

BADAL(WEATHER APP)

M.K. Liza¹, Ashish Angadi², Bhaskar Chowdhury³

¹ Assistant Professor ,Department of Information Technology, SRM University, Chennai
India

² Under Graduate Student ,Department of Information Technology, SRM University, Chennai
India

² Under Graduate Student ,Department of Information Technology, SRM University, Chennai
India

ABSTRACT

With the View ever changing climate and unaccounted rain it has been impossible to be certain of current day weather. The order of the day and schedule gets disrupted and daily chores come to a standstill. In such events a small notification on your phone can help. The Weather Application presented will not be different from others. However, it will be better. The application not only provides the user with current weather and forecast but also provides the users with variety of new options. This Application will notify the user if and only if there is a forecast of heavy rains or showers. However, the application will also notify you in the events of news related to your current location. The various tabs include

Now- The current weather forecast with maximum accuracy. Gives a graph of events in last 24 hours.

News - Directs you to the news and weather related headlines near your location.

Traffic Route - The Application uses Traffic detection which in turn leaves the user with option of time in hand and best possible route to work or home.

The Notification leniency -

As for most applications the notifications are either permanent or floating . Which leaves user irritated and constant beeping of the phone pocket dials as the phone switches on on receiving a notification.

Keyword: - JavaScript, JSON, API, Database

1. INTRODUCTION

A weather app is one of the most rarely used application yet it finds a place in every android or iOS device. Coming to the roots of it, the android OS finds a place in almost 60% of the device in the entire world. Globally speaking, 3 out of 5 people are android users picked at random. It is an extremely huge market with very high potential and jurisdiction. With a high reach like this, android programming falls into the category of programmer's attraction. A weather application is also an in built application but the amount of usage of that application is lesser than the usage of a radio apps these days.

With a view of limited users in huge market, its is quite certain to reckon the fact that this product might not reach places, however what matters is how the product is presented. Not many apps are marketed the way they should be done but then this application offers way more than what a ordinary application will ever. The design and user

interface is not made fancy or mind-bending but is simple and clean, because the focus of this application is not towards attracting the users towards the design of the application but to create a measure of maximum safety. Which forms the basis of a weather application.

1.1 Objective

The main objective of this mobile application is to provide users a weather application with some brand new features which are not included in any other weather app but ours. This weather app will change the experience of the users forever. The app will not only give you weather reports but will also provide some extra features which will help users at times of need.

1.2 Organization of the report

The report is divided into 4 parts and each part deals with the different aspects of the system.

- (i) **Design:** This part talks about the existing system, how they are designed and the issues associated with them. Furthermore, it describes the features of the system proposed and the requirements for operating it.
- (ii) **Module Description:** This part describes each module implemented in the system, i.e., how the data is processed in each and what are the steps involved from the user's point of view.
- (iii) **Implementation:** This part deals with an overview of the platform for which the system is developed for. It also talks about the parameters needed for running the system and provides a sample of code used, along with screenshots of the output.
- (iv) **Conclusion:** This part concludes the report and discusses the possible enhancement that can be implemented in the future improve the quality.

1.3 Challenges Overcome

The user is verified via Single Sign-on method with Google, Facebook verification method. The guest Sign-in will therefore have only weather forecast privileges. With respect to intensity of rainfall or climatic changes, the panic button's alert message will be accordingly increased. The Maps API is provided by Google and Weather API by certified Weather provided also used in ongoing inbuilt applications.

2. LITERATURE REVIEW

This specific project is not only the most easy topic but also rarely used. For instance there exist more than 15 Weather Applications and only 3 have managed to climb the cliff of 100k Downloads. With the market so low, entering into this was indeed difficult and the greatest risk. As Steve Jobs once said "There is no such thing as a simple project, make it complex", just what I did here. With every review, every guidance, every step I have reached one step closer to what is now a baby. The project, the Application in itself is enough explain and need no marketing at any point of time. Application stands for itself. It is indeed very important to take care of oneself, no one ever can ever predict rain, nor a storm. Climatic change is subject to risk. In interest of social service and user convenience that this app is perfect of a person who is climate conscious and doesn't use his/her phone well. This app will stand out from others for sure. For almost whole review I have referred to the project as the app and not Badal because it is not published yet and the main goal stays undone. For every possible hurdle is overcome now stays beta Testing and Debugging after which the project will be done with. This stands to reason how this idea came to mind that, I, myself have been a part of the Chennai floods and I can't imagine how it was to witness a disaster like this, This application is a tribute to Chennai Floods '16. This application in full force can save atleast 20% of people, Purpose of this project is served and we are happy. There will be no ads and glitches for this project is totally non profit and every penny invested is sort of a donation.

