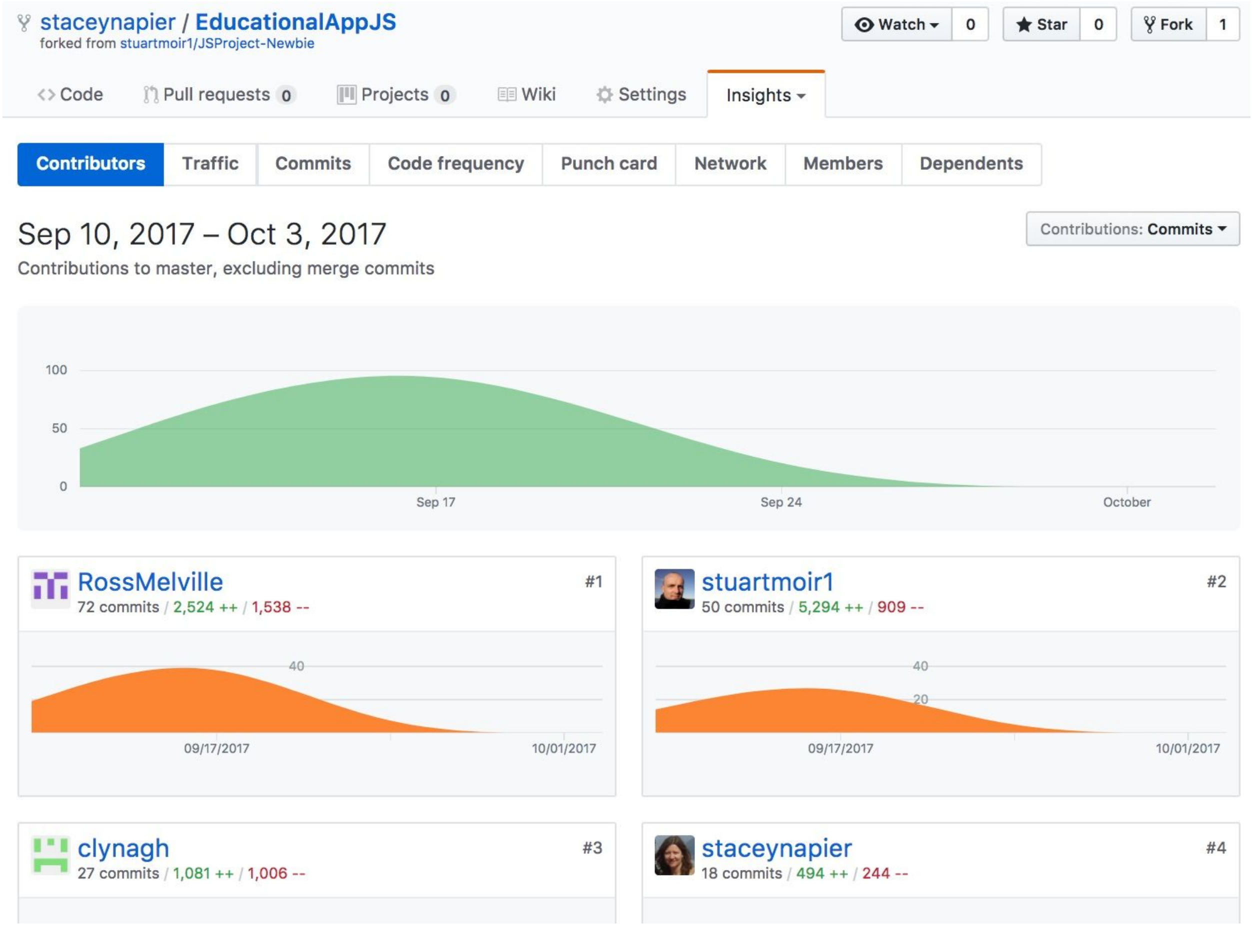
Stacey Napier Project Evidence

P1 - Group GitHub



staceynapier / EducationalAppJS
forked from stuartmoir1/JSProject-Newbie

Sep 10, 2017 – Oct 3, 2017
Contributions to master, excluding merge commits

RossMelville #1
72 commits / 2,524 ++ / 1,538 --

stuartmoir1 #2
50 commits / 5,294 ++ / 909 --

clynagh #3
27 commits / 1,081 ++ / 1,006 --

staceynapier #4
18 commits / 494 ++ / 244 --

P2 - Project Brief

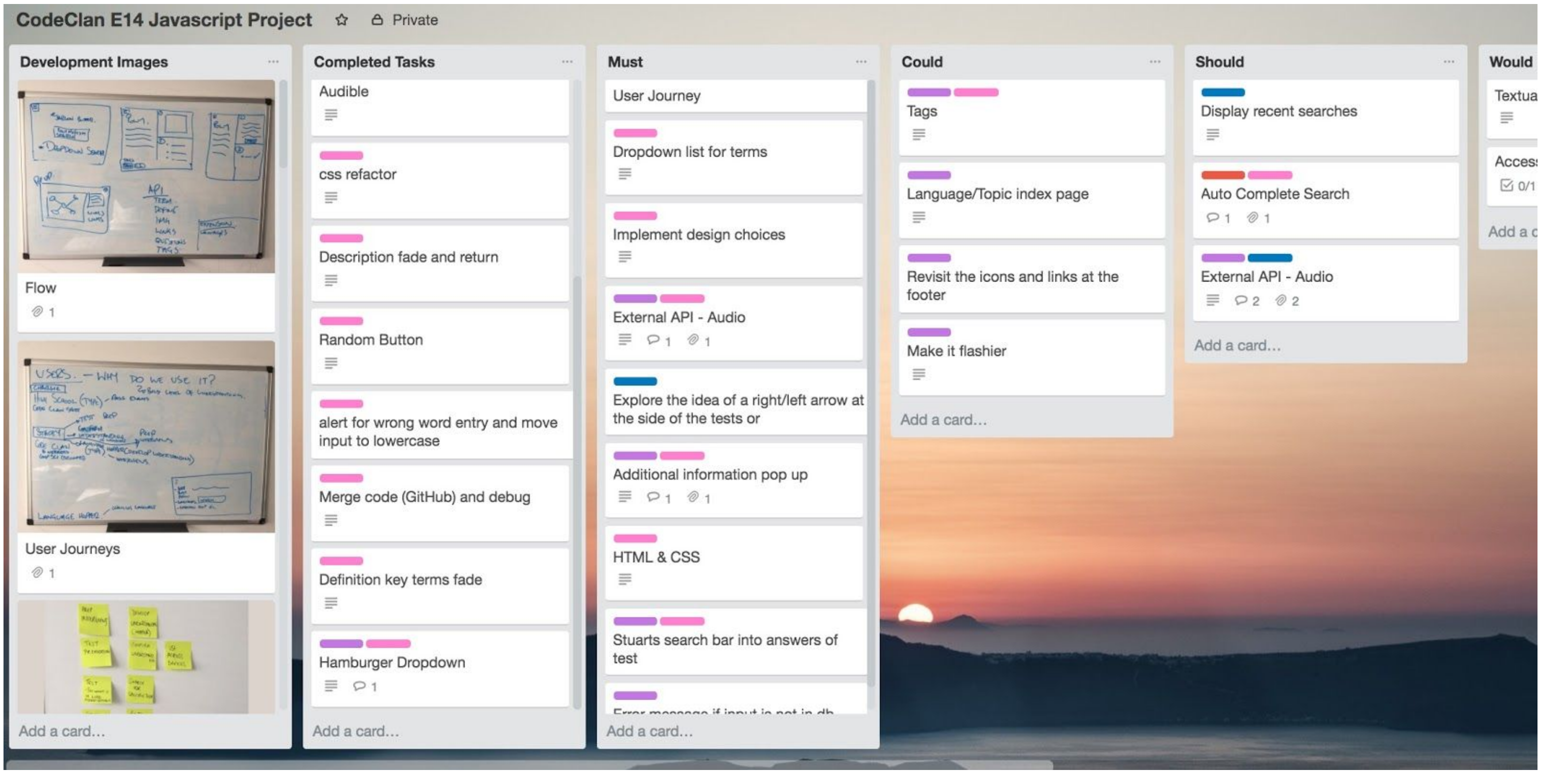
Create an online educational tool which is fun and interactive which will assists the user in understanding programming principles.

The app should allow the user to search for keywords that they are looking to have a greater understanding of.

The search should provide the user with a definition of the subject and subsequent options to allow them to further their learning.

MVP
Ability to search db and get definition.
User activity by interactive options.

