Software Design Document for the Weather Data Application Programming Interface (API)

Version 1.1

Mutesasira Jovan

Kasozi Mulondo Moses

Kyebambe Sarah

Okwii Stanley

Table of Contents

Table of Figures

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of the Weather Data Application Programming Interface (API). It stipulates details of how the Weather Data API will be implemented. The document consists of a narrative and graphical models of the software design for the API[1]. This document is intended to encompass all the design details of the software implementation. It is also useful for background reading for anyone involved in developing or using the weather data API.
The API shall provide an abstraction of WDR weather observation data without direct access to the underlying database. It shall facilitate development of applications that shall use the Weather data, process it and provide it to the weather modelling tools and Message switching system at National Meteorological Center.

1.2 Scope

Weather Data Repository (WDR)[2] is an existing system which receives data from both the Automatic Weather Station (AWS) and manual weather stations across Uganda. The proposed project aims at improving the existing system through the following ways: -

- Create an API that will retrieve weather data from WDR and avail it to the developers interested in building applications that require the use of Uganda weather data, this data is represented in JSON format.

- Develop an application that shall format the API data into Table Driven Code Format (TDCF) and transmit it to message switching system at National Meteorological Centre (NMC), Entebbe through the use of emails and short messaging services (SMS) from where this data is transmitted to the World Meteorological Organization (WMO) regional Centre in Nairobi.
- Develop an application which shall format the API data into Little_R format that can be ingested by the weather modeling tool to make weather predictions.

Objectives

- To implement an API that provides an abstraction of the WDR data in JSON format.
- To develop an application that uses the API to send WDR data to the weather modelling software.
- To develop an application that uses the API to format and send the WDR data to the Entebbe NMC message switching system.

Goal

- Facilitate the access of local weather data by the weather modelers, Uganda National Meteorological center and other developers who may require to use this data in developing their applications.

Benefits

- The API shall promote innovation around the weather data without necessarily having access to the database.
- National Meteorological Center (NMC) will receive weather data in TDCF, a standard format that is recommended by World Meteorological Organization (WMO).
- The weather modelers will have access to a repository of Little_R files created from the local weather data that will be used to make improve the current estimates of weather forecasts.

1.3 Overview

The SDD is divided into sections and subsections. The sections include the following;-

Introduction.

This section provides introductory information related to this document (e.g. purpose, overview, scope, terms, definitions etc.)

System overview

This section provides a general description of the functionality, context and design of the Weather Data API.

System architecture

This section describes the modular program structure and explains the relationships between the modules to achieve the complete functionality of the system. It also contains architectural design, decomposition description and design rationale of the system.

Data design

This section provides detailed information about the data structures, data dictionary and data descriptions.

Component design

This section looks at what each component does in a more systematic way.

Human interface design

This section describes how users will interact with the weather data API. It contains screen mockups with detailed procedures user perform to achieve a specific task.

Requirements matrix

This section contains a table that lists each requirement and tracks the disposition of each requirement throughout the project life cycle.

1.4 Reference Material

[1] "WIMEA-ICT | University of Bergen." [Online]. Available: https://www.uib.no/en/rg/meten/57609/wimea-ict. [Accessed: 07-Jun-2018].

[2] "WIMEA-ICT – WIMEA-ICT." [Online]. Available: https://wimea-ict.net/. [Accessed: 07-Jun-2018].

1.5 Definitions and Acronyms

WDR	Weather Data Repository
WIMEA	Weather information management in East Africa
API	Application Programming Interface
SDD	System Design Document
UNMA	Uganda National Meteorological Authority
FR	Functional Requirement
WRF	Weather Research and Forecasting model

2. SYSTEM OVERVIEW

Weather Information Management in East Africa ICT (WIMEA-ICT) project aims at improving the accuracy and access to weather information by the communities in the East African region through suitable ICTs.

Weather Data API shall provide means of accessing weather data from Weather Data Repository (WDR) which is a component of WIMEA-ICT. WDR receives data from both the Automatic Weather Station (AWS) and manual weather stations across Uganda. This data is required for weather assimilation, modelling, forecasting and also as an input to the message switching system at the National meteorological center.

The Weather Data API shall provide an abstraction of the WDR data without direct access to its underlying database. We shall develop two applications that shall provide input data to Weather Research and Forecasting model (WRF) which is a weather modelling tool and the Message switching system at National Meteorological Center, this data shall be accessed through the weather data API.

This section consists of the general description of the project's functionality, context and design that include;

The API shall be web based thus running on the latest browsers such as google Chrome, Mozilla Firefox and many others. It shall be accompanied by an interface that has a Home, Documentation and About page. On the Home page, users have to sign-in to acquire an account so as to access the API key which is unique to every user. This key appended to the API URL shall enable the user to access the API services. The documentation page contains the guidelines on how to use the API to access the weather data. The About page explains who can access the API data and what kind of data ranges can be acquired, for example the current weather data, Weather data recorded for 7 days and Weather data recorded for 30 days.

The Weather Data messaging application shall be a web based application running on a Linux server, the application shall transmit (send) the data through an email or SMS to the message switching system in the Table Driven Code Format (TDCF) where the data is analyzed and later sent to Nairobi.

The Little_R generator Application shall also run on Linux platform. The application shall send the data in its required data format (Little-R) to the weather research and forecasting modelling software where this data shall be used for modelling weather predictions.

