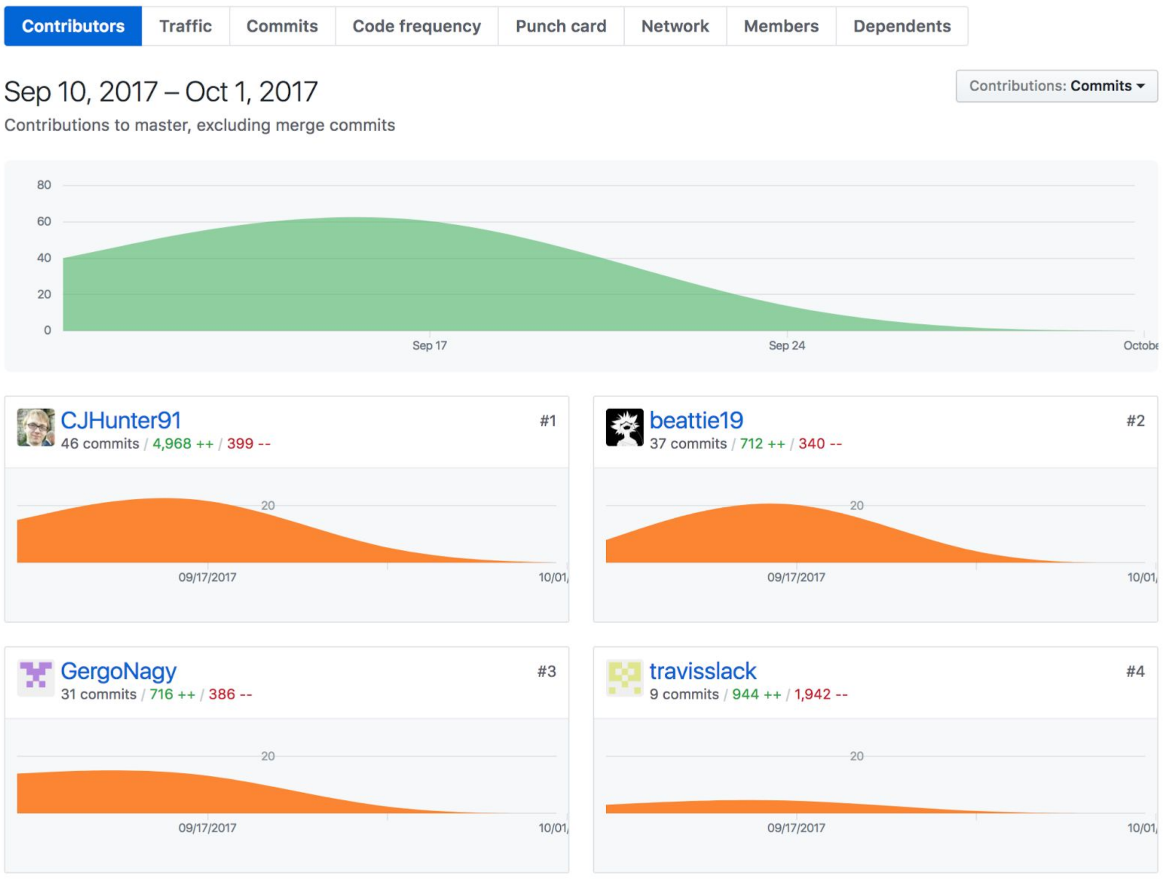
Gergo Nagy - Project Unit - Evidence (SQA PDA: Software Development)

P1 Group project - Education programming timeline



Sep 10, 2017 – Oct 1, 2017

Contributions to master, excluding merge commits

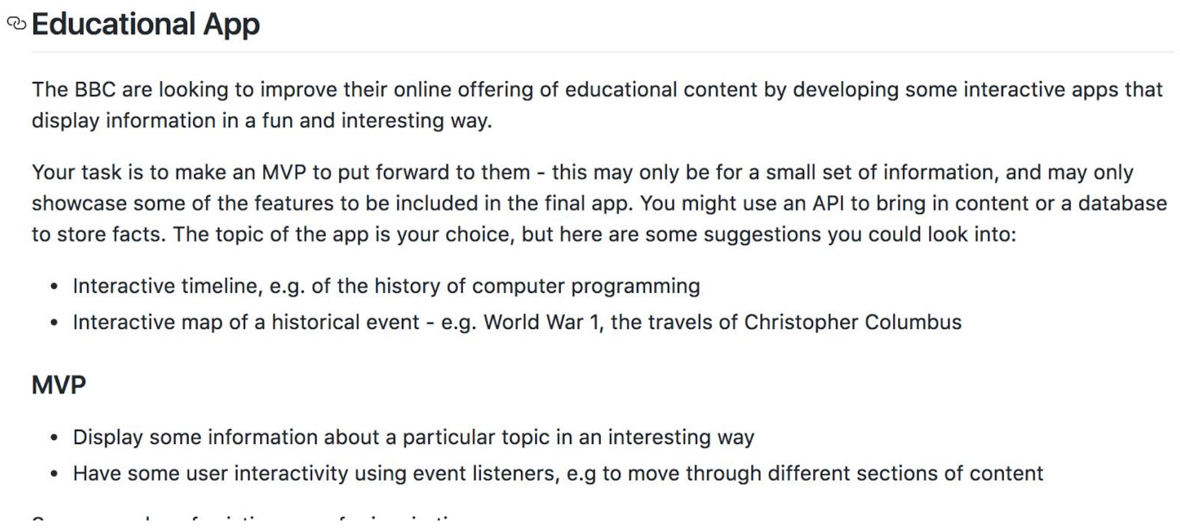
CJHunter91 #1
46 commits / 4,968 ++ / 399 --

beattie19 #2
37 commits / 712 ++ / 340 --

GergoNagy #3
31 commits / 716 ++ / 386 --

travisslack #4
9 commits / 944 ++ / 1,942 --

P2 Group Project Brief



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.

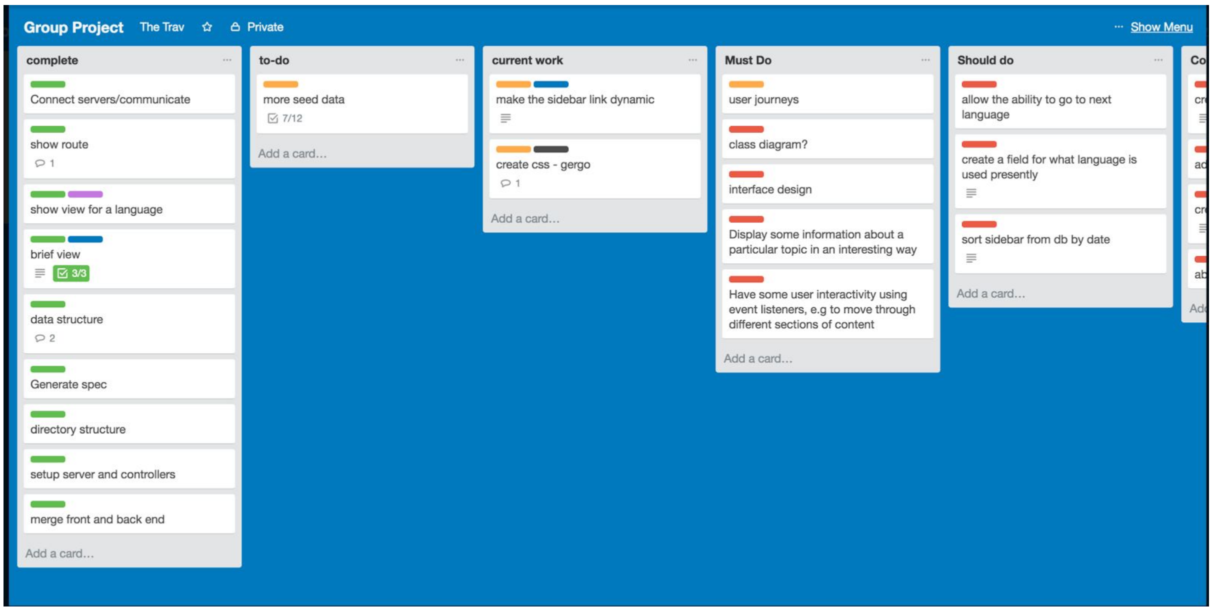
Your task is to make an MVP to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store facts. The topic of the app is your choice, but here are some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


