

**UNIVERSITY**

**MAKERERE**

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

DEPARTMENT OF NETWORKS

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

BSE 4100: SOFTWARE ENGINEERING PROJECT 2

SOFTWARE DESIGN DOCUMENT (SDD) FOR:

**A SECURE LAYER FOR DESKTOP-WEATHER DATA REPOSITORY**

**PROJECT MEMBERS [GROUP BSE20-4]**

|  |  |  |
| --- | --- | --- |
| **NAME** | **REG NO** | **STUDENT NO** |
| ASINGWIRE DALLINGTON | 16/U/127 | 216000708 |
| ISABIRYE TAIBU | 16/U/20495 | 216022278 |
| KUNYA NATHERN | 16/U/6372/EVE | 216006158 |
| KAGANZI TIMOTHY | 16/U/5134/EVE | 216007287 |

**SUPERVISOR**: MARY NSABAGWA

Table of contents

[1. Introduction 1](#_Toc34759510)

[1.1. Purpose 1](#_Toc34759511)

[1.2. Scope 1](#_Toc34759512)

[1.3. Document Overview 1](#_Toc34759513)

[1.4. Reference Material 2](#_Toc34759514)

[1.5. Definitions and Acronyms 2](#_Toc34759515)

[2. System Overview 3](#_Toc34759516)

[3. System Architecture 5](#_Toc34759517)

[3.1 Architectural design 5](#_Toc34759518)

[3.2 Decomposition Description 6](#_Toc34759519)

[3.2.1 Log user actions component 7](#_Toc34759520)

[3.2.2 Data encryption component 8](#_Toc34759521)

[4. Data Design 10](#_Toc34759522)

[4.1 Data Description 10](#_Toc34759523)

[4.2 Data Dictionary 10](#_Toc34759524)

[5. Component Design 14](#_Toc34759525)

[5.1 Log user actions component 14](#_Toc34759526)

[5.2 Data encryption component 14](#_Toc34759527)

[5.3 user level data access 14](#_Toc34759528)

[5.4 Tracking the typing patterns of the user 15](#_Toc34759529)

[6. Human Interface Design 16](#_Toc34759530)

[6.1 Overview of User Interface 16](#_Toc34759531)

[6.2 Screen Images 16](#_Toc34759532)

[6.3 Screen Objects and Actions 17](#_Toc34759533)

[7. Requirements Matrix 19](#_Toc34759534)

**Table of Figures**

[Figure 3.1. 1. client server architecture for a security layer of the WDR system. 8](#_Toc34757922)

[Figure 3.2. 1. Levels of Security 9](#_Toc34757926)

[Figure 3.2. 2 log user actions sequence diagram 11](#_Toc34757928)

[Figure 3.2. 3 sequence diagram for encrypt data component 11](#_Toc34757929)

[Figure 3.2. 4 Sequence diagram for add user levels and roles component 12](#_Toc34757930)

[Figure 4.1. 1 Data design for the weather data repository system 13](#_Toc34757931)

[Figure 6.2. 1 Search user logs 18](#_Toc34757936)

[Figure 6.2. 2 User logs interface 19](#_Toc34757937)

[Figure 6.2. 3 Add New User Role Interface 19](#_Toc34757938)

**List of Tables New page for each section**

[Table 1 Acronyms 2](#_Toc34759560)

[Table 2 Description of the different levels of security. 7](#_Toc34759561)

[Table 3: User Entity for inserting a new system user 10](#_Toc34759562)

[Table 4: Role Entity for the system roles 11](#_Toc34759563)

[Table 5: User logs Activity for storing user logs 11](#_Toc34759564)

[Table 6: Observation slip activity for storing weather data 12](#_Toc34759565)

[Table 7: Screen objects and their actions 17](#_Toc34759566)

[Table 8 A requirements traceability Matrix 19](#_Toc34759567)

# Introduction

## Purpose

This Software Design Document (SDD) specifies the architecture and system design of a secure layer for Desktop-Weather Data Repository (WDR)[1] system. The WDR is a system used to collect and store weather data from both manual and automatic weather stations. The system shall enhance the security of weather data stored in the desktop version of WDR, a system which shall be used by the Uganda National Meteorological Authority (UNMA)[2]. The intended audience includes; system developers, system testers, project managers, system maintainers and UNMA staff.

