EECS 118 Final Report

Group name: My Hero on the Go

Group members: Quan Chau, Raul Madrigal, Jon Apostol and Ryan Morrison

Introduction

Project Overview

Our group is working on a native Android app running on Android Smartphone for My Hero. This project is the mobile gateway to connect the user to the large database of inspirational heroes from MyHero.com. We aim to provide user with inspirations wherever they go and whatever difficulties they may face.

Project Deliverables

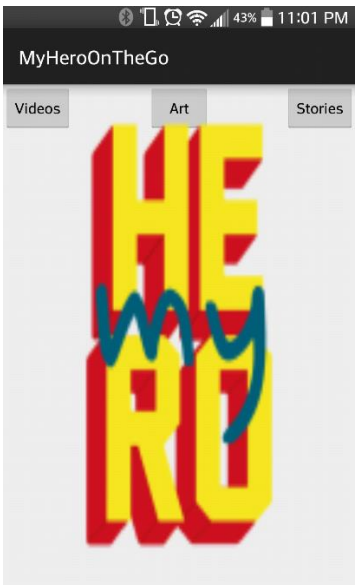
For this project, our group will deliver an Android application that will allow the user to do the following functions:

- Get and display the stories from MyHero.com
- Get and display the arts from MyHero.com
- Get and display the movies from MyHero.com

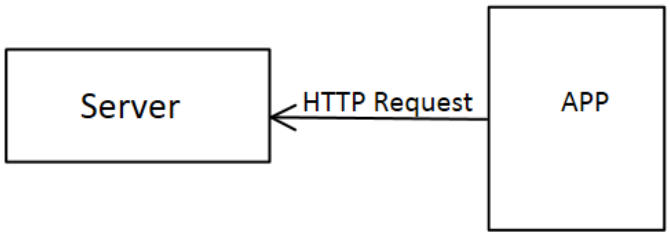
Project Organizations

Process Model

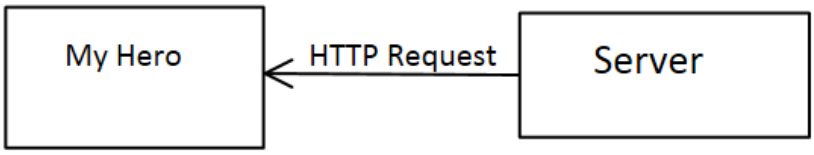
First the user starts the app



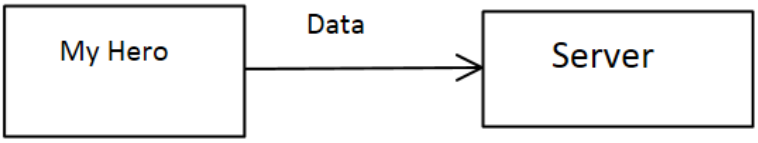
Once a button is clicked the app will first send an http request to our sever to get the desired content



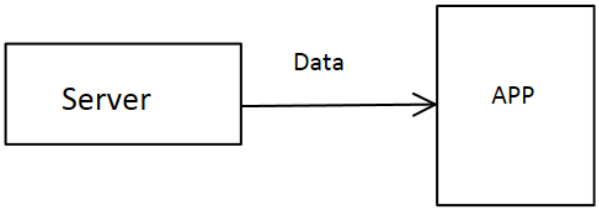
Then our sever will send an http request to my hero to get back certain content on the desired my hero page



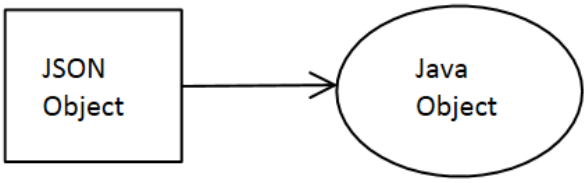
Once the data is fetched, My Hero will send back the required data to our server



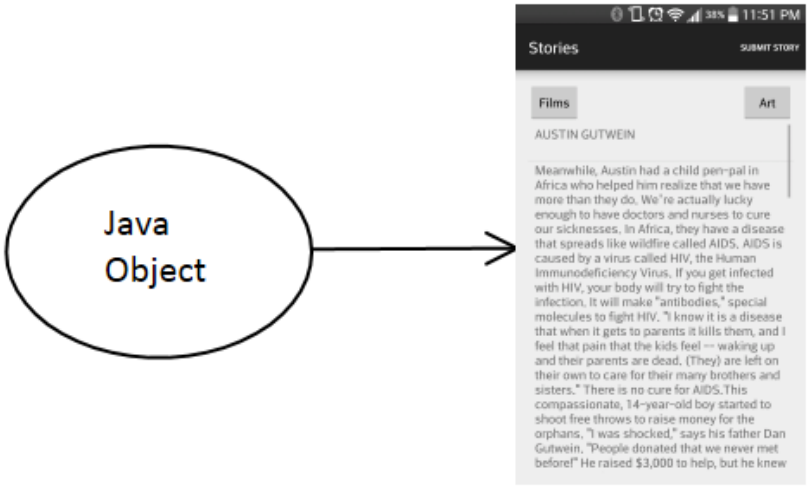
Then our server will send back the received data to our app



The app will then parse the data from a JSON object to a Java object



We will then display the Java object into a readable user friendly format (An example is shown below)



Organization Structure

This project is divided into four main parts:

