**EGERTON** **UNIVERSITY**

**SYSTEM DESIGN DOCUMENT**

**FOR**

**ONLINE STUDENT CLEARANCE SYSTEM**

**PREPARED BY: MBOYA ALBERT ONYANGO**

**SUPERVISOR: MR. MUNGAI**

**COORDINATOR: DR. WILFRED GIKARU**

**05/04/2017**

# **INTRODUCTION**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the Web-based Clearance System project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

## **Purpose and Scope**

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

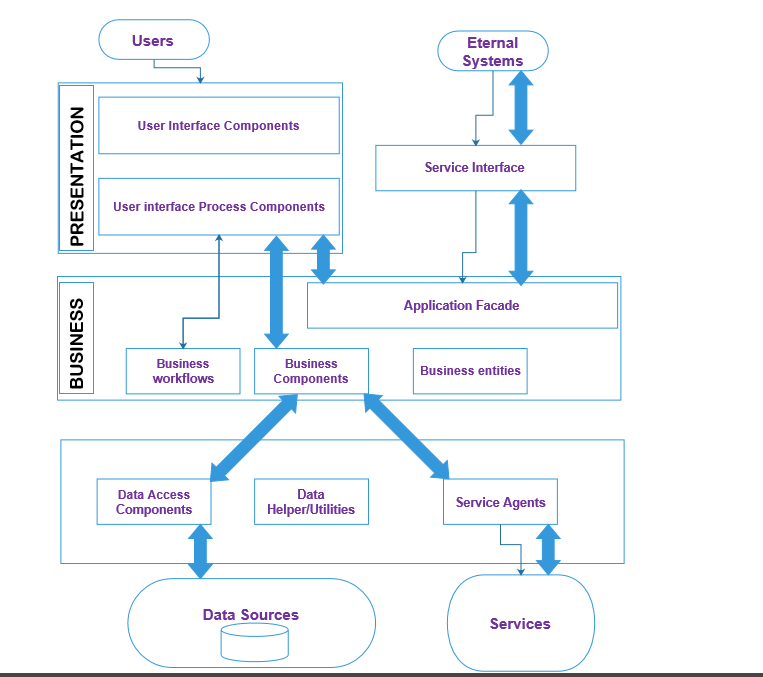
## **Project Executive Summary**

The online clearance system is an internet based system that will help ease the queuing system in the university’s clearance process. It is a web-based system that is designed to store and analyze information in a networked platform. Online student clearance system will build an effective information management system that is very convenient to use in universities since it is internet based and can be accessed from anywhere. The system will serve as a more reliable and effective means of undertaking students clearance, remove all forms of delay and stress as well as enable a student to know how to do his or her clearance online.

This application’s design is based on the Web Framework(WF) or web Application Framework(WAP). This framework enables application developers to design an application that automates the overhead associated activities performed in web development.

### **System overview**

This section describes the requirements of the distinct proposed software architecture. This system is a web-based application thus it is a client-server software application in which the client (user interface) runs in a web browser while the application’s resources and database resides in the web server. The system’s users can then access the application through a Web browser or a specialized web agent. The browser creates HTTP requests for the specific URLs that map the resources on the Web Server. The server renders and returns HTML and JSP pages to the client, which the browser can then display to the system user.



## **Fig. 1**

## **1.2.2 Design Constraints**

Several constraints are beyond the scope of the Clearance Application System and should be carefully factored into the system design. To date, the following constraints have been identified:

* The software to be built should take advantage of open source libraries and supporting software, such as databases and web containers, unless an adequate open source product is not available or creatable for use.
* The system must be compatible with existing infrastructure in the University to include network tools and applications, security requirements, server capabilities, and network management hardware. These constraints will impact the design because the I must ensure the system’s coding and formats meet the capabilities of the infrastructure. This will limit the system in certain areas—although the capabilities will still far exceed those of the legacy maintenance.
* The software should adhere to locally or nationally recognized standards and should follow the National Information Exchange Naming and Design Rules.

## **Future Contingencies.**

The main future contingency that may arise is the interface agreements. Although this may not be a big issue, if it arises, then the I will have to gather information on the preferred interface by the users and make necessary changes to the system.

## **Document Organization**

While the software requirement specification (SRS) document is written for a more general audience, this document is intended for individuals directly involved in the development of OSCS application. This includes software developers, project consultants, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document.

