**­­­­­­**

**Egerton University**

**Software Requirements Specification**

**For**

**E-Notice Board**

**Prepared by** Peter Mose Makori

**Supervisor** Mr. Jones Katiku

**Coordinator** Dr. Ing. Wilfred Gikaru

**March 31, 2019**

**Version 1.0**

# **Table of Contents**

[**Revision History** iv](#_Toc7174161)

[**1** **Introduction** 1](#_Toc7174162)

[**1.1** **Purpose** 1](#_Toc7174163)

[**1.2** **Document Conventions** 1](#_Toc7174164)

[**1.3** **Intended Audience and Reading Suggestions** 1](#_Toc7174165)

[**1.4** **Project Scope** 1](#_Toc7174166)

[**2** **Overall Description** 3](#_Toc7174167)

[**2.1** **Product Perspective** 3](#_Toc7174168)

[**2.2** **Product Features** 3](#_Toc7174169)

[**2.3** **User Problem Statement** 4](#_Toc7174170)

[**2.4** **User Objectives** 4](#_Toc7174171)

[**2.5** **Operating Environment** 5](#_Toc7174172)

[**2.5.1** **Software Requirement** 5](#_Toc7174173)

[**2.5.2** **Hardware Requirement (Minimum)** 5](#_Toc7174174)

[**2.6** **Design and Implementation Constraints** 5](#_Toc7174175)

[**2.7** **User Documentation** 6](#_Toc7174176)

[**2.8** **Assumptions and Dependencies** 6](#_Toc7174177)

[**2.9** **Constraints** 6](#_Toc7174178)

[**3** **System Features** 8](#_Toc7174179)

[**3.1** **User Registration** 8](#_Toc7174180)

[**3.1.1** **Description and Priority** 8](#_Toc7174181)

[**3.1.2** **Stimulus/Response sequence** 8](#_Toc7174182)

[**3.1.3** **Functional Requirements** 8](#_Toc7174183)

[**3.1.4** **Technical Issues** 9](#_Toc7174184)

[**3.1.5** **Dependencies with other Requirements** 9](#_Toc7174185)

[**3.2** **User Login/Authentication** 9](#_Toc7174186)

[**3.2.1** **Description and Priority** 9](#_Toc7174187)

[**3.2.2** **Stimulus/Response** 9](#_Toc7174188)

[**3.2.3** **Functional Requirements** 9](#_Toc7174189)

[**3.3** **Create Notice** 10](#_Toc7174190)

[**3.3.1** **Description and Priority** 10](#_Toc7174191)

[**3.3.2** **Stimulus/Response** 10](#_Toc7174192)

[**3.3.3** **Functional Requirements** 10](#_Toc7174193)

[**3.4** **Delete Notice** 10](#_Toc7174194)

[**3.4.1** **Description and Priority** 10](#_Toc7174195)

[**3.4.2** **Stimulus/Response** 11](#_Toc7174196)

[**3.4.3** **Functional Requirements.** 11](#_Toc7174197)

[**3.5** **View Notice** 11](#_Toc7174198)

[**3.5.1** **Description and Priority** 11](#_Toc7174199)

[**3.5.2** **Stimulus/Response** 11](#_Toc7174200)

[**3.5.3** **Functional Requirements** 11](#_Toc7174201)

[**3.6** **View profile** 12](#_Toc7174202)

[**3.6.1** **Description and Priority** 12](#_Toc7174203)

[**3.6.2** **Stimulus/Response** 12](#_Toc7174204)

[**3.6.3** **Functional Requirements** 12](#_Toc7174205)

[**3.7** **Update Profile** 12](#_Toc7174206)

[**3.7.1** **Description and Priority** 12](#_Toc7174207)

[**3.7.2** **Stimulus/Response** 12](#_Toc7174208)

[**3.7.3** **Functional Requirements** 12](#_Toc7174209)

[**3.7.4** **Change Password** 13](#_Toc7174210)

[**3.7.5** **Description and Priority** 13](#_Toc7174211)

[**3.7.6** **Stimulus/Response** 13](#_Toc7174212)

[**3.7.7** **Functional Requirements** 13](#_Toc7174213)