- An HTTP server to get the contents from MyHero.com server, parse it and pack the data into a nice Json format
- An Android software module that will display the content of the stories and allow user to navigate through the list of stories
- An Android software module that will display the content of the arts and allow user to navigate through the list of arts
- An Android software module that will play the content of the movies and allow user to navigate through the list of movies

Project Responsibilities

In order to complete the project, we divide the work in our group in the following fashion:

- Quan: implement the http server to feed and receive contents to the phone
- Raul: implement the Android module to communicate with Quan's HTTP server and parse the sever data into a format Jon can use to display it to the user's phone.
- Jon: implement the Android module to display stories and arts
- Ryan: implement the Android module to display movies

3. Technical Process

a. Methods, Tools, and Techniques

Here are the list of tools and techniques that we used for this project

On the HTTP server:

- Flask library to receive and send HTTP packets
- BeautifulSoup library to parse and process the HTML files from MyHero.com
- Requests library to get the HTML files from MyHero.com

On the Android application:

- Android Studio
- java.io.Serializable;
- android.content.Intent;
- android.graphics.Bitmap;
- android.graphics.BitmapFactory;
- android.os.AsyncTask;
- android.support.v7.app.ActionBarActivity;
- android.os.Bundle;
- android.util.Log;
- android.view.Menu;
- android.view.MenuItem;
- android.view.View;
- android.widget.AdapterView;
- android.widget.ArrayAdapter;
- android.widget.ImageView;
- android.widget.ListView;
- android.widget.TextView;
- java.io.InputStream;
- java.util.ArrayList;
- org.apache.http.client.ClientProtocolException;
- org.json.JSONArray;
- org.json.JSONException;
- org.json.JSONObject;
- java.io.IOException;
- java.util.ArrayList;
- android.app.ProgressDialog;
- android.content.Intent;
- android.media.MediaPlayer;
- android.net.Uri;
- android.text.method.ScrollingMovementMethod;
- android.widget.MediaController;

- android.widget.VideoView;
- org.apache.http.client.ClientProtocolException;
- org.apache.http.client.HttpClient;
- org.apache.http.client.ResponseHandler;
- org.apache.http.client.methods.HttpGet;
- org.apache.http.impl.client.BasicResponseHandler;
- org.apache.http.impl.client.DefaultHttpClient;

A key point in this project is the agreement on the HTTP requests format from the Android app to the Flask server. Our group had a meeting where we decide on what functionalities we'd need from the HTTP server. Afterwards we code the HTTP interface in the way that we feel most comfortable. Here is a code snipset that shows the HTTP URL that the Flask server expects from the Android app.

```
@app.route('/')
def hello_world():
	return "Hello world!"
#----GROUP OF STORIES-----
@app.route('/getAllStory')
def return_all_story():
	return json.dumps(myhero_story.get_story_type_description())
@app.route('/storyList/<story_category>')
def return_story_in_category(story_category):
	return json.dumps(myhero_story.get_story_in_type(story_category))
@app.route('/story/<story_link>')
def return_story_content(story_link):
	return json.dumps(myhero_story.get_story_content(story_link))
#----GROUP OF ART--------
@app.route('/getArtMedium')
def return_art_medium():
	return json.dumps(myhero_art.get_art_medium_list())
@app.route('/artList/<art_category>/<page_num>')
def return_art_list(art_category, page_num=1):
	return json.dumps(myhero_art.get_art_list(art_category, page_num))
@app.route('/art/<art_key>')
def return_artwork(art_key):
	return json.dumps(myhero_art.get_artwork(art_key))
#-----GROUP OF MOVIE----
@app.route('/getAllMovie')
def return_movie_list():
	return json.dumps(trim_objectid_list(list(client.db.movie_list.find())))
@app.route('/movie/<movie_key>')
def return_movie(movie_key):
	return json.dumps(trim_objectid(dict(client.db.movie_list.find({'movielink':
movie_key})[0])))
```

b. Software Documetation

Below are the UML diagrams of our projects. These diagrams explain how the software modules are structured and how the user interacts with our applications.

