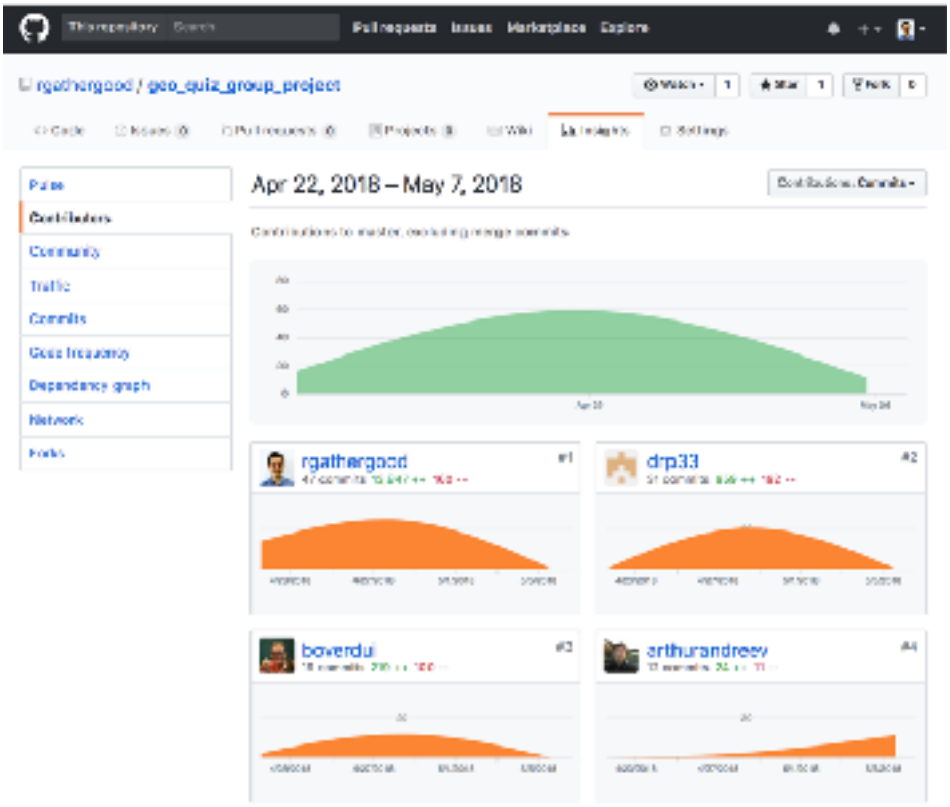
Evidence for Project Unit

Rob Gathergood
E19

P.1 - Group Project contributor's page on Github



P.2 - Group project brief

Geo Quiz Group Project

Week 14 Group Project

https://trello.com/b/mDndm0uJ/group-project

Educational App

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.

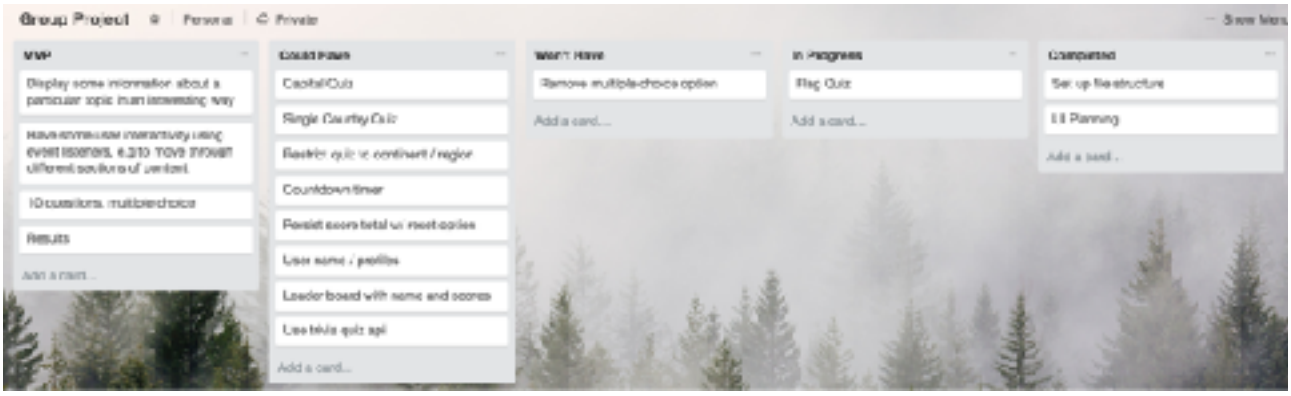
Your task is to make an MVP to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store facts. The topic of the app is your choice, but here are some suggestions you could look into:

- Interactive timeline, e.g. of the history of computer programming
- Interactive map of a historical event – e.g. World War 1, the travels of Christopher Columbus

MVP

- Display some information about a particular topic in an interesting way
- Have some user interactivity using event listeners, e.g to move through different sections of content

