|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  | |  | |  | |  | |
| Gift Generator  Project Description Document |
| Pooja Khare  CS443 Fall 2017 Prof. Bo Sheng December 18, 2017 |

Table of Contents

Project Statement 1

Application Design 2

Main Activity 2

Search Results 2

Display Product 4

App Launcher 4

Settings 5

Application Implementation and Evaluation 6

MainActivity Class: 6

AmazonURIGenerator Class: 6

Specify the Secret Key corresponding to the Access Key ID as taken from the Account page. 6

SearchResults Activity Class: 7

AmazonXmlParser Class: 7

Display Product Activity Class: 7

Testing the App 8

References 10

Experiences and Thoughts 11

# Project Statement

I have developed an app that can help a user find presents for someone who can be a friend, family or a relative. This app can be used to buy gifts for a person at any age and gender be it a male, a female, a child or an adult. I had recently attended many weddings in my family and I had no time to buy wedding presents, which let to chaos and last minute shopping. I realized that there must be other people facing same situation. If there could be an app specifically designed to help anyone buy a present for someone without spending too much time, thought or effort then this could be revolutionary app. It is especially helpful during holiday season when you need to buy presents for family and you don’t have the time to find the right gift for every member of your family. Below diagram explains the scenario when you have to buy a birthday gift for nine-year-old boy then this app suggest on buying either a toy car or a soccer ball.

This app will provide many item choices from which you can purchase a gift under your budget. The user must connect their mobile device to Internet (Wi-Fi/Mobile Data) before using this app so that he/she can access the various amazon links to purchase any item they like.