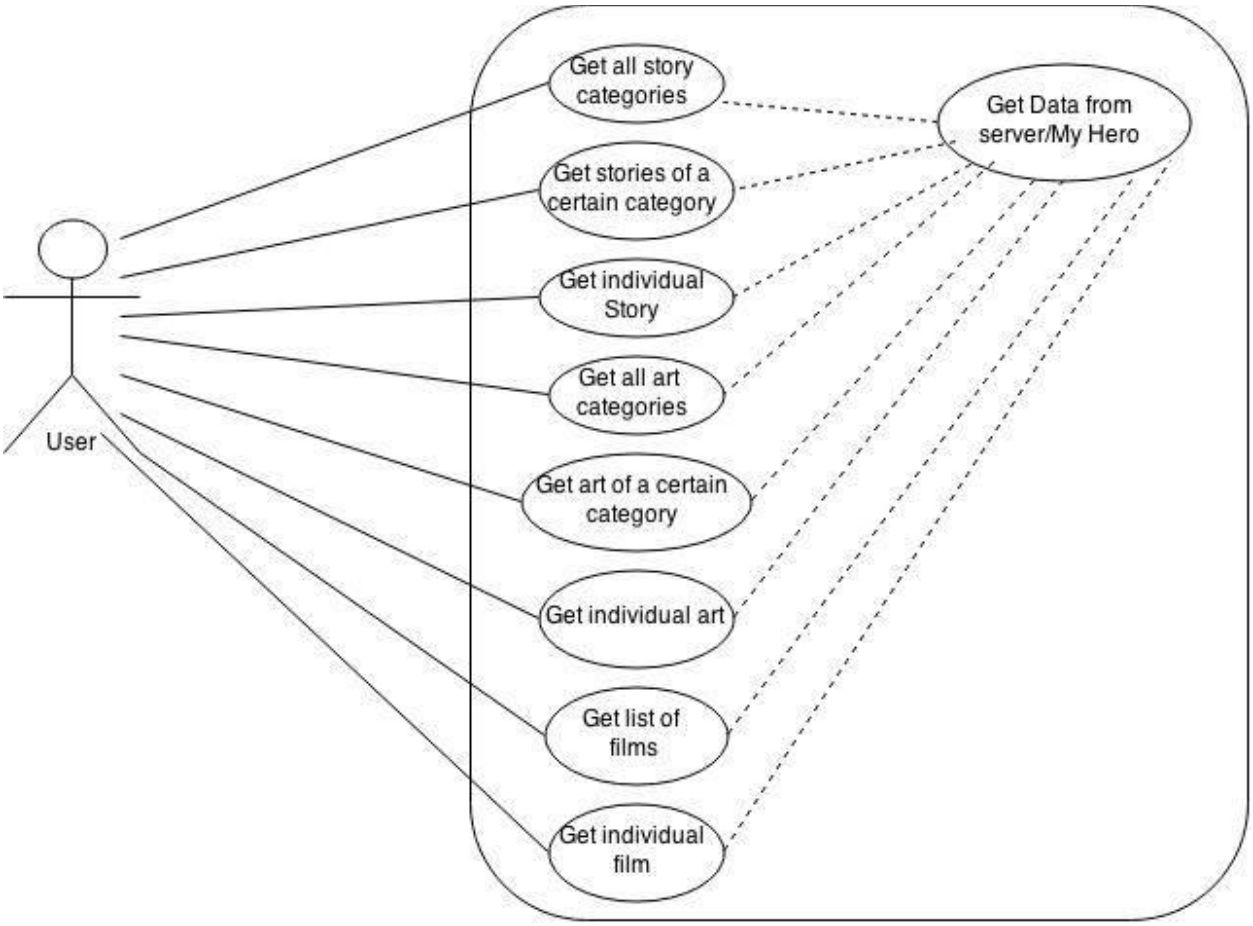


Diagram 1: How user interacts with the Android Application

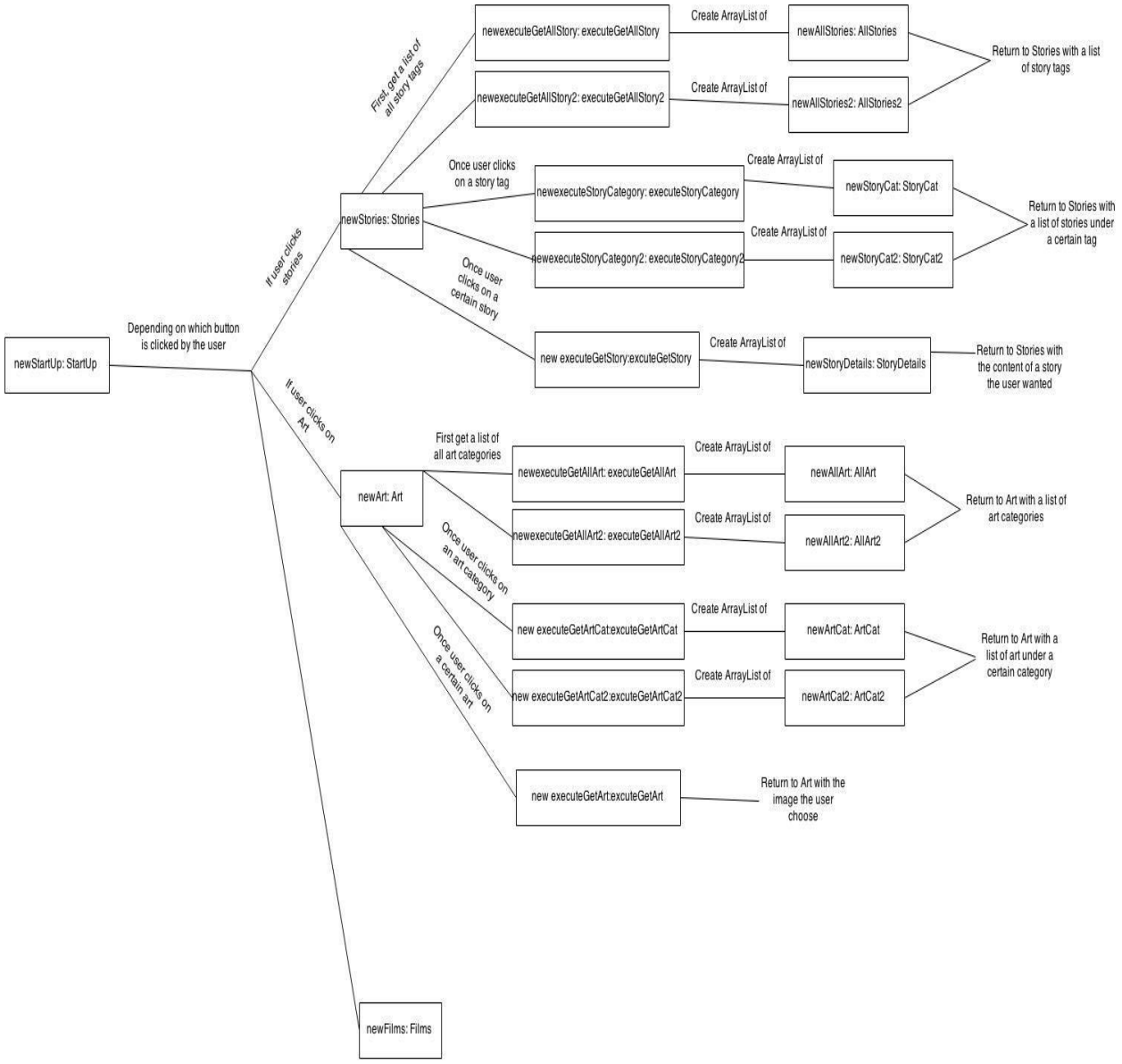


Diagram 2: The inside structure of the Android Application