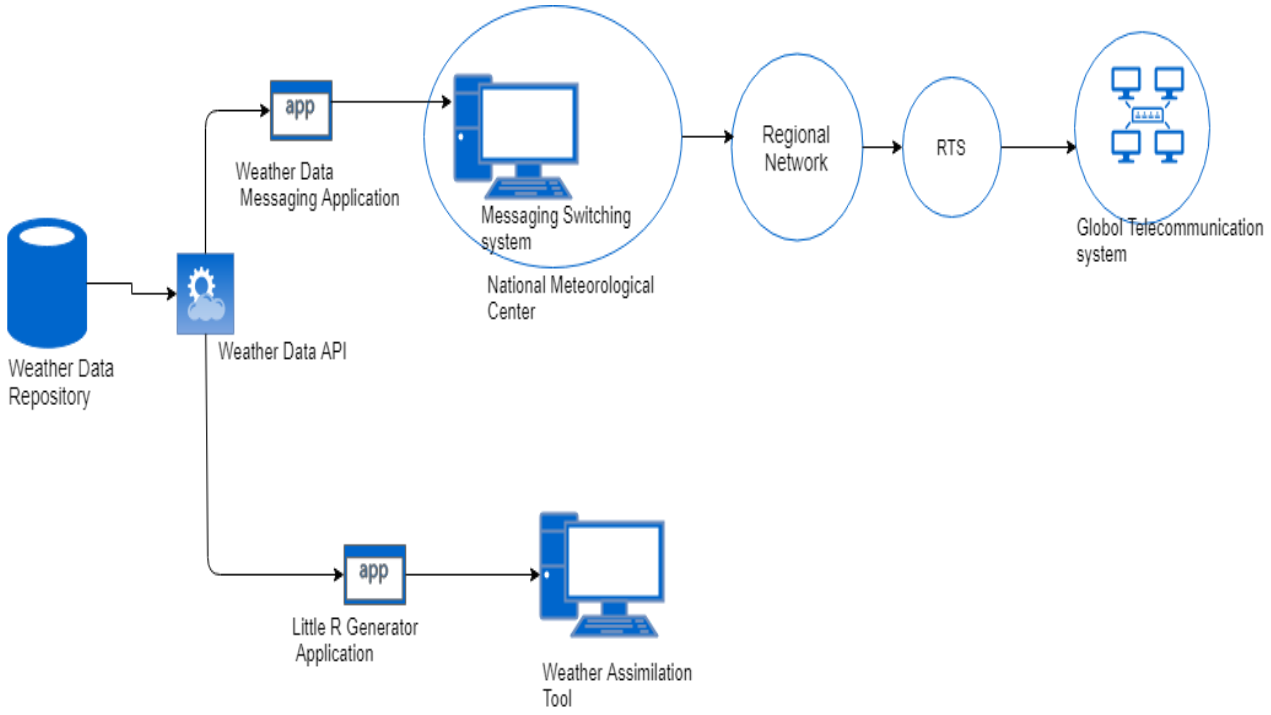

Figure 1 : Diagram showing the overall description of the system

3. SYSTEM ARCHITECTURE

3.1 Architecture diagram.

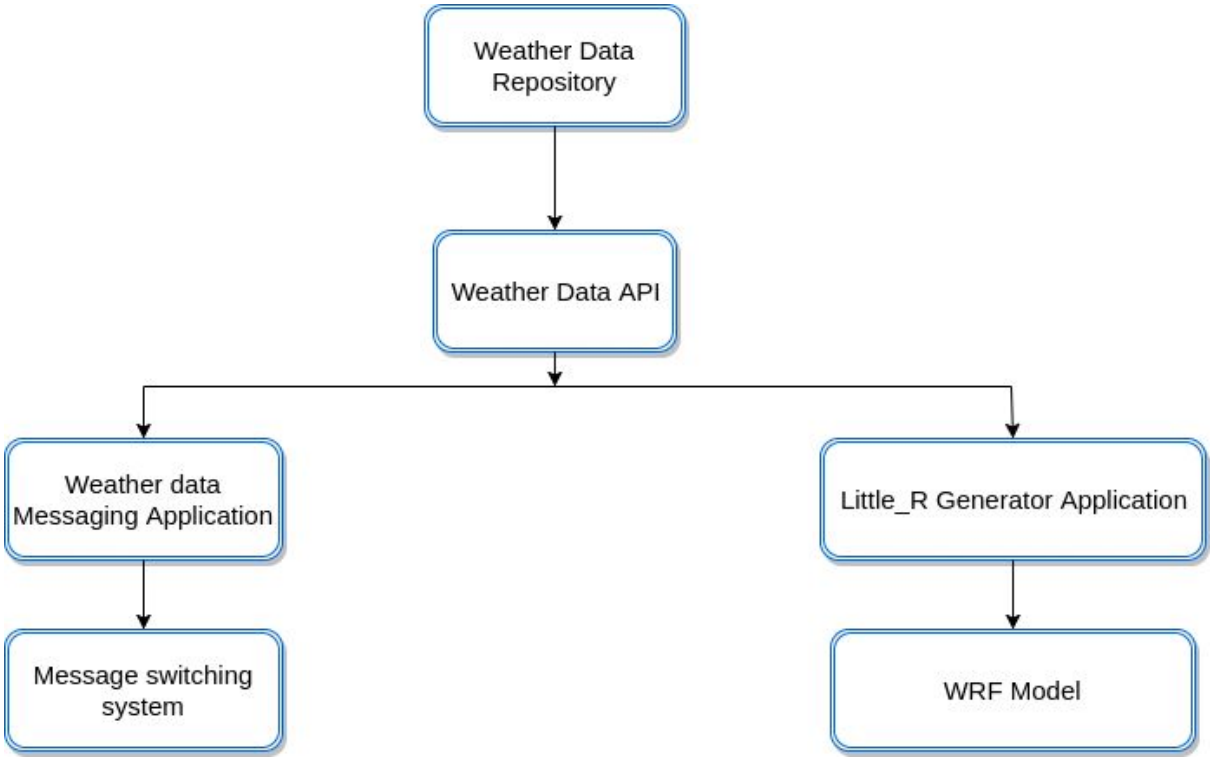

Figure 2: Architectural Diagram for the Weather Data API

Weather data repository is a data store that contains local weather data which is collected daily from both the automatic and the manual weather stations in Uganda. The weather data API will provide an abstraction of the weather data to Weather Data messaging application and Little_R Generator application. Weather Data messaging application gets the weather data from the Weather Data API converts it from JSON to TDCF format and automatically sends it to the message switching system through use of email and SMS. The Little_R generator Application gets the weather data automatically from the weather Data API, converts it to Little_R format and automatically starts the weather modelling tool. The API User first signs up with the API web interface to obtain an API key which shall be appended to the API URL thus used to obtain weather data.

3.2 Data flow diagrams.

Context diagram

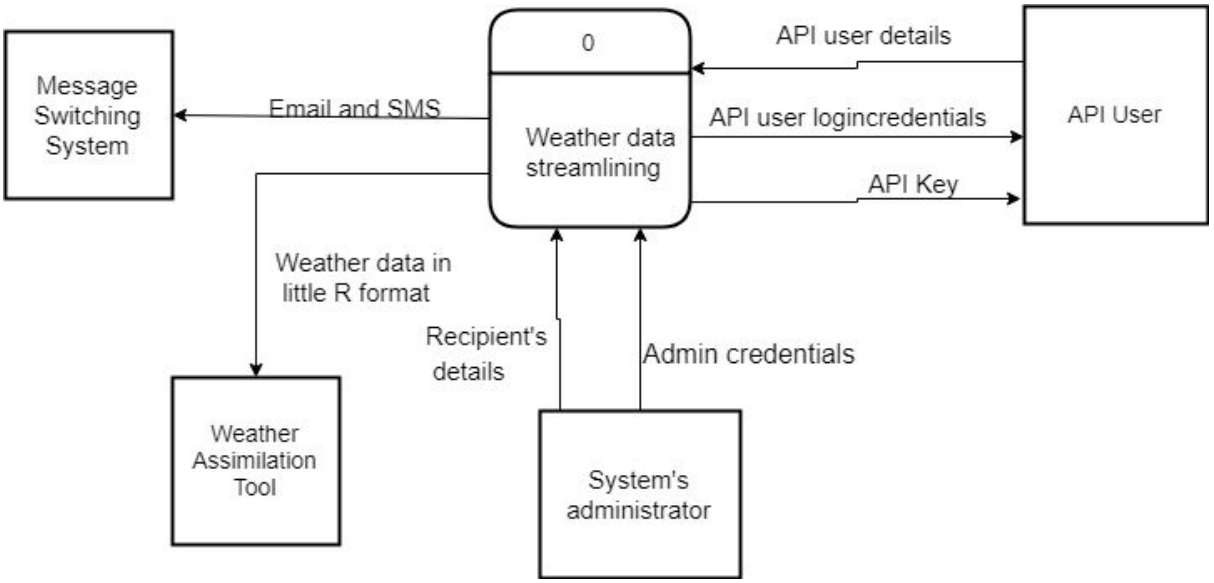

Figure 3: Context Diagram showing the data flow of the system

Level one diagram

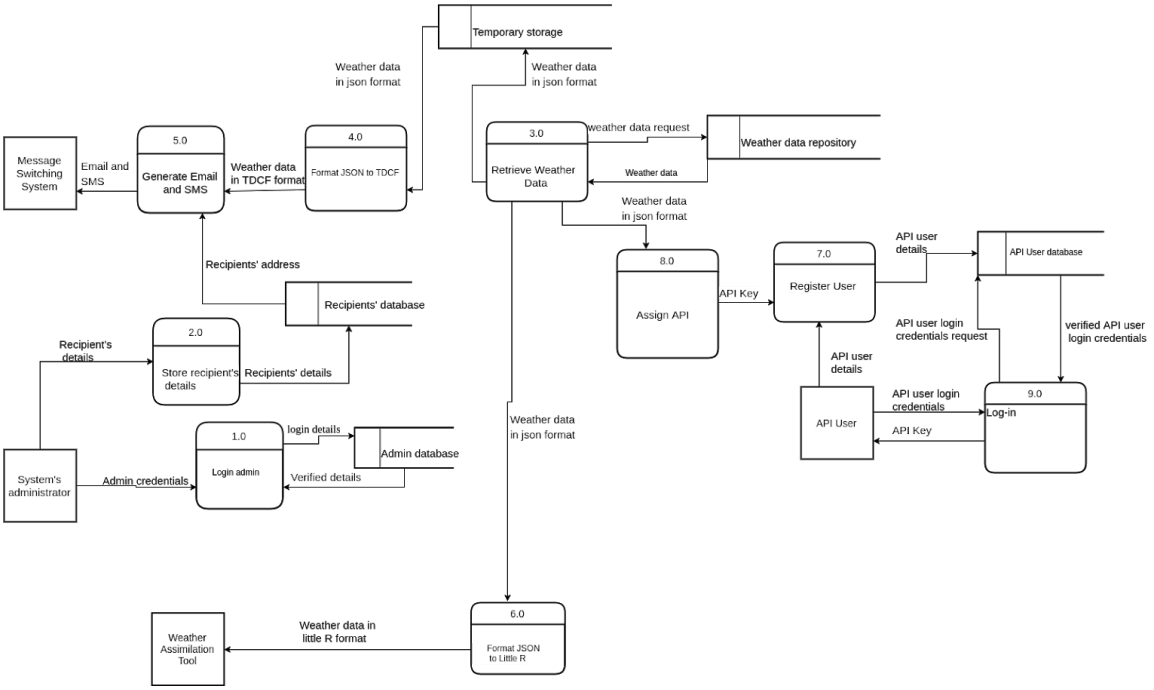

Figure 4: Level one diagram for the data flow of the system

3.3 Rationale for choosing DFDs.

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. The main aim of the Weather data API is to provide a mean to transmit local weather data to the weather modelling tool and the Message switching system. The DFDs well illustrate the flow of weather data from the Weather Data Repository (WDR) up to the intended end points.

4. DATA DESIGN

4.1 Data Description

The system consists of four databases. The API database stores data about users of the API and the encryption keys generated for each user, the WDR database stores weather data of the WDR system, the administrator database stores the credentials of the authorized administrators and the recipients' database that stores all the names of the authorized authorities to receive the data through the weather data messaging application. There are two applications, this data from the WDR database shall be sent to the national meteorological Centre in Entebbe via an email and SMS this shall be enforced through the conversion of the data from Json format to Table driven code format (TDCF) in the Weather Data messaging application and the data can also be used for assimilation through the use of the Little_R generator application where the Json data from the API is used to create Little_R files containing Little_R data which is used for assimilation by the weather data modelers and also for making prediction charts .

