Name: Muyu Tong

Andrew ID: muyut

# Project4Task1Writeup

# 1

## a

The app contains TextView and EditText (Input IP). It get an IP and gives back its location.


## b

The app asks the user to input an IP and then gives back the location.

## C

The LocateIP.class contains a getLocation function that will make a request to the server on heroku for the location.

```
public class LocateIP {
    MainActivity activity;
    String ip;
    Gson gson;
    String message;
    Location location;
    int code;

    public LocateIP(MainActivity activity, String ip) {
        gson = new Gson();
        this.activity = activity;
        this.ip = ip;
    }

    public void locate() {
        if (!ip.matches("^([0-9]{0,3}\\.){3}[0-9]{0,3}$")) {
            activity.setView(400, "Illegal ip format", null);
        } else {
            new BackgroundTask().start();
        }
    }

    private class BackgroundTask {
        // run the HTTP request task in a separate thread
        public void start() {
            new Thread(new Runnable() {
                @Override
                public void run() {
                    try {
                        getLocation(ip);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    activity.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            activity.setView(code, message, location);
                        }
                    });
                }
            }).start();
        }

        private void getLocation(String ip) throws IOException {
            //request the heroku to ask the IP
            System.out.println("get location");
```

```
            URL url = new URL("https://pacific-everglades-38530.herokuapp.com/ip-servlet/getIP?ip="+ip);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
            StringBuilder strB = new StringBuilder();
            String str;
            while((str = in.readLine())!=null) {
                strB.append(str);
            }

            //parse the response to Location if there is one
            Rep rep = gson.fromJson(strB.toString(), Rep.class);
            code = rep.getCode();
            if (code == 200) {
                location = rep.getLocation();
                System.out.println(location.toString());
                message = "success";
            } else {
                message = rep.getMessage();
            }
        }
    }
}
```

# d

As it is shown in the part c in the getLocation function. The response will be wrapped into Rep.class instance which also contains a Location.class object

The Rep.class and Location.class are shown as follow:

```
public class Rep {
    int code;
    Location location;
    String message;
    public Rep(int code, Location location, String message) {
        this.code = code;
        this.location = location;
        this.message = message;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public int getCode() {
```

```
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public Location getLocation() {
        return location;
    }

    public void setLocation(Location location) {
        this.location = location;
    }
}

```

```
public class Location {
    private String country;
    private String regionName;
    private String city;

    public Location() {};
    public Location(String country, String regionName, String city) {
        this.country = country;
        this.regionName = regionName;
        this.city = city;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }

    public String getRegionName() {
        return regionName;
    }

    public void setRegionName(String regionName) {
        this.regionName = regionName;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
```

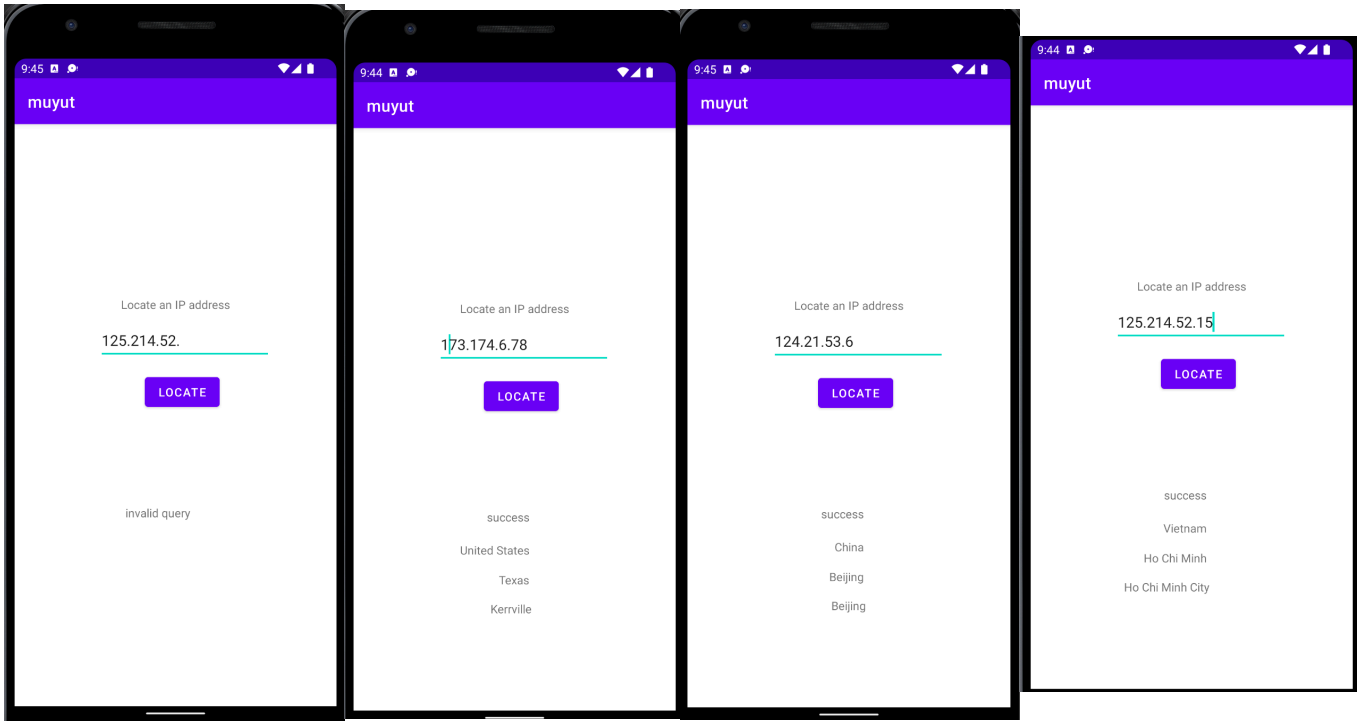
```
    }

    @Override
    public String toString() {
        return "Location{" +
                "country='" + country + '\'' +
                ", regionName='" + regionName + '\'' +
                ", city='" + city + '\'' +
                '}';
    }
}

