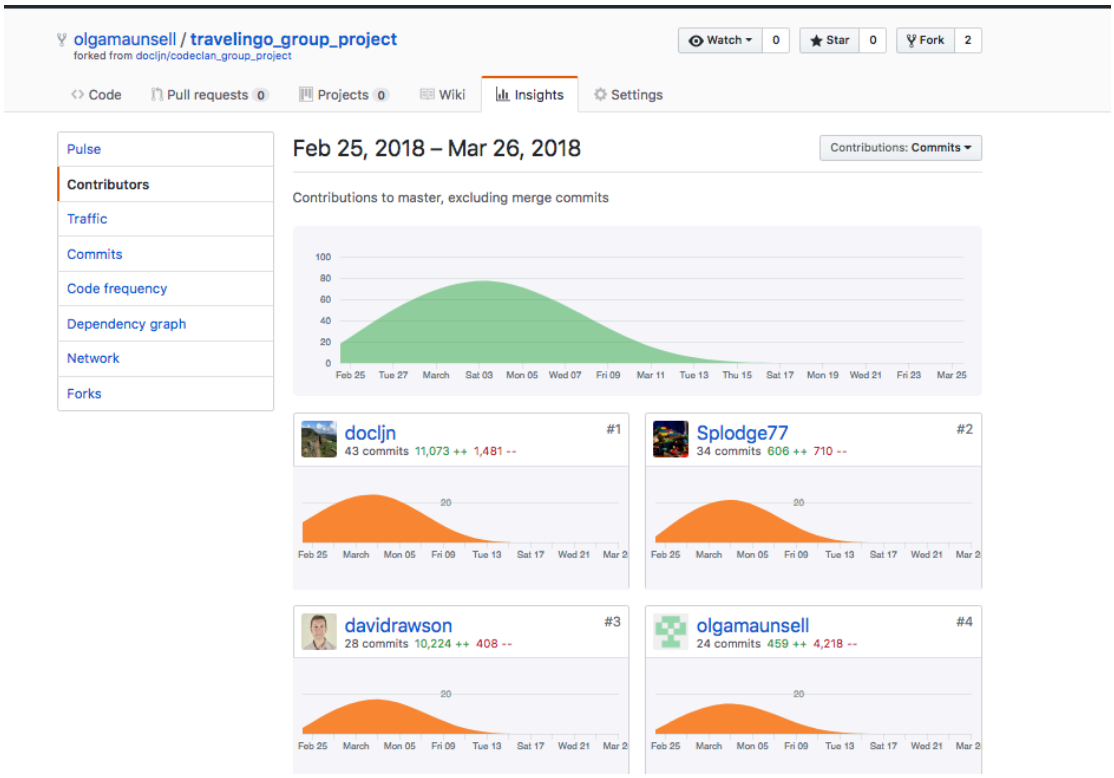
Evidence for Project Unit

Name: Olga Maunsell
Cohort: E18

P1 – Contributors page – Group project



P2 – Screenshot of project brief – 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

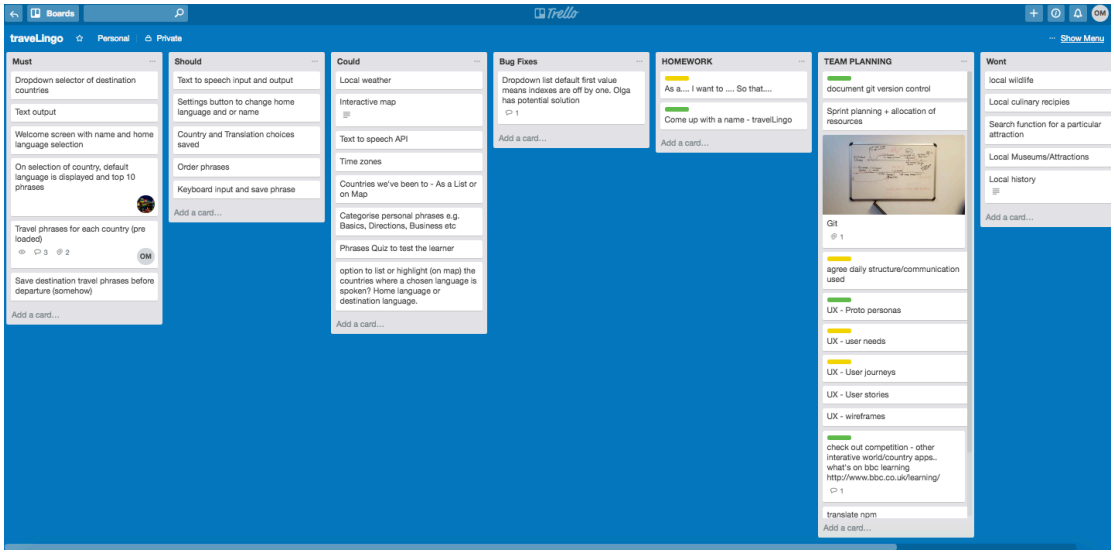
- Display some information about a particular topic in an interesting way
- Have some user interactivity using event listeners, e.g to move through different sections of content

Some samples of existing apps for inspiration:

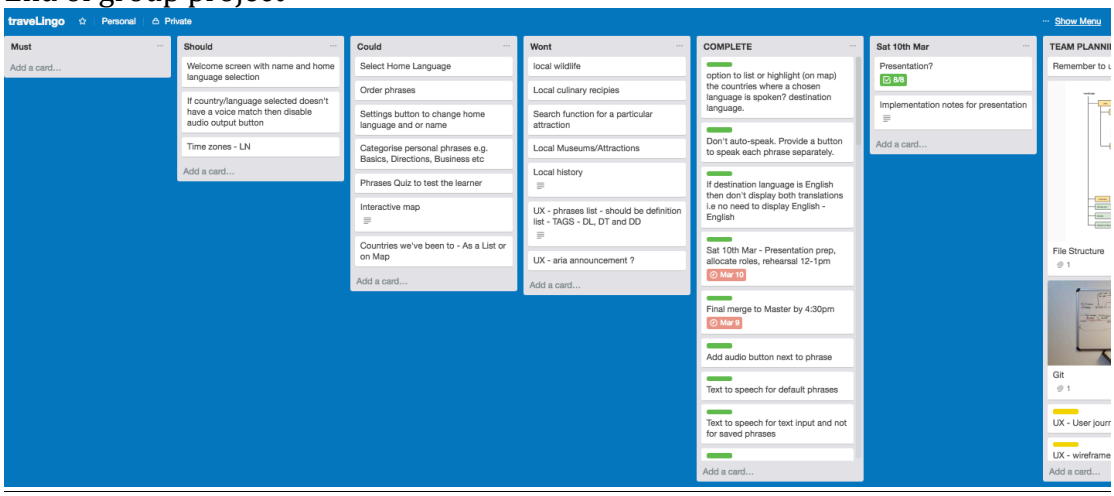
- http://chemistryset.chemheritage.org/#/
- http://www.royalmailheritage.com/main.php
- http://education.iceandsky.com/
- http://histography.io - may only work in Safari
- http://worldpopulationhistory.org/map/1838/mercator/1/0/24/