Group Project The Trav Private ··· Show Menu

complete
- Connect servers/communicate
- show route 1
- show view for a language
- brief view 3/3
- data structure 2
- Generate spec
- directory structure
- setup server and controllers
- merge front and back end

Add a card...

to-do
- more seed data 7/12

Add a card...

current work
- make the sidebar link dynamic
- create css - gergo 1

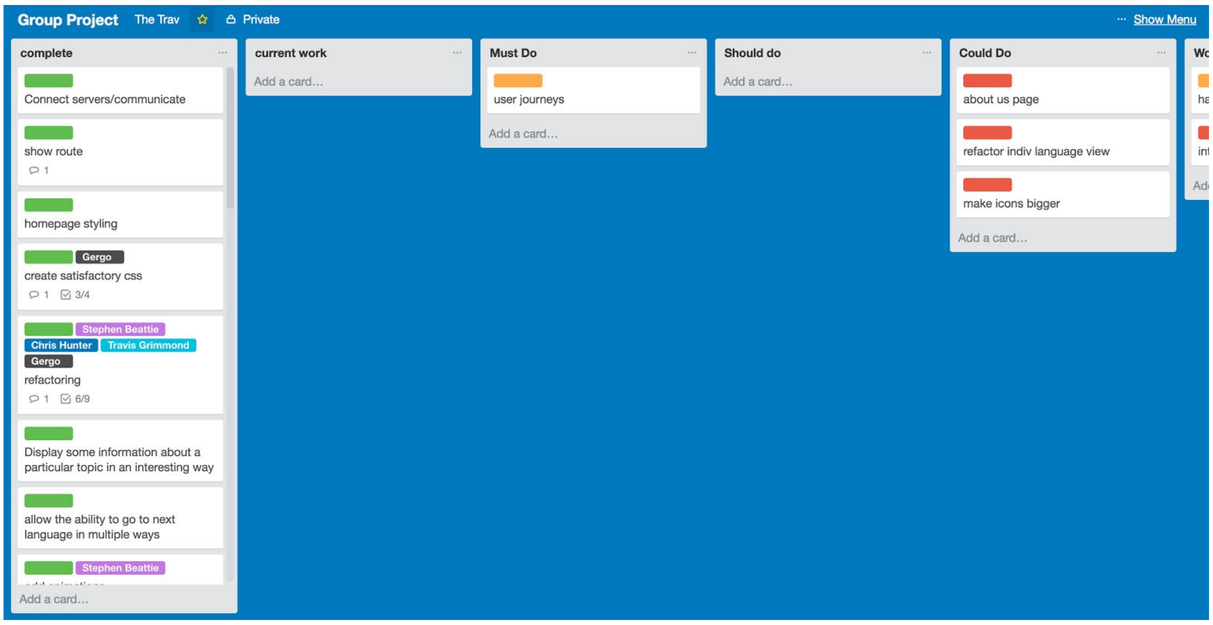
Add a card...

Must Do
- user journeys
- class diagram?
- interface design
- 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

Add a card...

Should do
- allow the ability to go to next language
- create a field for what language is used presently
- sort sidebar from db by date

Add a card...


Group Project The Trav Private ··· Show Menu

complete
- Connect servers/communicate
- show route 1
- homepage styling
- Gergo create satisfactory css 1 3/4
- Stephen Beattie Chris Hunter Travis Grimmond Gergo refactoring 1 6/9
- Display some information about a particular topic in an interesting way
- allow the ability to go to next language in multiple ways
- Stephen Beattie

Add a card...

current work

Add a card...

Must Do
- user journeys

Add a card...

Should do

Add a card...

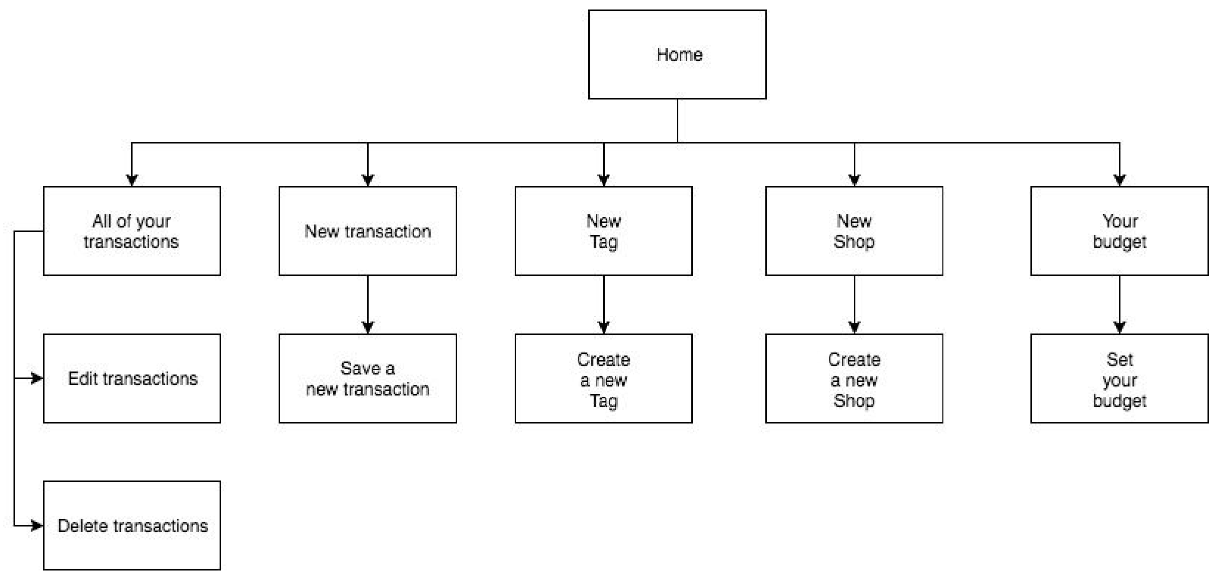
Could Do
- about us page
- refactor indiv language view
- make icons bigger

Add a card...

