

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation

```
1. Private void getCurrentLocation(){
2. LocationManager locationManager = (LocationManager)
   getSystemService (Context.LOCATION_SERVICE);
3.     if (ActivityCompat.checkSelfPermission(this,
   Manifest.permission.ACCESS_FINE_LOCATION)
4.     !=PackageManager.PERMISSION_GRANTED &&
   ActivityCompat.checkSelfPermission(this,
   Manifest.permission.ACCESS_COARSE_LOCATION)
5.     != PackageManager.PERMISSION_GRANTED) {
6.
7.         return;
8.     }
9. Location mylocation =
   locationManager.getLastKnownLocation(LocationManager.PASSIVE_PR
   OVIDER);
10.
11.         fromlatitude = mylocation.getLatitude();
12.         fromlongitude= mylocation.getLongitude();

13.     latilongiuser = new LatLng(fromlatitude,fromlongitude);
14. }
```

Rows 1 – 20 contain the creation of the current location search method. Rows 2 – 15 contain get permission to access location at current location, Location mylocation is filled current location get from locationManager. Rows 17-18 contain fromlatitude is filled with mylocation to getLatitude its get position Latitude current location, and fromlongitude is filled with mylocation to getLongitude for getting position Longitude current location. Rows 19 contain variable LatLng latilongiuser is filled with fromlatitude and fromlongitude.

```
15.     private void calculDistance() {
16.         urls="https://maps.googleapis.com/maps/api/distancematrix
   /json?
   units=imperial&origins="+fromlatitude+", "+fromlongitude+"&desti
   nations=-7.023636,110.403083&key=AIzaSyBGKxOSR_MTA7bT15oH2dR3H9
   sMuVSw7cQ";
```

```

17.     StringRequest caridistance = new
        StringRequest(Request.Method.GET, urls,
18.         new Response.Listener<String>() {
19.             @Override
20.             public void onResponse(String
                response) {
21.                 try {
22.                     JSONObject objectrow = new
                        JSONObject(response);
23.
24.                     JSONArray arrayrow =
                objectrow.getJSONArray("rows");
25.                     JSONObject objectElement =
                arrayrow.getJSONObject(0);
26.                     JSONArray arrayElement =
                objectElement.getJSONArray("elements");
27.                     JSONObject objectDistance =
                arrayElement.getJSONObject(0);
28.                     JSONObject objectJarak =
                objectDistance.getJSONObject("distance");
29.
                hasilJarak = objectJarak.getString("value");
30.                 hasilJarakConvert = Integer.parseInt(hasilJarak);

```

Rows 21 – 36 contain method to calculate distance from latitude and longitude with latitude and longitude destination. Rows 22 contain variable urls is filled name web for get Google api to calculate distance, to calculate distance needs from latitude, longitude and latitude, longitude destination. Rows 23 contain variable to get all the contain from calling the variable urls. Rows 28 – 34 contain the results of calculating the distance obtained from variable urls with different name variables like rows, elements, distance and the contain from result of distance in value. Rows 35 contain create variable hasilJarak to get result of calculate distance from variable urls. Rows 36 contain variable hasilJarakConvert is filled with variable hasilJarak convert to Integer from String parameter.

```

31.     int n = cekStatusArray.length();
32.
33.     arrayJarak = new int[n];
34.     int[] arraykodeantrian = new int[n];
35.     int[] arrayiduser = new int[n];
36.     String[] arraynama = new String[n];
37.     String[] statusUser = new String[n];
38.
39.     int temp = 0;
40.     int temp1 = 0;
41.     int temp3 = 0;
42.     String temp2 = "";

```

```

43.     String temp4 = "";
44.     int o = arrayJarak.length;