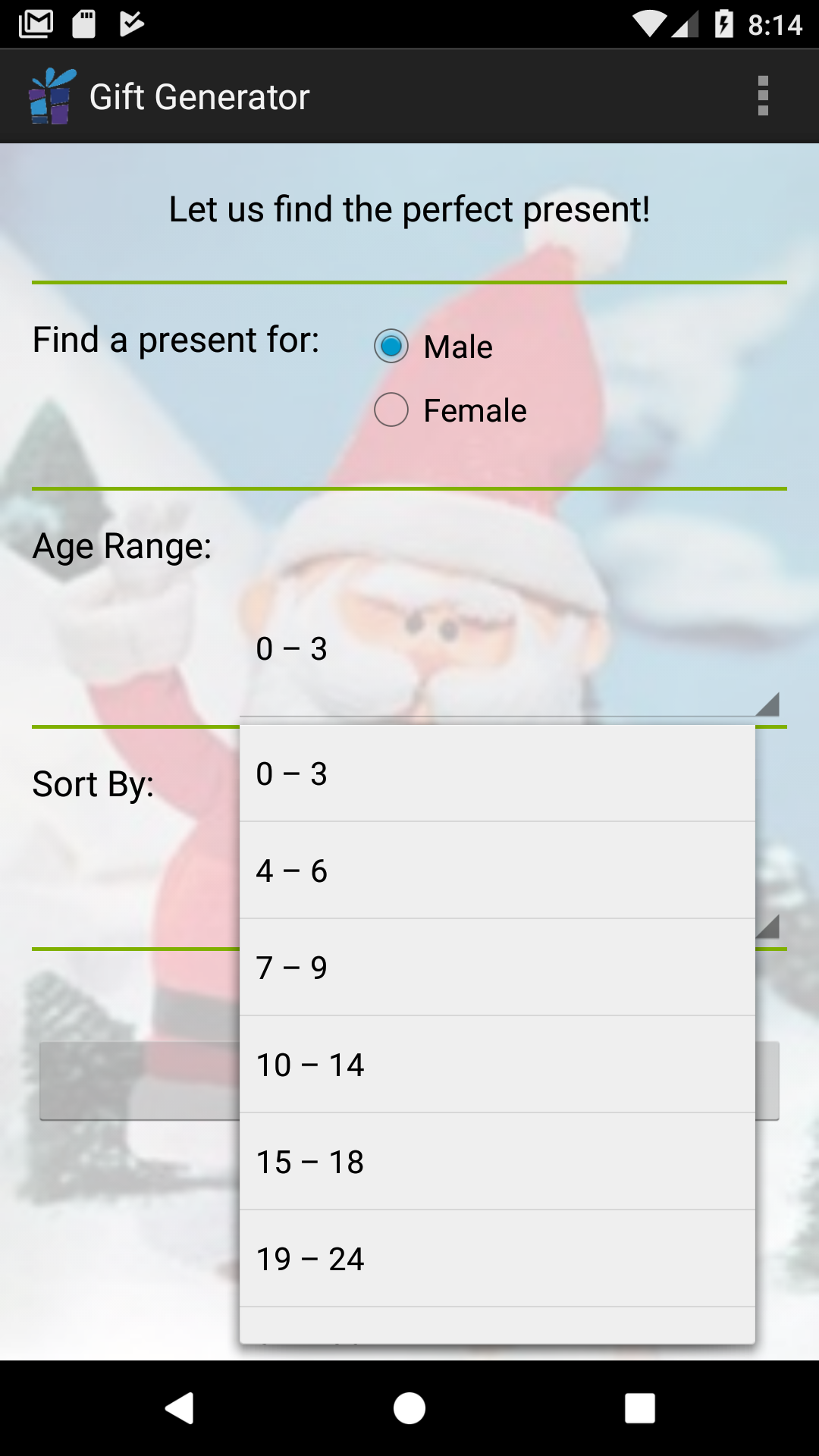
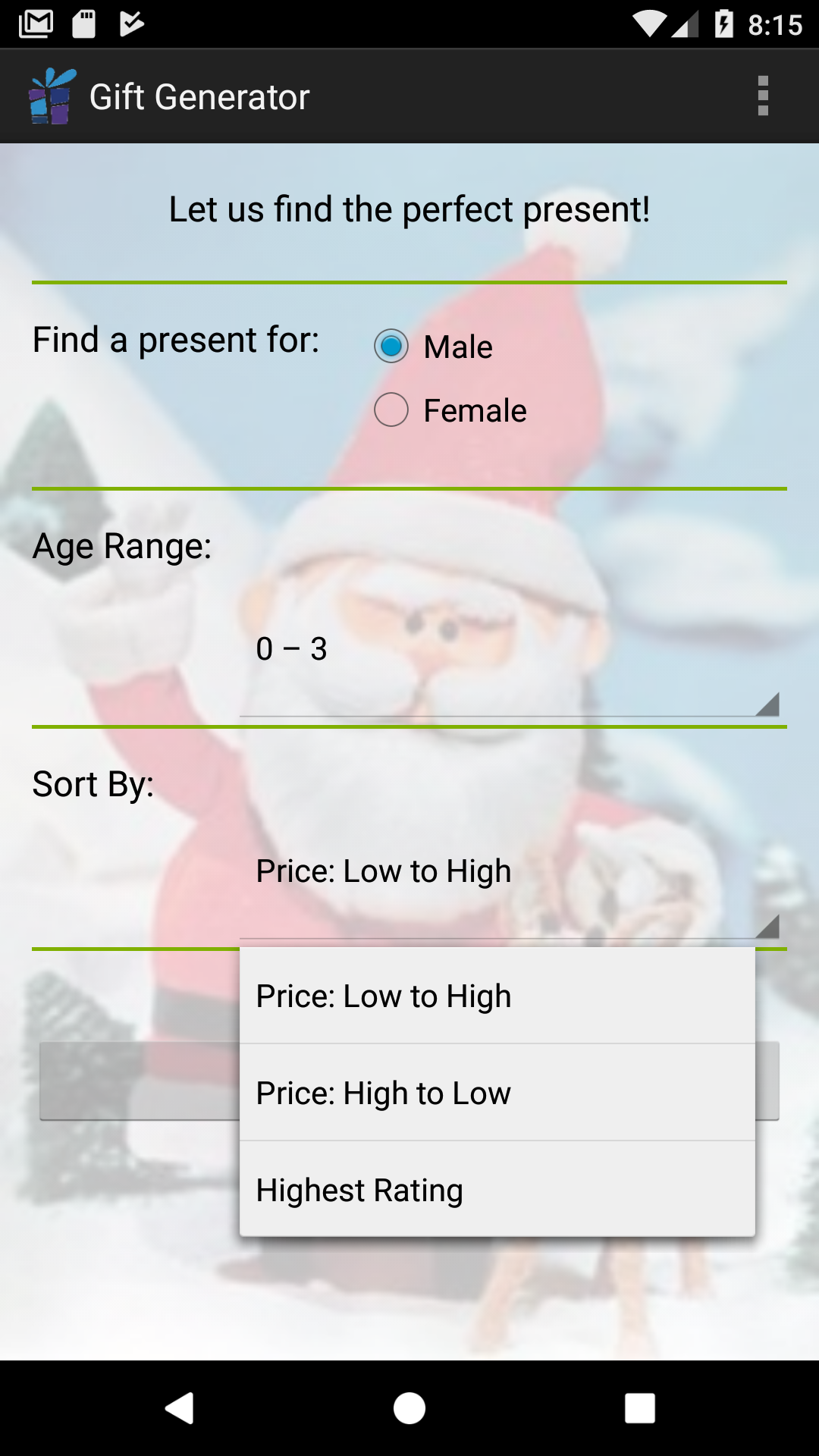
# Application Design

There are following modules in the app:

* Main Activity
* Search Results
* Display Product
* Settings
* App Launcher

## Main Activity

Here the user is asked to feed in the gender, age and either price ordering preference or highest rating preference values. The user selects one radio button from the given two genders i.e. Male or Female. User selects the age group and price sorting order from the dropdown list menu as shown below:

Screenshot 2 Price Ordering Menu

Screenshot 1 Age Range Menu

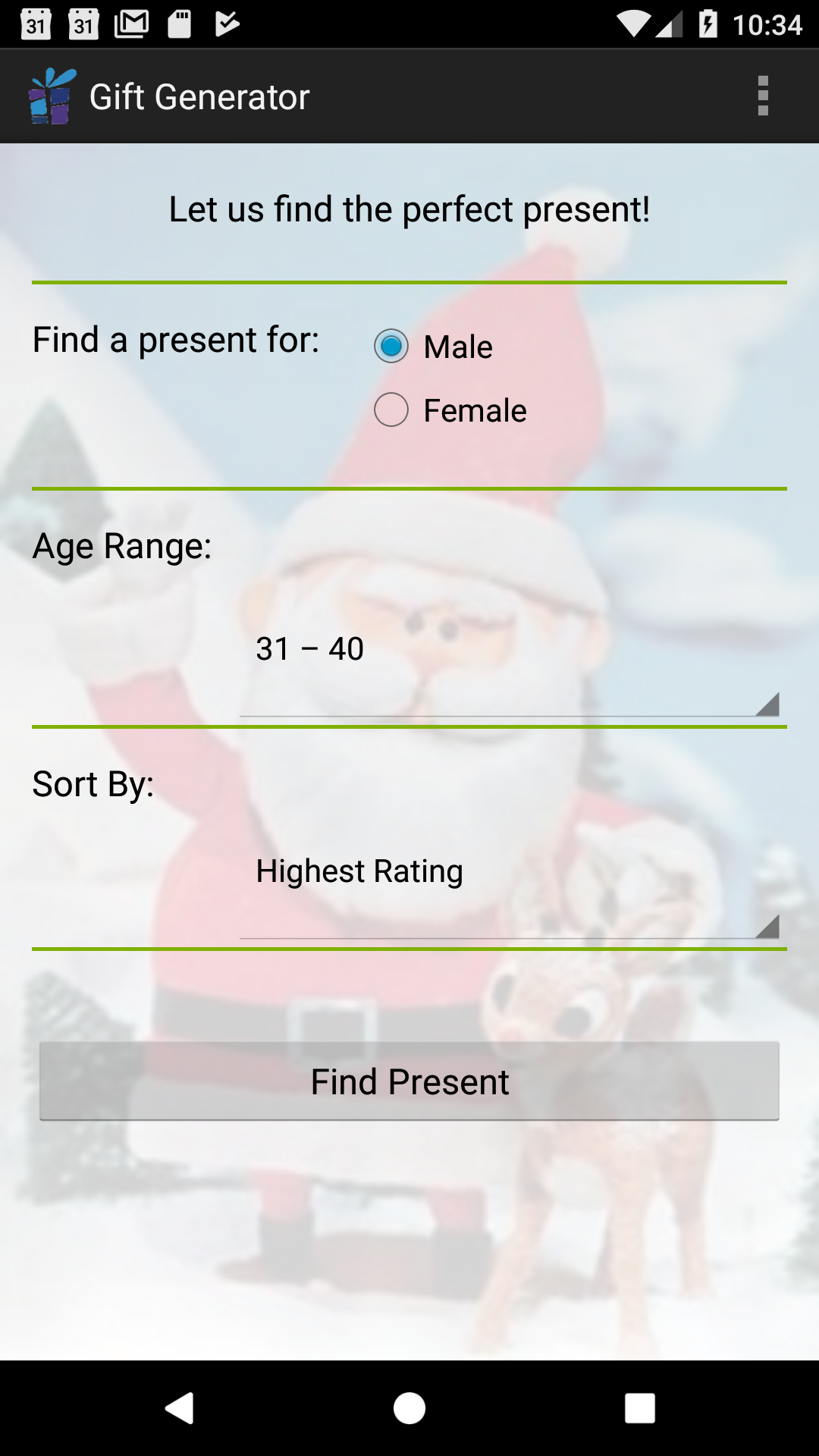
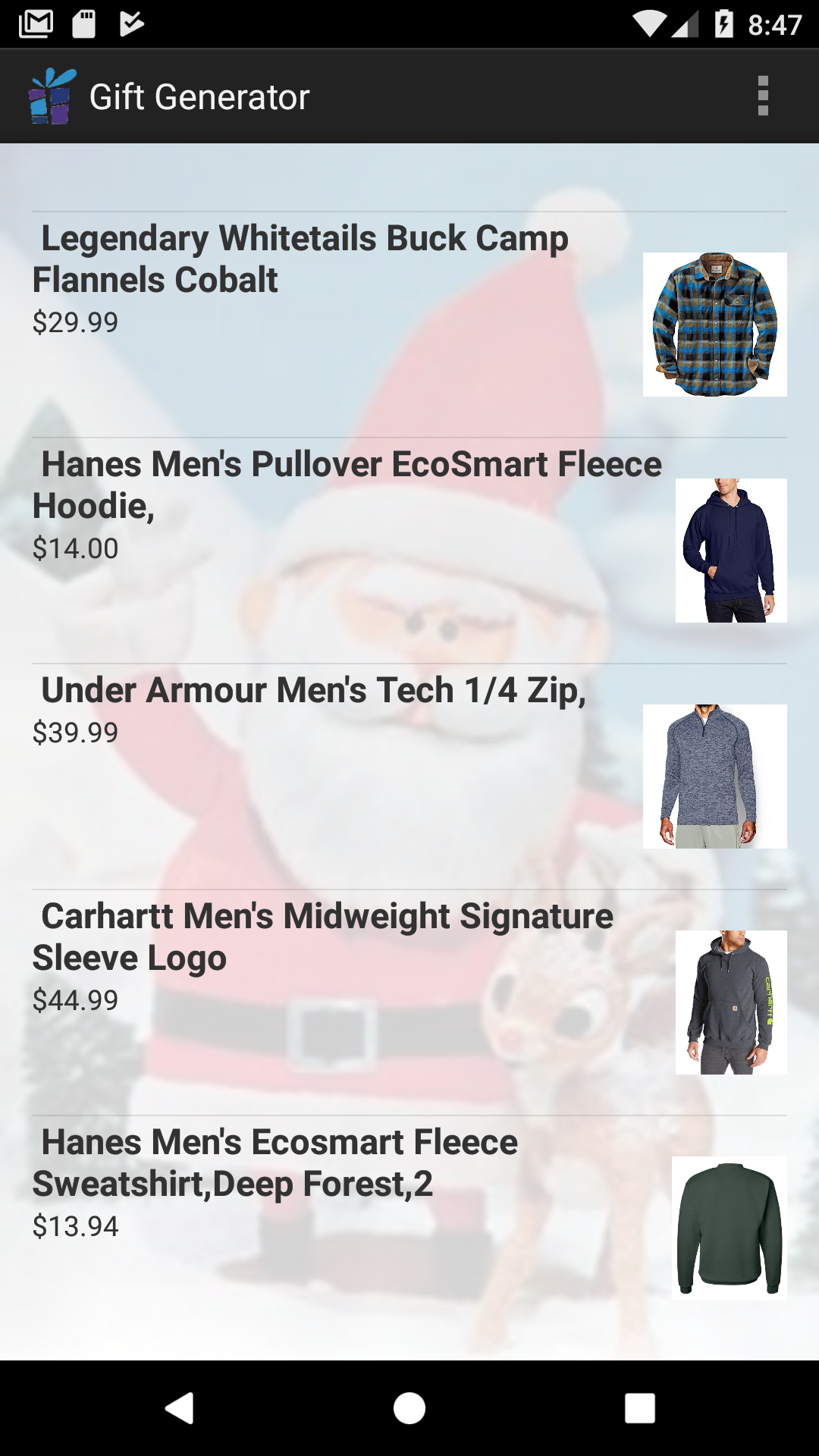
If If the user wants to choose from bestseller’s list then he/she selects the Highest Rating option from the drop down menu.

