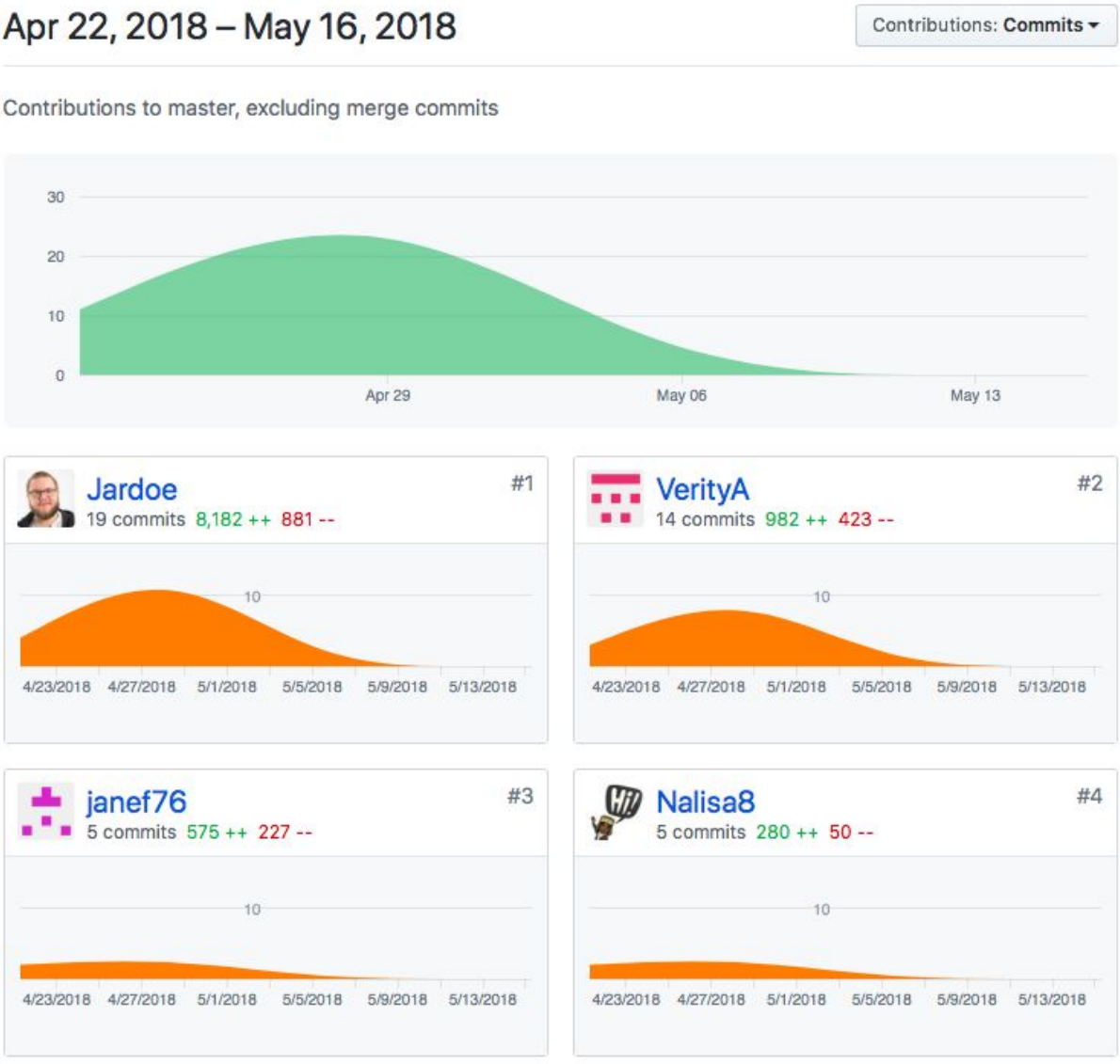
Evidence for Project Unit

Jane Flucker
E19

P. 1 Github contributors page



P. 2 Project brief

Route Planner

Visit Scotland are look for ways to encourage people to walk and cycle. Your task is to create an app that allows users to search for cycling and hiking routes, view routes on a map, save routes to a wishlist and mark a route done.

You could use GoogleMaps Directions API:

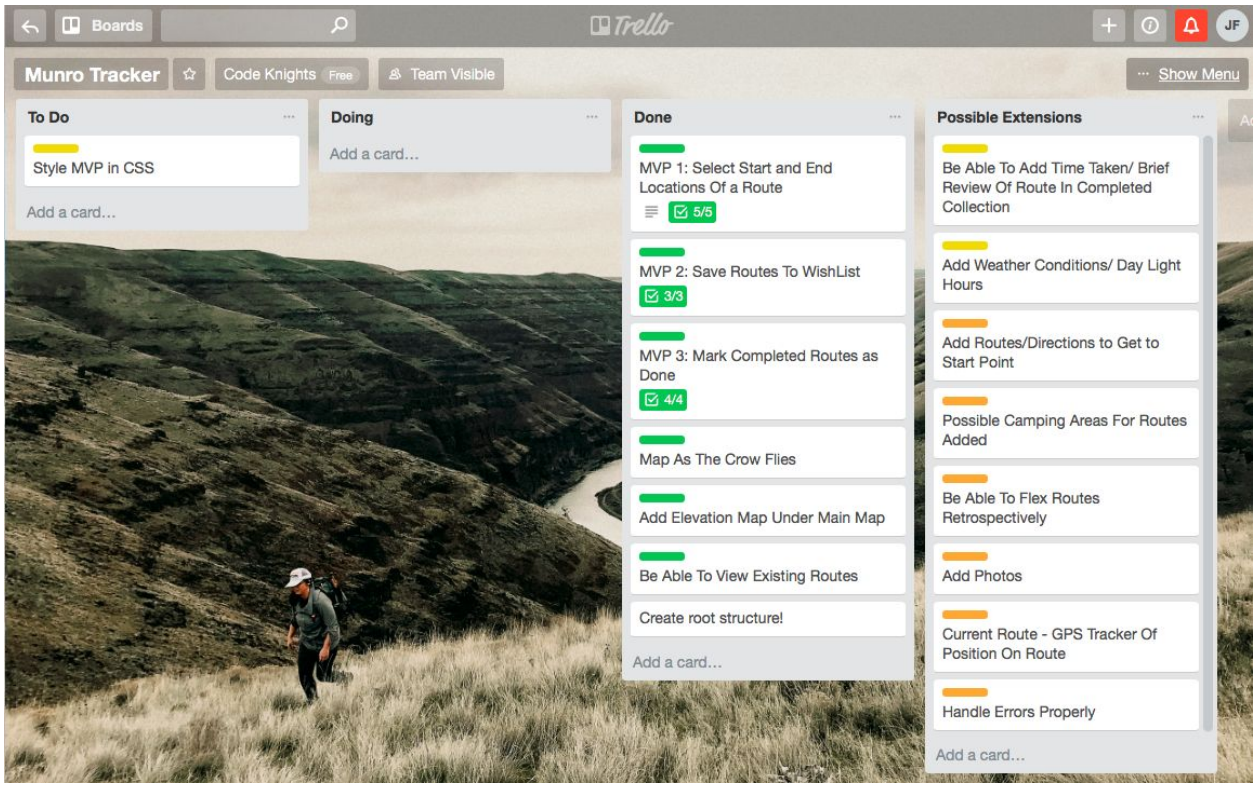
- https://developers.google.com/maps/documentation/directions/

