

EGERTON UNIVERSITY

software design document (SDD)

for

E-NOTICEBOARD SYSTEM

prepared by: PETER MAKORI

reg. no.: S13/09722/15

project supervisor: mR. kATIKU JONES

project cordinator: DR. ING. WILFRED GIKARU

7TH May, 2019

version: 1.0

**Overview**

This System Design Document will seek to describe the system requirements of the E-Noticeboard, the operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

Table of Contents

[1 INTRODUCTION 1](#_Toc9312443)

[1.1 Purpose and Scope 1](#_Toc9312444)

[1.2 Project Executive Summary 1](#_Toc9312445)

[1.2.1 System Overview 2](#_Toc9312446)

[1.2.2 Design Constraints 2](#_Toc9312447)

[1.2.3 Future Contingencies 3](#_Toc9312448)

[1.3 Document Organization 3](#_Toc9312449)

[1.4 Points of Contact 3](#_Toc9312450)

[1.5 Project References 4](#_Toc9312451)

[1.6 Glossary 4](#_Toc9312452)

[2 SYSTEM ARCHITECTURE 5](#_Toc9312453)

[2.1 System Hardware Architecture 5](#_Toc9312454)

[2.2 System Software Architecture 6](#_Toc9312455)

[2.3 Internal Communications Architecture 8](#_Toc9312456)

[3 FILE AND DATABASE DESIGN 9](#_Toc9312457)

[3.1 Database Management System Files 9](#_Toc9312458)

[3.1.1 Entity-Relationship (E-R) Diagram 9](#_Toc9312459)

[3.2 Non-Database Management System Files 9](#_Toc9312460)

[3.2.1 Media Files 9](#_Toc9312461)

[3.2.2 Static Files 10](#_Toc9312462)

[4 HUMAN-MACHINE INTERFACE 11](#_Toc9312463)

[4.1 Inputs 11](#_Toc9312464)

[4.2 Outputs 15](#_Toc9312465)

[5 Detailed Design 16](#_Toc9312466)

[5.1 Hardware Detailed Design 16](#_Toc9312467)

[5.2 Software Detailed Design 16](#_Toc9312468)

[5.2.1 System Flowchart 16](#_Toc9312469)

[5.2.2 Class Diagrams 16](#_Toc9312470)

[5.2.3 Use Cases 17](#_Toc9312471)

[5.3 Internal Communications Detailed Design 18](#_Toc9312472)

[6 External Interfaces 20](#_Toc9312473)

[7 SYSTEM INTEGRITY CONTROLS 21](#_Toc9312474)

# INTRODUCTION

## Purpose and Scope

The purpose of this document is to describe and depict the software design of the E-Noticeboard system. The subsections that will follow the Software Design Document (SDD) will seek to give the design of the system. This document will also define a detailed system architecture and functionality design of the E-notice board system to pave way for the development process. The document will describe the operating environment, system and architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and the external interfaces of the E-Noticeboard.

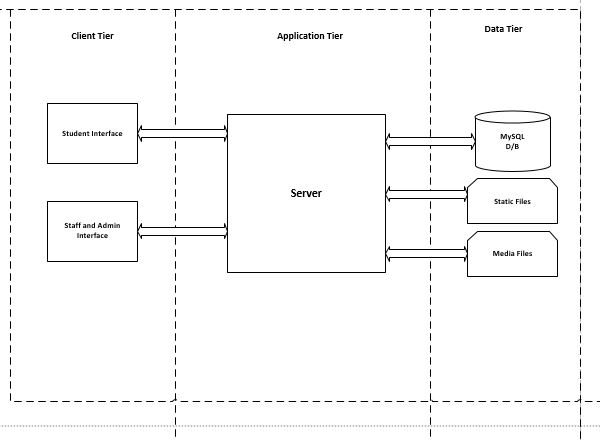
The E-Notice Board system will have four primary users who will be the system administrator, the dean, Chairperson of Department and the student. The administrator will provide general management and keep track of the activities performed in the system while the dean and the COD will create, update and delete notices. The student will view the notices with respect to the faculty and department they belong and also send some feedback to the departments and faculty when need arises.