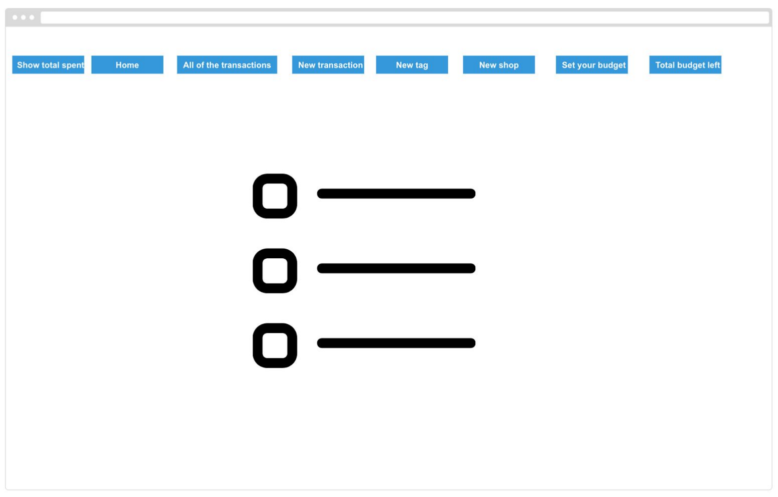
P 4 Acceptance Criteria

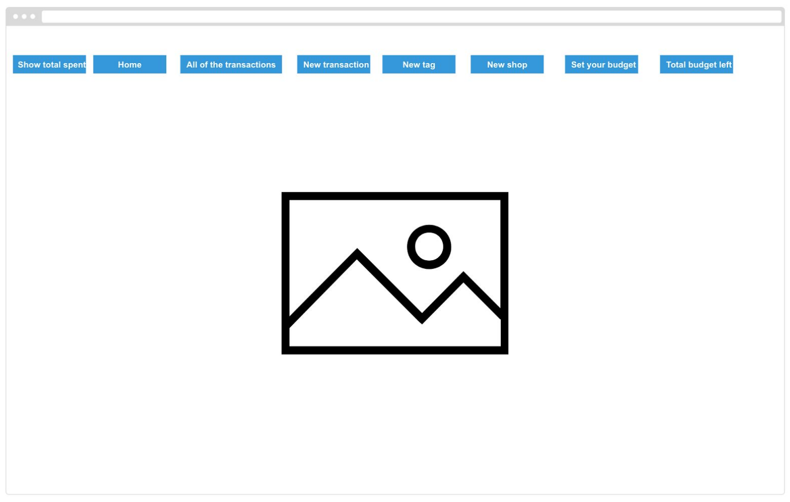
Acceptance Criteria	Expected Result/Output	Pass/Fail
A user can choose between programing language	User click on the icon on the timeline	Pass
A use can see the programing language page by page	Use click the back or next button to see the next language	Pass
A user can move on the timeline	User click on the next/prev button on the timeline	Pass

P 5 Site Map

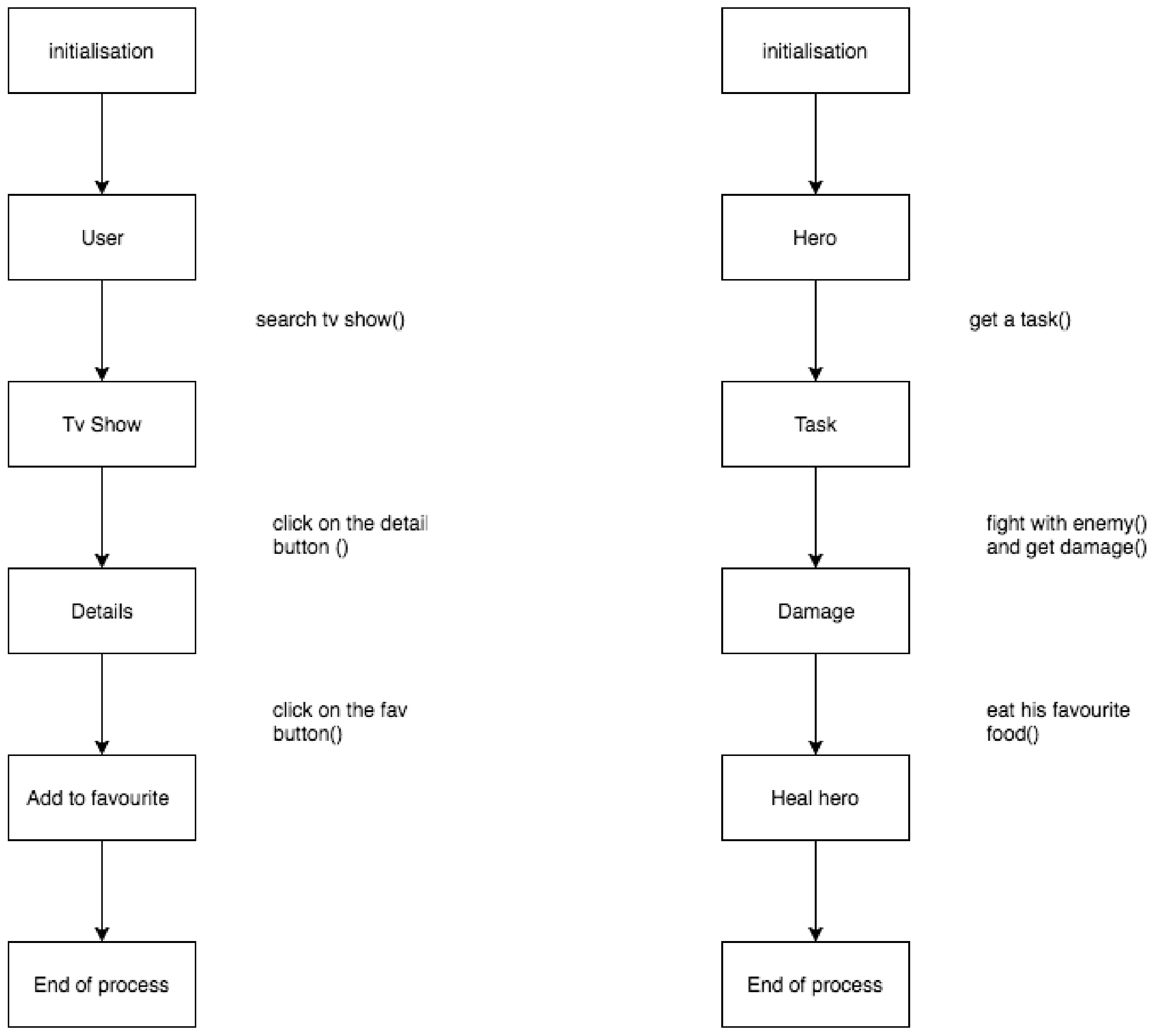


P 6 Wireframe:

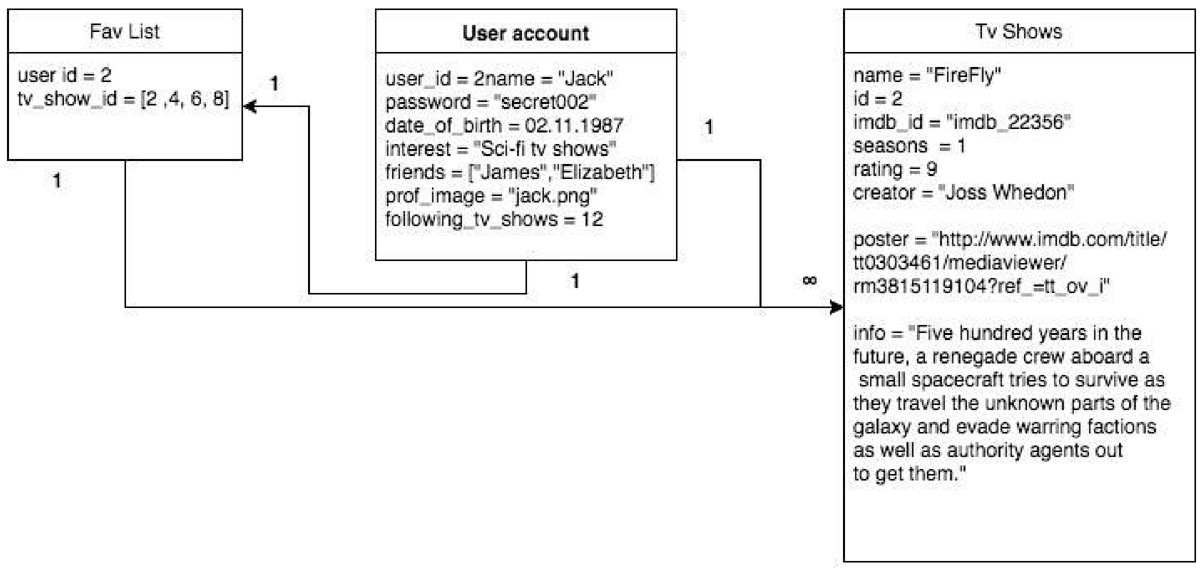


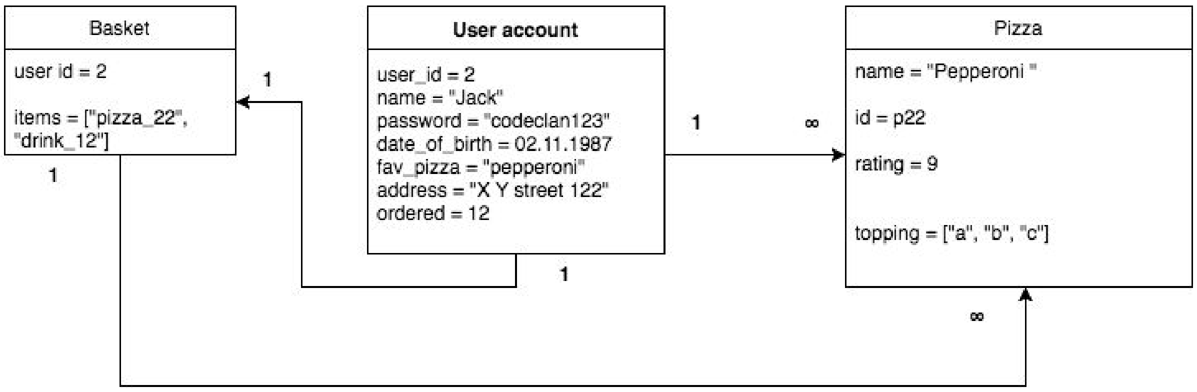


P7 System interaction diagrams



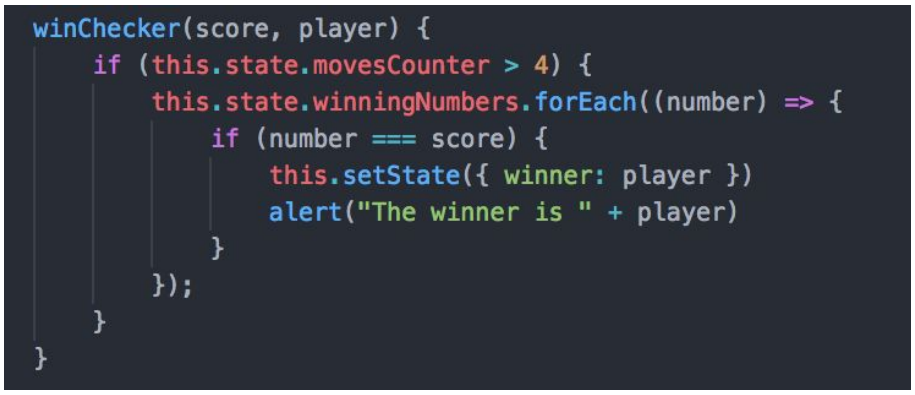
P8 Object diagrams


Fav List
user id = 2
tv_show_id = [2 ,4, 6, 8]
1
1
User account
user_id = 2name = "Jack"
password = "secret002"
date_of_birth = 02.11.1987
interest = "Sci-fi tv shows"
friends = ["James","Elizabeth"]
prof_image = "jack.png"
following_tv_shows = 12
1
1
∞
Tv Shows
name = "FireFly"
id = 2
imdb_id = "imdb_22356"
seasons = 1
rating = 9
creator = "Joss Whedon"
poster = "http://www.imdb.com/title/tt0303461/mediaviewer/rm3815119104?ref_=tt_ov_i"
info = "Five hundred years in the future, a renegade crew aboard a small spacecraft tries to survive as they travel the unknown parts of the galaxy and evade warring factions as well as authority agents out to get them."

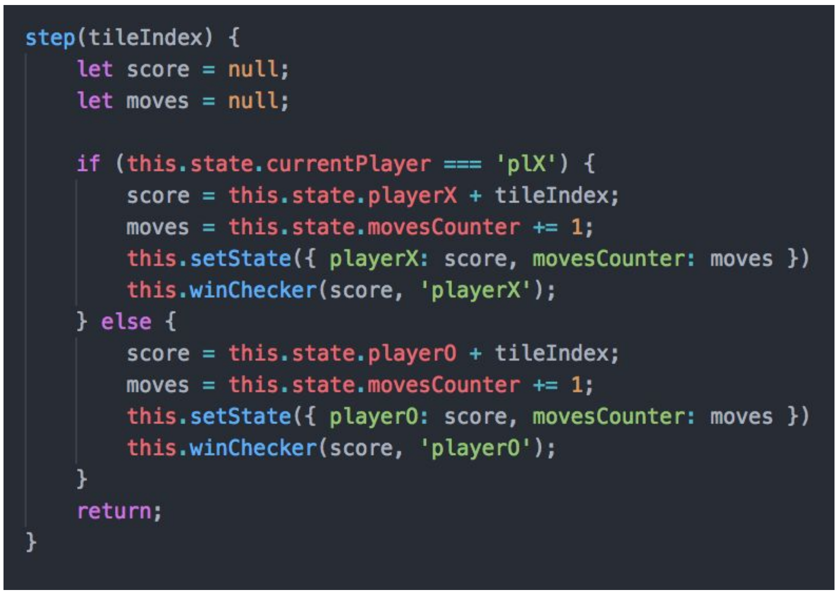


P9 Algorithms

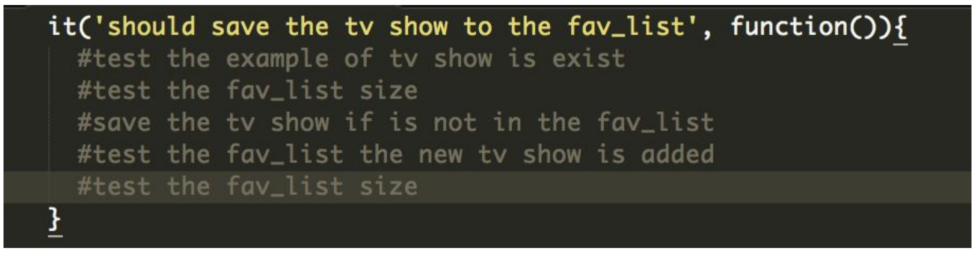
In this algorithm check a number of array if the moves counter bigger than 4.
If the win number is equal the player get the winner status end the game send a alert to the users with the winner name.


```
winChecker(score, player) {
    if (this.state.movesCounter > 4) {
        this.state.winningNumbers.forEach((number) => {
            if (number === score) {
                this.setState({ winner: player })
                alert("The winner is " + player)
            }
        });
    }
}
```

In this algorithm check which player had the last step and swap to the next player.
Also increase the player score and the move counter and calling another algorithm to check if the player score is equal with one of the winner numbers.

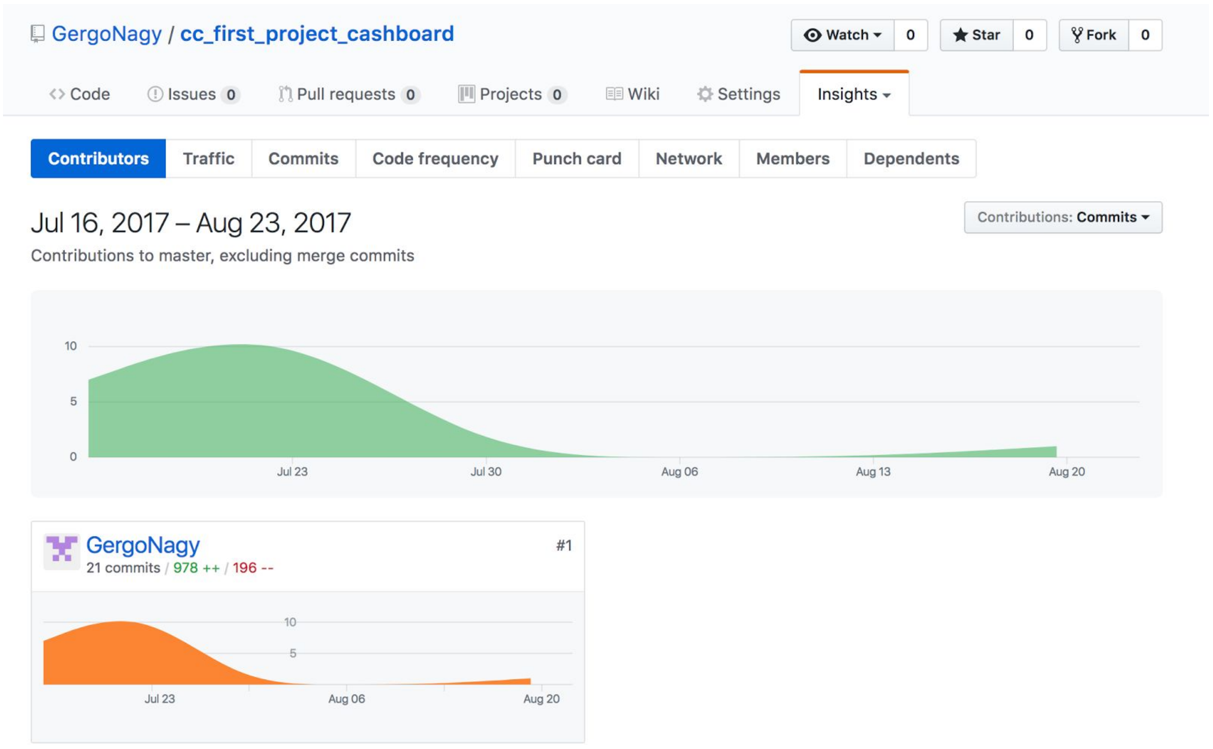

```
step(tileIndex) {
    let score = null;
    let moves = null;

    if (this.state.currentPlayer === 'plX') {
        score = this.state.playerX + tileIndex;
        moves = this.state.movesCounter += 1;
        this.setState({ playerX: score, movesCounter: moves })
        this.winChecker(score, 'playerX');
    } else {
        score = this.state.playerO + tileIndex;
        moves = this.state.movesCounter += 1;
        this.setState({ playerO: score, movesCounter: moves })
        this.winChecker(score, 'playerO');
    }
    return;
}
```