P3 – Planning – Group Project

Start of group project



End of group project

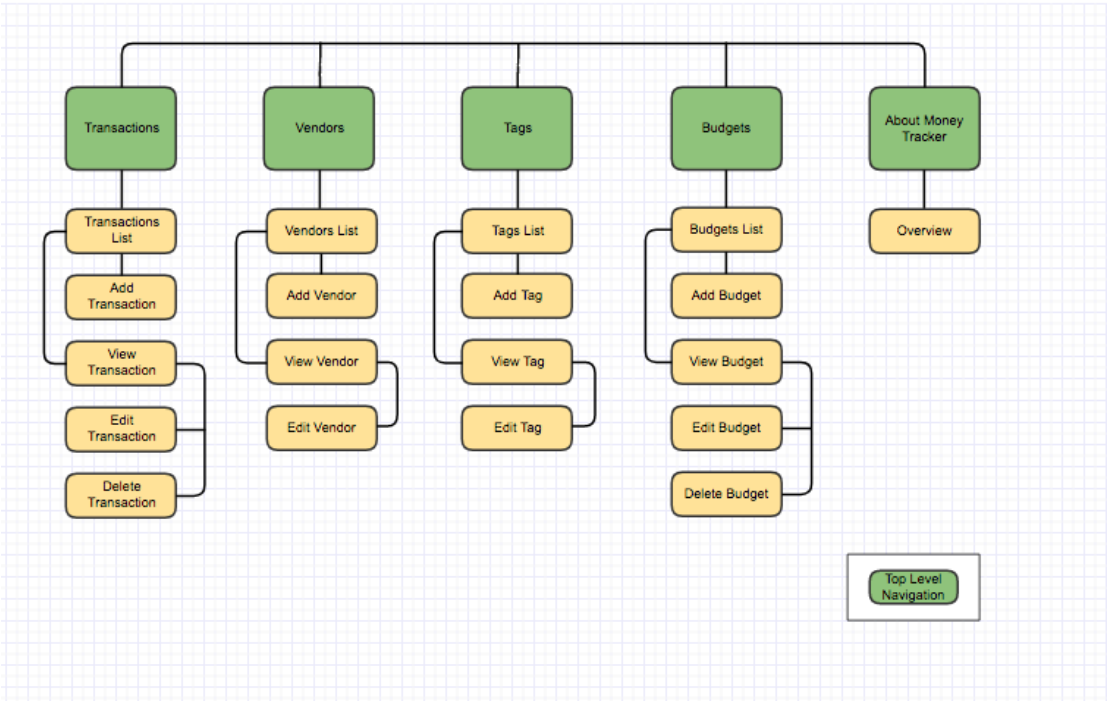


P4 – Money Tracker App – Acceptance Criteria and Testplan

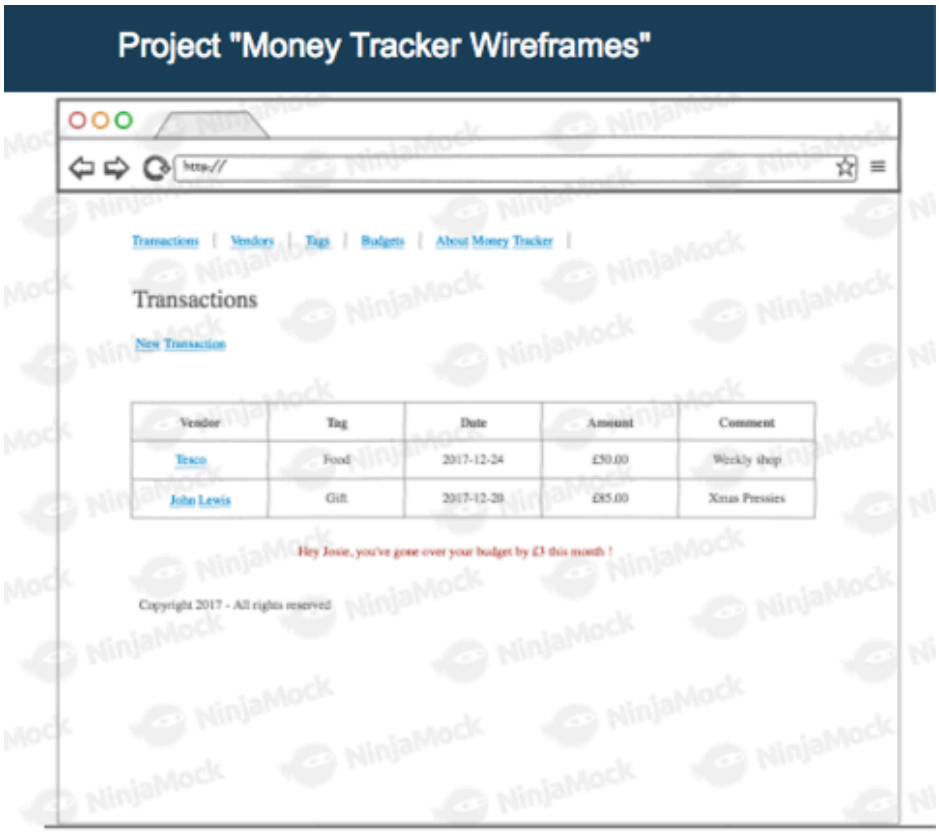
Acceptance Criteria	Expected Result/Output	Pass/Fail
An option is provided to add a new transaction	When a user selects "New Transaction" a new screen will appear allowing user to enter Vendor, Tag, Amount, Date, Comment and save the transaction	Pass
When a user enters and saves a transaction, transaction is created and can be viewed in the transaction list.	When user enters transaction details and saves transaction a message "Transaction created" is displayed. The new transaction is now also displayed on the main transactions screen.	Pass
When a user enters a transaction but does not select the save button, the transaction is not saved	Transaction is not saved. Transaction is not displayed on the main transactions screen.	Pass
User can view all transactions added	The main transactions screen lists all transactions added by the user in descending date order	Pass
User can view total amount of all transactions	The total amount of all transactions is displayed to 2 decimal places at the bottom of the transactions list.	Pass
User can select an individual transaction	User can select a transaction from the main transactions screen , a new screen will appear displaying the transaction details with options to edit or delete	Pass
User can edit a transaction	When user selects to edit a transaction a new screen is displayed allowing user to edit and update transaction details. On update the main transaction screen will display the new details.	Pass
User can delete a transaction	When user selects to delete a transaction, the transaction is deleted. The transaction can no longer be viewed on the main transactions screen	Pass

P5 – User Site Map – Money Tracker App

Gliffy / **Money Tracker Sitemap, v2**

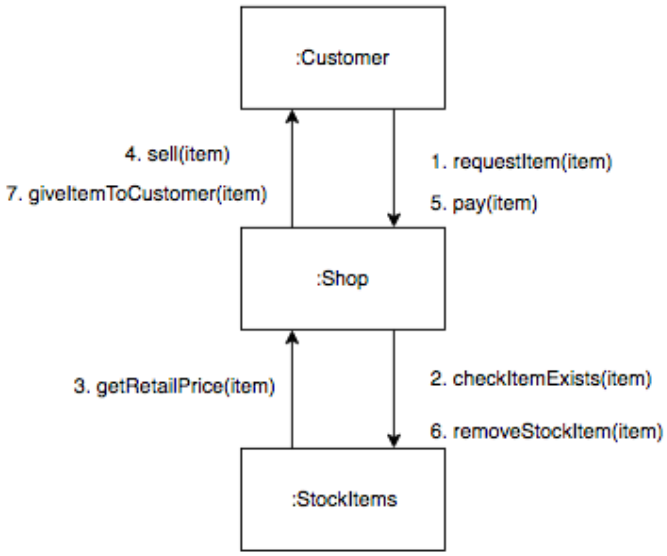


P6 – Two Wireframe Designs – Money Tracker App

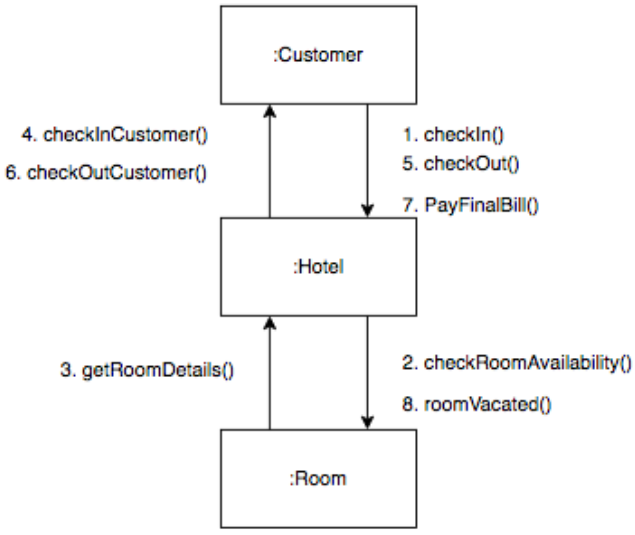


P7 – System Interaction Diagrams

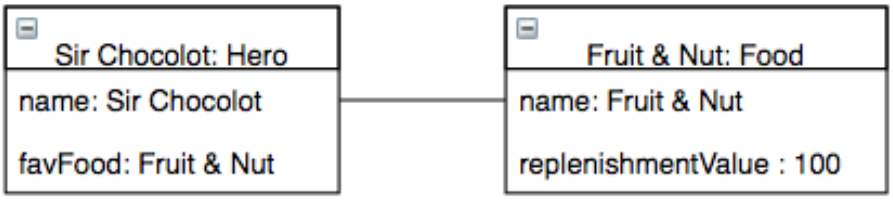
Music Shop - Collaboration Diagram

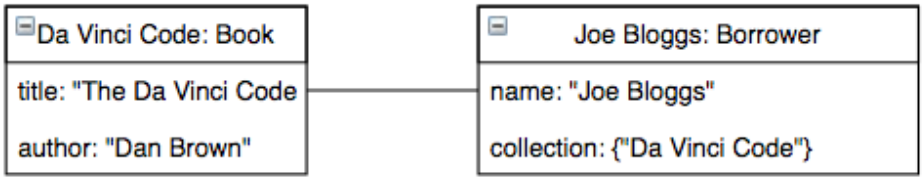


Hotel - Collaboration Diagram



P8 – Object Diagrams





P9 – 2 Algorithms
https://github.com/olgamaunsell/week_11_day3_enumeration_hw

I chose this as the code is easy to follow, so good for another coder to understand. This function's purpose is to find duplicate values in array, returning a new array of the duplicates

