James Maycock

S1219938

Computing (Information Systems Development)

*I confirm that the code contained in this file (other than that provided or authorised is all my own work and has not been submitted elsewhere in fulfilment of this or any other award*.

*James Maycock*

# Code Explanation

## MainActivity

The main activity is the first thing the user sees. It displays a news feed to the user and provides the user with a menu to navigate the application. In order to create the news feed several classes were used. The first was a "newsItem" class. This class simply contained getters and setters for the variables which correspond to the data I wished to retrieve from the RSS feed and a "toString" method in order to display the objects in the list view.

A class called "feedClass" was also used. This was used to read in the data from the RSS feed. This class had one function of type String called "getNewsFeed" which took a URL as a parameter. This method then opens a connection and reads in the data from the URL supplied. This information is then stored in a string and returned by the function.

In order to get the appropriate data from the feed the data was stored in a string and this was then parsed by the "parserClass". This class also had one function of type LinkedList called “parseData”. Using the xml pullparser the data was able to be parsed by identifying the appropriate tags I wished to pull data from. In order to store this data a LinkedList of “newsItem” objects was created. Each time a tag was found a temporary variable took it that data and was passed as a parameter for the objects set method. This would keep being done until the end of document was found. Once the end of document was found the LinkedList containing all the news objects now filled with appropriate data would be returned. This list would then be used in conjunction with an array adapter and then the list view would be populated and displayed using the “populateListView” method.

The other thing in this activity was the menu. This was created by creating a menu in xml specifying which items would be in it. There was then a switch statement which called an intent and started up a new activity depending on which option was picked. The only exceptions to this were the “Quit” option which simply closed the program and the “About” option which showed a dialogue.

## Google Activity

A class called “mapClass” was created for the use of google maps. In it two “latLng” variables were created each which specific coordinates that pointed to different Disney resorts. The “BitmapDescriptor” class was also used in order to create a custom marker which was a picture from the drawable folder. A google map object was created in order to allow manipulation of the map. The main functionality comes within the “onClick” method. Depending on which button is clicked the map will go to a specific destination. The zoom was also hardcoded in order to provide the appropriate zoom to save the user from having to zoom in themselves. There are also two buttons that allow the user to change the camera to satellite view. This is done by calling the “setMapType” and choosing satellite view.

## Preference Activity

In order to load and save user preferences a “sharedPreferences” object was created. In the method that is used to save the preferences the “sharedPreferences” editor was used which has a method called “putString”. This method takes two parameters, a key in the form of a string and a value. The values are taken from the text fields that the user enters and storing those values in strings. These are then passed into the save preferences method and this acts as the value for the key value pair.

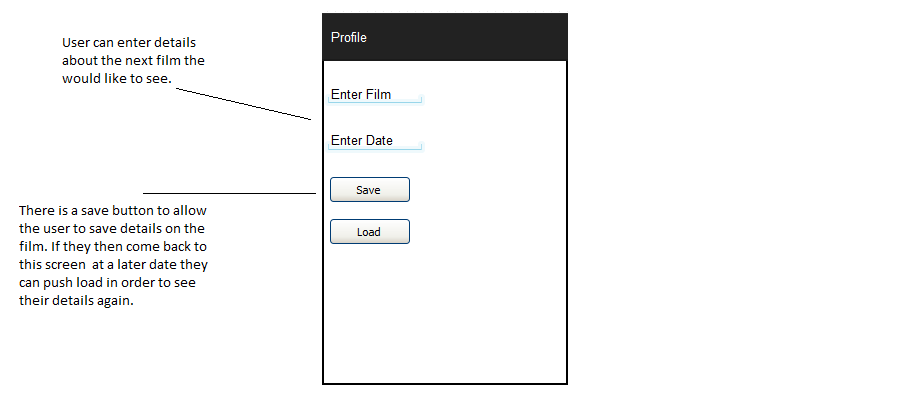
To load these preferences the method “getString” is called this takes the key value as a parameter and also a default value. Once again strings are created in order to store the values taken in from the “getString” method and then used to set the textfields.