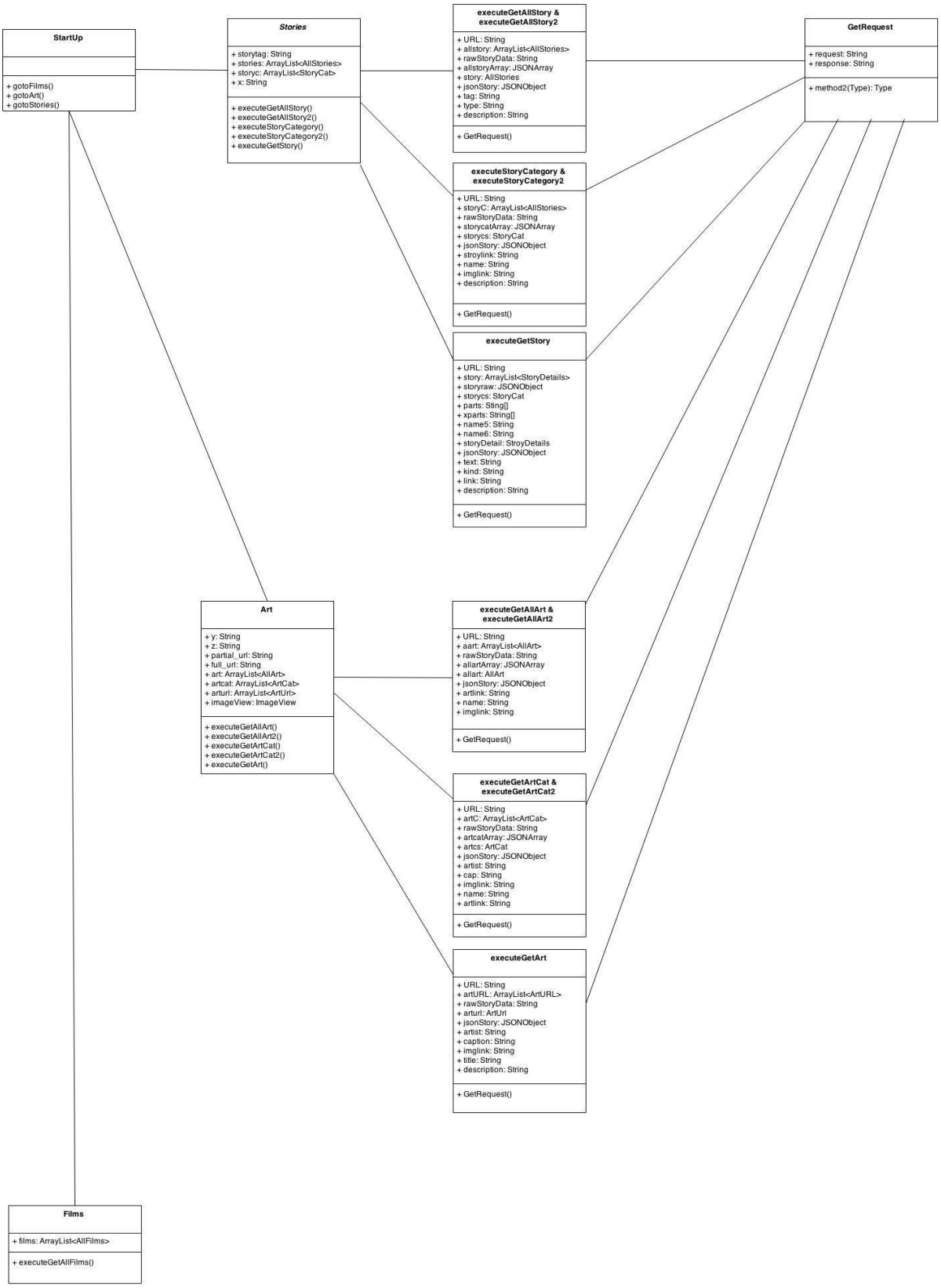


Diagram 3: The description of method and their arguments on Android Application

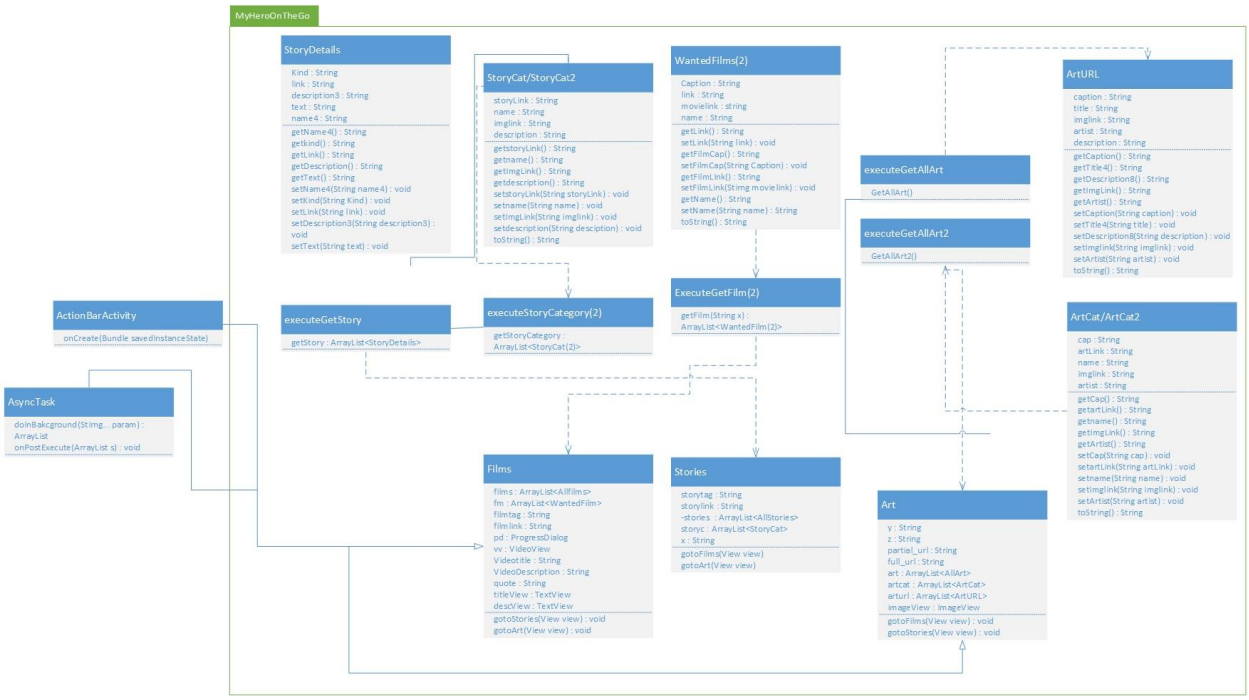


Diagram 4: Detail UML diagram