## Project Executive Summary

This project is aimed at developing a campus E-Notice Board system. The system will have three four primary users who will be the administrator, the dean, chairperson of the department (COD) and the student. The admin will be expected to keep track of all the processes that are being carried out by the system and provide general monitoring. The admin will also register or create accounts for the dean and COD. On the other hand, the dean and COD will create notices, update and delete them at the faculty and departmental level respectively. Notices coming from the faculty level will be managed by the dean while the COD will create, update and delete notices coming from the departmental level. The students will be expected to view the notices targeted to them by the department or faculty to which they belong. They will also be giving a feedback in terms of a query, comment or suggestion in case any need arise which will help the university improve their services. While the student will be the only one expected to register before accessing any service, all the users will be required to login before they perform any role within the system.

### System Overview

This software will focus on eliminating the rigorous process of students has to go to the specific locations of notices to view them. The system will allow the university to post notices on an online platform and the students will receive an alert. They will then login and view the notice with respect to their faculty and department. The system will hence prove significant in ensuring that students can get information timely and regardless of the relative place to the university at the time of posting.



### Design Constraints

The major constraint for the E-Notice board system is time. The time allocated for project development is close to one and half months. This also includes the configuration and setting up of the development environment as well as databases configurations. The testing, and documentation of the project is also expected to be complete in a span of one month. Jointly, the system developer has minimal experience with the Django framework and therefore significant timeframe will be dedicated towards learning the framework. Thus, time will be a major constraint. This may lead to less features in the initial design and release despite that the core functionalities of the system will not be affected.

### Future Contingencies

Due to the evaluation and personal level perspectives, the system may be subject to change the user interfaces design. Moreover, the system may be subject to change the flow of activities because the functionalities of the system are limited and expected to comply with the flow of information in the University. For this reason, the system orientation and operations of the major activities may change incase need arises.

## Document Organization

This SDD document is in Times New Roman theme fonts. The headings are numbered in numeric numbers of 1,2,3,4…while the subheadings are numbered as x.1, x.2, x.x.1… and so on, depending on the level of the subheading. All the headings and subheadings are in bold and in 14 font size. The rest of the content is in normal style and 12 font size. The entire document is exactly spaced. The document is organized into topics and each topic address specific aspect according to the conventions of a Software Design Document. The document will consist of the glossary, design considerations, data design with class and table structures, system architecture with components and user interface designs.

## Points of Contact

|  |  |
| --- | --- |
| Person | Role |
| System Manager | The person has a full understanding of the project and manages the whole project and all the system developers. The person facilitates communication between the developers and the project sponsor. |
| Quality Assurance | Entitled to ensuring that the system functionalities work perfectly as documented and there are no any flaws prior to system deployment. The person also ensures that the software system meets the ISO standards. |
| Security Manager | Responsible for system security and protecting the data of the system against security breaches and malicious intruders. |
| User Organization. | Consists of all the system users. |
| Configuration Manager | Ensures that all the system components are integrated together and communicate to work efficiently as a wholesome system. |

## Project References

This section provides a bibliography of key project references and deliverables that have been produced before this point.

## Glossary

COD- Chairperson of Department

RegNo-Registration Number

HTTP- Hyper Text Transfer Protocol

CRUD- Create Read Update Delete

E-R- Entity Relationship

Django- Python framework for web applications.

# SYSTEM ARCHITECTURE

This section will provide the system architecture. The tools that will be used for the development of the system will also be described in this section.

## System Hardware Architecture

The system will be built using Django and MySQL running on localhost but will later be transferred to a server during deployment. Using a basic three-tier-architecture, the system will be split into three levels; the data tier, application tier and client tier. Users or machines at the client tier will request web pages by sending requests to the web server which, at the application tier, will find

the correct resource and run the script. The server retrieves any data it needs from the data tier by

querying MySQL and formats the results into a dynamic page for the user to view and interact with.

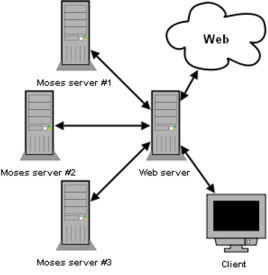
The results of the user’s interaction are then dealt with by the Django code and any affected database elements are updated. Three tier applications deliver higher performance than the standard client-server

architecture. The diagram below illustrates how this architecture allows each tier to focus solely on its assigned task and the critical middle tier can be inexpensively scaled to virtually any level by adding

inexpensive industry-standard computing resources, meaning the system could be easily scaled up to a

regional level as requested. This architecture also allows multiple users to run the administration

interface from anywhere on the network and change the content being displayed on the front office system.



