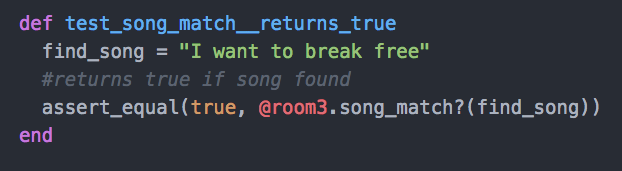
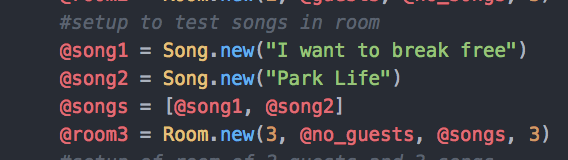
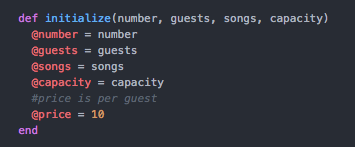
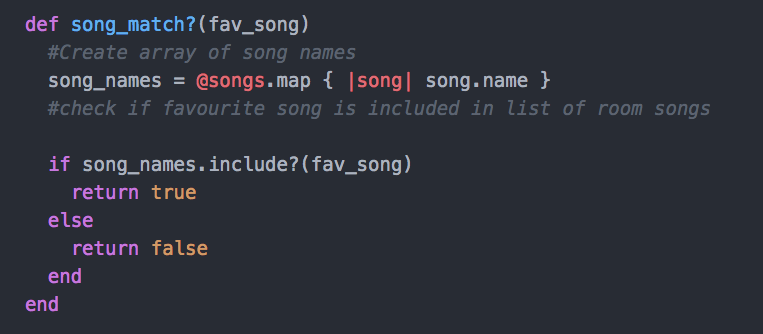
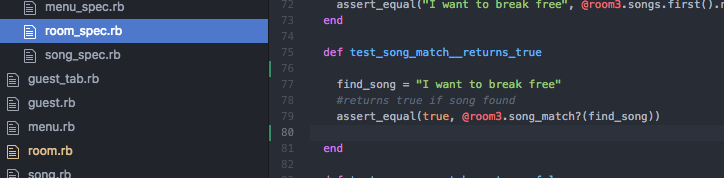
**Week 2 – I.T 5 – Use of an array in a program**

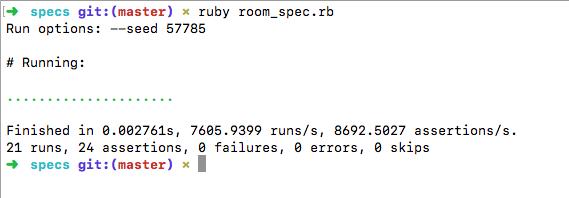






Result of function running

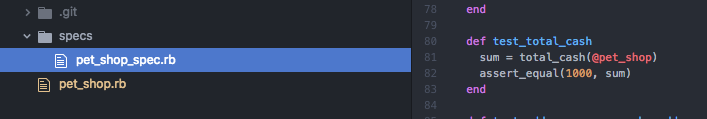


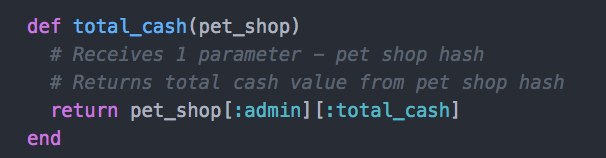


Week 2 – I.T 6 – Use of a hash in a program

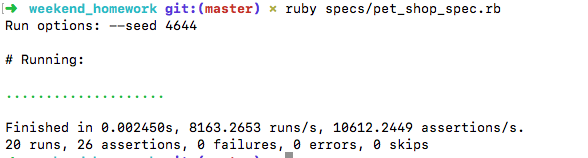
“admin” hash contained within “pet\_shop” hash



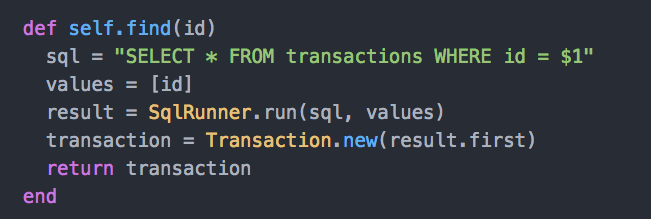




Result of function running

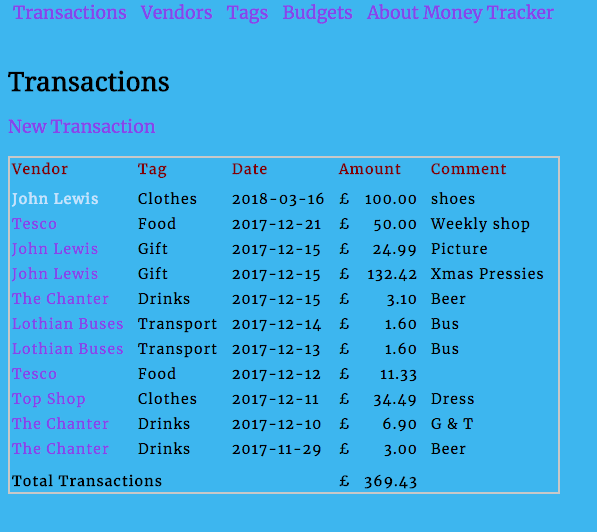


Week 3 – I.T.3 – Demonstrate searching data in a program

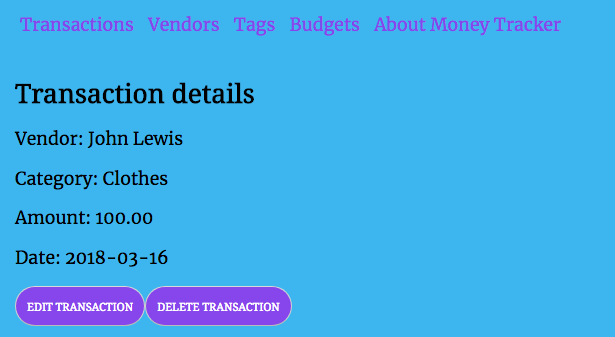


Result of search function running

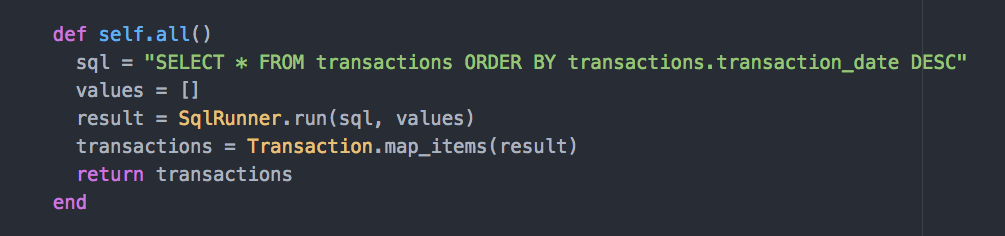
Screenshot of all transactions



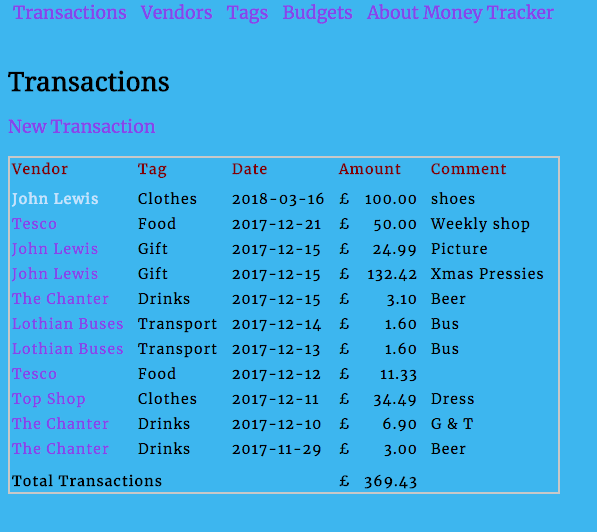
User selects to view 1 transaction and the above search function is then run to produce the result of transaction searched/ found and displayed



