**ANDROID WEATHER APPLICATION**

**A Project Report**

***Submitted by:***

**ABHINAV SINGH (1421506)**

**MAYANK (1421553)**

**MASROOR AHMAD (1421552)**

**CHHOTU KUMAR (1421524)**

in partial fulfilment for the award of the degree

of

**BACHELOR OF TECHONOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

at

****

**SHAHEED BHAGAT SINGH STATE TECHNICAL CAMPUS**

**MOGA ROAD ,NH 95,PUNJAB (INDIA) -152004**

**(AFFILIATED TO I. K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, JALANDHAR, PUNJAB )**

**(Formerly known as Punjab Technical University)**

**May, 2018**

**DECLARATION**

I hereby declare that the Major Project Report entitled **Android Weather App** is an authentic record of our own work as requirements of Major Project during the period from Jan,2107 to May,2018 for the award of degree of B.Tech. (Computer Science & Engineering), Shaheed Bhagat Singh State Technical Campus, Ferozepur, affiliated to I.K. Punjab Technical University, Jalandhar, Punjab(India), under the guidance of Mr. Navtej Singh Ghumman.

**(Signature of student)**

**Abhinav Singh**

**1421506**

**Place:**

**Date:**

It is to be certified that the above statement made by the student is correct to the best of our knowledge and belief.

#### Signatures

#### Internal Examiner:

#### 1. 2. 3. 4.

External Examiner: Signature  
 Name:

**CERTIFICATE**

This is to certify that the project titled **“Sunshine”** is the bonafide work carried out by Abhinav Singh(1421506), Mayank(1421551), Masroor Ahmad(1421552) and Chhotu Kumar(1421524) a student of B.Tech(CSE) of Shaheed Bhagat Singh State Technical Campus, Ferozepur, affiliated to I.K. Punjab Technical University, Jalandhar, Punjab(India) during the academic year 2018, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology(Computer Science and Engineering ) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature **Mr. Navtej Singh Ghumman (Project Guide)**

**Place:**

**Date:**

**ACKNOWLEDGEMENTS**

We take this opportunity to express our profound gratitude and deep regards to our tutor Mr. Navtej Singh Ghumman and Ms. Khushjeet Kaur for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We am obliged to staff members of SBSSTC, for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Lastly, We thank almighty, our parents and our classmates for their constant encouragement without which this assignment would not have been possible.

**ABSTRACT**

Our aim for this six Major Project was to learn how to create an application in android, the Google operating system for mobile devices. The design of such an application is made in slightly modified Java.

The purpose of the final project was to build a cloud-connected Android app, and understand the tools, principles, and patterns that underlie all Android development. The aim of the project was to make the students understand the challenges associated with developing for the mobile environment (and how to overcome them), learn how to build a great user experience for Android devices, and apply this knowledge for building awesome Android applications in future.

The idea behind the “Sunshine” project was to simply create a cloud based weather android app with a user-friendly interface which will provide the user with the weather forecast of next 2 weeks for a particular city (based upon the city id entered by the user).

The “Sunshine” project was created using agile approach methodology using Android Studio i.e. this project was divided into small chapters (stages) like creating project Sunshine, connecting to the cloud, creating new activities and navigate app with intents, activity lifecycle and data, content providers , hooking it up with loaders, rich & responsive layouts and services & notifications.

This Major Project allowed us to increase our knowledge in Java, a language with which we had many difficulties. Also going through different stages of developing Sunshine helped us to earn a lot of information about the difficulties being faced by a programmer during android app development.

**TABLE OF CONTENTS**

**Chapter 1 Project Profile**

1.1 Project definition…………………………………………………………………………………...

1.2 Scope and objective of project……………………………………………………………………..

**Chapter 2 System Study and Problem formulation**…………………………………………………………

2.1 Data Analysis……………………………………………………………………………………….

2.2 Proposed System……………………………………………………………………………………

2.3 Advantages of proposed System……………………………………………………………………

2.4 Feasibility study…………………………………………………………………………………….

2.4.1 Technical……………………………………………………………………………………...

2.4.2 Operational……………………………………………………………………………………

2.4.3 Economical…………………………………………………………………………………...

2.5 System requirements………………………………………………………………………………...

2.6 object oriented analysis……………………………………………………………………………...