The function has 2 algorithms:

i) Creating a countHash – the code loops through the array, for each number in the array if the number doesn't exist in the hash the number is added to the hash with a value of 1 otherwise if it exists the value is incremented by 1.

ii) Builds a new duplicatesArray – the code retrieves the keys of the countHash, for each key it checks if the key's value > 1 i.e. it's a duplicate and if so it puts the number in the duplicates array. The duplicates array is returned by the function.

```
61          findDuplicates: function (arr) {
62
63                  let countHash = {};
64
65                  arr.forEach(function(number){
66                          if (number in countHash){
67                                  countHash[number] +=1;
68                          }
69                          else{
70                                  countHash[number] = 1;
71                          }
72                  });
73
74                  const keys = Object.keys(countHash);
75
76                  let duplicatesArray = [];
77                  for (key of keys) {
78                          if (countHash[key] > 1){
79                                  const number = parseInt(key);
80                                  duplicatesArray.push(number);
81                          }
82                  }
83
84                  return duplicatesArray;
85          },
```

Test

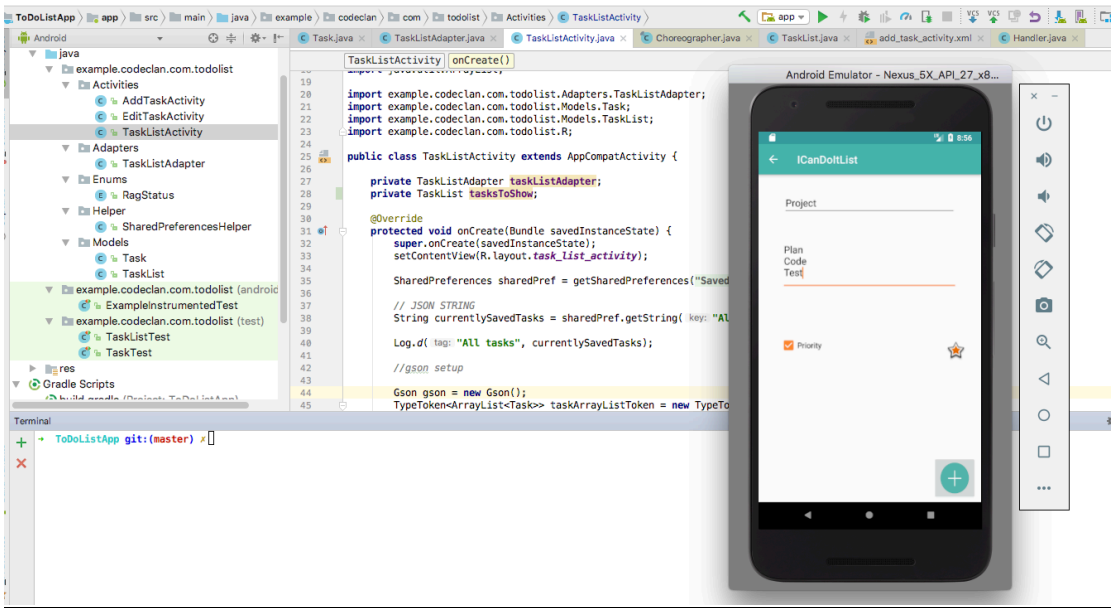
```
it('EXTENSION - should find duplicate values in an array, returning a new array of the duplicates', function () {
        const arr = [1, 2, 3, 4, 4, 5, 5, 5]
        assert.deepStrictEqual(arrayTasks.findDuplicates(arr), [4, 5])
})
```

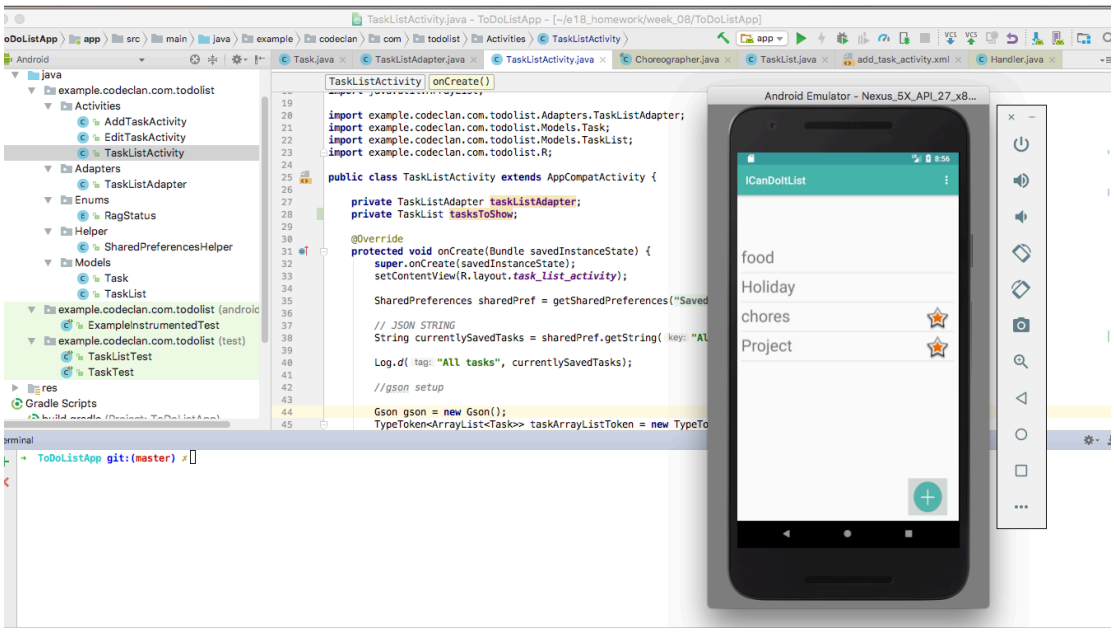
P 10 - An example of pseudocode for a function