MVP

Users should be able to:

- Select start and finish locations for their route
- Save routes to a wishlist
- Mark completed routes as 'done'

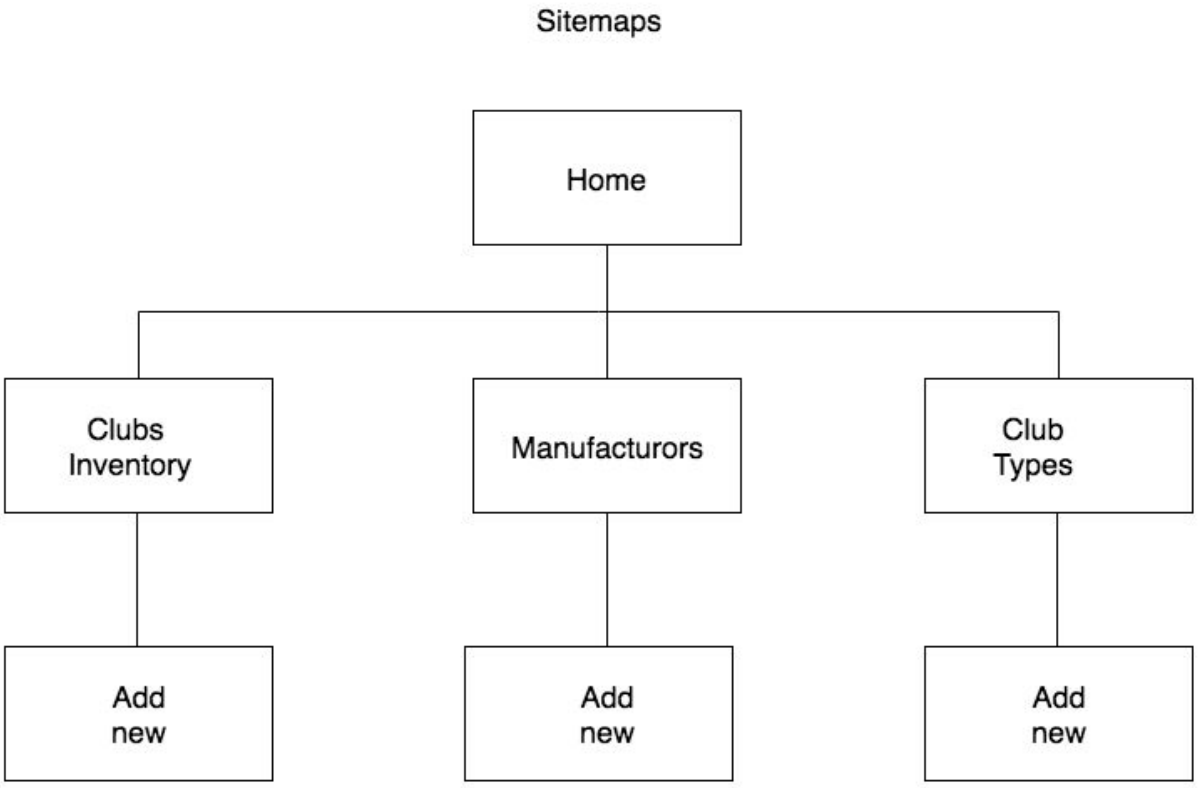
P. 3 Use of Trello



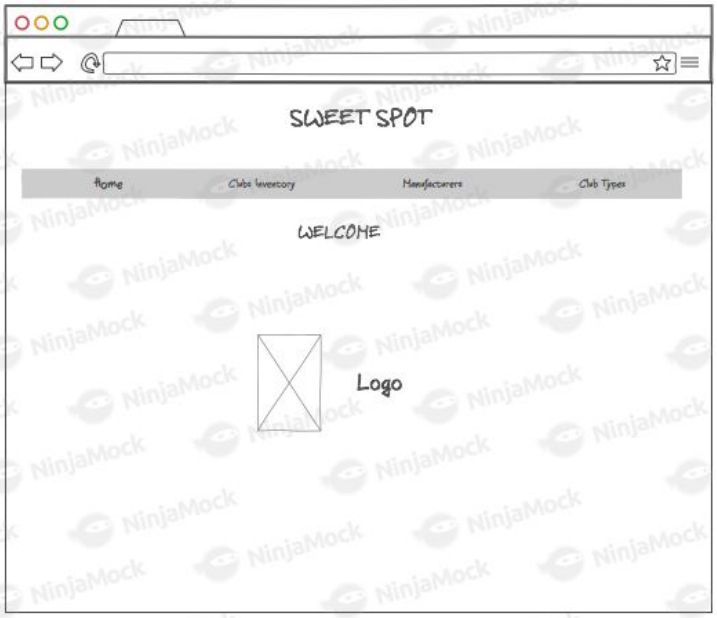
P. 4 Acceptance criteria

Acceptance Criteria	Expected Result/Output	Pass / Fail
User can see the map of Scotland on app.	On the app starting, a map is loaded and displayed.	PASS
User can select start and end points.	Using the mouse click to select the start and end points on the map shown by marker.	PASS
User can save the route into the wish-list	Once the two markers on map click the save button and this route details are saved to and displayed in the wish-list table on screen.	PASS
User can choose to mark a route in wish-list as complete.	Once a route is marked complete it should be displayed in completed list table on screen.	PASS