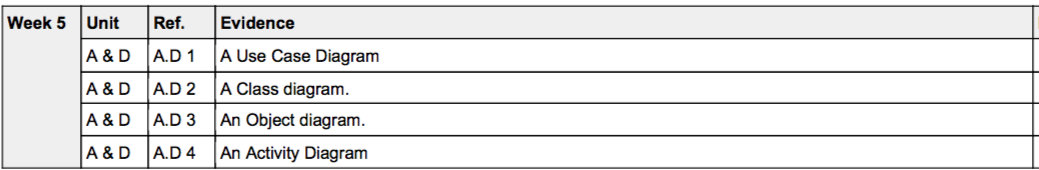
**Week 3 – I.T.4 – Demonstrate sorting data in a program**



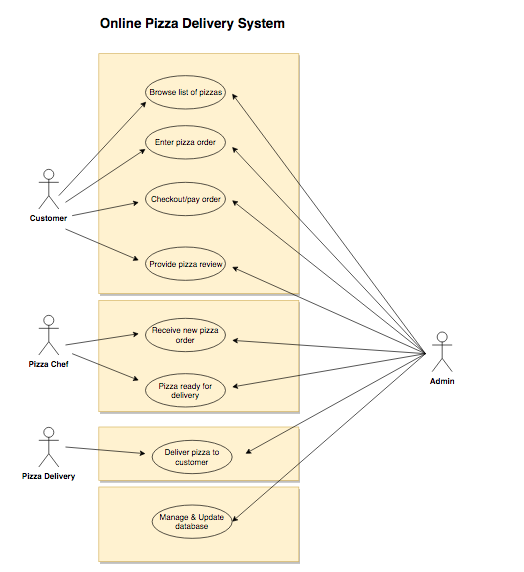
Transactions are retrieved and sorted in descending order to be displayed on the screen in descending order



Week 5



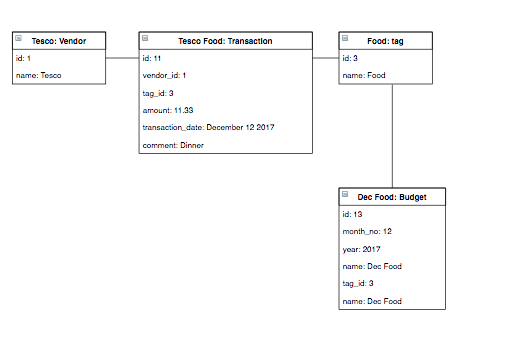
A.D 1 – A Use case diagram – Online Pizza Delivery System



A.D 2 – Class Diagram – Money Tracker App

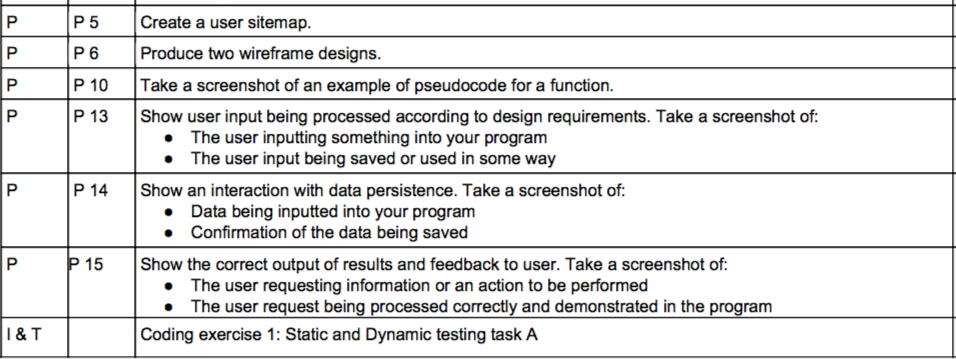


A.D 3 – Object Diagram – Money Tracker App

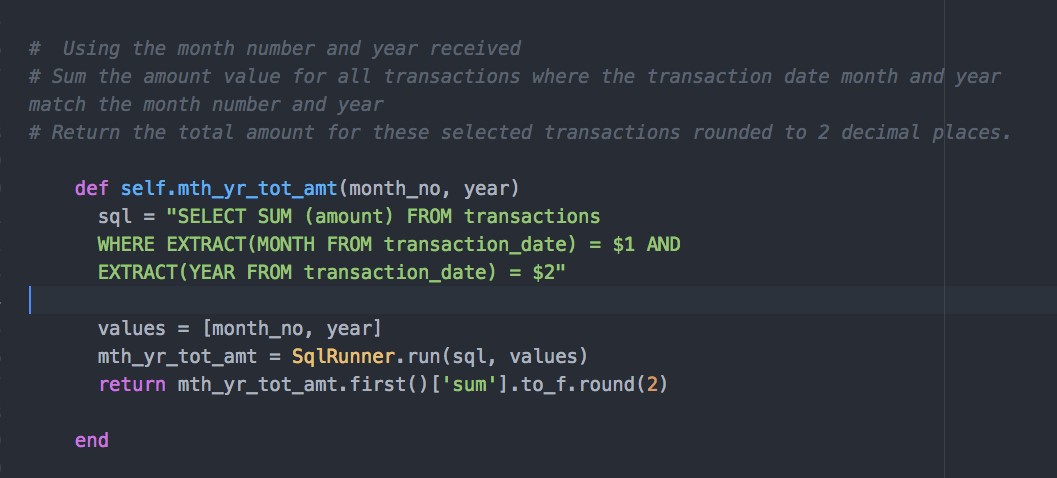


**A.D. 6 - travelingo Project - Implementation Constraints**

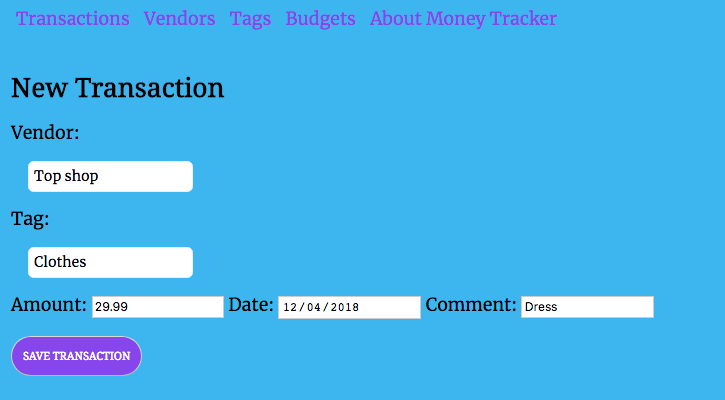
|  |  |  |
| --- | --- | --- |
| **Constraint** | **Possible effect of constraint on product** | **Solution** |
| Hardware and software platforms | Web app requires wifi to perform api requests to translate phrases, access country information and weather | Provide user documentation to clarify wifi access is required. A reduced app could be provided which displays only phrases that user has stored in the database. |
| Performance requirements | If the user stores large number of phrases for each language it could cause a delay in displaying the information to the screen. | Provide user documentation to clarify that large number pf phrases stored could cause a delay and recommend deleting phrases not required. In addition a “loading” icon can be displayed to keep user informed while the database is loading information. |
| Persistent storage and transactions | This version of the product which uses a mongo database is not designed for a large number of phrases to be stored in the database. | Provide user documentation clarifying database constraints. If a large volume database is required, a new database design should be considered for the next version. |
| Usability | Not all languages listed have a voice to speak the phrase | Disable the speak button for languages where is no voice for that language and provide user documentation with the latest language list that has a voice to speak the language. |
| Budgets | CodeClan project had no budget available. All tools/technologies needed to be existing tools already obtained by students. | Make use of all free tools/languages available. For example access to free apis. |
| Time | Project to be delivered in 1 week | Initial planning session to discuss and confirm MVP to be delivered and any extensions if time available. Daily planning sessions to confirm project is on track to be delivered to timescales. |



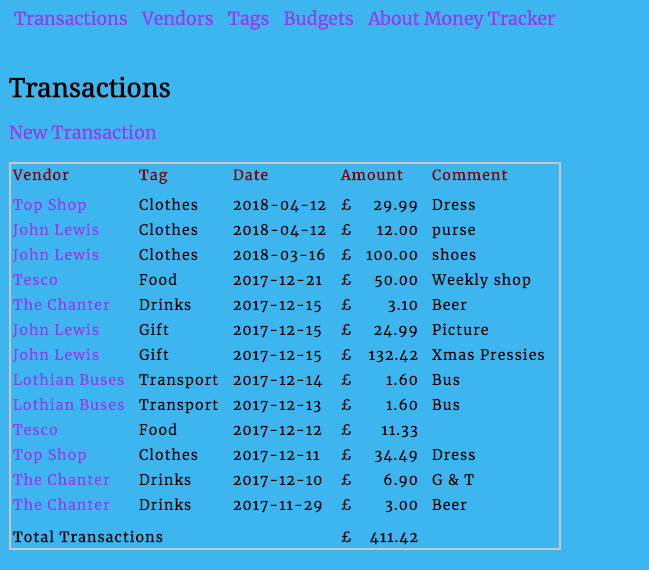
P 10 - An example of pseudocode for a function



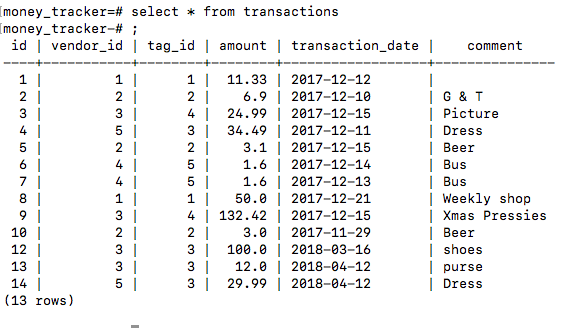
**P13 – User input being processed to design requirements.**



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.