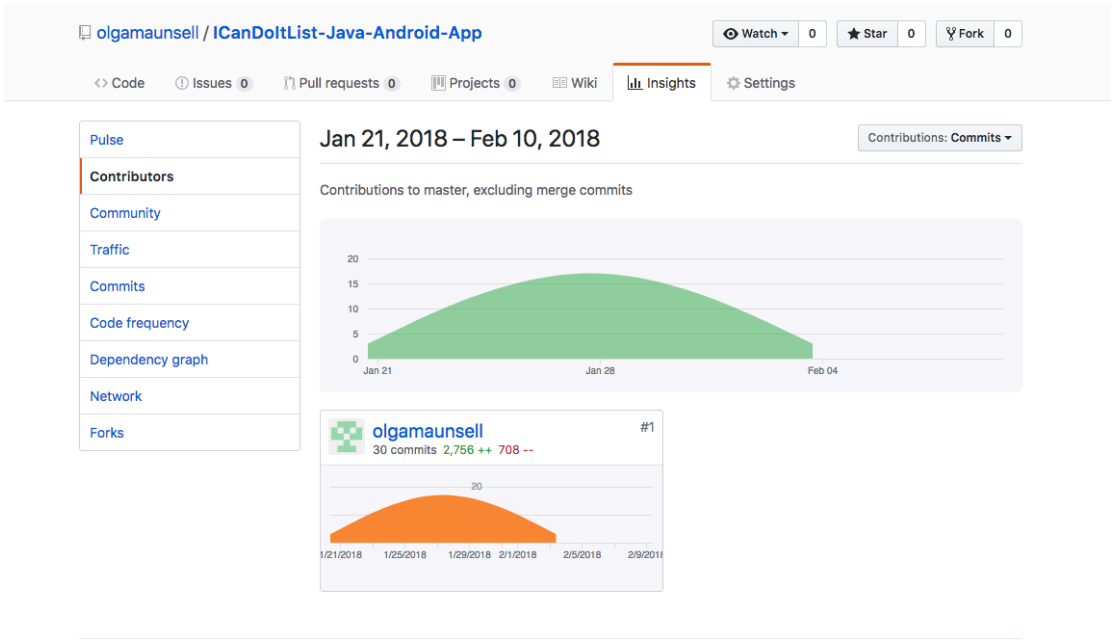
```
#  Using the month number and year received
# Sum the amount value for all transactions where the transaction date month and year
match the month number and year
# Return the total amount for these selected transactions rounded to 2 decimal places.

    def self.mth_yr_tot_amt(month_no, year)
      sql = "SELECT SUM (amount) FROM transactions
      WHERE EXTRACT(MONTH FROM transaction_date) = $1 AND
      EXTRACT(YEAR FROM transaction_date) = $2"

      values = [month_no, year]
      mth_yr_tot_amt = SqlRunner.run(sql, values)
      return mth_yr_tot_amt.first()['sum'].to_f.round(2)

    end
```

P11 – Screenshots of my 2nd Individual project - a to do list app I called "ICanDoItList"

https://github.com/olgamaunsell/ICanDoItList-Java-Android-App

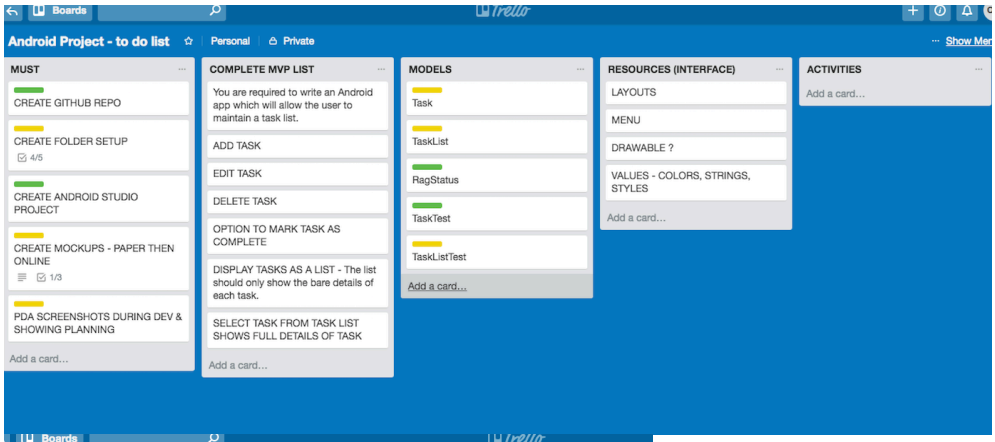


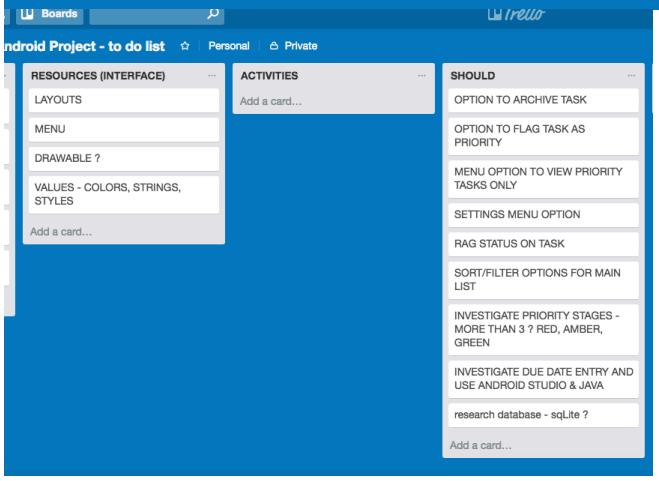

TaskListActivity.java - ToDoListApp - [~/e18_homework/week_08/ToDoListApp]
Android Emulator - Nexus_5X_API_27_x8...
ICanDoItList
food
Holiday
chores
Project
ToDoListApp git:(master) ✗