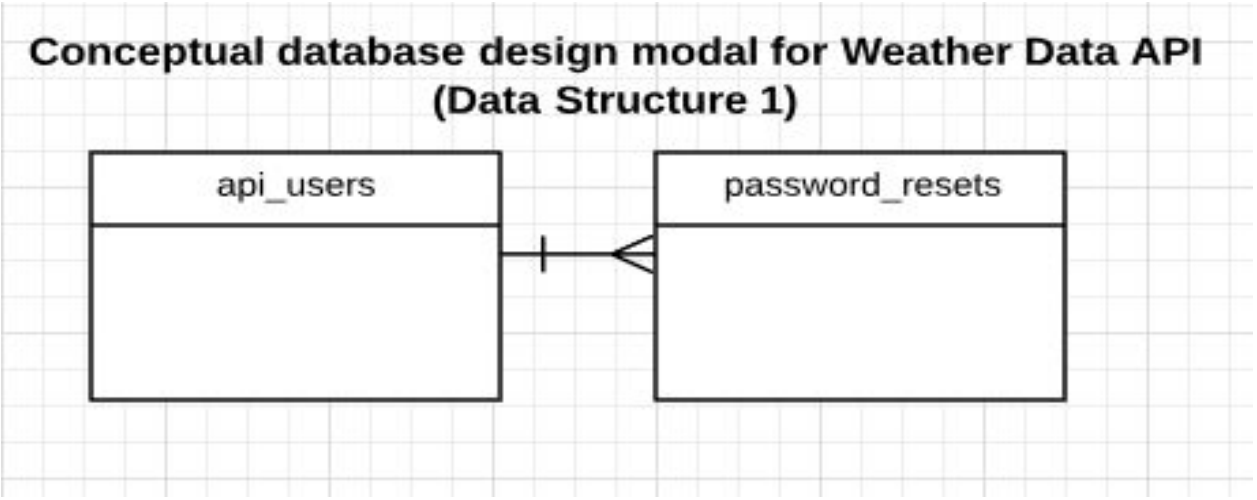

Figure 5: Conceptual design for the Weather data API

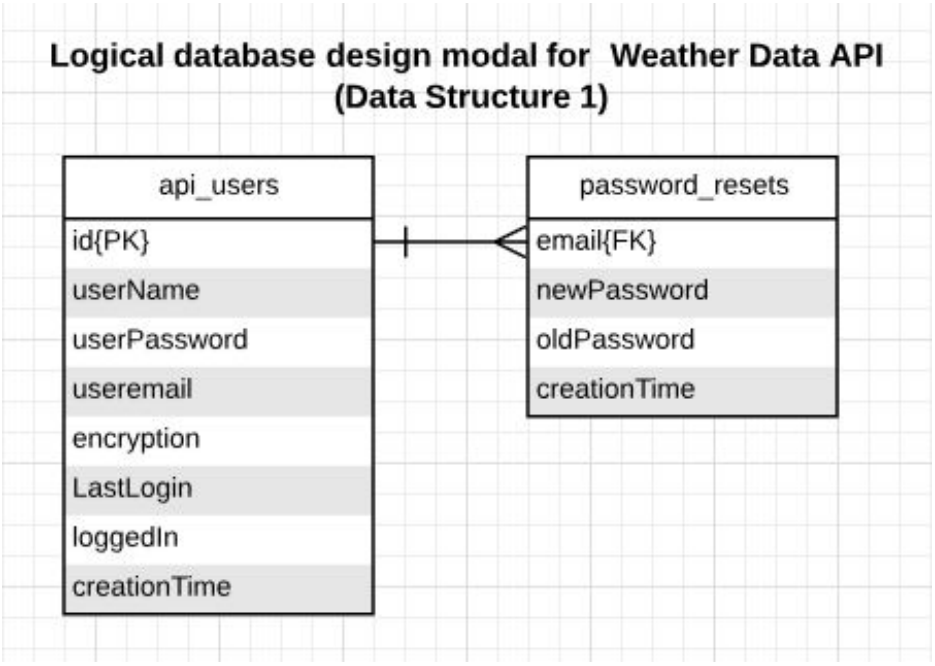

Figure 6: Logical design for API system

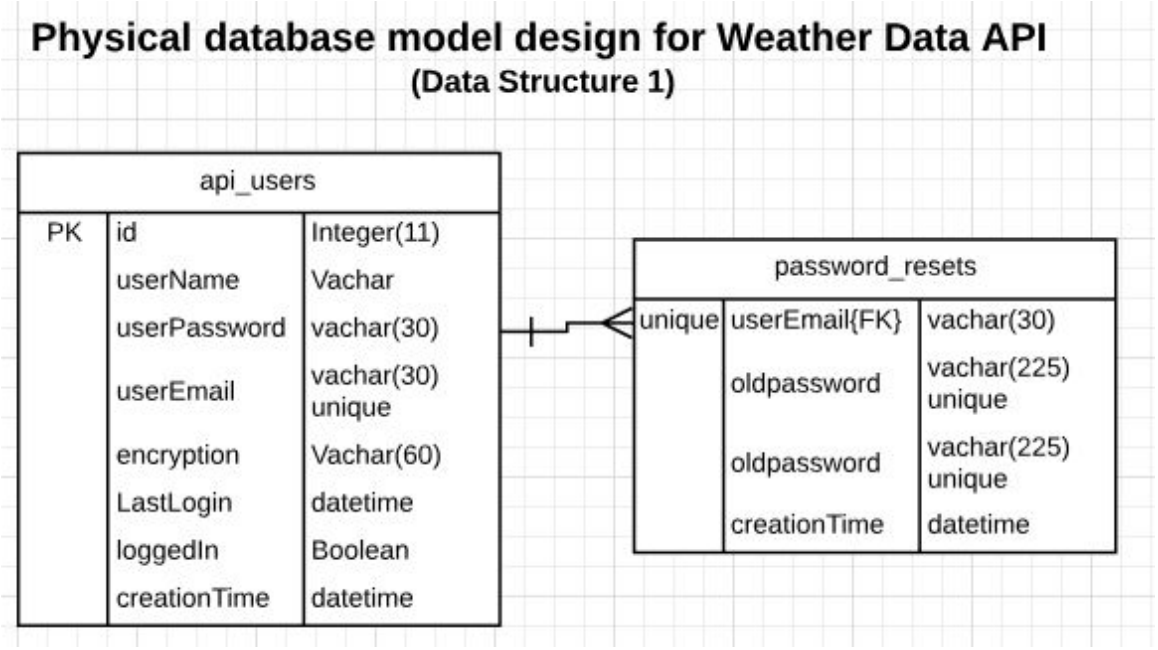

Figure 7: Physical design for the API system

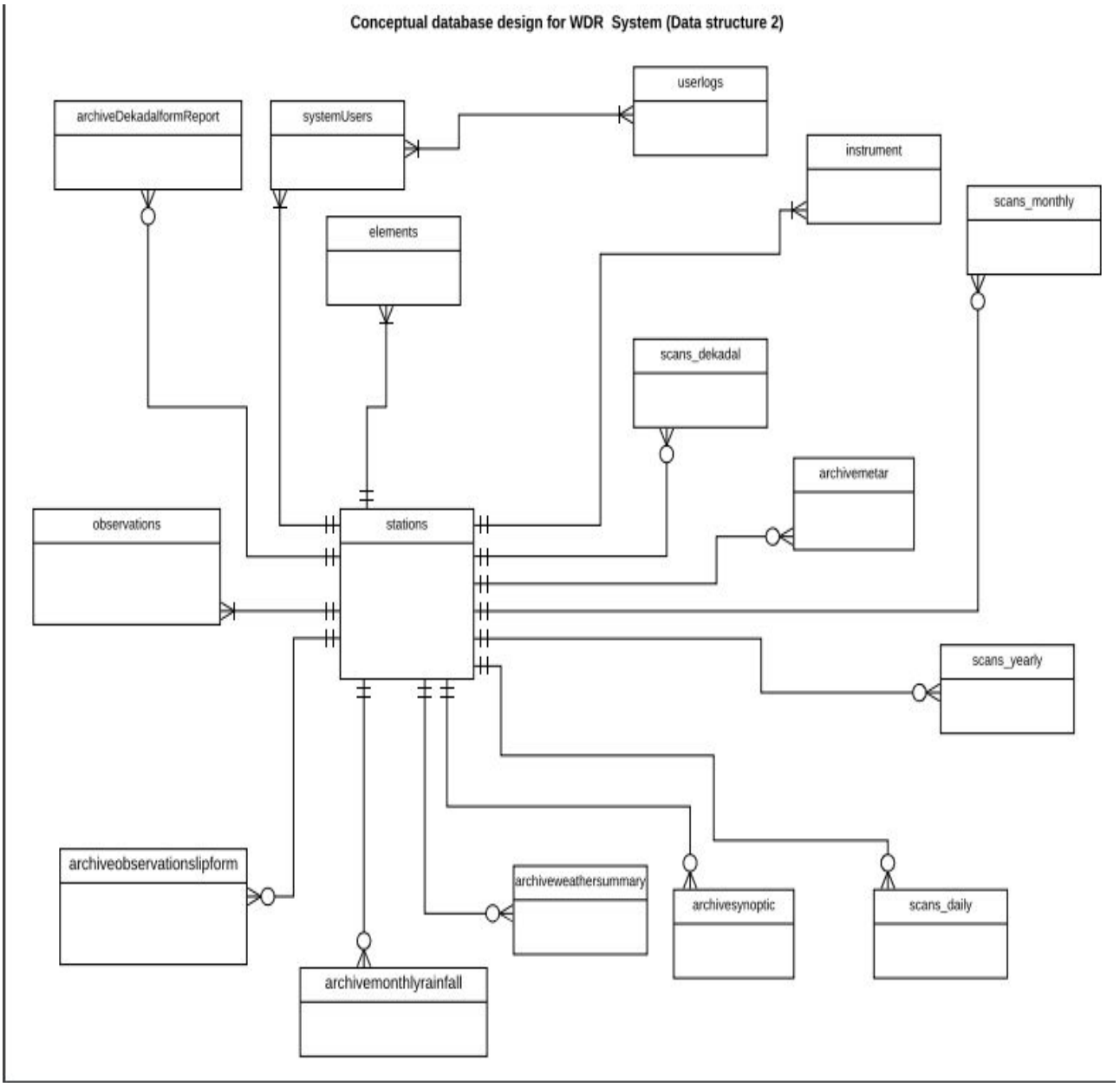

Figure 8: Conceptual design for the WDR system

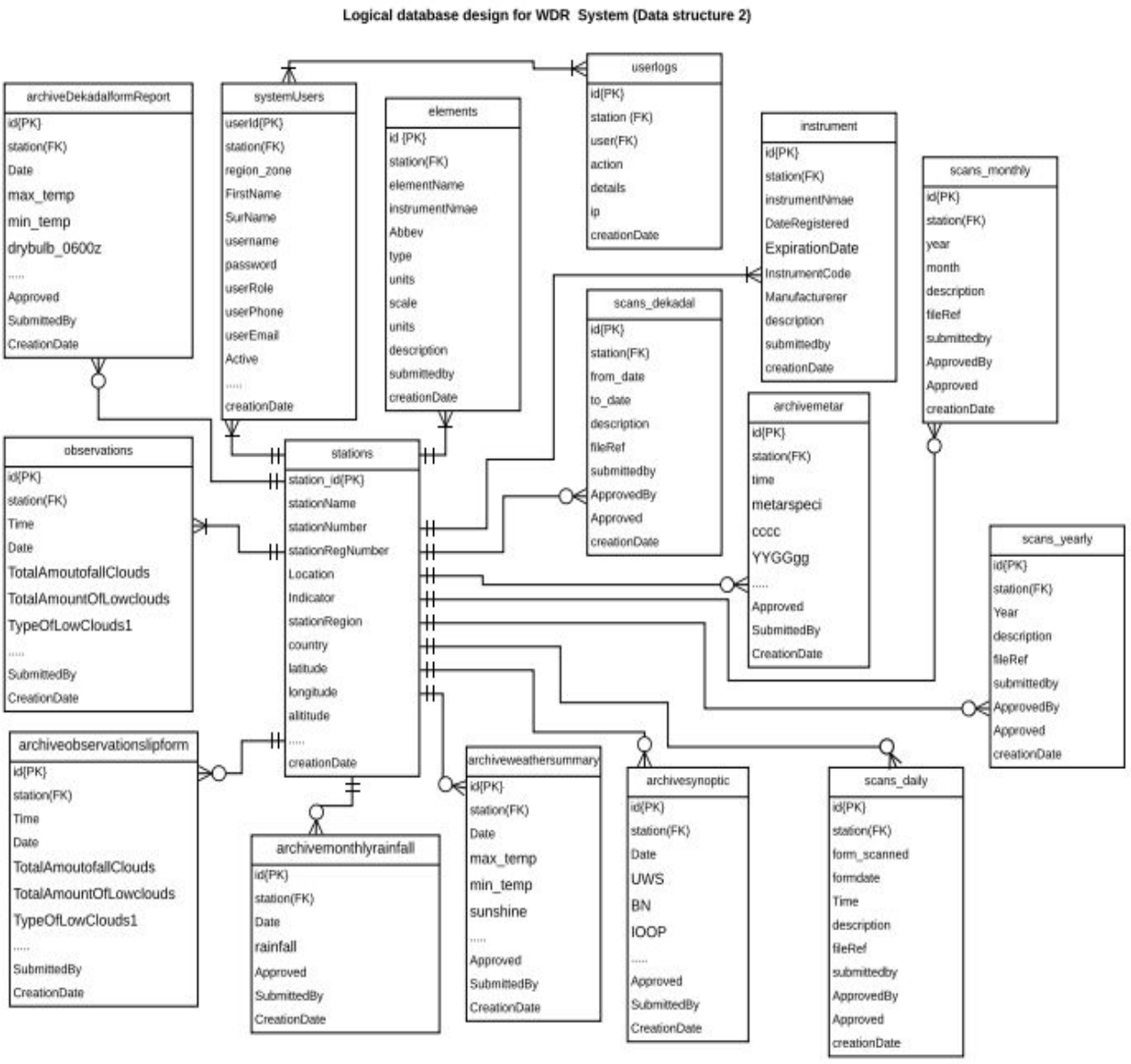

Figure 9: Logical design for the WDR system

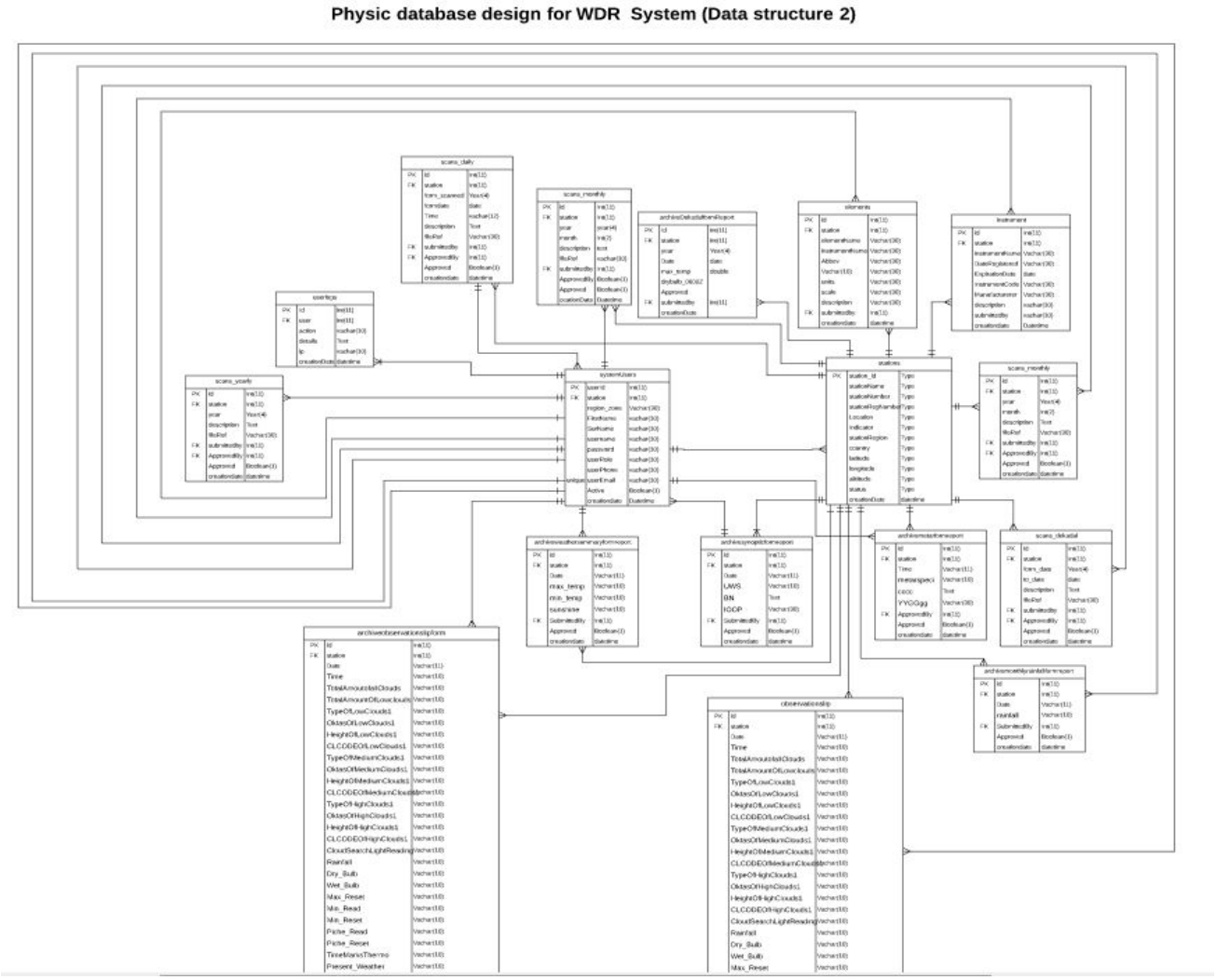

Figure 10: physical design for the WDR system

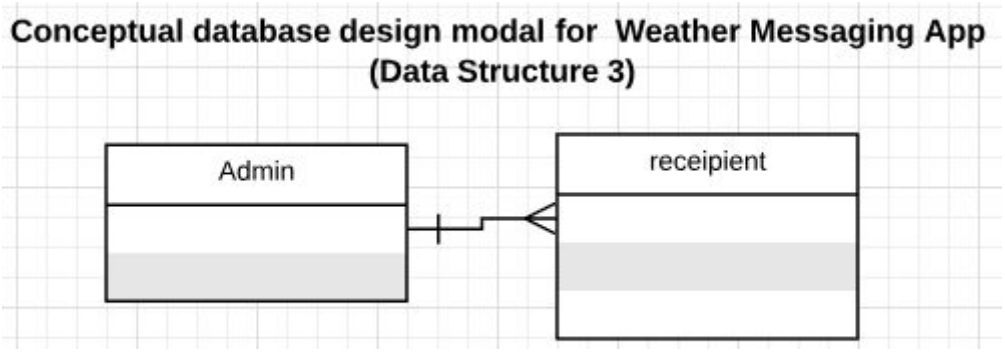

Figure 11: Conceptual design for the weather data messaging application

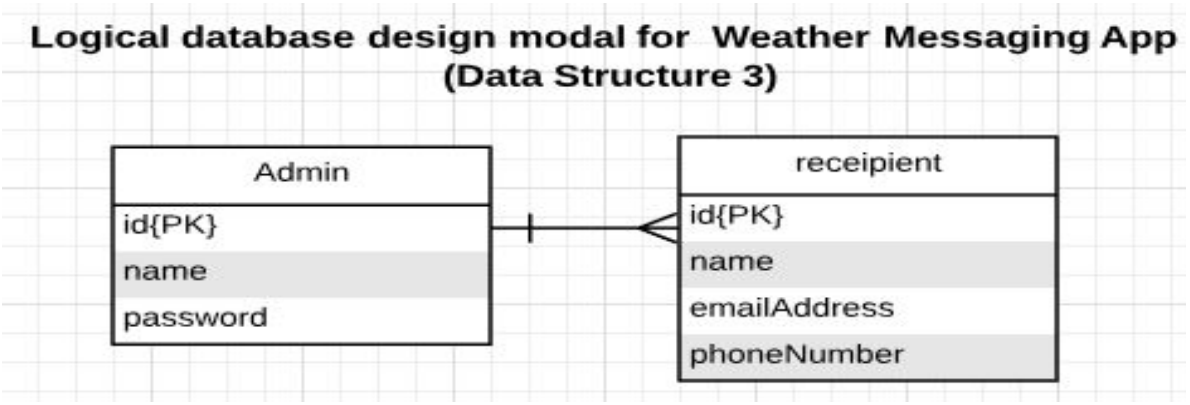

Figure 12: Logical design for the weather data messaging application

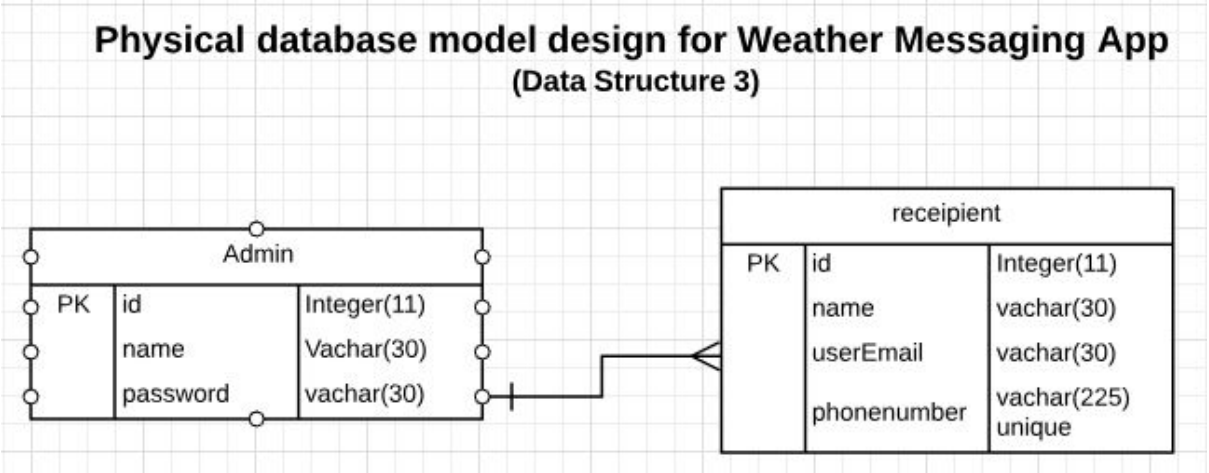

Figure 13: physical design for the weather data messaging application

Sample of JSON data output by the API

Figure 14: Sample data output by the API

4.2 Data dictionary for stations table in the WDR database containing details.

Fieldname	Datatypes	Data format	Field size	Description	Example