of the relationship between classes on Android application

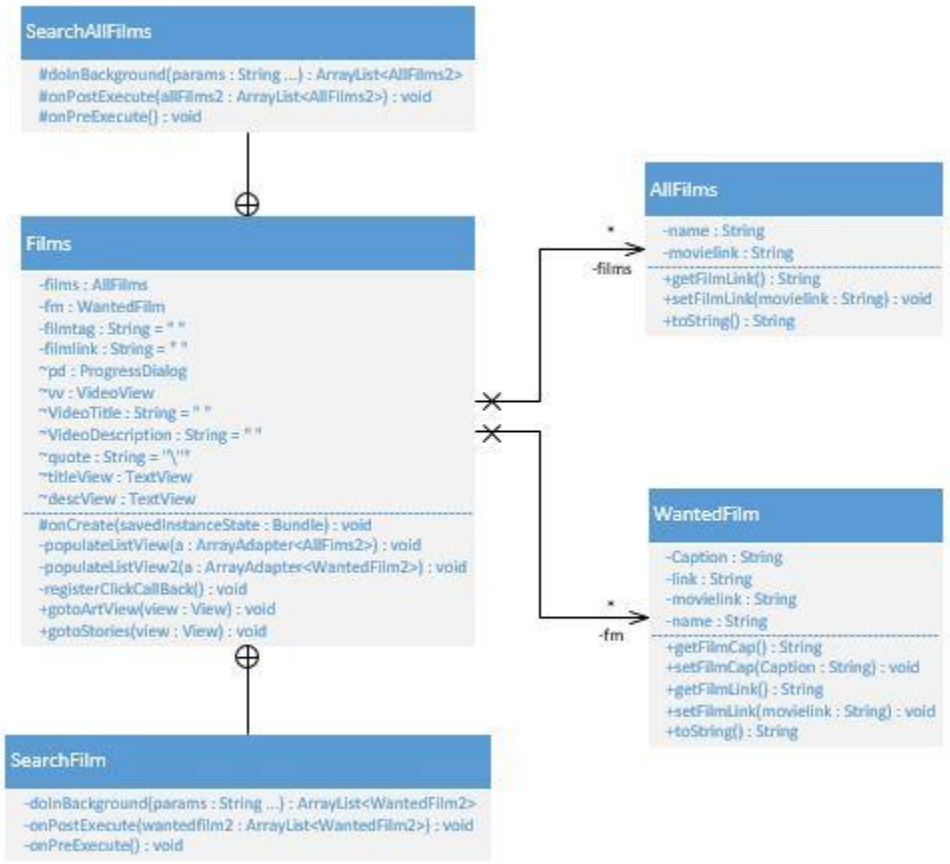


Diagram 5: Detail UML diagram of the Android functions that render the list and contents of movies