olgamaunsell / ICanDoItList-Java-Android-App
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Pulse
Contributors
Community
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
Jan 21, 2018 – Feb 10, 2018
Contributions: Commits
Contributions to master, excluding merge commits
20
15
10
5
0
Jan 21
Jan 28
Feb 04
olgamaunsell
#1
30 commits 2,756 ++ 708 --
20
1/21/2018
1/25/2018
1/29/2018
2/1/2018
2/5/2018

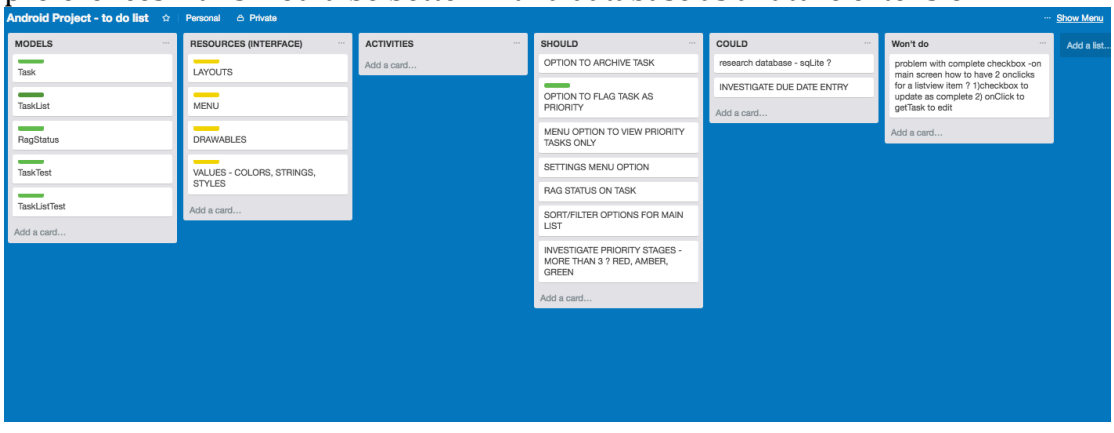
P12 – Screenshots of planning and the different stages of development to show changes

Day 2 of individual Java Android Studio project (27th Jan)

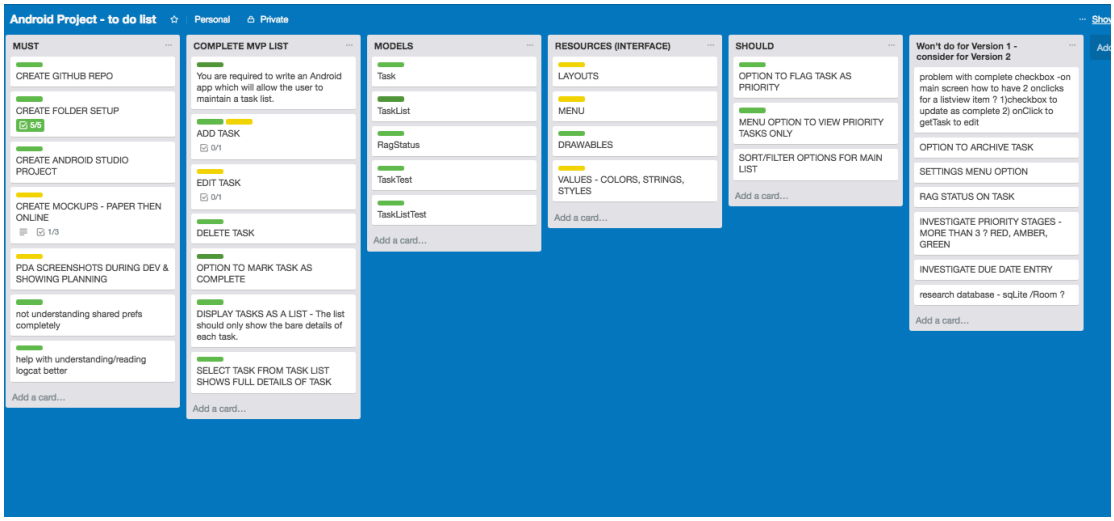




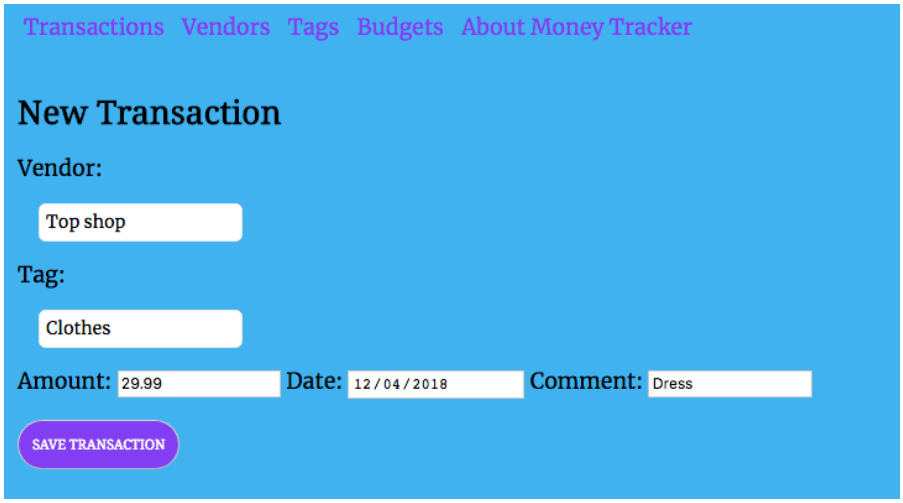
During the project week I decided after research not to include “completed task” checkbox on Main List screen – too complicated/difficult to do with Shared preferences – this would be better with a database as a future extension.