[**3.7.8** **Technical Issues** 13](#_Toc7174214)

[**3.7.9** **Dependencies with other Requirements** 13](#_Toc7174215)

[**4** **External Interface Requirements** 14](#_Toc7174216)

[**4.1** **User Interfaces** 14](#_Toc7174217)

[**4.2** **Hardware Interfaces** 14](#_Toc7174218)

[**4.3** **Software Interfaces** 15](#_Toc7174219)

[**4.4** **Communication Interfaces** 15](#_Toc7174220)

[**5** **Other Non-Functional Requirements** 16](#_Toc7174221)

[**5.1** **Performance Requirements** 16](#_Toc7174222)

[**5.2** **Safety Requirements** 16](#_Toc7174223)

[**5.3** **Security Requirements** 17](#_Toc7174224)

[**5.4** **Software Quality Attributes** 17](#_Toc7174225)

[**6** **Preliminary Object-Oriented Domain Analysis** 18](#_Toc7174226)

[**6.1** **Inheritance Relationships** 18](#_Toc7174227)

[**6.2** **User Classes and Characteristics** 18](#_Toc7174228)

[**6.2.1** **Abstract or Concrete** 18](#_Toc7174229)

[**6.2.2** **List of Super Classes** 18](#_Toc7174230)

[**6.2.3** **List of Sub Classes** 19](#_Toc7174231)

[**6.2.4** **Purpose** 19](#_Toc7174232)

[**6.2.5** **Collaborations** 19](#_Toc7174233)

[**6.2.6** **Attributes** 19](#_Toc7174234)

[**6.2.7** **Operations** 19](#_Toc7174235)

[**6.2.8** **Constraints** 19](#_Toc7174236)

[**7** **Preliminary Budget and Schedule** 20](#_Toc7174237)

[**7.1** **Budget** 20](#_Toc7174238)

[**7.2** **Schedule** 20](#_Toc7174239)

[**8** **Other Requirements** 21](#_Toc7174240)

[**8.1** **References** 21](#_Toc7174241)

[**8.2** **Appendices** 21](#_Toc7174242)

[**Appendix A: Glossary of Definitions, Acronyms and abbreviations** 21](#_Toc7174243)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
| E-Notice Board |  |  | 1.0 |
|  |  |  |  |

# **Introduction**

## **Purpose**

The purpose of this document is to evaluate and depict the software requirements of the E-notice board system. The subsections that will follow the Software Requirements Specification (SRS) document will seek to give a framework of the whole SRS. This document will also define the scope and functionality of the E-notice board system.

## **Document Conventions**

This SRS 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 double spaced.

## **Intended Audience and Reading Suggestions**

This document is intended to be read by students, lecturers, and developers. The rest of this document consists of the software requirements; both functional and non-functional requirements for the E-Notice Board System.

## **Project Scope**

The E-Notice Board system will be dedicated to helping the affected parties of manual noticeboard to handle the core involved activities in a more reliable and techno approach.

The system will chiefly be beneficial users because, firstly, it will be exclusively aimed to disseminate information to students in an informative way without involving the paperwork. It will also be beneficial since access to information will be guaranteed and most interesting are that it will be advantageous in ensuring scalability is improved since a number of functionalities will be provided in the system.

The users will spend less effort in accessing and passing within the intended players. There will be easy tracking of notices and also, they the notices will be easily read by the students without availing themselves to the school premises and particularly the specific locations of the noticeboards. The system will enhance the reviewing of feedbacks, goal setting and reminders or alerts can easily be set.

# **Overall Description**

## **Product Perspective**

The product defined within this SRS is a new, self-contained product. Therefore, this segment will seek to describe the general requirements that will be explored later for clear understanding. The requirements specified here for the product are not specific in any manner but highlights the overall issues that will influence the product.

## **Product Features**

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.

1. COD /Dean
2. Manage users
3. Create notice
4. Update notice
5. Delete notice
6. View notice
7. Student
8. View notice
9. Post feedback
10. Manage their Profile

## **User Problem Statement**

The users anticipate that they will help to:

1. Eliminate the paperwork
2. Incorporate an enhanced way to disseminate and access information
3. Promote an improved way to access the feedback from students concerning the notices
4. Tracking of information to ensure that references to the previous notices can easily be made.