* Part 1 (Introduction)
  + This section offers a summary of the OSCS application, including goals and objectives, project scope, general system details, and some major constraints associated with the intended platform.
* Part 2 (Architectural and Component-Level Design)
  + This section describes the OSCS application class by class, including interface details, class hierarchies, performance/design constraints, process details, and algorithmic models.
* Part 3 (Data Design)
  + Readers interested in how OSCS organizes and handles data should consult this section, which covers data structures and flow patterns utilized by the system.
* Part 4 (User Interface Design)
  + This section covers all of the details related to the structure of the graphical user interface (GUI), including some preliminary mockups of the OSCS application. Readers can view this section for a tentative glimpse of what the final product will look like.
* Part 5 (Restrictions, Limitations, and Constraints)
  + This section discusses the general constraints imposed upon the project
* Part 6 (Testing Issues)
  + Readers interested in the software testing process should consult this section, which offers a list of test cases, expected responses, and other pertinent information.
* Part 7 (Appendices)

**1.4 Points of Contact**

The points of contact in this project includes the System Super Administrator(SSA), who will be able to assign staff members to various departments where clearance should be done, should be able to view clearance requests sent by clearing students and should also be able to view the activity log of each user in the system, will also be able to generate clearance certificate for the cleared students. The Standard Administrators(SA) are as signed by the Super Administrator to work in various university departments, they will be receiving clearance requests from clearing students, they will also be able to check through the students’ records for any arrears, possessions or damages made before they clear a student from a department. Students will be able to log in to their accounts, make a clearance request, view their profiles, view requests and receive feedback from the system administrator.

# **Project References**

This document references the **Online Student Clearance System** proposal document and the **Online Student Clearance System** requirement specification document.

## **Glossary**

1. WAP - Web Application Framework
2. OSCS – Online Students’ Clearance System
3. WP - Web Framework.
4. JSP - Java Server Pages.
5. SSA – System Super Administrator.
6. SA – Standard Administrator.
7. HTTP –Hypertext Transfer protocol.
8. URL - **Uniform Resource Locator.**
9. WAMP - Windows, Apache, MySQL, and PHP.
10. MYSQL - structure query language ([SQL](https://techterms.com/definition/sql)), which is used for adding, removing, and modifying information in the database.

# **SYSTEM ARCHITECTURE**

OSCS is a client server application that is composed of different modules that will be accessed in the web server. The application is highly portable and can run of different operating systems.

## **2.1 System Hardware Architecture**

The main hardware components required for this application are:

1. A computer with the following requirements with at least 2 GB of RAM
2. A web/Application server with Intel 34500 Dual Core, 2Gig RAM, 120 GB SATA HD, Windows Server 2012. Database Server with intel Q6600 Quad Core, 2 Gig RAM, Dual 120 GB SATA RAID 1 and Windows Server 2012.

## **2.2 System Software Architecture**

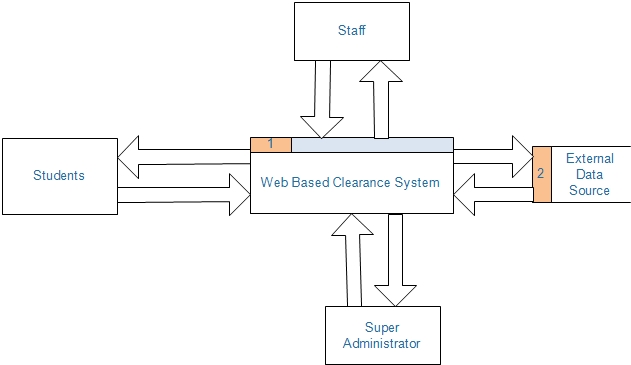
The Clearance Application is broken up into two major components: a client-side application and a server-side application and MYSQL database.

The client-side application is also separated into two parts: the functional component and the graphical component. The functional component forms the core of the Clearance System. It fetches data from server and posts data to server. The graphical component, as the name implies, is simply the graphical user interface. It provides all of the buttons, text boxes, and other onscreen elements which allow the user to access all of the features provided by the application.

The following software components will be used:

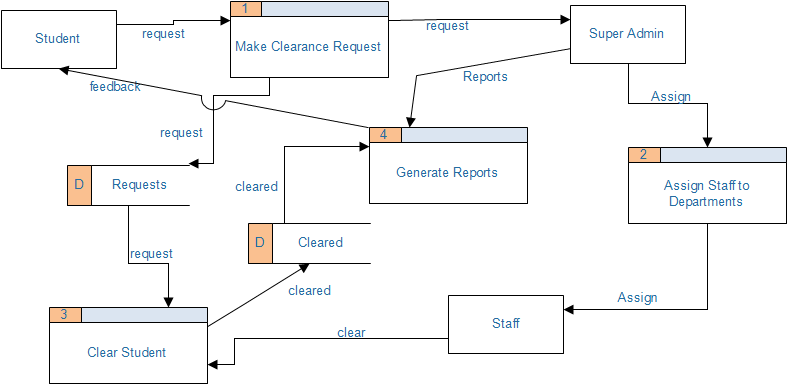
* Eclipse- This is the IDE in which the application will be built.
* Web browser- e.g. Mozilla Firefox, Chrome, Explore. It will be used by users to access the system from the server.
* WAMP server on which the database server and the application will be hosted.
* MYSQL DBMS which provides centralized storage for synchronized data.

Below is a context diagram that only shows the top-level communication between the system and external entities. It demonstrates how the Clearance System interacts with some of the external entities such us the students, administrators and eternal sources of data required for the operation of the system.



**Fig. 2:** *top level communication between the system and external entities.*

The Web Based Clearance System contains four processes, three external entities and two data stores as shown in the figure below.