P12 (Continued) - Wed 31st Jan 2018 - Updated plan on last Day



P13 – User input being processed to design requirements.

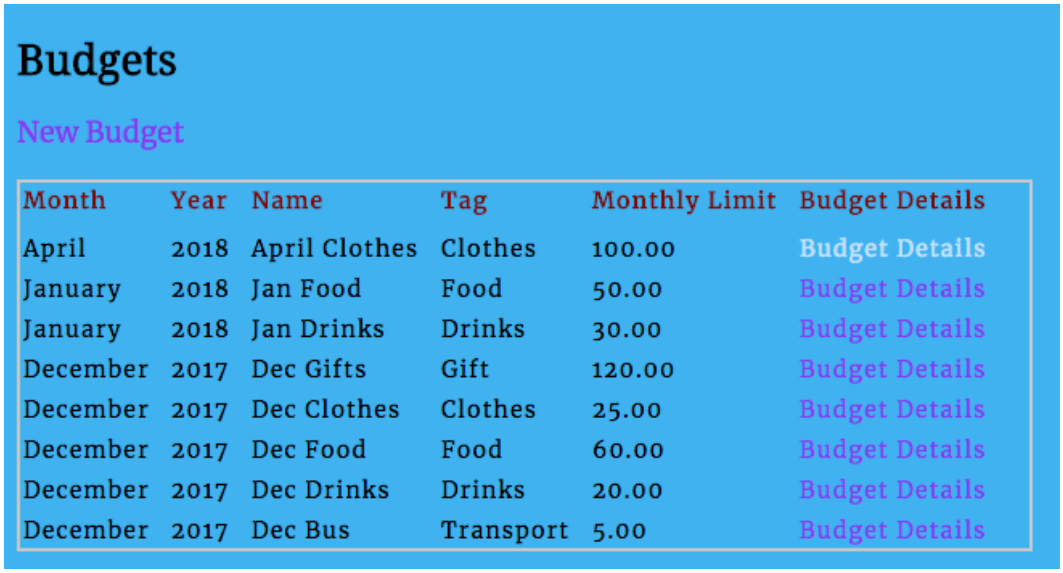


User input being saved – shows transaction has been saved and added to list of transactions – most recent "Dress" transaction at the top of the list.

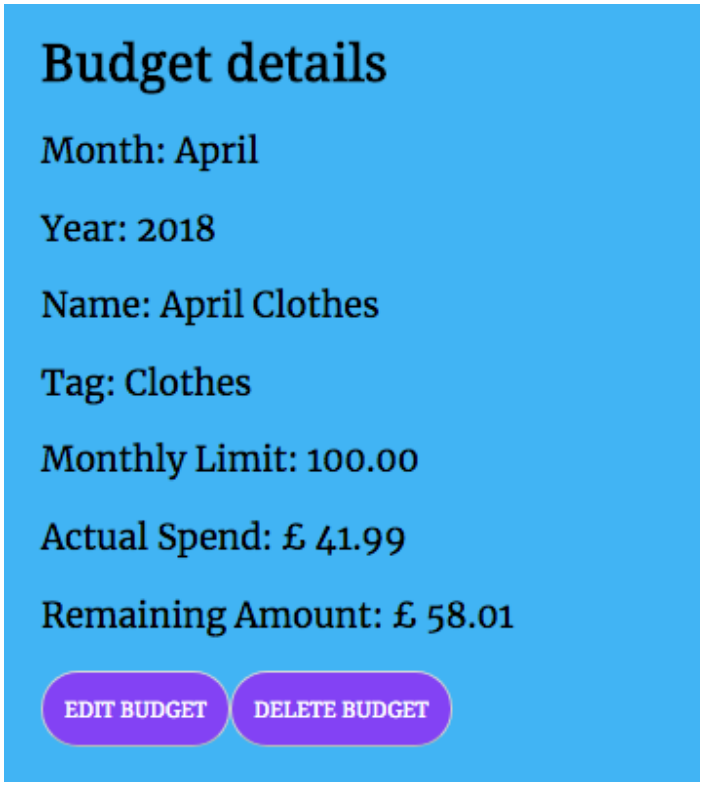
P14 – Interaction with data persistence – data input in P13 saved to database

```
money_tracker=# select * from transactions
money_tracker-# ;
 id | vendor_id | tag_id | amount | transaction_date |    comment
----+-----------+--------+--------+------------------+---------------
  1 |         1 |      1 |  11.33 | 2017-12-12       |
  2 |         2 |      2 |    6.9 | 2017-12-10       | G & T
  3 |         3 |      4 |  24.99 | 2017-12-15       | Picture
  4 |         5 |      3 |  34.49 | 2017-12-11       | Dress
  5 |         2 |      2 |    3.1 | 2017-12-15       | Beer
  6 |         4 |      5 |    1.6 | 2017-12-14       | Bus
  7 |         4 |      5 |    1.6 | 2017-12-13       | Bus
  8 |         1 |      1 |   50.0 | 2017-12-21       | Weekly shop
  9 |         3 |      4 | 132.42 | 2017-12-15       | Xmas Pressies
 10 |         2 |      2 |    3.0 | 2017-11-29       | Beer
 12 |         3 |      3 |  100.0 | 2018-03-16       | shoes
 13 |         3 |      3 |   12.0 | 2018-04-12       | purse
 14 |         5 |      3 |  29.99 | 2018-04-12       | Dress
(13 rows)
```

P15 - User wishes to view budget details for April 2018 and clicks the "Budget details" link for that month