**Chapter 3 Project Plan**………………………………………………………………………………………..

3.1 Programming languages……………………………………………………………………………..

3.2 Development tools…………………………………………………………………………………...

3.2 Object oriented design……………………………………………………………………………….

3.3 Reuse of existing software components……………………………………………………………..

**Chapter 4 structured analysis and structured design**……………………………………………………….

4.1 data flow diagrams…………………………………………………………………………………..

4.2 Flowchart/algorithm used……………………………………………………………………………

**Chapter 5 Detailed Design**…………………………………………………………………………………….

5.1 Design Strategy……………………………………………………………………………………...

5.1.1 Major modules…………………………………………………………………………………..

5.1.2 Sub modules…………………………………………………………………………………….

5.2 Module Design………………………………………………………………………………………

5.3 Interface Design……………………………………………………………………………………...

**Chapter 6 Testing and Implementation**……………………………………………………………………...

6.1 System Testing……………………………………………………………………………………...

6.2 Test plan…………………………………………………………………………………………….

6.2.1 Functional, performance and stress test………………………………………………………..

6.3 Summary of results obtained

**Chapter 7 Project legacy**………………………………………………………………………………………

7.1 Current status of project……………………………………………………………………………

7.2 Remaining areas of concern………………………………………………………………………..

7.3 Technical and managerial lessons learnt…………………………………………………………...

7.4 Future recommendations…………………………………………………………………………...

Bibliography

**Chapter 1**

**Project Profile**

* 1. Project Definition

Sunshine is a cloud based weather android app with a user-friendly interface which will provide the user with the weather forecast of next 2 weeks for a particular city (based upon the city id entered by the user). The Sunshine app provides the user with a detailed weather forecast for a particular city which involves the minimum and maximum temperature along with different factors like humidity, pressure and wind speed (with directions).

Features:

* Weather forecast for next 14 days. Including high/low temperatures, wind speed, atmospheric pressure and humidity level.
* Ability to share the forecast with your friends and family via your own messaging/email apps like messenger, whatsapp , gmail etc.
* See the location you are getting the weather for on the map.
* Get a notification once a day about the weather information for that day for easy access.
  1. Scope & Objective

The purpose of the final project was to build a cloud-connected Android app, and understand the tools, principles, and patterns that underlie all Android development. The aim of the project was to make the students understand the challenges associated with developing for the mobile environment (and how to overcome them), learn how to build a great user experience for Android devices, and apply this knowledge for building awesome Android applications in future.

**Chapter 2**

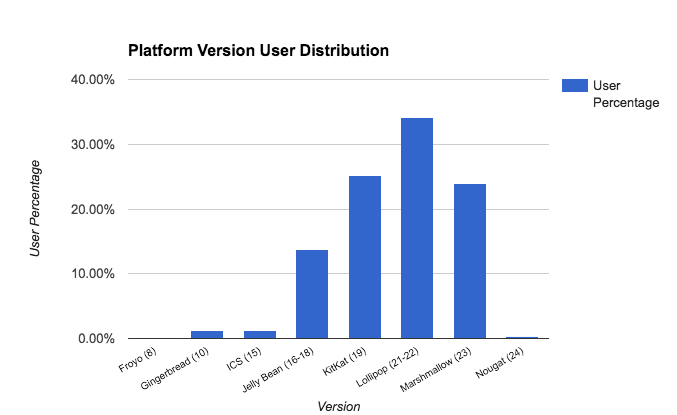
**System Study and problem formulation**

2.1 Data analysis

* Android min and target versions

Android 1.0 launched in 2008, and in just the 8 years since - there have been **13** new major platform releases. By convention, each release is named off of a sugary treat, and the releases are named alphabetically.

One can almost see a vaguely bell shaped curve, with the oldest releases on the left, their popularity dropping off as devices are upgraded or replaced. The largest proportion of devices are in the middle, representing devices about 2 years old. The newest platforms, which gain in popularity as new phones are released or updates go out, are on the right. On the basis of this data analysis the minimum sdk version of “Sunshine” was set to Jelly Bean (Api 16-18) so that our app must work on maximum android devices .



