(Team Name)

(AWS Monitor)

Software Design Document

Name (s):

Lab Section:

Workstation: WIMEA-ICT LAB

Date: (08/16/2020)

Table of Contents

1. INTRODUCTION

Automatic Weather Stations (AWSs) collect and transmit weather data without human intervention, enabling them to operate in remote areas. AWSs, which use Wireless Sensor Networks (WSNs) technology, in which distributed sensors collect varying parameters at predetermined intervals are the focus of this document. While in remote deployments, WSNs face challenges such as coverage, packet loss and limited energy among others, which lower their health and life time [1, 2]. Thus, the need to monitor these AWSs arises. The AWS Monitor wirelessly monitors the conditions of these AWSs so as to enhance the performance, health and life time.

1.1 Purpose

This software design document describes the architecture, system design and the way the modules of the Automatic Weather Station Monitor (AWS Monitor) modules interact to achieve a particular goal.

1.2 Scope

AWS Monitor is a web application for monitoring the conditions of the automatic weather stations deployed in the different parts of the country. AWS monitor is specifically to monitor the conditions of these AWS widely deploy in different areas on one computer and to inform concerned stakeholders about the issues.

- **Goal(s)**
  - To monitor the conditions of the automatic weather stations.
- **Objectives**
  - To determine aws performance.
  - To determine problems faced by aws.
  - To determine aws status.
  - To examine station data.
  - To extract and receive weather data from station data

- **Benefits**
    - Improve automatic weather stations performances.
    - Improve troubleshooting capacity to the technical team.
    - Improve response time in case a problem arises.
    - Aws monitor reduces costs compared to manual monitoring.
    - Easier to monitor widely spread stations allover a region.
    - Reduce downtime.

1.3 Overview

This document is written according to the standards for Software Design Documentation explained in "IEEE Recommended Practice for Software Design Documentation" [1].

Sections 3 – 5 contain discussions of the designs for AWS Monitor with diagrams, section 6 shows samples of UI from the system, and section 7 contains requirements matrix matched with components that satisfy the functional requirements. Section 8 contains appendices; they contain extra details about the algorithms applied in the AWS monitor modules.

1.4 Reference Material

References added in the appendix section.

1.5 Definitions and Acronyms

Term	**Definition**
AWS	Automatic Weather Station
Reports	Refers to what is received from the AWSs by the data receiver.
Node	Each aws consists of four "nodes", these contain a set of sensors on each of them performing different tasks.

Sensors	These are devices attached to nodes to detect weather parameters such as wind speed, precipitation etc.
Listener	Refers to one of the modules in change of data reception(reports).
Weather parameters	These are the instances of weather such as humidity, ultra violet light, etc.
WSN	Wireless Sensor Network

2. SYSTEM OVERVIEW

Background.

Automatic Weather Stations (AWSs) perform automatic collection and transmission of weather data. These AWSs face challenges, which lower their performance. Hence, a need for regular monitoring to reduce down time.

Overview.

The **AWS Monitor** performs condition monitoring, comprised of a *data receiver (Listener), analyzer, problem classifier* and *reporter & visualizer*, to mine data relationships, identify possible causes of problems and perform *reporting* of AWSs.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

