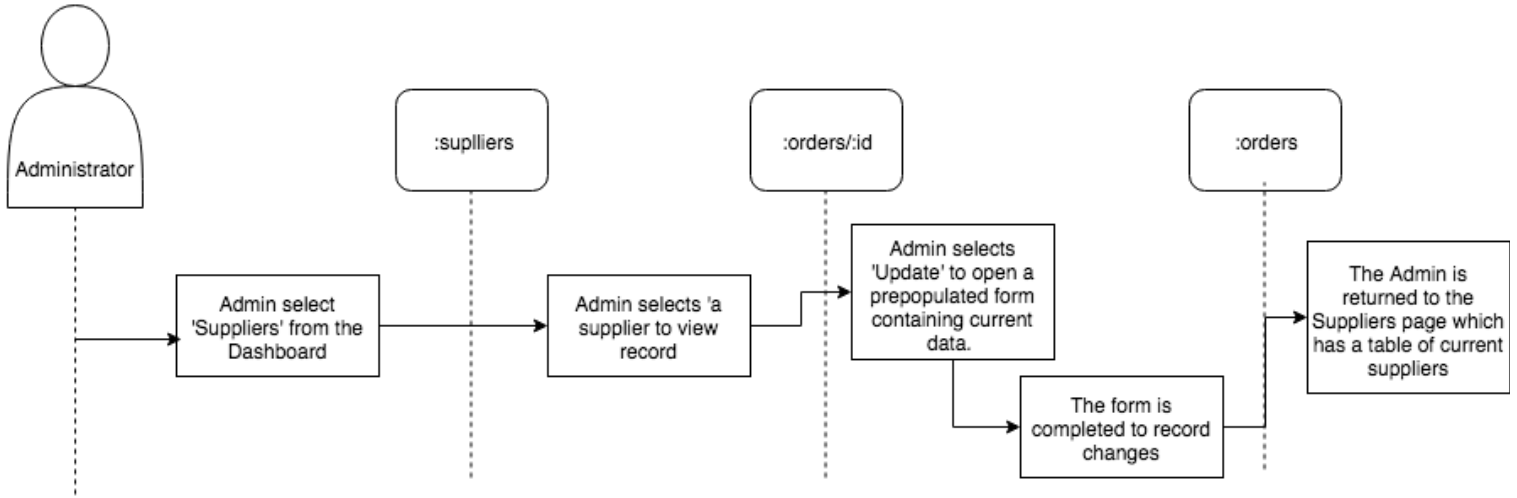
P 6 Two Wireframe Designs



Project Unit
Paul McPhail Stevenson
Cohort E21

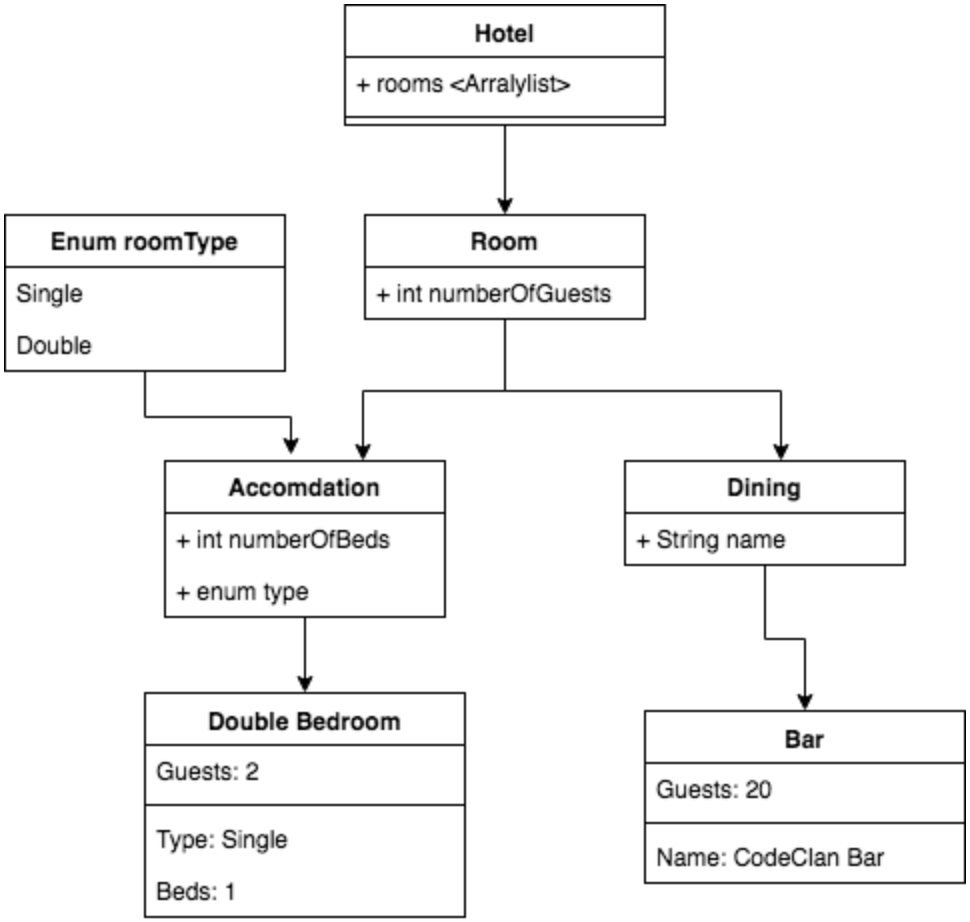
P 7 Two System Interaction Diagrams





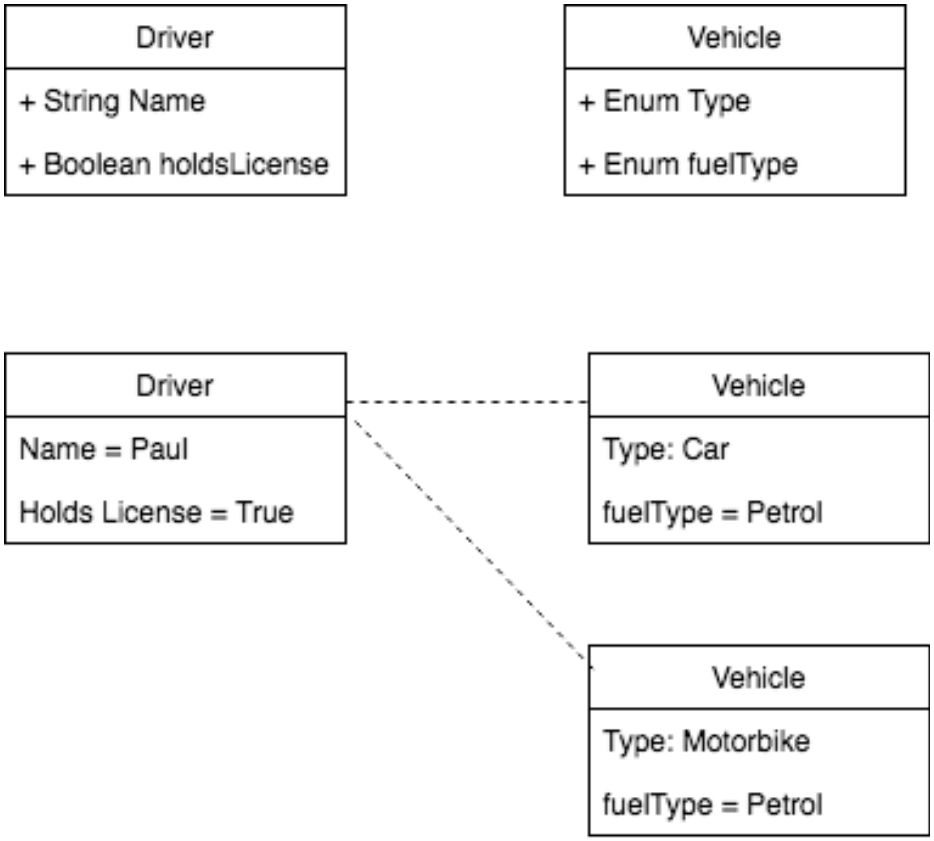
Project Unit
Paul McPhail Stevenson
Cohort E21

P 8.1 Two Object Diagrams



Project Unit
Paul McPhail Stevenson
Cohort E21

P 8.2 Two Object Diagrams



Project Unit
Paul McPhail Stevenson
Cohort E21

P 9.1 Select Two Algorithms

To the right, the array films contains film objects which contain a title, genre, release date and runtime.

The below algorithm loops through the array to pull the film titles. Instead of a for loop, enumeration was used with the map() method.

```
const Cinema = function (films) {
  this.films = films;
};

Cinema.prototype.filmTitles = function () {
  const titlesArray = this.films.map((film) => {
    return film.title;
  });
  return titlesArray;
};
```

```
beforeEach(function() {
  moonlight = new Film('Moonlight', 'drama',
    2016, 111);
  bladeRunner = new Film('Blade Runner 2049',
    'sci-fi', 2017, 164);
  dunkirk = new Film('Dunkirk', 'history', 2017,
    96);