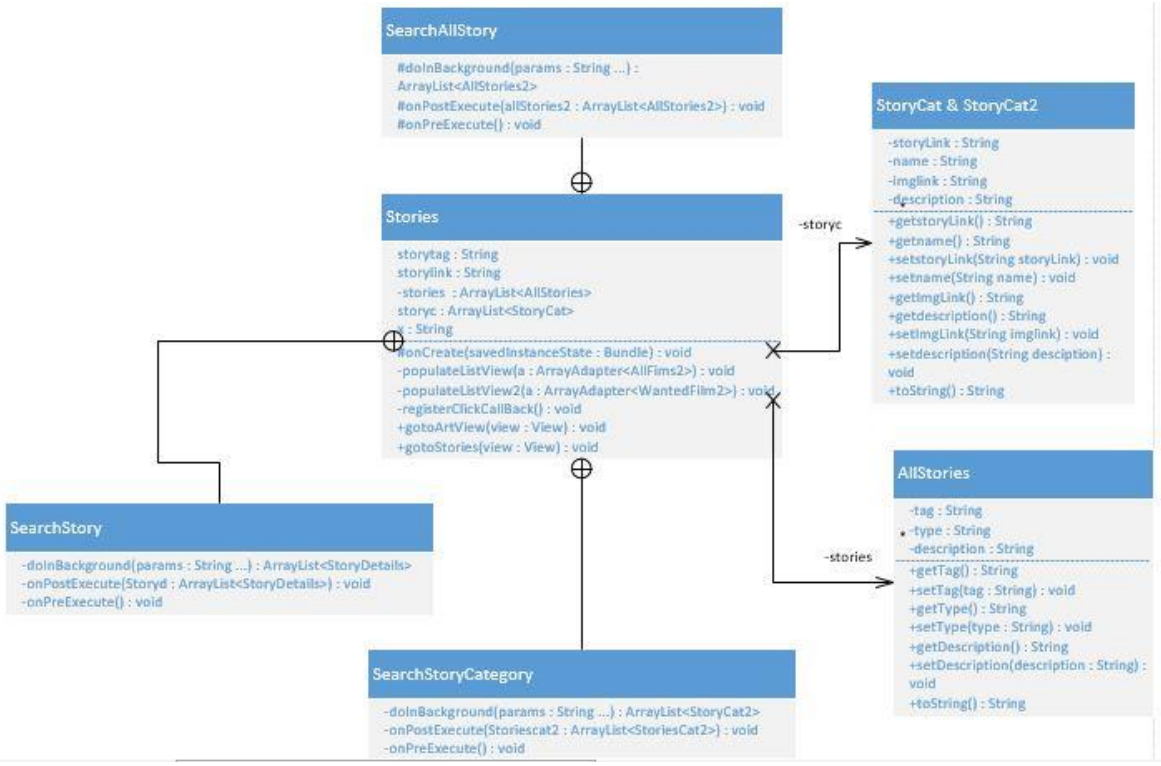


Diagram 6: The relationship of classes and methods that display list and content of stories on Android