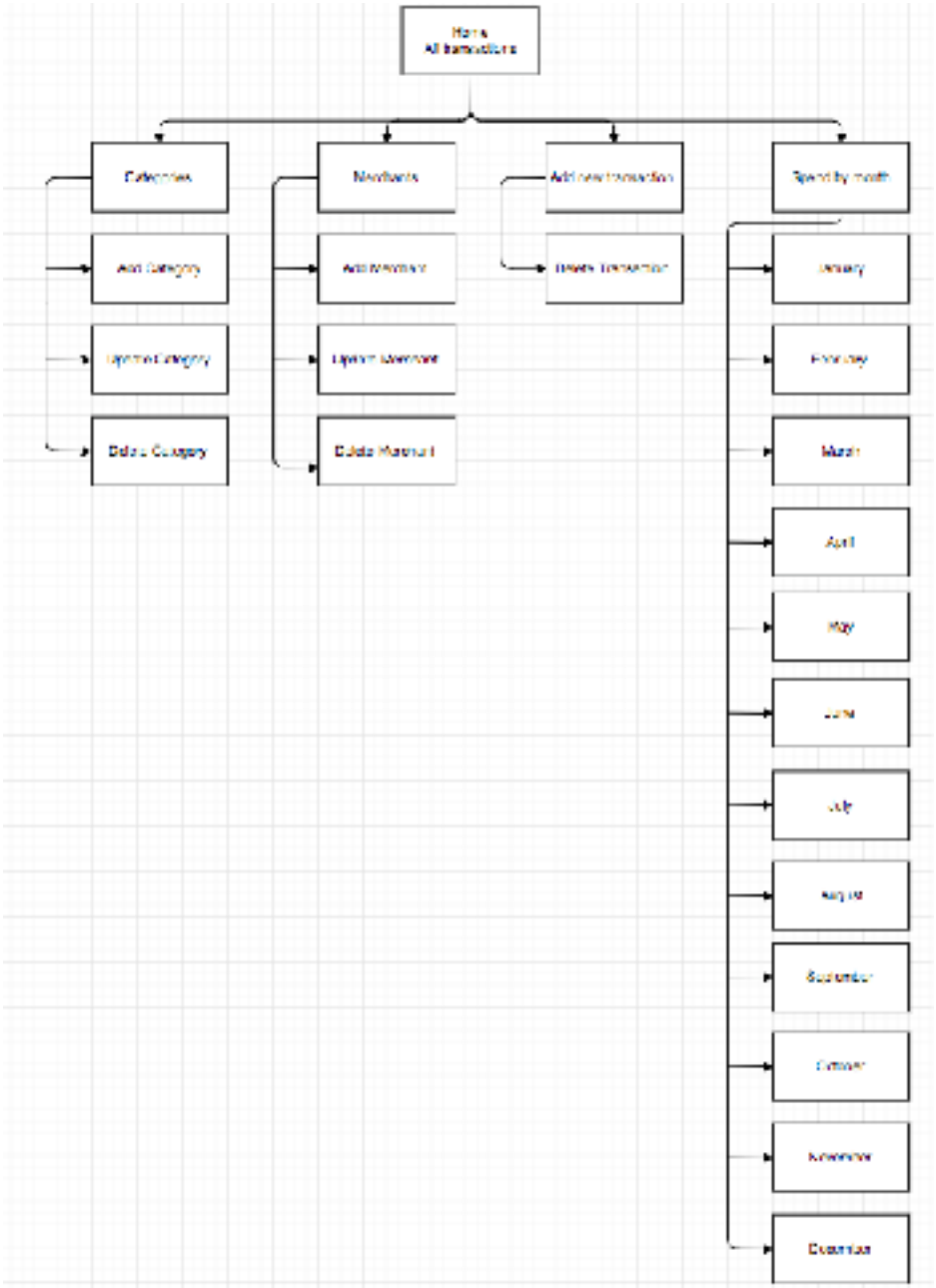
P.3 - Group Planning (trello)



