# Software Requirements Specification

# for

# wimea-ict project WDR (RC2)

Version 1.0 approved

Prepared by Muhumuza Joshua and Mutesasira Jovan

wimea

24th July 2017

Table of Contents

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| **Muhumuza Joshua** | 24/7/2017 | Created it | 1 |
|  |  |  |  |

# Introduction

## Purpose

This document is mainly describes the requirements of the WDR RC2 component that encompasses data capturing from both the automatic and manual weather stations.

## Document Conventions

## Intended Audience and Reading Suggestions

This document is created for the different users interested in understanding what the system is expected of in order to help in the design, development and testing phase

## Product Scope

WIMEA RC2 component aims at improving the accuracy and reliability of the weather information by communities in the East African region through suitable use of ICTs. The purpose of this component is to provide a secure and efficient means of data capturing from the different weather stations and a platform for producing organized data which can be used in the dissemination component RC4

It has three (3) different Versions which include WIMEA Web, WIMEA Desktop and WIMEA Mobile. The Desktop and Mobile Version of the system are majorly used for climatic data entry into the various forms of the system while the Web Version will have all the system functionalities ranging from data entry, to view of data entry reports and all administrative roles of the system.

All data entered from the different versions of the system is stored in one central web server (repository). The Mobile and Desktop system versions can enable users submit data even without accessible to internet and later when you are connected to the internet, the data will be transmitted to the central web server. Observed Data (Reports) stored on the web server can be downloaded to a Personal Computer (PC) over the internet of which this data will be stored in MS Excel.

## References

1. Overall Description

## Product Perspective

This system is a self contained product with an aim of capturing the data in an efficient and safe manner. The figure below shows the abstract data flow diagram of the RC2 component of the application



## Product Functions

## Use case diagram for the entire WDR component

## User Classes and Characteristics

* + 1. **Observer**

**Data Entry** will be able to scan all previous documents concerning weather and thereafter upload (Achieve) the Scanned document(s).Before uploading the Scanned documents they will be required to enter some information concerning the scanned document they want to upload. This information may include things like the date of the scanned documents when entered.

**Export Data:** Users (Observers) will be able to export any data from the system into different formats like excel, csv.

**Import Data:** Observers will be able to import data from different previous systems like climsoft and clidata and also from different file formats like Csv, excel. This data may be Meta data that data about stations, elements, instruments.

* + 1. **Automatic weather stations**

**Automatic Data Entry:** Automatic Weather Stations (Devices) send data to the system.

* + 1. **Manager**

**Enter Meta Data Into The System:** Metadata refers to information about stations, elements, instruments, the units that will be used to measure the elements and other objects in the system. Station information must be in place before data for that station can be added. We anticipate this information being added by the station Manager.

**Approve Data Entered:** The Manager will be able to approve all the data collected by the observer in the WDR System. Once data has been approve by the Manager it will be transferred to a permanent database. Describes the quality control facilities available for checking the consistency of data that have been entered, and for correcting values when necessary.

**Defining User Privileges**: This will describes administration functions, such as defining users of the system, and updating the metadata (information about stations, elements and other entities described in the database). The User rights can be read, hide, insert, update and delete and apply to individual parts of the application as well as to individual parts of the tables.

**View Data Analysis**: Manager will be able to use the system to be able to come up with better analysis on the information that has been collected. A Manager will use the system to generated information concerning basic computations on the data e.g Mean, Average, display reports, graphs Monthly data and charts, Cumulative precipitation Charts. The user defines only what he/she wants to calculate and the system carries out the calculations automatically without interaction. The definition of the formulas used for the calculation is very flexible, and the user is able to define very complex calculations.

## Operating Environment

* + 1. **WDR Web Application**

|  |  |
| --- | --- |
| Hardware |  |
| Operating system | Windows, linux, mac OS |
| Cpu |  |
| RAM |  |
| User | Observer, manager, automatic weather station |
| Software | All latest web browsers i.e. chrome, Mozilla firefox, safari  Server |

* + 1. **WDR mobile Application**

|  |  |
| --- | --- |
| Hardware | Smart phones |
| Operating system | Android version 4.1.1 + |
| Cpu | None |
| RAM | None |
| User | Observer, manager, automatic weather station |
| Software | None |

* + 1. **WDR desktop application**

|  |  |
| --- | --- |
| Hardware |  |
| Operating system | Windows, linux, mac OS |
| Cpu |  |
| RAM |  |
| User | Observer, manager, automatic weather station |
| Software | None |
|  |  |

