Project4Task1Writeup **Kritica Sinha - kriticas**

1. Implement a native Android application

a. Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, or anything that extends android.view.View).

The Android app contains the following different types of view:

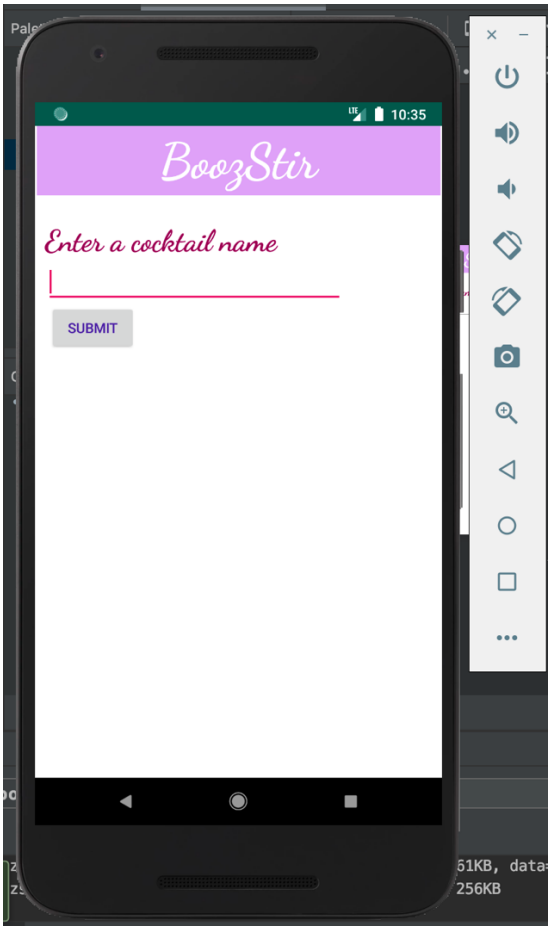
TextView, AutoCompleteTextView, Image View, ScrollView and a button

Three TextViews provides the prompt to user to enter the cocktail name, display the drink information that is drink name, ingredients, measures, instructions, and the third TextView is for the app logo display

AutoCompleteTextView takes user Input and starts prompting with suggestions of prepopulated drink names

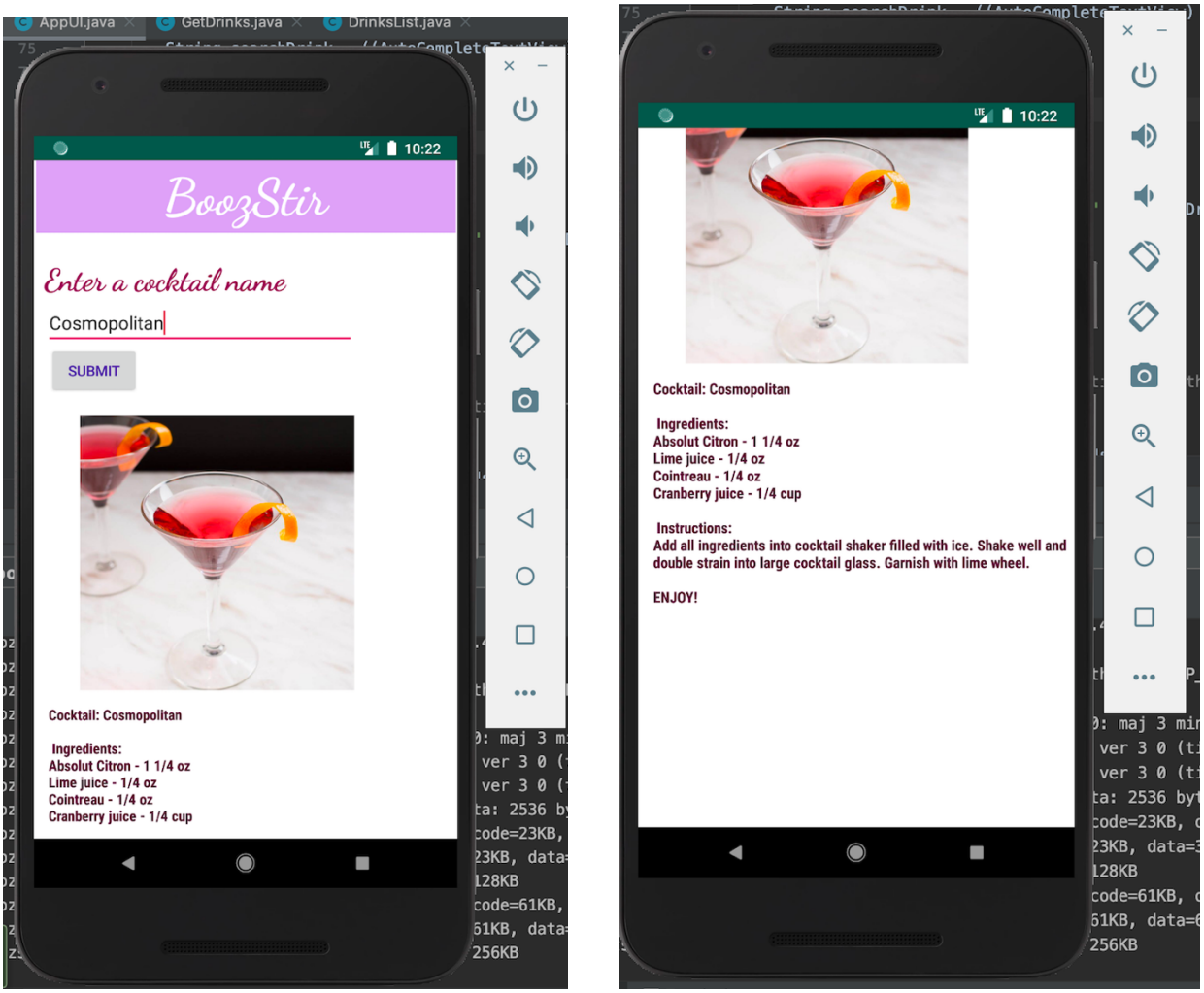
ImageView displays the searched drinks image