**Fig. 3:** *a detailed communication between the system and external entities.*

## **2.3 Internal Communications Architecture**

The communication process on the application will be achieved by availability of internet connection to enable users to retrieve data from the database and also to enable the various modules in the system to communicate with one another.

# **FILE AND DATABASE DESIGN**

## **Database Management System Files**

The Web Based Student Clearance System is meant to provide complete interaction between Administrators and Student during clearance period. The system is meant to automatically generate reports and certificates for various students. The system administrator can generate and store this information and this will ensure efficiency and consistency in stored information about the graduating students.

MySQL is used as the primary storage for all the data that is in the application. The reason why the database is used because it has the following qualities:

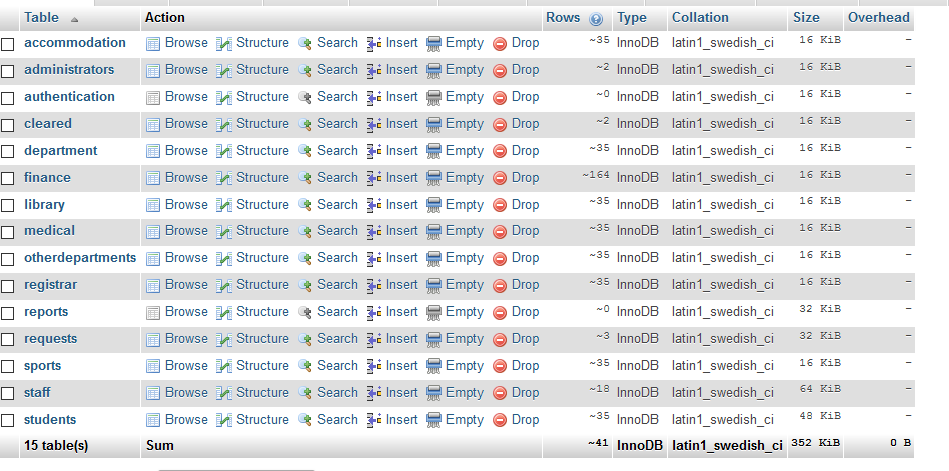
* It supports Structured Query Language(SQL)
* It is open and free.
* It supports referential integrity and transaction processing

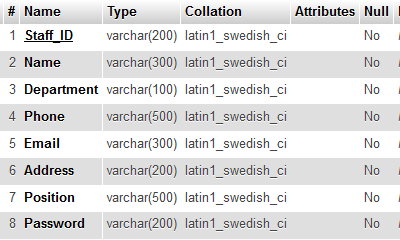
The database also comes with associated rich set of DBMS applications that facilitate query processing, analysis of data and making reports.

To facilitate the integrity, confidentiality and security of the sto red data, the database will only be accessible by the system administrator. There will be user interfacing activities on the various modules of the system to further facilitate data access and report generation.

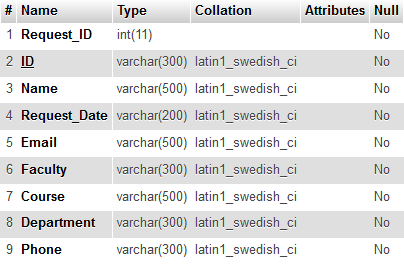
**Database entities**

The following entities will form the database schema.