## Design and Implementation Constraints

* + 1. **WDR mobile application**
* The mobile application is constrained to allow data entry by users whose location matches to the location of the station they are registered to.
* The mobile application shall only allow non- duplicate data entry into the database.
* The mobile application shall support only smart phones with android 4.1.1 and above.
* The mobile application is constrained to supporting only observation form data and more form fields data entry into the central repository
* The user will have to login inorder to acquire data entry services provided by the mobile application
* The mobile application will delete data stored offline if that data has been submitted online by the user
* All transactions provided by the mobile application will be done in less than 2 seconds
  + 1. **WDR desktop application**
* This desktop application is constrained to only support data entry i.e. more form data and observation form data.
* The desktop application will run only on Ubuntu, windows and mac os x
* All transactions of this application will be completed in less than 2 seconds
* A notification is required upon any transaction made by the user for guidance.
* An offline database is required to support offline data entry
* Synchronization of offline data to the database must happen at the user’s opinion.
* The application should be able to detect the state of network availability on the machine its installed on in less than 2s.
* At first login, a user is required to have an internet connection for verifying their details

## User Documentation

User manuals for the web application, mobile application and desktop application will be provided to guide users.

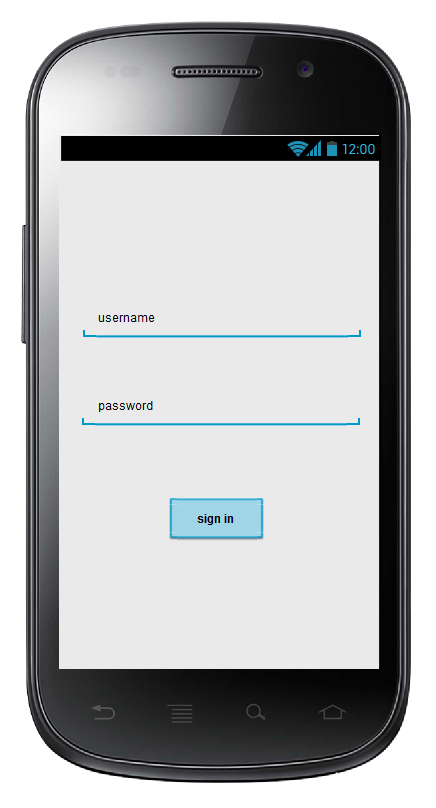
## Assumptions and Dependencies

* + 1. **WDR Mobile application**
* One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.
* The GPS components in all phones work in the same way.
* The mobile phone has access to internet at any point in time of data submission
* The mobile phone supports touch gestures.
* A user has already been registered into the system i.e. he/she has login credentials

# External Interface Requirements

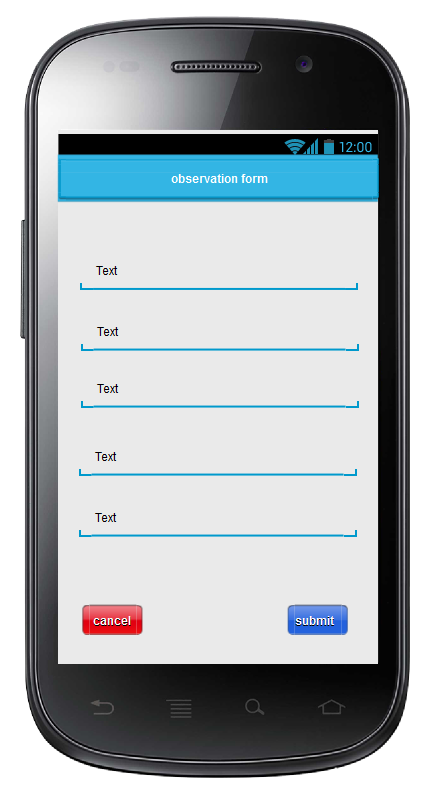
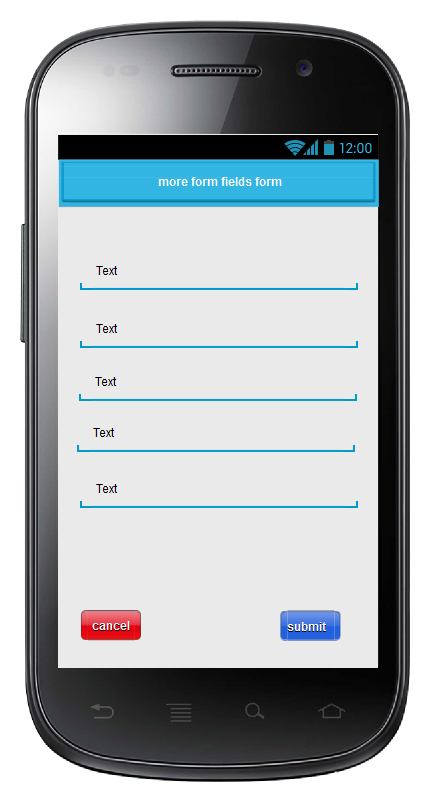
## User Interfaces

A first-time user of the mobile application should see the log-in page when he/she opens the application.