To specify the gender I have used radio group button within which there are two radio buttons having value stored as Male and Female in the strings.xml file under values folder. Group radio button ensures only one of the radio buttons are set as true. The values both in Age and Sort By menus are generated from spinner in which I have applied an array adapter, which uses spinner layout defined in activity\_main.xml file.

There is a Find Present button, which will call findGifts method when it is pressed set in the onclick attribute of Button in activity\_main.xml layout file

## Search Results

On clicking Find Present button we get the list of Amazon Items in a List View form through which we can scroll through the items and view its image, price and product name all taken from Amazon’s website.

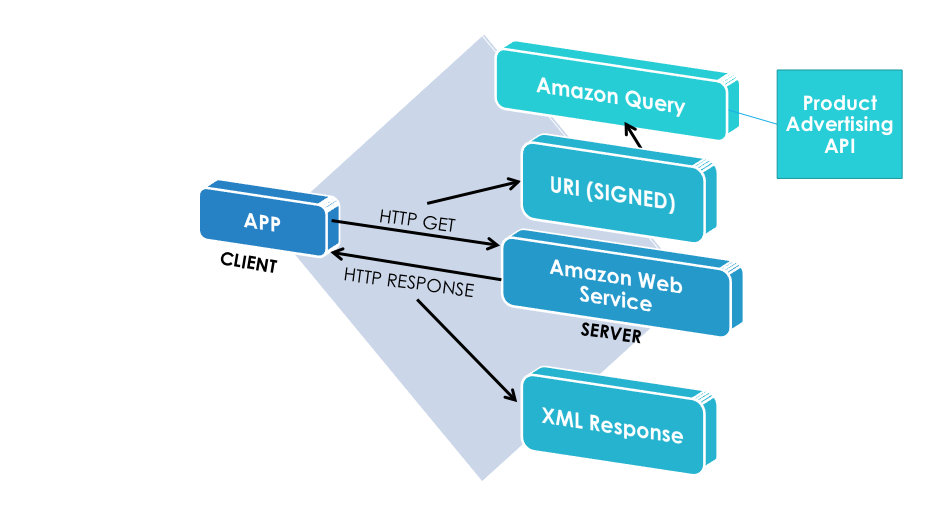
 

Screenshot 4 List View results of Amazon Items

Screenshot 3 Find Present button

Amazon web services are used to gather list of items that is displayed to the user using Product Advertising API. Product Advertising API operations open the doors to Amazon's databases so that we can take advantage of Amazon's e-commerce data and functionality and build our own app to sell Amazon items.

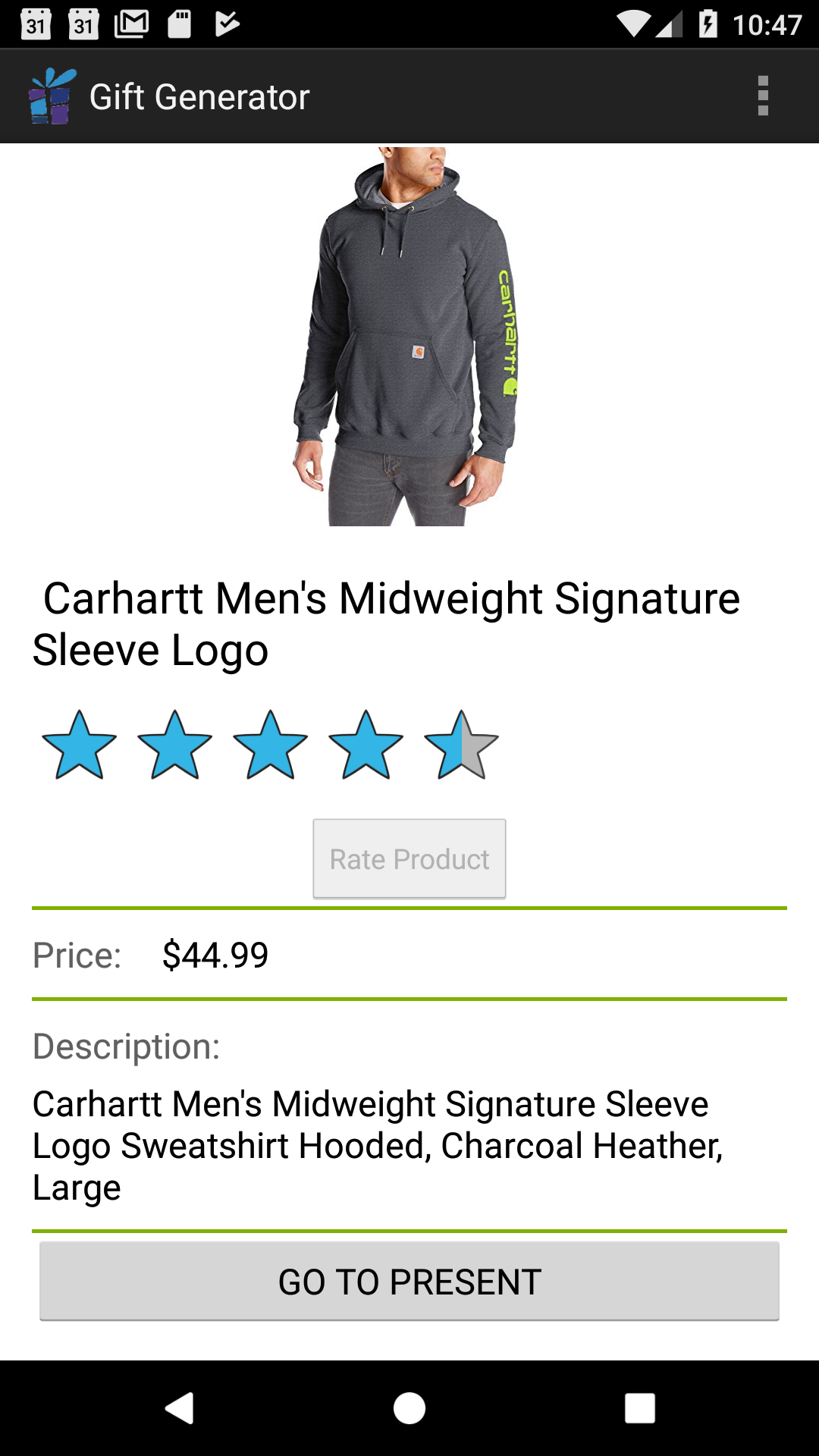
Below flow diagram shows how to get Amazon related data such as image URL, amazon item URL, item attributes, price, etc. from the Amazon product advertising API:



On pressing the Find Present button, we are making a call to findGifts() method in the Main Activity which will start Search Results activity. The Search Results activity queries the Amazon web services using the Product Advertising API to get the results. On the Search Results Activity the user can select individual gift item, which will start the Display Product Activity. The Display Product Activity also uses the Product Advertising API to get information about the selected gift item.