P3 - Planning



CodeClan E14 Javascript Project ☆ Private

Development Images
- Flow — 1
- User Journeys — 1
- Add a card…

Completed Tasks
- Audible
- css refactor
- Description fade and return
- Random Button
- alert for wrong word entry and move input to lowercase
- Merge code (GitHub) and debug
- Definition key terms fade
- Hamburger Dropdown — 1
- Add a card…

Must
- User Journey
- Dropdown list for terms
- Implement design choices
- External API - Audio — 1, 1
- Explore the idea of a right/left arrow at the side of the tests or
- Additional information pop up — 1, 1
- HTML & CSS
- Stuarts search bar into answers of test
- Error message if input is not in db
- Add a card…

Could
- Tags
- Language/Topic index page
- Revisit the icons and links at the footer
- Make it flashier
- Add a card…

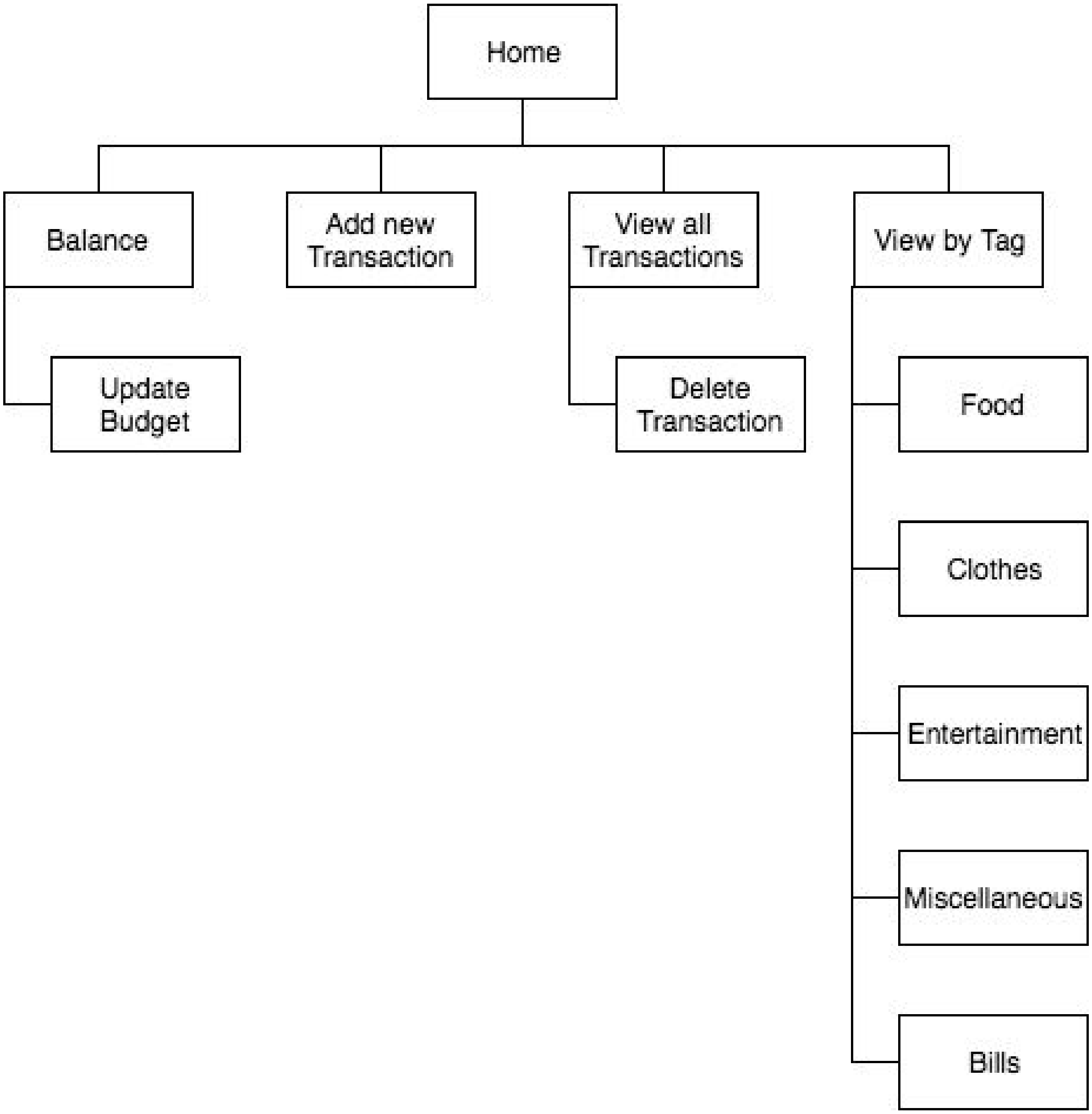
Should
- Display recent searches
- Auto Complete Search — 1, 1
- External API - Audio — 2, 2
- Add a card…

Would
- Textua
- Acces — 0/1
- Add a c

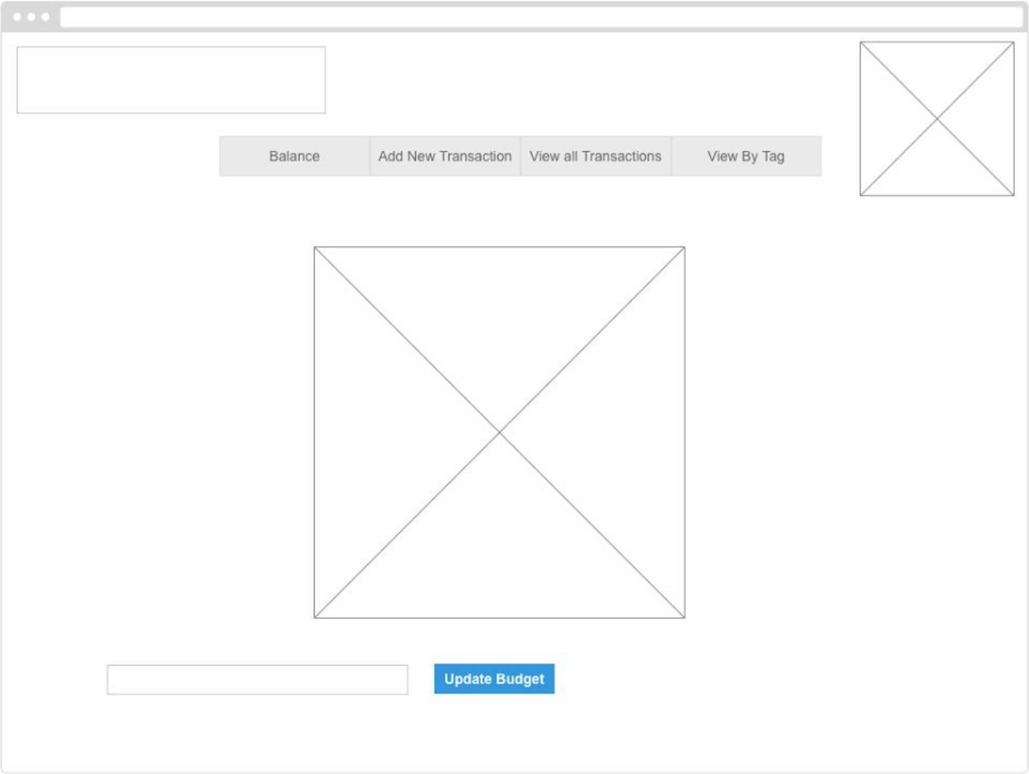
P4 - Acceptance Criteria

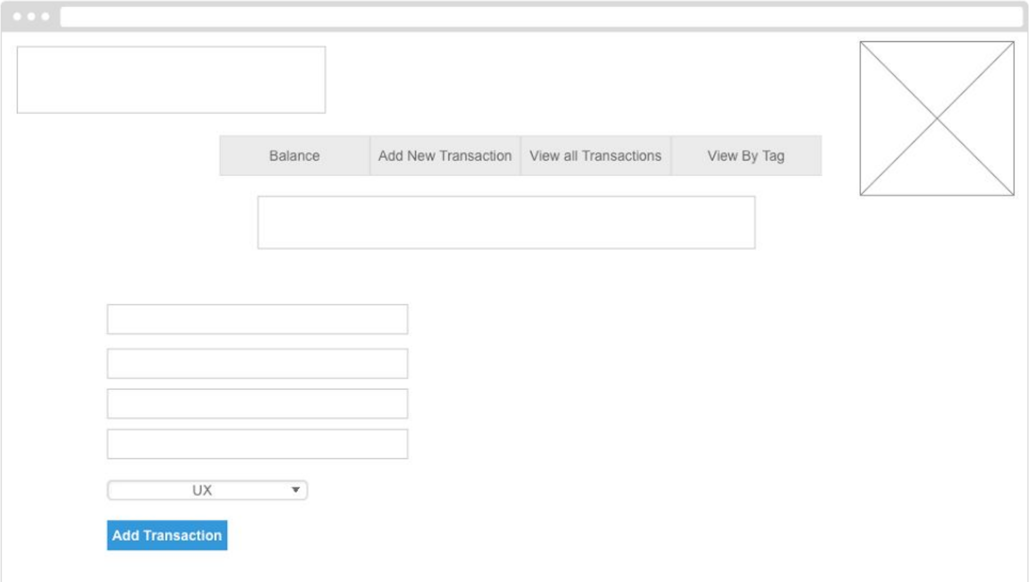
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user is able to search for a particular keyword	Word is displayed	Pass
A user is able to use a menu to view list of keywords	Menu displays showing keywords	Pass
A user is able to check their understanding through an interactive test.	User should have to input data which is checked and confirmed if correct	Pass
A user can view more information about a word	A button press will present further information in a pop up	Pass