If you were to shop on Amazon , would you rather

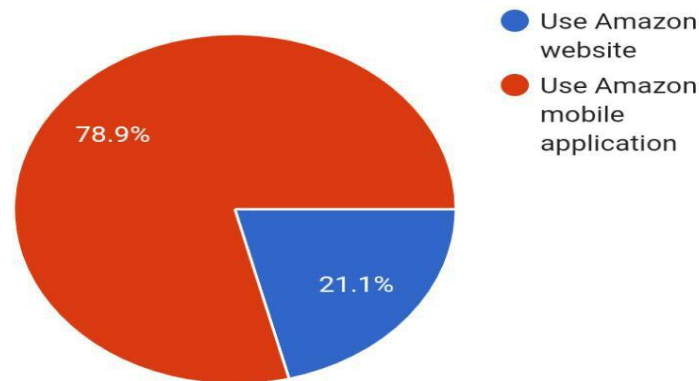


Chart -1(pie chart): user's preference to mobile application over web application

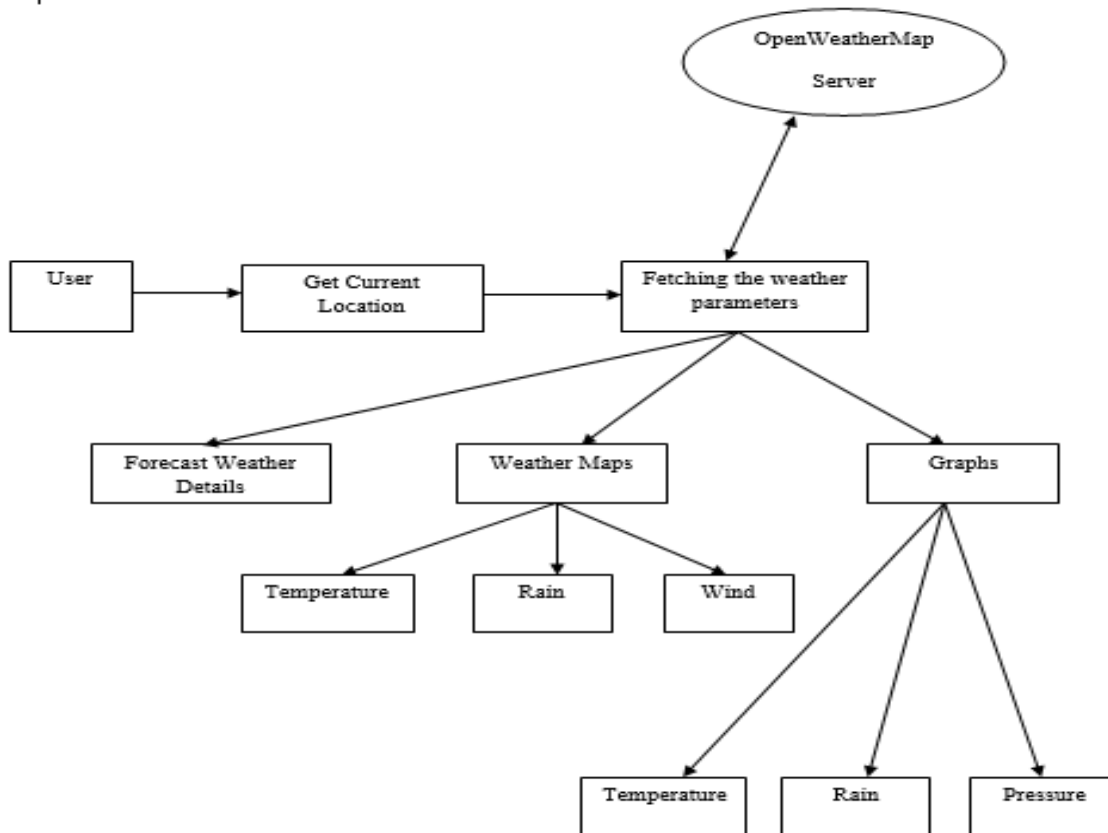
2.1 Existing System

The existing weather applications are AccuWeather , NBC , Weather so on and so forth. These applications are purely weather applications and do not provide any safety measures nor do they actually do anything. They're either informative or provide facts and figures of temperature and air-pressure. Although, NBC provides new which gives weather updates from all over the world.This was a new revolution in the market. However the age old problem of notifications. The application irritated so much with their notifications, furthermore the application even had permanent notifications adding on to the already existing problems. With that being said, the application have exquisite User Interface and powerful design and the real competition is between the design and User Interface and how the weather is presented and no scope for innovation. This application is what will change the whole perspective of people looking at a weather application.

2.2 Proposed System

Thousands of apps are released every day, stable as well as with bugs. This weather app has features and new implementations as follows: i). Light weighted ii). Support multiple units iii). Support for widget iv). Simple design and user interface v). Detailed graphical representation Our App is totally based upon the OpenWeatherMap Server. The data is provided by the OpenWeatherMap to users by the means of API key. The API key is used to provide the security to cloud as only authorized user can access the weather information from the cloud. To extract the information from the API key we use JSON (JavaScript Object Notation) Object Android concept.JSON object holds key/value pair. Each key is represented as a string in JSON and value can be of any type. The cloud is updated continuously over the time which is more accurate as compared to previous approaches.

2.3 Application Architecture



2.4 Future Enhancements

- The offline notify service could improve with instant sharing options.
- The traffic route would be made App-Specific and generate a overall climate map whilst it is showing the fastest and convenient.
- The User Interface will improve as per requirement.

2.5 Features Of The Application

There are many great features but some of the features are worth mentioning:

- Map access from within the application to get shortest route to destination in case of heavy downpour
- Panic button is provided in-case of emergency
- News can be accessed from within the application which gives real-time weather updates pertaining to only the user's location.
- Application runs foreground and not background so no unnecessary and annoying news updates.
- No background running so no battery drainage.
- Speedometer to notify user in the events of rainfall and speed of the user is above 60 km/hr (specific vibration pattern).