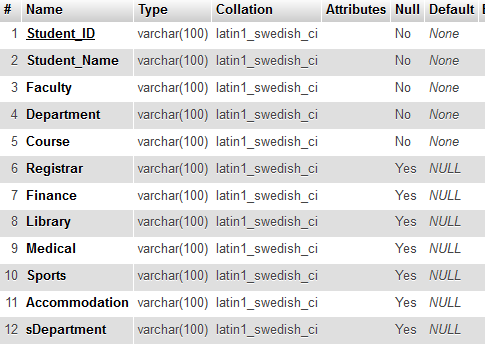


****

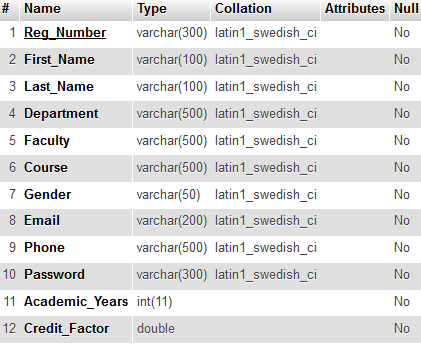
**Staff table**

****

**Requests table – stores requests from students.**

****

Cleared table – stores cleared students’ details



Students table

## **Non-Database Management Files**

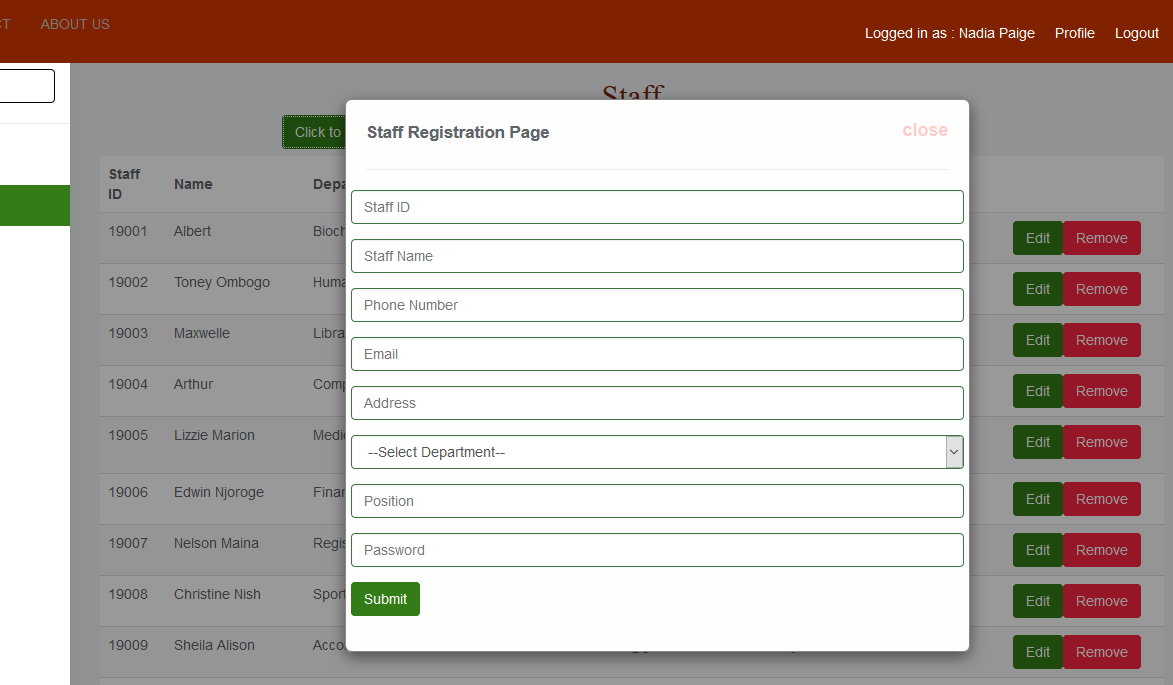
All data will be stored in relational database to facilitate the process of central access of data by all the system users. The management of data shall be done strictly using a database. Despite all this, reports generated from the database will be stored in the computer hard drive as excel worksheets. They shall be obtained from the computer hard drive.

# **HUMAN-MACHINE INTERFACE**

The system will consist of a registration module where all the users of the system will be registered. The users will be divided into three types. These are the system administrator, concerned staffs and the students. Users will use their login credentials to log into the system. After authentication, the users shall be able to perform their respective activities as stipulated in the Software Requirements Specification. Each user, upon successful login will be directed to their respective modules that will enable them to perform their respective activities.

## **Introduction**

The system has three sections with each user having its own section. The users are divided into three domains. These are the System Administrator, staff and students. Students are already in the system so they are not being registered in this system. The administrator section consists of a registration module where all the concerned staffs will be registered. The system administrator’s account is predefined in the system so he/she doesn’t need to be registered into the system as a user. The system administrator will open a web browser log into his/her account, access the registration module where he/she can register a new staff member into the system. The staff details will be recorded and stored in the respective relations in the database. After that, the staff will then log in to the system and access it using the details already stored in the database. All the system users are required to input their credentials in the input module to log into the system.