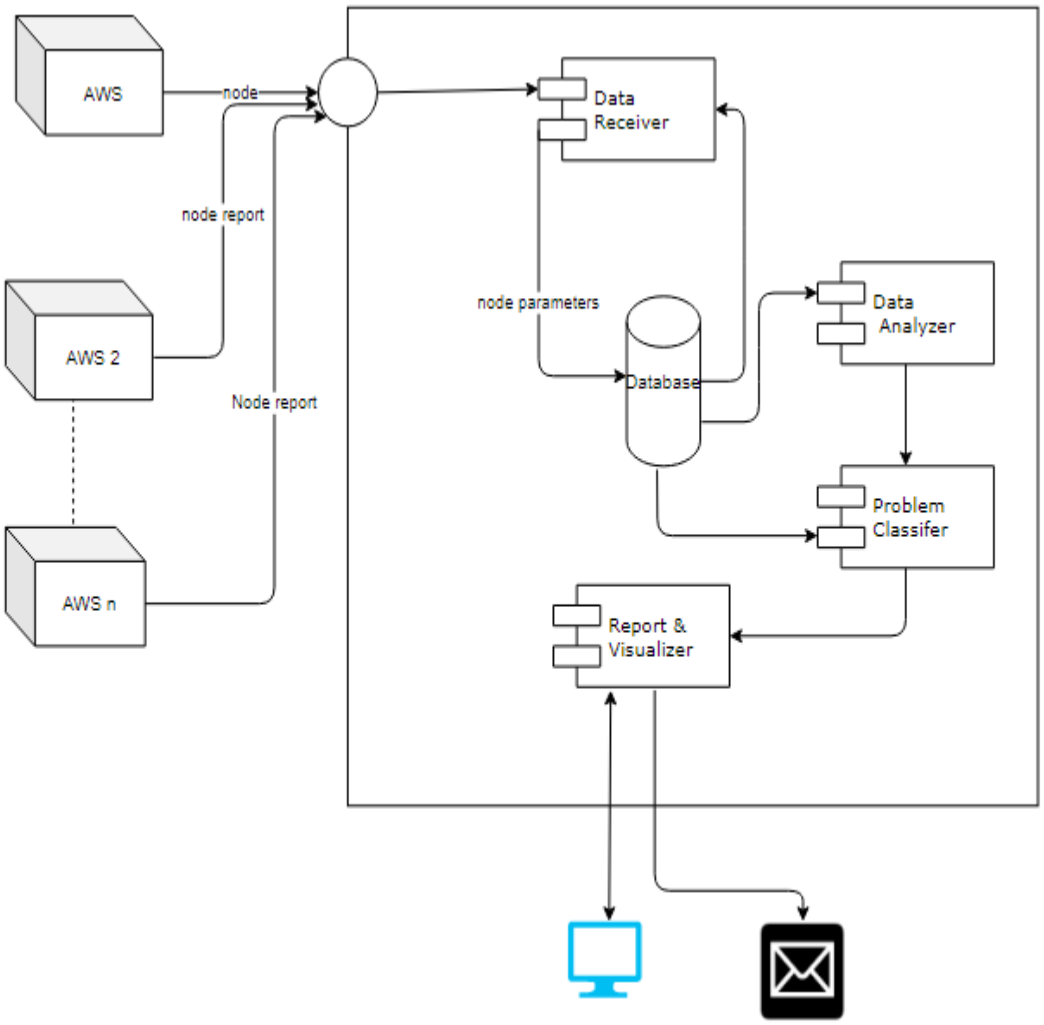

Figure 1 Architecture of AWS Condition Monitor running at the server.

AWSs

These perform automatic collection and transmission of weather data to a server port.

Data Receiver

The data receiver listens to the incoming TCP connection and receive data (weather data) from multiple AWS at roughly at time through a specified port. It writes this data to a file and to the database.

Data Analyzer.

This module traverses though the packets received from the aws for patterns and anomalies and as per a given AWS and determines if there are any problems with the data and AWSs.

Problem Classifier.

The problem classifier interacts with the analyzer to establish reason/causes of problems detected by the analyzers using condition monitoring algorithms [1].

Report & Visualizer.

After problems have been identified and classified by the analyzer and classifier respectively reports are sent to stakeholders under the mail list. Visuals are displayed on the web.

AWS Monitor web.

The web is where all the monitoring is down, performance is observed, user management among others.