Result – user request being processed correctly – budget details for April 2018 displayed



P16 – Show an API being used within a program

app.js

```
const app = function(){
  const url = "https://restcountries.eu/rest/v2/all"
  make_request(url);
}

const make_request = function(url){
  const request = new XMLHttpRequest();
  request.open("GET", url);
  request.addEventListener('load', convertJSONToCountries);
  request.send();

}

const convertJSONToCountries = function(){
  if(this.status !== 200) return;

  const countries = JSON.parse(this.responseText);
  populateLanguages(countries);
};
```

Part 1 of method populateLanguages

app.js

```
const populateLanguages = function(countries){

  const languagesList = {};
  countries.forEach(function(country, index){
    const languages = country.languages;
    const countryLanguagesName = languages.map(function (language) {
      return language.name;
    });

    const countryIndex = index;

    countryLanguagesName.forEach(function(language){

      const countryLanguages = languages;

      if(language in languagesList){
        languagesList[language].push(countryIndex)
      } else {
        languagesList[language] = [countryIndex]
      }

    });

  }.bind(this));

  // Create word cloud
  // create data array to pass to word cloud

  let languagesDataArray = [];
  const keys = Object.keys(languagesList);

  for (key of keys) {

    const language = key;
    const countryCount = languagesList[key].length
    languageHash = {
      name: language,
      weight: Math.log(countryCount)
    }
```

P16 (continued) - Part 2 of method populateLanguages

app.js

```
// Create word cloud
// create data array to pass to word cloud

let languagesDataArray = [];
const keys = Object.keys(languagesList);

for (key of keys) {

  const language = key;
  const countryCount = languagesList[key].length
  languageHash = {
    name: language,
    weight: Math.log(countryCount)
  }

  languagesDataArray.push(languageHash);
}

console.log("languagesDataArray", languagesDataArray);

const wordCloudContainer    = document.querySelector("#word-cloud");

const wordCloudDetails    = new WordCloudDetails(languagesDataArray, wordCloudContainer);

new WordCloud(wordCloudDetails.wordCloud);

populateLanguagesDropdown(keys, languagesList, countries);
};
```

The API being used by the program whilst running

Choose a language to see a list of countries that speak that language

Danish | Denmark

- Country name: Denmark
- Population: 5717014
- Capital City: Copenhagen

P17 – Travelingo Project - Bug Tracking Report

Issue	Status	Resolution	Status
User can enter a phrase to be translated without selecting country/language first	Failed	Change order of user entry so that the country/language dropdown is selected first before user can enter a phrase	Pass
User can leave phrase blank and select submit to translate	Failed	Update validation to ensure user has entered a phrase. If user clicks submit without phrase entry then a message will appear prompting user to enter a phrase to translate.	Pass
Speak button is provided for languages where there is no voice synthesis	Failed	Update code validation and screen display to ensure that the speak button is not available for languages which there is no voice output e.g. Sri Lanka	Pass
User selects country with English as main language. This displays and translates from English to English which is not required.	Failed	Update code validation and when country language is English do not provide translation, only display Country weather, flag and where that language is spoken on world map	Pass
When phrase to be deleted contains a question mark, the phrase isn't deleted	Failed	The ? is a special character so the code requires to be updated to use encodeURIComponent so that the phrase can be found in the database and deleted	Pass

P18 - Testing

```java
public class ConferenceRoomTest {

    ConferenceRoom conferenceRoom;
    Guest guest1;

    @Before

    public void before() { conferenceRoom = new ConferenceRoom( capacity: 15,  chargeable: true,  name: "Sun Room",  rate: 90.0); }

    @Test

    public void canGetName(){
        assertEquals( expected: "Sun Room", conferenceRoom.getName());

    }

    @Test

    public void canChangeName(){
        conferenceRoom.setName("Sunny Room");
        assertEquals( expected: "Sonny Room", conferenceRoom.getName());
    }

    @Test

    public void canGetRate(){
        assertEquals( expected: 90.0, conferenceRoom.getRate(),  delta: 0.01);
    }

    @Test

    public void canCheckInGuestToRoom() {
        conferenceRoom.checkInGuest(guest1);
        assertEquals( expected: 1, conferenceRoom.getGuestCount());
    }
}
```

```
ConferenceRoomTest
4 tests done: 1 failed - 47ms
ConferenceRoomTest 47ms
    canGetName 2ms
    canGetRate 1ms
    canChangeName 44ms
    canCheckInGuestToRoom 0ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/bin/java ...

org.junit.ComparisonFailure:
Expected :Sonny Room
Actual   :Sunny Room
 <Click to see difference>

 <2 internal calls>
    at ConferenceRoomTest.canChangeName(ConferenceRoomTest.java:28) <23 internal calls>

Process finished with exit code 255
```

```java
    ConferenceRoom conferenceRoom;
    Guest guest1;

    @Before

    public void before() { conferenceRoom = new ConferenceRoom( capacity: 15,  chargeable: true,  name: "Sun Room",  rate: 90.0); }

    @Test

    public void canGetName(){
        assertEquals( expected: "Sun Room", conferenceRoom.getName());

    }

    @Test

    public void canChangeName(){
        conferenceRoom.setName("Sunny Room");
        assertEquals( expected: "Sunny Room", conferenceRoom.getName());
    }

    @Test

    public void canGetRate(){
        assertEquals( expected: 90.0, conferenceRoom.getRate(),  delta: 0.01);
    }


    @Test

    public void canCheckInGuestToRoom() {
        conferenceRoom.checkInGuest(guest1);
        assertEquals( expected: 1, conferenceRoom.getGuestCount());
    }
}
```

```
ConferenceRoomTest
All 4 tests passed - 3ms
ConferenceRoomTest 3ms
    canGetName 2ms
    canGetRate 0ms
    canChangeName 1ms
    canCheckInGuestToRoom 0ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```