P. 5 User sitemap

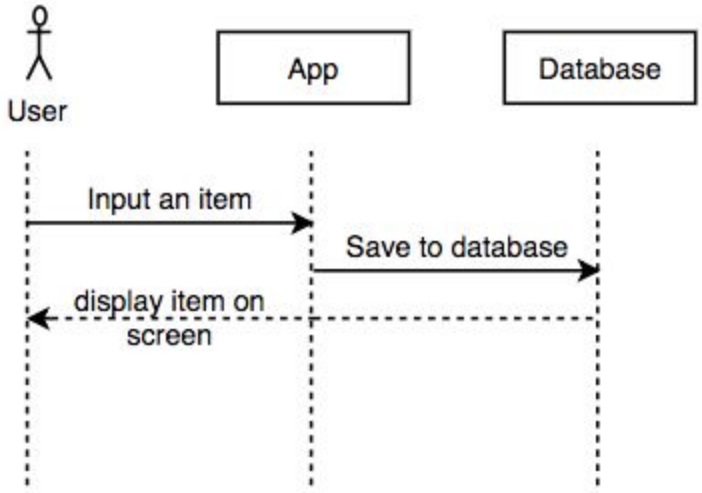


P. 6 Wireframe designs

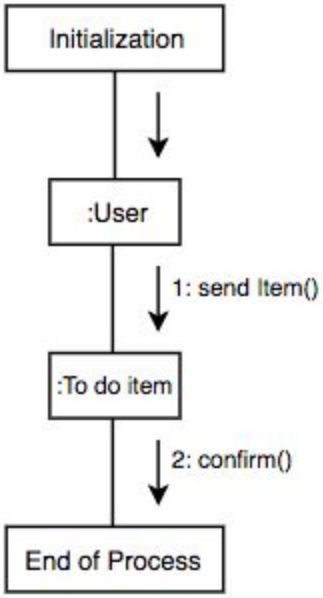


P. 7 System interactions designs

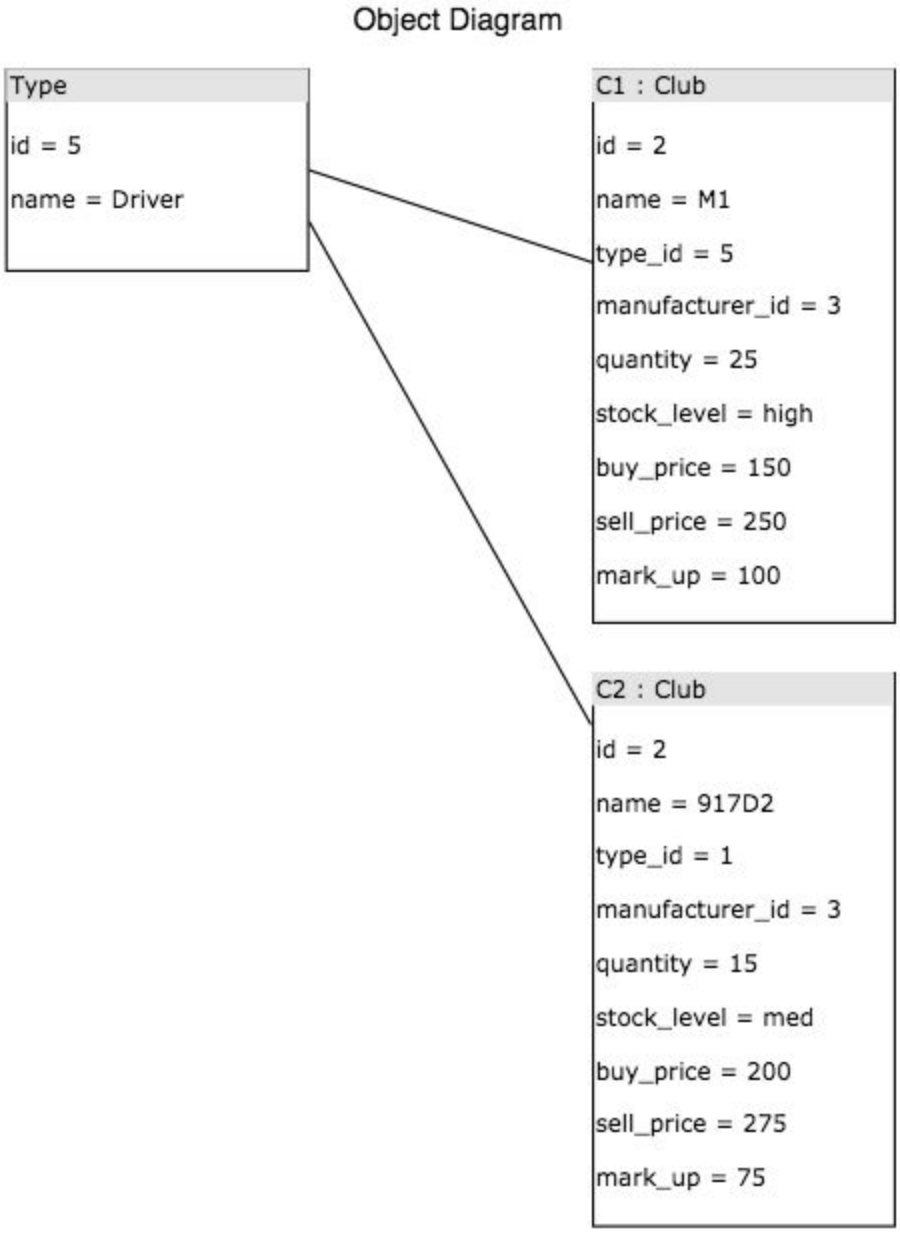
Sequence Diagrams



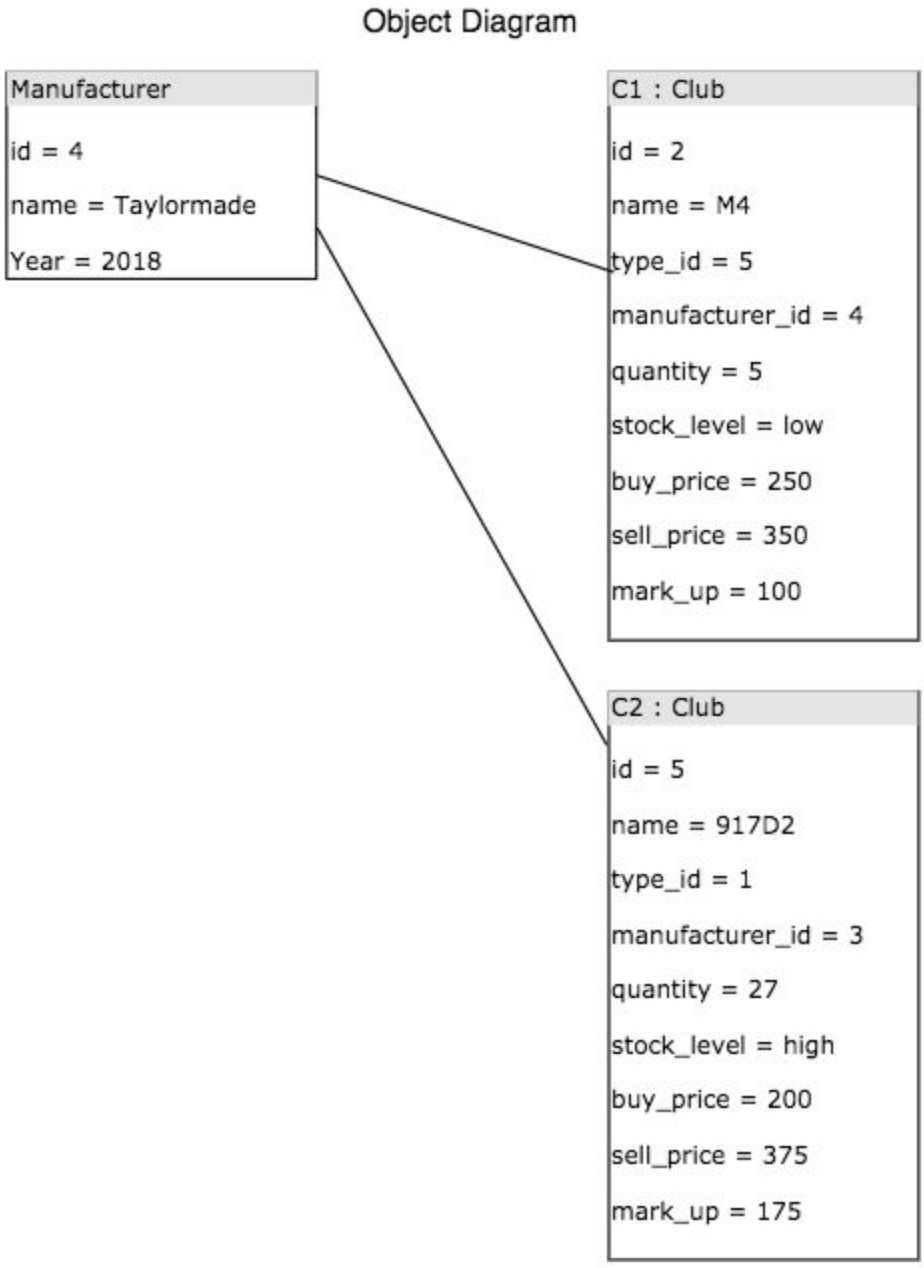
Collaboration Diagram



P. 8 Two object diagrams



P. 8 Two object diagrams...cont



P. 9 Choice of two algorithms

```
<% for club in @clubs %>
<tr>
  <td><%= club.name %></td>
  <td><%= Type.type_name(club.type_id) %></td>
  <td><%= Manufacturer.manufacturer_name(club.manufacturer_id) %></td>
  <td><%= club.quantity %></td>
  <td><%= club.level_indicator %></td>
  <td><%= club.buy_price %></td>
  <td><%= club.sell_price %></td>
  <td><%= club.mark_up_value %></td>
  <td><a href="/clubs/<%= club.id %>" class="update_buttons">Click Here</a></td
</tr>
<% end %>
```

I have chosen this code as it brings together the three database tables for displaying all the information for the user to display on app.

```
def level_indicator()
  if @quantity > 0 && @quantity <= 10
    return "low"
  elsif @quantity > 10 && @quantity <= 20
    return "med"
  else
    return "high"
  end
end
```

I have chosen this code as it is dependant on the quantity as to the information it holds in table for a display on app that is at a glance for user.