## **User Objectives**

The primary users of the system will be Egerton University students, the departmental chairs and the deans of faculties. The level of expertise that will be required is very minimal for crating notices by the specified Egerton University staff as well as a reading by the students because it will be simply done through a graphical user interface facilitated by a click of buttons.

The educational level of the users is expected to be low and the novice user experience will be required. The technical expertise required will be basically low.

1. To develop a system that will allow the chairpersons of departments and Deans to post notices.
2. To develop a system that will allow students to post feedback concerning posts.
3. To develop a system that allows only qualified clients to have working accounts.
4. To develop a system that will allow students to view posts in preference to departments and faculties.

## **Operating Environment**

### **Software Requirement**

1. Microsoft Windows operating system (all versions) or,
2. Linux operating system.
3. Any text editor (preferably atom).
4. Python-Django framework.
5. Html
6. Web browser
7. Internet connection

### **Hardware Requirement (Minimum)**

1. Intel Pentium 600 MHz compatible of faster processor
2. 512 megabytes (MB) of RAM or more (Recommended)
3. 1 gigahertz (GHz) or faster

## **Design and Implementation Constraints**

1. Being a web-based application, the system will only function on a web browser such as Mozilla, Chrome, Safari, and Opera
2. The system will be accessed through the internet connection
3. The client-server database architecture will be used in the implementation
4. Django framework, bootstrap, and JavaScript will be in development.

## **User Documentation**

The E-Noticeboard system will be delivered together with user manuals and online help. The online help will be included in the system to guide users on how to navigate through the system and carry out their tasks. The help menu will also guide the users on how to use the various links and functionalities provided by the system.

## **Assumptions and Dependencies**

1. The assumption made in the system is that all students will enter the correct faculty and department in which they belong to during registration.
2. The notices posted by a given department or faculty only target the students that belong there.
3. All the notices posted by the registrar and Halls department will target all the students in the university.
4. Notices posted by a given faculty or department will not target students from other departments and faculties.
5. Information flow is top-down from the admin to the Faculties, then to the department and then to the students.

## **Constraints**

1. **COD and Dean**
2. Fully understands how the system works.
3. They are in charge of disseminating institutional information or notices to the students.
4. **Students**
5. Have access to internet services
6. Have general knowledge on how the system works
7. Must include their faculties and departments during signing up.

# **System Features**

This chapter will seek to analyze how the users of the system will interact with the system.

## **User Registration**

### **Description and Priority**

This is a high priority feature because each user will be required to have an active account before they can access the system services. This process will be significant to all users of the system

### **Stimulus/Response sequence**

The users will enter their details in the fields provided within a form. The system will perform validations if the entered details are valid as expected. If all the fields are valid, the user will be redirected to another page. But if the entered details are invalid, then an error message will pop in the screen and the user will be asked to try again.

### **Functional Requirements**

1. The registration page has the registration form will have to load before the user can register. All the fields in the registration form will have "TBD" placeholder to indicate what is to be entered in that specific field.
2. The username, password and confirm password fields will have short descriptions of what valid input is expected.
3. The system will redirect to another page and display a "Registration successful" message if there are no errors.
4. If errors are found an error message will be displayed and the user will remain on the same page.

### **Technical Issues**

Implementation of password and username match verification.

### **Dependencies with other Requirements**

No dependencies will be required herein.

## **User Login/Authentication**

### **Description and Priority**

This is a high priority feature as it will allow authenticated users to perform tasks provided by the system. Only upon authentication will a user be able to access the features of the system.

### **Stimulus/Response**

The user will enter the username and password. The system will check if they match with the credentials stored in the database for a registered account. If they match, the user will be redirected to the home page and a successful login message will be displayed. If they do not match, then a login failed message will be displayed and they will remain in the same login page.

### **Functional Requirements**

1. The login page has the login form will load before the user can log in. The two fields in the login form will have "TBD" placeholder to indicate what is to be entered in that specific field.
2. The system will redirect to the home page and display a "login successful" message if there are no errors.
3. If errors are found an error message will be displayed and the user will remain on the same login page.