ScrollView to scroll the page vertically to show all the content.



b. Requires input from the user

The display on user Input and the user can scroll to view the entire content



The results are displayed in the TextView
To simplify the app working the users have been constrained to put drink names which are present in the DrinkList.java. These drink names are the most famous and popular cocktail names which have been retrieved from 3rd Party API and form a subset of the larger set. More drink names can be added, and the list can be increased.

The popular cocktail list is as follows –

"Amaretto","Afternoon","Affair","Afterglow","Acid","Bellini","Bluebird","Brooklyn","Blackthorn","Bloody Mary","Blue Lagoon","Cherry Rum", "Cuba Libre","Cream Soda","Casa Blanca","Cosmopolitan","Derby","Daiquiri", "Dragonfly","Egg Cream","English Highball","French 75","Frisco Sour","French Martini", "Frozen Daiquiri","Gin Fizz","Gimlet","Godchild","Gin Sling","Halloween Punch","Hawaiian Cocktail", "Hemingway Special","Irish Cream","Irish Spring","Imperial Fizz","Jelly Bean","Jackhammer","Jello Shots", "Jamaican Coffee","Kir","Kamikaze","Kiwi

Lemon","Long Vodka","Long Island Tea","Martini","Margarita", "Mojito","Mimosa","Moscow Mule","Manhattan","Negroni","Old Fashioned","Paloma","Pink Lady","Pink Gin", "Pina Colada","Rum Sour","Royal Flush","Spritz","Scooter","Snowball","Sangria","Tequila Fizz","Tequila Sour", "Vodka Fizz","Vodka Martini","Vodka Russian","Whisky Mac","Wine Punch","White Lady","White Russian","Yellow Bird","Zombie", "Screwdriver", "Rum Screwdriver", "Absolutly Screwed Up", "Mint Julep", "Sex on the Beach" ,"Screwdriver", "Campari Beer", "Homemade Kahlua", "Bailey's Dream Shake", "B-52"

The information on these drinks are fetched from the API

c. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does an HTTP POST request in GetDrinks.java. The HTTP request is posted to the URL:

https://blueberry-sundae-22185.herokuapp.com/getMyCocktail

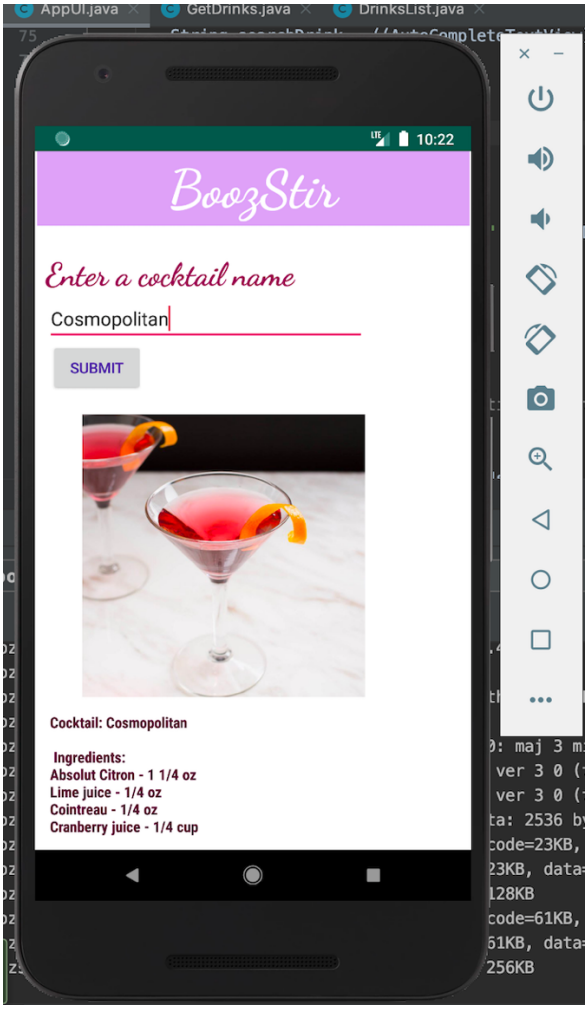
The user search string is made into a Json and the sent to the webservice which deployed on Heroku. The webservice then contacts the TheCocktailDB API and gets the information in form of Json string and this is converted into information to be displayed by the app as shown above

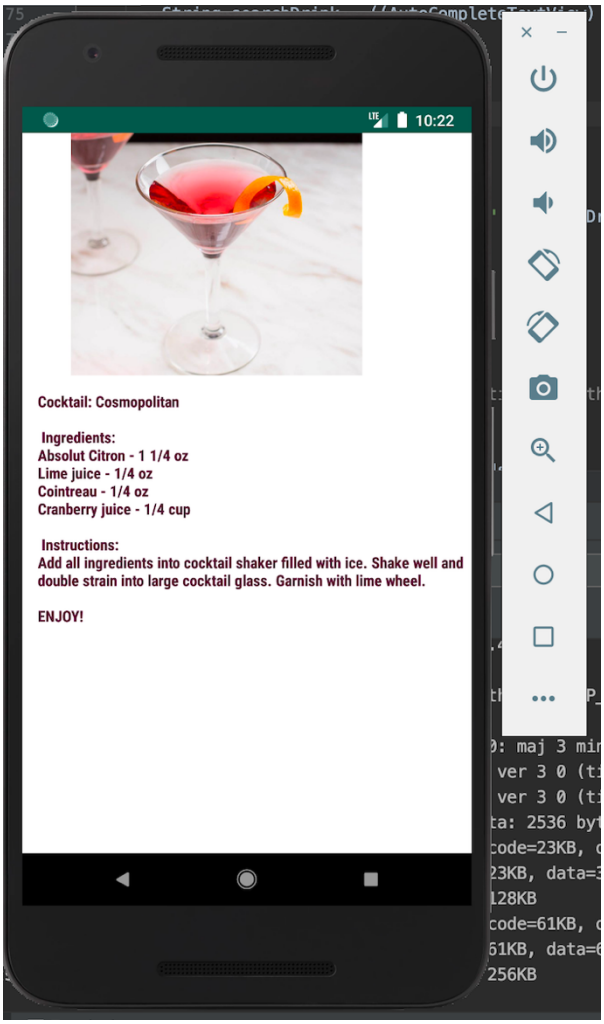
d. Receives and parses an XML or JSON formatted reply from your web service

{"drinks":[{"name":"Gin Fizz","instructions":"Shake all ingredients with ice cubes, except soda water. Pour into glass. Top with soda water.","ingredients":"Gin - 2 oz ,Lemon - Juice of 1/2 ,Powdered sugar - 1 tsp ,Carbonated water - null","image":"https://www.thecocktaildb.com/images/media/drink/xhl8q31504351772.jpg"}]}