**Observation form interface**

This interface comprises of different form fields in the observation form that are entered by the user

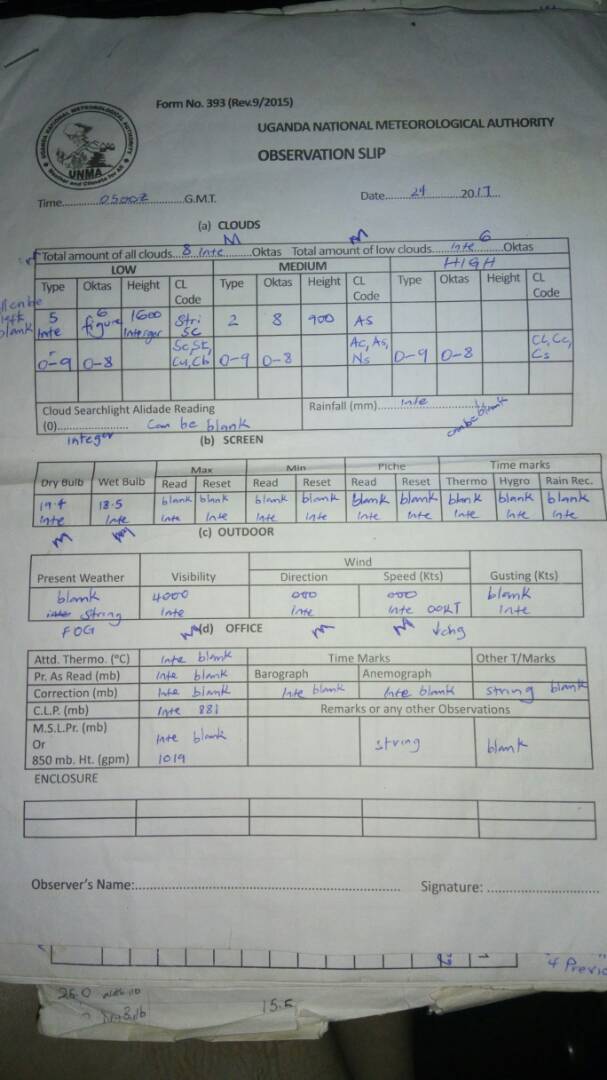
**Form fields**

These different forms take in different form fields that are specified as shown below

**Observation form**

This form is used by the observer to capture details read from the manual weather stations. The fields entered are then used to generate different reports

**Example of the manual observation form**

****

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Field description | Field type | Required? |
| Date | Describes the data entered capture date | Date | Yes |
| Station name | Describes the station name | varchar | Yes |
| Station number | Describes the station number | Int | Y |
| Time | Describes the time of data capture | Time in zulu | Y |
| Total amount of all clouds | Represents the amount of all clouds available | Int | Y |
| Total amount of low clouds | Describes the amount of low available clouds. It highyly depends on the number entered as the total amount of clouds | Int | Y |
| Type of low clouds | Describes the type of a low cloud | Int | Y |
| Oktas of low cloud | Describes the oktas of low cloud | Int | Y |
| Height of low cloud | Describes the height of the low cloud | Int | Y |
| Clcode of low cloud | Describes the clcode of the low cloud | Enum | Y |
| Type of medium cloud | Describes the type of the medium cloud | Int | Y |
| Oktas of medium cloud | Describes the oktas of medium cloud | Int | Y |
| Height of medium cloud | Describes the oktas of the medium cloud | Int | Y |
| Clcode of medium cloud | Describes the clcode of the medium | Enum | Y |
| Type of high cloud | Describes the type of the medium cloud | Int | Y |
| Oktas of high cloud | Describes the oktas of medium cloud | Int | Y |
| Height of high cloud | Describes the oktas of the medium cloud | Int | Y |
| Clcode of high cloud | Describes the clcode of the medium | Enum | Y |
| Cloud search light reading |  | integer | N |
| Rainfall | Describes the rainfall reading | integer | N |
| Dry bulb |  | Integer | Y |
| Wet bulb |  | Integer | Y |
| Max read |  | Integer | N |
| Max reset |  | Integer | N |
| Min read |  | Integer | N |
| Min reset |  | Integer | N |
| Piche read |  | integer | N |
| Piche reset |  | Integer | N |
| Time martks thermo |  | integer | N |
| Time marks hygro |  | integer | N |
| Time marks rain record |  | integer | N |
| Present weather |  | String | N |
| Visibility |  | Integer | Y |
| Wind direction |  | integer | Y |
| Wind speed |  | integer | Y |
| Gusting |  | integer | N |
| Attd thermo |  | integer | N |
| PrAsRead |  | integer | N |
| Correction |  | integer | N |
| CLP |  | integer | N |
| MSLPr |  | integer | N |
| Time marks Barograph |  | integer | N |
| Time marks anemograph |  | integer | N |
| Other TMarks |  | String | N |
| Remarks |  | String | N |