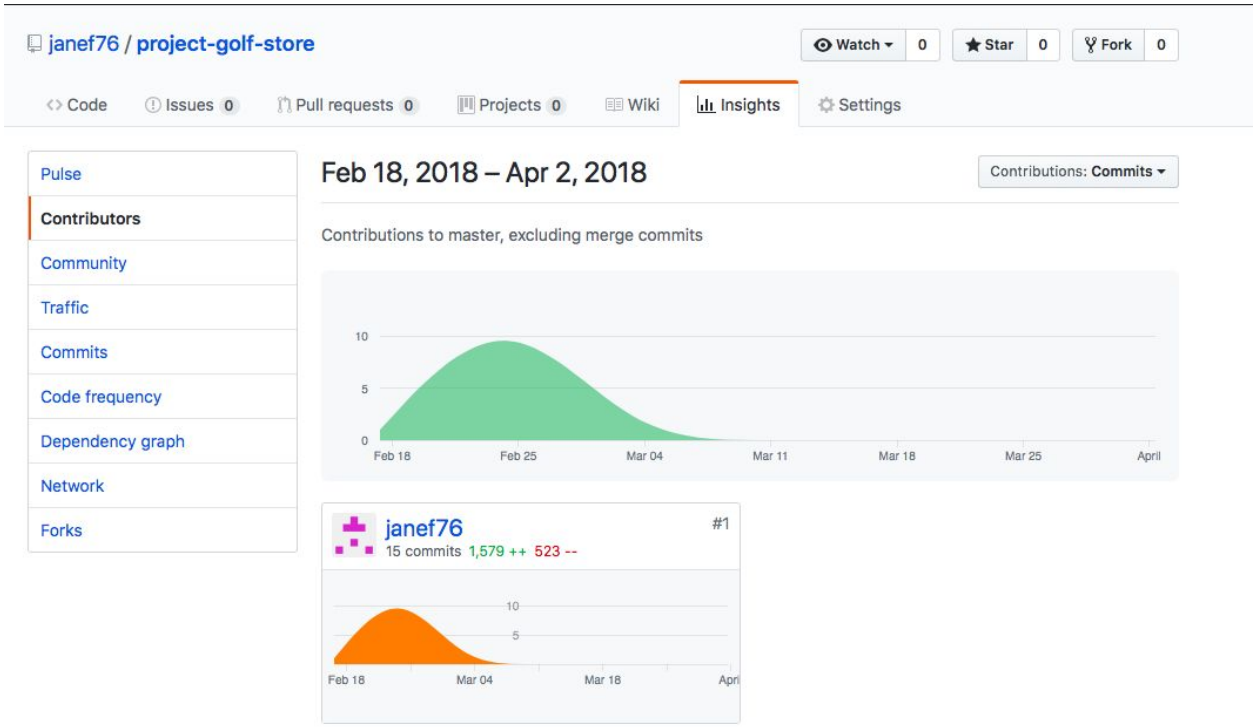
P.10 Pseudocode

Pseudocode — ~/my_stuff/PDA

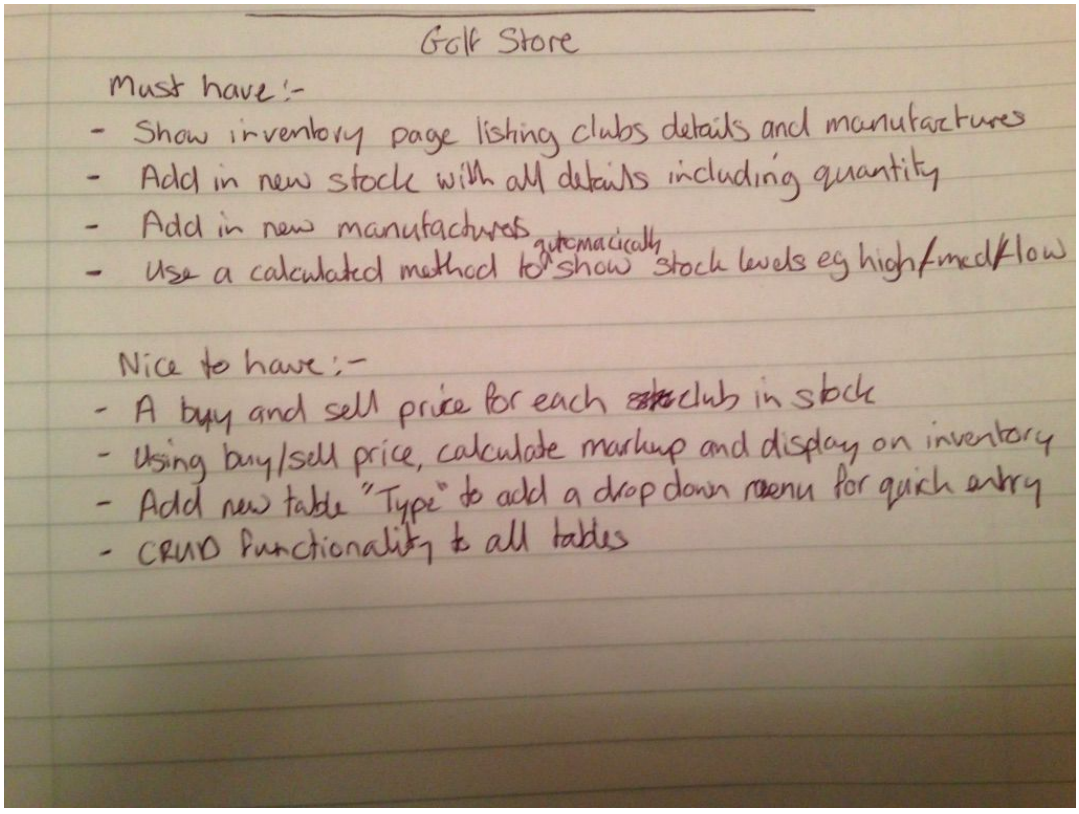
Pseudocode

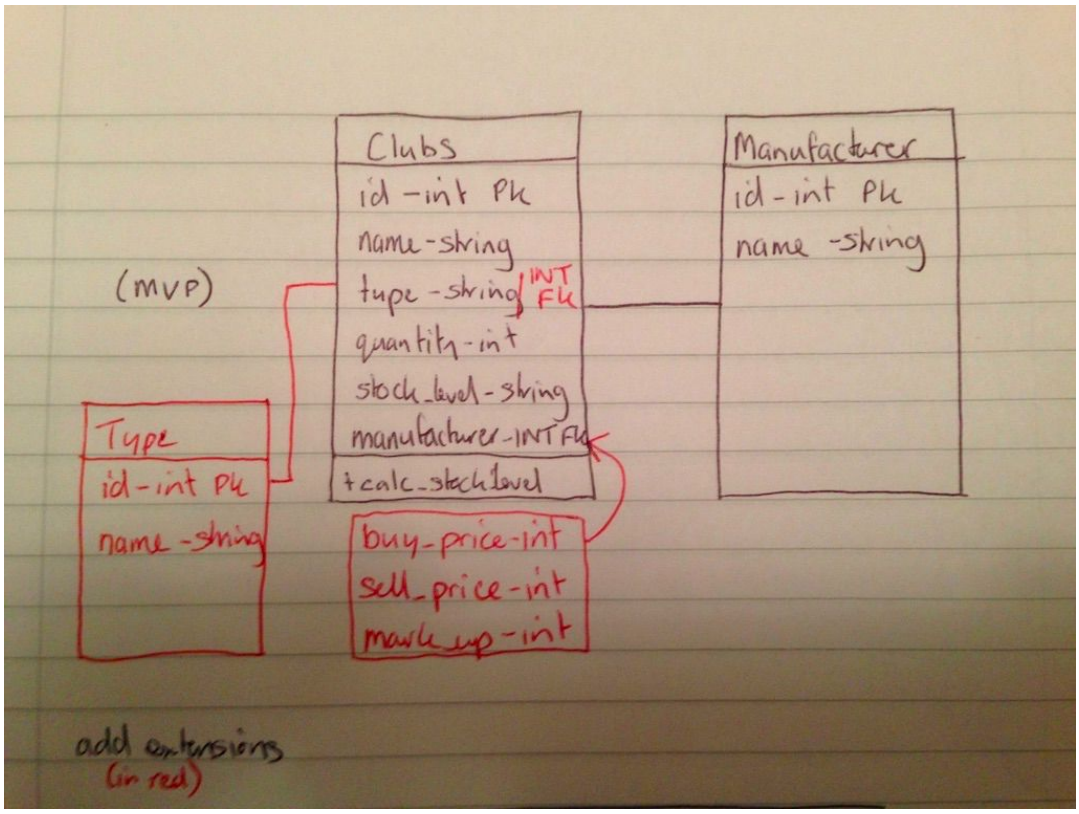
```
1  Pseudocode for a function
2
3  Start function level indicator
4  if quantity passed in is between 0 and 10
5  show stock level as low