  blackPanther = new Film('Black Panther',
    'action', 2018, 134);
  trainspotting = new Film('T2 Trainspotting',
    'drama', 2017, 117);

  films = [moonlight, bladeRunner, dunkirk,
    blackPanther, trainspotting];
  cinema = new Cinema(films);
});
```

Project Unit
Paul McPhail Stevenson
Cohort E21

P 9.2 Select Two Algorithms

The second algorithm uses the filter() method sort through the array of film objects based on a specific property. For example, the array could be looped through sorting the data by genre.

```
Cinema.prototype.filmsByProperty = function
  (property, value) {
 const objectArray = this.films.filter((film) => {
   return film[property] === value;
 });
 return objectArray;

};
```

Project Unit
Paul McPhail Stevenson
Cohort E21

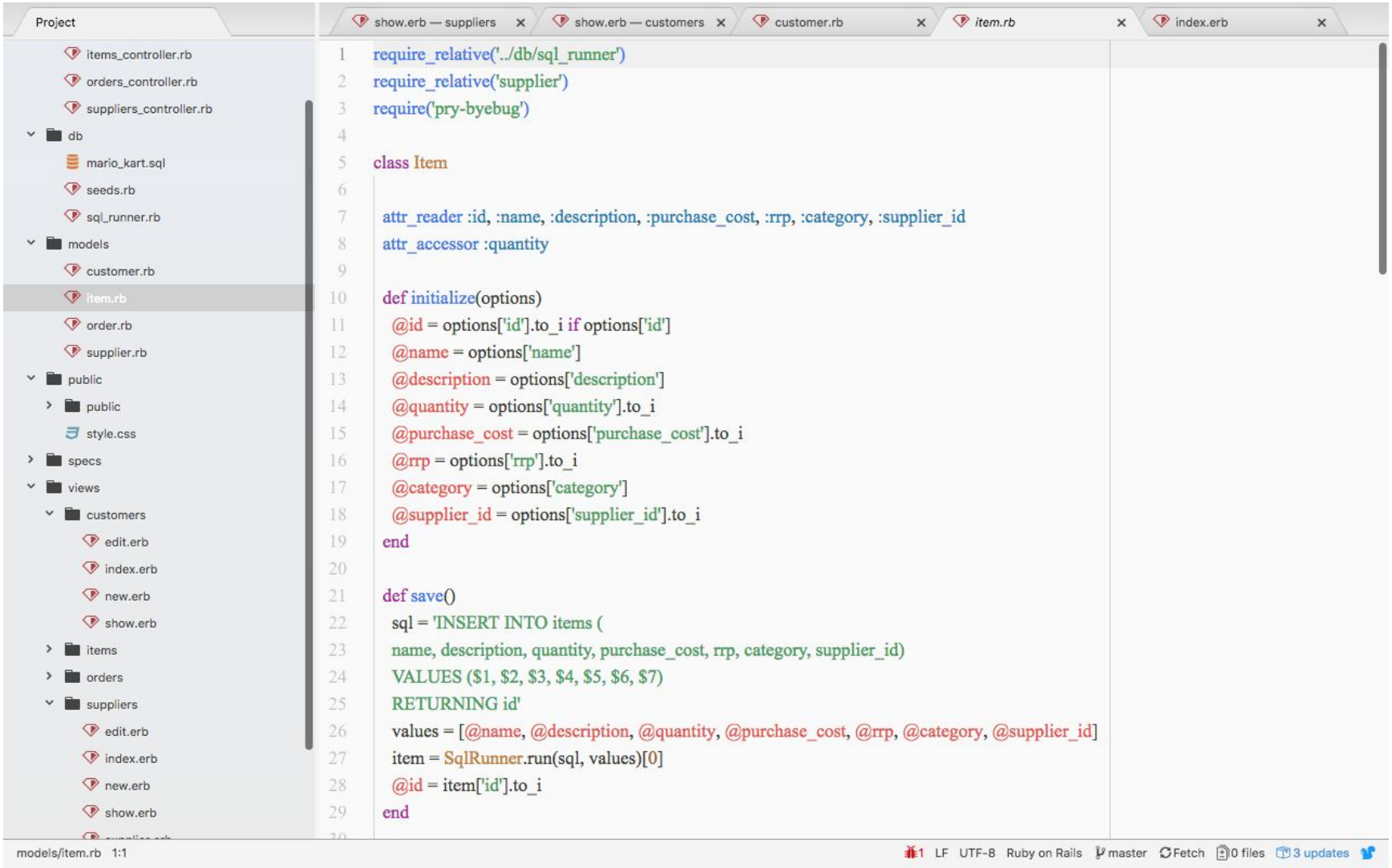
P 10 Pseudocode for a function

```
@Override
public int calculateTotalPurchaseCost() {
    return this.purchaseCost + this.additionalCost;
}
/*  calculateTotalPurchaseCost() should take in the object properties (purchaseCost and additionalCost)
    It will add these two properties together
    The result will equal the total amount spent of purchasing the object.

 */
```