P 10 Pseudocode



```
it('should save the tv show to the fav_list', function(){
  #test the example of tv show is exist
  #test the fav_list size
  #save the tv show if is not in the fav_list
  #test the fav_list the new tv show is added
  #test the fav_list size
}
```

P 11 Project in Github



GergoNagy / cc_first_project_cashboard

Jul 16, 2017 – Aug 23, 2017

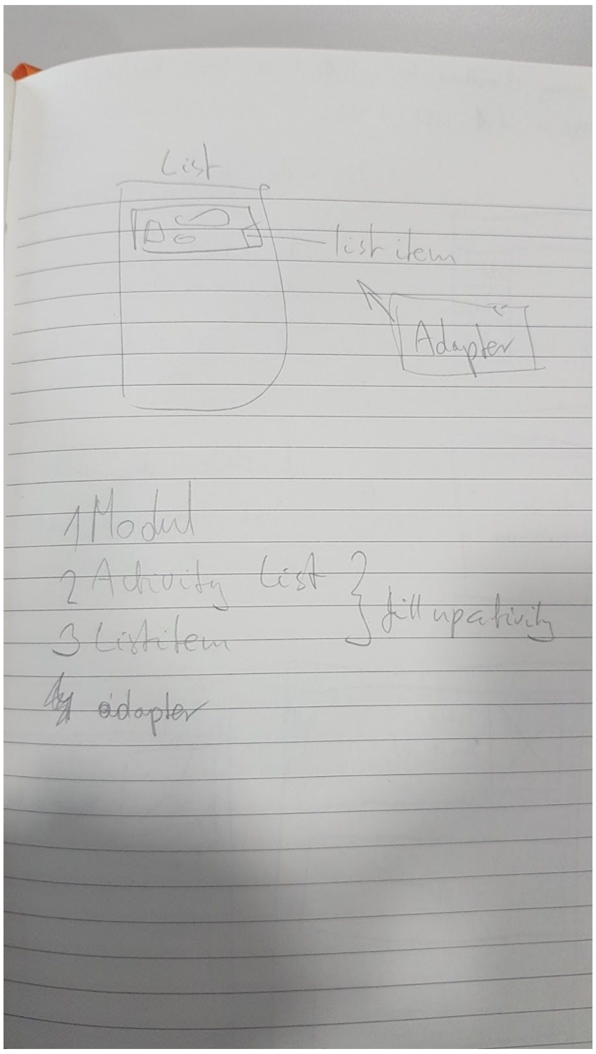
Contributions to master, excluding merge commits

GergoNagy #1

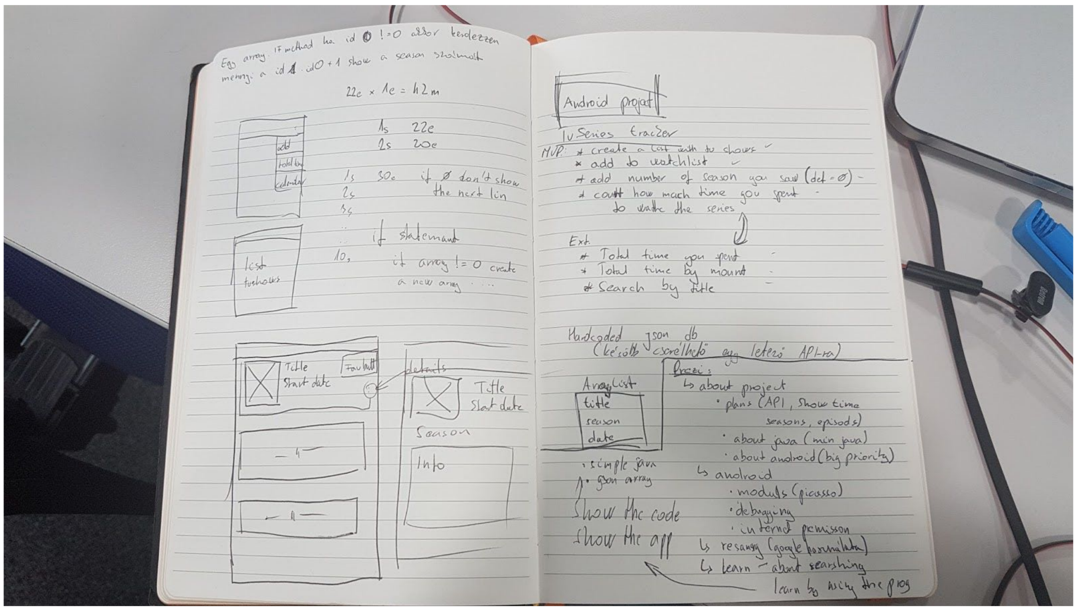
21 commits / 978 ++ / 196 --

P 12 Planning stages

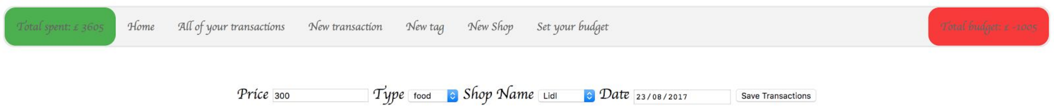
Stage 1:



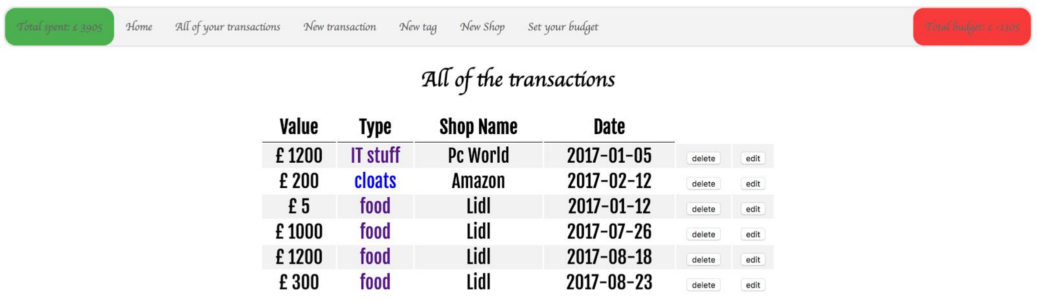
Stage 2:



P 13 Show user input being processed according to design requirements:



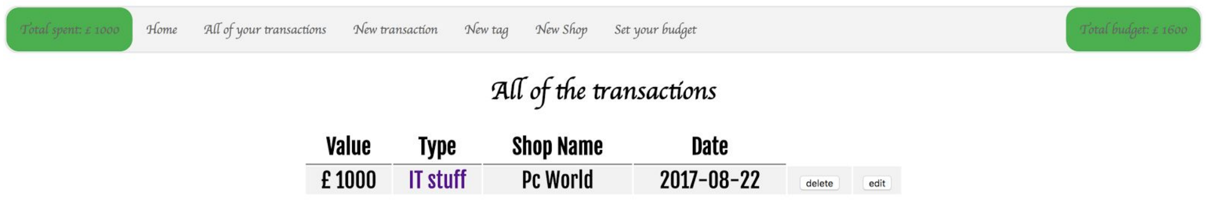
Price 300 Type food Shop Name Lidl Date 23/08/2017



All of the transactions

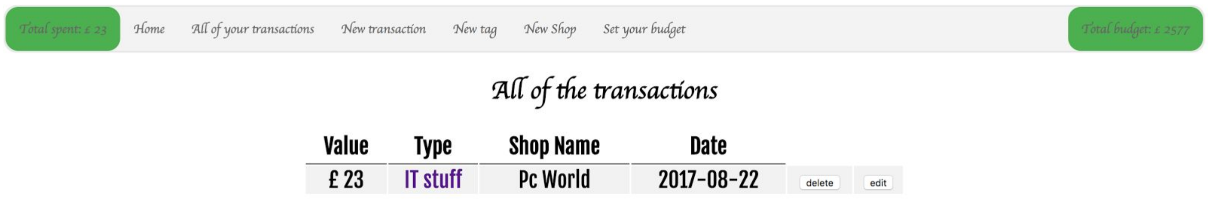
Value	Type	Shop Name	Date		
£ 1200	IT stuff	Pc World	2017-01-05	delete	edit
£ 200	cloats	Amazon	2017-02-12	delete	edit
£ 5	food	Lidl	2017-01-12	delete	edit
£ 1000	food	Lidl	2017-07-26	delete	edit
£ 1200	food	Lidl	2017-08-18	delete	edit
£ 300	food	Lidl	2017-08-23	delete	edit

P 14 Show an interaction with data persistence:



All of the transactions

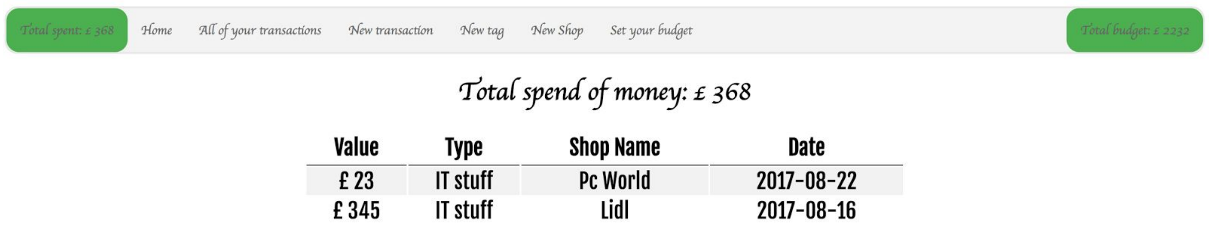
Value	Type	Shop Name	Date		
£ 1000	IT stuff	Pc World	2017-08-22	delete	edit



All of the transactions

Value	Type	Shop Name	Date		
£ 23	IT stuff	Pc World	2017-08-22	delete	edit

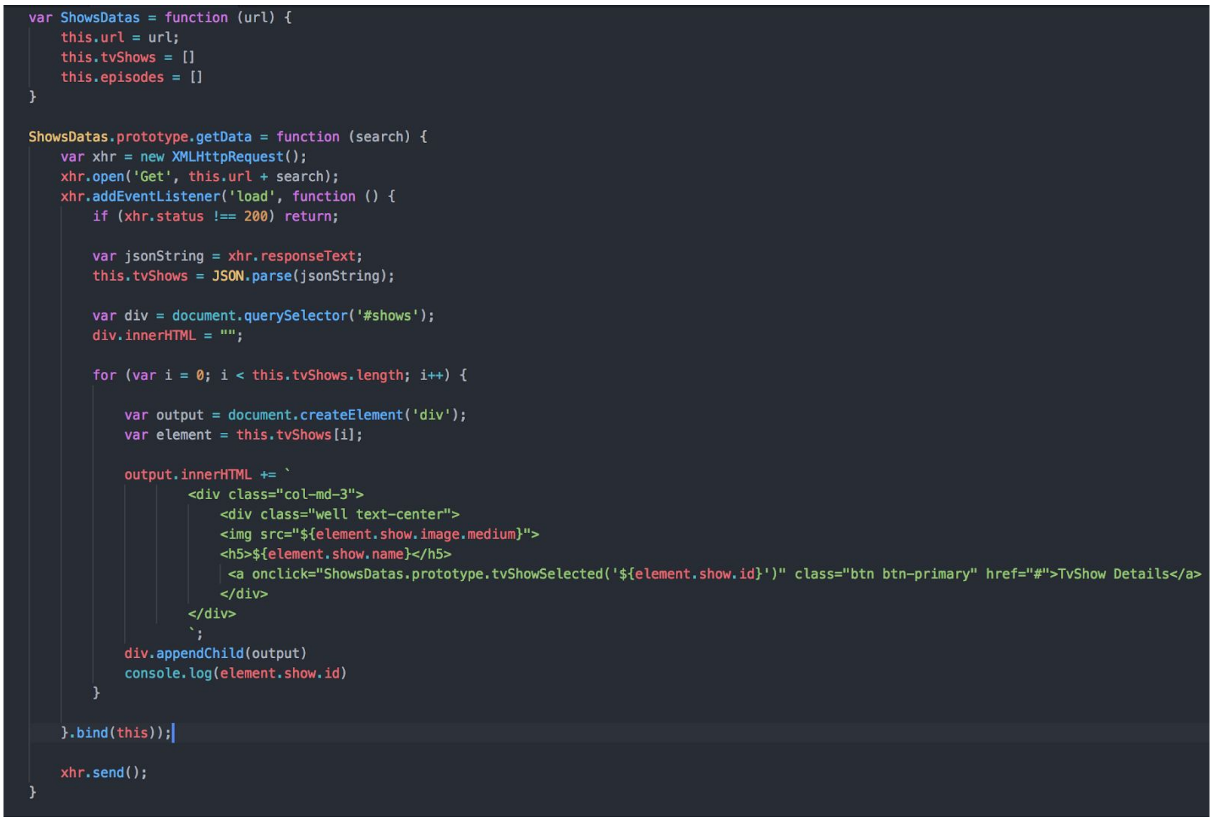
P 15 Show the correct output of results and feedback to user:



Total spend of money: £ 368

Value	Type	Shop Name	Date
£ 23	IT stuff	Pc World	2017-08-22
£ 345	IT stuff	Lidl	2017-08-16

P16 API being used



```
var ShowsDatas = function (url) {
    this.url = url;
    this.tvShows = []
    this.episodes = []
}

ShowsDatas.prototype.getData = function (search) {
    var xhr = new XMLHttpRequest();
    xhr.open('Get', this.url + search);
    xhr.addEventListener('load', function () {
        if (xhr.status !== 200) return;

        var jsonString = xhr.responseText;
        this.tvShows = JSON.parse(jsonString);

        var div = document.querySelector('#shows');
        div.innerHTML = "";

        for (var i = 0; i < this.tvShows.length; i++) {

            var output = document.createElement('div');
            var element = this.tvShows[i];

            output.innerHTML += `
                    <div class="col-md-3">
                        <div class="well text-center">
                        <img src="${element.show.image.medium}">
                        <h5>${element.show.name}</h5>
                         <a onclick="ShowsDatas.prototype.tvShowSelected('${element.show.id}')" class="btn btn-primary" href="#">TvShow Details</a>
                        </div>
                    </div>
                    `;
            div.appendChild(output)
            console.log(element.show.id)
        }

    }.bind(this));

    xhr.send();
}
```