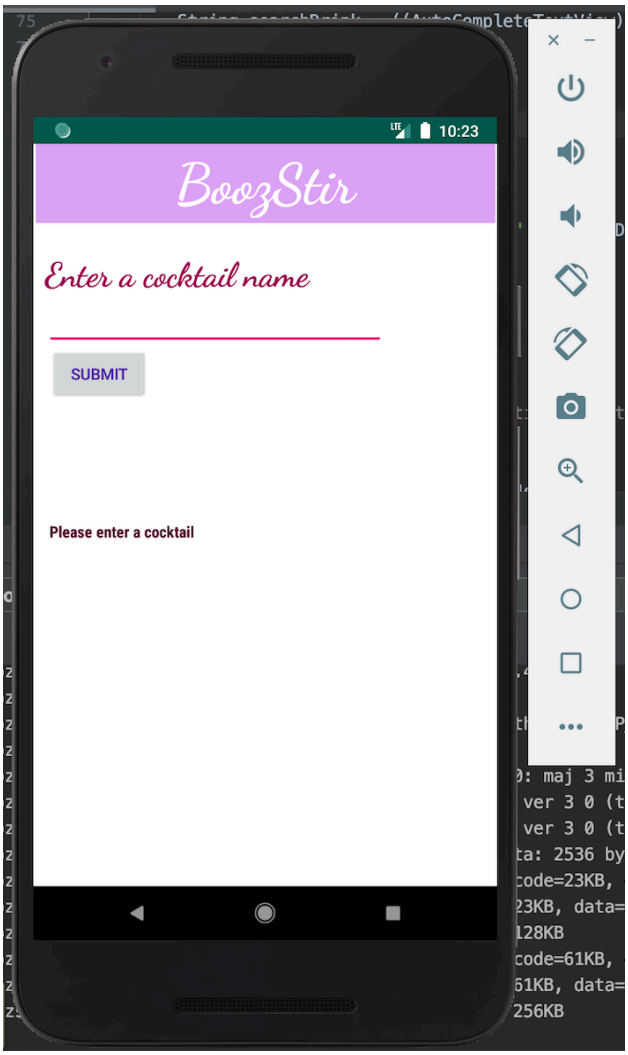
e. Displays new information to the user

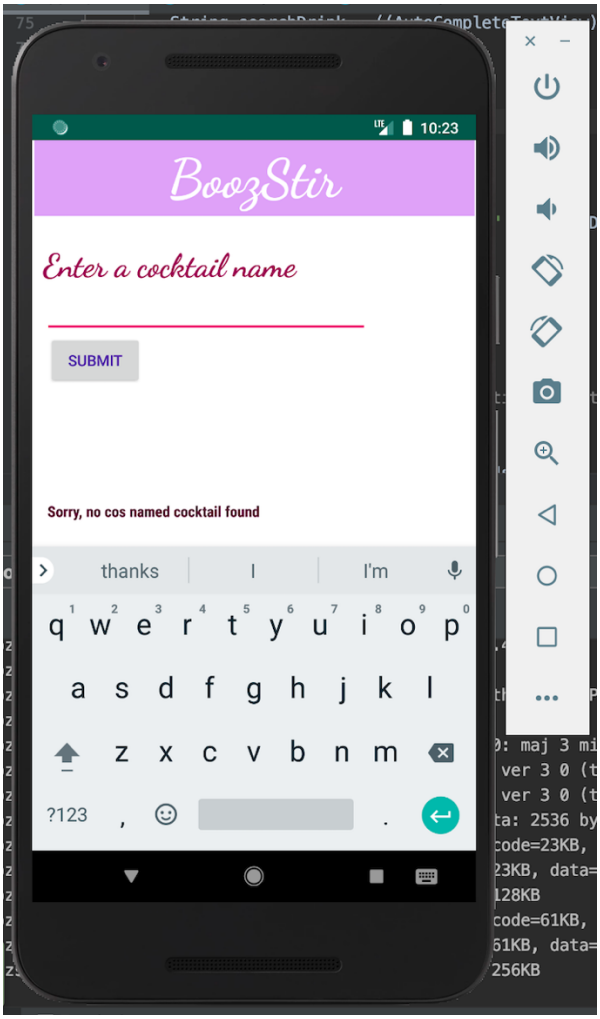
Screenshots displaying the information to the user