## **Create Notice**

### **Description and Priority**

This is a high priority feature as it will allow the CODs and Deans to create and post announcements in the system.

### **Stimulus/Response**

The dean or COD will create a notice by entering a message in a textbox. Upon clicking the ‘Send' button, the system will give a "notice created" message.

### **Functional Requirements**

1. The dean or COD must have been able to log in via the login page what is to be entered in that specific field.
2. The system will display “announcement created” message if there are no errors.

## **Delete Notice**

### **Description and Priority**

This is a high priority feature. It will allow the Dean and COD to delete notices from the system once they are deemed to be unnecessary or their effective date expires

### **Stimulus/Response**

Once the dean or COD click the ‘Delete' button, the notice will be removed from the system and the students will not be able to view it again.

### **Functional Requirements.**

1. User must have registered via the register page.
2. User must be logged in through login page.

## **View Notice**

### **Description and Priority**

This is a high priority feature that will allow students to view notices.

### **Stimulus/Response**

The user will click the link to view notices and will be redirected to a page where they will be able to view the posted notices.

### **Functional Requirements**

1. The user must be having an account holder, having gotten it by creating through the register page.
2. The user must be logged in via the login page.

## **View profile**

### **Description and Priority**

This is a low priority feature that will allow users to view their profile or details as stored in the database. They will be able to see their username, full name, and email.

### **Stimulus/Response**

The user will click the link ‘view profile’ and they will be redirected to a page that will display their profile.

### **Functional Requirements**

1. The user must have an account.
2. The user must be logged in.

## **Update Profile**

### **Description and Priority**

This is a medium priority feature that will allow the users to update their profile by changing their name or email.

### **Stimulus/Response**

The user will be redirected to the update profile page once they click the link ‘Update Profile’. Once they change any of their credentials, a ‘profile updated successfully’ message will be displayed and they will be redirected to the ‘view profile’ page.

### **Functional Requirements**

1. The user must have an active account.
2. The user must be logged in.

### **Change Password**

### **Description and Priority**

This is a medium priority feature that will allow the users to change or update their password.

### **Stimulus/Response**

Once the user clicks the ‘Change Password’ link, they will be redirected to another page where a form with 3 fields; current password, new password and confirm new password will be displayed. The user will enter the details and if they are valid, a message ‘password updated’ will be displayed and they will be redirected to the home page. If there is an error, an error message will be displayed and they will remain on the same page.

### **Functional Requirements**

1. The user must be having an account.
2. The user must be logged in.

### **Technical Issues**

1. Message passing will be implemented.
2. Verification of current password, new password and confirm new password must be implemented such that the fields match accordingly.

### **Dependencies with other Requirements**

The new password must be verified if it meets the criteria of being valid such as a short password or numeric password characters only will not be valid.

# **External Interface Requirements**

This section will seek to give a description of the user software, hardware and communication interfaces. It will depict sample images of how the interfaces will look, the messages to be displayed as part of the system.

## **User Interfaces**

1. The system will start with a login page to offer user authentication feature accordingly.
2. The system is designed to have easily accessible buttons for all major activities of the application.
3. Key features and current events will be highlighted to attract user attention
4. The user interfaces will be compatible with all major web browsers
5. All pages will assume standard GUIs with consistent theme and structure

## **Hardware Interfaces**

The software application will interact with the hardware resources of the device on which it will run. While any system can support the core software, a later version of Windows TM or Android TM app has internet access is recommended to ensure real-time activities are synched faster and the whole system runs efficiently.

The system will be hosted on a web server which will be expected to listen to client requests through Port 8080.

## **Software Interfaces**

Windows TM, Android TM app, Linux TM AND Mac OS will support this software application. It is expected to have an involving interaction with an external database server which will be dbsqlite3. This database server will however later be updated to MySQL 5 for production. It may also be integrated with other systems like Egerton University Student Portal if deemed necessary to automatically fetch and use students’ details.

The system will be developed using Django 2.1 framework or any later version.

## **Communication Interfaces**

An internet connection will be necessary for the proper function of the system. The internet connection will allow user devices to connect to the online database, send requests to the web server and receive responses from the web server. The system will use the HyperText Transfer Protocol (HTTP) to transmit data.