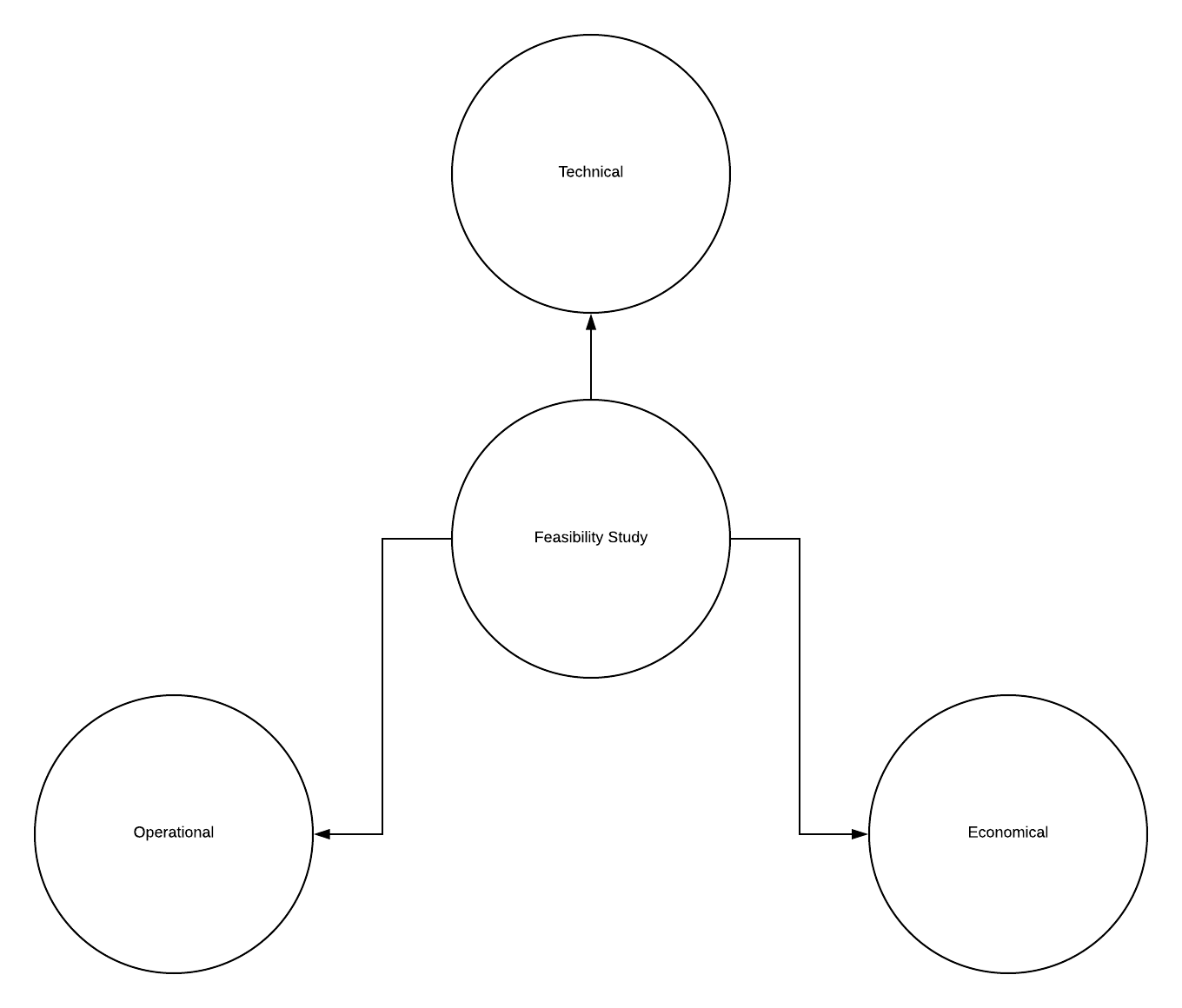
2.2 Proposed System

The proposed system which would be a compact weather app is a user friendly app which will keep the user updated with the weather forecast . The information and notification can be easily accessed by one touch on the android application. The application provides a live notification on the daily basis for the users who sets the notification ON .