Project Unit
Paul McPhail Stevenson
Cohort E21

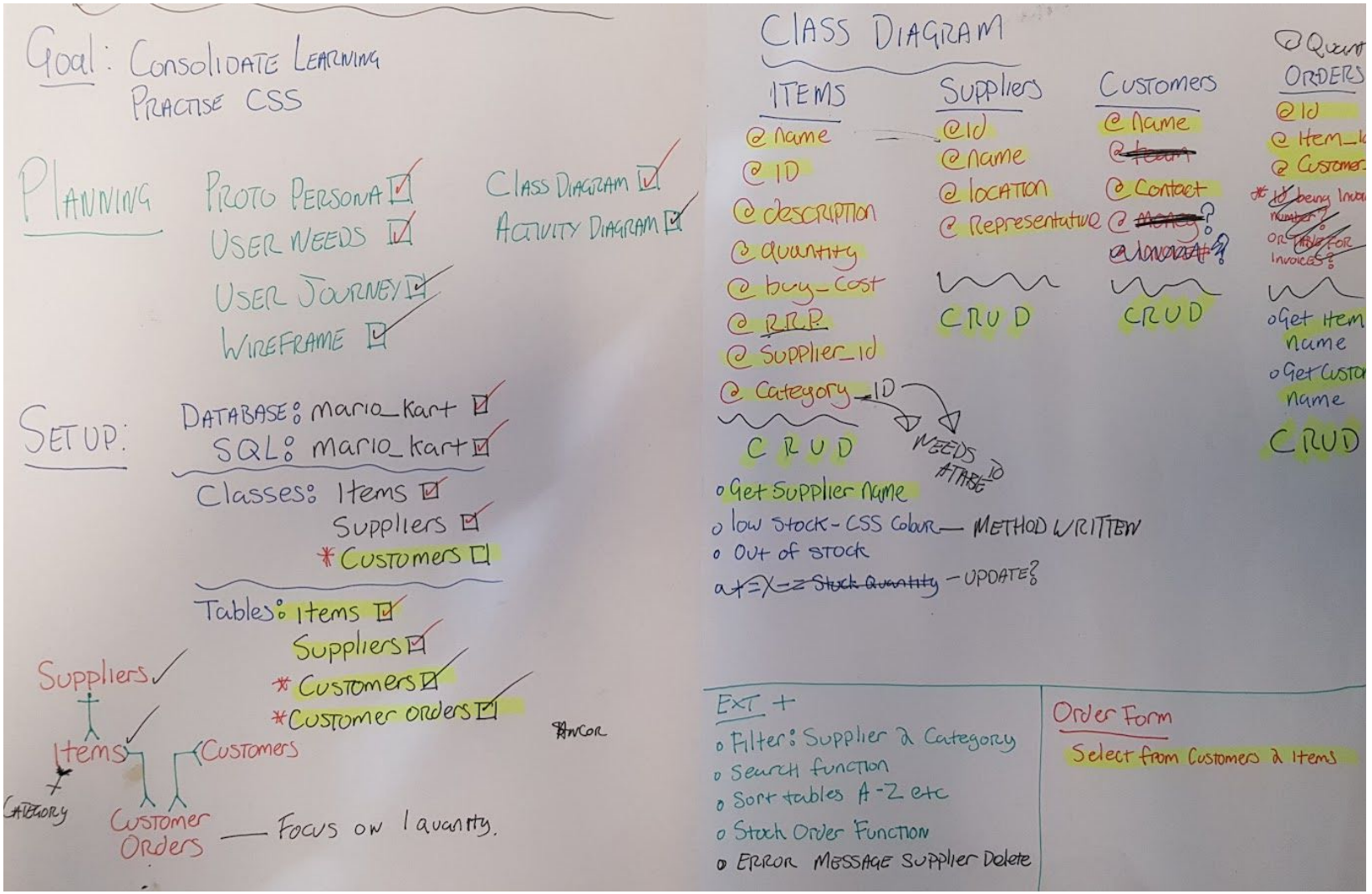
P 11 Solo Project with Github Link

https://github.com/PaulStevenson/project_mariokartshop_ruby_sql_html_css



Project Unit
Paul McPhail Stevenson