The system is also expected to use the File Transfer Protocol in the case where file uploads will be deemed necessary.

# **Other Non-Functional Requirements**

These will be the requirements that will not influence the operation and functionality of the system directly. The section will focus on the performance, security, safety, software quality attributes, usability requirements, and other requirements attributes.

## **Performance Requirements**

The system does not essentially require anything special to accomplish its basic operations. But a complete software with all its set of components running effectively will serve to be its performance requirements. Except for the fact of viewing pre-loaded notices and files in Android applications, an internet connection will be needed so that the system features become available.

Moreover, the database will store unlimited records. Searches are expected to take the shortest time possible of less than 6 seconds.

## **Safety Requirements**

The system will have to implement a desirable level of security to guarantee the safety of the data and information that are to be stored in the database by the software. The database will be trustworthy and non-leakage to ensure no data loss occurs.

## **Security Requirements**

1. The system will implement user authentication absolutely and non-by-passable to ensure nobody gains access to the system without being authenticated. Bo user will be able to access the system without providing proper authentication.
2. In the case of guest users, only public notices and events will be accessible to them.
3. The passwords will have to meet some minimum requirements during account creation. The password will also be hashed and salted to ensure no security breaches.
4. Database access and viewing by the users will not be accepted unless for the admin when deemed necessary or need arises.

## **Software Quality Attributes**

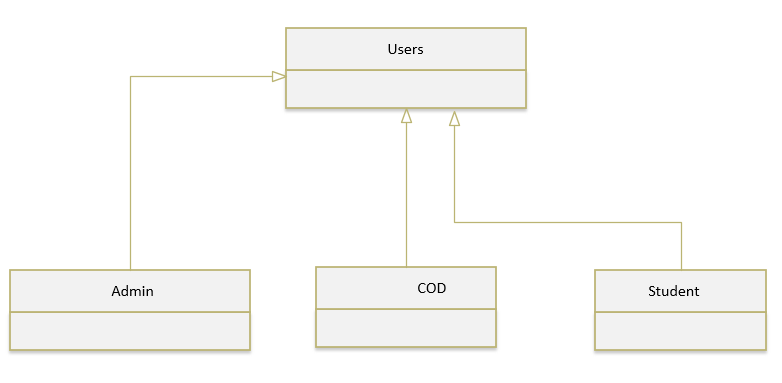
There are several quality attributes of the software which will be significant to the users and developer such as usability, testability, portability, maintainability, and correctness.

For the case of correctness, the developer must ensure a desirable and effective method is used during the design and coding. Usability will be realized by developing a product that is user-friendly as much as possible. In the same way, maintainability and testability will play a major role in the long life of the system.

# **Preliminary Object-Oriented Domain Analysis**

This section will present models of the fundamental structure of objects of the system as described in the requirements. It will analyze and show a structural view of how objects and classes will relate to one another.

## **Inheritance Relationships**



## **User Classes and Characteristics**

### **Abstract or Concrete**

### **List of Super Classes**

Users

Notice

### **List of Sub Classes**

COD

Dean

Student

ViewProfile

UpdateProfile

### **Purpose**

Admin- creates accounts for COD and Dean as well as for general management of the system.

Student- view notices and give comments, feedback or suggestions.

Notice- Address the aspect of notices and announcements.

ViewProfile- used by students, deans, and CODs to view their profile.

UpdateProfile- used by the students, deans, and CODs to update their profile.

### **Collaborations**

### **Attributes**

### **Operations**

### **Constraints**

# **Preliminary Budget and Schedule**

## **Budget**

|  |  |
| --- | --- |
| **Expense** | **Amount** |
| Internet | Kshs.1200 |
| Software download and Activation | Kshs.700 |
| System Documentation | Kshs.400 |
| Stationery | Kshs.200 |
| Miscellaneous | Kshs.350 |
| **Total** | **Kshs.2850** |

## **Schedule**

# **Other Requirements**

There are no other requirements for this system.

## **References**

## **Appendices**

## **Appendix A: Glossary of Definitions, Acronyms, and abbreviations**

COD- Chairperson of Department

OS- Operating System

**Appendix B: Analysis Models**