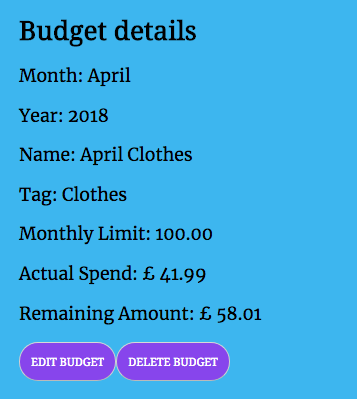
**P14 – Interaction with data persistence – data input in P13 saved to database**



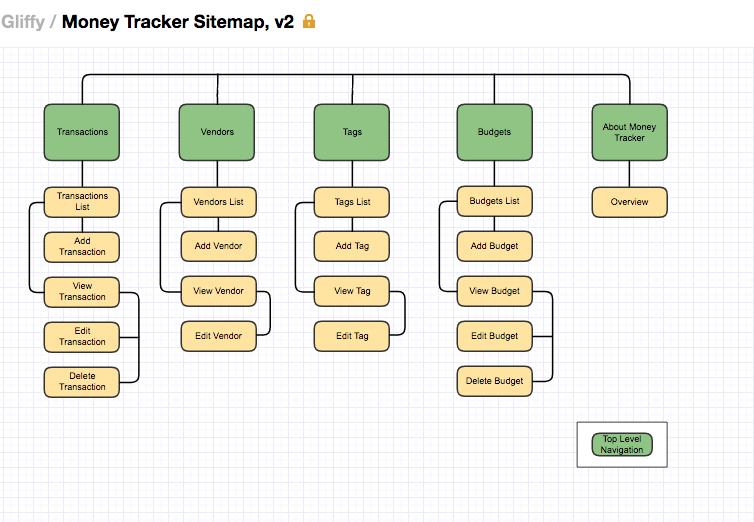
**P15 - User wishes to view budget details for April 2018 and clicks the “Budget details” link for that month**



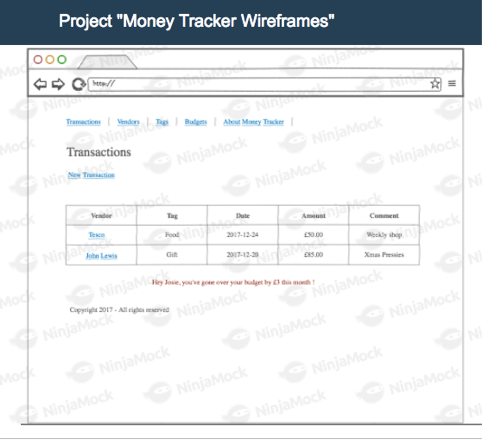
**Result** – user request being processed correctly – budget details for April 2018 displayed

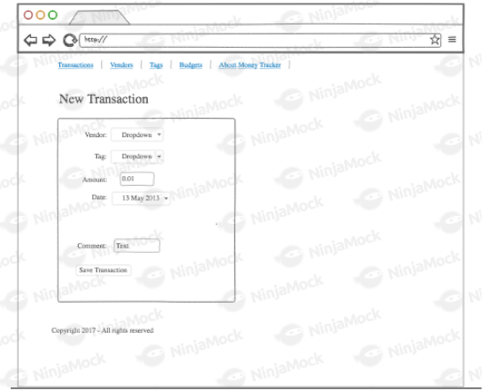


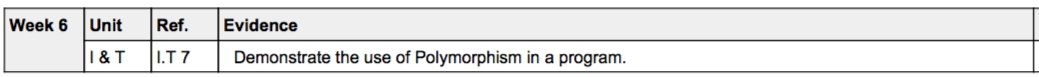
**P5 – User Site Map – Money Tracker App**



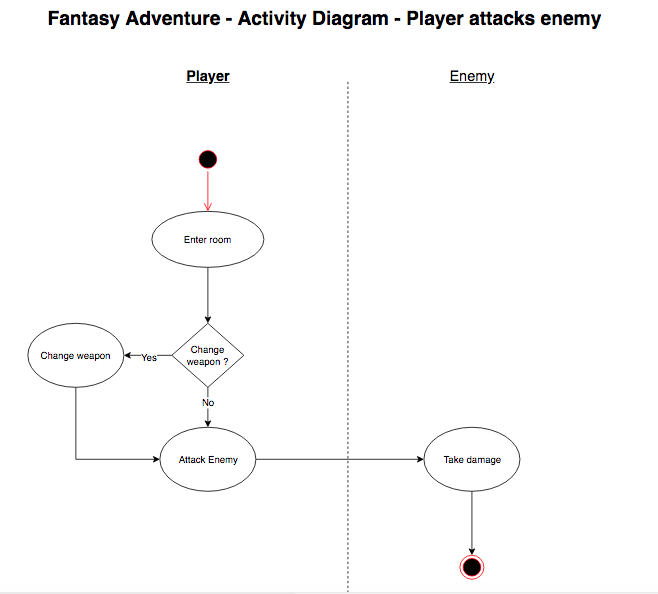
**P6 – Two Wireframe Designs – Money Tracker App**