## Scope

The system shall implement a secure layer for the desktop version of the WDR. The project will cover security features including logging user activities, weather data encryption, access control through user roles and levels among other security measures.

The objectives of the project are as follows.

1. To design a weather data encryption algorithm in order to secure data stored in the database.
2. To design keystroke monitoring algorithm which shall verify the user identity.
3. Are these all?

The system shall offer the following benefits to UNMA.

1. UNMA will use the system to track transactions performed on their data in order to establish the source of errors as a result of data input and modification.
2. The data shall be protected from malicious people since it will be encrypted and any authorized person who gains access to it will not be able to make sense out of it.
3. The system will help to prevent unauthorized access to data through keystroke monitoring ?? Revise
4. What about data integrity?

## Document Overview

The document how many?? sections. First is an introduction, which contains the purpose of the document, scope, reference material, definitions and acronyms. The second ???revise which describes the system being proposed. Section three is the system architecture which has the architectural design and decomposition description of the system. Section four is the data design, which has data description and dictionary. Section five is component design which has an overview of the components of the proposed system. Section six gives the human interface design which contains an overview of the human interfaces, screen images, screen objects and actions of the system intended to be developed. Section seven is the requirements matrix which is used to trace system requirements.

## Reference Material

[1] WIMEA-ICT, “No Title,” 2020. [Online]. Available: http://wimea.mak.ac.ug/wdr/. [Accessed: 10-Mar-2020].

[2] UNMA, “Uganda National Meteorological Authority (UNMA, 2016 and 2017). Climate data recorded in 2016 and 2017,” 2017.

## Definitions and Acronyms

Table 1 Acronyms

|  |  |
| --- | --- |
| **Acronym/Abbreviation** | **FULL FORM** |
| SDD | Software Design Document |
| UNMA | Uganda National Meteorological Authority |
| WDR | Weather Data Repository |
| AES | Advanced Encryption Standard |

# System Overview new chapters start on new pages!!! Simple instruction

As UNMA (Uganda National Meteorological Authority)[2] moves to a desktop version of the weather data repository, security of this data isincreasingly becoming of concern. Being in a distributed environment, this data is prone to improper manipulations. This data may further be illegally distributed, hence misleading the public through wrong predictions. Because of the importance of weather data, there is need for mechanisms to secure this data from the time it is collected to the time it is used for forecasting and prediction. This set up comes with its challenges. For instance, unlimited system access, denial of service by the authorized roles, and insider attacks among others.

We are proposing a what??? which will improve security of weather data accessed and manipulated via the desktop version of the WDR. the security of the system will be implemented in three levels.

Level 1 will provide user access control. That is, through user authentication, role-based access control and session timeouts. Keystroke dynamics which will involve the use of an AI model to keep track of the user typing patterns. The ?? *which one?? You never talked about models in the earlier sentences* will have to retrieve the user’s typing speed and pattern and compare them as the user is typing to login. Through role-based access control, users will be assigned roles and through those roles, perform specific tasks on data.

Level 2, also known as….. will provide data protection through data encryption and data validation checks. Advanced Encryption Standard (AES) data encryption will be used to secure the data in forms which are not easily understood. The weather data will be encrypted as it is stored into the database and decrypted when the user is granted access after authentication. The user provides the key for the data to be decrypted.

Level 3, also known as activity tracking will be achieved through activity logging and suspicious behaviour reporting. With activity logging, the different activities that will be performed by a user will be recorded in logs. The logs shall enable all responsible persons to view a summarised list of data access and manipulation activities.

the activities that will be logged include:- log on attempts both successful and unsuccessful, account changes that is account creation and deletion with the person who carried out the activity is to be logged, the various tasks performed by authenticated users, updating of data and inserting of new data.

The number of successful and failed application authentication attempts per user when logging into the system. The information that will be logged for the different activity will be timestamp of the system, user account associated with the event, user’s IP address, error codes, event carried out, the result of the event that was carried out.