station_id	Integer	NNNNNNNNNN	11	Unique number ID for a station	1111111111
stationName	Text		30	Name of the station.	jovanmuteesasira
stationNumber	Text		30	Unique number assigned to a station	
StationRegNumber	Text		30	Registration number of the station	mutesasiraj@gmail.com
Location	Text		50	Location of the station	QW5Sgsh7bdd5625262fsgvxbxb5
Indicator	Text		30	Indicator of the station	14-02-2011 05:05:55
stationRegion	Text		30	Region of the station	YES
country	Text		30	Country	14-02-2011 05:05:55
Latitude	decimal		11	Latitude	0.876363535
Longitude	decimal		11	Longitude	30.0766
Altitude	decimal		11	Altitude	1010
Stationstatus	Text		30	Status whether active	active
stationType	Text		30	Type of station	synoptic
opened	date	dd-mm-yyyy	10	Date of opening station	14-02-2011
closed	date	dd-mm-yyyy	10	Date station closed	14-02-2011

submittedBy	Integer	NNNNNNNNNN	11	Person registered station	Jeo emma
creationDate	date	dd-mm-yyyy: hh:mm:ss	20	Date station created	14-02-2011

Table 1: Table showing the data dictionary for the WDR system

Data dictionary for API_Users table in the API database containing details about a given user.