P5 - User Sitemap

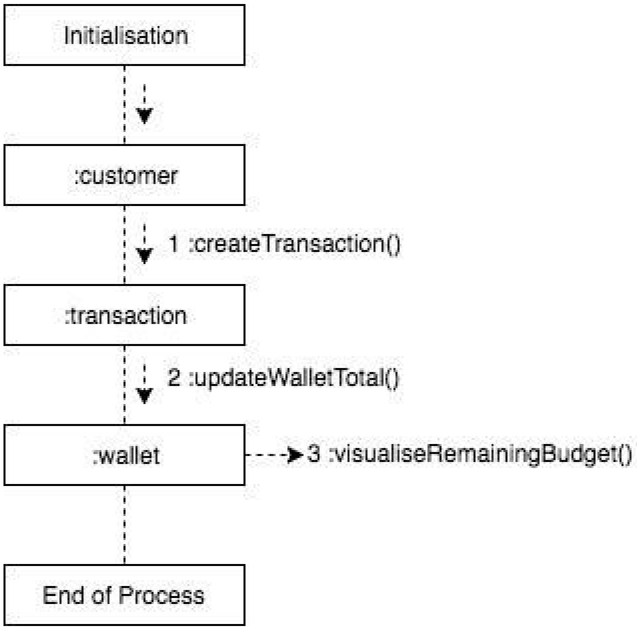


P6 - Wireframe

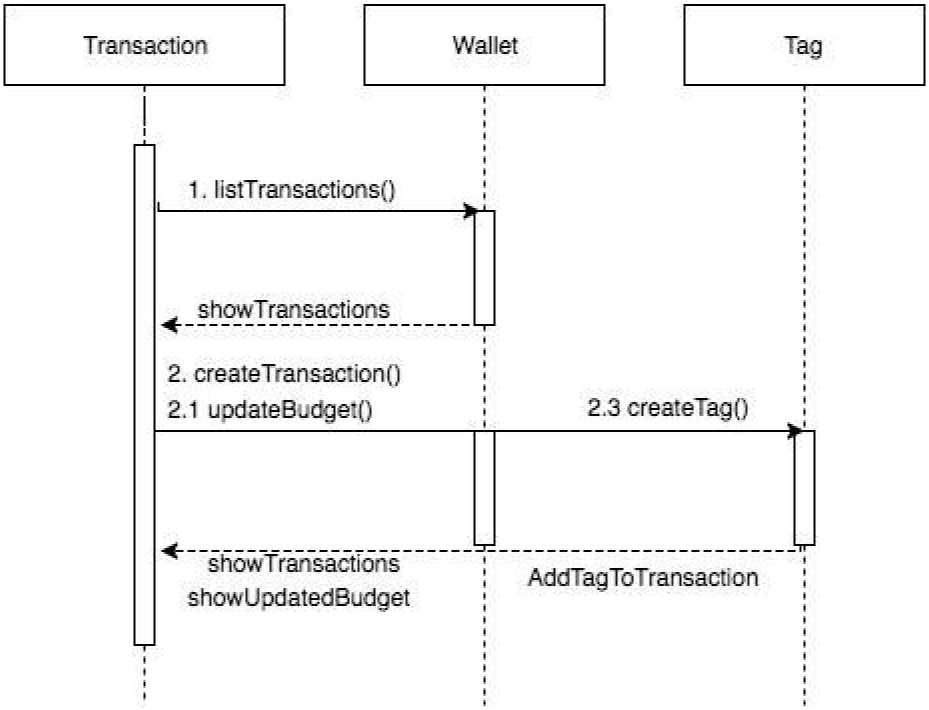




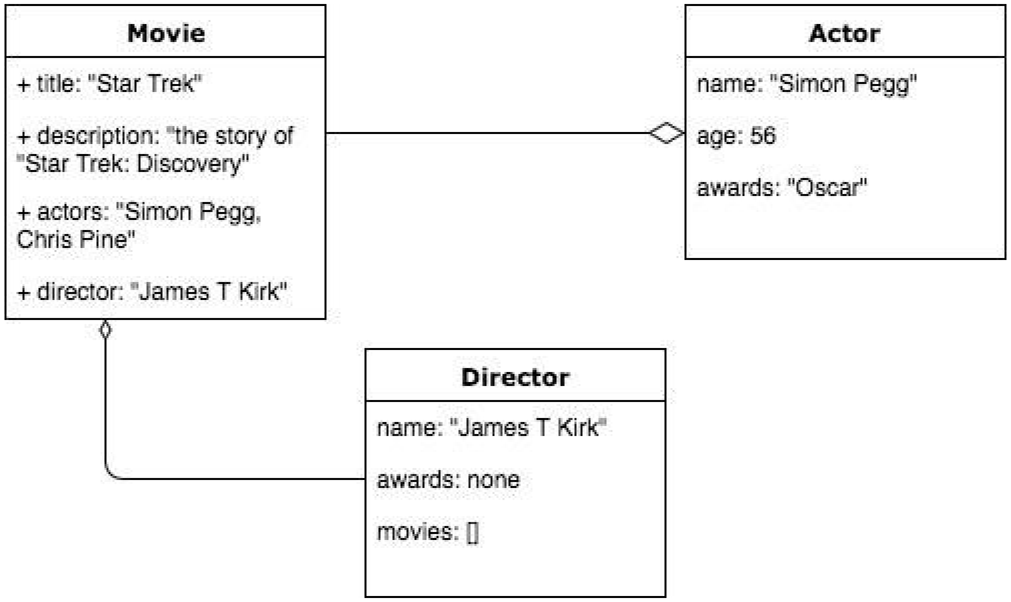
P7 Collaboration Diagram

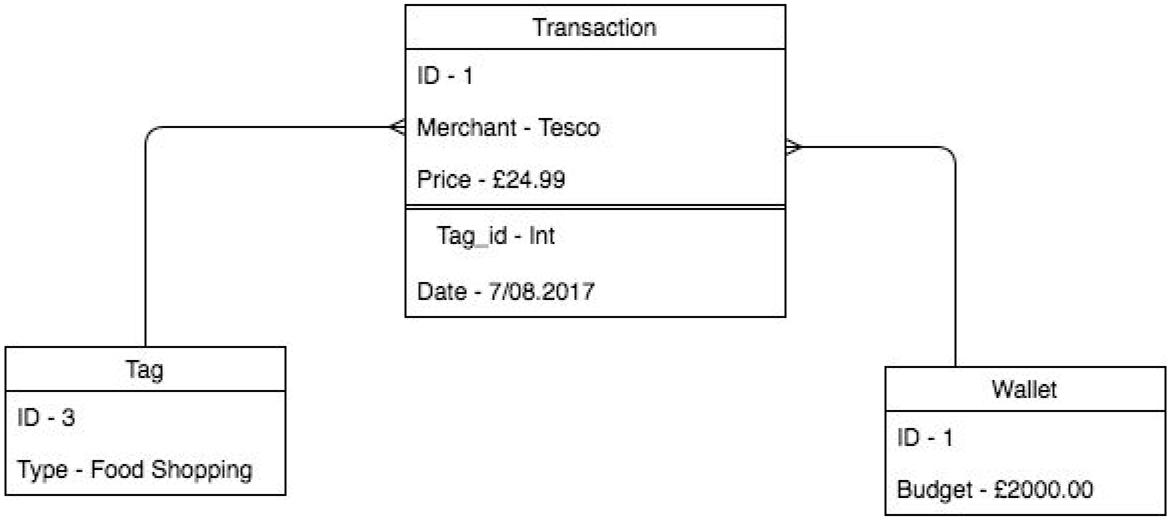


Sequence Diagram

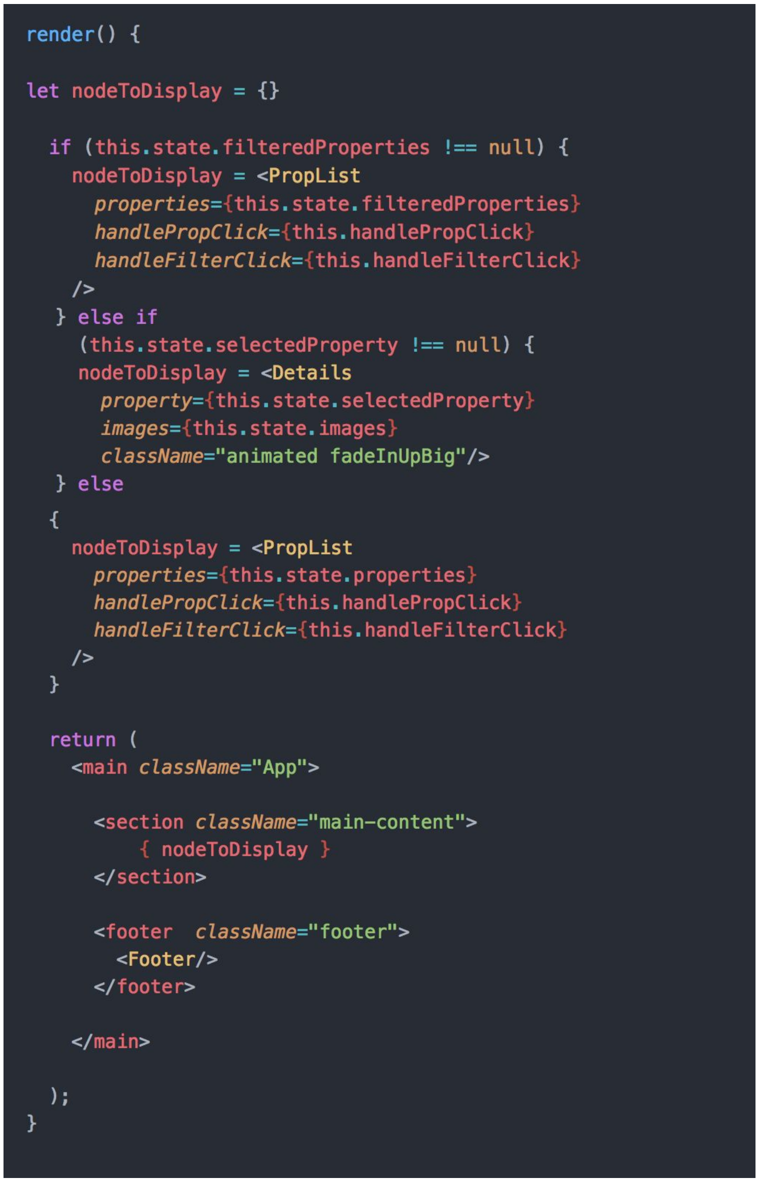


P8 - Object Diagram



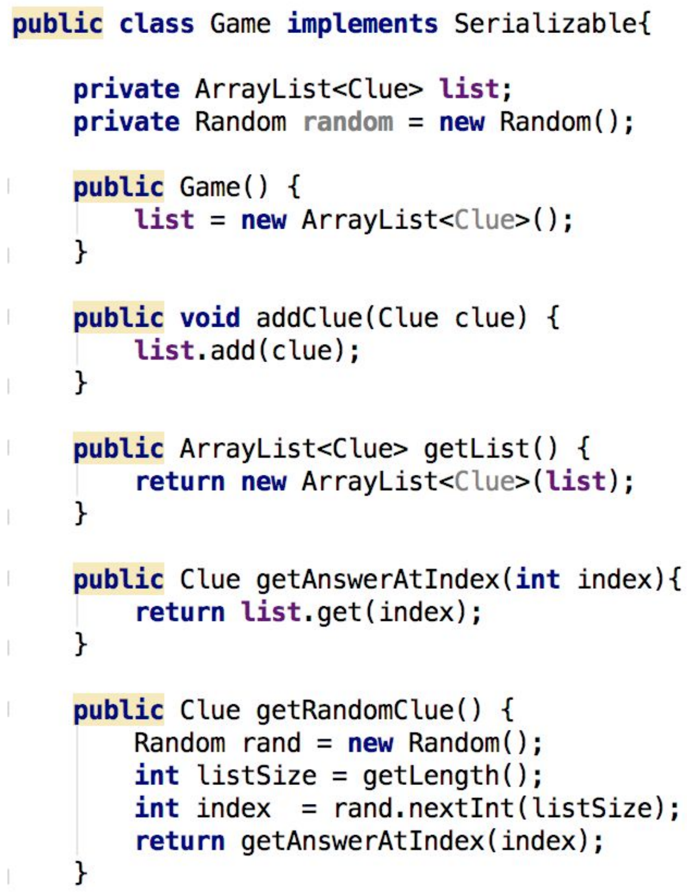


P9 - Algorithms



```
render() {

let nodeToDisplay = {}

  if (this.state.filteredProperties !== null) {
    nodeToDisplay = <PropList
      properties={this.state.filteredProperties}
      handlePropClick={this.handlePropClick}
      handleFilterClick={this.handleFilterClick}
    />
  } else if
     (this.state.selectedProperty !== null) {
     nodeToDisplay = <Details
       property={this.state.selectedProperty}
       images={this.state.images}
       className="animated fadeInUpBig"/>
  } else
  {
    nodeToDisplay = <PropList
      properties={this.state.properties}
      handlePropClick={this.handlePropClick}
      handleFilterClick={this.handleFilterClick}
    />
  }

  return (
    <main className="App">

      <section className="main-content">
          { nodeToDisplay }
      </section>

      <footer  className="footer">
        <Footer/>
      </footer>

    </main>

  );
}
```

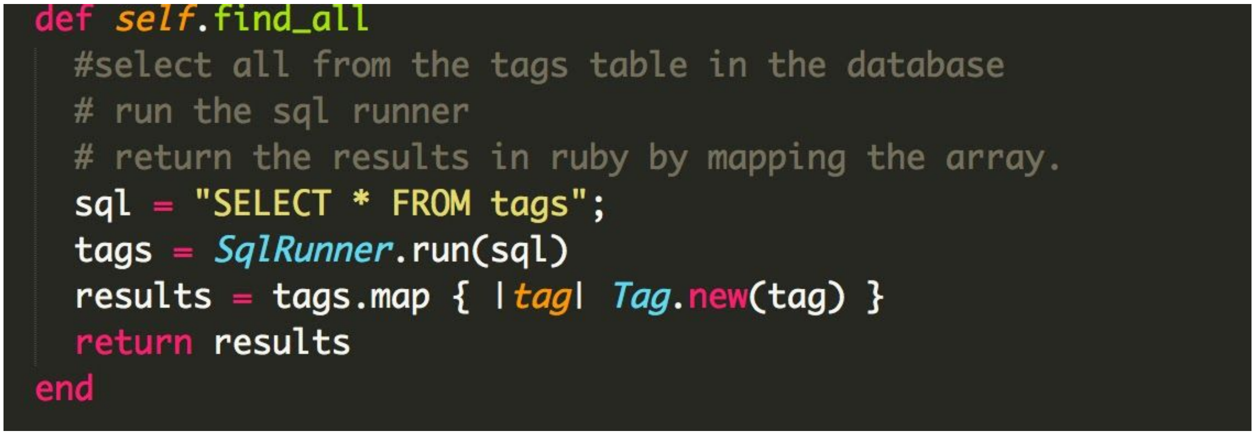