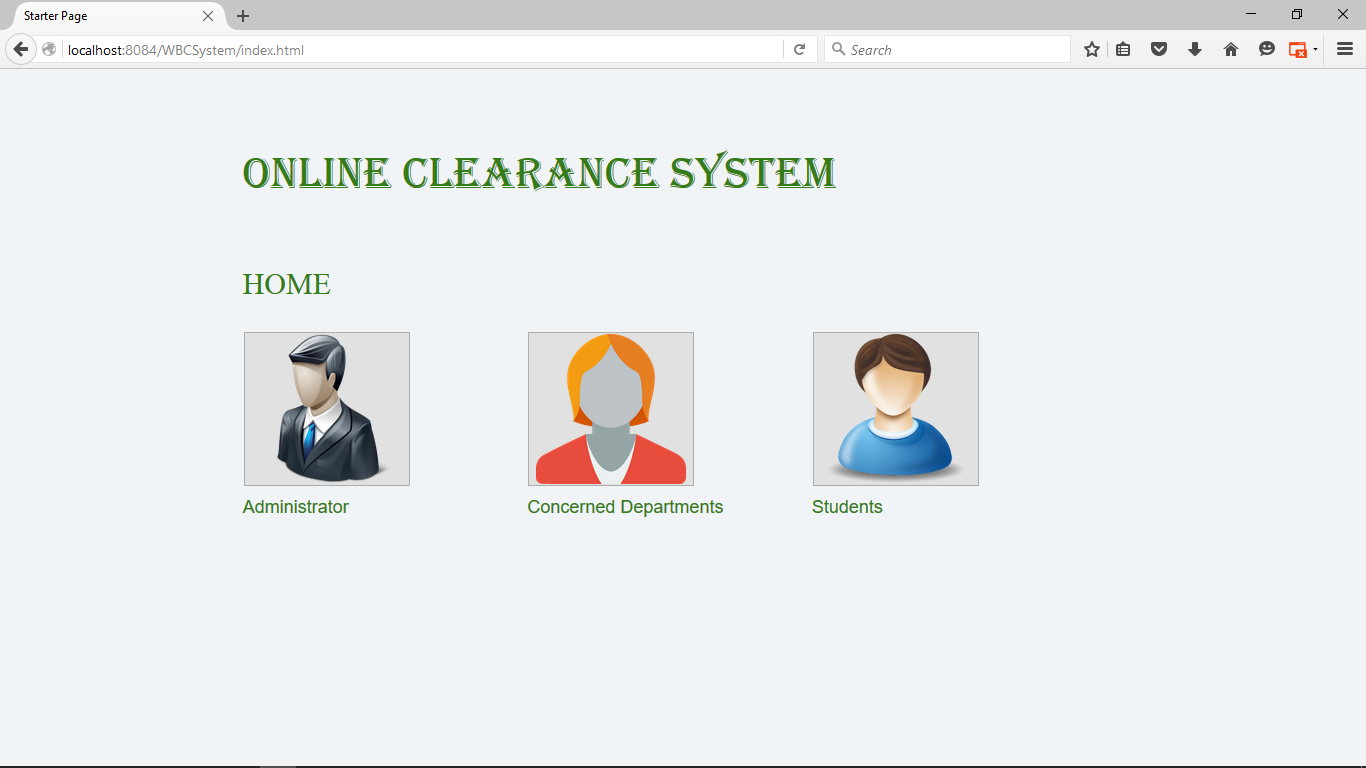


**Fig. 4:** *registration module for concerned staff members.*

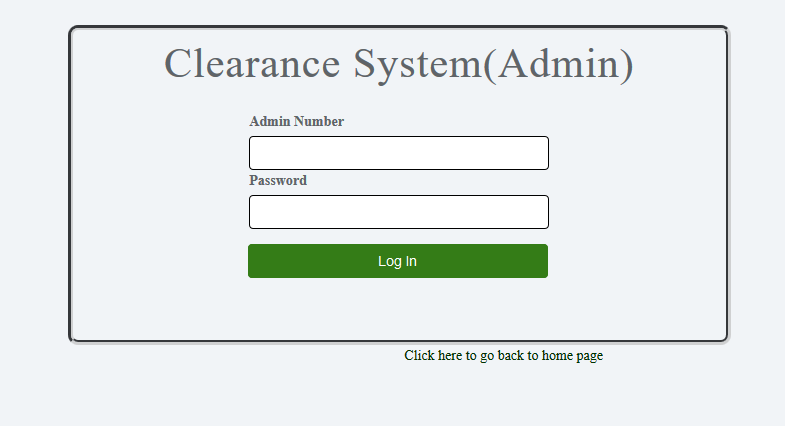
## **Inputs**

All data inputs will be in text formats. The input fields include text areas, text fields, drop downs and checkboxes as described in the input screens below. For each text input, the acceptable input will include the alphanumeric characters. The range of the inputs will be set as required. Failure to conform to the range of the inputs will be generate an error.

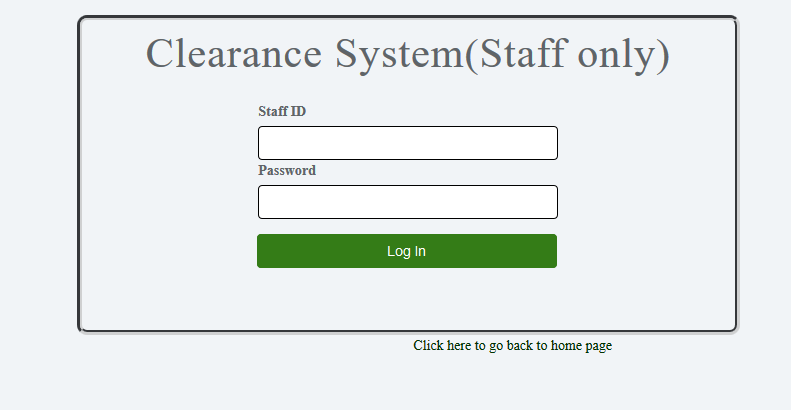
There will be a client side and server side validation of each input data to ensure that there are no null entries and that also the input types are valid. Security will also be checked on the client side by encrypting user passwords before they are stored in their respective relations in the database.