Fieldname	Datatypes	Data format	Field size	Description	Example
Id	Integer	NNNNNNNNNN	11	Unique number ID for all API users.	1111111111
userName	Text		30	API login user name for the user.	Jovanmuteesasira
userPassword	Text	xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx	32	API user login password	
userEmail	Text		30	API user email address	mutesasiraj@gmail.com
EncryptionKey	Text		50	Users Key used to access	QW5Sgsh7bdd5625262fs gvxbxb5
LastLogin	datetime	dd-mm-yyyy: hh:mm:ss	12		14-02-2011 05:05:55
Loggedin	Boolean		1	Tells whether user currently logged in or not	YES
creationTime	datetime	dd-mm-yyyy: hh:mm:ss		Tracks the creation time of the user account	14-02-2011 05:05:55

Table 2: table showing the data dictionary for the API users

Data dictionary for observation table of the WDR database containing details about every observation/raw data submitted from different stations.

Fieldname	**Datatypes**	**Data format**	**Field size**	**Description**	**Example**
Id	Integer	NN	11	Unique number ID for an observation.	1111111111
Date	Date	dd-mm-yyyy	10	Date of the observation.	14-02-2011
Station	Integer	NN	11	Foreign key to station details table.	1
Time	Text		6	Time observation was taken in Zulu time.	12:30Z
TotalAmountofAllclouds	Integer	NN	11	Total amount of all clouds.	5

TotalAmountofLowclouds	Integer	NN	11	Total amount of low clouds	5
TypeOfLowClouds1	Integer	NN	11	Type of Low Clouds (Type one)	4
OktasOfLowClouds1	Integer	NN	2	Oktas of Low Clouds (Type one)	5
HeightOfLowClouds1	Integer	NN	2	Height of Low Clouds (Type one)	3
CLCODEOfLowClouds1	Text		30	CL code of Low Clouds (Type one)	Sc
TypeOfLowClouds2	Integer	NN	2		3
OktasOfLowClouds2	Integer	NN	2		2
HeightOfLowClouds2	Integer	NN	2		2
CLCODEOfLowClouds2	Text		30	CL Code of low clouds (Type two)	Sc
TypeOfLowClouds3	Integer	NN	2	Type of low clouds (Type three)	2
OktasOfLowClouds3	Integer	NN	2	Oktas of low clouds (Type three)	4
HeightOfLowClouds3	Integer	NN	2	Height of low clouds (Type three)	4
CLCODEOfLowClouds3	Text		30	CL Code of low clouds (Type three)	Sc
TypeOfMediumClouds1	Integer	NN	2	Type of medium clouds (Type one)	5
OktasOfMediumClouds1	Integer	NN	2	CL Code of medium clouds (Type one)	4
HeightOfMediumClouds1	Integer	NN	2	Height of medium clouds (Type one)	4
CLCODEOfMediumClouds1	Text		30	CL code of medium clouds(Type one)	Ac
TypeOfMediumClouds2	Integer	NN	2	Type of medium clouds (Type two)	5
OktasOfMediumClouds2	Integer	NN	2	Oktas of medium clouds (Type two)	6
HeightOfMediumClouds2	Integer	NN	2	Height of medium clouds (Type two)	7
CLCODEOfMediumClouds2	Text		2	CL code of medium clouds (Type two)	Ac
TypeOfMediumClouds3	Integer	NN	2	Type of medium clouds (Type three)	5
OktasOfMediumClouds3	Integer	NN	2	Oktlas of medium clouds (Type three)	5
HeightOfMediumClouds3	Integer	NN	2	Height of medium clouds (Type three)	5
CLCODEOfMediumClouds3	Text		2	CL code of medium clouds (Type three)	Ac

TypeOfHighClouds1	Integer	NN	2	Type of high clouds (Type one)	5
OktasOfHighClouds1	Integer	NN	2	Oktas of high clouds (Type one)	4
HeightOfHighClouds1	Integer	NN	2	Height of high clouds (Type one)	4
CLCODEOfHighClouds1	Text			CL code of high clouds (Type one)	Cl
TypeOfHighClouds2	Integer	NN	2	Type of high clouds (Type two)	4
OktasOfHighClouds2	Integer	NN	2	Oktas of high clouds (Type two)	4
HeightOfHighClouds2	Integer	NN	2	Height of high clouds (Type two)	4
CLCODEOfHighClouds2	Text			CL code of high clouds (Type two)	Cl
TypeOfHighClouds3	Integer	NN	2	Type of high clouds (Type three)	4
OktasOfHighClouds3	Integer	NN	2	Oktas of high clouds (Type three)	4
HeightOfHighClouds3	Integer	NN	2	Height of high clouds (Type three)	4
CLCODEOfHighClouds3	Text			CL code of high clouds (Type three)	Cl
CloudSearchLightReading	Integer	NN	2	Cloud Search Light Reading (Type three)	4
Rainfall	Integer	NN	2	Rainfall measured in mm	4
Dry_Bulb	Decimal		5	Dry Bulb	
Wet_Bulb	Decimal		5	Wet Bulb	7.4
Max_Read	Decimal		5	Max Read	5.44
Max_Reset	Decimal		5	Max Reset	0.99
Min_Read	Decimal		5	Min Reset	4.50
Min_Reset	Decimal		5	Piche Read	4.45
Piche_Read	Decimal		5	Piche Reset	3.45
Piche_Reset	Decimal		5	Time Marks Hydro	2.34
TimeMarksThermo	Decimal		5	Time Marks Rain Rec	2.35
TimeMarksHygro	Decimal		5	Present weather	3.45
TimeMarksRainRec	Decimal		5	Time Marks Rain Rec	0.34
Present_Weather	Decimal		5	Present Weather	4.45
Present_WeatherCode	Decimal		5	Present Weather Code	3.45
Past_Weather	Text		30	Past Weather	
Visibility	Text		30	Visibility	
Wind_Direction	Text		30	Wind Direction	
Wind_Speed	Text		30	Wind Speed	
Gusting	Decimal		5	Gusting	2.35
AttdThermo	Decimal		5	Attended Thermo	4.36

PrAsRead	Decimal		5	Pr.As Read(C)	0.27
Correction	Decimal		5	Correction	3.94
CLP	Text		30	C.L.P(mb)	
MSLPr	Decimal		5	M.S.L.Pr(mb) or 850mb. Ht.(gpm)	2.90
TimeMarksBarograph	Decimal		5	Time Marks barograph	3.95
TimeMarksAnemograph	Decimal		5	Time MarksAnemograph	2.35
OtherTMarks	Decimal		5	Other Time Marks	3.95
Remarks	Text		30	Remarks	
SubmittedBy	Text		30	Submitted By	Musa Joe
Approved	Boolean		1	Whether Approved or not	Yes
creation_date	Datetime		10	Submission time of the observations/raw data	02-08-2011 08:30:38
SoilMoisture	Decimal		5	Soil moisture readings	2.09
SoilTemperature	Decimal		5	Soil Temperature readings	30.70
sunduration	Text		30	Sun Duration	
trend	Text		30	Trend	
windrun	Text		30	Wind run	
speciormetar	Text		30	Specifies whether a special phenomenon is observed	
UnitOfWindSpeed	Text		30	Unit Of Wind speed	
IndOrOmissionOfPrecipitation	Text		30	Ind Or Omission Of Precipitation	
TypeOfStation_Present_Past_Weather	Text		30	Type Of station present past weather	
HeightOfLowestCloud	Text		30	Height of Lowest cloud	
StandardIsobaricSurface	Text		30	Standard Isobaric Surface	
GPM	Text		30	Geopotential Of Standard Isobaric Surface	
DurationOfPeriodOfPrecipitation	Text		30	Duration Of Period Of Precipitation	
GrassMinTemp	Text		30	Grass Minimum temperature	
CI_OfPrecipitation	Text		30	Character and Intensity of Precipitation	
BE_OfPrecipitation	Text		30	Beginning or End of Precipitation	
IndicatorOfTypeOfInstrumentation	Text		30	Indicator Of Type Of Instrumentation	
SignOfPressureChange	Text		30	Sign Of Pressure Change	

Supp_Info	Text		30	Supplementary Information	
VapourPressure	Integer	NN	2	Vapour Pressure	100
T_H_Graph	Text		30	TH Graph	
DeviceType	Text		30	Type of device used to submitted observation/raw data	web

Table 3: Table showing data dictionary for the Observation table of the WDR database

5. COMPONENT DESIGN

5.1 Weather Data Messaging Application.

Algorithm

Step 1: Start.

Step 2: Weather Data messaging application requests for weather data from weather data API.

Step 3: Check if there is current weather data.

Step 4: If it's there, proceed else wait for 30 minutes and request again.

Step 5: Receive data, convert the JSON data into Table Driven Code Format and generate email or SMS.

Step 6: Generate email and SMS.