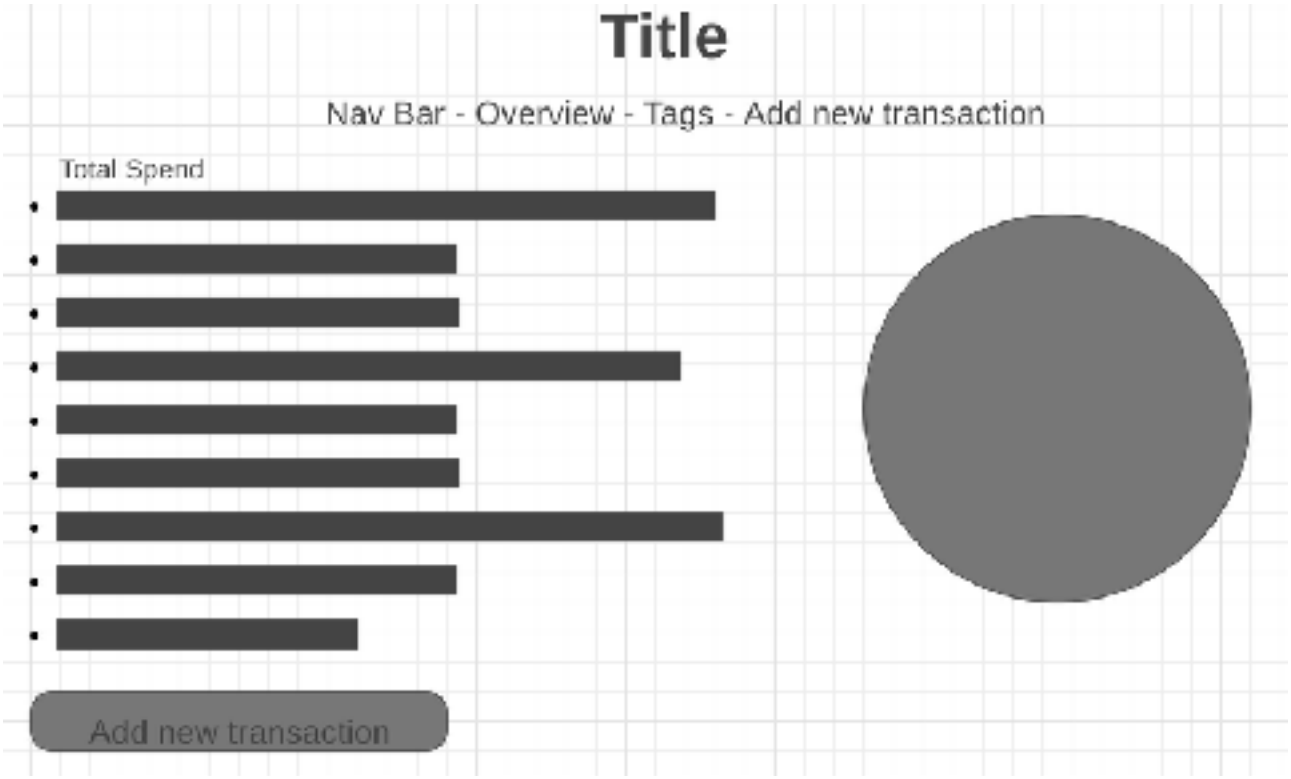
P.4 - Acceptance criteria and test plan

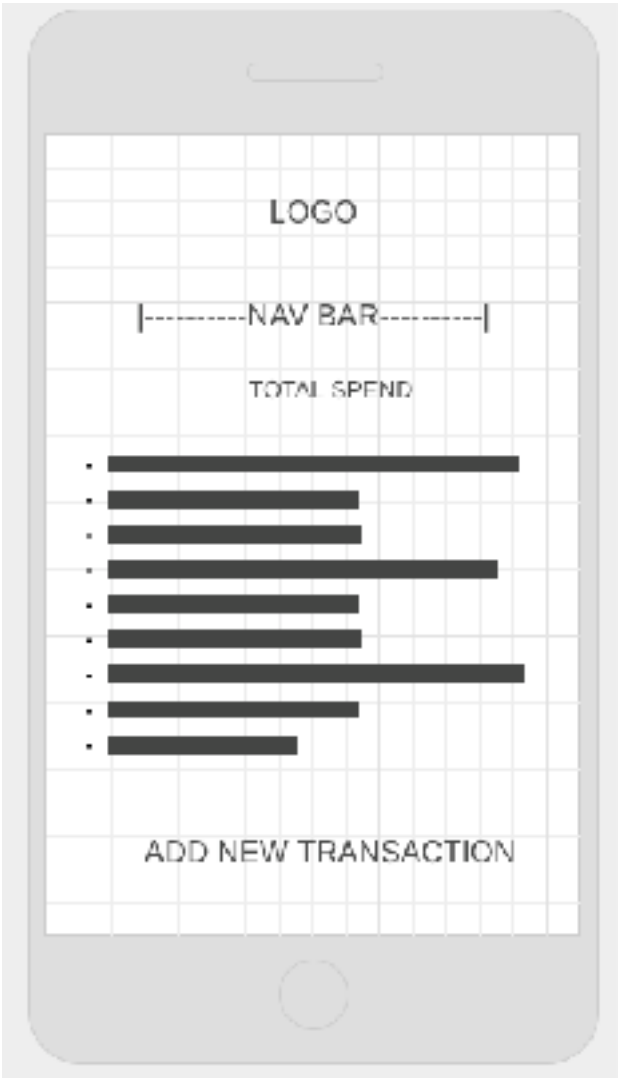
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user is able to add a new task to their to do list	A fragment form is displayed when the user clicks on the add task button	Pass
A user is able to sort their current task by priority	The drop menu allows for tasks to be viewed by low, medium or high priority	Pass
A user is able to update a task's information	A fragment form is displayed when the user clicks on the update task button	Pass
A user should be able to mark tasks as complete	A confirmation fragment appears to confirm that the task is completed	Pass