****

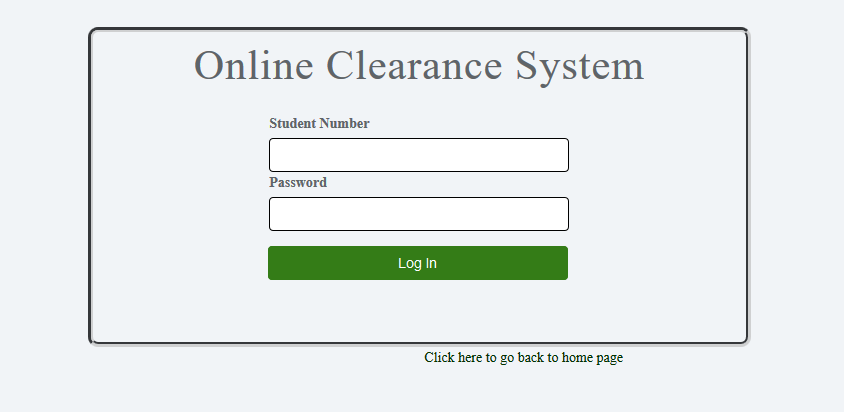
**Fig. 5** *Main page.*

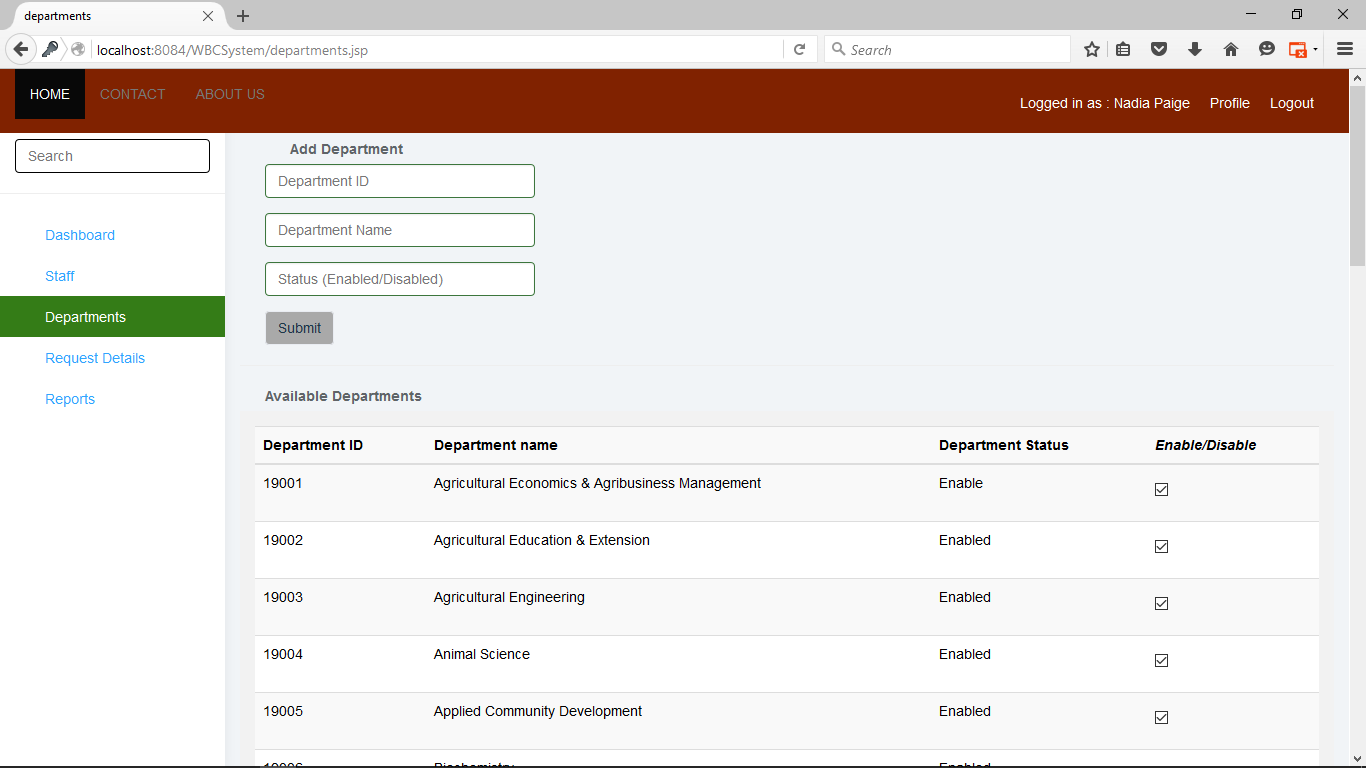


**Fig. 6** *Admin login page.*



**Fig. 7** *Staff login page*

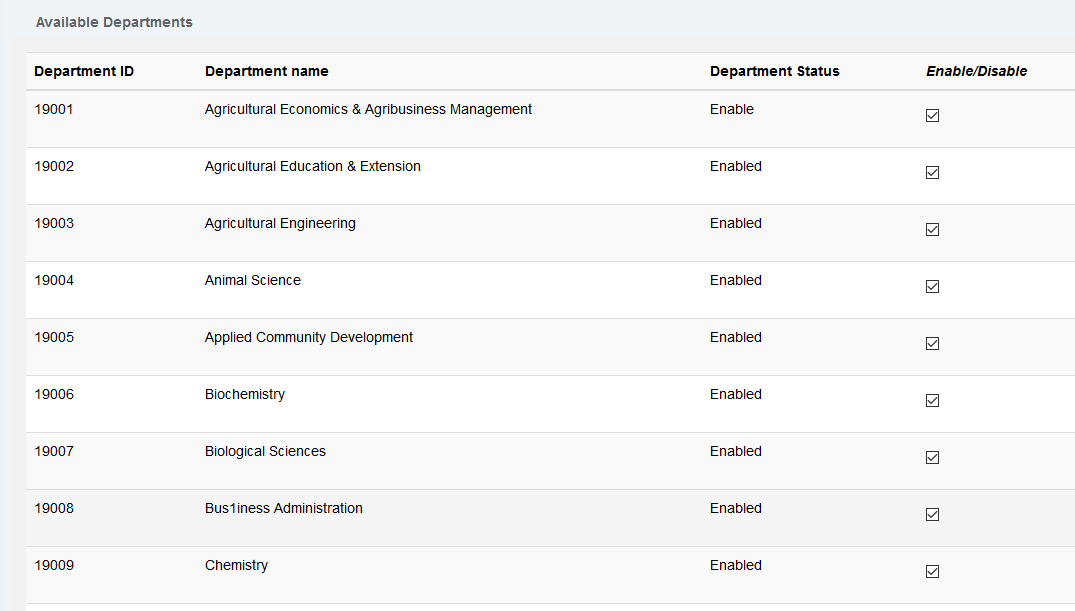
  
**Fig. 8** *Students’ login page*



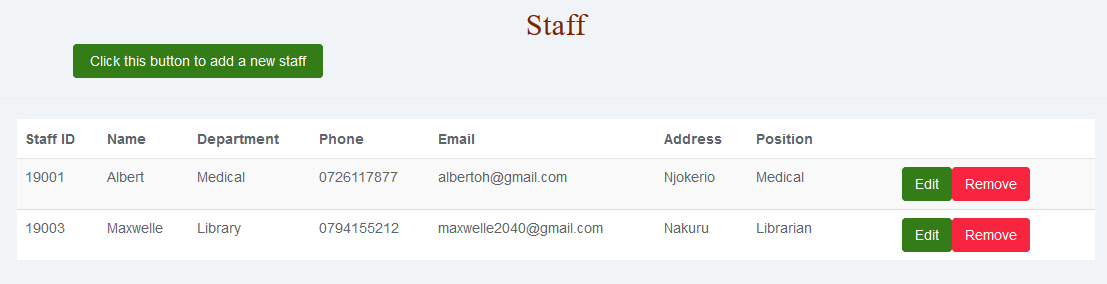
**Fig. 10** *department registration page*

## **Outputs**

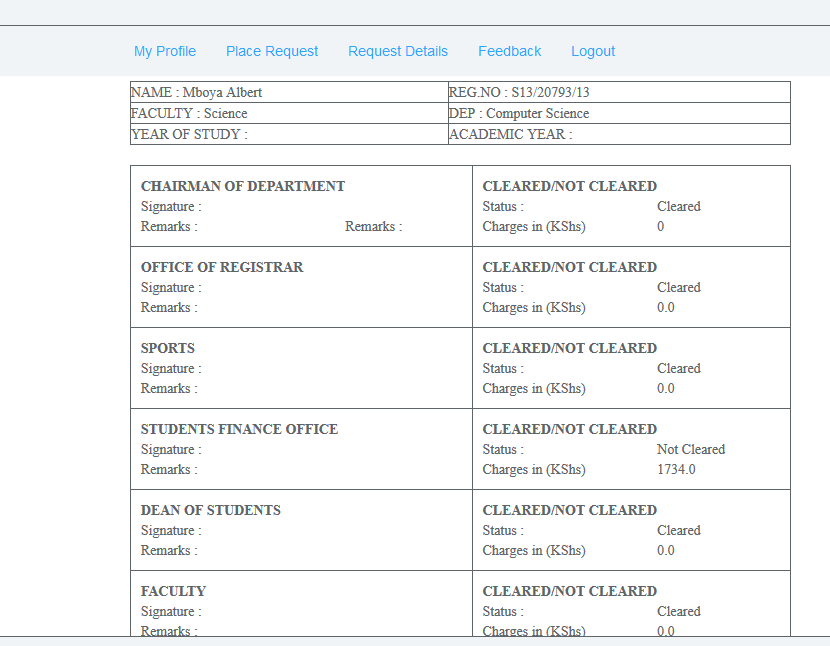