## System Software Architecture

1. **Python Programming Language**

Python is a powerful, high-level, interpreted, multi-purpose programming language that is simple with easy-to-use syntax. It is dynamically typed and supports multiple programming paradigms such as procedural, object-oriented and functional programming. Its interpreter can be available in most operating systems. Python programming language will be used in the backend development of the E-Notice Board system. However, it should be noted that python alone supports desktop applications and does not support web applications unless used in conjunction with a framework such as Django or Flask frameworks.

1. **Django Framework**

Django is a high-level Python Web framework that encourages rapid development design of systems and follows the model-view-template architectural pattern. It is free and open source. Its primary goal is to ease the creation of complex, database-driven websites by encouraging code reuse and ‘plug ability’ of components. For this reason, the Django framework will be used in the development of this system in the backend.

1. **Atom**

Atom is a highly complex text editor that provide significant features such as code completion and indentation, inspection and advanced debugging that gives a good and admirable view. The text editor is used in computer programming irrespective of any language. It is cross-platform with Windows, mac, and Linux versions and therefore will not limit its usage to any specific Operating System environment. Thus, the atom text editor will be used in the development of this system.

1. **MySQL**

MySQL is a free and an open-source relational database management system. It is reliable, ease-to-use, and has become the leading database choice for web-based applications. MySQL will be used in the storing of data of the system and querying for the same data. However, this database will be adopted during the production stage since Django has an inbuilt dB SQLite which gives the functionalities almost as those provided by MySQL. Therefore, dB SQLite will be used during the early stages but later on the database will be migrated to MySQL.

1. **Microsoft Visio 2016**

The Unified Modelling Language (UML) is a general-purpose, standardized language for constructing, visualizing and documenting diagrams for system software. The language can be used in business modelling and architectural designs for non-software systems. On this note, UML will be used in the design and modeling of the E-Noticeboard system. The Microsoft Visio 2016 is a software application that comes packaged with all the components and capability that allow the modelling of diagrams and gives a quite good taste for its outstanding interface. Thus, this application will be used to draw diagrams which include use cases, sequence diagrams, class diagrams, activity diagrams, E-R diagrams and data flow diagrams, among others.