3.2 Decomposition Description

Top level data flow diagram

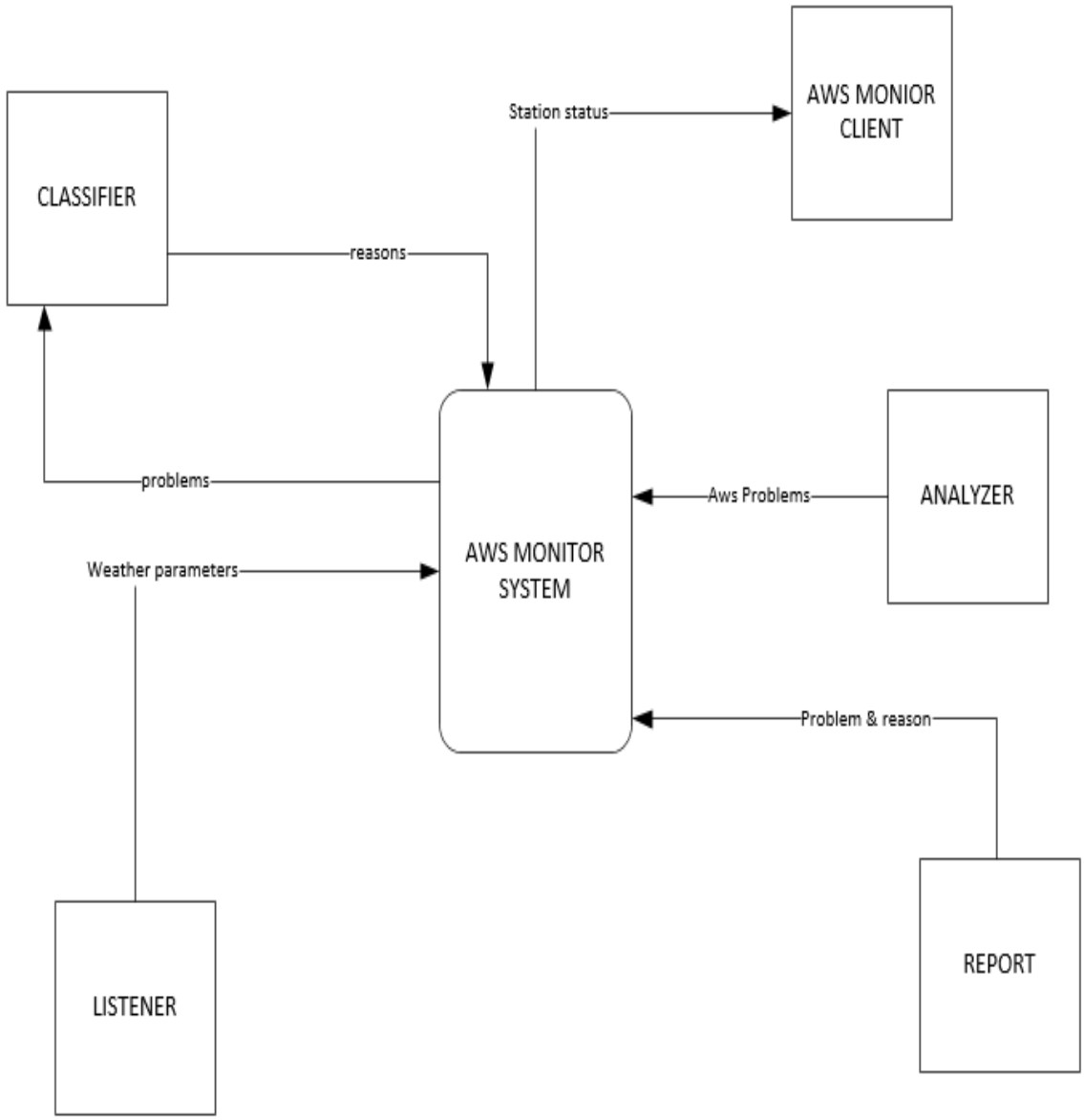

Figure 2 High level view of the AWS MONITOR.