I chose the above algorithm to help determine which page should be displayed in the browser. The algorithm checks the state held in the constructor and selects which component is to be rendered to the browser.



```java
public class Game implements Serializable{

    private ArrayList<Clue> list;
    private Random random = new Random();

    public Game() {
        list = new ArrayList<Clue>();
    }

    public void addClue(Clue clue) {
        list.add(clue);
    }

    public ArrayList<Clue> getList() {
        return new ArrayList<Clue>(list);
    }

    public Clue getAnswerAtIndex(int index){
        return list.get(index);
    }

    public Clue getRandomClue() {
        Random rand = new Random();
        int listSize = getLength();
        int index  = rand.nextInt(listSize);
        return getAnswerAtIndex(index);
    }
```

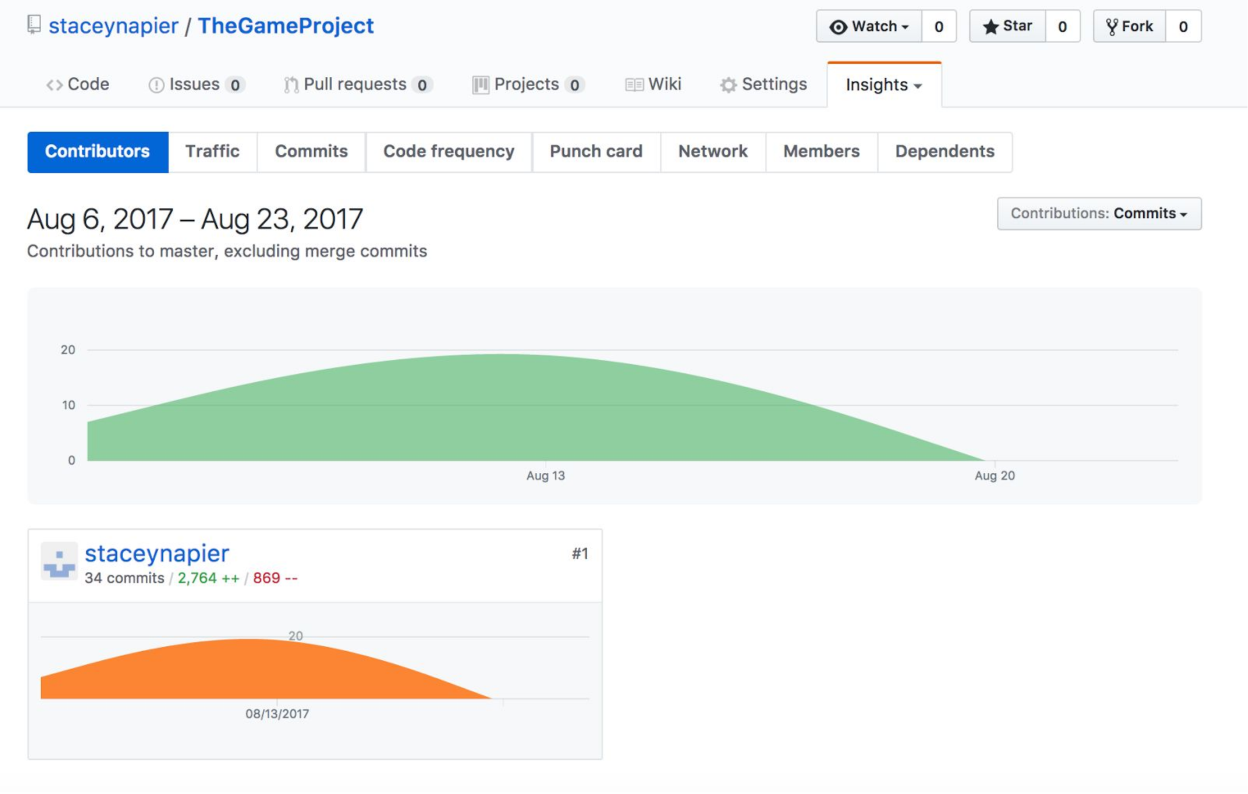
The above ‘getRandomClue’ algorithm was used as the app I created needed to randomly select a clue from an array of clues. This was to be presented on screen as part of a game. To get the random clue, I needed to get the length of the array, choose a number at random that is less than the length and then apply this to the array using the ‘getAnswerAtIndex’ function. This algorithm was used as Java doesn’t offer built in functions like Ruby or Javascript to choose an item at random.