**Other fields**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Field description** | **Field type** |
| Time | Captures the UTC time for the record | Zulu time |
| Unit of wind speed | Captures an integer value for the wind speed | Int |
| indorOmissionOfprecipitation |  | Int |
| Type of station present past weather |  | Int |
| Height of lowest cloud |  | Int |
| Standard isobaric surface |  | Int |
| GPM |  | Int |
| Duration of period of preciptation |  | Int |
| Past weather |  | Int |
| Grass min temp |  | Int |
| CI of precipitation |  | Int |
| BE of precipitation |  | Int |
| Indicator of type of instrumentation |  | Int |
| Duration of sunshine |  | Int |
| Sign of pressure change |  | Int |
| Supp\_info |  | Text |
| Vapour pressure |  | Int |
| Wind run |  | Int |
| T\_H\_Graph |  | Int |

## Hardware Interfaces

**3.2.1 Mobile application**

Mobile devices: the application will run on all android devices supporting android version 4.1.1 and above

**3.2.2 desktop application**

Desktop devices: the application will run on all ubuntu, windows and mac platforms

## Software Interfaces

**3.3.1 Operating systems supported**

* **Mobile**: android version 4.1.1 and above
* **Desktop**: ubuntu, windows, mac (still needs identification)
* **Web**: all operating systems as long as the browsers are latest i.e. safari, chrome, mozilla

# Mobile app and Desktop app System Features

This section describes the different mobile app system features

## Data capture

**Description and Priority**

This feature allows data capture for observation form and more form fields.

It uses the pending uploads feature to determine internet connection availability for decisions on whether to upload the data online or offline

It provides forms for both observation and more form fields as described in the respective fields.

**Stimulus/Response Sequences**

It is triggered when a person logs in. When a person logs in it provides the different forms that the user enters data in and on submission it triggers the internet availability tester feature

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirements id** | **Description** |
| MW001 | This feature shall display the observation form |
| MW002 | This feature shall display the more form fields form |
| MW003 | This feature shall notify the users upon successful submission of data online |
| MW004 | This feature shall validate the user entered fields |

## Internet availability tester

**Description and Priority**

This feature tests for the availability of internet on the mobile device. The internet availability determines which course of action to take. The offline mode and online mode of the application is determined by this feature

**Stimulus/Response Sequences**

It is triggered when a user clicks to submit data entered in the capture data feature. If internet is not available, it triggers pending uploads feature else it triggers the GPS location tracker feature .

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirements id** | **Description** |
| MW005 | This feature shall determine the network state of the device |
| MW006 | This feature shall notify the users on their network state. |

## Pending uploads

**Description and Priority**

This feature will keep track of data stored in the mobile phone database that requires to be submitted incase internet is available

**Priority**: high.

**Stimulus/Response Sequences**

It is triggered when a user’s internet is off and a user needs to submit data.

It notifies the user that data is to be submitted offline

It then adds that data submitted to the pending uploads menu.

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirements id** | **Description** |
| MW007 | This feature shall store data offline incase internet is not available |
| MW008 | This feature shall provide a menu of pending uploads |
| MW009 | This feature shall allow a user to upload offline data incase internet is available |

## Login

**Description and Priority**

This feature prevents users to access the application without the authorized credentials.

**Priority**: high

**Stimulus/Response Sequences**

When a user launches the application, this feature is trigered and launched.

If a user enters no credentials, this feature notifies them

If a user enters wrong credentials , this feature notifies them

If a user enters correct credentials, this feature routes them to the data capture feature

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirement ID** | **Description** |
| MW010 | The application shall check user credentials before allowing them access to the application |
| MW011 | The application shall notify the user about wrong credentials entered in case the credentials entered do not match the real credentials |
| MW012 | The application notify users incase they don’t enter any credentials and try to login |
| MW013 | The feature shall route the user to the data capture section if the credentials are valid |
| MW014 | This feature shall detect offline state to determine if login should be made offline or online |

## GPS location tracking

**Description and Priority**

This component shall determine user location coordinates and match them with the station they are entering data for.

**Stimulus/Response Sequences**

It is triggered when a user clicks to submit data to the repository from the weather station

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirement ID** | **Description** |
| MW015 | The application shall track user location |
| MW016 | The application shall notify the user their location when a user tries to submit data while online |
| MW017 | The application shall limit the user on submission of data if user location doesn’t match the station they are submitting data for |

# Other Nonfunctional Requirements

## Performance Requirements

## Safety Requirements

## Security Requirements

All users will be required to login before accessing the application

The mobile application will allow data entry at only locations with same cordainates as the weather station

The

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>