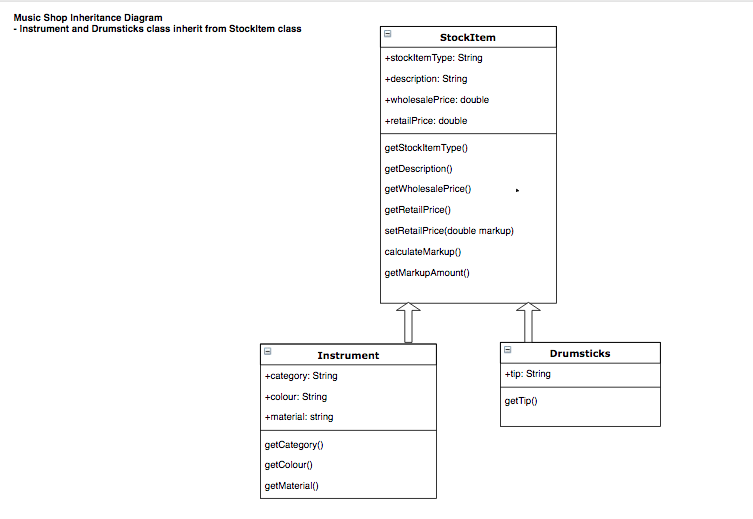




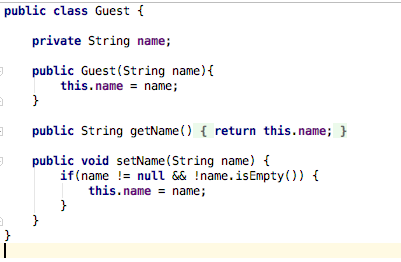
**A.D 4 – Activity Diagram**



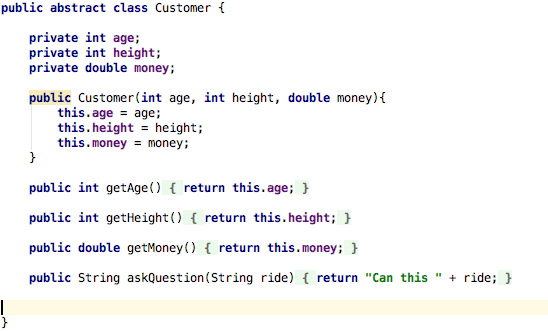
**A.D 5 - Inheritance Diagram**



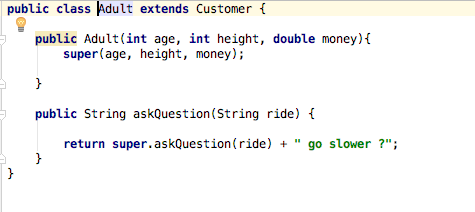
**I.T. 1 - Encapsulation in a program**



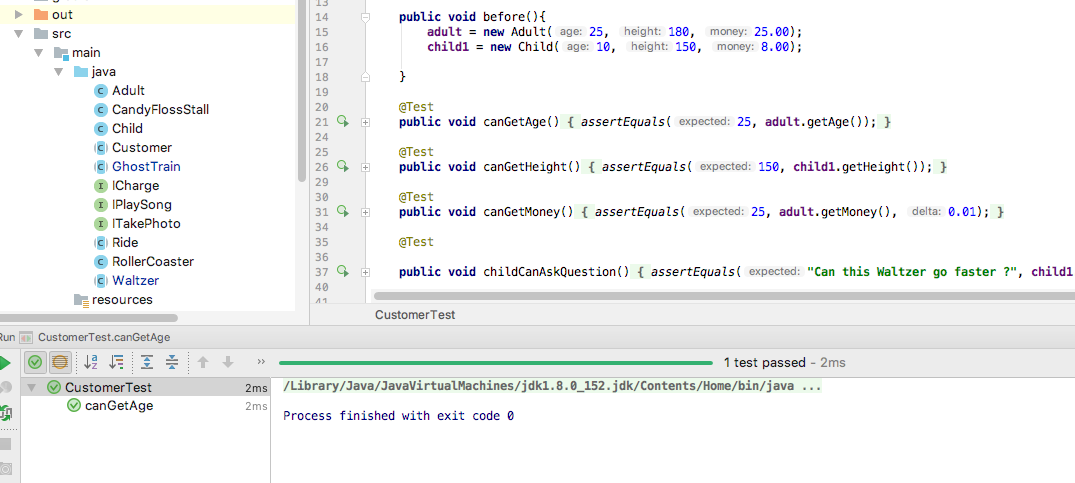
**I.T.2 – Inheritance**



Adult class inherits from Customer class

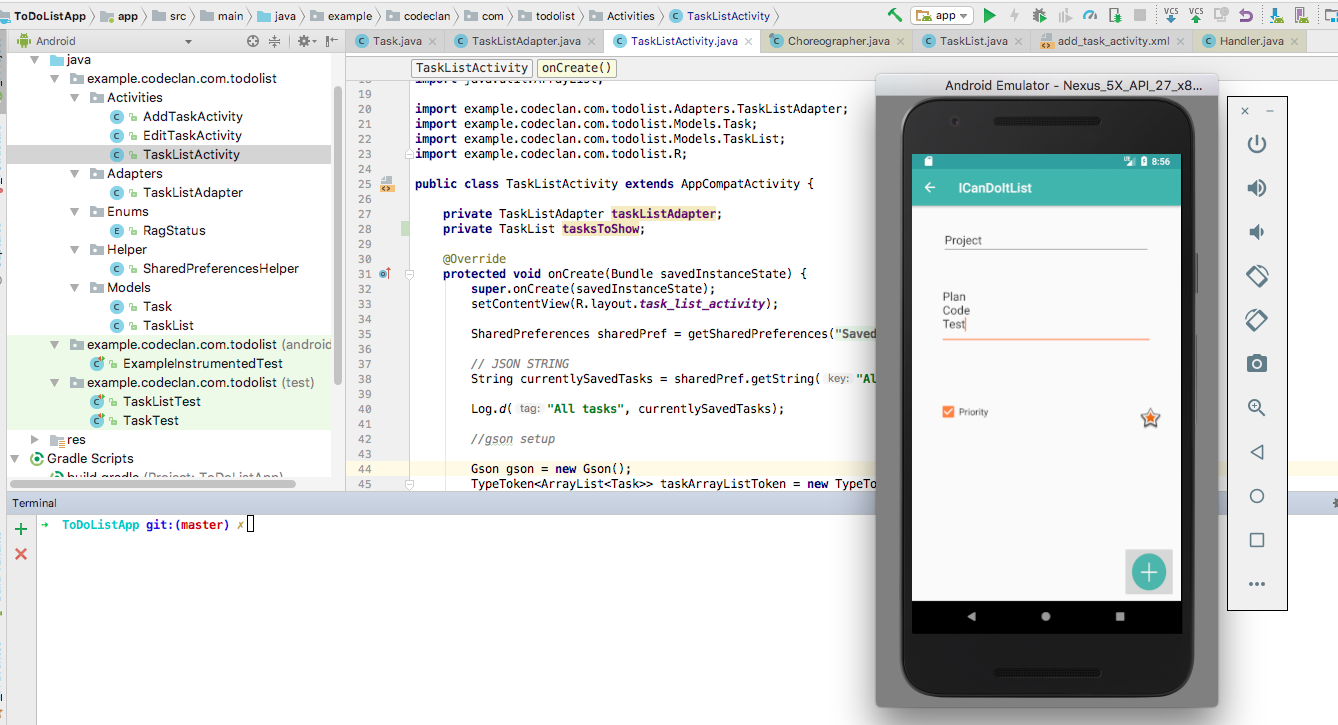


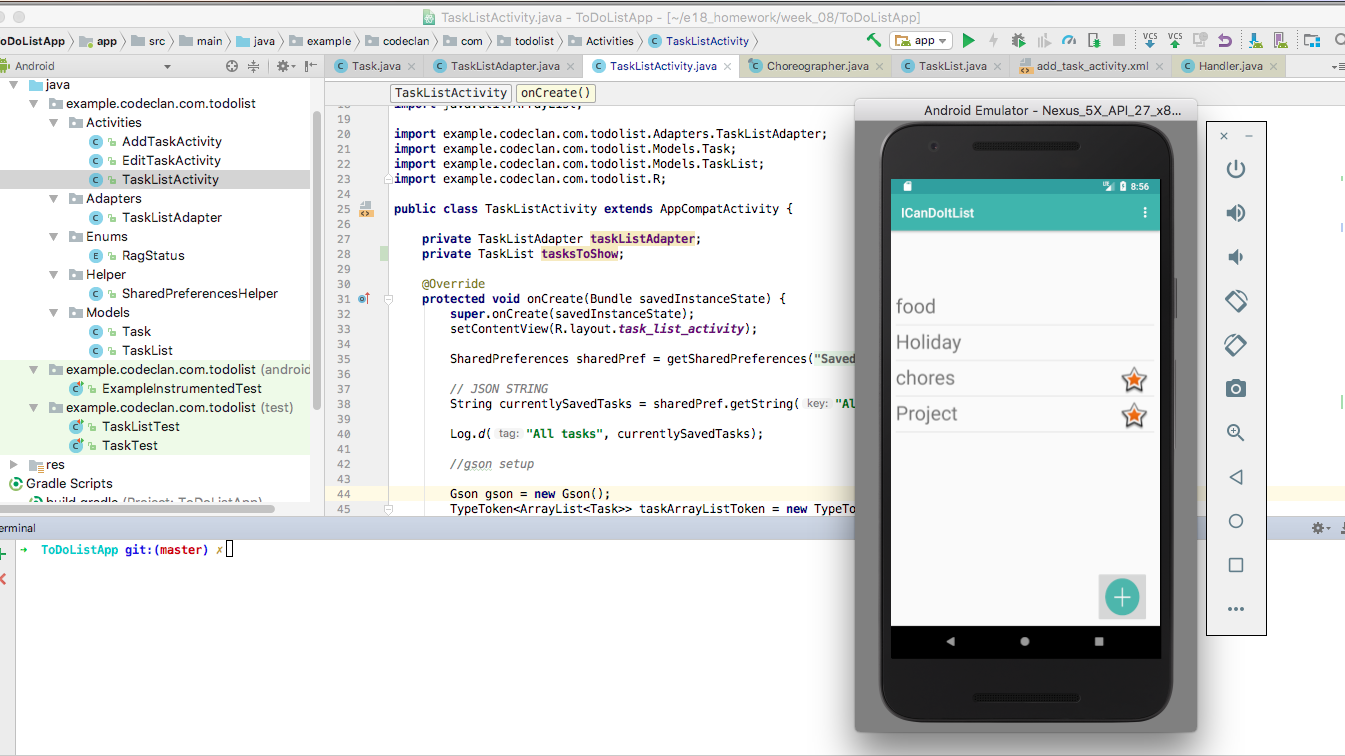
Adult object and .getAge method inherited from Customer class