```

This illustration explain method LoadStatusAntrian for display all data like queue, quota, queue user and to sorting the queue user. Rows 37 – 43 contain variable preparation for sorting queue user and rows 45 – 50 same like rows 37 – 43.

```

45.     for(int i= 0; i<o; i++) {
46.         for (int j = 1; j < (o - i); j++) {
47.             if (arrayJarak[j - 1] > arrayJarak[j]) {
48.                 temp = arrayJarak[j - 1];
49.                 arrayJarak[j - 1] = arrayJarak[j];
50.                 arrayJarak[j] = temp;
51.                 temp1 = arraykodeantrian[j-1];
52.                 arraykodeantrian[j-1] = arraykodeantrian[j];
53.                 arraykodeantrian[j] = temp1;
54.                 temp3 = arrayiduser[j-1];
55.                 arrayiduser[j-1] = arrayiduser[j];
56.                 arrayiduser[j] = temp3;
57.                 temp2 = arraynama[j-1];
58.                 arraynama[j-1] = arraynama[j];
59.                 arraynama[j] = temp2;
60.                 temp4 = statusUser[j-1];
61.                 statusUser[j-1] = statusUser[j];
62.                 statusUser[j] = temp4;
63.             }
64.         }
65.     }

```

Rows 51 – 68 contain method for sorting the queue user, method sorting queue user used swapping.