Structural decomposition diagram

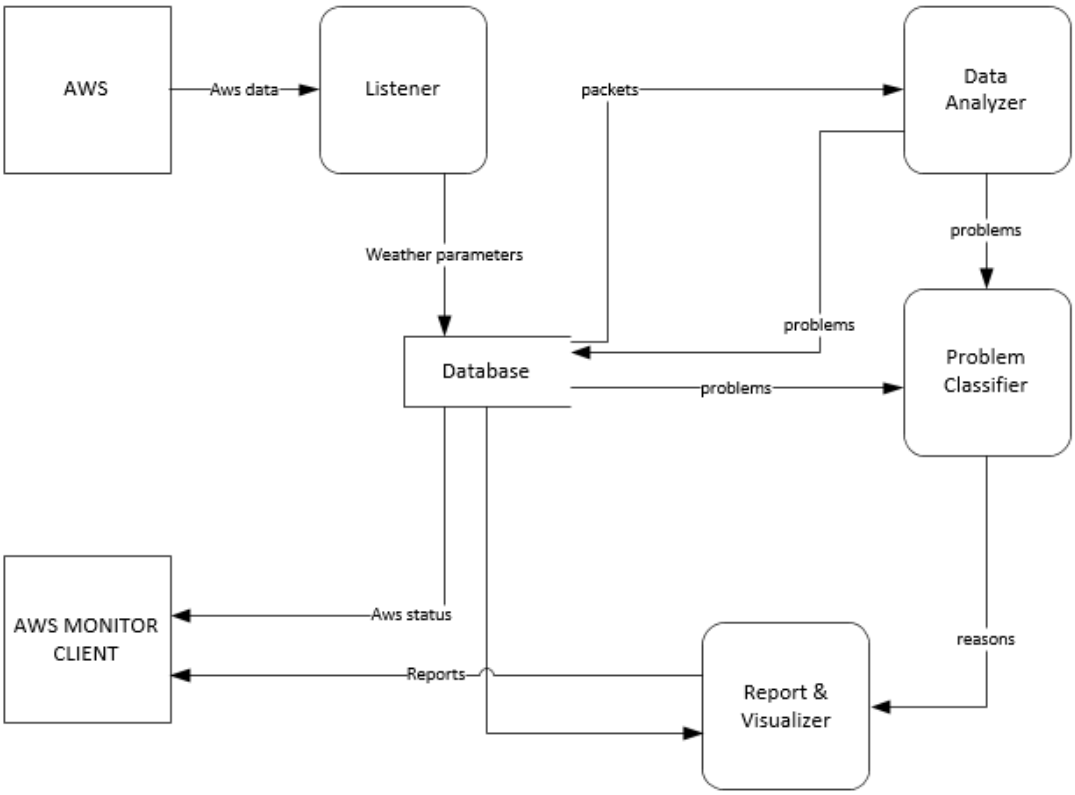

Figure 3 high level decomposition diagram.

3.3 Design Rationale

The monitoring process becomes more complex as the network of AWSs being monitored grows variations in AWS data. Furthermore, receiving big volumes of data centrally becomes more challenging as the data arrival and processing rates vary, causing data packets to drop. We have proposed an architecture in figure 1 for monitoring a network of AWSs distributed over a wide area, consisting of a data receiver which, receives and stores data at rates as low as 1ms using M/M/1/k queuing model. The data receiver is able to perform an infinite number of parallel connections at the same time, hence facilitating reception of packets from many AWSs at the same time [2].

4. DATA DESIGN

4.1 Data Description

The reports from the AWSs are received by the listener and written to files and a MYSQL database. Reports are directly stored in .dat file data structure format and weather parameters are distinctively stored in the database using simple MSQL queries.

4.2 Data Dictionary

TABLE	COLUMN	TYPE	NULL	DEFAULT
REPORTINTERVALCLUSTERS	***id***	Int (11)	No	
	stationID	Int (20)	No	
	Node	Varchar (200)	No	
	time_of_last_running_analyzer	Varchar (200)	No	
	cluster	Varchar (10000)	No	