P10 - Pseudocode



```
def self.find_all
  #select all from the tags table in the database
  # run the sql runner
  # return the results in ruby by mapping the array.
  sql = "SELECT * FROM tags";
  tags = SqlRunner.run(sql)
  results = tags.map { |tag| Tag.new(tag) }
  return results
end
```

P11 Solo Project



staceynapier / TheGameProject

Aug 6, 2017 – Aug 23, 2017

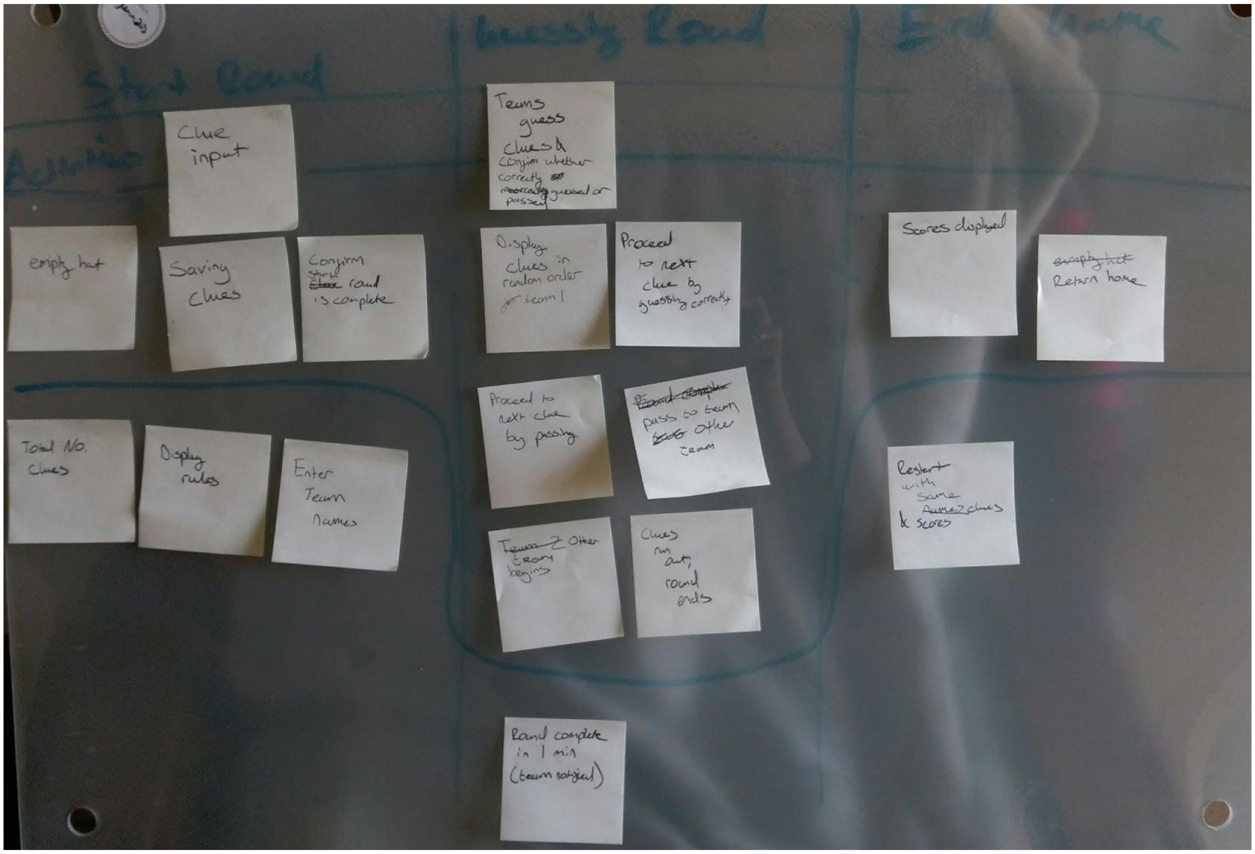
Contributions to master, excluding merge commits

staceynapier

34 commits / 2,764 ++ / 869 --

https://github.com/staceynapier/TheGameProject

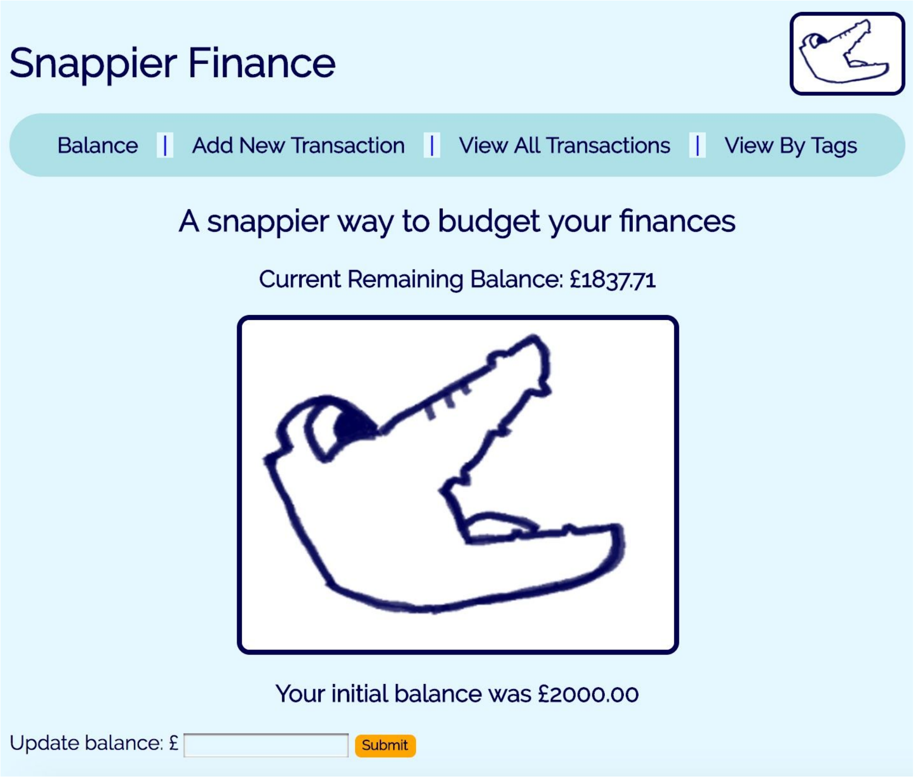
P12 - Planning


Start Round
Guessing Round
End Game
Activities
Clue input
empty hat
Saving clues
Confirm start ~~Clue~~ round is complete
Total No. clues
Display rules
Enter Team names
Teams guess clues & confirm whether correctly ~~or~~ ~~incorrectly~~ guessed or passed
Display clues in random order for team 1
Proceed to next clue by guessing correctly
Proceed to next clue by passing
~~Round complete~~ pass to team ~~two~~ other team
~~Team 2~~ Other team begins
Clues run out, round ends
Round complete in 1 min (team notified)
Scores displayed
~~empty hat~~ Return home
Restart with same ~~Names~~ clues & scores

The image above indicates a planning session, in which I noted down all of the steps that would be required for the game to work. From here, I was able to establish which steps were necessary for the MVP - indicated by the blue line.

P13

User inputs a new budget which alters the available balance left.


Balance | Add New Transaction | View All Transactions | View By Tags
Submit

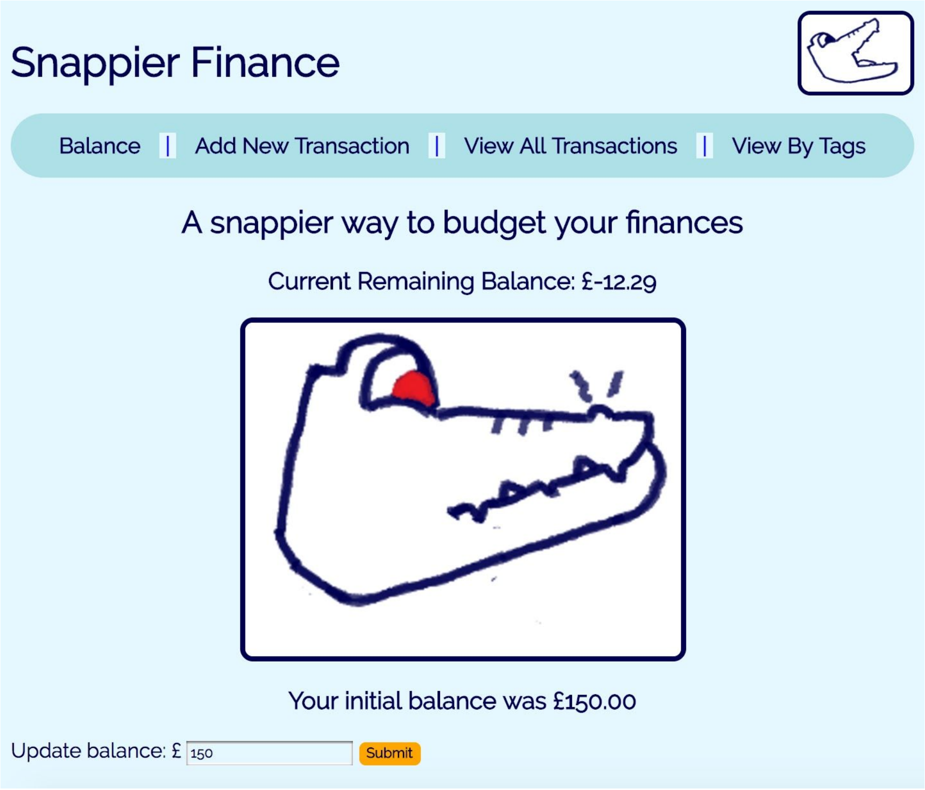
Snappier Finance

A snappier way to budget your finances

Current Remaining Balance: £1837.71

Your initial balance was £2000.00

Update balance: £


Balance | Add New Transaction | View All Transactions | View By Tags
Submit

Snappier Finance

A snappier way to budget your finances

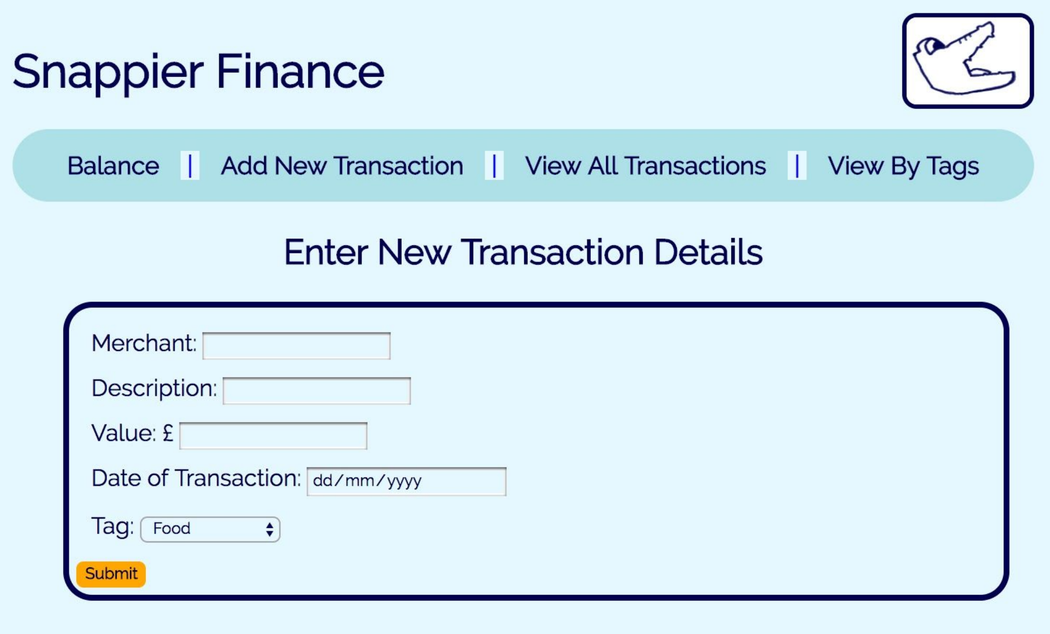
Current Remaining Balance: £-12.29

Your initial balance was £150.00

Update balance: £ 150

P14

User inputs new transaction details. Showing all transactions confirms transaction saved in the database.