Cohort E21

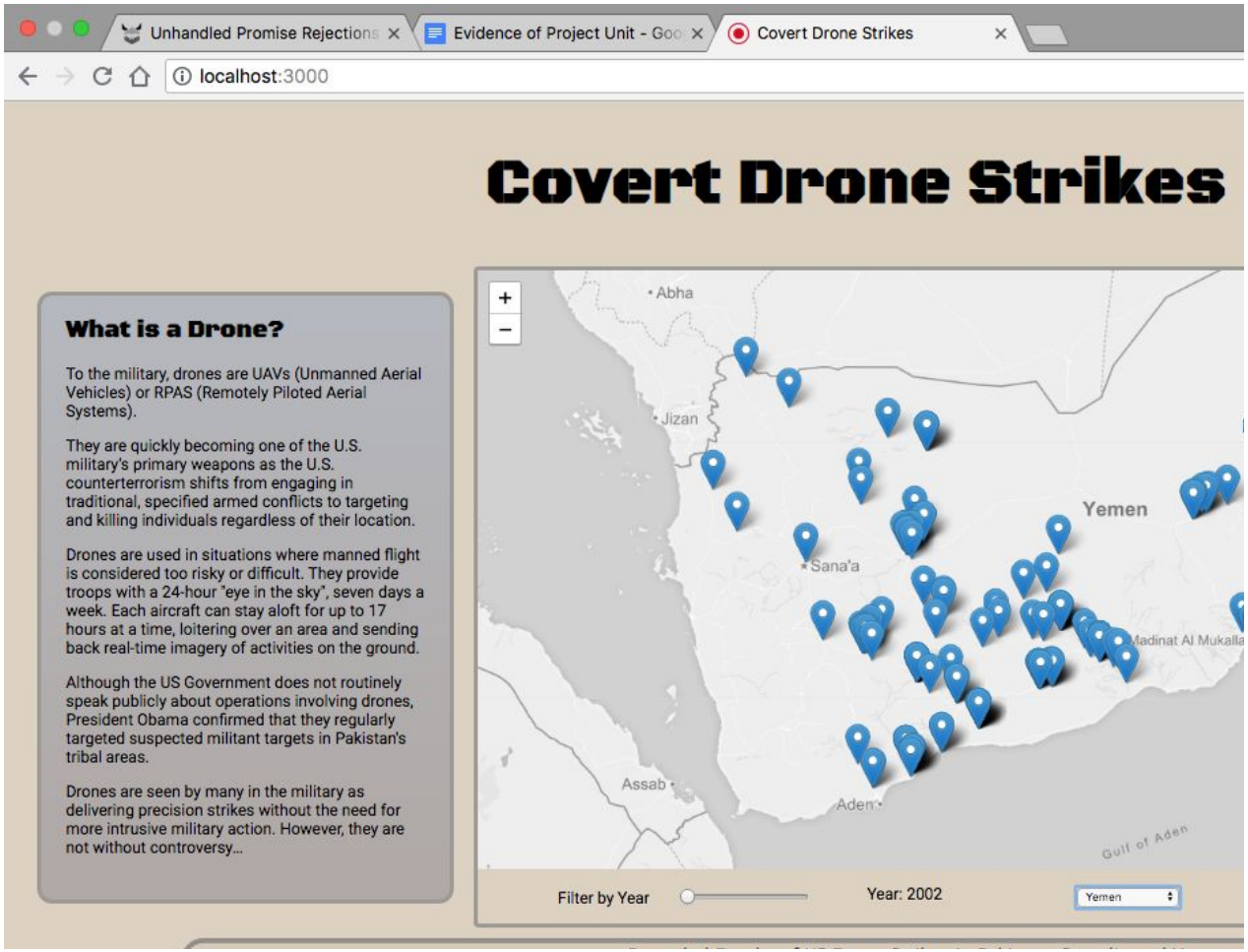
P 12 Project Planning with changes



Project Unit
Paul McPhail Stevenson
Cohort E21

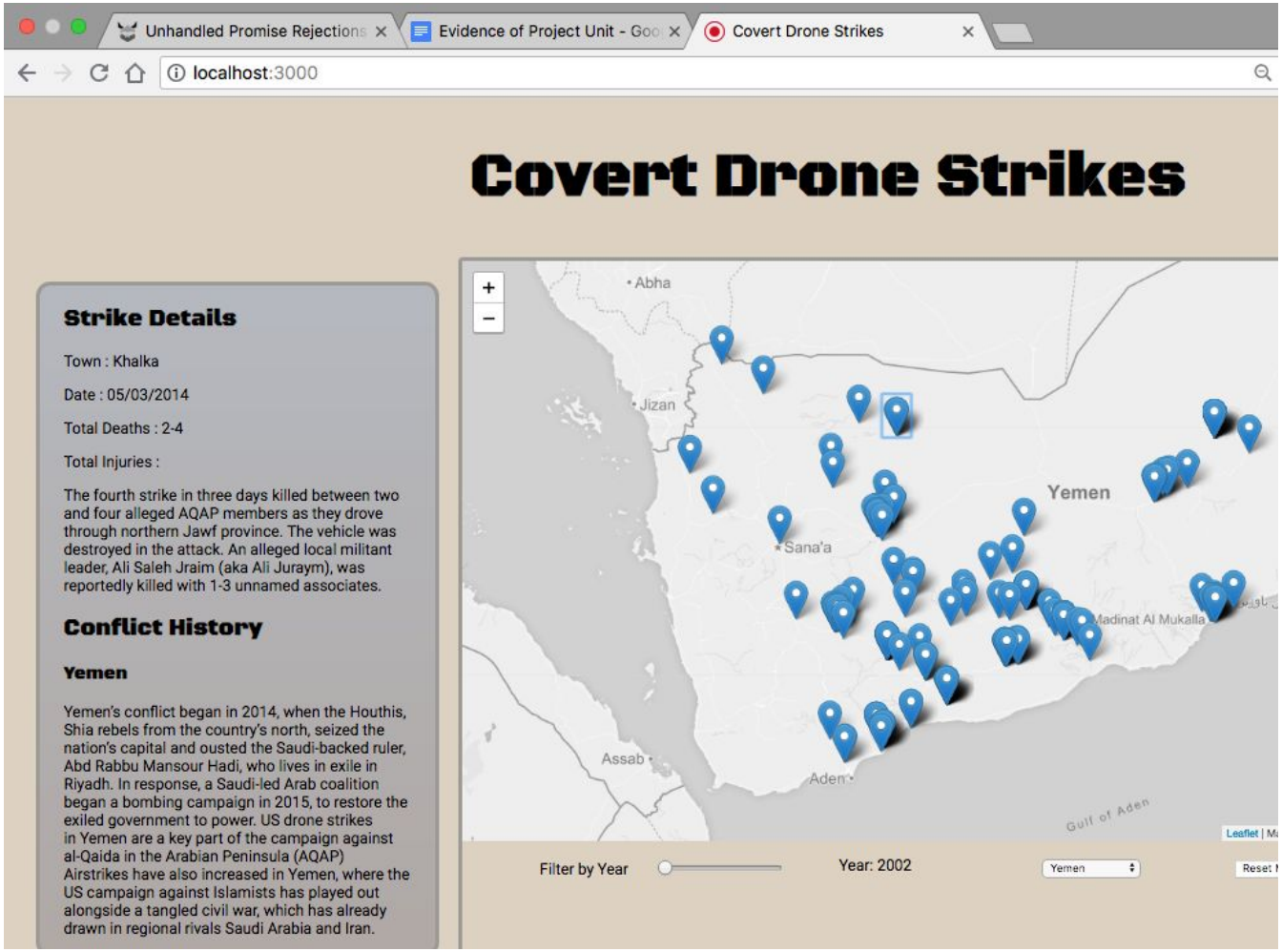
P 13 Processing User Input

User selects a marker on the map.