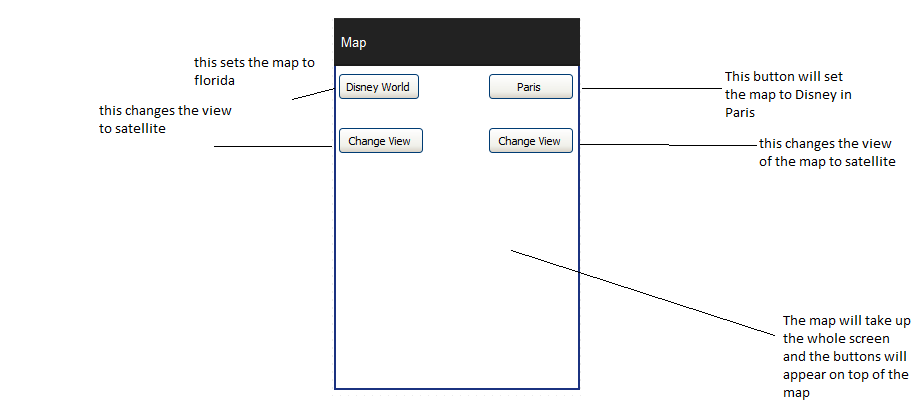
## Character Activity

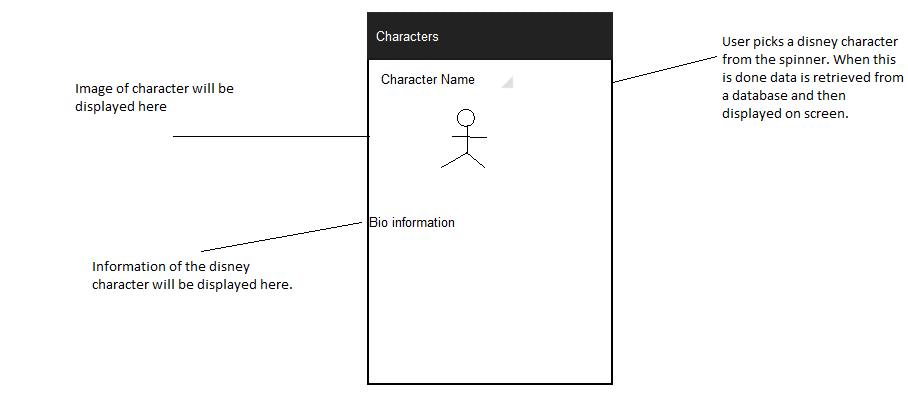
In order to read in the data for the characters in the app a SQLlite database was used. In order to do this an empty database is created and then overwritten with my own database. This is done by first using the “dbCheck” method to see if a database doesn’t exist. If it doesn’t then an empty database is created and overwritten with my own database using the “copyDBFromAssets” method. This is all done with the class “databaseClass” which extends “SQLiteOpenHelper”. This class also has a function that reads in a row from the database that I need. This is done in the “findCharacter” function which returns a “characterClass” object. In order to retrieve the correct data the “characterDisplay” class was also used. This class involves the user picking a name from the spinner. When the user picks a name the “getText” method on the spinner is called and its contents are stored in a string. For example if the user picks Donald Duck this will go into the string. This string is then passed as a parameter into the “findCharacter” function from the “databaseClass” and is used for the data in the where clause of the SQL query. The SQL query is stored in a string. This string is then used as a parameter for the “rawQuery” method and the contents of that query are then stored in a cursor. The data in the cursor is then used to set the values for a “characterClass” object and this object is then returned. This data is then used to fill the text field and will display the appropriate data for the characters bio. For the image of the character an image path is set up leaving the specific image name to be found at the time depending on what record is returned as the image column has the image name not the image itself. Once the image path is built the method “setImageResource” is then called on the image view passing in the variable with the path to the image.

# UI Design









# Bibliography

Developer.android.com, (2014). *Design | Android Developers*. [online] Available at: http://developer.android.com/design/index.html [Accessed 19 Dec. 2014].

Google Developers, (2014). *Getting Started*. [online] Available at: https://developers.google.com/maps/documentation/android/start [Accessed 19 Dec. 2014].

Stackoverflow.com, (2014). *Stack Overflow*. [online] Available at: http://www.stackoverflow.com [Accessed 19 Dec. 2014].

Youtube.com, (2014). *YouTube*. [online] Available at: http://www.youtube.com [Accessed 19 Dec. 2014].