Snappier Finance
Balance | Add New Transaction | View All Transactions | View By Tags
Submit

Enter New Transaction Details

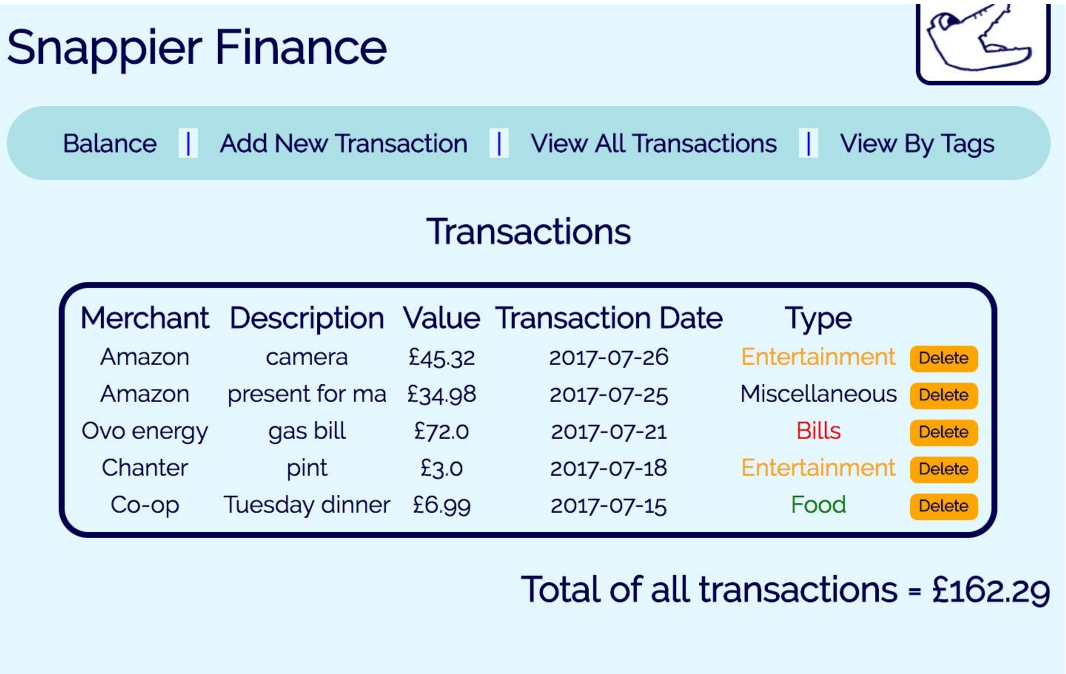
Merchant:

Description:

Value: £

Date of Transaction: dd/mm/yyyy

Tag: Food


Snappier Finance
Balance | Add New Transaction | View All Transactions | View By Tags

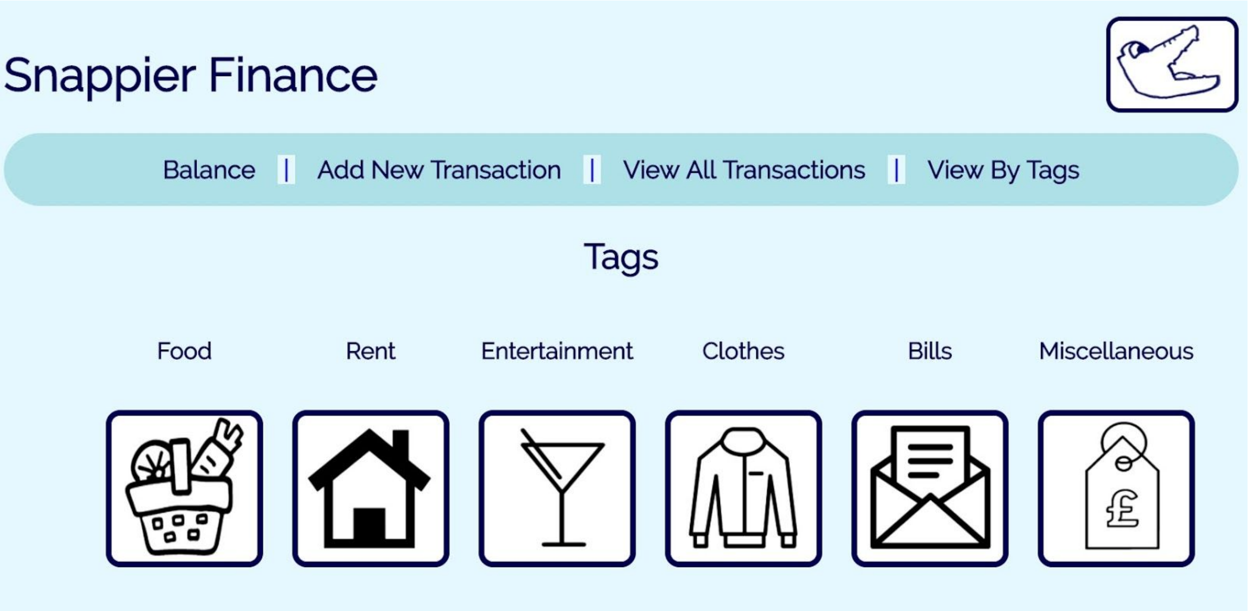
Transactions

Merchant	Description	Value	Transaction Date	Type	
Amazon	camera	£45.32	2017-07-26	Entertainment	Delete
Amazon	present for ma	£34.98	2017-07-25	Miscellaneous	Delete
Ovo energy	gas bill	£72.0	2017-07-21	Bills	Delete
Chanter	pint	£3.0	2017-07-18	Entertainment	Delete
Co-op	Tuesday dinner	£6.99	2017-07-15	Food	Delete

Total of all transactions = £162.29

P 15

User selects a tag and is taken to a new page with a list of all the transactions with that tag.