GENERALTABLE	***id***	int (200)	No	
	stationID	int (11)	No	111
	RTC_T	Varchar (200)	Yes	NULL
	V_BAT	Varchar (200)	Yes	NULL
	SOC	Varchar (200)	Yes	NULL
	V_MCRTC_T	Varchar (200)	Yes	NULL
	REPS	Varchar (200)	Yes	NULL
	DATE_OF_DB_INSERTION	Varchar (200)	Yes	NULL
	TIME_OF_DB_INSERTION	Varchar (200)	Yes	NULL
	stationname	Varchar (200)	Yes	NULL

	hoursSinceEpoch_of_db_insertion	double	Yes	NULL
	DATE	Varchar (200)	Yes	NULL
	TIME	Varchar (200)	Yes	NULL
	hoursSinceEpoch	Varchar (200)	Yes	NULL
USERS	***id***	int (10)	No	
	name	Varchar (191)	No	
	email	Varchar (191)	No	
	phone	Varchar (30)	No	
	password	Varchar (191)	No	
	admin	tinyint (1)	No	0

	remember_token	Varchar (100)	Yes	NULL
	created_at	timestamp	Yes	NULL
	updated_at	timestamp	Yes	NULL
STATIONS	***station_id***	int (11)	No	
	StationName	Varchar (30)	No	
	StationNumber	Varchar (30)	Yes	NULL
	StationRegNumber	Varchar (30)	Yes	NULL
	Location	Varchar (50)	No	
	Indicator	Varchar (30)	Yes	NULL
	StationRegion	Varchar (30)	No	

	Country	Varchar (30)	No	Uganda
	Latitude	double	No	
	Longitude	double	No	
	Altitude	double	Yes	NULL
	StationStatus	Enum ('on', 'off')	No	on
	StationType	Varchar (30)	No	
	Opened	datetime	Yes	NULL
	Closed	date	Yes	NULL
	SubmittedBy	Varchar (30)	Yes	NULL
	Creation_Date	datetime	No	CURRENT_TIMESTAMP
	UpdateDate	timestamp	Yes	NULL

	stationCategory	Enum ('manual', 'aws')	No	
GROUNDNODE	***id***	int (200)	No	
	stationID	int (11)	No	111
	RTC_T	Varchar (200)	Yes	NULL
	NAME	Varchar (200)	Yes	NULL
	E64	Varchar (200)	Yes	NULL
	V_A1	Varchar (200)	Yes	NULL
	V_A2	Varchar (200)	Yes	NULL
	P0_LST60	Varchar (200)	Yes	NULL

	T1	Varchar (200)	Yes	NULL
	RSSI	Varchar (200)	Yes	NULL
	TTL	Varchar (200)	Yes	NULL
	LQI	Varchar (200)	Yes	NULL
	SEQ	Varchar (200)	Yes	NULL
	UP_TIME	Varchar (200)	Yes	NULL
	T	Varchar (200)	Yes	NULL
	V_IN	Varchar (200)	Yes	NULL
	V_MCU	Varchar (200)	Yes	NULL
	DATE	Varchar (200)	Yes	NULL

	TIME	Varchar (200)	Yes	NULL
	Parameter checked	Varchar (200)	No	false
	Trend checked	Varchar (10)	Yes	false
	Ignore on calc trend	Varchar (10)	No	false
	hoursSinceEpoch	double	Yes	NULL
DETECTEDANALYZERPROBLEMS	***id***	int (11)	No	
	Problem	Varchar (200)	Yes	NULL
	Value	Varchar (20)	Yes	NULL
	NodeType	Varchar (255)	Yes	NULL
	stationID	int (255)	Yes	NULL

	when_reported	timestamp	No	CURRENT_TIMESTAMP
	status	Varchar (200)	No	reported
SINKNODE	***id***	int (200)	No	
	stationID	int (11)	No	111
	RTC_T	Varchar (200)	Yes	NULL
	NAME	Varchar (200)	Yes	NULL
	E64	Varchar (200)	Yes	NULL
	P_MS5611	Varchar (200)	Yes	NULL
	V_IN	Varchar (200)	Yes	NULL
	V_MCU	Varchar (200)	Yes	NULL