## 5.2 Testing

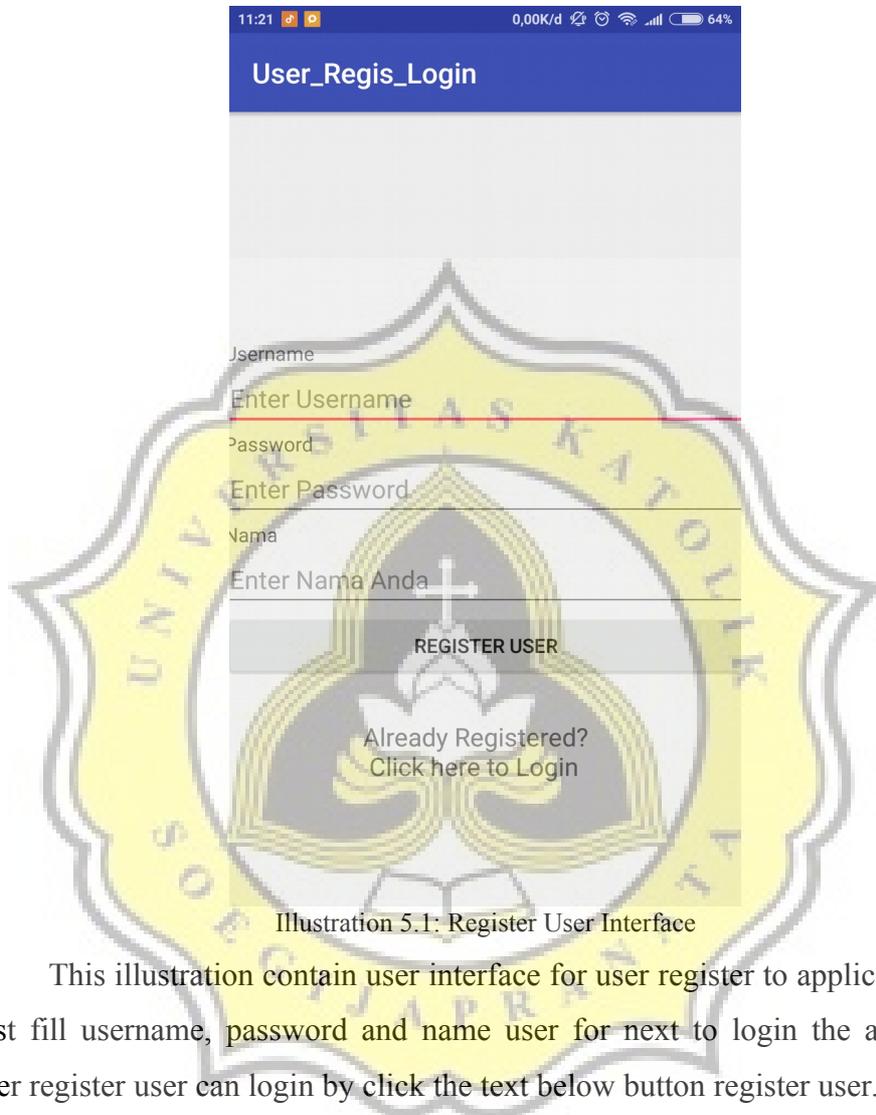


Illustration 5.1: Register User Interface

This illustration contain user interface for user register to application. User must fill username, password and name user for next to login the application. After register user can login by click the text below button register user.

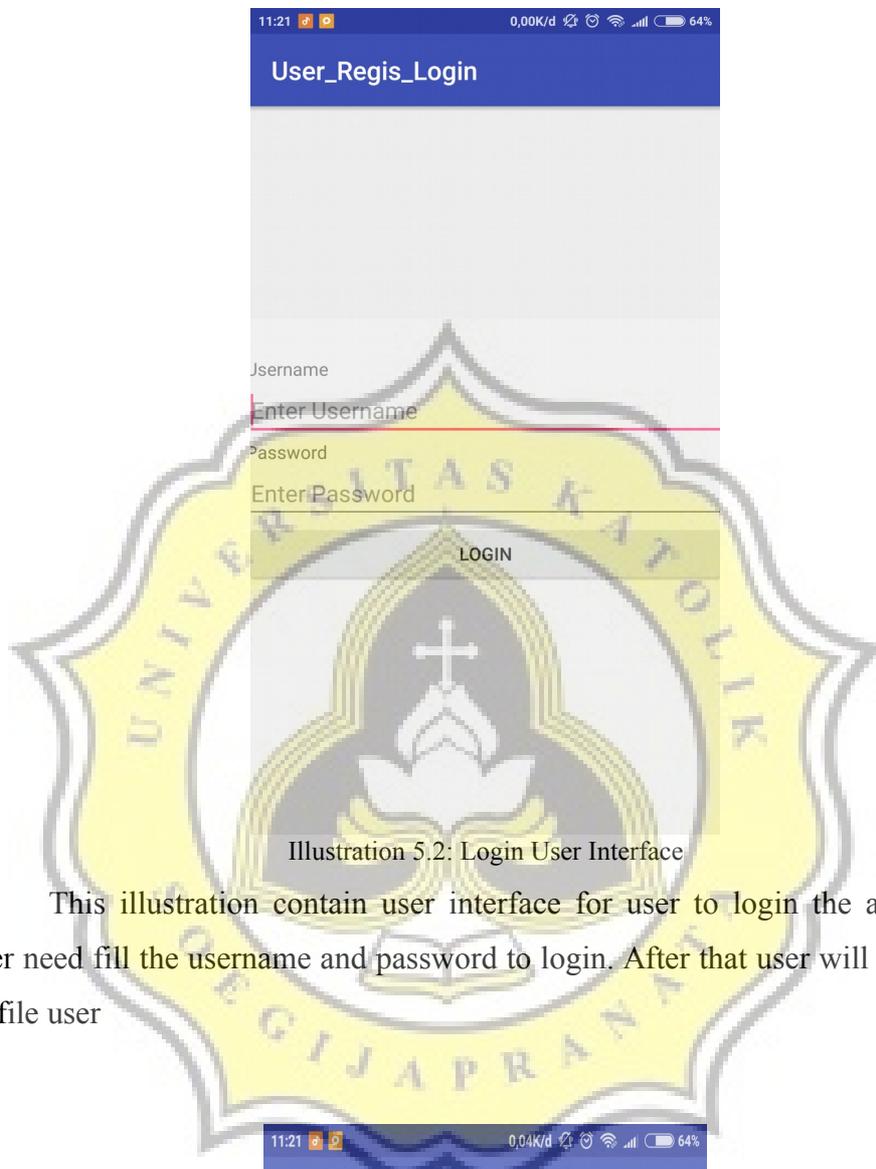


Illustration 5.2: Login User Interface

This illustration contain user interface for user to login the application. User need fill the username and password to login. After that user will be enter to profile user

