

SOFTWARE DESIGN SPECIFICATION

NAME:PRASHANTH MURALI

ID:1211418361

PROJECT : WINTERHASCOME(WEATHERAPP)

/**

*** Copyright © 2016 by Prashanth Murali. Permission is granted to use, modify, and distribute this document.**

***/**

SOFTWARE DESIGN DOCUMENT

1 Introduction

This document is meant to describe the features of weatherapp. So as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other. The weatherapp gathers weather data of the required location from the openweathermap api and displays the details on the screen. The weather details are obtained as a JSON file. This document provides an in depth review of the internal functioning and design of the app.

1.1 Objective

The primary goal of weatherapp is to provide accurate and instant weather updates as and when required by the user for the user's desired location. The app in itself is very simple, fetching the weather data from openweathermap servers when requested. The UI of the app is very simple, consists of the weather data and a change city button to change city.

1.1.1 Search City and Display

The weatherapp is used to provide weather data such as temperature, sunrise time, sunset time, wind direction, current conditions for the location of the user's choice. The app allows the User to enter the location i.e town or city for which they require the weather data. The app is designed to perform the data collection operations in the background method which reduces the load on the target android device.

1.2 Project Overview and Scope

The weatherapp is used to provide weather data such as temperature, sunrise time, sunset time, wind direction, current conditions for the location of the user's choice. The app allows the User to enter the location i.e town or city for which they require the weather data. The app is designed to perform the data collection operations in the background method which reduces the load on the target android device.

1.2.1 Core Features

The android application will have the following core features.

- **Android Device with Android V 6.0.1 or higher**

The app requires the user to operate an android device with android V 6.0.1 or higher in order to function.

- **Openweathermap Api**

The app requires the use of a weather api such as Openweathermap to obtain the weather details from and parse through.

- **Access to the Internet**

The weatherapp cannot function if the device is not connected to the internet.

- **Display**

The app must display all the weather information obtained from the each JSONObject appropriately.

- **Change Location**

The app must include a button to enable the user to request weather data for a location of their choice.

3.2 Technical Issues

The weatherapp works on a client server architecture where the user's device acts as the client and the Openweatherapp api works as the server and handles user requests. In case of a Server shutdown, the app will not be able to retrieve data from the server.

1.2.2 Additional Features

The features discussed below are

Performance Requirements

The product is expected to perform optimally with no errors or significant reduction in performance at any load of operation. The onbackground method of the AsyncTask ensures that the major operations are performed in the background.

GUI Requirement

The GUI consists of several TextViews to display the various weather condition details. An ImageView is used to download the weather icons and display them along with the other details. The user can provide input and search for weather details of the city they desire.

Security Requirements

The app does not have any security requirements as it does not handle any safety critical data. It obtains the data from the Openweatherapp servers and displays it on the screen.

Reliability

The weatherapp consumes very little memory and hence has very little toll on the internal RAM of the user's android device. Thus even when using a large amount of memory and under stress, the app would be able to run as it consumes minimal amount of RAM(20MB). All the weather data is obtained from the Openweathermap database which is exceptionally reliable. All coding standards and Javadoc documentation standards are followed in developing the app which makes it easily maintainable. The app can be converted to previous sdk versions with relative ease which makes it very portable. The functioning of the app is very trivial and the user interface is simple and easy to understand.

Maintainability

All coding standards and Javadoc documentation standards are followed in developing the app which makes it easily maintainable.

Portability

The app can be converted to previous sdk versions with relative ease which makes it very portable.

Availability

All the weather data is obtained from the Openweathermap database which is exceptionally reliable and has an average downtime of 0.2% hours per week.

1.3 Software Context

The weatherapp will be available for all phones running Android 6.0.1 or higher. The weatherapp is used to provide weather data such as temperature, sunrise time, sunset time, wind direction, current conditions for the location of the user's choice.

1.4 Major Constraints

One of the chief limitations for the implementation of the weatherapp is the difficulty in finding reliable servers that provide weather data.

1.5 Intended Audience

The intended audience of this piece of documentation is the management or stakeholders with knowledge about computer jargons and obviously to developers.

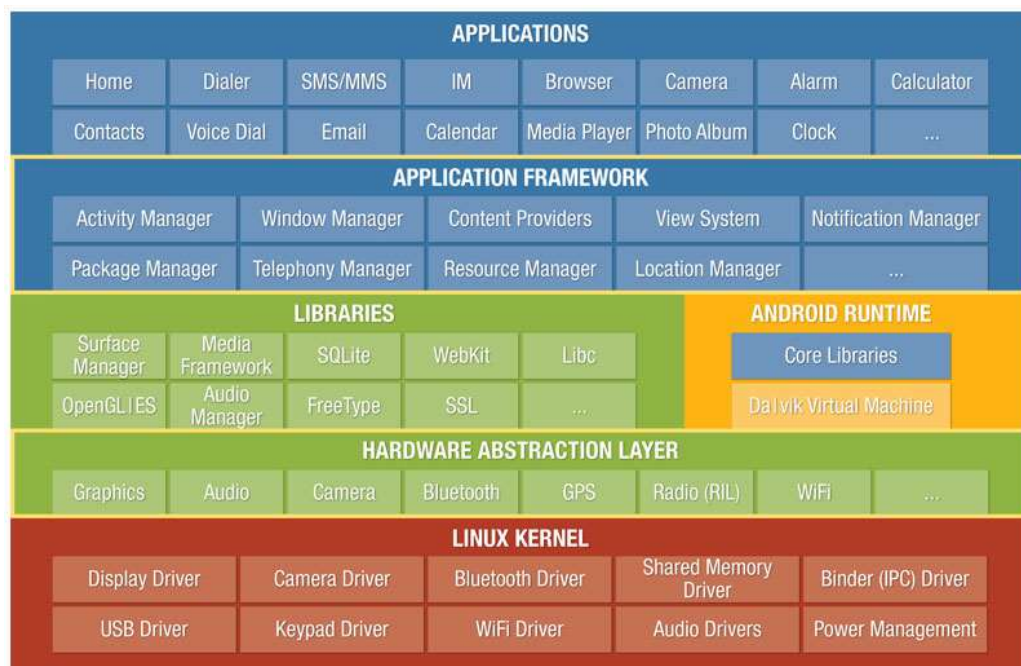
2 Architectural and Component Level Design

2.1 System Architecture

The weatherapp will be developed using java, JSON, android-XML. Client-Server Architecture is used.

2.1.1 Justification

The architecture of the System is represented below:



2.1.2 UML Diagrams