6  if quantity passed in is between 11 and 20
7  show stock level as med
8  otherwise
9  show stock level as high
10 End function
11
```

P.11 Github link to one of your projects

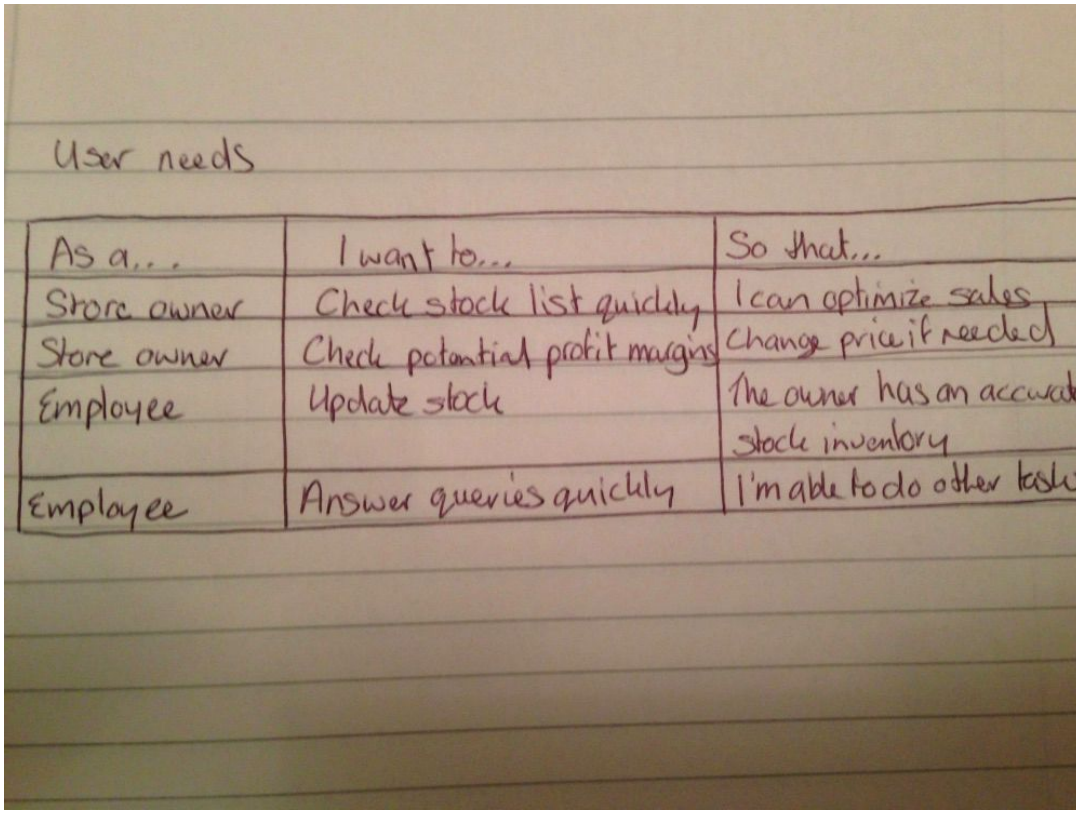


P.12 Screenshot of your planning and the different stages of development to show changes