Step 6: Send email and SMS to Message switching system in Entebbe.

Step 7: End.

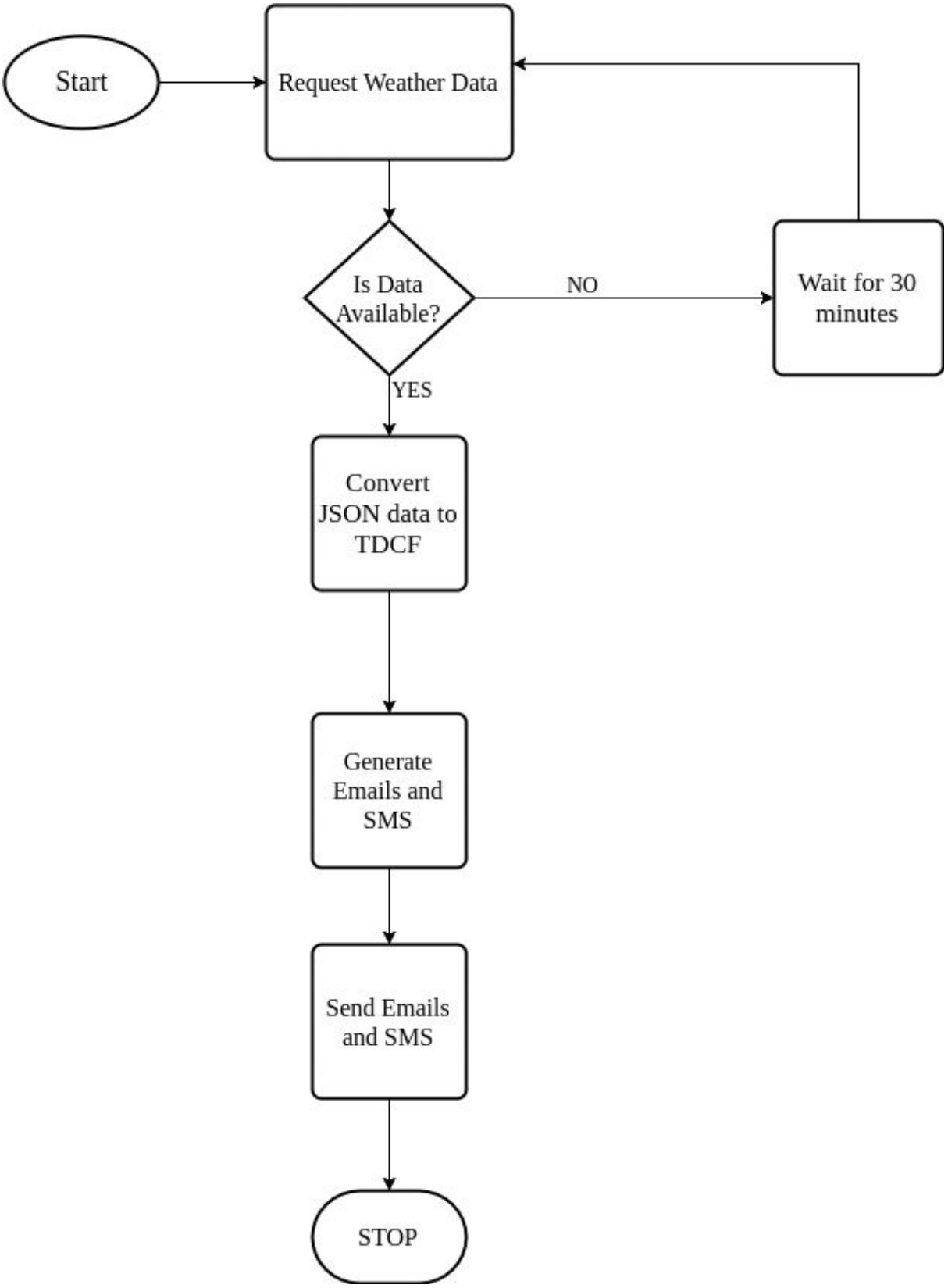

Figure 15: How Weather data is transmitted to the Message Switching System.

5.2 Little_R Generator

Algorithm

Start.
Little_R generator Application requests for weather data.
Receive the data and converts it to Little_R
Start modelling tool.
Stop.

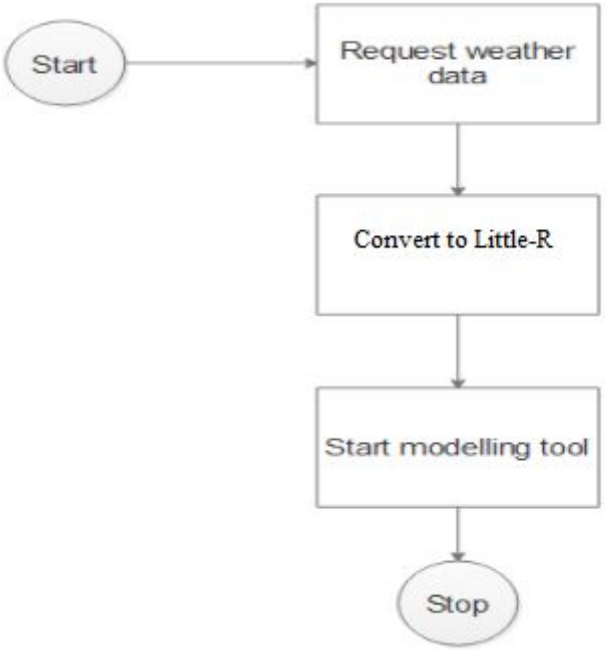

Figure 16: showing how weather data is transmitted to the Weather Modelling tool

5.3 API User.

Algorithm

Step 1: Start.
Step 2: Sign up for Weather data API key.
Step 3: Assign API key to user.
Step 4: Append the API key to the URL.
Step 5: Check if enter key is correct,
Step 6: If the key is correct, proceed else try again and enter the correct key.
Step 7: Access API services.
Step 8: Stop.

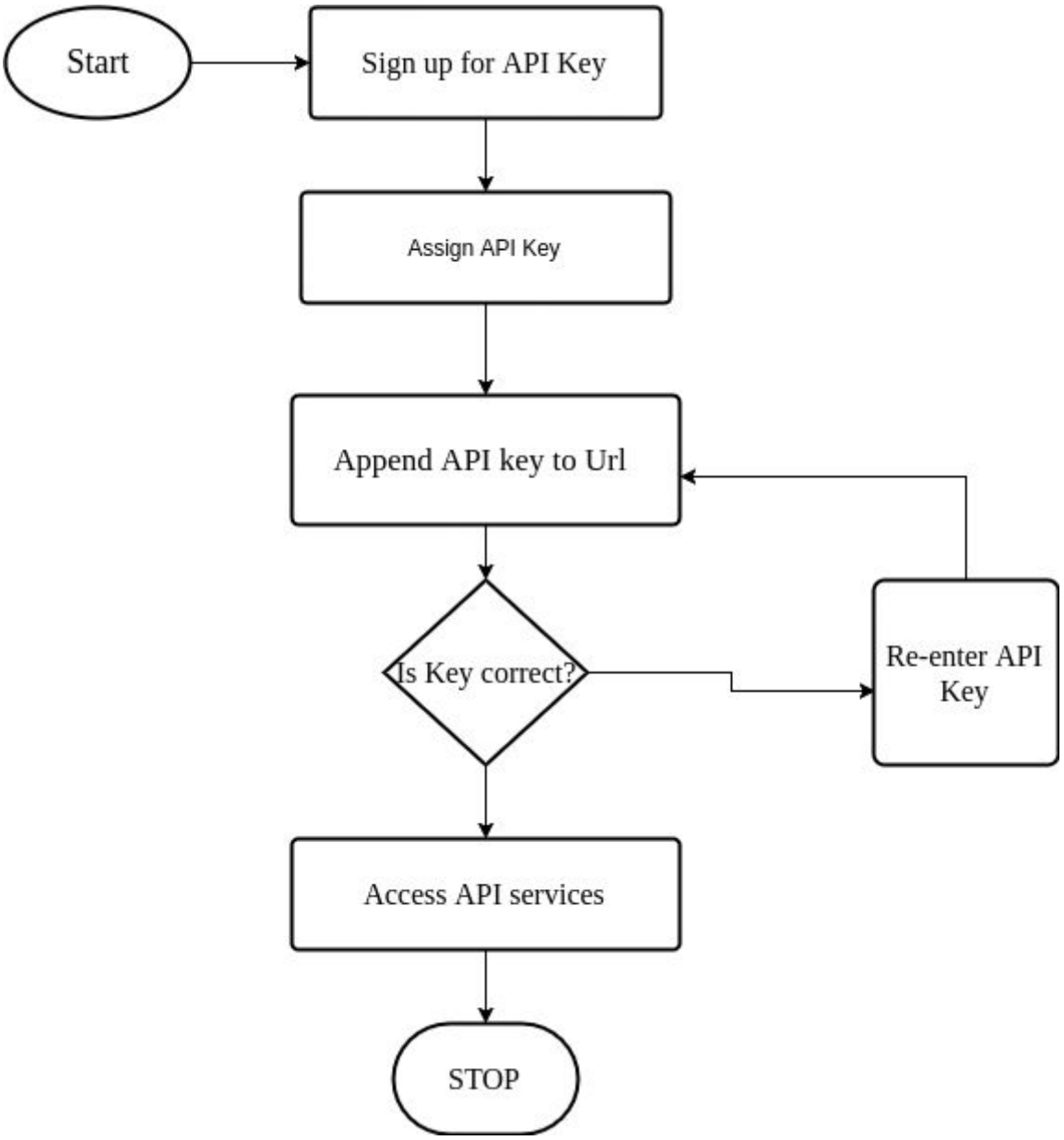

Figure 17: showing data flow in the weather data API

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The user shall be required to sign-up for an API key in order to use the API. The user shall enter their username, email, Password and password confirmation or else sign in with only the username and password.
The user shall be availed with the opportunity to change their username and passwords once they are logged in.
Upon logging in, the user shall be able to view a randomly generated API key that has been assigned to that particular user.
With the API key available, users can access weather data from the WDR database while maintaining a certain level of abstraction.
The user shall be able to view the documentation once he/she clicks on the DOCUMENTATION tab, this shall contain help information regarding how to use and acquire the API and also a sample of the JSON structure.
The user shall also be able to view more information about the API and the developers from the blog link which shall be accessed by clicking on the BLOG tab.
The user shall be able to know who can access what data from the weather data API and which ranges is the data received when he/she clicks on the ABOUT tab.