2.3 Advantages of proposed system

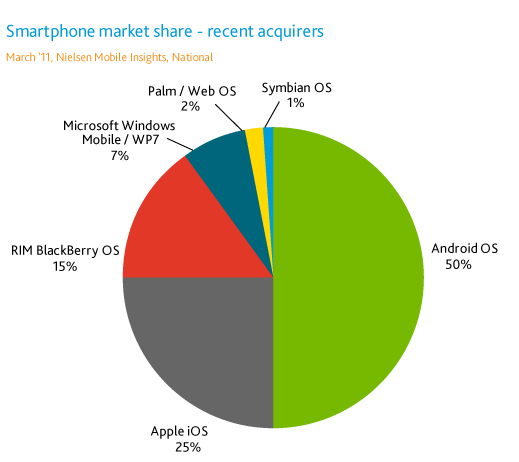
* The application will be available for the public free of cost on the Google Play Store.
* The application will provide the user with a detail view of the daily weather along with the factors like the humidity, maximum and minimum temperature, pressure , wind speed etc.
* User friendly interface.
* Easy to use.

2.4 Feasibility study

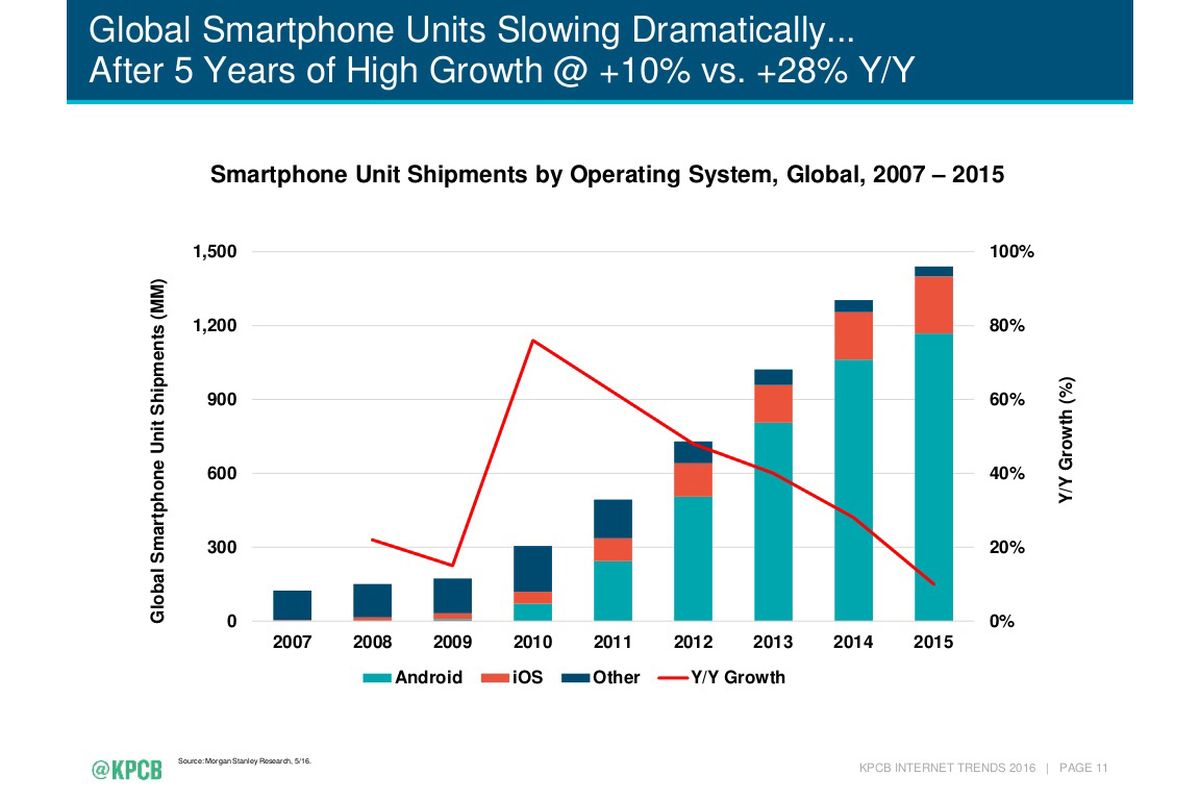


2.4.1 Technical

* A weather application can be developed for many mobile platforms like iOS, Windows , Symbian, Android etc , but since Android is one of the most used mobile OS , hence it is one of the most appropriate platform in order in order to reach more and more people.



* The technologies used in this project are JAVA for core development of project, XML for designing of the application and interactive user interface and SqLite for database creation and management. All these technologies are still emerging and has a future scope.
* Weather API has been used as a framework for this project. The API used in the project is provided by openweathermap.org.
* All the weather data is fetched from the server of openweathermap.org in Json format.
* The concept of Json parsing is used which is implemented in ForeastAdapter.java file in the project.
* Two database tables are used: one for storing the information of the location and other for storing the information of the details of weather forecast.
* The application requires the permission for accessing the internet and the user’s permission will be required for accessing an application which has the map’s services for determining the user’s location on the map.
* For the debugging purposes the testing of the database, URI , location is important which is implemented in the TestDb.java, TestUri.java, TestUtilities.java and TestWeatherContract.java files.



2.4.2 Operational

* The project is implemented using the Android studio software provided by Google.
* Android Studio is the official Integrated Development Environment (IDE) for Google’s Android operating system built on JetBrain’s Intellij IDEA software and designed specifically for android development.
* In android the data can be stored in two ways :

1. Shared Preferences: Android provides many ways of storing data of an application. One of this way is called Shared Preferences. Shared Preferences allows to save and retrieve data in the form of key,value pair. In order to use shared preferences, we need to call a method getSharedPreferences() that returns a SharedPreference instance pointing to the file that contains the values of preferences.