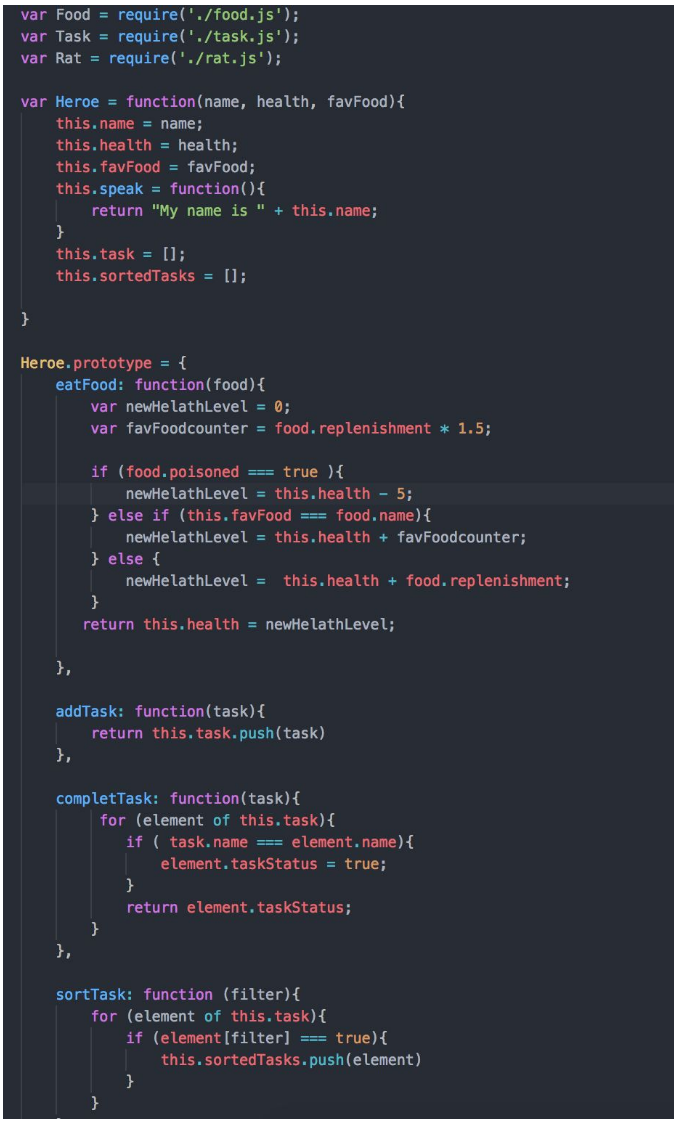
P17 Bug tracking report

A user can search tv shows	Fail	Use search on a tv show api by title	Pass
A user can see the details of the tv show	Fail	Create a details button to show the details	Pass
A user can see the Imdb rate	Fail	Use a imdb api the show the rate of tv show on the details page	Pass
A use can see the tv show is still running	Fail	Create a section to show the channel and the day when the tv show is running	Pass
A use can see a bigger size of the poster	Fail	Use different size setting on the details page	Pass