# System Architecture

## Architectural design

The system will follow client-server architecture where by the client is represented by the user interface.



Figure 3.1 1 client server architecture for a security layer of the WDR system.

Each user, through a client WDR desktop application sends requests to its instance of local database server. The server handles those requests and sends a response back to the client. When a client is connected to internet, the data on the local database server is directly synced with the data on the remote main database server. Otherwise, the client has to wait until it gains Internet connection to sync with the revise database.

the system shall consist of the following modules:-

* User access control module: This module is used to authenticate users so that only authorized users can access the application
* Data protection module: Data protection module validates, encrypts and decrypts weather data
* Activity logging module

This module logs every activity done by any application user to both a secured file and database.

The above modules are combined to provide a secure layer for the desktop weather data repository application.

## Decomposition Description



Figure 3.2. 1 Levels of Security

Table 2 Description of the different levels of security.

|  |  |
| --- | --- |
| Arrow Number | Description |
| 1 | The arrow shows the flow that after user authentication, there is data validation check for the data that is being recorded. The weather data that is recorded has to be consistent thus the validation checks. |
| 2 | After data validation checks, user activites are logged. This will help to track of the different actions performed by the user to help in audit tracking of the system. |
| 3 | ? there is encryption of the data that is to be stored in the database. This helps to provide more security of the data stored in the database, thus limiting the number of people who can access the data. |
| 4 | The arrow shows that after authentication, there is logging of the activity of authentication that is if the authentication was successful or not, the number of times the authentication failed. |
| 5 | after authentication and the user fails to login several times, that is more than five times. It is the recorded as suspicious behavior. |
| 6 | The arrow shows that if any suspicious behavior is detected, it is then logged. |

### Log user actions component

Logging user activities involves keeping track and storing the state and outcome of all activities performed by the user. The sequence diagram in figure where??? Get serious!!! shows the flow of data and actions for this component. When the user performs an action on system data, a new log record is created and inserted into the database. This record includes the action performed plus the data variables changed.

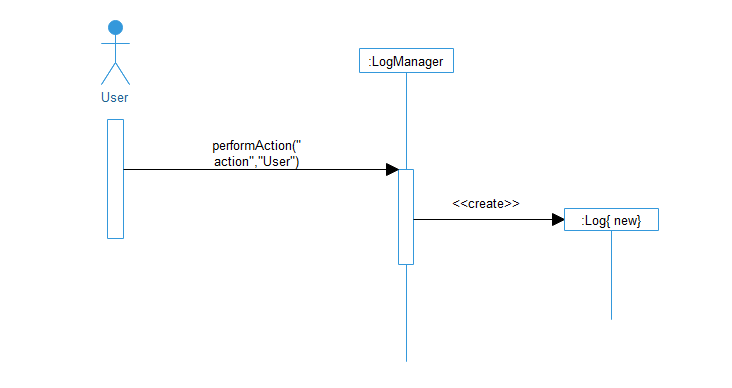


Figure 3.2. 2 log user actions sequence diagram

### Data encryption component

Data encryption will involve translation of data into another form, or code, so that only people with access to a secret key (the decryption key) can read it. When a user enters data into a form and clicks on the submit button. The encryption component encrypts the data and inserts the encrypted data into the database. The sequence diagram in figure 3 illustrates this.

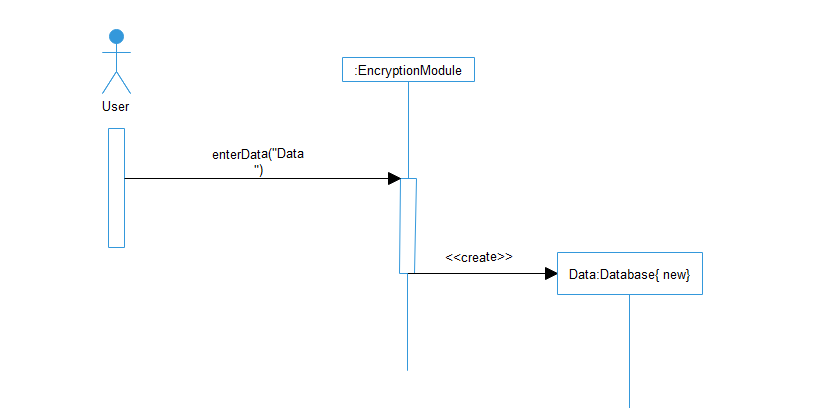


Figure 3.2. 3 sequence diagram for encrypt data component

* + 1. Create user levels and roles.