2. SQLite database: SQLite is a open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features. In order to access this database, we don't need to establish any kind of connections for it like JDBC,ODBC e.t.c

* Android Studio is a replacement for the Eclipse Android Development tools (ADT) as primary IDE for native Android application development.
* In this project the data is stored using SQLite since using key value pairs for storing the weather and the location data won’t be an efficient way to store the data .
* In SQLite the data can be stored in a properly organised and efficient manner in the form of rows and columns .
* All the weather data is collected from the server of openweatermap.org and stored in the database.
* The data regarding the location of the user is also stored in the database depending upon the user input of the loctaion id.
* The concept of inner join is implemented for merging of both the location table and the weather data table.

2.4.3 Economical

* For the basic functionalities of the weather API a free API key has been provided by the openweathermap.org which has been implanted in this project.
* For exploring and implementing more advanced features for a weather application like the analysis of the weather data for a particular time period, hourly update of the minimum and maximum temperatures etc, further paid API keys are provided by the openweathermap.org.
* The purchase cost of Google developer console which is required for uploading android applications on the Google .
* The application will be available free of cost for public use on Google Play Store.

2.5 System Requirements

The minimum system requirements for “Android Weather Application” is Jelly Bean. Since Android supports backward compatibility and the target version is Marshmallow (API 23) so “Sunshine” works properly on all Android devices except the first two versions of Android.

Sunshine only requires the permission to access the internet in a user’s device and Internet permissions falls under the category of normal permissions so, it won’t ask the user to grant access to any private data .

The user will simply need to install the .apk file in his/her Android device and must use the app over wifi or mobile data.

A **marshmallow** is a sugar-based candy or type of sweets that in its modern form typically consists of sugar, water and gelatin whipped to a squishy consistency, molded into small cylindrical pieces, and coated with corn starch. Some marshmallow recipes call for eggs. This is the modern version of a medicinal confection made from *Althaea officinalis* the marshmallow plant.