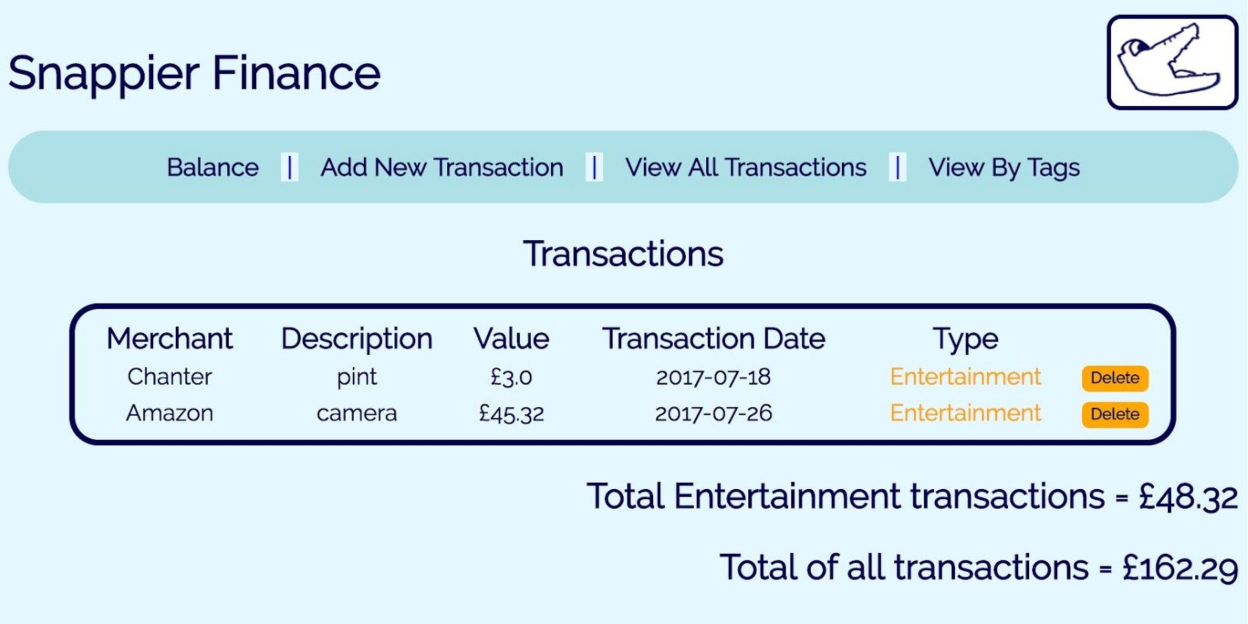


Snappier Finance

Balance | Add New Transaction | View All Transactions | View By Tags

Tags

Food
Rent
Entertainment
Clothes
Bills
Miscellaneous
£



Snappier Finance

Balance | Add New Transaction | View All Transactions | View By Tags

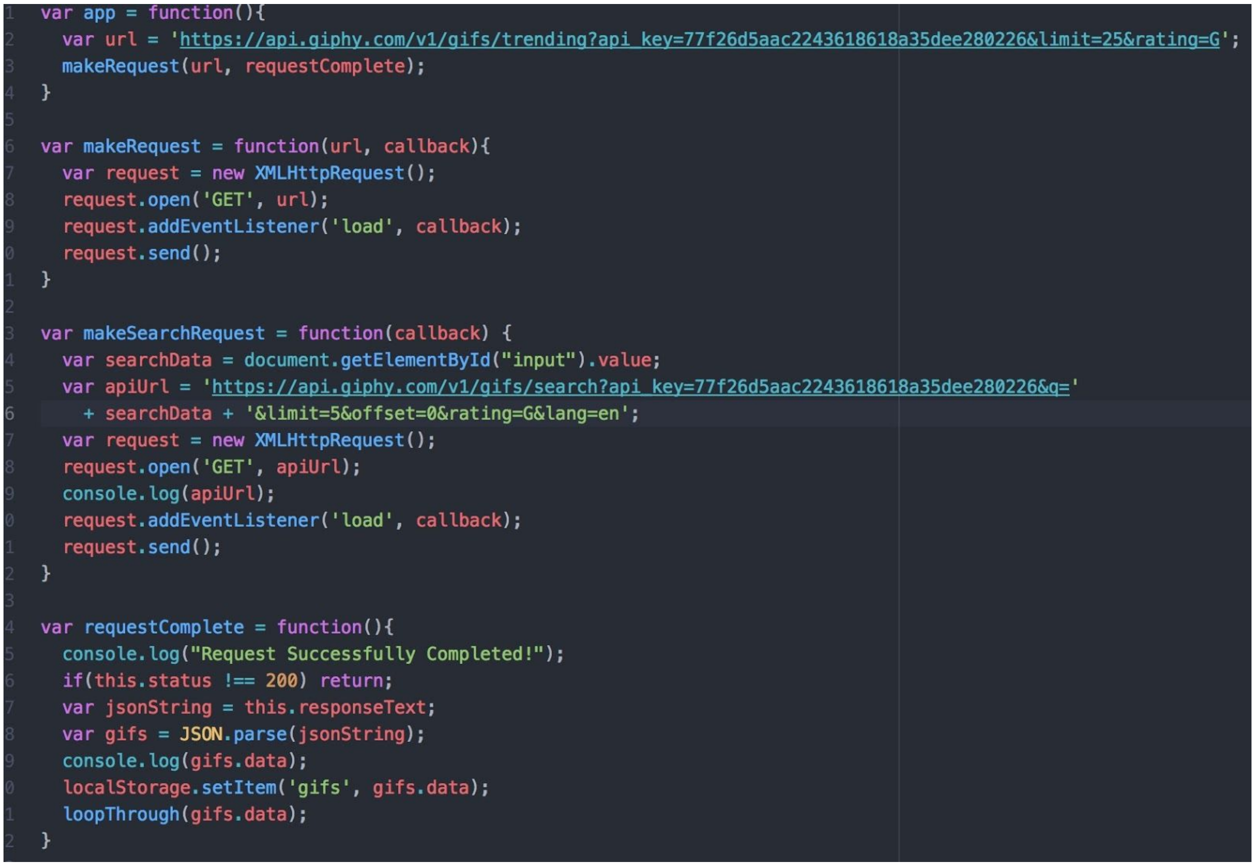
Transactions

Merchant	Description	Value	Transaction Date	Type	
Chanter	pint	£3.0	2017-07-18	Entertainment	Delete
Amazon	camera	£45.32	2017-07-26	Entertainment	Delete

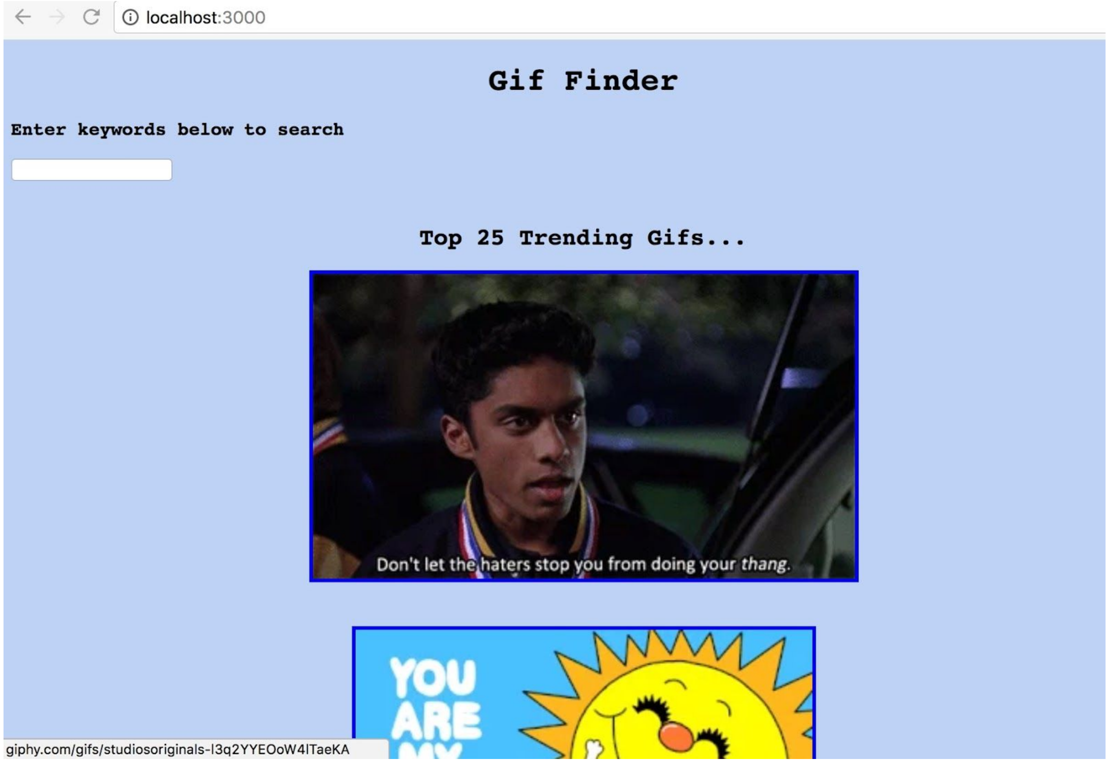
Total Entertainment transactions = £48.32

Total of all transactions = £162.29

P16 API



```
var app = function(){
  var url = 'https://api.giphy.com/v1/gifs/trending?api_key=77f26d5aac2243618618a35dee280226&limit=25&rating=G';
  makeRequest(url, requestComplete);
}

var makeRequest = function(url, callback){
  var request = new XMLHttpRequest();
  request.open('GET', url);
  request.addEventListener('load', callback);
  request.send();
}

var makeSearchRequest = function(callback) {
  var searchData = document.getElementById("input").value;
  var apiUrl = 'https://api.giphy.com/v1/gifs/search?api_key=77f26d5aac2243618618a35dee280226&q='
    + searchData + '&limit=5&offset=0&rating=G&lang=en';
  var request = new XMLHttpRequest();
  request.open('GET', apiUrl);
  console.log(apiUrl);
  request.addEventListener('load', callback);
  request.send();
}

var requestComplete = function(){
  console.log("Request Successfully Completed!");
  if(this.status !== 200) return;
  var jsonString = this.responseText;
  var gifs = JSON.parse(jsonString);
  console.log(gifs.data);
  localStorage.setItem('gifs', gifs.data);
  loopThrough(gifs.data);
}
```



Gif Finder

Enter keywords below to search

Top 25 Trending Gifs...

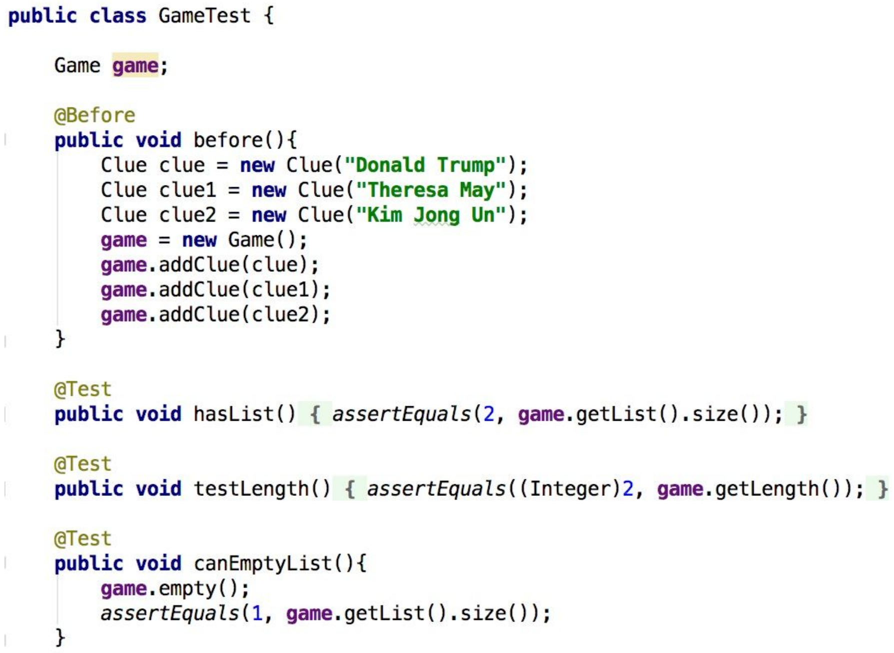
giphy.com/gifs/studiosoriginals-l3q2YYEOoW4lTaeKA