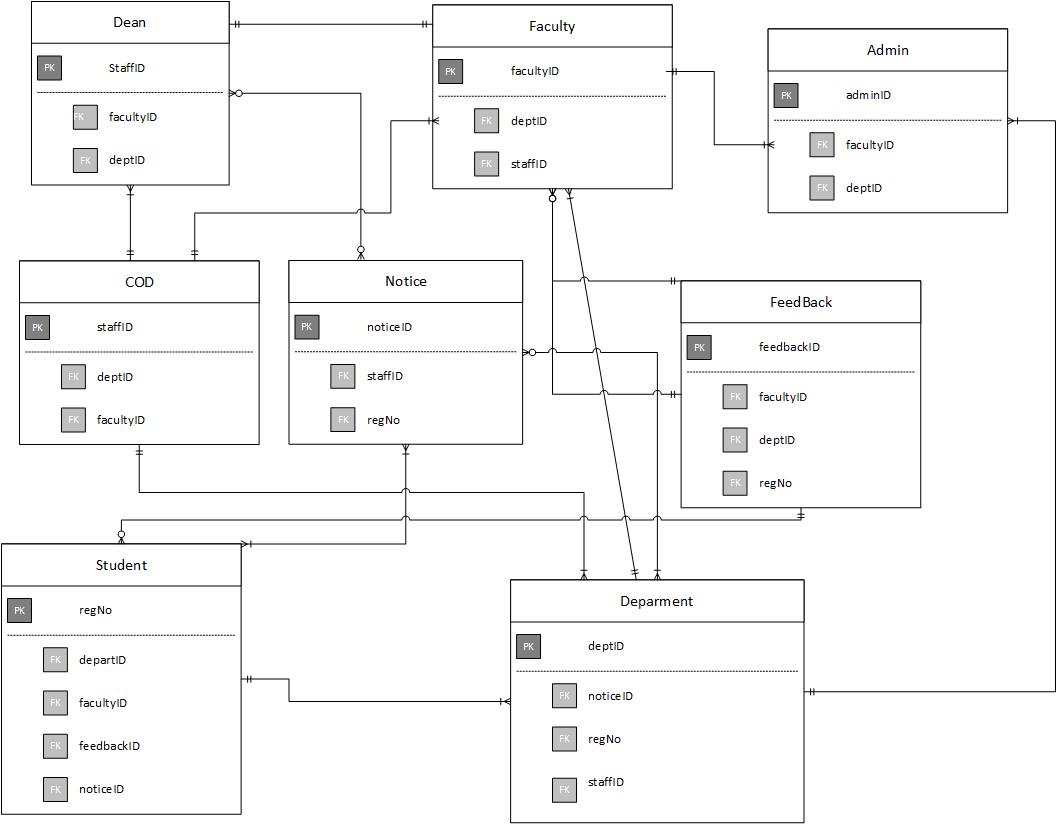
## Internal Communications Architecture

There will be no any special internal communications architecture that will be implemented in this system because the system will be communicating using the HTTP port 80 in data transfer.

# FILE AND DATABASE DESIGN

## Database Management System Files

### Entity-Relationship (E-R) Diagram



## Non-Database Management System Files

### Media Files

This will consist of the files such as images which will not be stored in the database, but instead will be stored at the root directory of the application system. They may however be referenced by their name in the database.

### Static Files

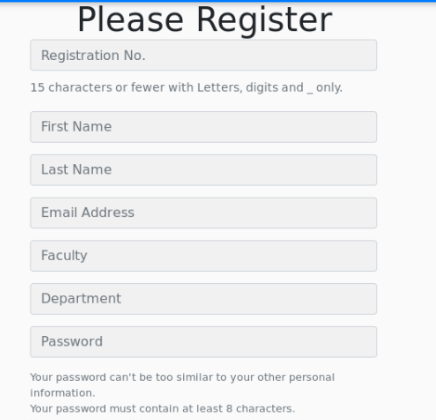
These files include CSS and JavaScript. They are stored in the static directory of the root directory of the application.

# HUMAN-MACHINE INTERFACE

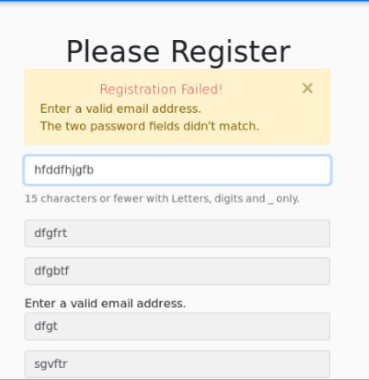
## Inputs

The data will be input to the E-Notice Board system mainly by the use of forms which will be comprised of input fields, password fields and selection list. The user will enter data which will then be validated by the system before being stored in the database. The stored data can be retrieved updated or deleted depending on the user needs. The occurrence of errors during input of data, will display meaningful error messages to the users with regard to what the errors are and what valid input is expected. Upon entering the valid data to the required fields, the user will be redirected to the next page and a success message will be displayed on the screen.

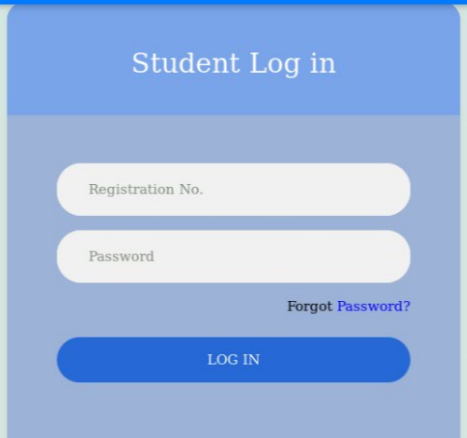
A typical Registration form for the students



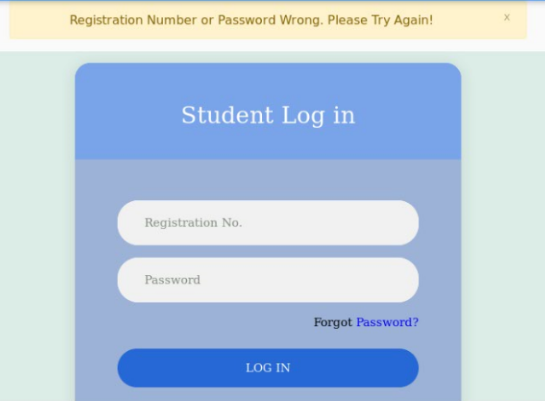
The Registration form when user inputs invalid data