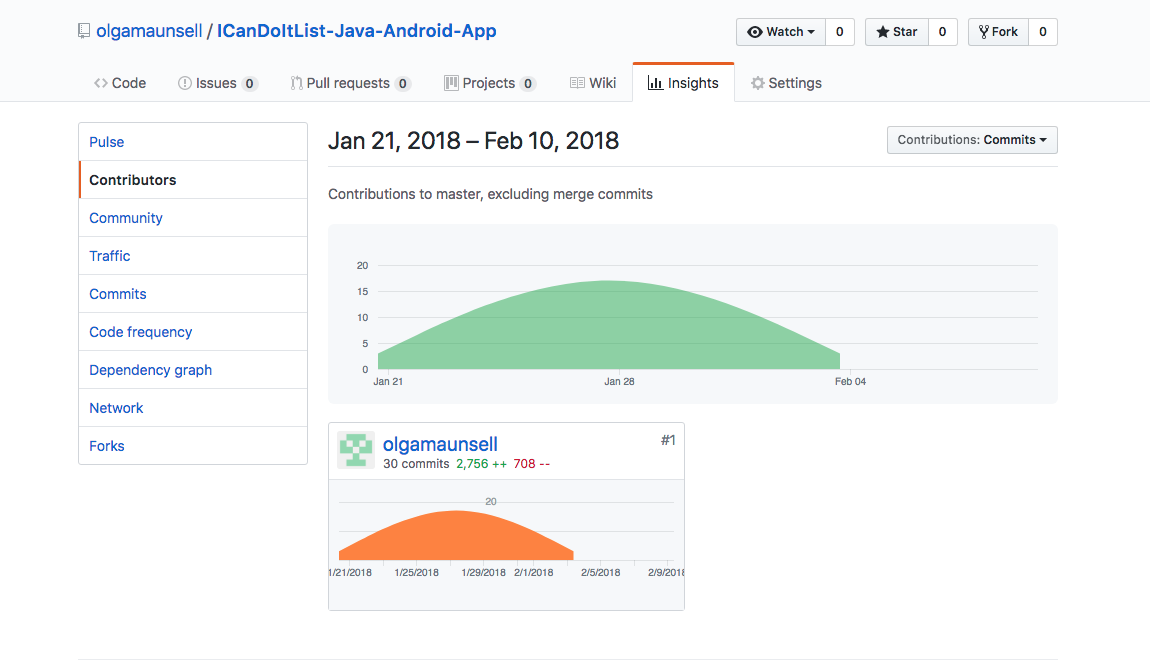


**P11** – Screenshots of my 2nd Individual project - a to do list app I called “ICanDoItList”

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

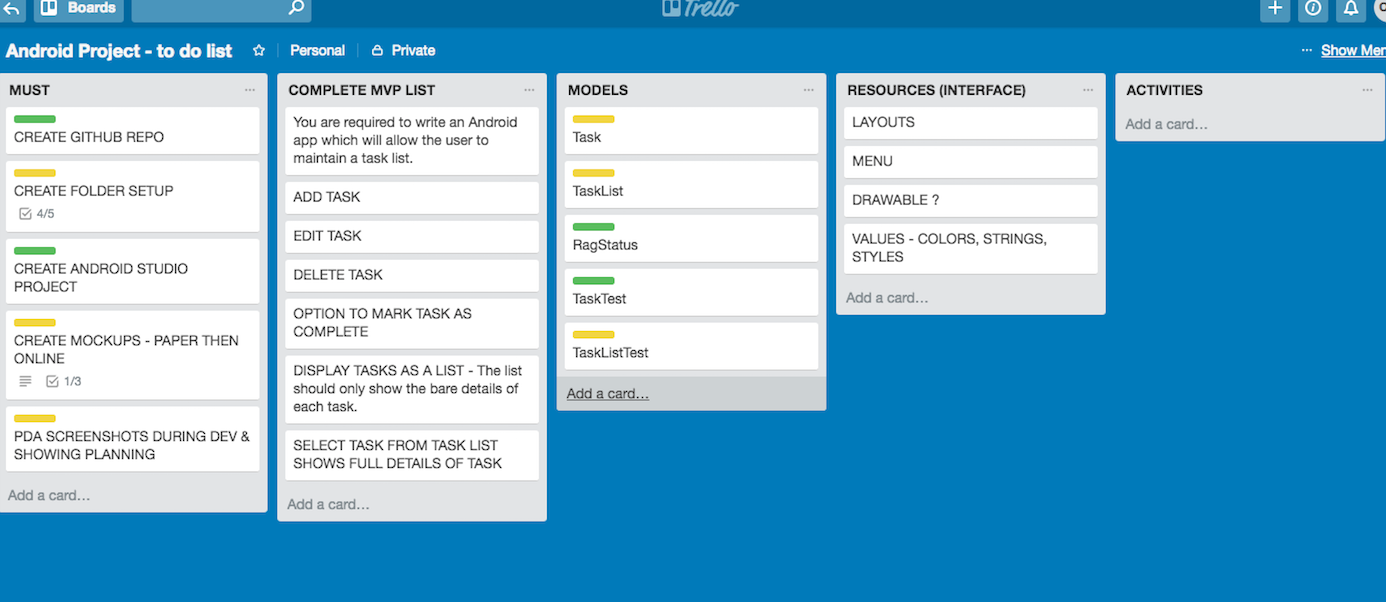


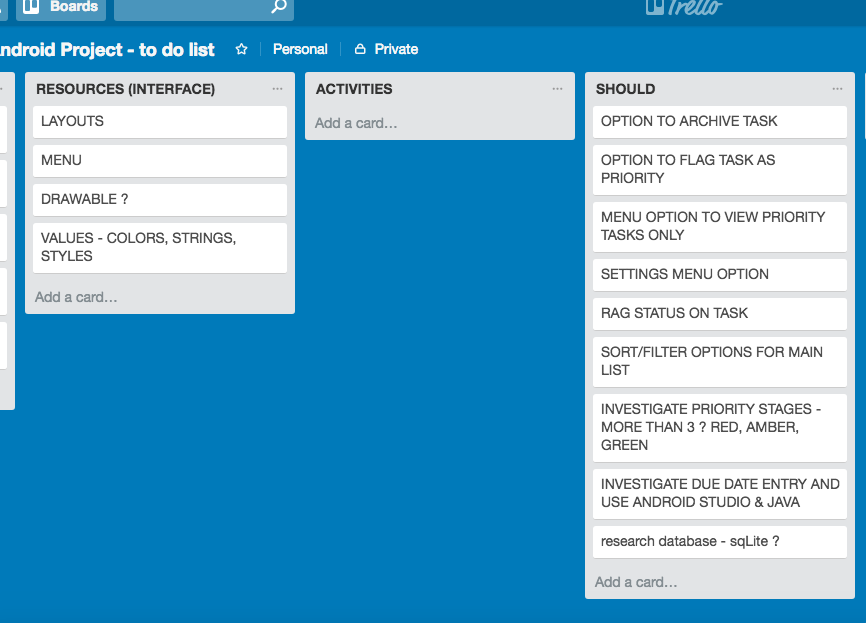




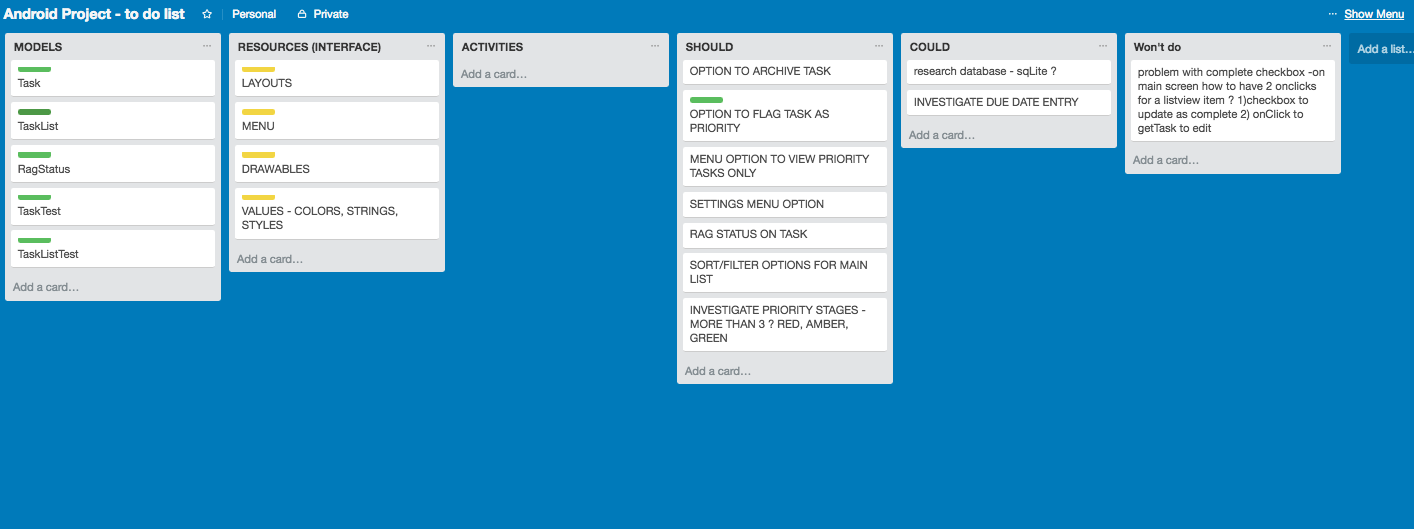
**P12 – Screenshots of planning and the different stages of development to show changes**

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