2.6 Object-Oriented Analysis

Create Project

Create new activities and navigate apps with intents

Content providers

Activity lifecycle and data

Connect Sunshine to the cloud

Rich and responsive layout design

Hooking it up with loaders

Services & notifications

2.6.1 Create Project

Creating Sunshine project in Android Studio and discovering what makes mobile, Android in particular a unique environment for app development.

2.6.2 Connect Sunshine to the cloud

Connecting Sunshine up to an Internet on back –end . Learning how to add permissions, initiate network i/o, and move time consuming tasks off UI thread using AsyncTask().

2.6.3 Create new activities and navigate apps through intents

Structuring “Sunshine” and create complex and separate activities for each funtion(.java files) for the purpose of connecting the app to Internet using , fetching data from the sever, json parsing, using SQLite for storing forecast’s data in the form of table, for showing the data in the form of lists as shown in the fig. Below.

2.6.4 Activity lifecycle and data

Finding out how the android framework manages the activity lifecyle and how it differs from what you might expect and dive into the world of persistent storage.



Fig. Android Activity lifecycle

2.6.5 Content providers

Learning how to create databases, using Android’s content providers to provide an abstraction layer between our data and our UI implementation.

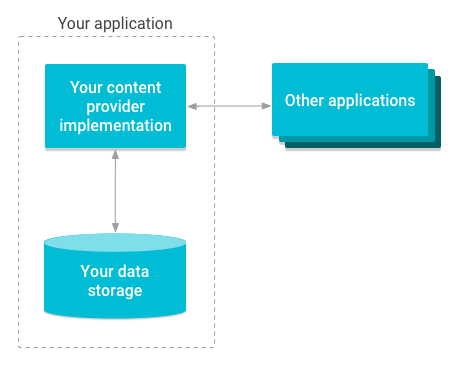


Fig. Content Provider

2.6.6 Hooking it up with loaders

Learn how to use loaders to asynchronously load data.

2.6.7 Rich and responsive layouts

Creating rich and responsive user interfaces that work across a variety of different hardware types and screen sizes.In order to create an outstanding UI design, it is essential for app developers to first segregate android mobile devices into various categories (buckets). Below is an illustration that depicts various categories of mobile devices based on their screen size and display.

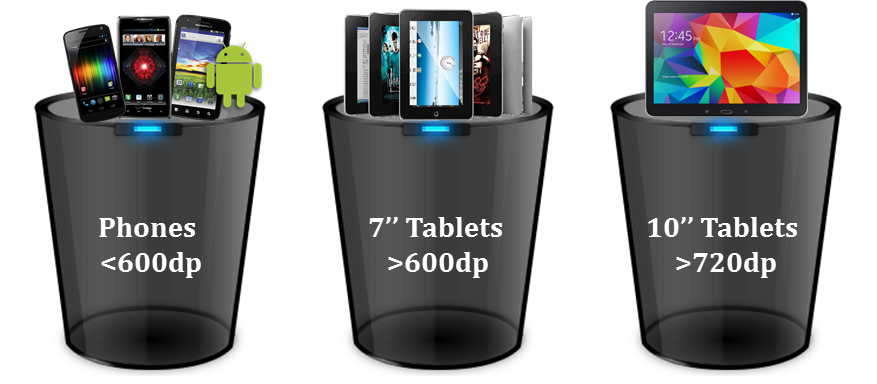


Fig. Different screen size devices

2.6.8 Services and notifications

Learning how the Android framework manages background apps and how to use Services & Notifications to make “Sunshine” active when it is not in the foreground.