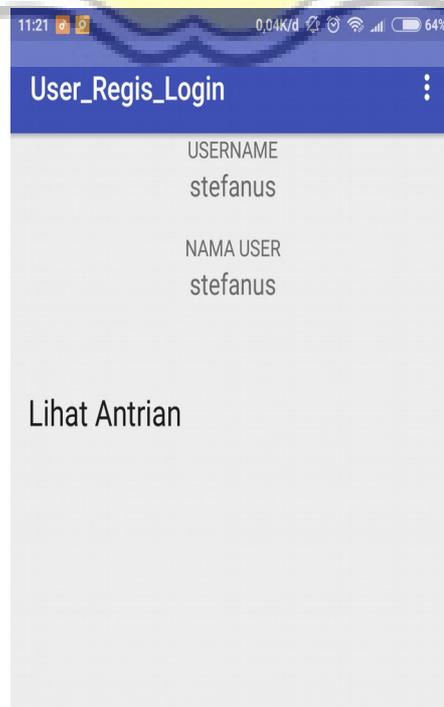


Illustration 5.3: Profile User Interface

This picture contain user profile with display username user and name user. In this interface user can do two things. There is look queue user and logout by clicking button at top right corner.

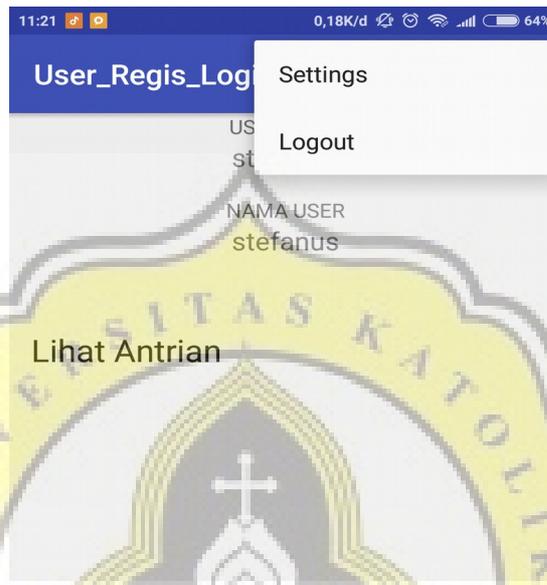


Illustration 5.4: Menu Logout User Interface

Button at the top right corner for user can logout from the application.



Illustration 5.5: Queue User Interface

In this queue user interface, user can queue, cancel the queue and refresh the queue when user move to other place and check the number of queue. When user queue, application ask user to get user location and calculate the distance from the user location to the the destination location. The user will get the queue number after the application calculates the distance from the user it is queuing up. So the application will display the user queue number in the user's application display and display the queue number of the user which is currently being served. If the user want to cancel user queue, the application will delete user queue number from display user queue number. If the user going to the destination, the user should press button refresh to refresh the queue number on the application

This application used data structure array to swap the queue number user. If the user press button refresh every time when user make a moves. User will swapping the queue number according to users distance.