The system will output a range of data based on the user needs. Most data are outputted as either read-only or editable based on the user roles and needs.



**Fig. 11** *An output of the available departments*



**Fig. 11** *An output of the registered staffs*



**Fig. 11** *Student page showing clearance status and progress*

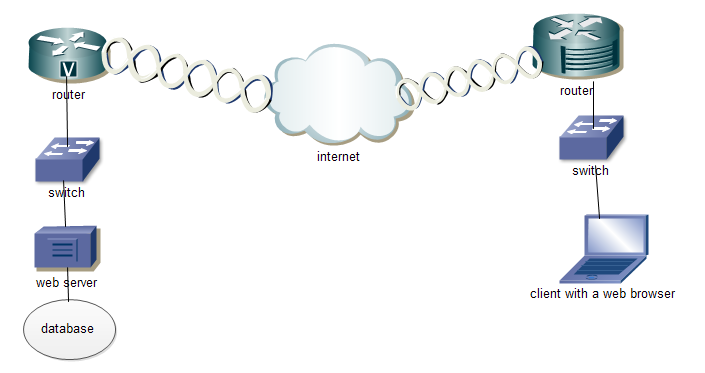
# **DETAILED DESIGN**

## **5.1 HARDWARE DETAILED DESIGN**

The system will run on a number of hardware components.