**Chapter 3**

**Project plan**

**3.1 Team Structure**

3.1 Programming languages

1. JAVA for creating the functional units of “Sunshine”.
2. XML for layout design.
3. SQLite

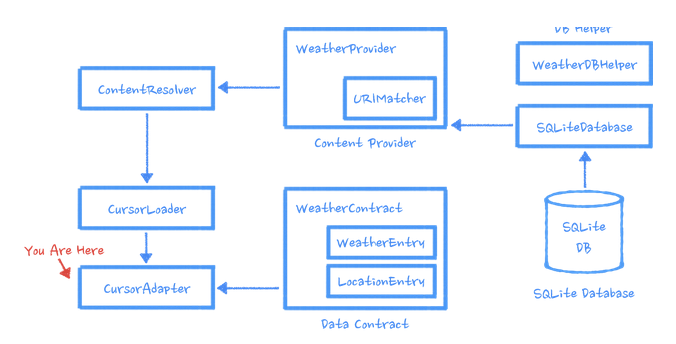
3.2 Development tools

1. Android Studio
2. Gradle
3. ListView & adapters
4. Threading and AsyncTask
5. HTTP requests, web APIs
6. Android permissions
7. App navigation with explicit intents
8. Implicit intents to incorporate 3rd party apps
9. Broadcast intent receivers
10. Activity lifecycle
11. Creating & using content providers
12. JUnit tests
13. Loaders & asynchronous loading of data
14. Fragments
15. Background services
16. Rich notification
17. Efficient data transfer with SyncAdapters
18. Custom Views
19. Json parsing
20. Android design principles

3.2 Object Oriented design

The input for object-oriented design is provided by the output of [object-oriented analysis](https://en.wikipedia.org/wiki/Object-oriented_analysis_and_design).

1. In Sunshine we used content providers to help us efficiently sync the data from the internet combined with utilities in the Android framework.
2. Using a content provider simplifies getting database content to & from our user interface.
3. We implement a content provider by extending the content provider class. In “Sunshine” we called this the WeatherProvider class to read & write the database.



3.3 Reuse of existing software components

The different modules of the “Sunshine” can further be used in order to fulfil the similar function requirements of the app such as the FetchWeatherTask() in the ForeCastFragment.java is used for json parsing and this method can be used in other apps for similar purpose.

The XML code for creating the ListView & the method used in java for populating the ListView can be used for other apps like a movie advertisement app which will show the images of the movie poster in the ListView.

**Chapter 4**

**Structured Analysis & Structured Design**

4.1 Data Flow diagrams

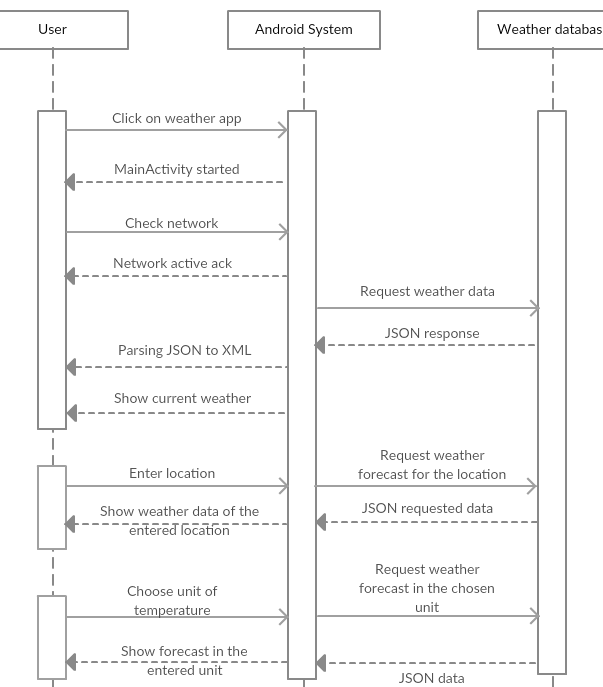


Fig. Data flow diagram

4.2 Flowchart/algorithm used

Services & notifications

Responsive layout design

Hooking it up with loaders

Content providers

Activity lifecycle & data

Navigate apps with intent

Connect to Internet

Create project

Fig. Flowchart (“Sunshine”)

**Chapter 5**

**Detailed Design**

5.1 Design Strategy

5.1.1 Major modules

Sunshine has three major modules :

1. Module 1 : In this module the user will see the forecast data of the default location (Moscow) at first in the as he opens the app for the first time on his/her device.
2. Module 2: The user can now go to the settings option in the action bar and can enter the desired location and unit of the temperature in which he wants to see the weather forecast.
3. Module 3: The user can know the weather forecast without even opening the weather the “Sunshine” app with the help of live running notifications if he/she enables the notifications in the settings menu.