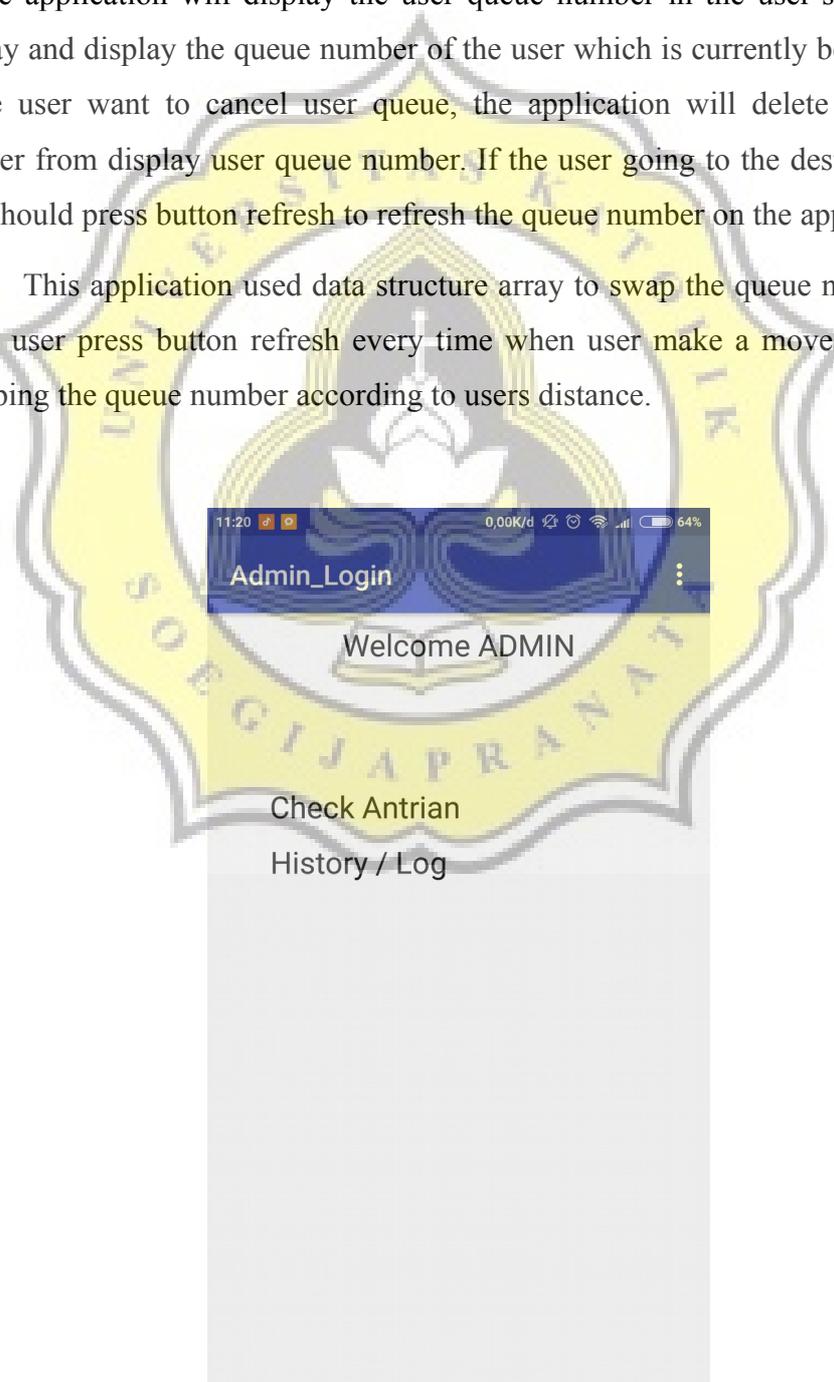


Illustration 5.6: Admin Profile Interface

In this illustration, admin of this application can check the queue user or check the user who has been previously queued up.



Illustration 5.7: Admin Check User Queue Interface

At this illustration admin can see all the user queue and monitoring the user queue number. If the admin ready to open the queue, admin just press panggil and then the number 1 of the queue number will be called and into the current queue. After the admin finished serving the user then admin just press the button selesai, if the admin is ready to serve other users then just pressing the button panggil again. Every a few minutes admin must press button refresh to refresh the queue number of users, to know who will be next call to enter the current queue.