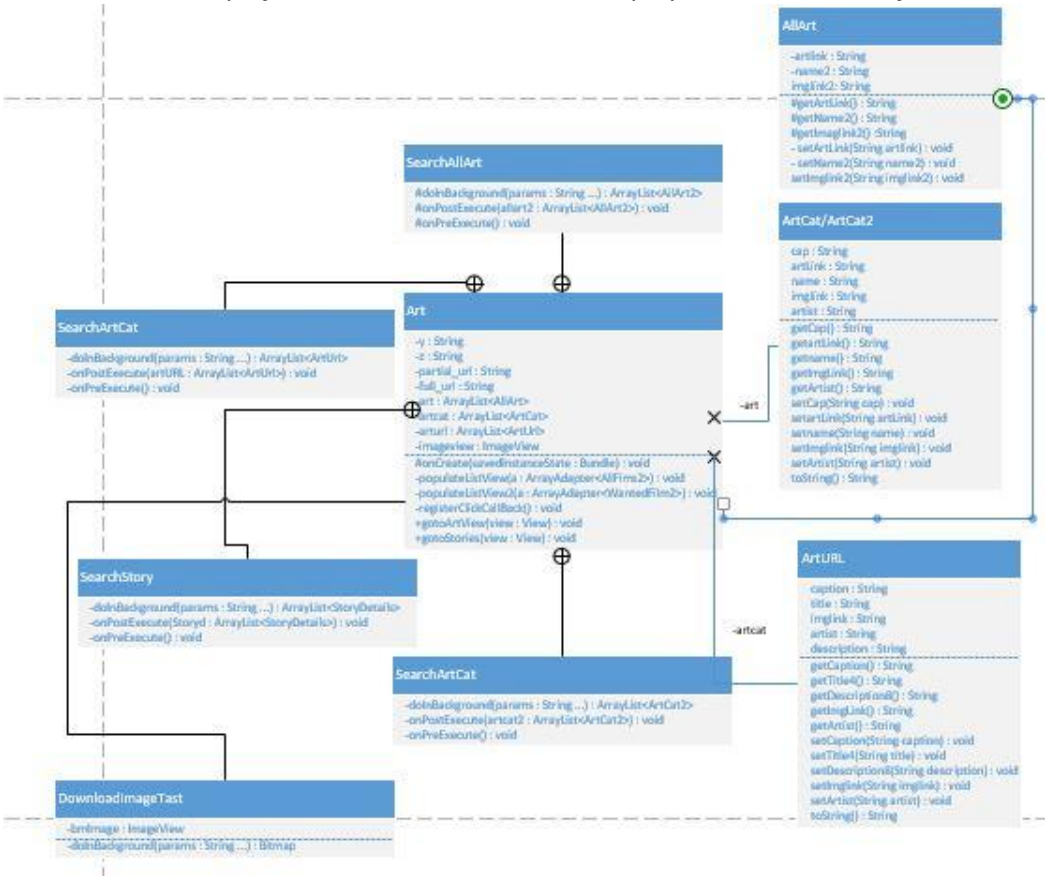


Diagram 7: The relationship of classes and methods that display list and content of arts on Android

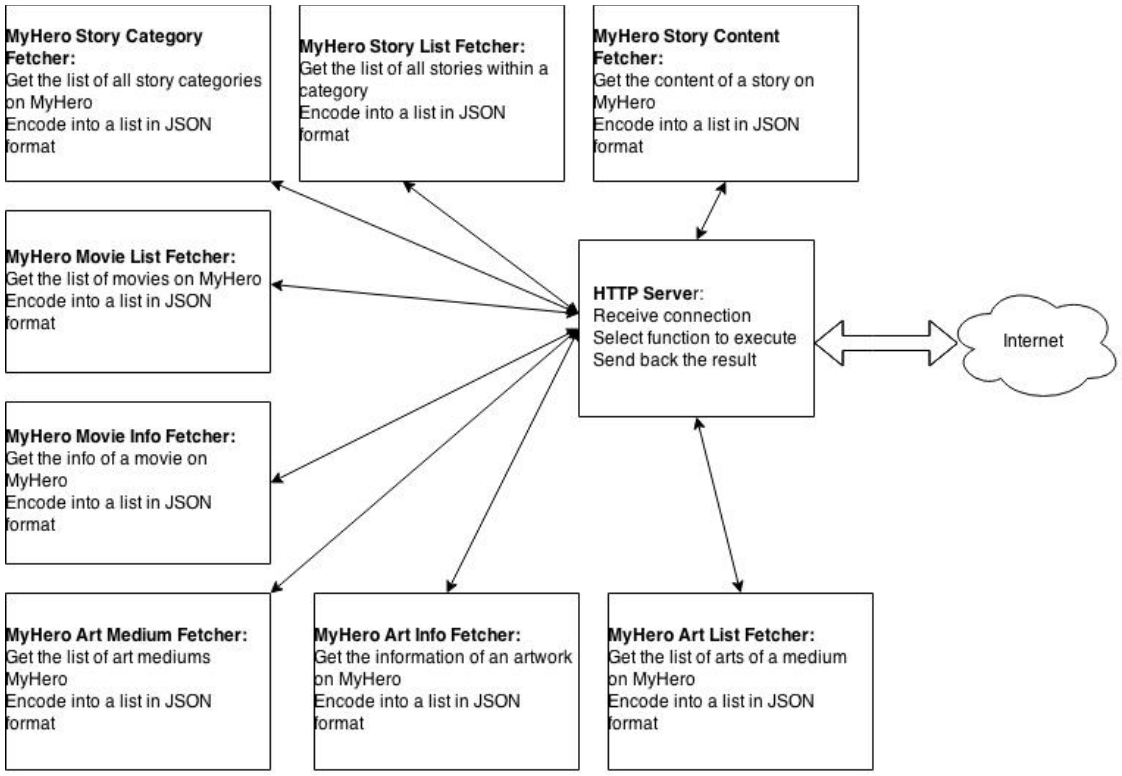


Diagram 8: The inner strucutres of the software module on the HTTP server

4. Testing

To test our software, we randomly select stories, arts and movies from MyHero.com and try to render that on the Android Application. If the app fails in the middle for any reasons, we investigate and fix the bugs. Below is a list of our test cases:

a) For stories:

- Austin Gutwein
- Alice in Wonderland
- Anne Shirley
- Balto
- Hua Mulan
- Woodrow Prater
- Alexandra Cousteau
- Caroline Cannon
- Deland Chan
- Dr. Vandana Shiva
- Gerald Durrell
- J.N. (Ding) Darling

b) For arts:

- Raja Weksler's Story by Doug Miller
- A great Russian painter
- Appointing of Cabinet Members by Arthur Zhinsman
- Family Hero
- Super Heart by Catalin from Cluj-Napoca, Romania
- Tatra Mountain and the small shrine by Krystyna Szymkowiak
- Denisa, the Vet by Denisa Kiss from Cluj-Napoca, Romania
- Doctor D by Alexandra from Cluj-Napoca
- Steve Jobs by Aubin Freville from Belo Horizonte