## Display Product

On selecting one of the listed items the app will take you to Display Product Activity, which displays the item’s larger size image, rating, price and a button that takes you to Amazon’s webpage upon clicking it.

Screenshot 6 Amazon webpage of the item for final purchase

Screenshot 5 Display Product of item chosen from the list

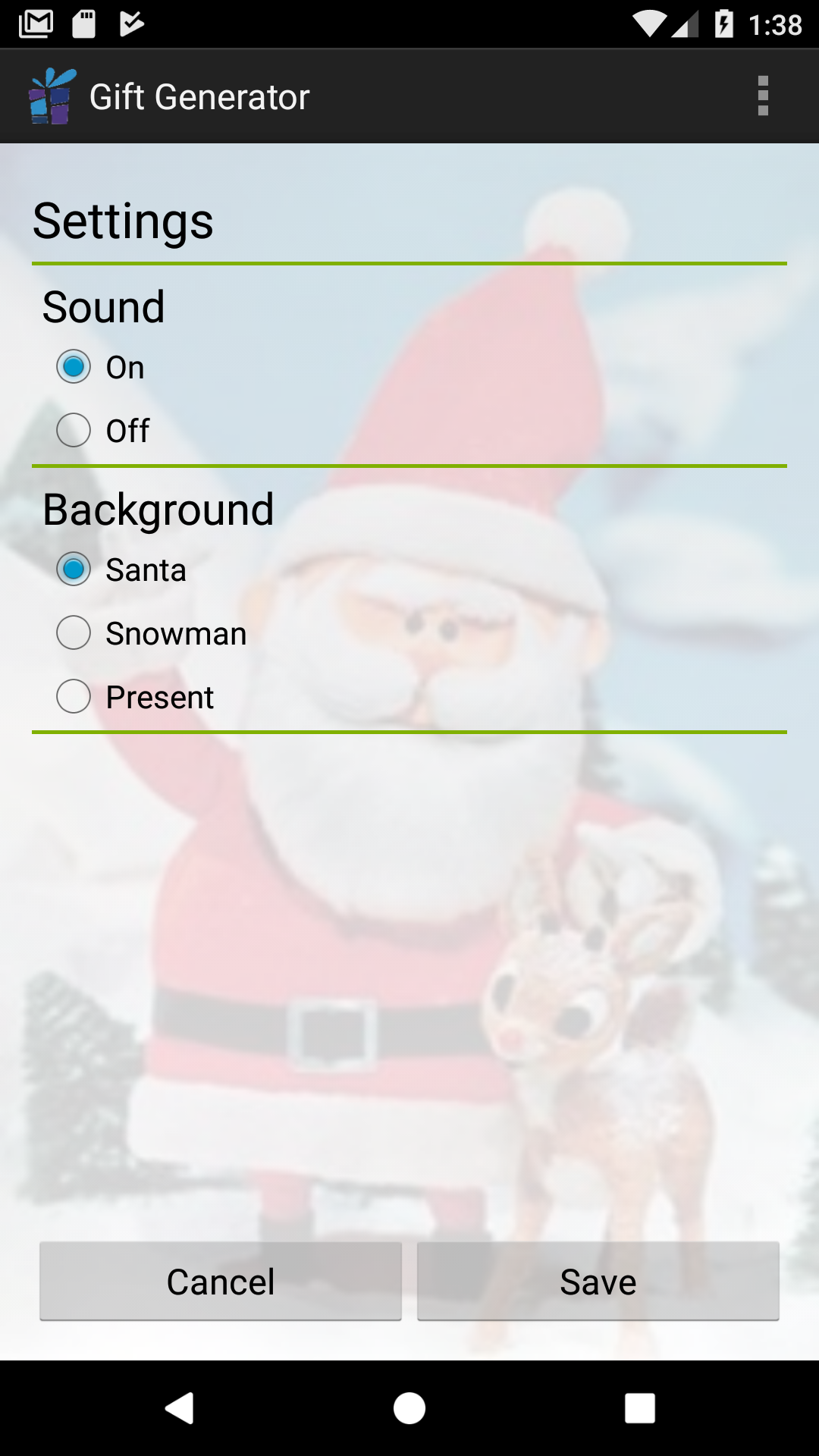


## App Launcher

During the launch of the app, Splash activity is called by androidmanifest.xml file. This activity will use MediaPlayer instance to play the mp3 sound file when the app launches.