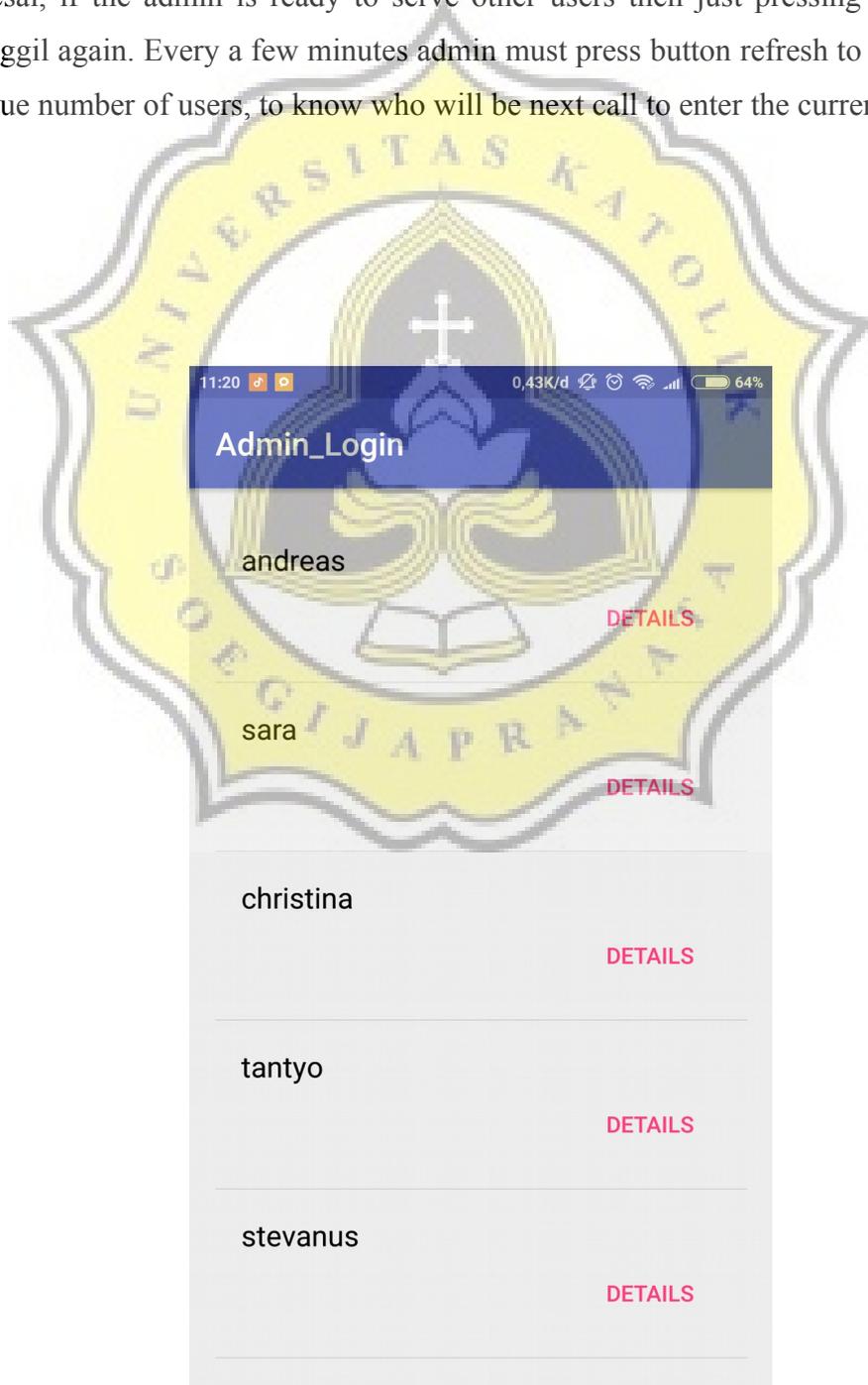


Illustration 5.8: Admin History User Interface

This illustration explain about the history of user who ever queued up. Each user will be recorded by name if it has been queued once. If the admin wants to know the completeness of the data from the user, by pressing the details text besides users name.