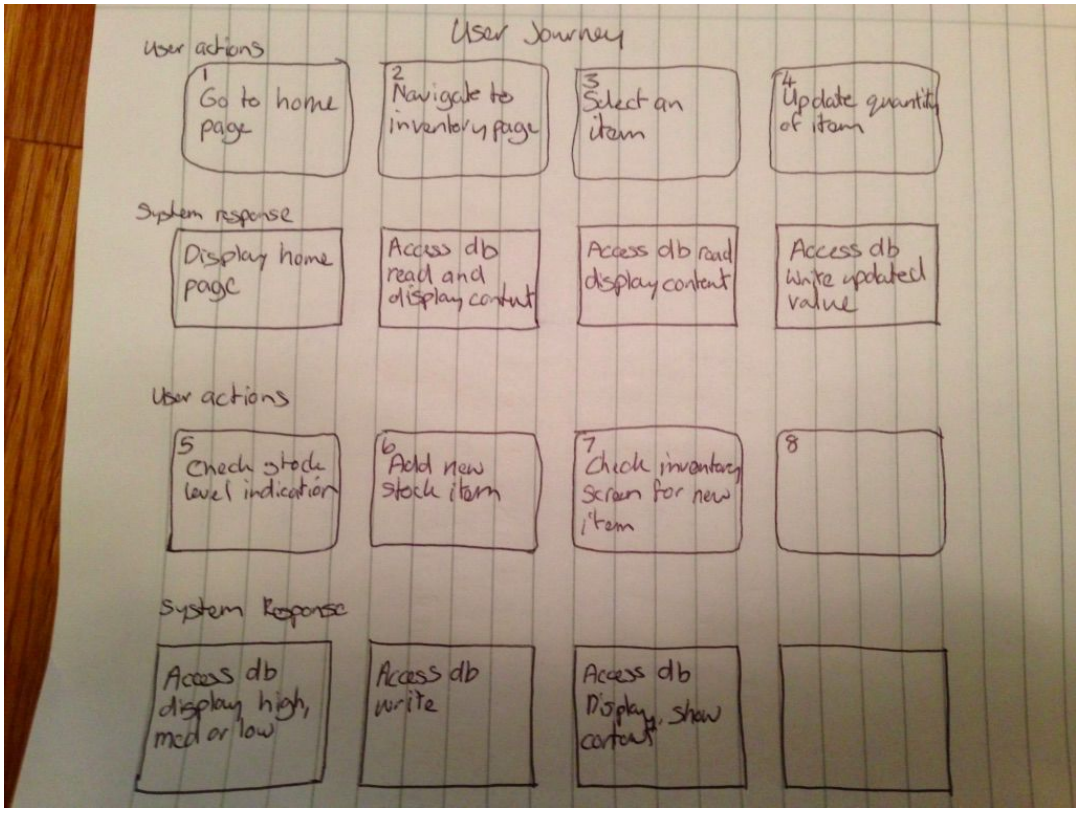


P.12 Screenshot of your planning and the different stages of development to show changes...cont

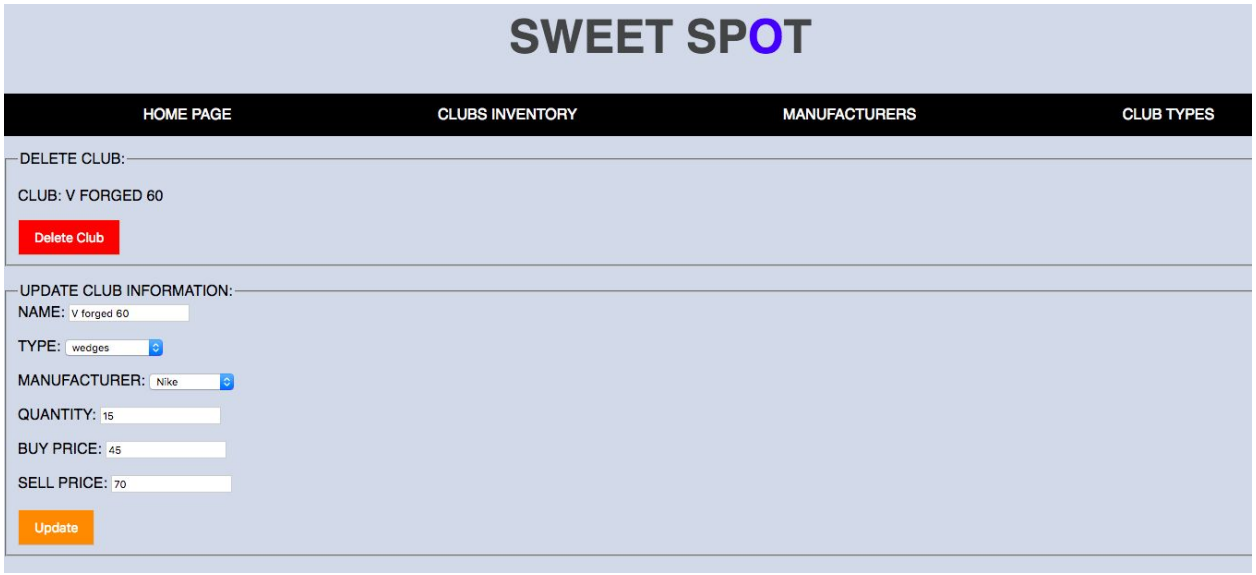


User needs

As a...	I want to...	So that...
Store owner	Check stock list quickly	I can optimize sales
Store owner	Check potential profit margins	Change price if needed
Employee	Update stock	The owner has an accurate stock inventory
Employee	Answer queries quickly	I'm able to do other tasks