P.5 - User sitemap

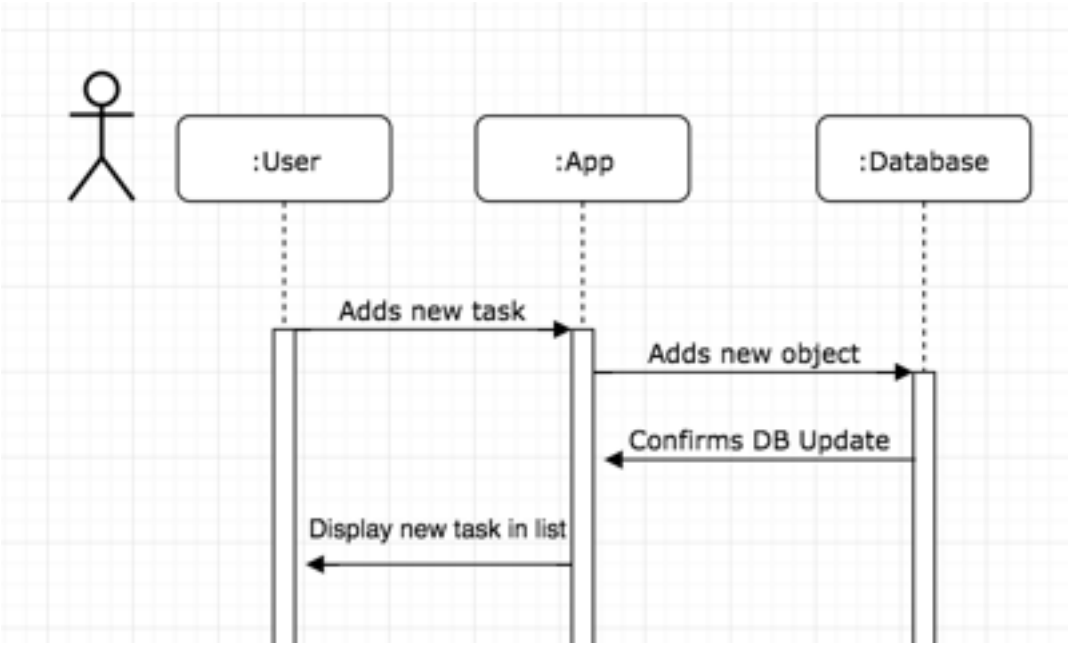


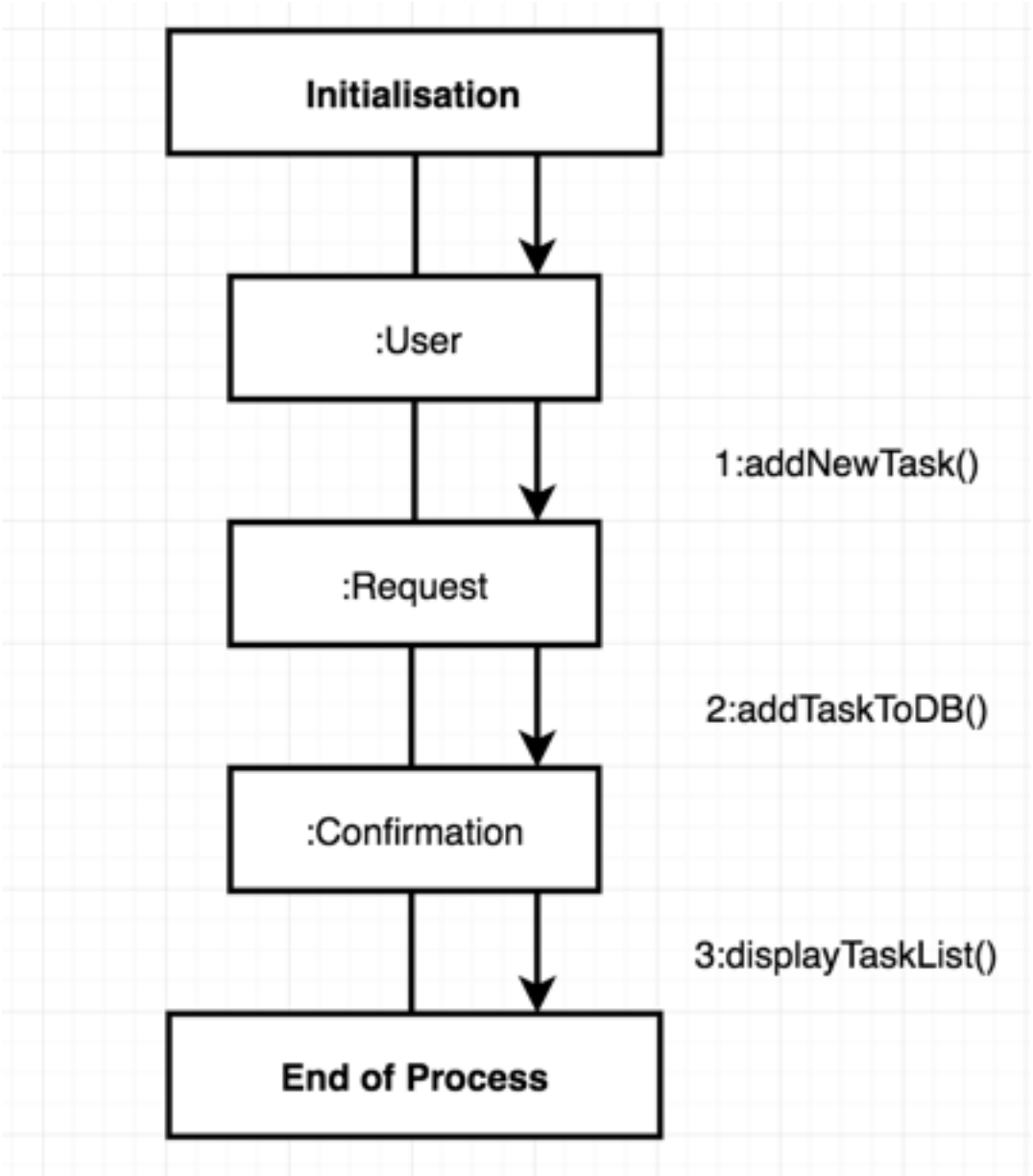
P.6 - Wireframe Designs



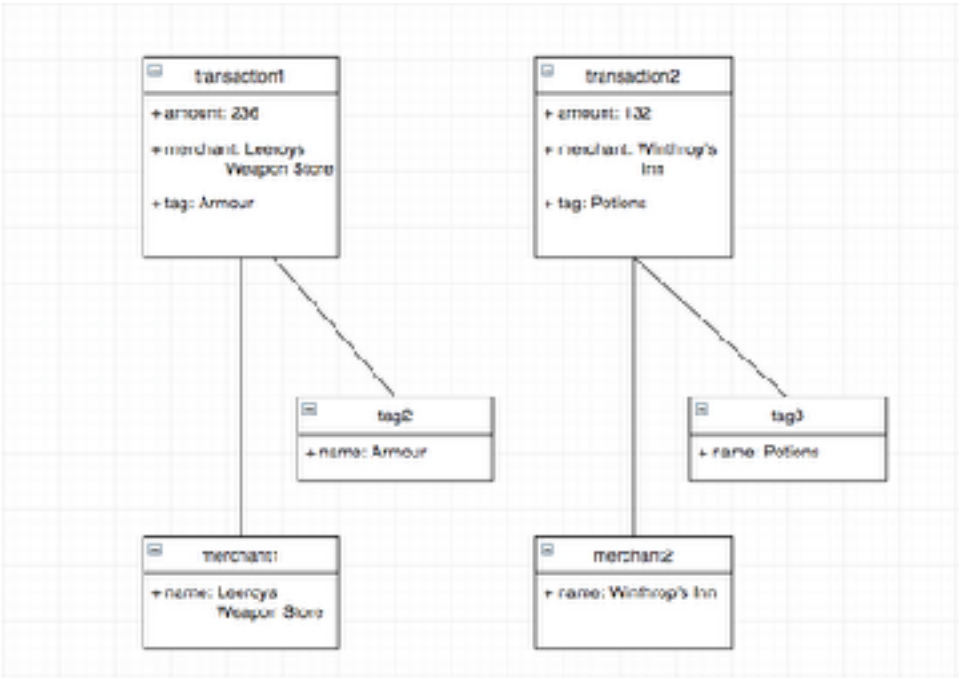


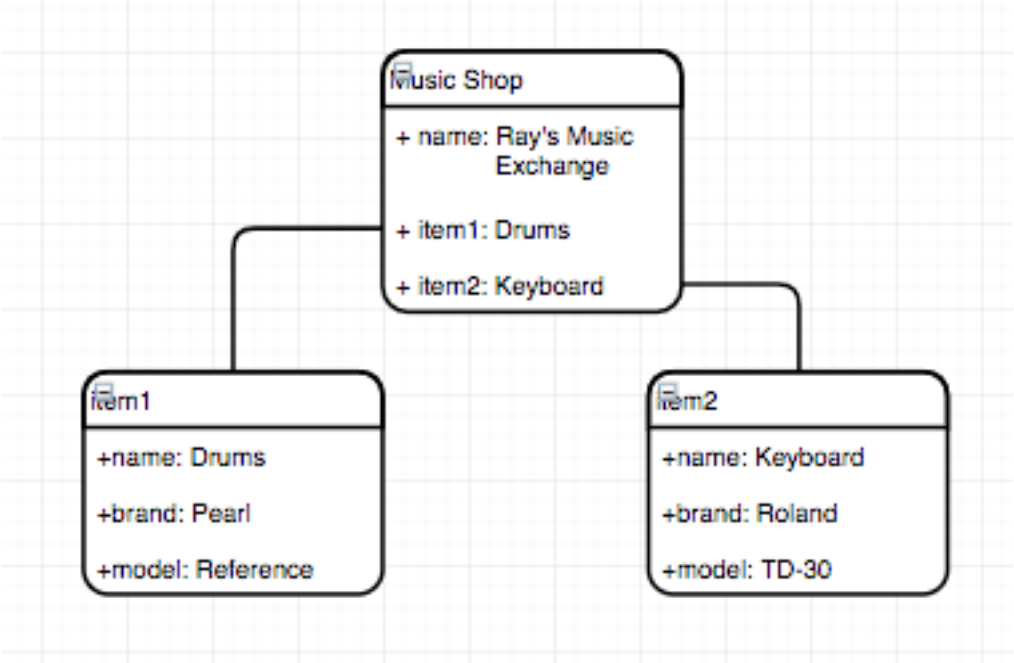
P.7 - Sequence Interaction Diagrams





P.8 - Object Diagrams





P.9 - Algorithms

- This algorithm takes an array from an api and displays each item by looping through and adding to a new array (hopNames). This information is then displayed as a string with each item e.g. 'Hops: Chinook, Ahtanum, Amarillo'.

```
const hopNames =[]
beer.ingredients.hops.forEach((hop) => {
  if (!hopNames.includes(hop.name)) {
    hopNames.push(hop.name);
  }
});
hops.textContent = `Hops: ${hopNames.join(", ")}`;
```

- This algorithm is used to filter the items in a list by priority level (id). This allows the items to be viewed by 'Low', 'Medium', 'High' or by all. These views are toggled by a dropdown menu.

```
@Override
public boolean onOptionsItemSelected(MenuItem menuItem) {
    if (menuItem.getItemId() == R.id.All) {
        refreshList();
    }

    if (menuItem.getItemId() == R.id.High) {
        refreshList();
        taskAdapter.filterTasks("High");
    }

    if (menuItem.getItemId() == R.id.Medium) {
        refreshList();
        taskAdapter.filterTasks("Medium");
    }

    if (menuItem.getItemId() == R.id.Low) {
        refreshList();
        taskAdapter.filterTasks("Low");
    }
    return true;
}
```

P.10 - Pseudocode

```
def sanitised_name(name)
  #"harry The blacksmith " == "Harry the Blacksmith"
  #downcase name parameter
  #determine if (name) contains 'and', 'the' or 'of'
  #upcase all words that are not 'and', 'the' or 'of'
  #return (name) in sanitised format
end
```