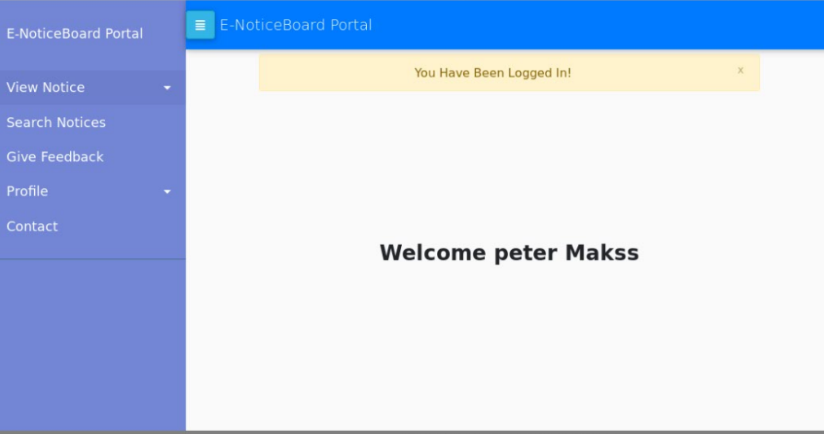
Login Page



Login Form with Invalid input



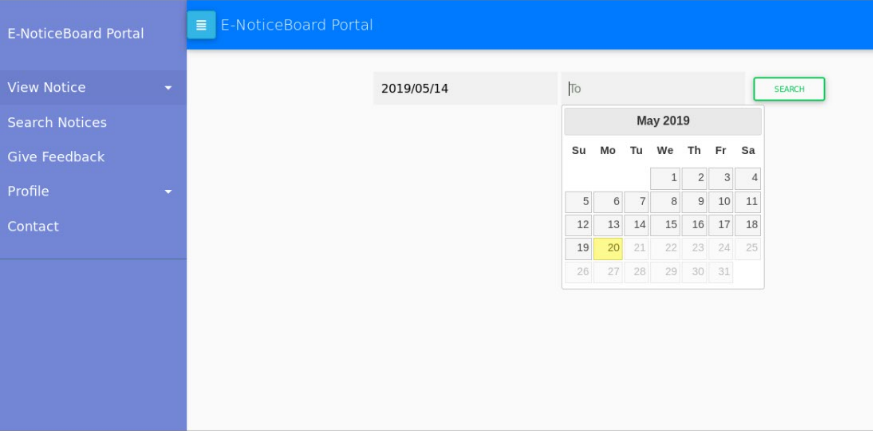
Homepage Displaying success logging in success message to the user



## Outputs

The data will be output and displayed in various ways depending on the type of data that the user requests. However, the standard formats will be kept and most importantly is that the data will always be displayed in the most effective way that will give an easier understanding to the user. This will enable the users to easily visualize and understand whatever output is being displayed. Different colors will be used to get the users’ attention wile at the same time ensuring that there is no use of many different colors that will overwhelm the users.

Sample output for date format when user searches for notice between a specific period