## Settings

On clicking the options button we get Settings option displayed. This will take us to Preferences Activity where you can change the background image of the app. You can also set on/off the sound played during the launch of the app as shown in next page:





# Application Implementation and Evaluation

These are the Java classes created under MAIN folder of the app’s SRC folder:

## MainActivity Class:

* findGifts() method:

1. Creates a Bundle and puts all the values taken from the user into it.
2. Puts the bundle into an Intent and open the Search Results Activity where this Bundle instance is passed on.

* getValues() method:

1. Gets user choice of gender stored in a String variable
2. Gets user choice of sort order using price or rating
3. Gets user choice of Age range from the spinner

* onCreateOptionsMenu() method: Inflates the options menu
* onOptionsItemSelected() method:

1. Opens the Preferences Activity page
2. Called when Settings is clicked in options menu

## AmazonURIGenerator Class:

### Specify the Secret Key corresponding to the Access Key ID as taken from the Account page.

* generateURI() method:

1. Set up the signed requests helper.
2. Set up the Search, Keywords, OrderBy parameters for different cases based on gender, price and age requests.
3. Mention the values of Service, Operation, AWSAccessKeyId, AssociateTag, ResponseGroup as well in the parameters to be sent to SignedRequestsHelper sign method.

## SearchResults Activity Class:

* XMLParse Class to create Async Task

1. doInBackground(): We execute the queryDatabase() to get the Amazon items. Next, we download the images for all the amazon items. All the downloaded images are stored in the app’s RAM so that it can be loaded again and again and save more time in uploading images. They are referenced by the amazon item ID as their filename.
2. onPostExecute(): We populate the List View with all the amazon items. Each Amazon item is clickable and launches a new activity using DisplayProduct Class.

* queryDatabase():

1. Based on the user input provided we generate the Amazon Request URI.
2. We use loadXmlFromNetwork() to execute HTTP GET for Amazon Request URI.

* loadXmlFromNetwork() and downloadUrl():

loadXmlFromNetwork() uses downloadUrl() to issue the HTTP GET for Amazon Request URI. The HTTP Response is in XML and it contains the results of the Amazon Request URI. Next, the XML data is parsed to get all the entries. Each entry contains all the metadata information such as ID, Title, Description, Amazon Link, Image URL, price, etc.

## AmazonXmlParser Class:

Parse(): It will parse the XML by using XmlPullParser which will make a call to reedFeed()

reedFeed(): method created to return the List of entries where data from XML is parsed into

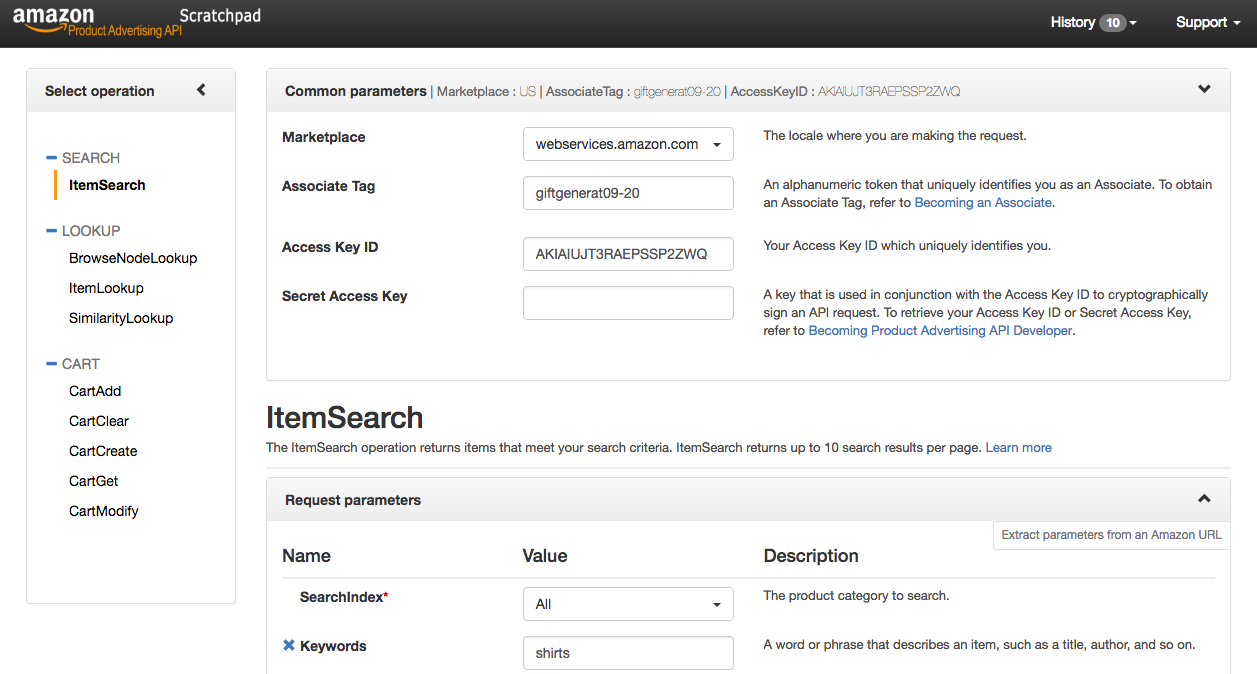
## Display Product Activity Class:

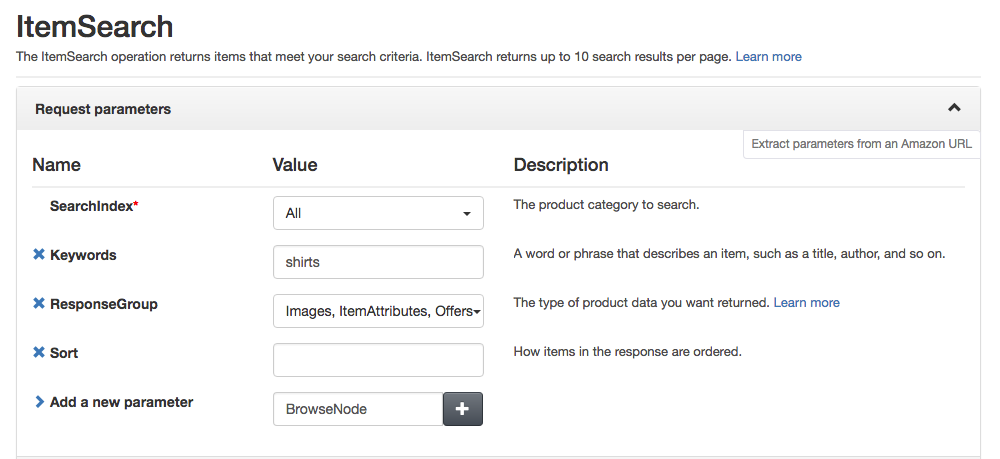
SearchResults bundles up information about particular amazon item chosen by the user. The Display Product activity unbundles this information and uses it to load the Amazon Product.

* getValues(): This method is used to unbundle all the values sent by Search Results Activity.
* XMLParse is the Async task which is used to load the page in background
* doInBackground() downloads the larger image URL for the particular Amazon item.
* goToGift(): When Go to Present button is clicked then this method is called. This method launches the browser with the Amazon item link URI. This will allow user to add this item on the cart on the Amazon web page.

## Testing the App

* Amazon URI generator was tested to make sure that proper Amazon request URI is generated for different user input. Breakpoint was used in AmazonURIGenerator in generateURI() method
* AmazonXmlParser was tested to ensure the proper parsing of XML response data. I tested that for each Amazon item, all the fields were correctly populated.
* Search Results were tested to make sure the download images were of the correct size. I also tested that the bundling of metadata for every amazon item was accurate. This was needed for the display product activity.
* To get the items according to the input values, an error occurred when the sort by value was chosen as Highest Ranking. As each SearchIndex has a particular Sort value associated with it to get the highest rank item, so if I save the value of order by set to “psrank” then it will display products for some cases. To rectify this, I needed to find out each and every case by changing the parameters passed in Sort and checking the generated response gives me no errors. I used Amazon product service API scratch pad to generate responses based on different parameter value sent to it.
* Below is the snapshot of scratchpad where I tested different parameters values of SearchIndex:





# References

* [Sign up as an Amazon Associate http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingAssociate.html](http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingAssociate.html)
* [Sign up for the Product Advertising API http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingDev.html](http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingAssociate.html)
* [What is authentication? http://docs.aws.amazon.com/AWSECommerceService/latest/DG/WhatIsAuthentication.html](http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingAssociate.html)
* [Using the Product Advertising API Scratchpad http://docs.aws.amazon.com/AWSECommerceService/latest/DG/Scratchpad.html](http://docs.aws.amazon.com/AWSECommerceService/latest/DG/becomingAssociate.html)
* Amazon API ItemSearch
* <http://docs.aws.amazon.com/AWSECommerceService/latest/DG/ItemSearch.html>

# Experiences and Thoughts

This app allowed me to create the basic infrastructure to search and display products using a web service. Due to lack of time I could not incorporate ebay products into this app. Ebay also has a similar web service where you can search for products. There are several such web services that could be incorporated into this infrastructure.

I tried implementing a widget for this app and a different navigation scheme to display the search results. I was unable to debug their functionality due to time shortage and that is why I kept it out of the project.

I learned a lot from homework assignments. I think we could have benefited from such small assignments. I was able to use the concepts learned in those assignments, particularly interaction with web services, in implementing this project.