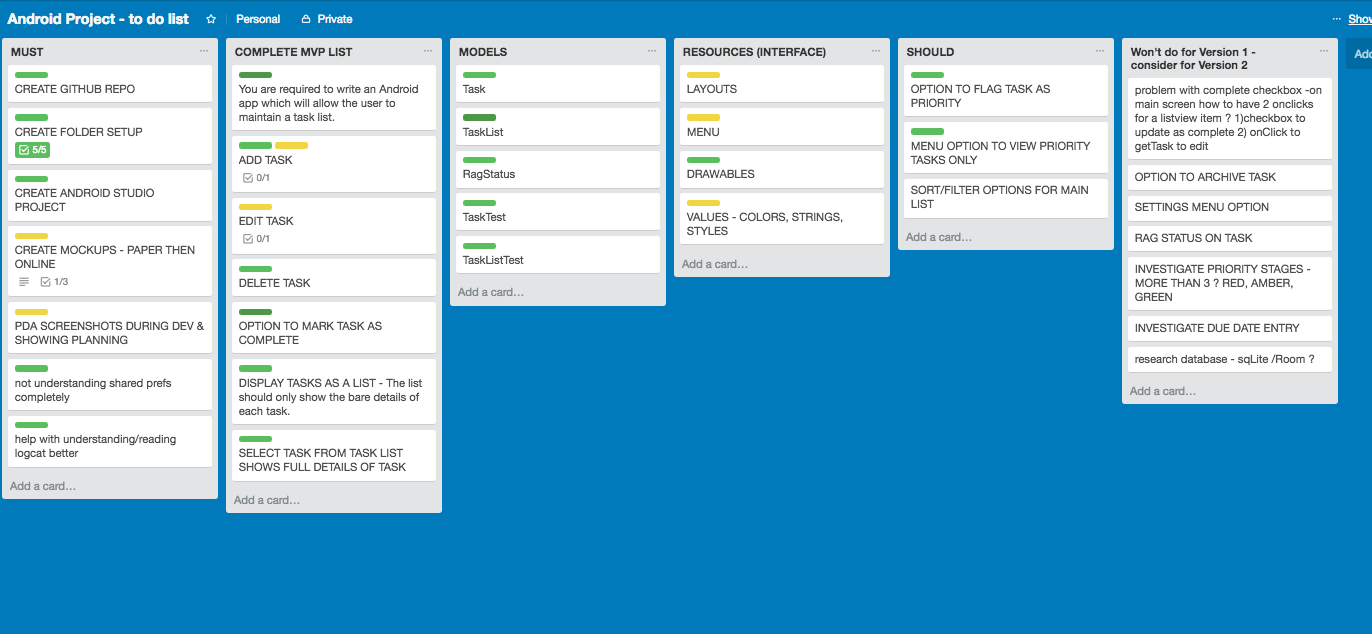




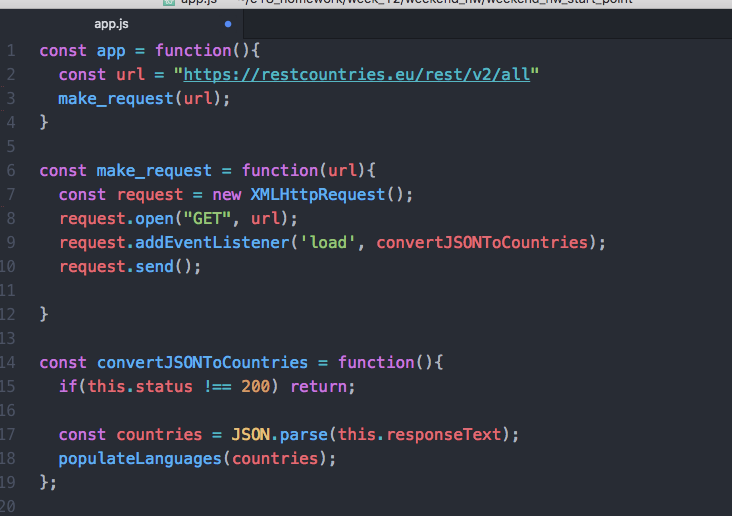
During the project week I decided after research not to include “completed task” checkbox on Main List screen – too complicated/difficult to do with Shared preferences – this would be better with a database as a future extension.



Wed 31st Jan 2018 - Updated plan on last Day



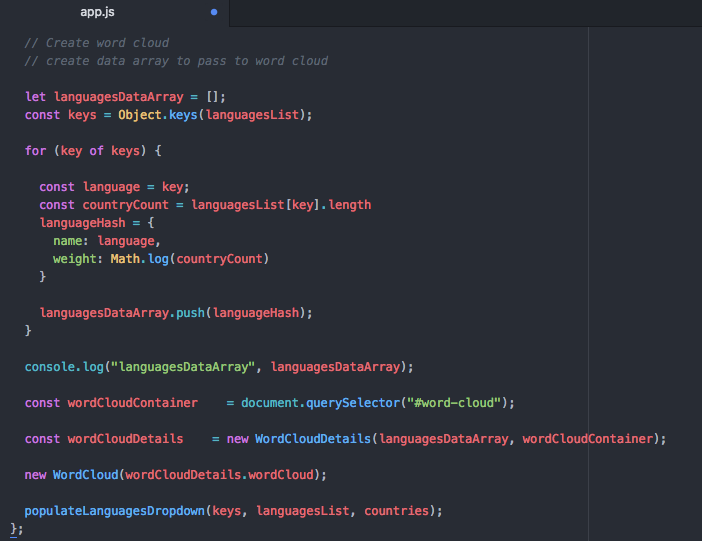
**P16 – Show an API being used within a program**