```

## e

Here are some results for different IPs and invalid input.




## f

The user don't have to restart to run. They can just enter a new IP.

# 2

## a

The IPServlet receive an request, parse the IP in that request and use to request the third party API.

```java
//ref: https://stackoverflow.com/questions/5175728/how-to-get-the-current-date-time-in-java
//ref: https://www.mongodb.com/docs/drivers/java/sync/v4.3/quick-start/;
@WebServlet(name = "ipServlet", value = "/ip-servlet")
public class IPServlet extends HttpServlet {
    private Gson gson;
    private DateFormat dateFormat;
    public void init() {
        gson = new Gson();
        dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        queryIP(request, response);
    }


    public void queryIP (HttpServletRequest request, HttpServletResponse response) throws IOException {
        String query = request.getQueryString();

        //get the ip address
        if (query==null || query.split("=").length < 2) return;
        String ipAddress = query.split("=")[1];

        ResBody resBody = IP.locateIP(ipAddress);

        PrintWriter out = response.getWriter();
        out.println(gson.toJson(resBody));
        response.setContentType("text/html");
    }

    public void destroy() {
    }
}
```

```java
public class IP {

    public static ResBody locateIP(String ip) throws IOException {
        ResBody resBody = new ResBody();
        Gson gson = new Gson();

        if (ip.matches("^([0-9]{0,3}\\.){3}[0-9]{0,3}$")) {
            //if the ip format is valid, connect the api and get the result map
            URL url = new URL("http://ip-api.com/json/" + ip);

            //connect to the third party API
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
```

```
14              StringBuilder strB = new StringBuilder();
15              String str;
16              while((str = in.readLine())!=null) {
17                  strB.append(str);
18              }
19              in.close();
20              HashMap<String, String> result = gson.fromJson(strB.toString(), HashMap.class);
21
22              String status = result.get("status");
23
24              //when success, create the location object
25              if (status.equals("success")) {
26                  String country = result.get("country");
27                  String regionName = result.get("regionName");
28                  String city = result.get("city");
29                  Location location = new Location(country, regionName, city);
30                  resBody.setLocation(location);
31                  resBody.setCode(200);
32              } else {
33                  String message = result.get("message");
34                  resBody.setMessage(message);
35                  resBody.setCode(400);
36              }
37
38          }
39          else {
40              resBody.setCode(400);
41              resBody.setMessage("Invalid IP format");
42          }
43          return resBody;
44      }
45  }
```

## b

IPServelet that receive requests from users are shown in 2.a

## c

IP.class will request the third party API and parse the result. The code is shown in 2.a

## d

In IPServlet the response object is converted to json (line 26-28)

```
1  PrintWriter out = response.getWriter();
2  out.println(gson.toJson(resBody));
3  response.setContentType("text/html");
```

The response object only contains a code, a message and a location

```java
public class ResBody {
    int code;
    String message;
    Location location;

    public Location getLocation() {
        return location;
    }

    public void setLocation(Location location) {
        this.location = location;
    }

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```