P.11 - Screenshot & Github Link
https://github.com/rgathergood/to_do_list_android_project

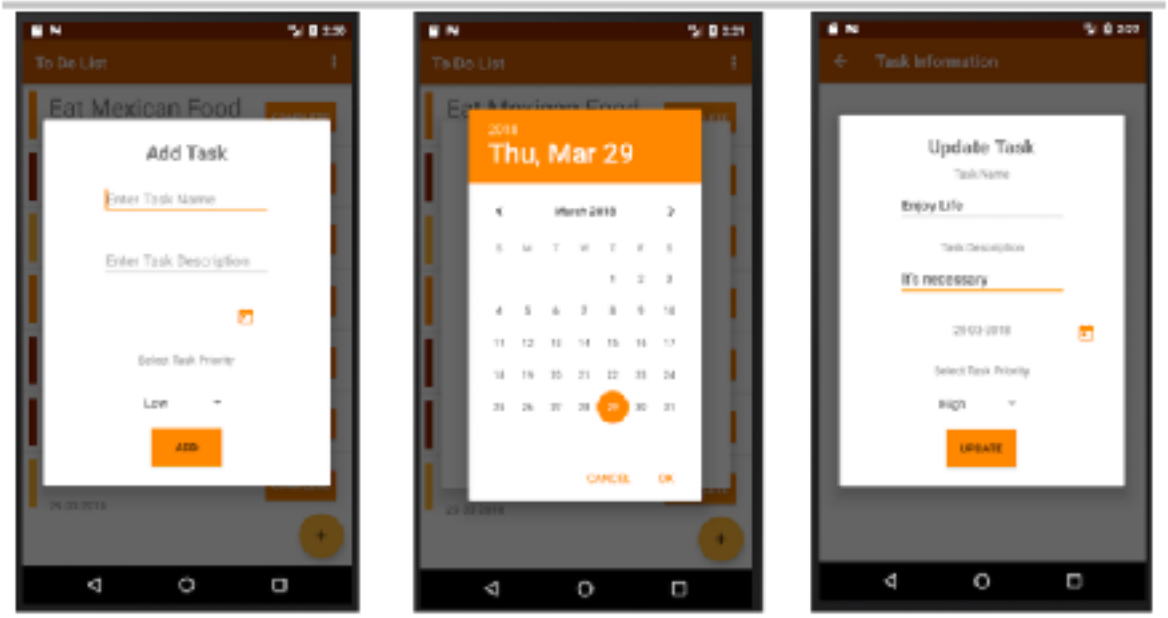
wk9_project

To-Do List

This was the second project I completed at CodeClan (Week 9) where I created a To Do List app in Android using Java and SQLite3.

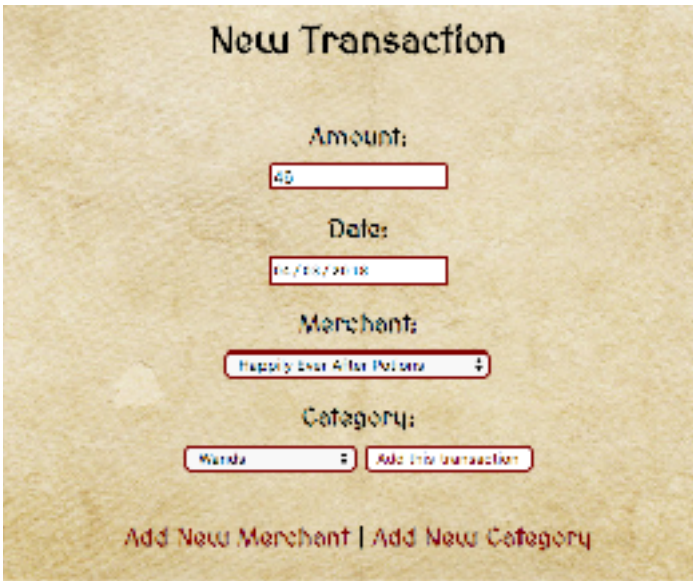
Project Specification

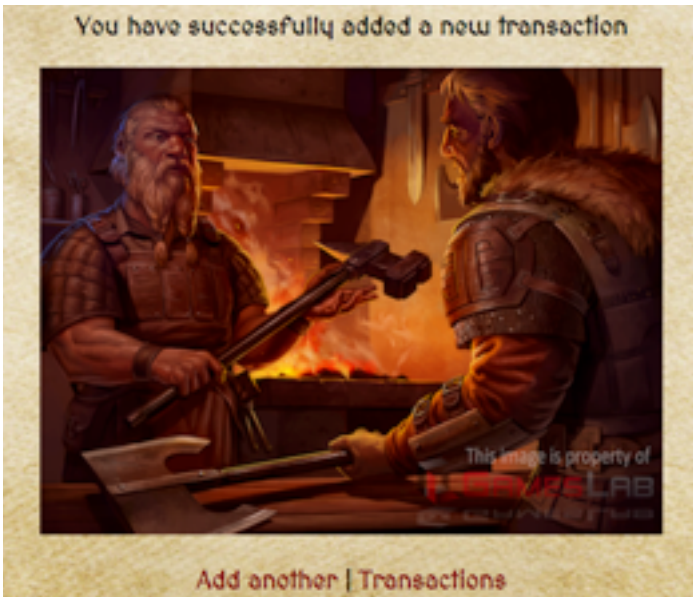
You are required to write an Android app which will allow the user to maintain a task list. The user should be able to enter tasks and display them in a list. The list should only show the bare details of each task. Selecting a task from the list should bring up another screen which shows full details of the task. Users should also be able to mark tasks as completed.