3. CODE

```
package com.sai.badalapp;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import com.google.android.gms.auth.api.Auth;

import com.google.android.gms.auth.api.signin.GoogleSignInAccount;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.gms.auth.api.signin.GoogleSignInResult;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.SignInButton;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthCredential;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.GoogleAuthProvider;

public class LoginScreen extends AppCompatActivity {

    SignInButton signInButton;

    FirebaseAuth mAuth;

    private final static int RC_SIGN_IN = 2;

    GoogleApiClient mGoogleApiClient;
```

```
FirebaseAuth.AuthStateListener mAuthListener;
```

```
Button button;
```

```
@Override
```

```
protected void onStart() {
```

```
    super.onStart();
```

```
    mAuth.addAuthStateListener(mAuthListener);
```

```
}
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_login_screen);
```

```
    signInButton = (SignInButton) findViewById(R.id.googlebtn);
```

```
    mAuth = FirebaseAuth.getInstance();
```

```
    button = (Button) findViewById(R.id.button);
```

```
    button.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View view) {
```

```
            startActivity(new Intent(LoginScreen.this, GuestSignIn.class));
```

```
        }
```

```
    });
```

```
    signInButton.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View view) {
```

```
            signIn();
```

```
        }
```

```
    });
```

```

mAuthListener = new FirebaseAuth.AuthStateListener() {
    @Override
}
};

GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).requestIdToken(getString(R.string.defa
ult_web_client_id)).requestEmail().build();

mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this /* FragmentActivity */, new GoogleApiClient.OnConnectionFailedListener() {