	UP_TIME	Varchar (200)	Yes	NULL
	T	Varchar (200)	Yes	NULL
	SEQ	Varchar (200)	Yes	NULL
	DATE	Varchar (200)	Yes	NULL
	TIME	Varchar (200)	Yes	NULL
	parameterChecked	Varchar (200)	No	false
	Ignore on calc trend	Varchar (10)	No	false
	Trend checked	Varchar (10)	No	false
	hoursSinceEpoch	double	Yes	NULL
TWOMETERNODE	***id***	int (200)	No	

	stationID	int (11)	No	111
	RTC_T	Varchar (200)	Yes	NULL
	NAME	Varchar (200)	Yes	NULL
	E64	Varchar (200)	Yes	NULL
	T_SHT2X	Varchar (200)	Yes	NULL
	checked	Varchar (20)	No	default-value
	RH_SHT2X	Varchar (200)	Yes	NULL
	checkr	Varchar (200)	No	default-value
	RSSI	Varchar (200)	Yes	NULL
	TTL	Varchar (200)	Yes	NULL

	LQI	Varchar (200)	Yes	NULL
	V_IN	Varchar (200)	Yes	NULL
	V_MCU	Varchar (200)	Yes	NULL
	SEQ	Varchar (200)	Yes	NULL
	UP_TIME	Varchar (200)	Yes	NULL
	T	Varchar (200)	Yes	NULL
	DATE	Varchar (200)	Yes	NULL
	TIME	Varchar (200)	Yes	NULL
	Parameter checked	Varchar (200)	No	false
	Trend checked	Varchar (10)	No	false

	Ignore on calc trend	Varchar (10)	No	false
	hoursSinceEpoch	double	Yes	NULL
TENMETERNODE	***id***	int (200)	No	
	stationID	int (11)	No	111
	RTC_T	Varchar (200)	Yes	NULL
	NAME	Varchar (200)	Yes	NULL
	E64	Varchar (200)	Yes	NULL
	V_A1	Varchar (200)	Yes	NULL
	V_A2	Varchar (200)	Yes	NULL
	P0_LST60	Varchar (200)	Yes	NULL

	V_AD1	Varchar (200)	Yes	NULL
	V_AD2	Varchar (200)	Yes	NULL
	RSSI	Varchar (200)	Yes	NULL
	TTL	Varchar (200)	Yes	NULL
	LQI	Varchar (200)	Yes	NULL
	SEQ	Varchar (200)	Yes	NULL
	V_IN	Varchar (200)	Yes	NULL
	V_MCU	Varchar (200)	Yes	NULL
	UP_TIME	Varchar (200)	Yes	NULL
	T	Varchar (200)	Yes	NULL

	DATE	Varchar (200)	Yes	NULL
	TIME	Varchar (200)	Yes	NULL
	Parameter checked	Varchar (200)	No	false
	Trend checked	Varchar (10)	No	false
	Ignore on calc trend	Varchar (10)	No	false
	hoursSinceEpoch	double	Yes	NULL

5. COMPONENT DESIGN

Data Receiver Algorithm.

1: listen to for incoming TCP connections on one port.

2: receive AWSs Reports.

3: Traverse through the database.

3.1: If StationName exists.

3:2 go to 4.

3.3: if StationName does not exist.

3.4: create new station.

4: write to file

5: insert into database.

6: repeat 1-5.

Analyzer Algorithm.

Algorithm 1: Successive Pairwise Record Differences (SPREDs)

1: aws_sensor_list, missed_node_count =0

2: if exists (new_aws_record)

3: reference_parameters = get_reference_parameter_list

4: while (exists more tokens in model_tokens list)

5: if (reference_parameter not found in new list)

6: record_sensor_level_miss (time, parameter)

7: node_record_time_diff = last_record_time – previous_record_time

8: if (node_record_time_diff not equal to previous diff)

9: record_change (old_ddiff, new_diff, time)

10: update_cluster_list (cluster_name, cluster_count)

11: if (missed_node_count equals len (aws_sensor_list))

12: record_aws_miss (aws_name)

Reporter.