Part 1 of method populateLanguages



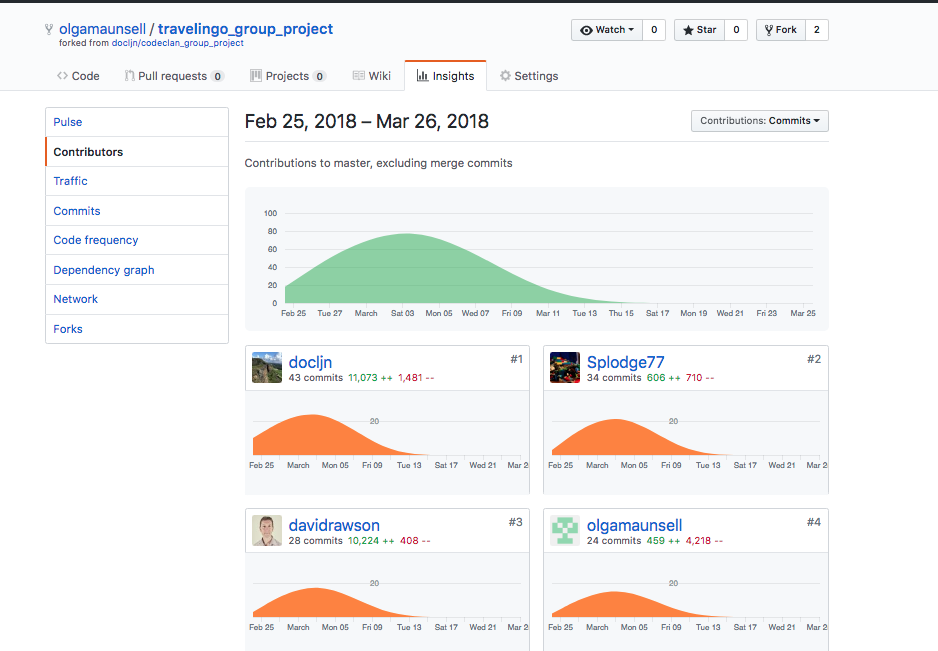
Part 2 of method populateLanguages



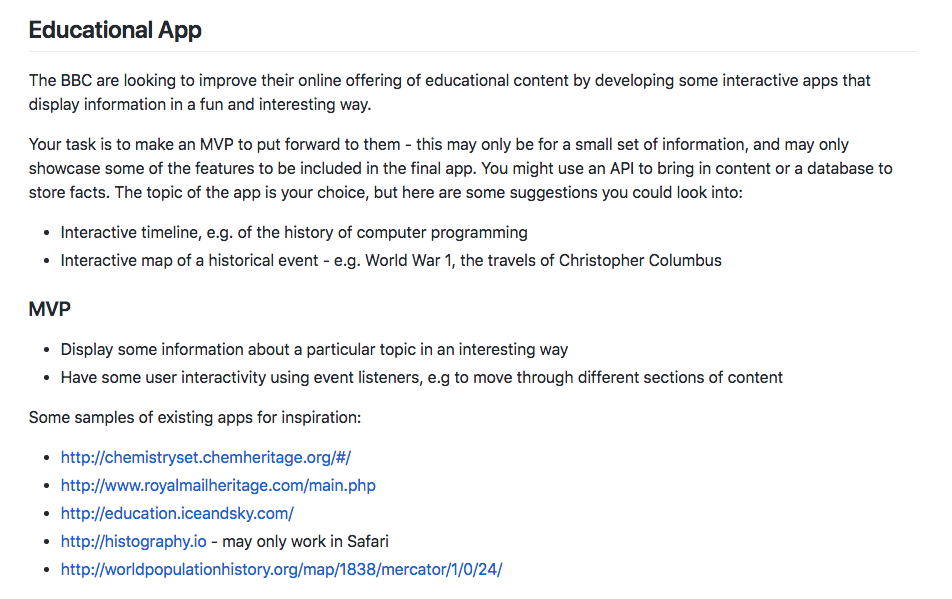
The API being used by the program whilst running