6.2 Screen Images

The Home Screen

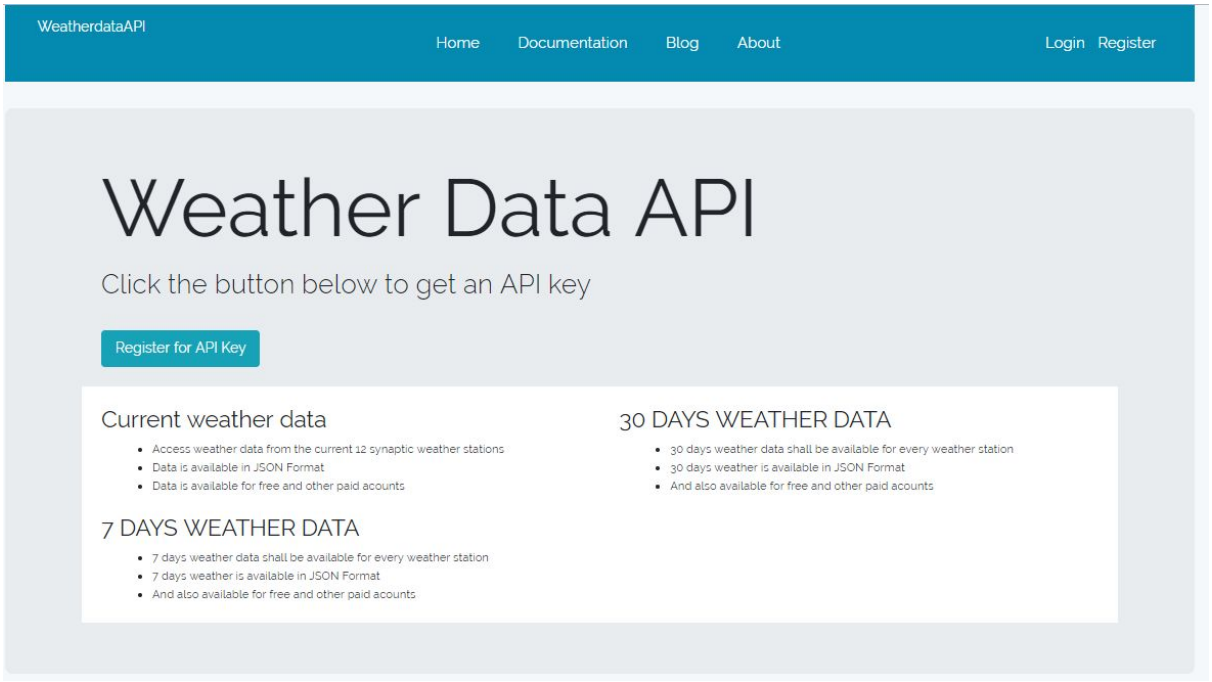

Figure 18: Home Screen of the Weather data API

This shows the different weather data ranges that the users shall be able to acquire from our API using the two applications too. It also has Log in and Register buttons for the user to acquire accounts.

Create New Account

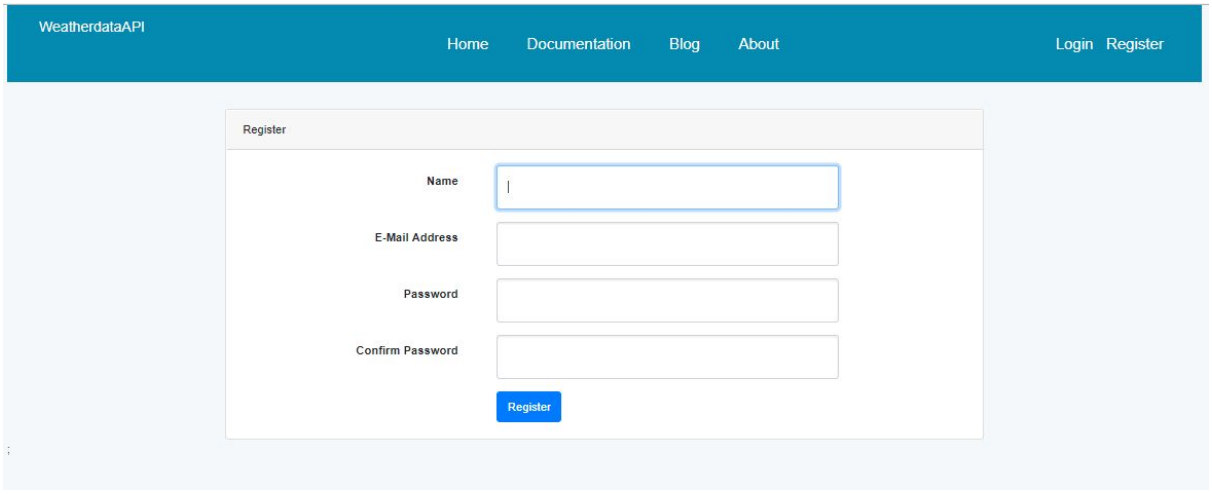

Figure 19: how user creates an account

Create account screen, this is the page where the user is able to create their account which details they will use each time they want to access the API for any data.

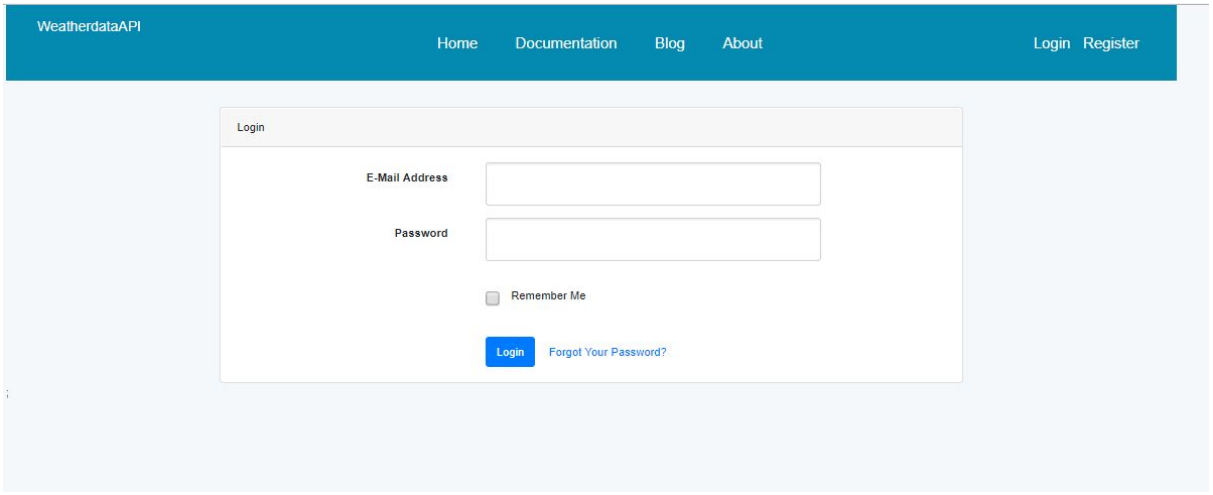

Figure 20: showing how a user logs into the system

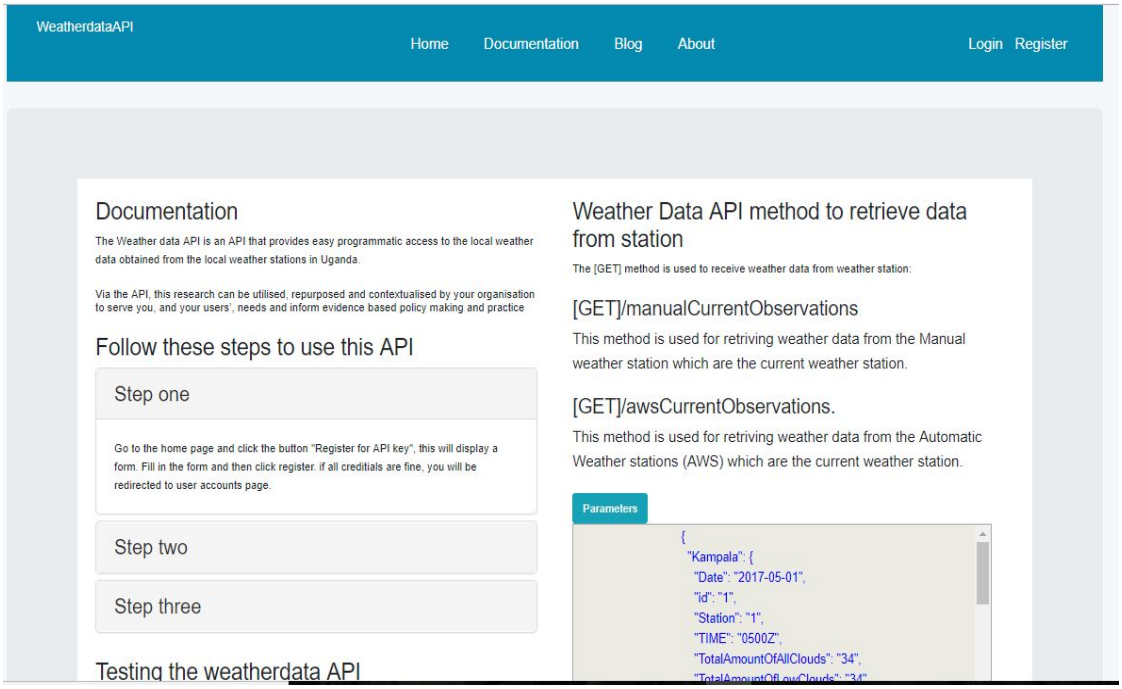

Figure 21: screen showing the documentation page

Registered Users Home Screen

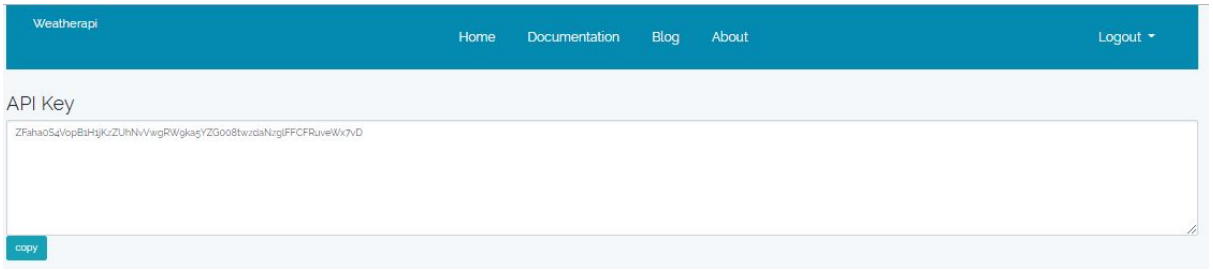

Figure 20: screen showing the API key

Registered user home screen, at this page the user is able to edit/change their log in details and also acquire an API key which he/she appends to the API URL.