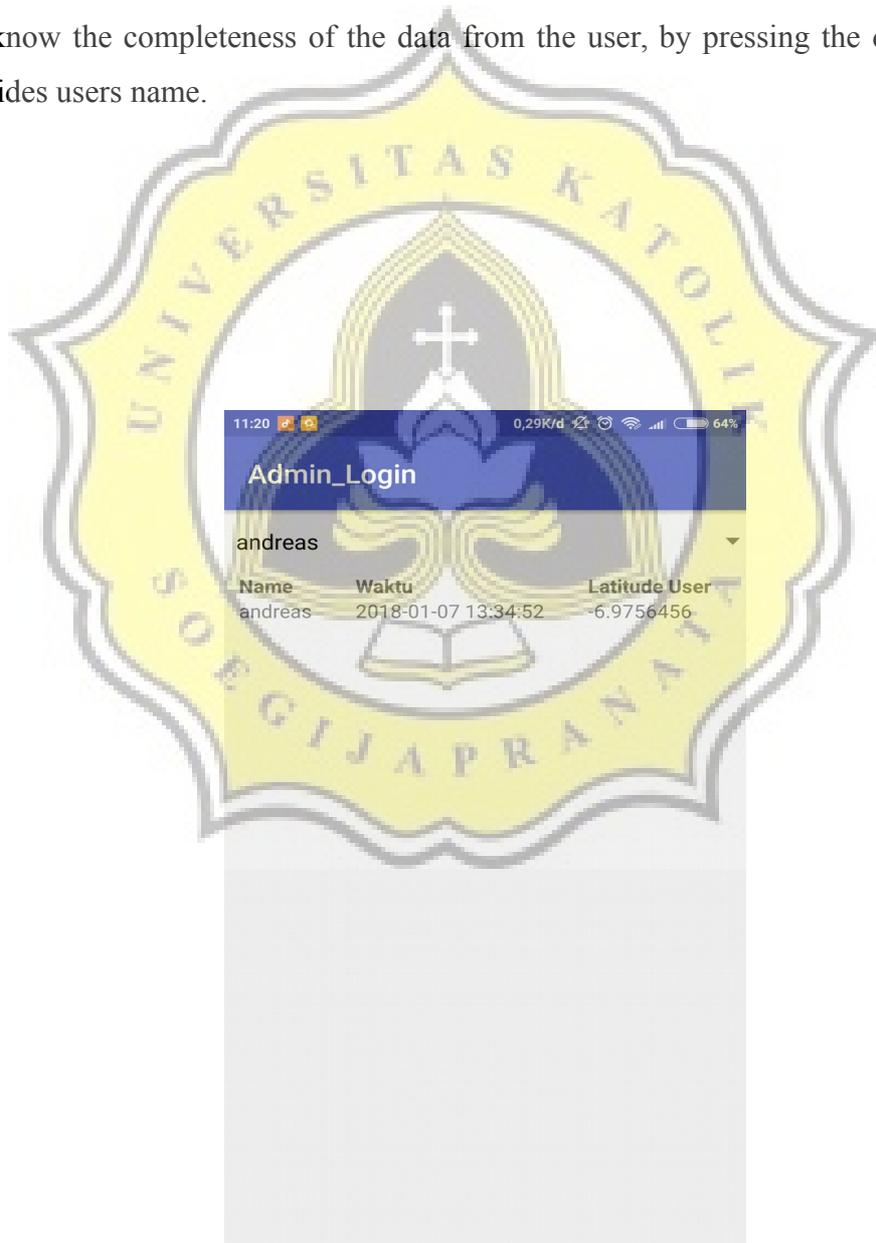


Illustration 5.9: Details User Queue Interface

In this interface, admin can look all the details about user after user make queue to the application. There is drop down button at the name of users, admin can choose many user to see the detail of users.