# Detailed Design

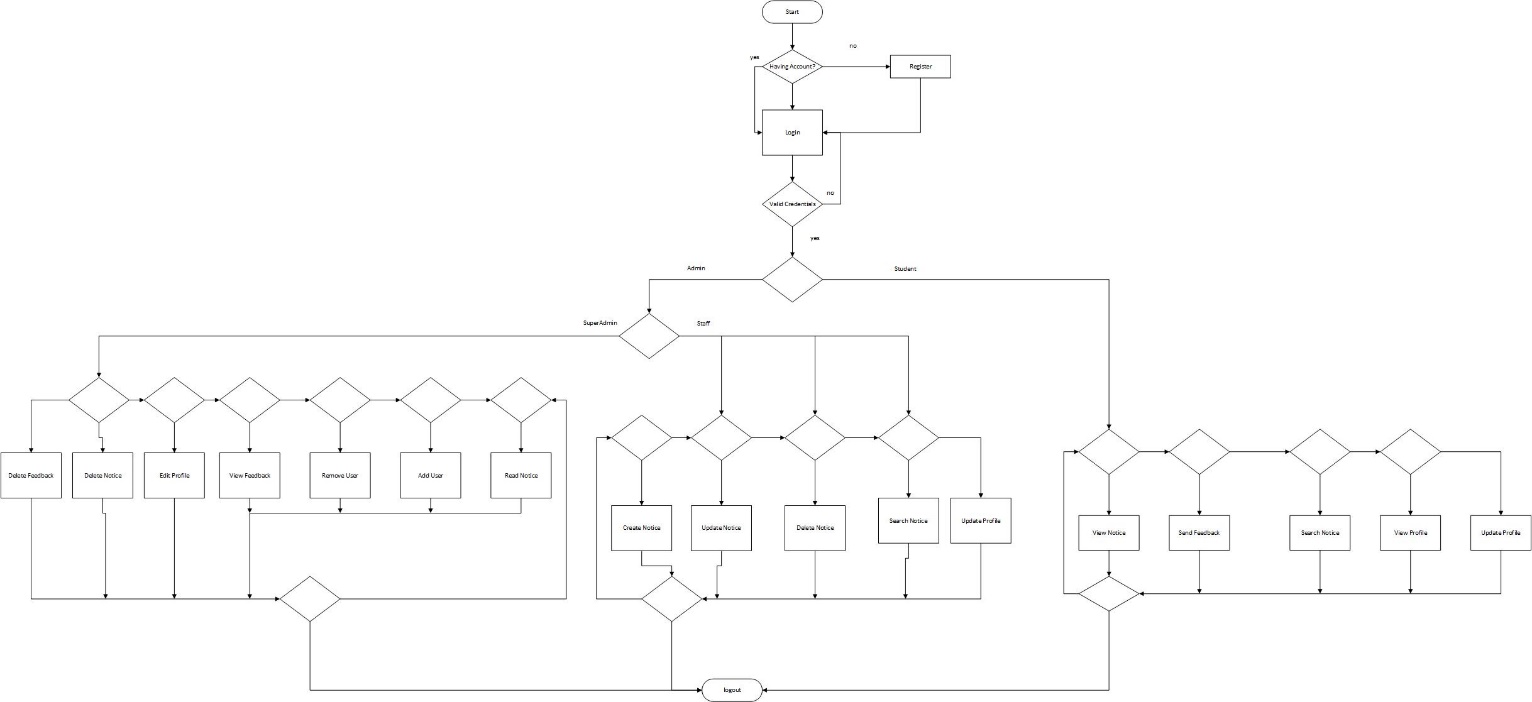
## Hardware Detailed Design

The is no much considerable hardware design because the E-Notice Board system will be a web-based system whose much processing will be based on the software part. The only hardware parts of the system will be a PC which will run on a web browser. All the other hardware components will be for deployment server.

## Software Detailed Design

### System Flowchart

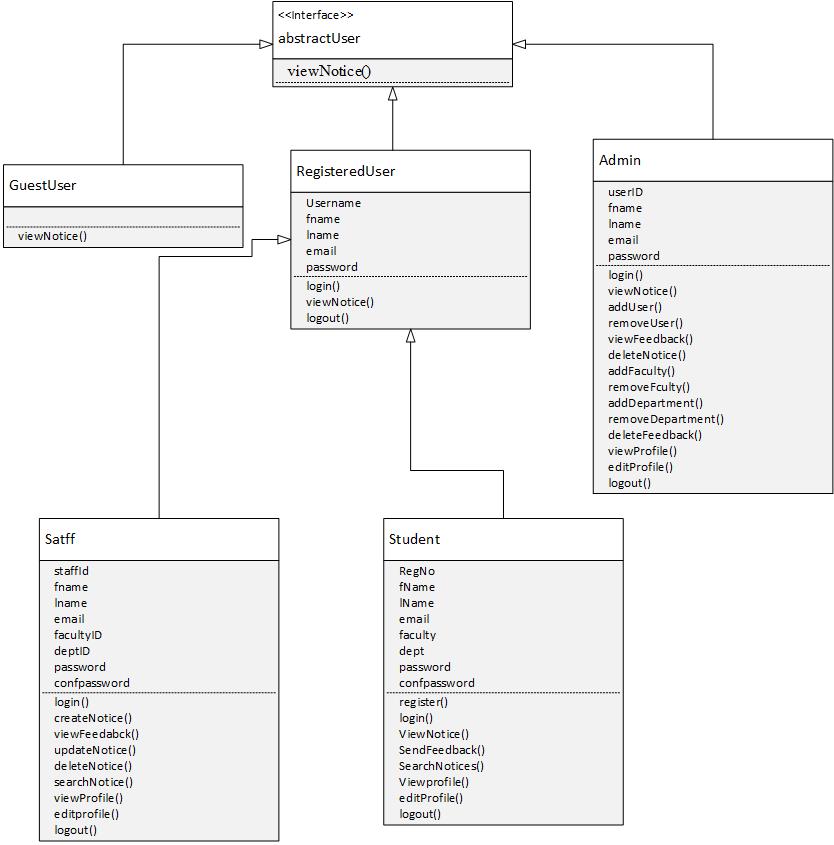
The figure below shows a basic flowchart for the E-Notice Board system. The flow chart shows the flow of activities from start to the end for every user of the system.