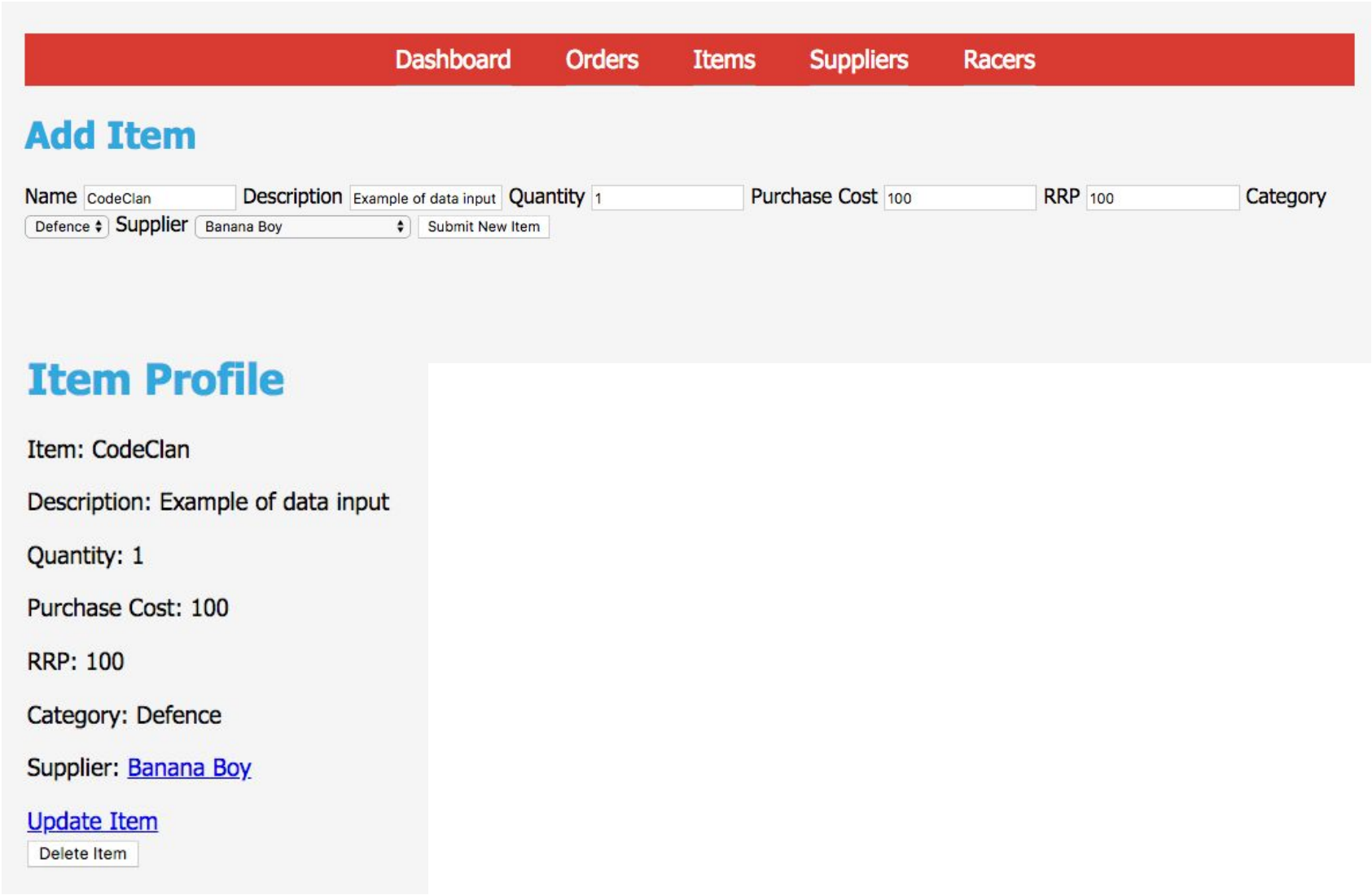
Project Unit
Paul McPhail Stevenson
Cohort E21

Once the marker is clicked, it populates the text container on the left of the map using data pulled in from the API as per the design requirement.