The system has different users each with rights to access given data according to their responsibility. When the user clicks add new user, the system loads a form for the user to enter data among which the user select the role of the new user being created. The system do yoy already have this. Check your tenses a user record in the database and notifies the user creating the new user. The sequence diagram in figure where??? What is wrong with you? illustrates the flow of data and actions.

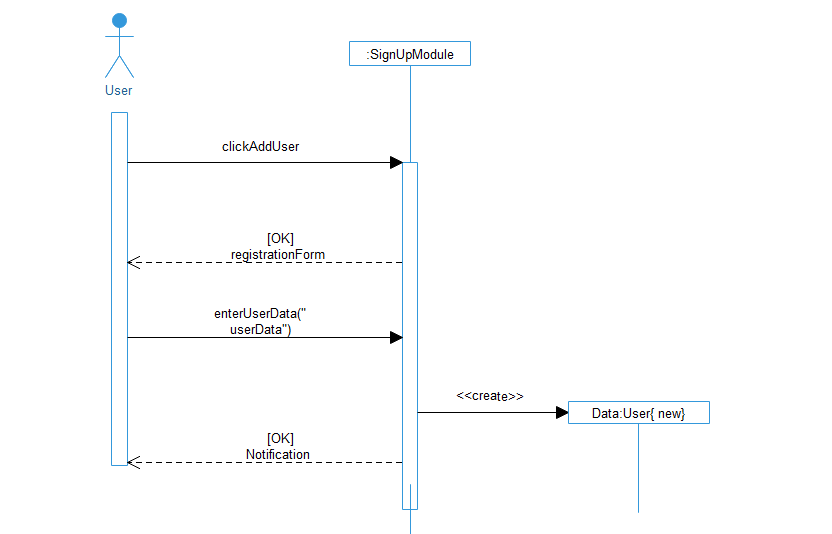
Delete figure its useless . 

Figure 3.2. 4 Sequence diagram for add user levels and roles component

# Data Design

## Data Description

Since the weather data repository system is an information-based system, the users of the system are expected to create, manipulate, update and delete data. This section therefore gives a description of the data structures of the system.

Fisrst describe and list all the entities and attributes here before you draw the ERD

Described here is the database structure composed of the Station, UserRole, User, Userlog and observationSlip tables. Fields indicated PK are primary keys for each table and fields marked FK are foreign keys.

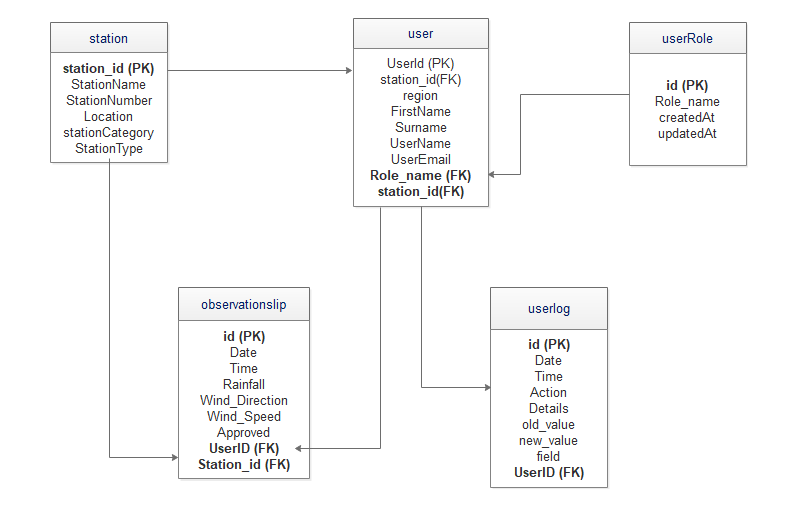


Figure 4.1. 1 ERD (oh my God!!!!!)