        @Override
        public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
            Toast.makeText(LoginScreen.this,"Its not me, its the weather",Toast.LENGTH_LONG ).show();
        }
    })
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .build();
}

Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
startActivityForResult(signInIntent, RC_SIGN_IN);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

}

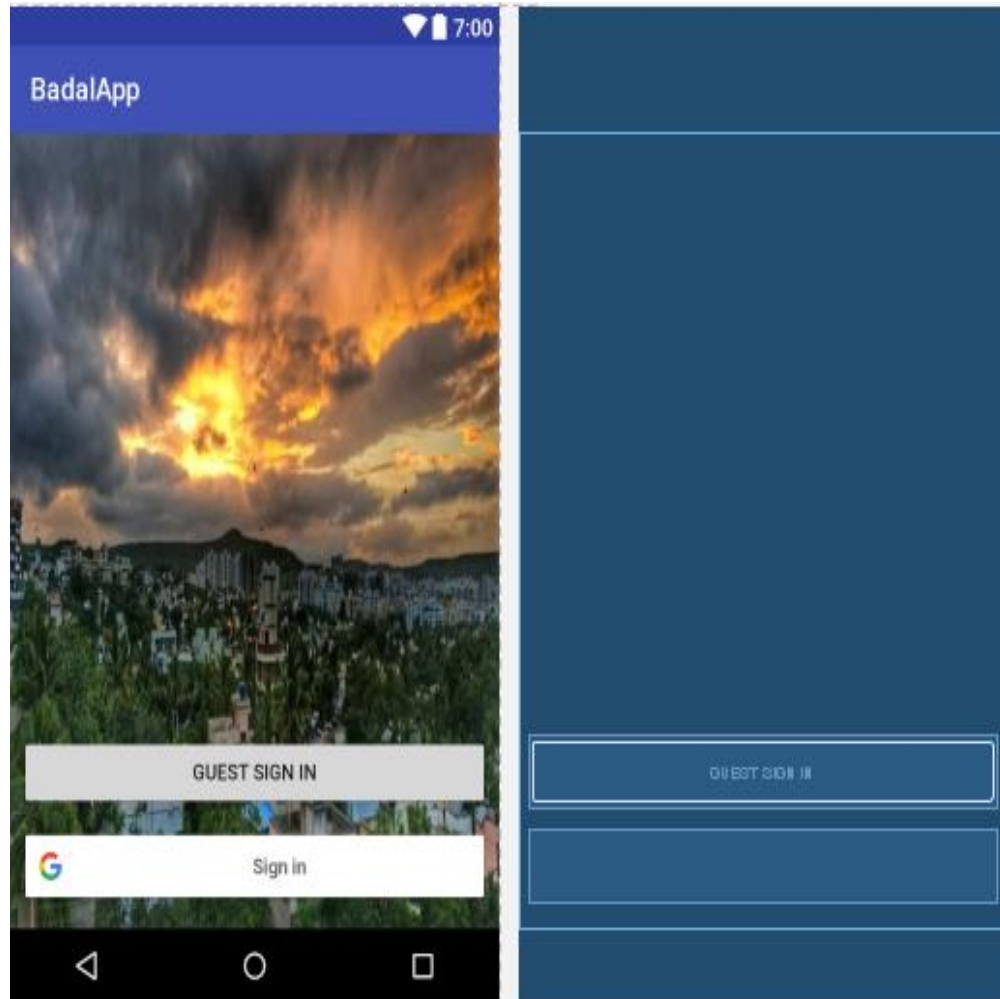
// ...

```



```
}  
});  
}  
}
```

4. SCREENSHOTS



5. CONCLUSION

- In this given time, Weather Applications are too many. But the Weather application is a daily used but least used application.
- If any application irritates the user, the user uninstalls it right away.
- A User-Friendly Application with less intervention and more use stands the sole purpose of the application.

6. ACKNOWLEDGEMENT

We, Bhaskar Chowdary and Ashish Angadi have dedicated our time heart and soul for this project. Still, it would be unfair not to mention a few names as without their constant support and backing, this would not have been what it is now. Therefore I would like to extend my sincere gratitude to all of them.

First of all I would like to thank our Guide of project Mrs.Liza for her sincere attention, guidance and impeccable teaching ability and also my HOD Dr.T.K Thivakaran for providing valuable insight and expertise.

In the end we thank SRM University, Ramapuram campus for providing the freedom and resources to pursue the project we wanted to.

7. REFERENCES

- [1]. Google Maps API, <https://developers.google.com/maps/documentation/android-api/start>.
- [2]. OpenWeatherAPI, <https://openweathermap.org/api>.
- [3]. Application Fundamentals, <https://developer.android.com/guide/index.html>.