1: New & confirmed problem*: There is supporting evidence of a new problem. Then report problem.*
2: Reported/re-reporting problem*: problem has been reported once and if it still persists, there is a chance of reporting it again.*
3: Persisting problem*: The number of times a problem has been reported has exceeded the threshold and reporting of the same has stopped.*

4: Fixed & archived problem*: Problem has been resolved and information stored for future reference.*

Classifier and visualizer

Details about the classifier and visualizer reference [2] .

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The aws monitor has two kinds of administrators namely: admin and super admin. The super admin has access to some pages that causal admin does not have access to. The super admin basically monitors the causal admin but can also access the pages the causal admin access. The causal admin only has access to the features limited to him/her which are enough to carry out the real core (Aws condition monitoring) of the system.

Super Admin.

The super admin login into the system with super admin credentials provided by the system developer(s). The only unique task/feature the super admin does that makes him/her superior, is the ability to add new users of the system and monitoring them otherwise the features remain the same for both users.

Admin

The admin (causal) gains access to the system through the super admin, who has to create an account for the admin. The causal admin logins into the system with credentials provided by the super admin. From that point the user can access any information on any of the AWS deployed in different parts of the country at the comfort of his/her desk. Some of the features the user will explore include map showing aws deployment, aws statuses, aws performance, aws problems among others.

6.2 Screen Images

Figure 4 below is a welcome page with login for both kinds of users for the AWS Monitor. It shows all the stations deployed by WIMEA-ICT all over the country.

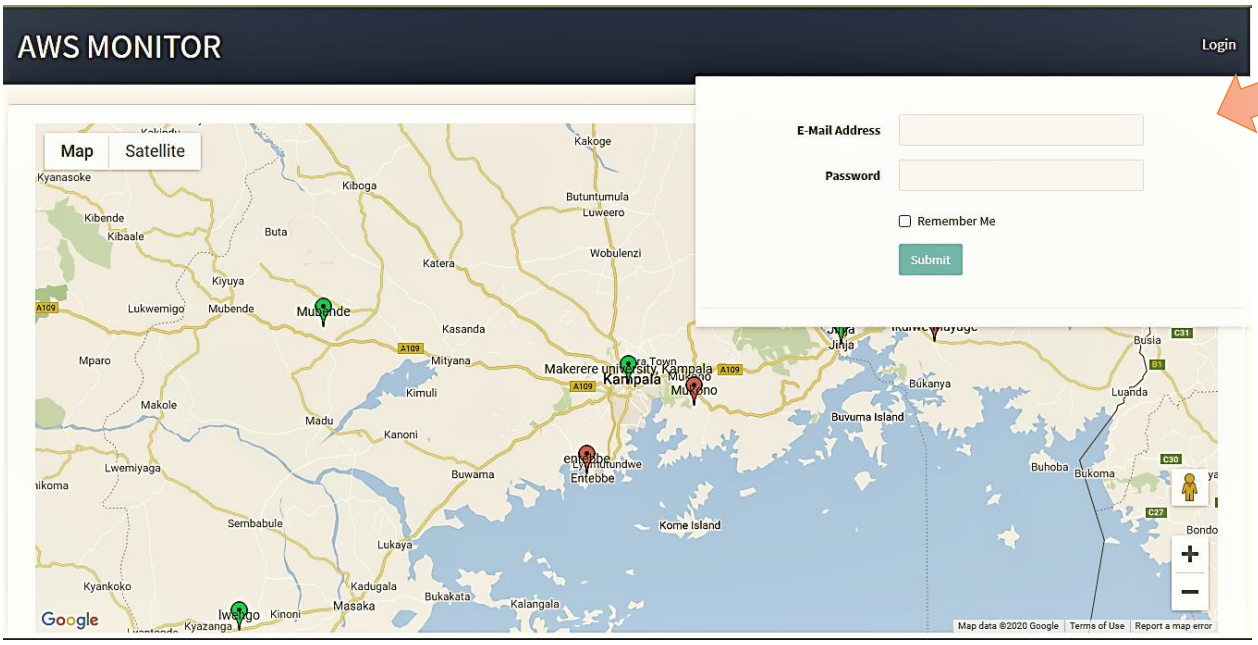

Figure 4 AWS Monitor welcome page.