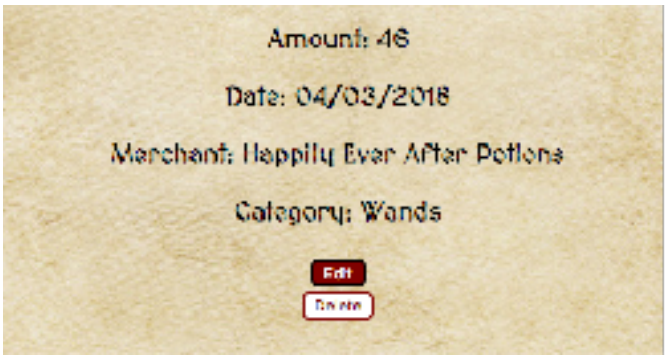
P.12 - Planning screenshots

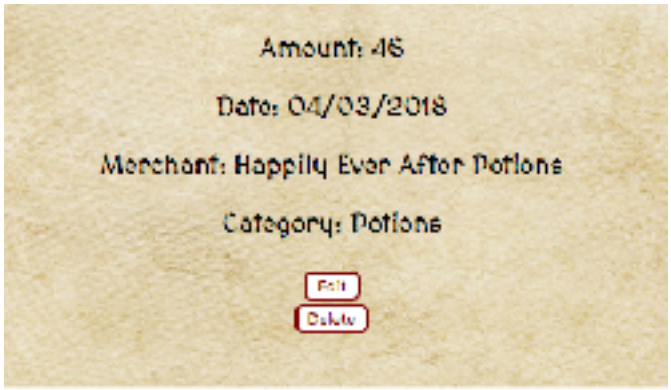
P.13 - User input being processed (user input/system response)



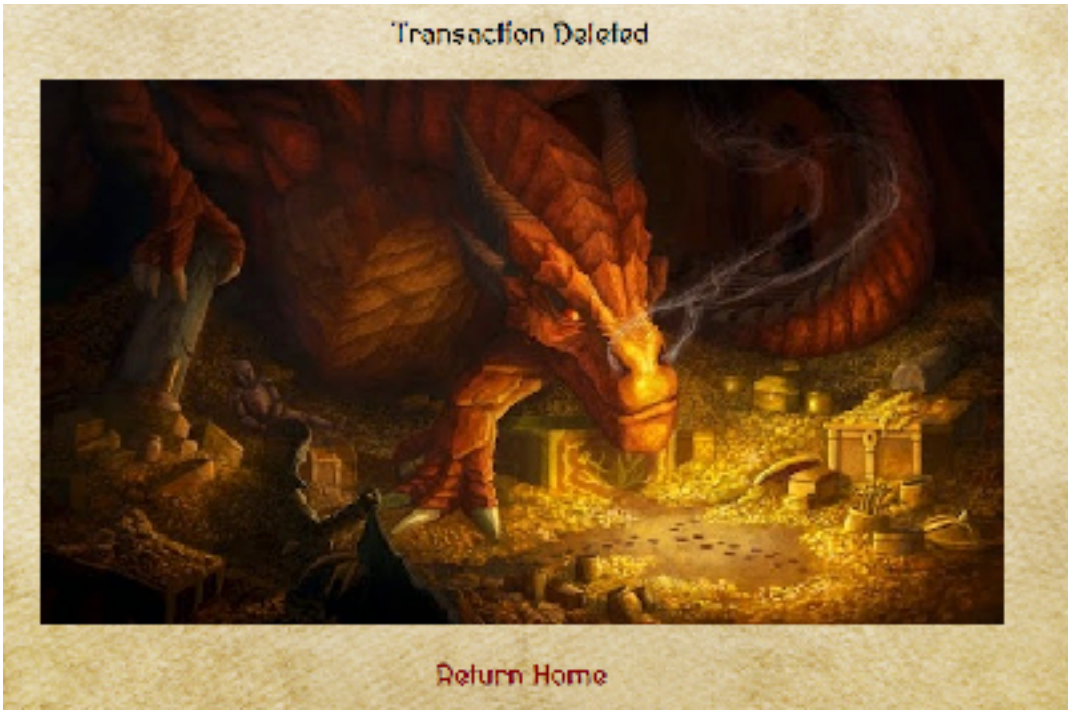


P.14 - Interaction with data persistence (Data input/save confirmation)





