Project4Task1Writeup – Steam Game Data App
By Sisi Rao Andrew id: sisir

Description

My mobile application will take user's Steam ID as input and display the total number of games the user owns on the Steam platform and the game that he played for the longest time. It will then display the user Steam id, game id, name of the game, image icon of a game and the total number time the user spent on this game.

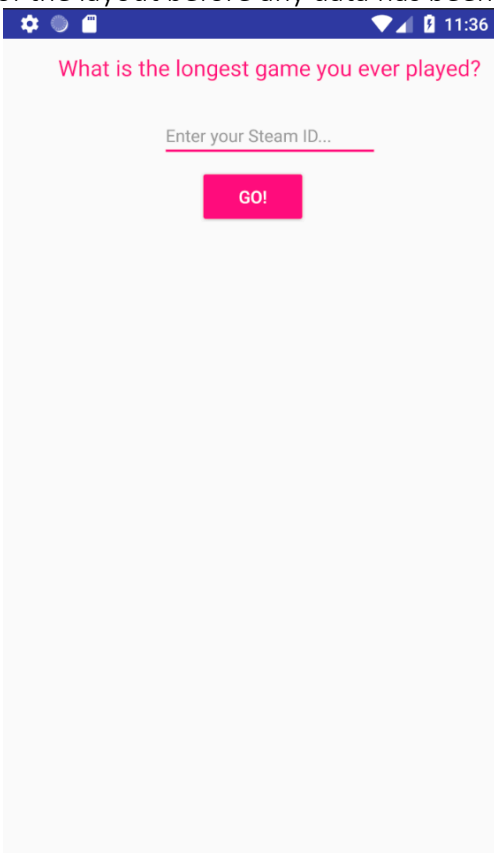
Here is how my application meets the task requirements

1. Implement a native Android application

The name of my native Android application project in Android Studio is: Project4Android.

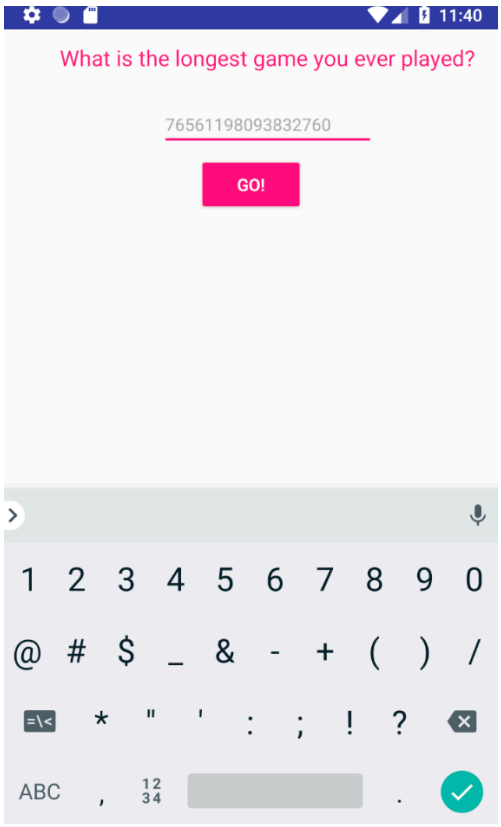
1.1. Has at least two different kinds of views in your Layout

My application uses TextView, EditText, Button, and ImageView. See content_main.xml for details of how they are incorporated into the LinearLayout. Here is a screenshot of the layout before any data has been fetched.



1.2. Requires input from the user

Here is a screenshot of the user searching for Steam data for user id 76561198093832760



1.3. Makes an HTTP request (using an appropriate HTTP method) to the web Service

My application does an HTTP GET request in GetGame.java. The HTTP request is "https://fathomless-beach-88572.herokuapp.com/SteamUserGameServlet?steamID="+steamID
Where steamID is the user's search input.
The search method makes this request of my web application, parses the returned JSON to find the user data and game url, then fetches the game logo picture, and returns the bit image of the picture.