Project Unit
Paul McPhail Stevenson
Cohort E21

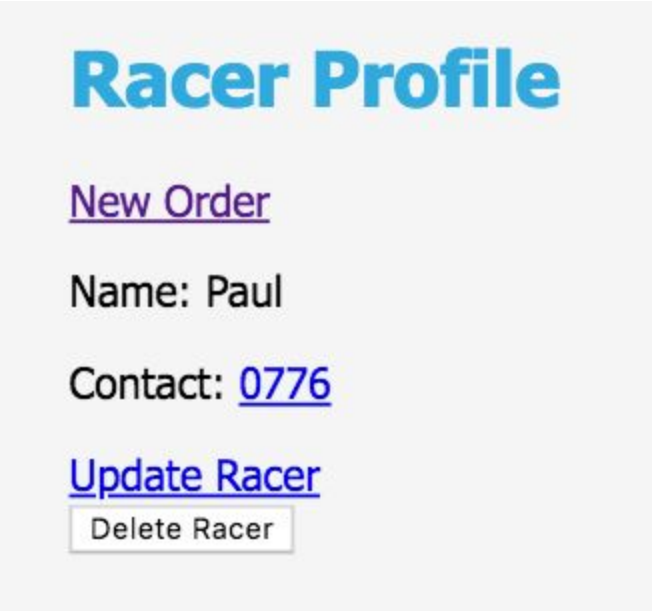
P 14 Data Persistence



Project Unit
Paul McPhail Stevenson
Cohort E21

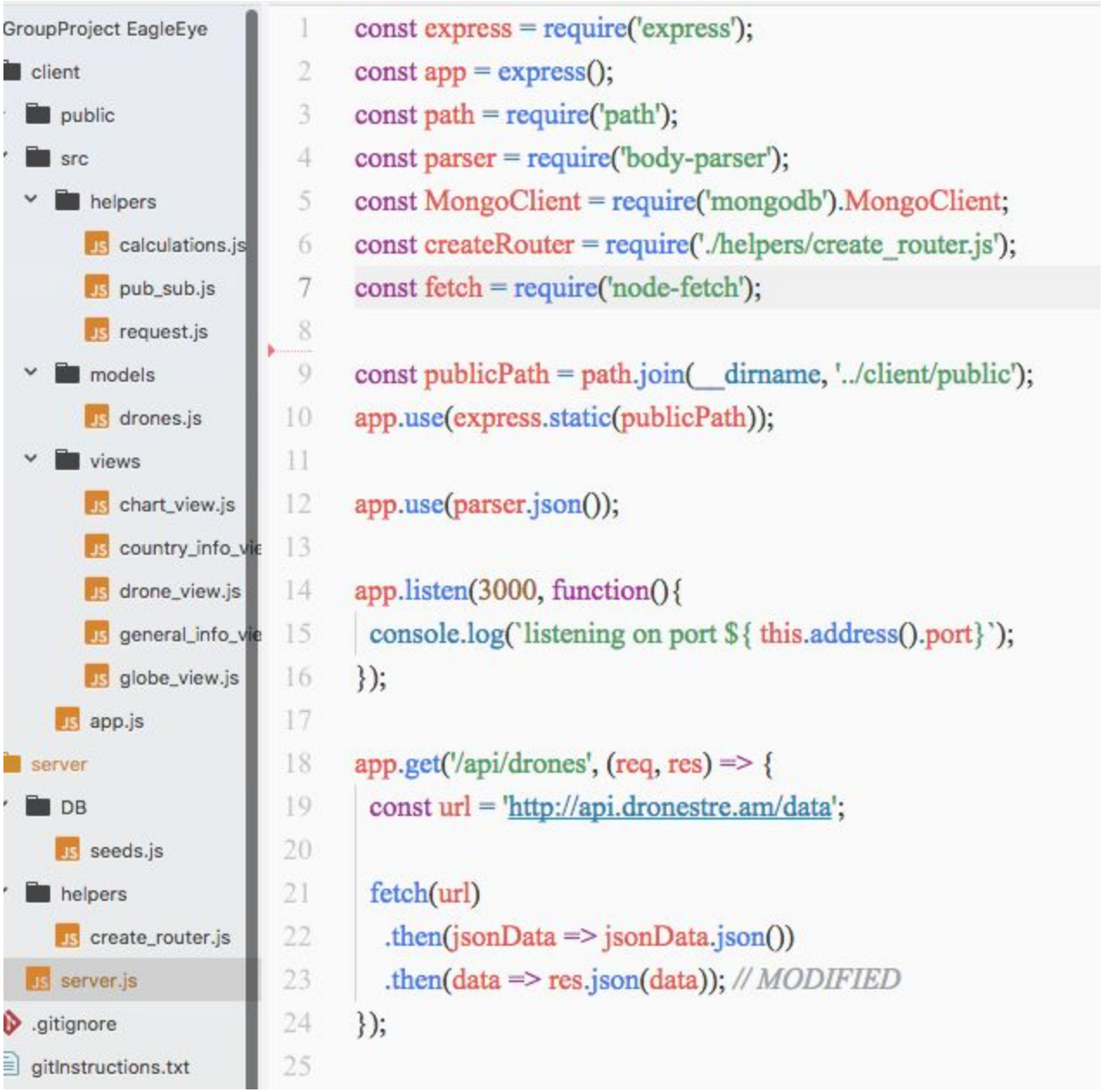
P 15 Show the correct output of results and feedback to user.