**Week 14 – P1 – Contributors page – Group project**

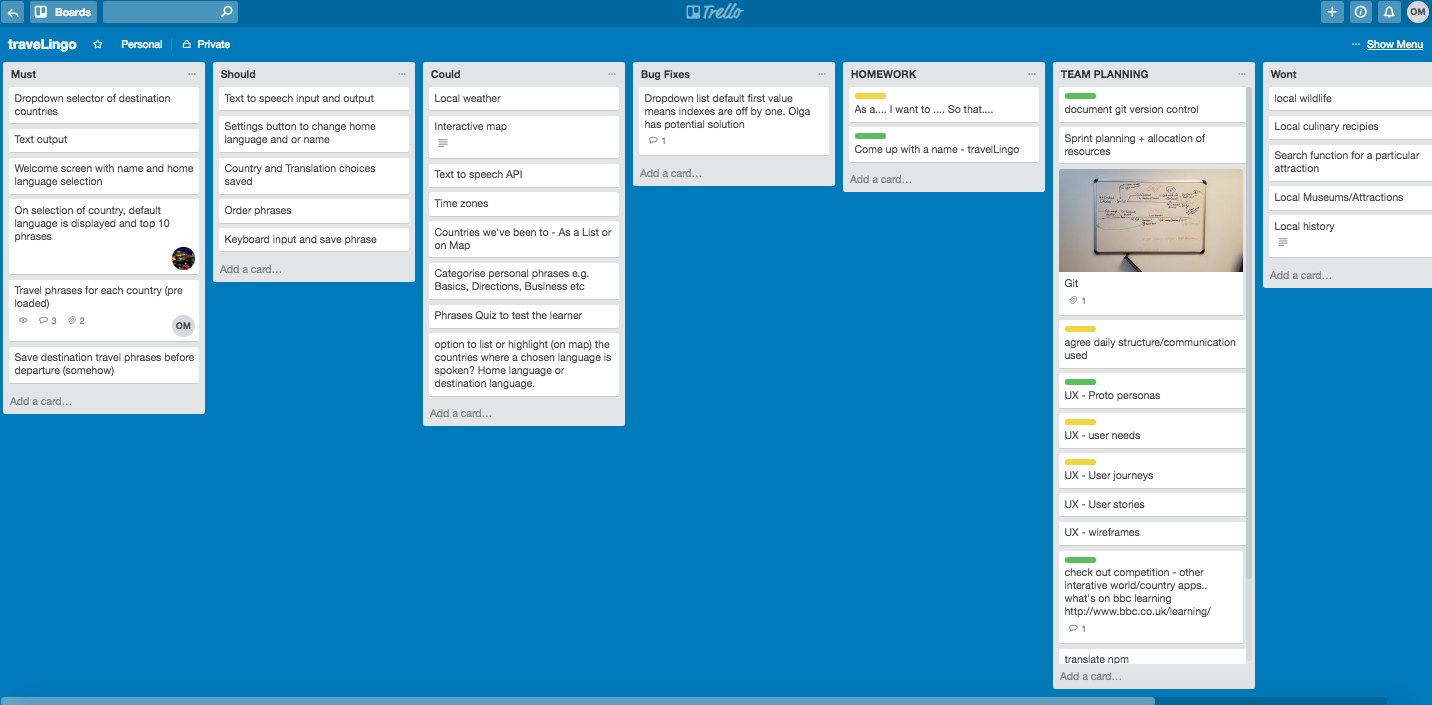


**P2 – Screenshot of project brief – Group Project**

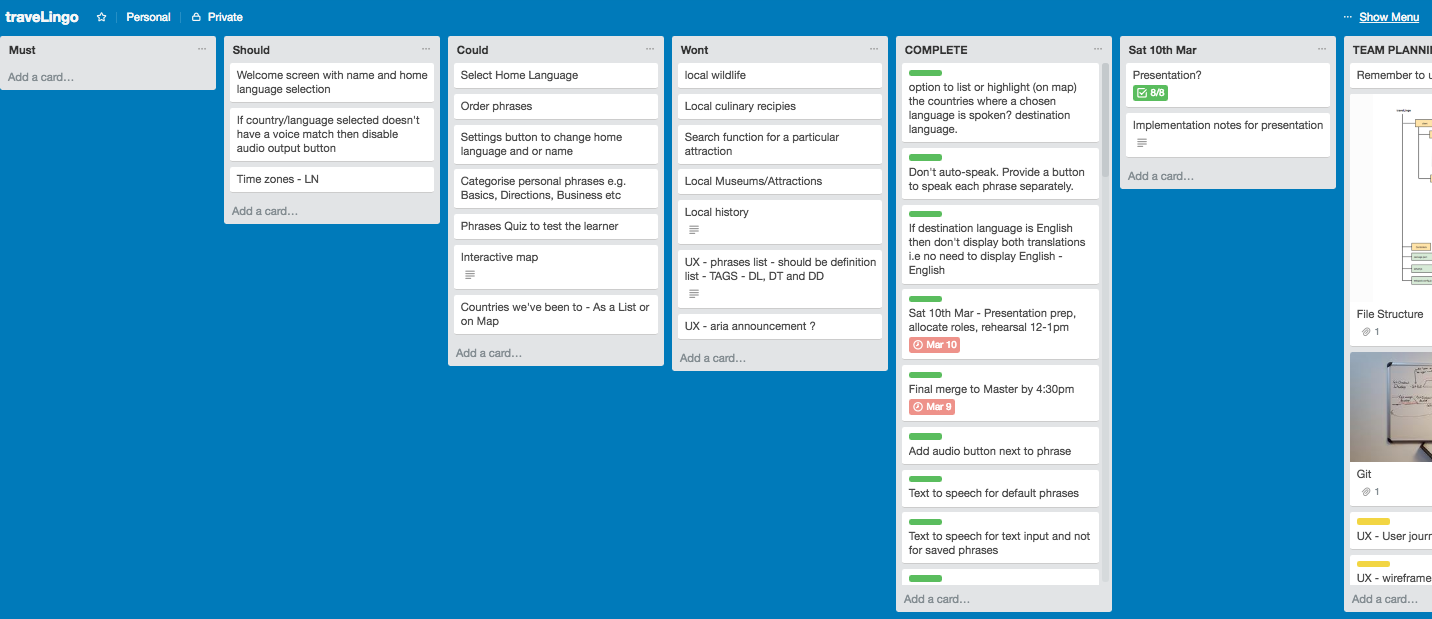


**P3 – Planning – Group Project**

Start of group project



End of group project

****

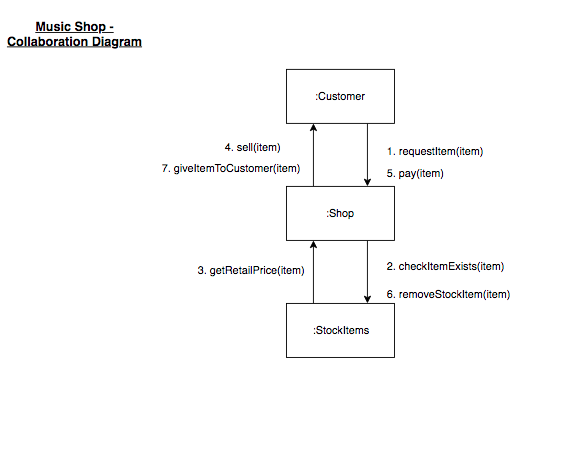
**P4 – Money Tracker App – Acceptance Criteria and Testplan**

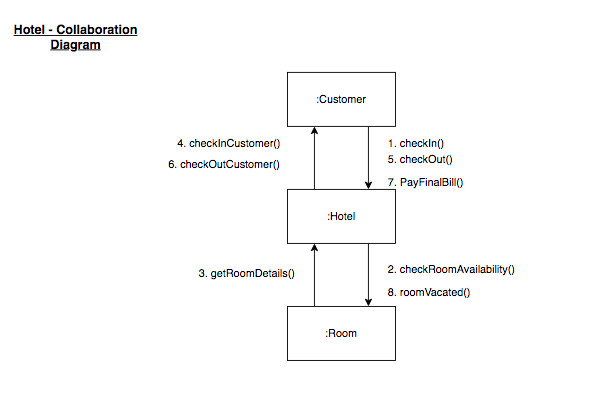
|  |  |  |
| --- | --- | --- |
| **Acceptance Criteria** | **Expected Result/Output** | **Pass/Fail** |
| An option is provided to add a new transaction | When a user selects “New Transaction” a new screen will appear allowing user to enter Vendor, Tag, Amount, Date, Comment and save the transaction | 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 |

**P17 – Travelingo Project - Bug Tracking Report**

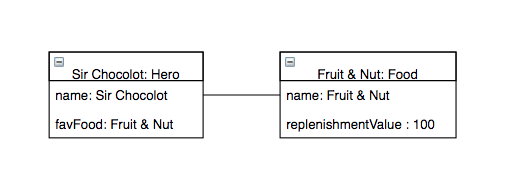
|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Status** | **Resolution** | **Status** |
| User can enter a phrase to be translated without selecting country/language first | Failed | Change order of user entry so that the country/language dropdown is selected first before user can enter a phrase | 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 |

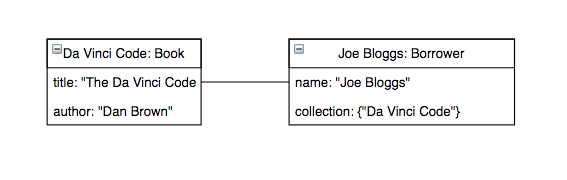
**P7 – System Interaction Diagrams**





P8 – Object Diagrams





**P9 – 2 Algorithms**

<https://github.com/olgamaunsell/week_11_day3_enumeration_hw>

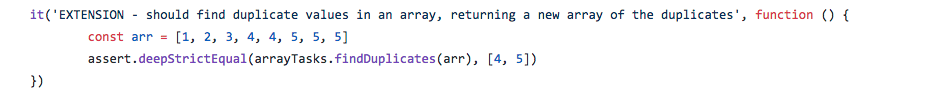
I chose this as the code is easy to follow, so good for another coder to understand. This function’s purpose is to find duplicate values in array, returning a new array of the duplicates

The function has 2 algorithms:

1. 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.
2. 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.



Test

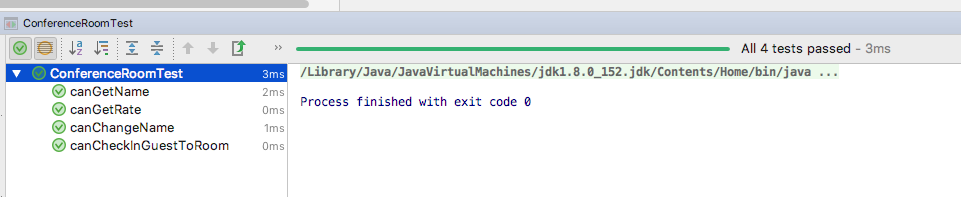


**P18 - Testing**



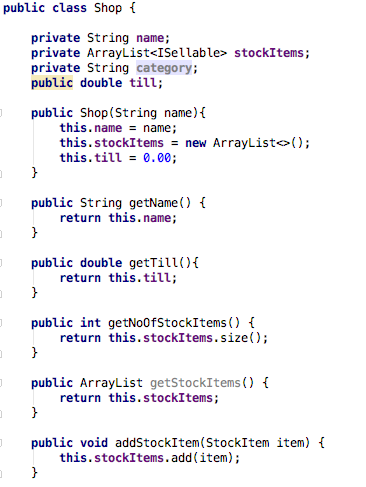






**I.T. 7 – Polymorphism**

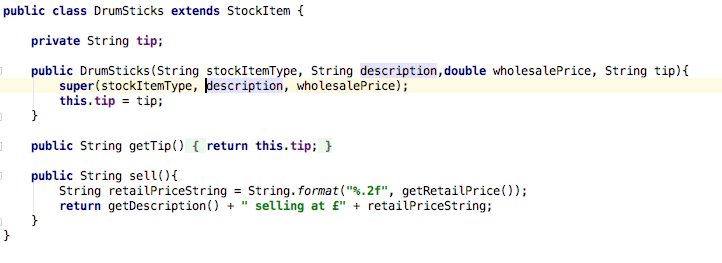
1. **Shop class has an ArrayList of ISellable stockItems**

****

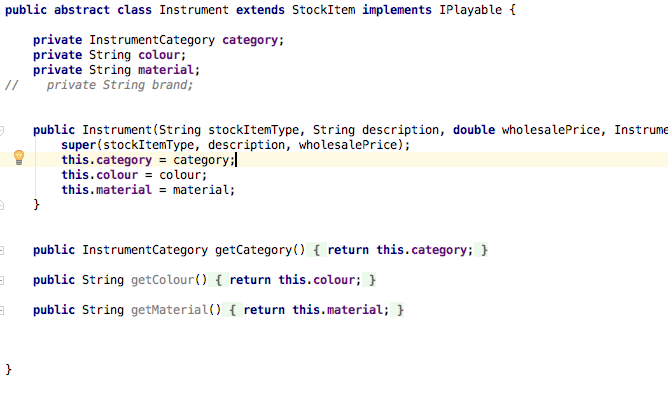
**2. StockItem class implements ISellable interface**



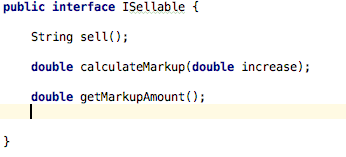
**3. DrumSticks class extends StockItem class and therefore implements ISellable interface**



**4. Instrument class extends StockItem class and therefore implements ISellable interface.**



**4. ISellable interface**



<End of Document>