Figure 5 the users after logging in with their respective credentials can now access the system features, monitor the conditions of the deployed AWSs.

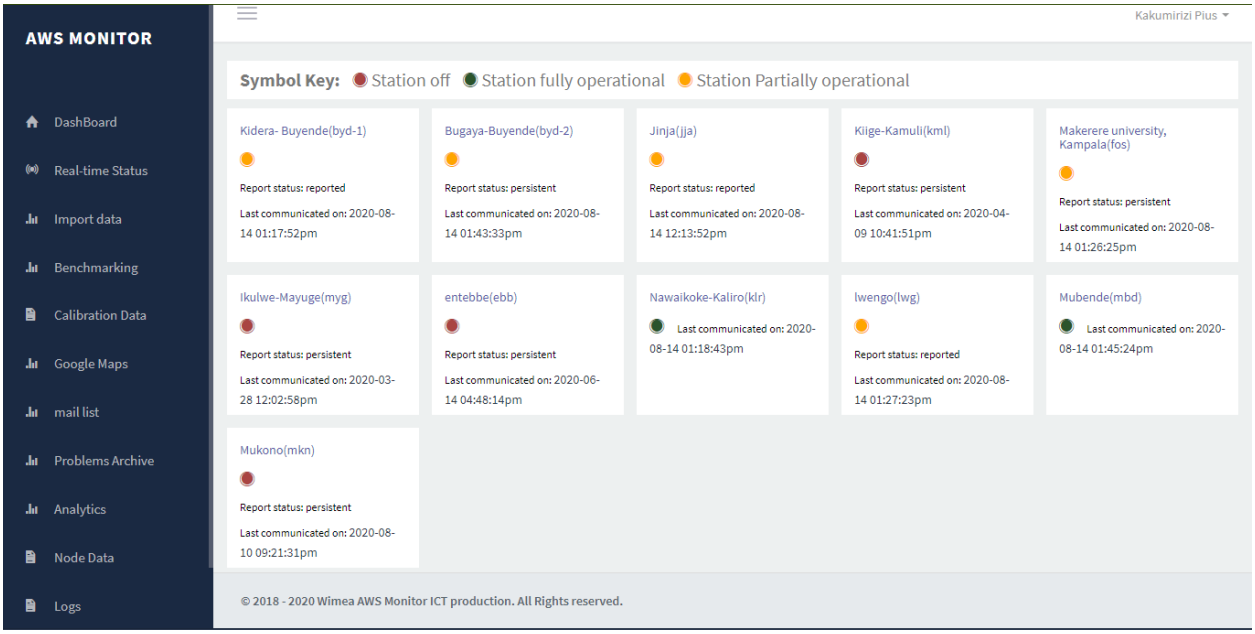

Figure 5. AWS status.

7. REQUIREMENTS MATRIX

Function Requirement	Listener Module	Analyzer Module	Classifier Module	Report Module	Web Monitor
Authenticate users					X
Receive Reports	X				
Extract Weather parameters	X				
Analyze reports		X			
Classify problems			X		
Visualize		X			X
Report Problem				X	
Show Station status					X
AWS performance		X	X		X
Manage mail list					X
Manage Users					X
Validate Data		X	X		

AWS statistics		X	X		X

8. APPENDICES

References

[1] IEEE, *IEEE Std 1016-1998 Recommended Practice for Software Design Descriptions*, 1998-09-23, The Institute of Electrical and Electronics Engineers, Inc., (IEEE).
IEEE, *IEEE 1016* Software Design Document (SDD) Template for CENG491

[2] J. S. O. ,. R. N. A. ,. G. N. R. M. M. B. a. B. P. Mary Nsabagwa, "Condition Monitoring for Wireless Sensor Network- Based Automatic Weather Stations," *EAI Endorsed Transactions on Internet of Things,* vol. I, pp. 1-11, 2018.