The delete button removes a racers profile from the table ‘Existing Racers’

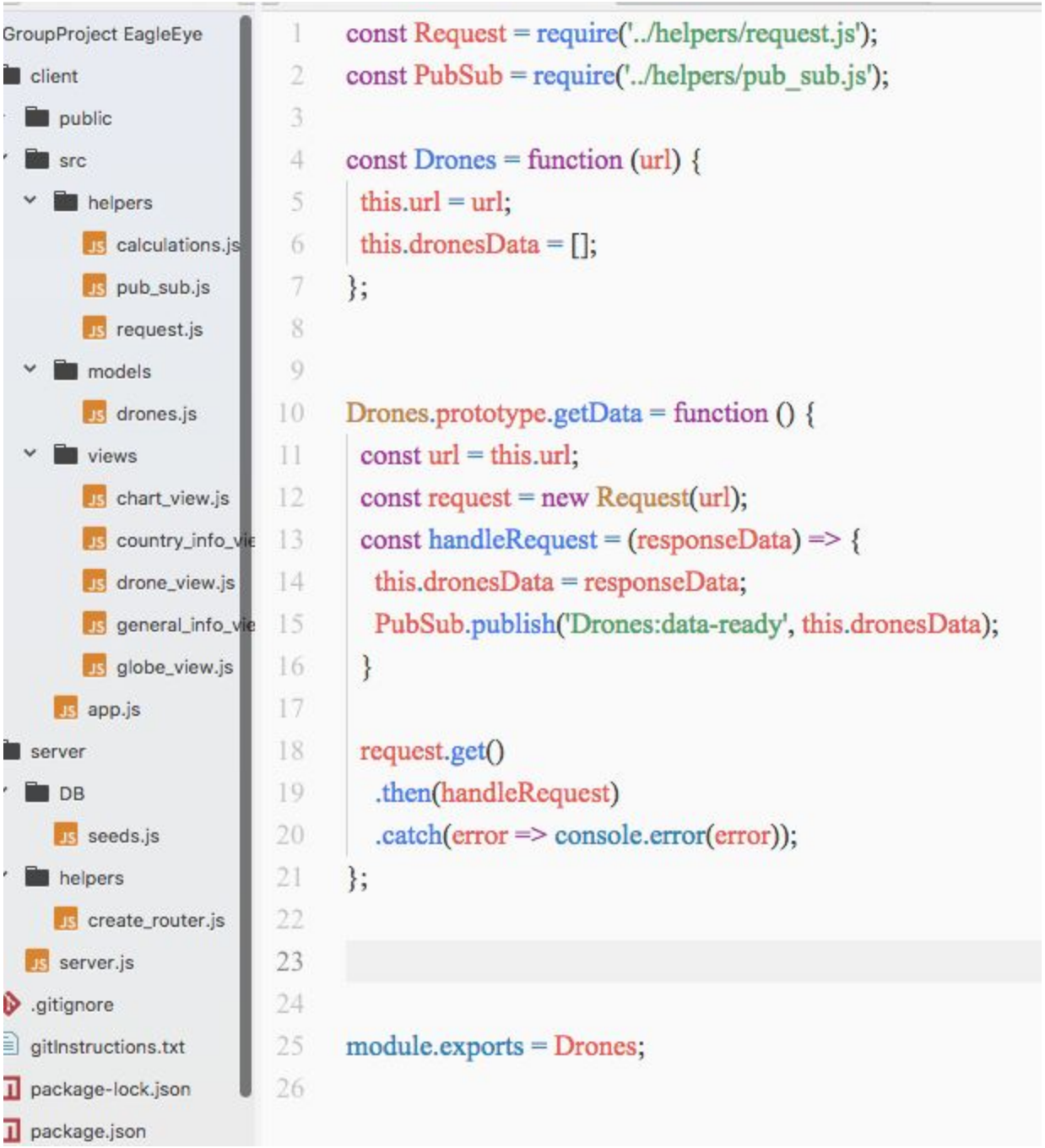


Project Unit
Paul McPhail Stevenson
Cohort E21

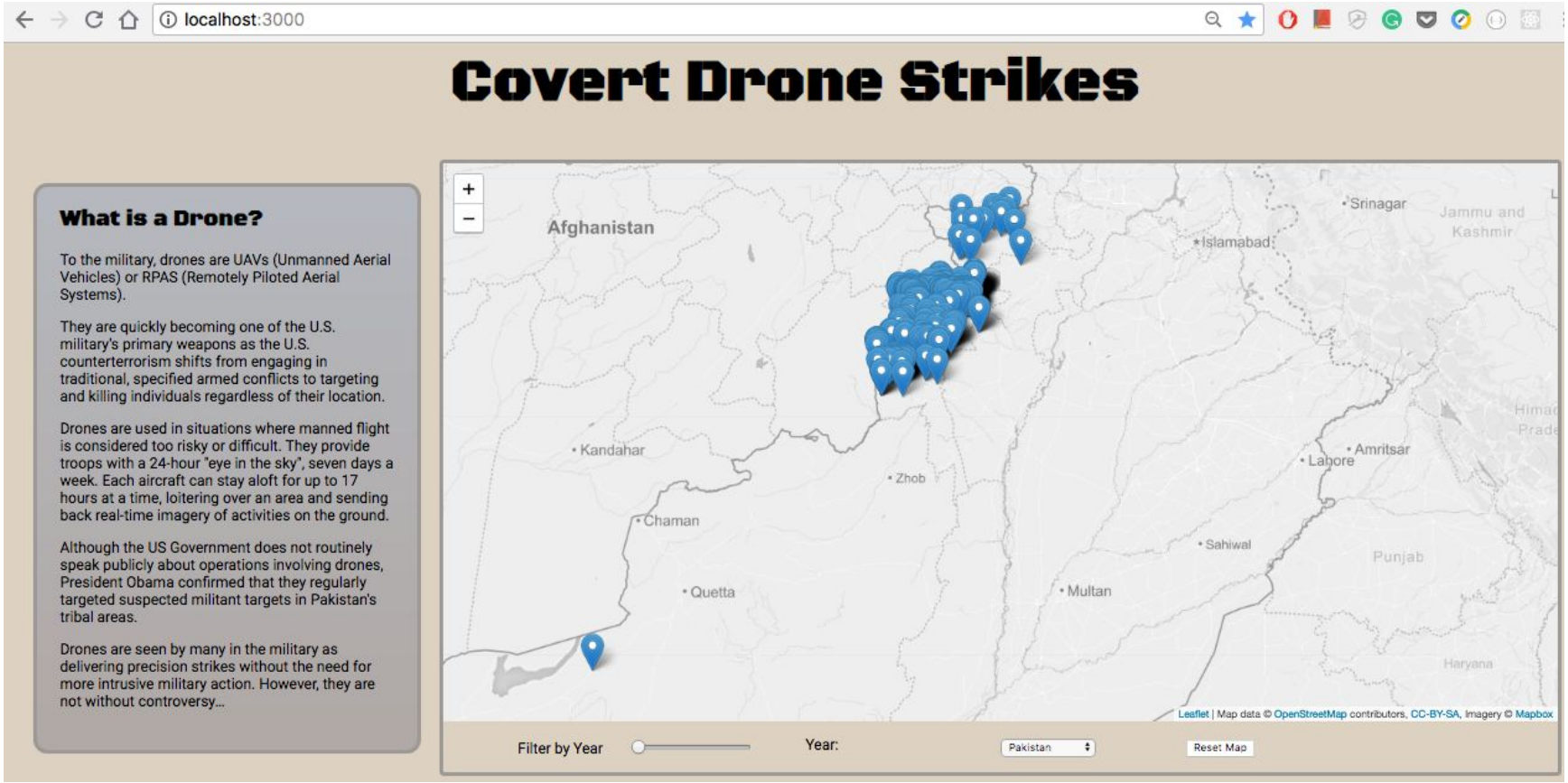
P 16 Use of an API



Project Unit
Paul McPhail Stevenson
Cohort E21



Project Unit
Paul McPhail Stevenson
Cohort E21



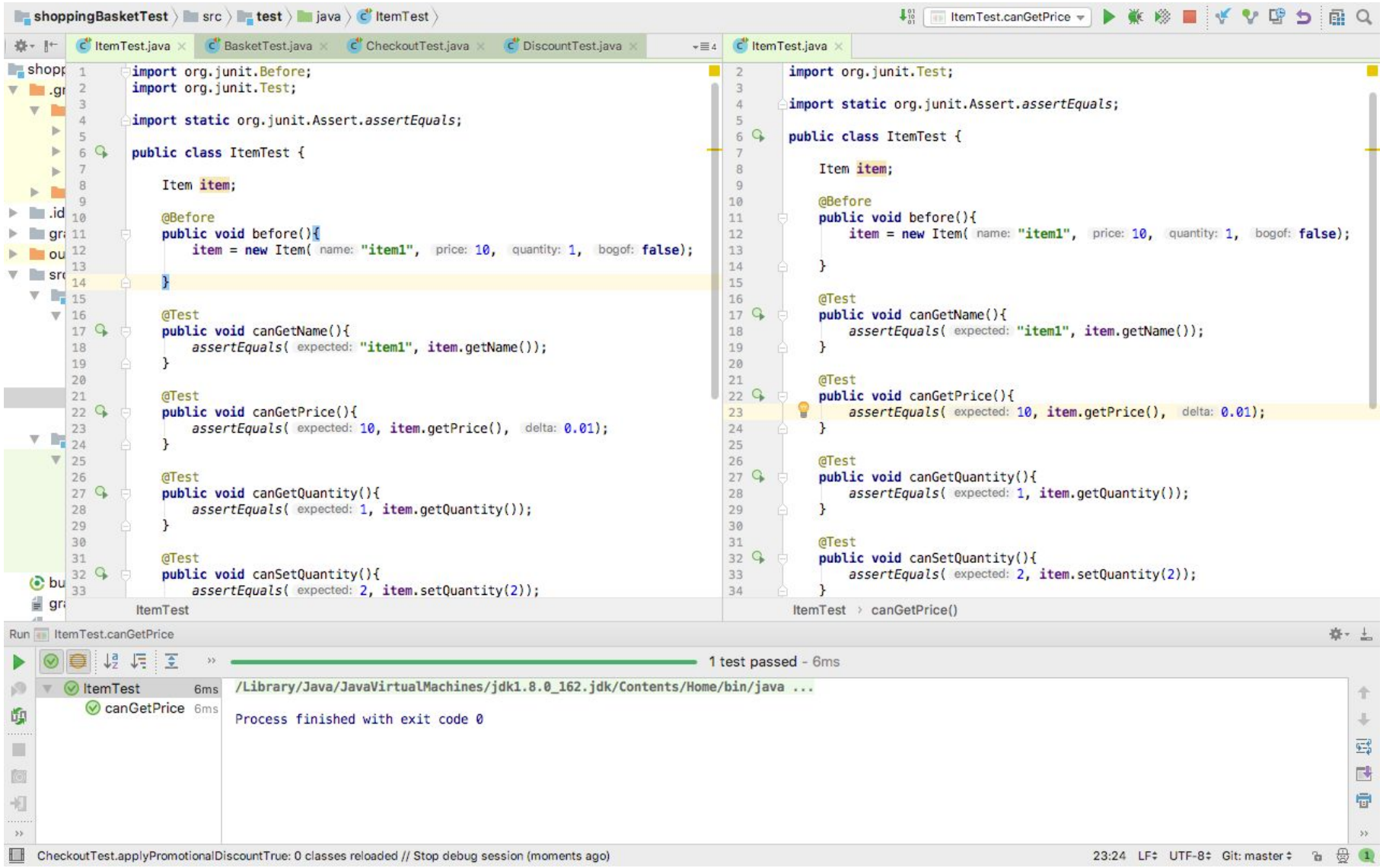
Project Unit
Paul McPhail Stevenson
Cohort E21

P 17 Bug Tracking Report

Test	**Fail**	**Solution**	**Pass/Fail**
Should be able to render markers on the map			Pass
Clicking a marker should populate the ‘Drone Strike’ info view			Pass
The slider should show the the markers rendering in chronological order	Failed	Formatted the Date key from the API so that it could be used more effectively	Pass
Selecting a country from the dropdown menu should zoom the map into that country	Failed	Seperated the map and marker rendering methods	Pass
The reset button should reset the map to default	Failed	Set the event to re-render the map zoom to centre	Pass

Project Unit
Paul McPhail Stevenson
Cohort E21

P 18 Demonstrate Testing In A Programme



Project Unit
Paul McPhail Stevenson
Cohort E21