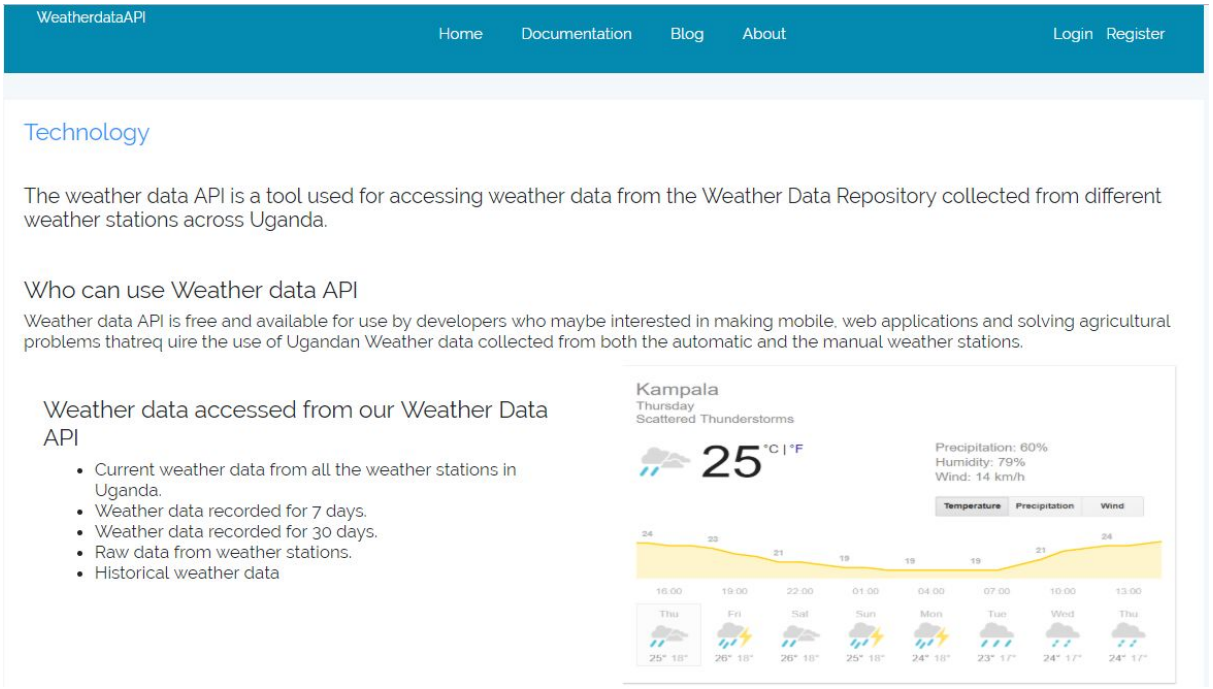

Figure 22: The About Screen

This page tells the API user who can access the API and what data ranges are displayed on the system.

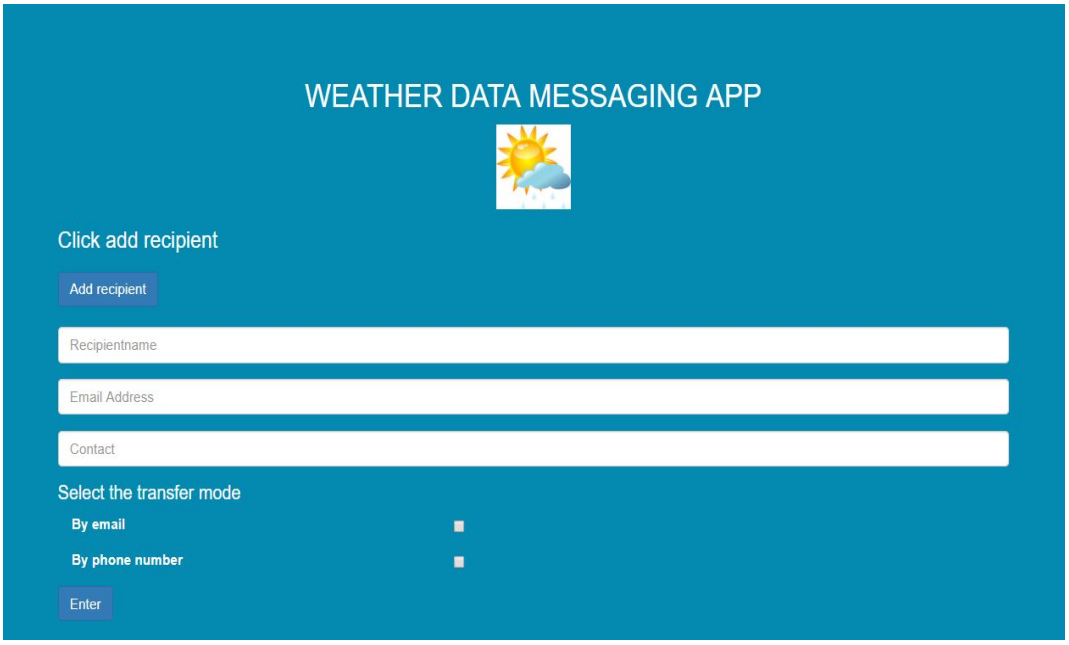

Figure 23: screen showing how an administrator adds a recipient

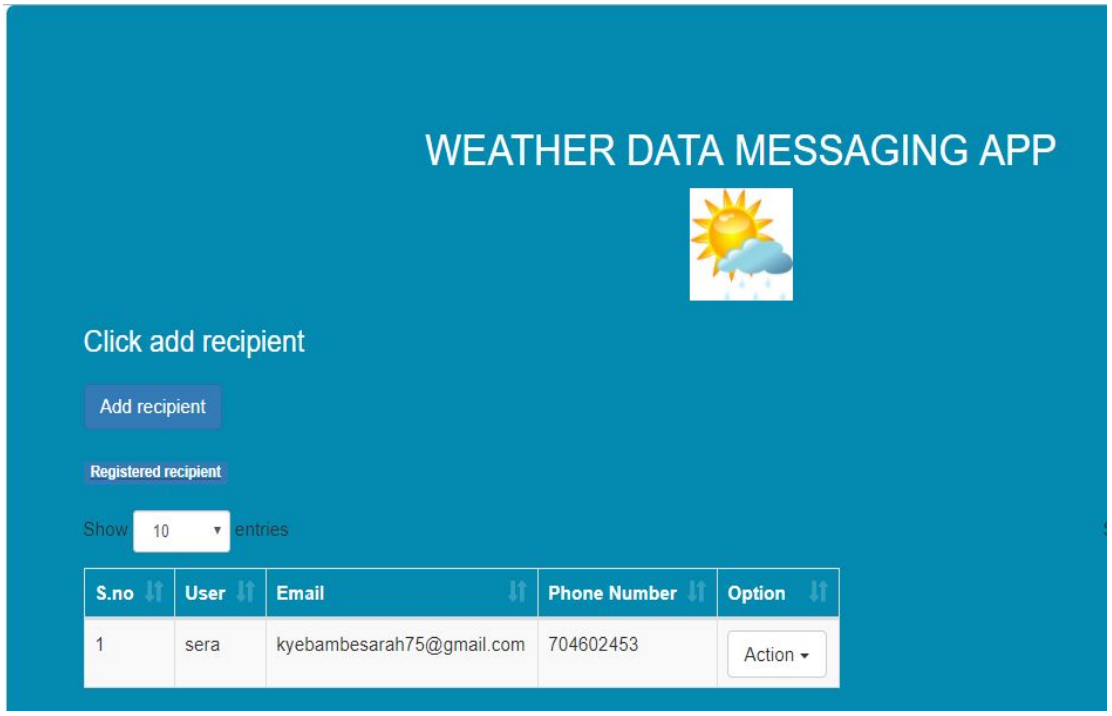

Figure 24: Weather data messaging system showing recipients available

6.3 Screen Objects and Actions

The Home screen

Login button

- This enables the user with an existing account to log in to the account using a username and password.

Register for API key button

- This enables a user without an account to acquire one.

The Create New Account screen

Username

- User name can be in the range of 6 to 20 letters (numbers), as a standard.

Email address

- The email range is left open but the email should be valid and working.

Password

- Password can be ranged from 6 to 20 letters (numbers), as a standard. No special characters, space.

Confirm Password

- Re-type the password to confirm it's the same as the first password.

Create account

- This is the final step; an account is immediately created on click.

The Registered Users Home Screen

Here the user is able to change his/her personal details and password

API key

- Since the system automatically generates a key for the user could only copy, append the key to the URL in the browser.

Log out

- The user is able to successfully leave the system on click.

The Documentation Button

- This screen contains information about how to use the API and sample of the JSON data.

The Blog button

- It is automatically connected to the blog link thus direct access to the blog.

The About screen

- The API gives a detailed information of the kind of data you are able to acquire.

Adding a recipient

- The administrator adds a recipient to the database by entering their name, email address and phone number.

List of recipients

- This shows a list of recipients who will be able to receive data from the weather data API using the emails and SMS.

7. REQUIREMENTS MATRIX

Functional Requirement code	Component 5.1	Component 5.2	Component 5.3	Data structure 1	Data structure 2
FR01			x	x	
FR02			x		x
FR03			x		x
FR04			x	x	
FR05			x	x	
FR06			x	x	
FR07			x	x	
FR08		x			x
FR09		x			x
FR10	x				x
FR11	x				x

Figure 24: Requirements Matrix