### Class Diagrams

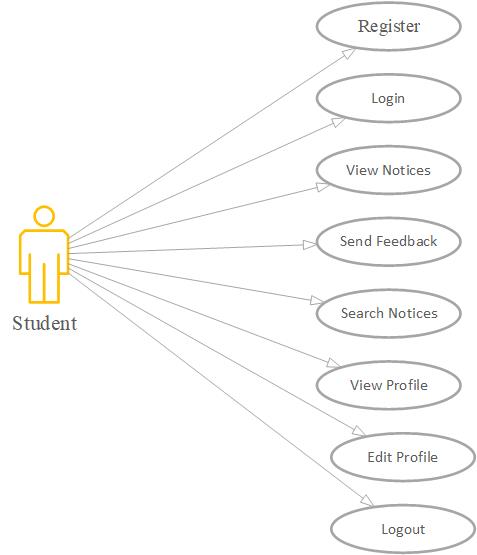
The class diagrams show how the functionalities of the system will be implemented.

The following is a class diagram showing the different users of the system with the methods that each user will use to manipulate data.

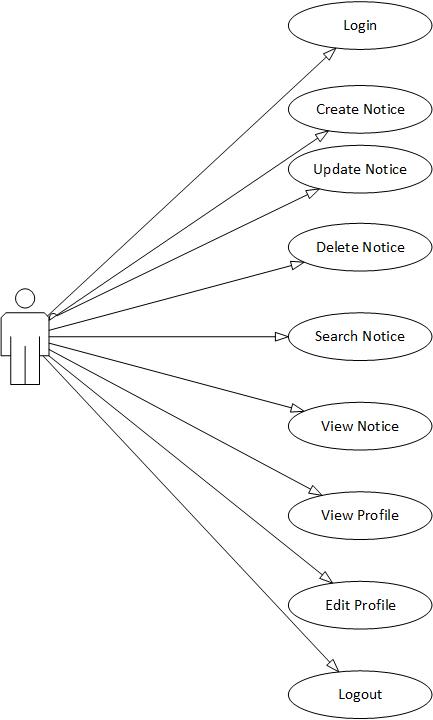


### Use Cases

#### Student Use Case Diagram



#### Staff Use Case



## Internal Communications Detailed Design

The E-Notice Board system will be using the client-server architectures. The users will be the clients who will be expected to send requests to the database through the web browser. These requests will then be channeled to the server which will process the requests. Processing of data either from the client to the database or from the database to the client will be done in the business layer which will be the business. The client PCs will communicate to the server using HTTP protocol port 80. Communication between the server and the database will take place via port 3306.

# External Interfaces

The E-Notice Board system will not connect to any external system for data exchange. For this reason, no external interfaces will be included in this section.

# SYSTEM INTEGRITY CONTROLS

The E-Notice Board system will bare sensitive information whose loss, misuse, modification, or unauthorized access will render the university in critical condition and may possibly affect the running of the university programs and affairs. The privacy will also be likely to be breached which is entitled to some notices and posts. For this reason, the following system integrity controls will be implemented:

1. **Password Encryption**

The passwords of the users will be hashed and encrypted before they are stored in the data store to ensure that anyone who gains access to the database does not manipulate the data of users by the sue of their passwords.

1. **Data validation**

Data will be validated at the client side or web browser before it is sent to the database for storage. Meaningful error messages will be given to the users when they enter invalid data like common or short password, invalid email access, usernames that are already taken or null fields.

1. **Integrity control**

Unauthorized users of guest users will not be allowed to access information belonging to other system users.

1. **Authorization Control**

Only webpages that are permissioned to each given user will be accessed by that specific user. Every user will be authenticated during login to ensure that they access the information or links they are authorized to access

1. **Database Restriction**

The users of the E-Notice Board will not be permitted to manipulate the data in the database. They will only be able to edit limited personal details such as their email address. The will however be able to view other information pertaining to their profile but will not edit most of it.