- Anna Pavlova by Rita Farias from Belo Horizonte

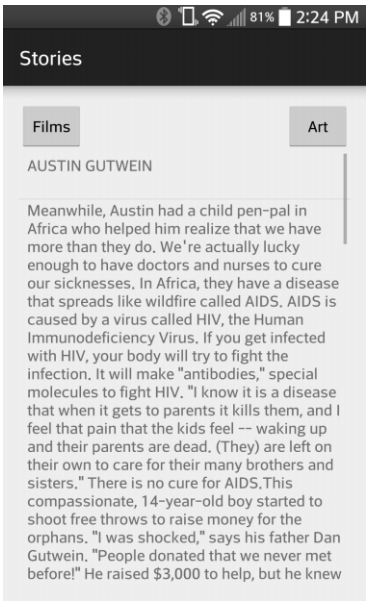
c) For movies:

- The Mailbox
- Savior of the seas
- Make A Wish
- Get A Clue
- Ellie Wen
- Mattie Stepanek

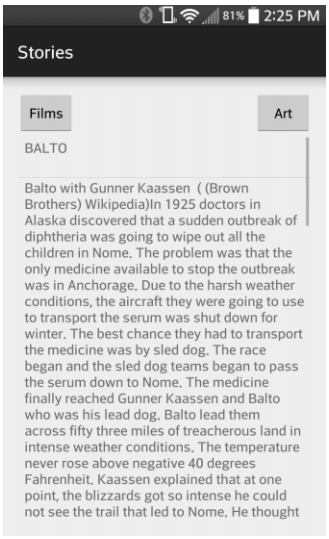
Below are the screenshots of some of our test results:

a) Stories:

Austin Gutwein

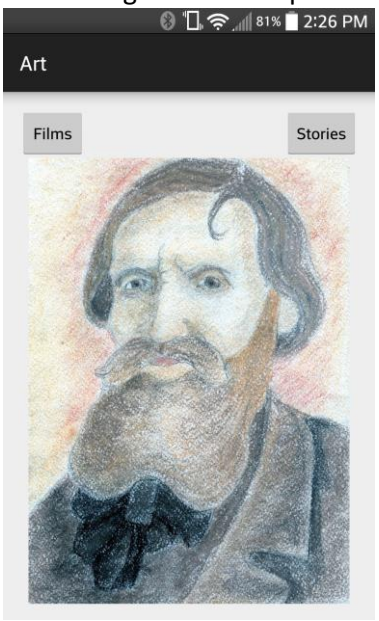


Balto

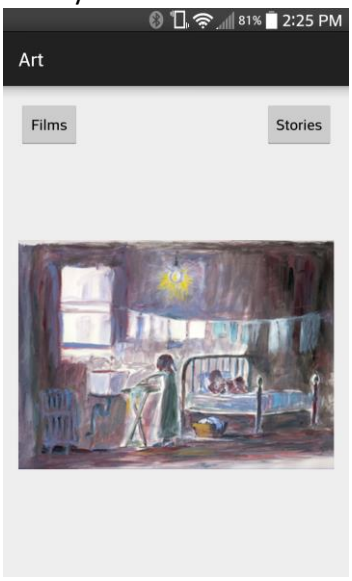


b) Arts:

A great Russian painter

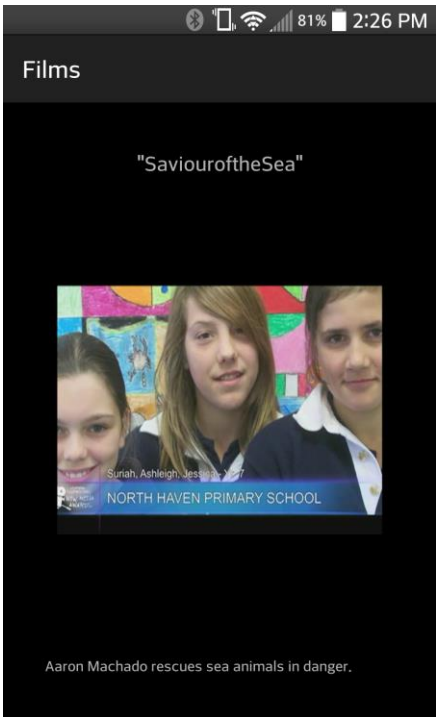


Family Hero

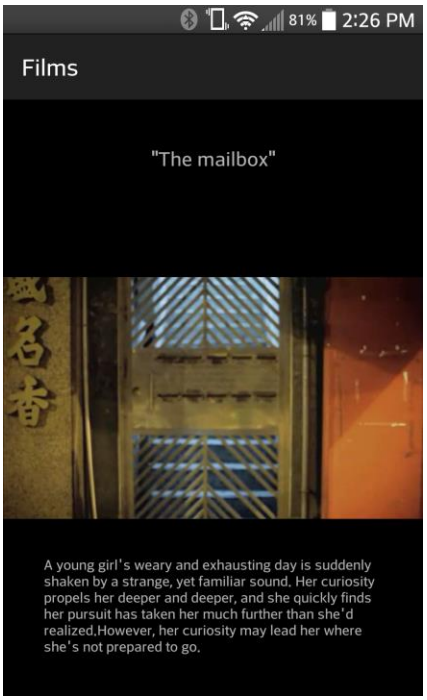


c) Movies:

Savior of the seas



The mailbox



5. Conclusion:

Our group has successfully implemented the Android application and HTTP server to serve the contents from MyHero.com. All of our tests passed and we were able to deliver the contents from MyHero.com to users' Android phones in a nice display. We believe this project will enhance MyHero.com's user experiences and improve their quality of life by providing inspirational contents at every moment.