5.1.2 Sub modules

1. As the user clicks on a day’s forecast a it will lead the user to the details activity where a detailed weather of the day is provided along with the humidity, pressure, wind speed along with the max & min temperatures and a separate weather depending upon the day’s weather.
2. In the details activity the user can share a day’s forecast data with his family or friends by simply clicking on the share icon through any messaging app like facebook, gmail, whatsapp etc.
3. In the home screen of the app the user can see the weather forecast of two weeks.
4. In the homescreen today’s weather forecast is shown highlighted at the top followed by next 13 days forecast.
5. In the list of wether forecast each day’s forecast is shown along with max & min temperatures in the right hand side and a weather icon is shown in the left hand side depending upon that day’s weather.
6. The user can watch his location in the map by simply clicking on the “Map Location” in the action bar.

5.2 Module design

**Chapter 6**

**Testing And Implementation**

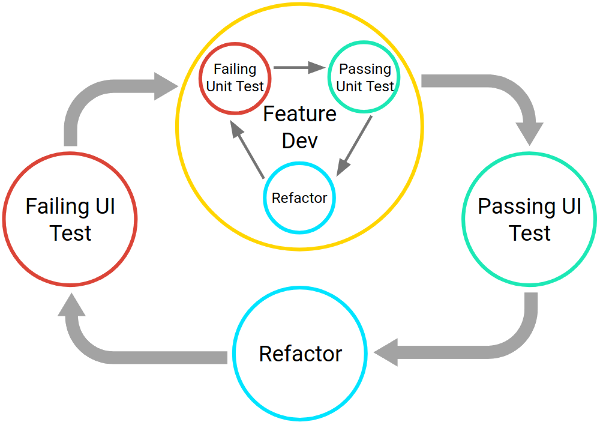
6.1 **System testing**

Users interact with Sunshine app on a variety of levels, from pressing a  button to view the details of weather to naviagate between multiple apps like Google Maps onto their device. Accordingly, we should test a variety of use cases and interactions as you iteratively develop our app.

As our app expands, we might find it necessary to fetch data from openweathermap.org server, interact with the device's sensors, access local storage, or render complex user interfaces. The versatility of our app demands a comprehensive testing strategy.

When developing a feature iteratively, start by either writing a new test or by adding cases and assertions to an existing unit test. The test fails at first because the feature isn't implemented yet.

It's important to consider the units of responsibility that emerge as we design the new feature. For each unit, we write a corresponding unit test. Our unit tests should nearly exhaust all possible interactions with the unit, including standard interactions, invalid inputs, and cases where resources aren't available.

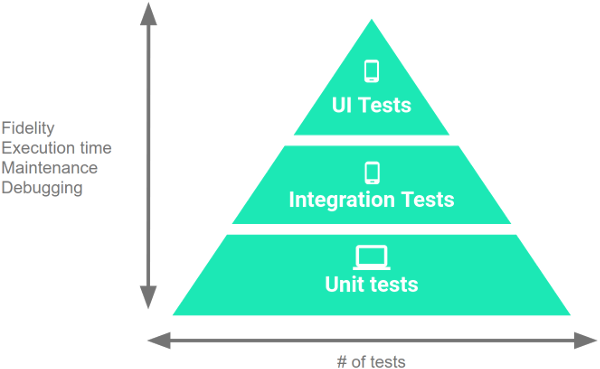


**Figure .** The two cycles associated with iterative, test-driven development

The full workflow, as shown in Figure , contains a series of nested, iterative cycles where a long, slow, UI-driven cycle tests the integration of code units. You test the units themselves using shorter, faster development cycles. This set of cycles continues until your app satisfies every use case.

Testing Pyramid

The Testing Pyramid, shown in Figure , illustrates how your app should include the three categories of tests: small, medium, and large:



**Figure,** Testing pyramid

Small tests are unit tests that you can run in isolation from production systems. They typically mock every major component and should run quickly on your machine.

Medium tests are integration tests that sit in between small tests and large tests. They integrate several components, and they run on emulators or real devices.

Large tests are integration and UI tests that run by completing a UI workflow. They ensure that key end-user tasks work as expected on emulators or real device.