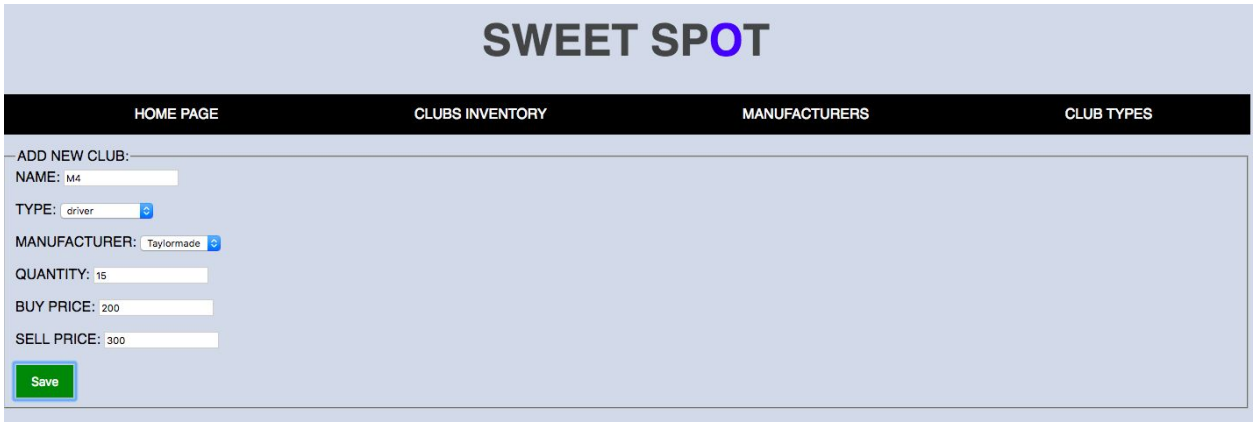
P.13 User input



P.13 User input...cont

P.14 Interaction with data persistence

P.14 Interaction with data persistence...cont



P.15 User output result

SWEET SPOT

HOME PAGE | CLUBS INVENTORY | MANUFACTURERS | CLUB TYPES

ALL CLUBS

CLUB NAME	CLUB TYPE	MANUFACTURER NAME	QUANTITY	STOCK INDICATOR	BUY PRICE	SELL PRICE	MARK UP	UPDATE/DELETE
M1 DRIVER	DRIVER	TAYLORMADE	15	MED	300	450	150	CLICK HERE
917 3 WOOD	FAIRWAY WOODS	TITLEIST	2	LOW	200	250	50	CLICK HERE
XJ20 3-PW	IRONS	CALLAWAY	25	HIGH	400	600	200	CLICK HERE
FX 2	PUTTERS	BEN SAYERS	11	MED	20	35	15	CLICK HERE
V FORGED 60	WEDGES	NIKE	15	MED	45	70	25	CLICK HERE
M4	DRIVER	TAYLORMADE	15	MED	200	300	100	CLICK HERE

ADD NEW CLUB

SWEET SPOT

HOME PAGE | CLUBS INVENTORY | MANUFACTURERS | CLUB TYPES

DELETE CLUB:

CLUB: FX 2

Delete Club

SWEET SPOT

HOME PAGE | CLUBS INVENTORY | MANUFACTURERS | CLUB TYPES

ALL CLUBS

CLUB NAME	CLUB TYPE	MANUFACTURER NAME	QUANTITY	STOCK INDICATOR	BUY PRICE	SELL PRICE	MARK UP	UPDATE/DELETE
M1 DRIVER	DRIVER	TAYLORMADE	15	MED	300	450	150	CLICK HERE
917 3 WOOD	FAIRWAY WOODS	TITLEIST	2	LOW	200	250	50	CLICK HERE
XJ20 3-PW	IRONS	CALLAWAY	25	HIGH	400	600	200	CLICK HERE
V FORGED 60	WEDGES	NIKE	15	MED	45	70	25	CLICK HERE
M4	DRIVER	TAYLORMADE	15	MED	200	300	100	CLICK HERE

ADD NEW CLUB

P.16 API being used within your program.

```
document.addEventListener('DOMContentLoaded', () => {
  const url = 'https://api.punkapi.com/v2/beers';
  makeRequest(url, requestComplete);
});
```

P.17 Bug tracking report

Bug Tracking Report

Stock level to display low if quantity 10 or below	Failed	Added in an equals-sign to the less than-sign.	Passed
Name type to be displayed on club inventory	Failed	ID showing first, fixed code to display name	Passed
Mark-up to be displayed on screen and update if price changed	Failed	Stored the value in database after performing calculation	Passed
Be able to delete an item in stock	Failed	Changed to allow deleting by ID so as only 1 item is deleted at a time	Passed
Show stock item in newest first	Failed	Changed SQL statement to order the items newest first	Passed

P.18 Testing in a program

card_game_spec.rb

```ruby
require("minitest/autorun")
require_relative("../card_game")
require_relative("../card")

class TestCardGame < MiniTest::Test

  def setup
    @card1 = Card.new("club", 1)
    @card2 = Card.new("heart", 3)
    @card3 = Card.new("spade", 6)
    @cardgame = CardGame.new()
  end

  def test_card_for_ace()
    card = @cardgame.checkforAce(@card1)
    assert_equal(true, card)
  end

  def test_card_for_not_ace()
    card = @cardgame.checkforAce(@card2)
    assert_equal(false, card)
  end

  def test_highest_card()
    value = @cardgame.highest_card(@card1, @card2)
    assert_equal(@card2, value)
  end

  def test_total_value_of_cards()
    total = CardGame.cards_total([@card1, @card2, @card3])
    assert_equal("You have a total of 10", total)
  end

end
```

P.18 Testing in a program...cont

```
➜ implementation_and_testing git:(master) ✗ ruby specs/card_game_spec.rb
specs/card_game_spec.rb:2:in `require_relative': /Users/janeflucker/codeclan_work/pda/implementation_and
_testing/card_game.rb:20: syntax error, unexpected keyword_end, expecting end-of-input (SyntaxError)
end # extra end statement, not ne
   ^
        from specs/card_game_spec.rb:2:in `<main>'
➜ implementation_and_testing git:(master) ✗
```

```
➜ implementation_and_testing git:(master) ✗ ruby specs/card_game_spec.rb
Run options: --seed 34395

# Running:

....

Finished in 0.001254s, 3189.7927 runs/s, 3189.7927 assertions/s.

4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
➜ implementation_and_testing git:(master) ✗
```