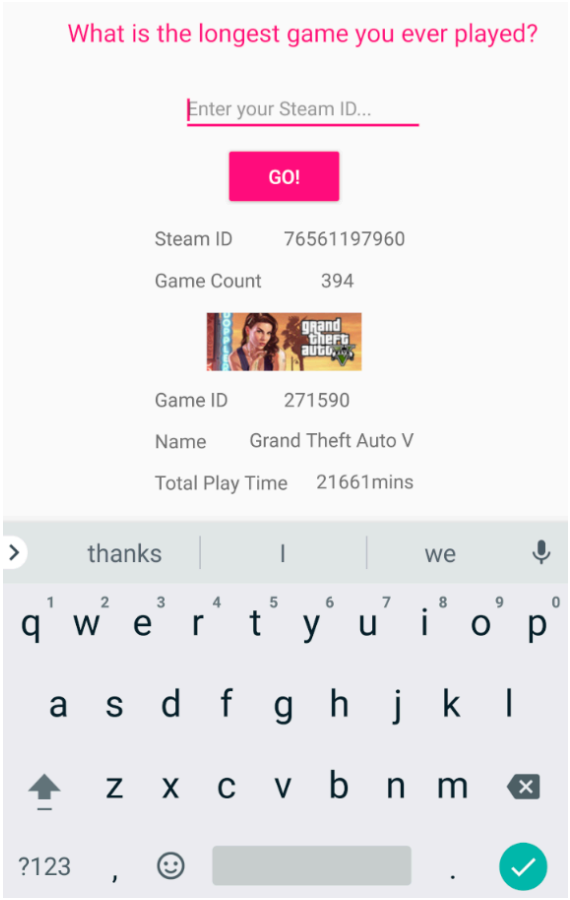
1.4. Receives and parses an XML or JSON formatted reply from the web service

An example of the received JSON reply is:


```
{
	"game_count":44,
	"imgurl":"http:\/\/media.steampowered.com\/steamcommunity\/public\/images\/apps\/292030\/2f22c2e5528b78662988dfcb0fc9aad372f01686.jpg\n",
	"appid":292030,
	"name":"The Witcher 3: Wild Hunt",
	"playtime":5383
}
```

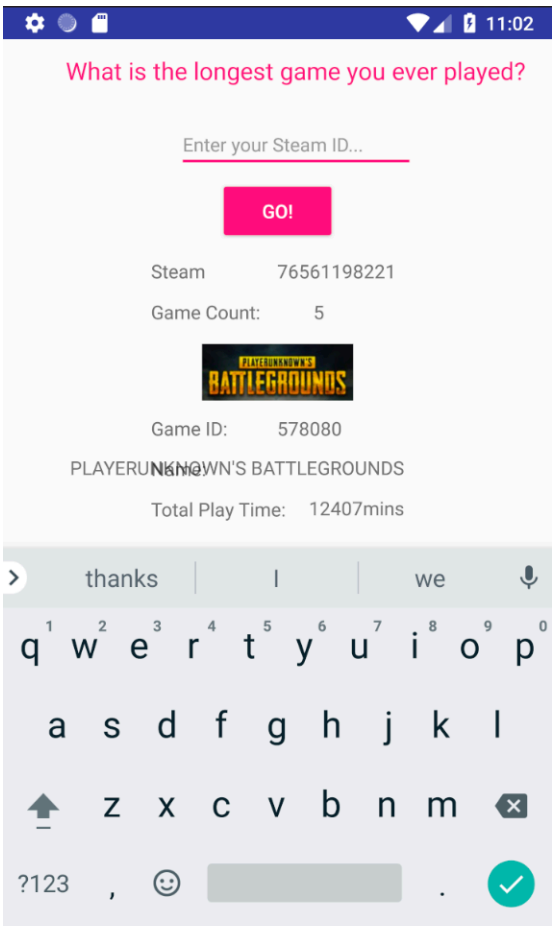

1.5. Displays new information to the user

Here is the screenshot after the user data is returned.



1.6. Is repeatable

The user can type in another search term and hit GO!. Here is an example of having typed in another Steam ID.



2. Implement a web application, deployed to Heroku

The URL of my web service deployed to Heroku is fathomless-beach-88572.herokuapp.com

2.1 Using an HttpServlet to implement a simple (can be a single path) API

In my web app project:
Model: SteamUserGameModel.java
View: index.html
Controller: SteamUserGameServlet.java

2.2 Receives an HTTP request from the native Android application

SteamUserGameServlet.java receives the HTTP GET request with the argument "steamID". It passes this search string on to the model.

2.3 Executes business logic appropriate to your application

SteamUserGameModel.java makes an HTTP request to:
"http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/?key=18C3271FAB442E0163DFDC03C70083A4&steamid="+steamID+"&format=json&include_appinfo=1"

It then parses the JSON response and extracts the parts it needs to respond to the Android application.

2.4 Replies to the Android application with an XML or JSON formatted response.

The model will format the response to the mobile application in a JSON format of my own design:


```
{
        "game_count":44,
        "imgurl":"http:\/\/media.steampowered.com\/steamcommunity\/public\/images\
/apps\/292030\/2f22c2e5528b78662988dfcb0fc9aad372f01686.jpg\n",
        "appid":292030,
        "name":"The Witcher 3: Wild Hunt",
        "playtime":5383
}
```