## 4.2 Data Dictionary

Table 3: User Entity for inserting a new system user

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Field Size for Display | Description |
| Id | integer | 11 | Unique identifier for each User |
| name | varchar | 255 | Users name |
| email | varchar | 255 | Users email address |
| password | varchar | 255 | Password for the user |
| createdAt | datetime |  | Date the user is created |
| updatedAt | datetime |  | Date the user is updated |

Table 4: Role Entity for the system roles

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Field Size for Display | Description |
| id | integer | 11 | Unique identifier for each user role |
| name | varchar | ever seen a name with very many characters? | Users role |
| createdAt | datetime |  | Date the user role is created |
| updatedAt | datetime |  | Date the user role is updated |

Table 5: User logs Activity for storing user logs

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Field Size for Display | Description |
| date | datetime |  | Date for the user log |
| id | integer | 11 | Unique identifier for the user log |
| Data\_id | integer | 20 | Unique identifier for each data item |
| userid | integer | 11 | Unique identifier for each user |
| Action | varchar | 50 | Action performed by the user e.g. login |
| details | Varchar | text | Details about the action performed |
| Ip address | Varchar | 25 | Ip address for each user |
| status | Enum (‘00’,’11’,’10’,’01’) |  | User log status |

Table 6: Observation slip activity for storing weather data

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Field Size for display | Description |
| id | integer | 11 | Unique identifier for each record |
| Date | datetime |  | Submission Date for the record entered |
| Time | varchar | 11 | Time in zulu…used by meteorological stations |
| Rainfall | varchar | 225 | Amount of rainfall collected from the rain gauge |
| Wind\_Direction | varchar | 255 | The direction of wind |
| Wind\_Speed | varchar | 255 | The speed of wind |
| Approved | varchar | 5 | Status of the observation slip (approved or not approved by the intended person) |
| TotalAmountOfAllClouds | int | 11 | Total amount of all clouds observerd at a particular time period |

# Component Design

This section provides the description of the components that constitute the architecture of the security layer of the weather data repository system. Sequence diagrams are used to describe the flow of control and data for each component.

## Log user actions component

Pseudo code

*If user is logged in and performs an action on data*

*Get user’s session identity*

*Get the modified data variables*

*Insert a record of modified variables and the user into a secured file and database*

*Print “Action performed successfully”*

*Else*

*Redirect user to login interface.*

## Data encryption component

Pseudo code

*While there is a new action performed on data*

*Read plain text x*

*Apply the advanced encryption standard algorithm to x to get xi.*

*Save xi into database*

*Else*

*Redirect user to login interface*

## user level data access

Pseudo code

*If user is logged in*

*If user role is x*

*Display specific dashboard (sub-menu) for user role x*

*Else*

*Redirect user to login interface.*

## Tracking the typing patterns of the user

*If user is logged inn and performs and action on data*

*Types in a text field.*

*System monitors the typing speed of the user.*

*The system retrieves the previous typing patterns*

*The system compares the current and retrieved typing patterns.*

*If**the current pattern matches the retrieved pattern.*

*All the user to continue the operation.*

***Else***

*System raises a warning to the manager data and mark the user’s data as suspected*

***Else***

*Redirect user to login interface.*

# Human Interface Design

## Overview of User Interface

The system will be a role-based system where the overall administrator is the Data manager. Data manager will be responsible for adding new system users and their roles as shown in figure

Every action performed by the user will be recorded and stored in the database. The user logs will be generated by the Data manager when needed.

The actions performed by the user will include login, logout, adding new observation slip (new weather data record), editing an observation slip and generating reports.

When logging into the system, the typing speed and pattern of a given user will be recorded and stored into the database. This pattern will later be used to compare the speed of the user will logging into the system.

If the user pattern does not match the one stored in the database, the user will be denied access to the system. This is to avoid malicious logins and system access by un authorized parties.