* Router
* Network switch
* Web server
* Client PC. With at least 1GB RAM and 1.5 GHZ processor speed.
* Client machine should have a browser. E.g. Mozilla Firefox

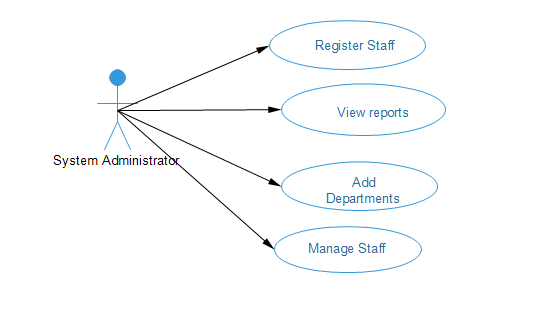
The physical component connection is depicted in the diagram shown below.

****

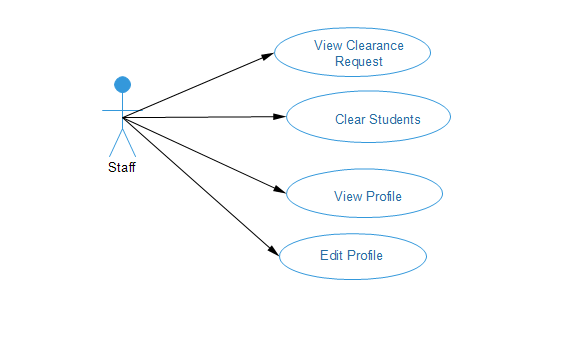
**Fig. 12** *Connection between the physical components of the system*

## **5.2 SOFTWARE DETAILDE DESIGN**

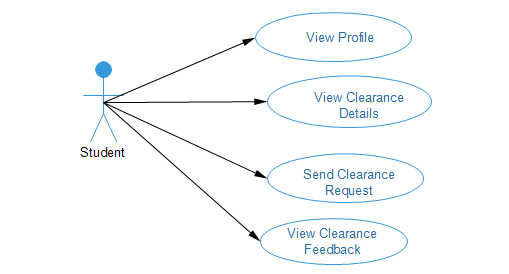
The system consists of the following modules. Each module is shown by a use case described below having the specific functionalities. The criterion of grouping these modules is based on the user roles. There are three actors in the system. They are the system administrator, Staff and students.



**Fig. 13** *System administrator module*



**Fig. 14** *Staff module*



**Fig. 15** *Student module*

## **5.3 INTERNAL COMMUNICATIONS DETAILED DESIGN**

The business flow of information from one module to another in the management system will make internal flow of information necessary. The internal flow of information will be in form of object communication among various objects of the classes in the system. The internal data communication will be handled by the HTTP protocol that will relay messages between various objects in the system.

Communication between the system and the database will be facilitated by various user interfaces describes earlier in the human- machine interface section which will ensure that data from the client is successfully stored in the database.

The interface will also ensure that data can easily be fetched from the database, updates and deletions can be done without affecting the consistency of data. External data communication between the system and the external environment will be realized by use of the internet connection between the system and the external environment through which data will be sent.

# **EXTERNAL INTERFACES**

Since communication is through TCP/IP, there is no need to reformat data before it is transmitted or received over the interface between two different communicating modules. Data exchange over the interface will be purely regulated by the protocol. Errors arising from such exchange will be handled dynamically by the protocol. Errors arising will be handled and a brief error description given to the user in HTML format.

# **SYSTEM INTEGRITY CONTROLS**

Based on the sensitivity and importance of the data that is passed between the various modules of the system, there is need to put in place integrity checks to ensure correctness of the stored data and accountability on the part of every system user. The following security control measures will be implemented to enforce integrity controls

1. Authorization control. This ensures that users can only access data that belongs to their user groups based on user roles.
2. Access to the data in the database will be restricted to various user groups based on the user roles. Users can only update data that they have supplied. Update of data affecting the whole system will only be restricted to the system administrator. Data deletions will also be limited to certain classes of users based on the user roles.
3. Integrity control. This ensures that data stored in the database cannot be violated or altered by any unauthorized parties in any way whatsoever.
4. Authenticity control. This aspect of security will verify user credentials at the time of login into the system. This ensures that only authorized users are allowed to access the system.
5. Confidentiality control. This is a security technique that prohibits unauthorized access of personal information.
6. There shall be a system audit mechanism that automatically collects data based on user transactions to monitor user activities in the system. The audit will be available to the system administrator in form of log information detailing user activities, user identifications and time so as to enhance accountability.

Server side and client side validation of data to ensure no wrong inputs are accepted into the system.

Due to the various types of attacks that are in existence today, the system has taken into considerations just but a few of these attacks. For example, SQL injection and session hijacking attacks have been greatly controlled in this system.