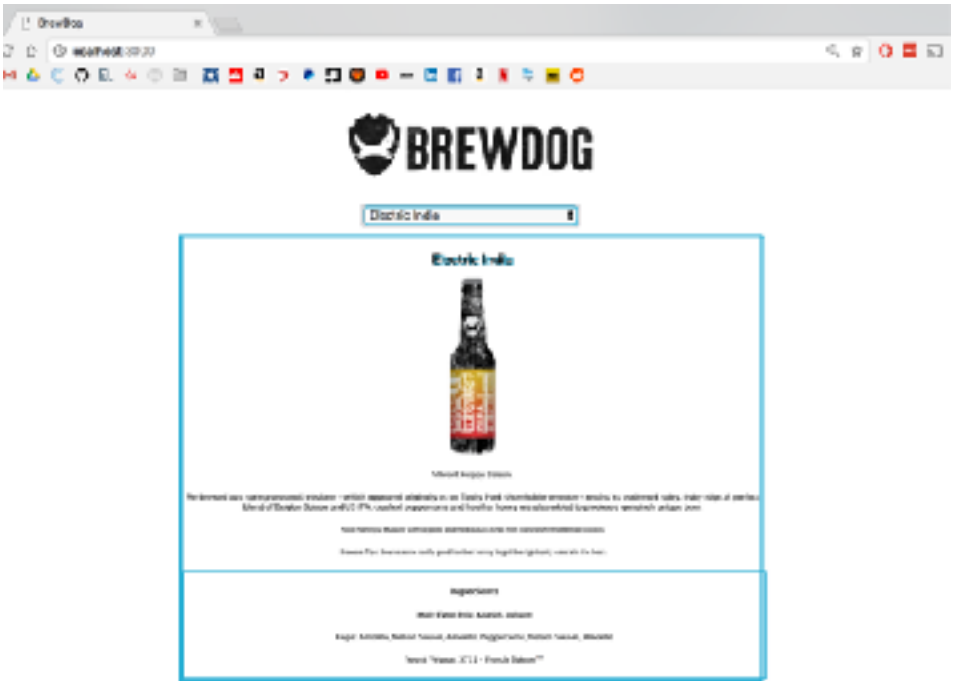
P.15 - Output of results to user (user request/request processed and outcome)



P.16 - API in use(code/in use by the program

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

  const select = document.querySelector('#beer-list');
  select.addEventListener('change', handleBeerSelection)
});
```



P.17 - Bug tracking report

DOM loads in Index			Passed
DOM loads api image	Failed	Image now references the correct api property	Passed
Timer counts in the background once the quiz is completed	Failed	timerReset() method called in results page	Passed
User needs to enter a name to start quiz	Failed	Name input is now required to start a new quiz	Passed
Audio plays on page load	Failed	Called playAnthem() on page load	Passed

P.18 - Testing(test code/code failing/errors corrected/code passing)

Failing

```ruby
require("minitest/autorun")
require_relative("testing_task_2.rb")
require_relative("card.rb")

class CardGameTest < MiniTest::Test

  def setup()
    @cardGame = CardGame.new
    @card1 = Card.new("Hearts", 10)
    @card2 = Card.new("Ace", 1)
  end

  def test_card_is_ace()
    assert_equal(true, checkforAce(@card2))
  end

  def test_card_is_not_ace()
    assert_equal(false, checkforAce(@card1))
  end

  def test_highest_card()
    assert_equal(10, highest_card(@card1, @card2))
  end

  def test_highest_card_swapped()
    assert_equal(10, highest_card(@card2, @card1))
  end
```

```
~  static_and_dynamic_tasks_a git:(master) ✗ ruby testing_task_2_spec.rb
Run options: --seed 52139

# Running:

EEEE

Finished in 0.001467s, 2725.6538 runs/s, 0.0000 assertions/s.

  1) Error:
CardGameTest#test_card_is_not_ace:
NoMethodError: undefined method `checkforAce' for #<CardGameTest:0x007fa929a208d8>
    testing_task_2_spec.rb:16:in `test_card_is_not_ace'

  2) Error:
CardGameTest#test_highest_card:
NoMethodError: undefined method `highest_card' for #<CardGameTest:0x007fa929a20240>
    testing_task_2_spec.rb:23:in `test_highest_card'

  3) Error:
CardGameTest#test_card_is_ace:
NoMethodError: undefined method `checkforAce' for #<CardGameTest:0x007fa92a04a9a0>
    testing_task_2_spec.rb:15:in `test_card_is_ace'

  4) Error:
CardGameTest#test_highest_card_swapped:
NoMethodError: undefined method `highest_card' for #<CardGameTest:0x007fa92a049c90>
    testing_task_2_spec.rb:27:in `test_highest_card_swapped'

4 runs, 0 assertions, 0 failures, 4 errors, 0 skips
```

Passing

```ruby
require("minitest/autorun")
require_relative("testing_task_2.rb")
require_relative("card.rb")

class CardGameTest < MiniTest::Test

  def setup()
    @cardGame = CardGame.new
    @card1 = Card.new("Hearts", 10)
    @card2 = Card.new("Ace", 1)
  end

  def test_card_is_ace()
    assert_equal(true, @cardGame.checkforAce(@card2))
  end

  def test_card_is_not_ace()
    assert_equal(false, @cardGame.checkforAce(@card1))
  end

  def test_highest_card()
    assert_equal(10, @cardGame.highest_card(@card1, @card2))
  end

  def test_highest_card_swapped()
    assert_equal(10, @cardGame.highest_card(@card2, @card1))
  end
```

```
→  static_and_dynamic_tasks_a git:(master) ✗ ruby testing_task_2_spec.rb
Run options: --seed 30313

# Running:

..F.

Finished in 0.001135s, 3524.2291 runs/s, 3524.2291 assertions/s.

  1) Failure:
CardGameTest#test_card_is_not_ace [testing_task_2_spec.rb:101]:
Expected: false
  Actual: nil

4 runs, 4 assertions, 1 failures, 0 errors, 0 skips
→  static_and_dynamic_tasks_a git:(master) ✗ ruby testing_task_2_spec.rb
Run options: --seed 52867

# Running:

....

Finished in 0.001302s, 2094.3560 runs/s, 2094.3560 assertions/s.

4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```