## Screen Images

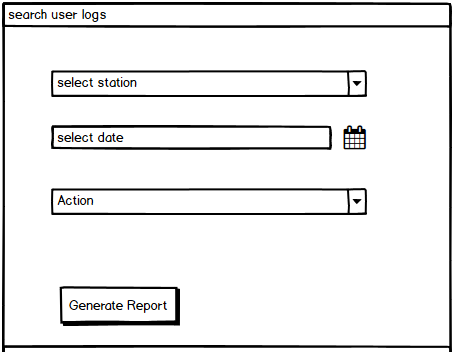


Figure 6.2. 1 Search user logs

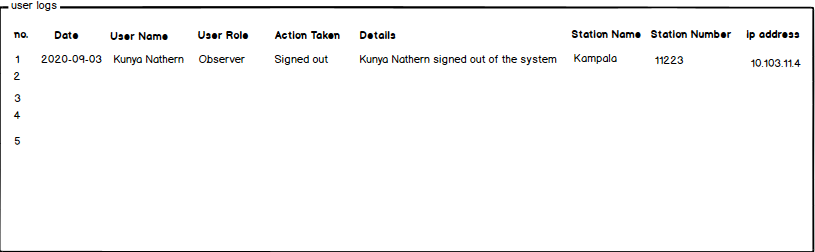


Figure 6.2. 2 User logs interface

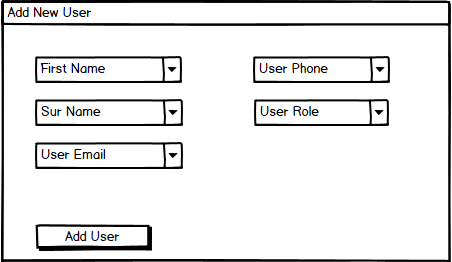


Figure 6.2. 3 Add New User Role Interface

## Screen Objects and Actions

Table 7: Screen objects and their actions

|  |  |  |
| --- | --- | --- |
| **Screen Object** | **Action** | **What happens on performing the action** |
| User name | User name of the person performing the action | Display user name of the person who performed the action |
| Action Taken | action performed by the user. For example, add/ edit action, login/logout etc | Display the action that was performed by the user |
| date | Date when the action was performed | Display the date when the action was taken |
| Station Name | Name of the station where the user is assigned. | Display the station name from which the action was performed |
| User role | The role played by the user in the system. For example, Observer. | Display the user role that performed the action |
| Details | What exactly the user did in the system. For example, added new observations lip, signed out | Display the actual action performed by the user |
| User email | Enter user email where the user logins will be sent for the user role being created | User email entered, verified  And stored into the database |
| User phone | Enter the phone number for the user being created | User phone number stored |
| Ip address | The internet protocol address of the user performing the action within the system | Display the ip address from which the action was performed |
| Station Number | The number of the station from which the user performed the action | Display the station number from which the action was performed |

# Requirements Matrix

Table 8 A requirements traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Business Use Case | Functional Requirement | | Priority | Test Case | |
| Requirement Id | Description | Id | Description |
| Logging of user activity | Req-1 | Authenticate all users before they access the system | High | TC01 | Verify if user is authenticated as they access the system |
| TC02 | Verify if user with wrong login details is not granted access |
| Req-2 | Logging of all user activities | High | TC03 | Verify if all user activities are logged |
| Encryption of data | Req-1 | Allow user enter raw data | Medium | TC04 | Verify if user provides correct input data |
| Req-2 | Encrypt data entered | High | TC05 | Verify that the system is able to encrypt data provided by user |
| Req-3 | Save encrypted data to the database | High | TC06 | Verify that the encrypted data is saved appropriately |
| Create user levels and roles | Req-1 | Allow data manager register new system users | High | TC07 | Verify if the user is able to register new users |
| Tracking of user typing patterns | Req-1 | Save user typing patterns | Medium | TC08 | Verify if the system is able to save the different user typing patterns |
| Req-2 | Retrieve and compare the typing pattern | High | TC09 | Verify if the system is able to identify a given user based on their typing pattern |
| Req-3 | Raise a warning in case of an unknown user | Medium | TC10 | Verify if the system is able to provide an alert in case of any intrusion |