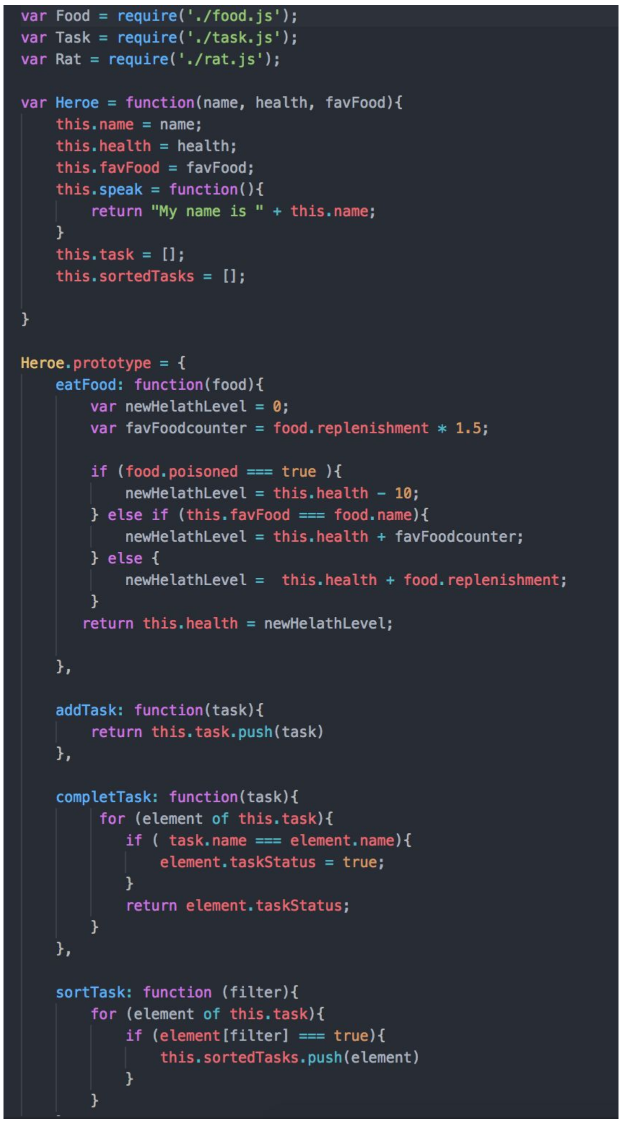
P18 Demonstration of testing

-example of the code

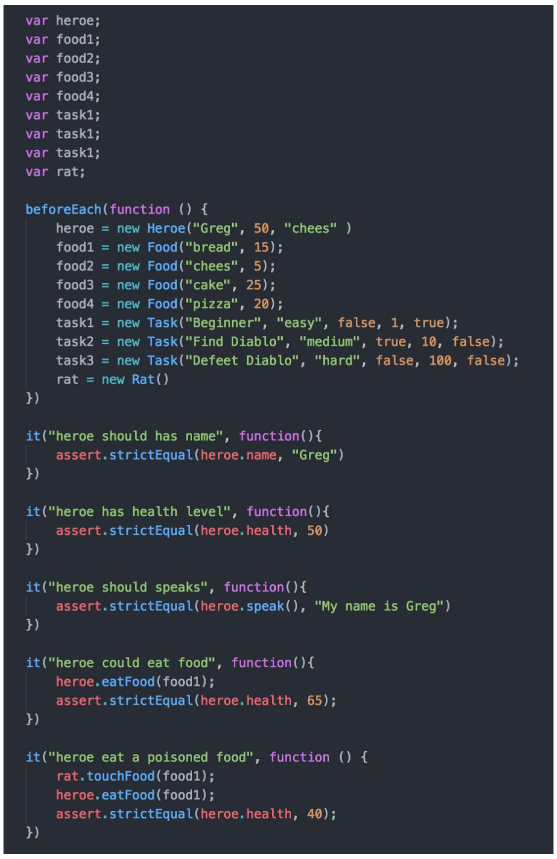
-example of the corrected code



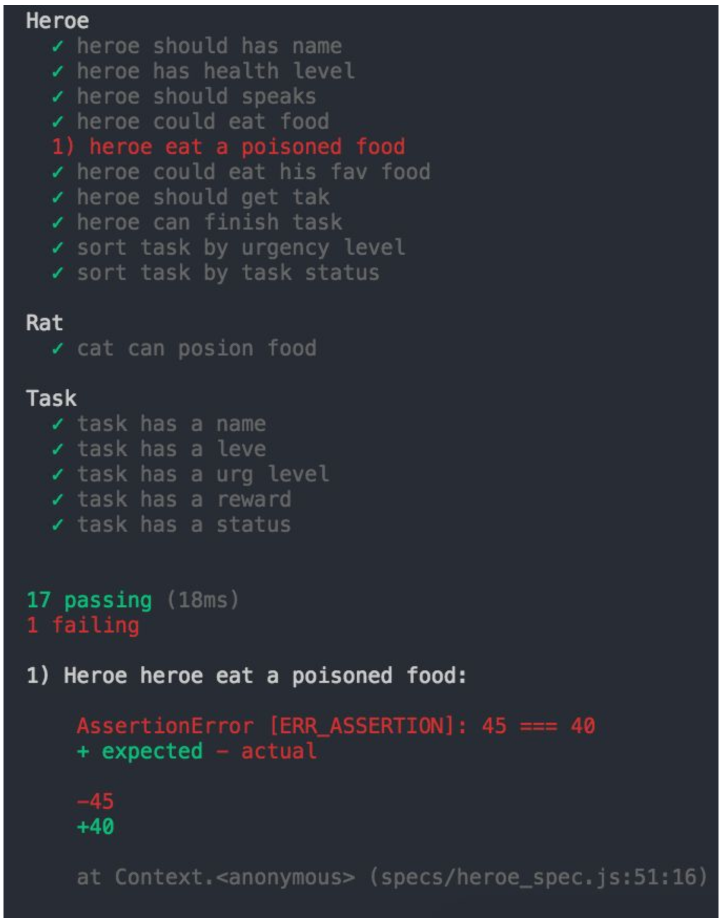
```
var Food = require('./food.js');
var Task = require('./task.js');
var Rat = require('./rat.js');

var Heroe = function(name, health, favFood){
    this.name = name;
    this.health = health;
    this.favFood = favFood;
    this.speak = function(){
        return "My name is " + this.name;
    }
    this.task = [];
    this.sortedTasks = [];

}

Heroe.prototype = {
    eatFood: function(food){
        var newHelathLevel = 0;
        var favFoodcounter = food.replenishment * 1.5;

        if (food.poisoned === true ){
            newHelathLevel = this.health - 5;
        } else if (this.favFood === food.name){
            newHelathLevel = this.health + favFoodcounter;
        } else {
            newHelathLevel =  this.health + food.replenishment;
        }
       return this.health = newHelathLevel;

    },

    addTask: function(task){
        return this.task.push(task)
    },

    completTask: function(task){
         for (element of this.task){
            if ( task.name === element.name){
                element.taskStatus = true;
            }
            return element.taskStatus;
        }
    },

    sortTask: function (filter){
        for (element of this.task){
            if (element[filter] === true){
                this.sortedTasks.push(element)
            }
        }
```



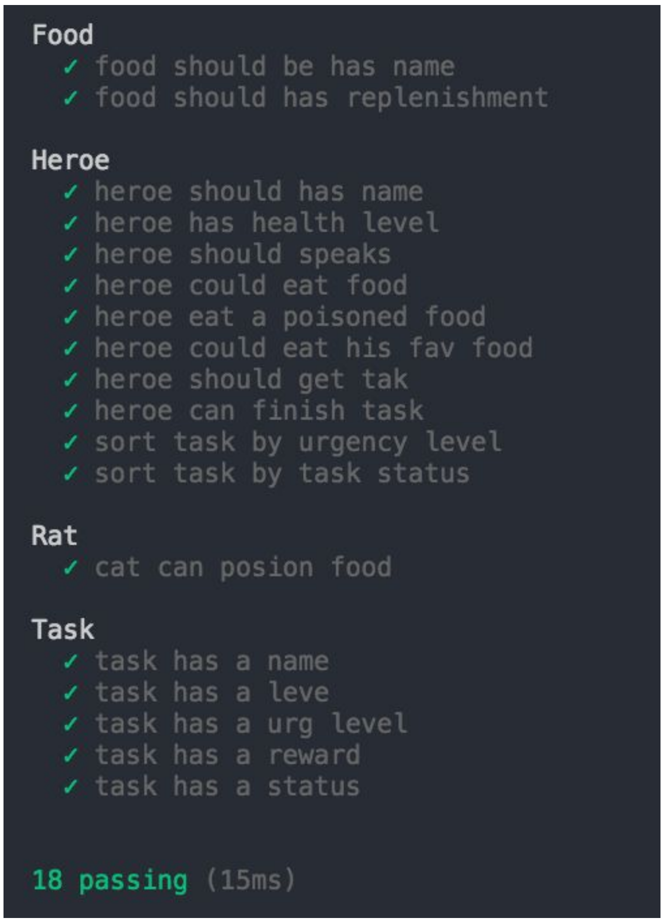
```
var Food = require('./food.js');
var Task = require('./task.js');
var Rat = require('./rat.js');

var Heroe = function(name, health, favFood){
    this.name = name;
    this.health = health;
    this.favFood = favFood;
    this.speak = function(){
        return "My name is " + this.name;
    }
    this.task = [];
    this.sortedTasks = [];

}

Heroe.prototype = {
    eatFood: function(food){
        var newHelathLevel = 0;
        var favFoodcounter = food.replenishment * 1.5;

        if (food.poisoned === true ){
            newHelathLevel = this.health - 10;
        } else if (this.favFood === food.name){
            newHelathLevel = this.health + favFoodcounter;
        } else {
            newHelathLevel =  this.health + food.replenishment;
        }
       return this.health = newHelathLevel;

    },

    addTask: function(task){
        return this.task.push(task)
    },

    completTask: function(task){
         for (element of this.task){
            if ( task.name === element.name){
                element.taskStatus = true;
            }
            return element.taskStatus;
        }
    },

    sortTask: function (filter){
        for (element of this.task){
            if (element[filter] === true){
                this.sortedTasks.push(element)
            }
        }
```



```
var heroe;
var food1;
var food2;
var food3;
var food4;
var task1;
var task1;
var task1;
var rat;

beforeEach(function () {
    heroe = new Heroe("Greg", 50, "chees" )
    food1 = new Food("bread", 15);
    food2 = new Food("chees", 5);
    food3 = new Food("cake", 25);
    food4 = new Food("pizza", 20);
    task1 = new Task("Beginner", "easy", false, 1, true);
    task2 = new Task("Find Diablo", "medium", true, 10, false);
    task3 = new Task("Defeet Diablo", "hard", false, 100, false);
    rat = new Rat()
})

it("heroe should has name", function(){
    assert.strictEqual(heroe.name, "Greg")
})

it("heroe has health level", function(){
    assert.strictEqual(heroe.health, 50)
})

it("heroe should speaks", function(){
    assert.strictEqual(heroe.speak(), "My name is Greg")
})

it("heroe could eat food", function(){
    heroe.eatFood(food1);
    assert.strictEqual(heroe.health, 65);
})

it("heroe eat a poisoned food", function () {
    rat.touchFood(food1);
    heroe.eatFood(food1);
    assert.strictEqual(heroe.health, 40);
})
```



```
Heroe
  ✓ heroe should has name
  ✓ heroe has health level
  ✓ heroe should speaks
  ✓ heroe could eat food
  1) heroe eat a poisoned food
  ✓ heroe could eat his fav food
  ✓ heroe should get tak
  ✓ heroe can finish task
  ✓ sort task by urgency level
  ✓ sort task by task status

Rat
  ✓ cat can posion food

Task
  ✓ task has a name
  ✓ task has a leve
  ✓ task has a urg level
  ✓ task has a reward
  ✓ task has a status


17 passing (18ms)
1 failing

1) Heroe heroe eat a poisoned food:

    AssertionError [ERR_ASSERTION]: 45 === 40
    + expected - actual

    -45
    +40

    at Context.<anonymous> (specs/heroe_spec.js:51:16)
```



```
Food
  ✓ food should be has name
  ✓ food should has replenishment

Heroe
  ✓ heroe should has name
  ✓ heroe has health level
  ✓ heroe should speaks
  ✓ heroe could eat food
  ✓ heroe eat a poisoned food
  ✓ heroe could eat his fav food
  ✓ heroe should get tak
  ✓ heroe can finish task
  ✓ sort task by urgency level
  ✓ sort task by task status

Rat
  ✓ cat can posion food

Task
  ✓ task has a name
  ✓ task has a leve
  ✓ task has a urg level
  ✓ task has a reward
  ✓ task has a status


18 passing (15ms)
```