P17 Bug Tracking Report

Bug		Fix	
Footer disappears from main page when scrolling	Fail	Amended CSS to ensure the footer ‘sticks’ to the page ad is now on show permanently	Pass
Content doesn’t respond to different screen sizes	Fail	Created flexbox container and changed font size from ‘px’ to ‘em’ to respond appropriately	Pass
Filter can only check by one item, should be able to filter by price and number of bedrooms	Fail	Amended logic to check both before presenting the filtered list	Pass
Unable to view all images that are linked to each property	Fail	Amended database and added image table which is connected to the property by id	Pass
Unable to get all relevant information in one xml request	Fail	Amended the property controller to automatically bring back all linked images in the same xml request	Pass

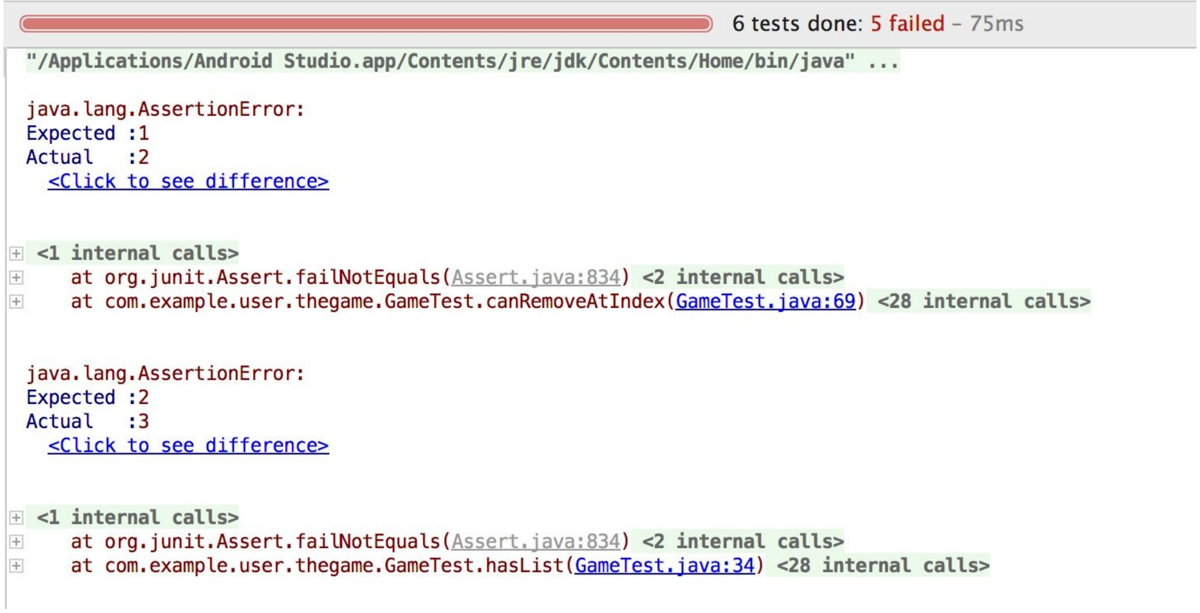
P18 Testing

Test code



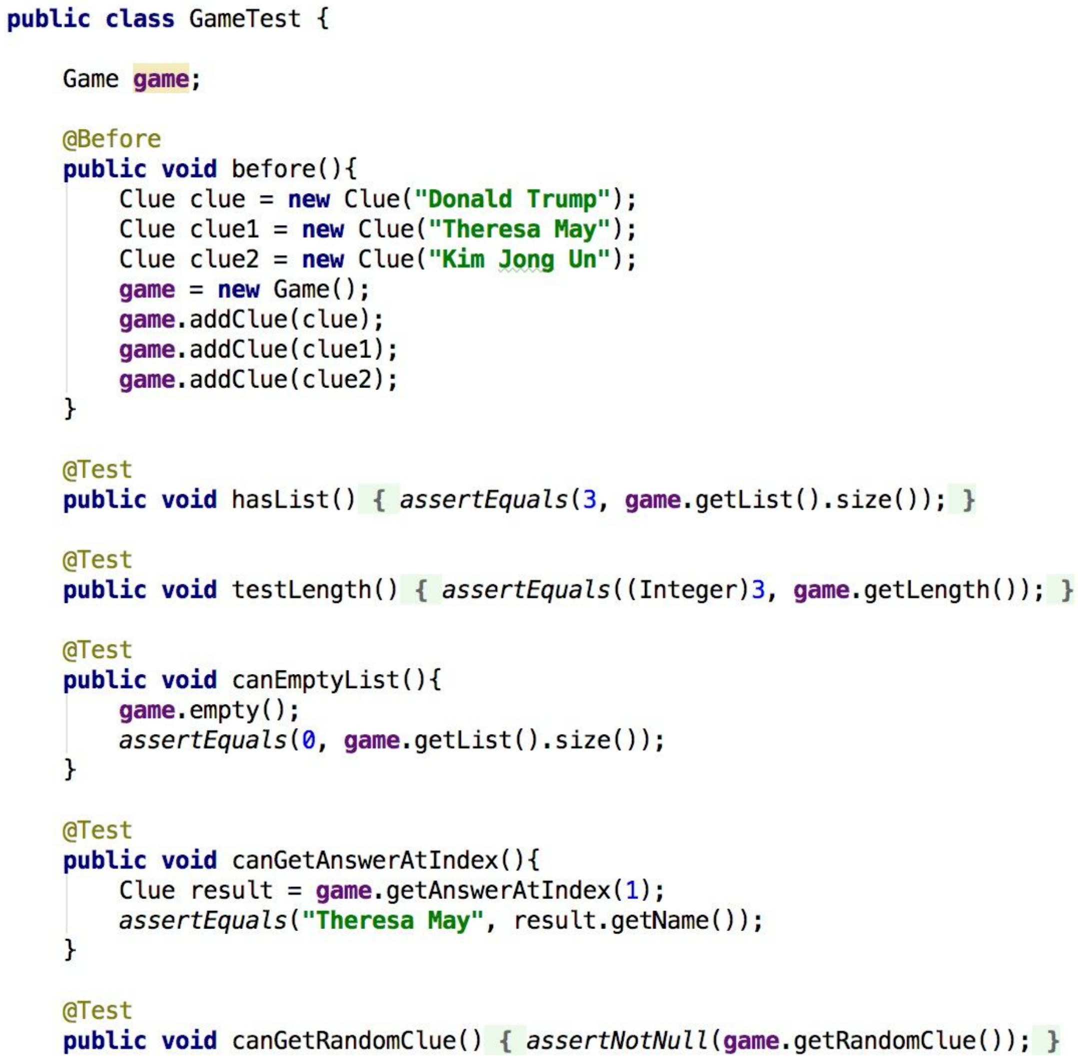
```
public class GameTest {

    Game game;

    @Before
    public void before(){
        Clue clue = new Clue("Donald Trump");
        Clue clue1 = new Clue("Theresa May");
        Clue clue2 = new Clue("Kim Jong Un");
        game = new Game();
        game.addClue(clue);
        game.addClue(clue1);
        game.addClue(clue2);
    }

    @Test
    public void hasList() { assertEquals(2, game.getList().size()); }

    @Test
    public void testLength() { assertEquals((Integer)2, game.getLength()); }

    @Test
    public void canEmptyList(){
        game.empty();
        assertEquals(1, game.getList().size());
    }
```

Tests failing



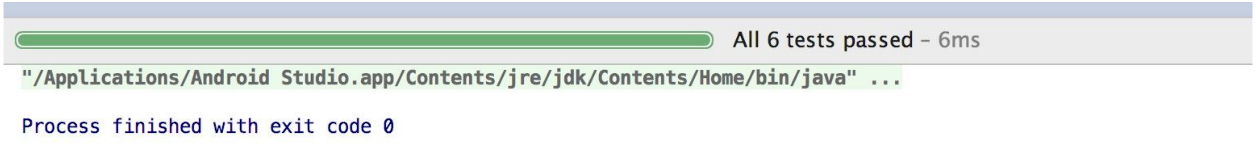
```
6 tests done: 5 failed – 75ms
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

java.lang.AssertionError:
Expected :1
Actual   :2
  <Click to see difference>


<1 internal calls>
    at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
    at com.example.user.thegame.GameTest.canRemoveAtIndex(GameTest.java:69) <28 internal calls>


java.lang.AssertionError:
Expected :2
Actual   :3
  <Click to see difference>


<1 internal calls>
    at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
    at com.example.user.thegame.GameTest.hasList(GameTest.java:34) <28 internal calls>
```

Code after changes



```java
public class GameTest {

    Game game;

    @Before
    public void before(){
        Clue clue = new Clue("Donald Trump");
        Clue clue1 = new Clue("Theresa May");
        Clue clue2 = new Clue("Kim Jong Un");
        game = new Game();
        game.addClue(clue);
        game.addClue(clue1);
        game.addClue(clue2);
    }

    @Test
    public void hasList() { assertEquals(3, game.getList().size()); }

    @Test
    public void testLength() { assertEquals((Integer)3, game.getLength()); }

    @Test
    public void canEmptyList(){
        game.empty();
        assertEquals(0, game.getList().size());
    }

    @Test
    public void canGetAnswerAtIndex(){
        Clue result = game.getAnswerAtIndex(1);
        assertEquals("Theresa May", result.getName());
    }

    @Test
    public void canGetRandomClue() { assertNotNull(game.getRandomClue()); }
```

Tests passing



```
All 6 tests passed – 6ms
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```