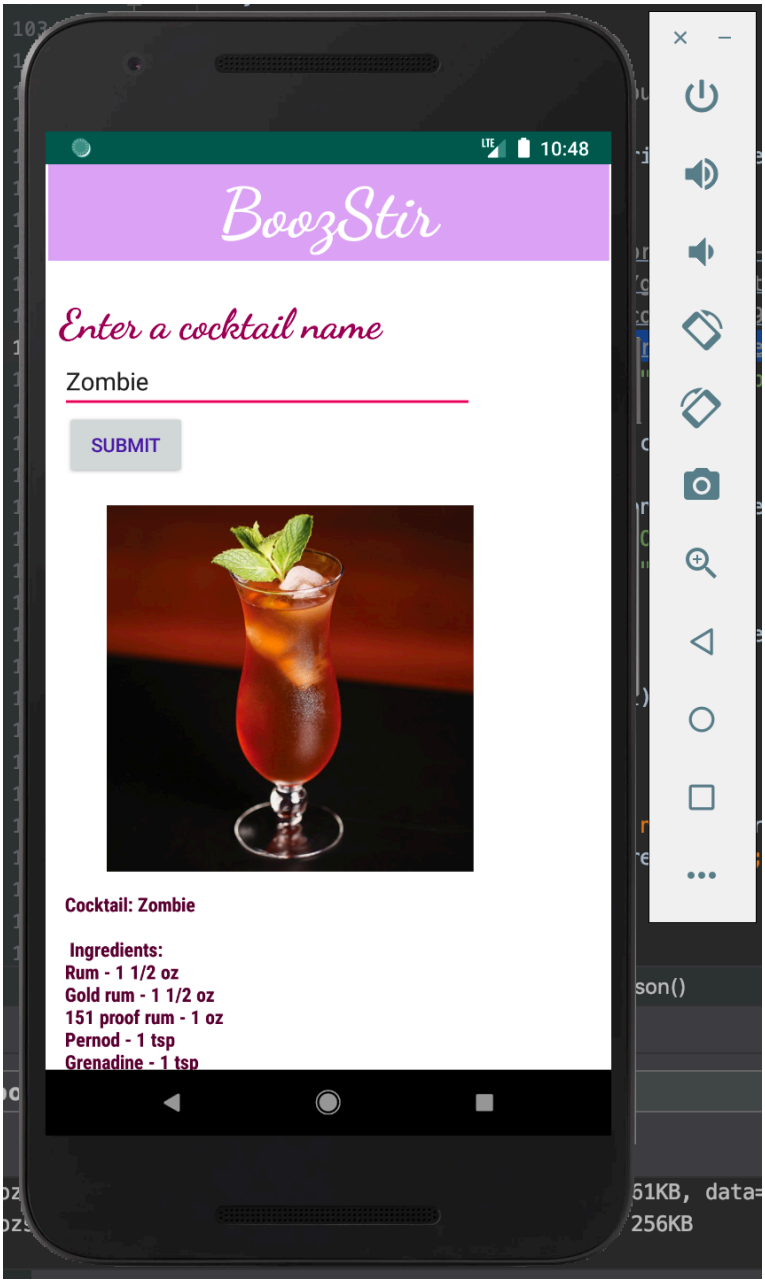
If user inputs nothing or wrong drink name, then following information displayed





f. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

It is repeatable after displaying either of the above examples, user can put a new cocktail name as Zombie and following is displayed



2. Implement a web service, deployed to Heroku

a. Using an HttpServlet to implement a simple (can be a single path) API. (It is recommended that you do not try to use JAX-RS / Jersey.)

The URL to web service deployed on Heroku: https://blueberry-sundae-22185.herokuapp.com/getMyCocktail

b. Receives an HTTP request from the native Android application

Project4Task1CocktailModel and Project4Task1CocktailServlet receive the HTTP request from the Android Application

c. Executes business logic appropriate to your application. This includes fetching XML or JSON information from some 3rd party API and processing the response.

Project4Task1CocktailModel fetches the Json from the 3rd party API and then formats the response into a Json String which sent as a response by the Project4Task1CocktailServlet back to Android App

d. Replies to the Android application with an XML or JSON formatted response. The schema of the response can be of your own design. Alternatively, you can adopt a standard schema that is appropriate to your application. (E.g. Common Alerting Protocol if your application deals with emergency alerts.)

Project4Task1CocktailModel fetches the Json from the 3rd party API and then formats the response into a Json String as below

{"drinks":[{"name":"Gin Fizz","instructions":"Shake all ingredients with ice cubes, except soda water. Pour into glass. Top with soda water.","ingredients":"Gin - 2 oz ,Lemon - Juice of 1/2 ,Powdered sugar - 1 tsp ,Carbonated water - null","image":"https://www.thecocktaildb.com/images/media/drink/xhl8q31504351772.jpg"}]}