**CMM007 - INTRANET SYSTEMS DEVELOPMENT**

**PROJECT RESEARCH PAPER SHARING APP**

1.0 INTRODUCTION

This project is the design, development and implementation of a web application that will facilitate the upload, identification, review, monitoring and tracking of research papers among group members. Research activities; report writing, review, editing and other associated task can be daunting particularly with the need to monitor and track the editorial progress of a number of papers within a limited span of time. This project in a bid to address the challenges faced in research editorial cycle will provide an opportunity for the creation of a platform that will support project teams in storing, uploading, assigning and reviewing research papers amongst groups members in a project.

The users of the system will be categorized into three classes, namely: Administrator, Student Team Leader and Students. All users will be registered members of the application i.e. members will be required to log in to perform designated operations. The administrator has the highest privileged role to manage the resources (projects, users and papers) on the platform. He (administrator) only has the right to create users, setup project groups, allocate team leader role and assign members to the project. The student team leader is responsible for allocating papers to members for review, keeping tab on review period and making document opened to all members after review. Students can upload papers that have been identified worthy of review, which will be submitted to the Student team leader, who will allocate it to any member(s) for review. The group member can only view and access the paper that has been allocated to he/her until the review process is completed, after which the paper and review are accessible to every member in the group.

2.0 PROJECT CONSTRAINTS

1. The web application must be hosted on a server with up-to-date code stored in Github
   1. The application will be hosted on a remote server accessible to the course facilitator/marker and other interested parties for use in the University.
   2. All codes will be made available on github at github.com/onwukaok
2. The application must contain both front end (client side) and (server side) code.
   1. The front-end is the presentation layer that the user interacts with to perform tasks on the platform. The front-end is the user-friendly GUI (Graphical User Interface) built to give the user a smooth, intuitive and seamless interaction with the application. It is built using HTML, CSS and Bootstrap.
   2. The back-end is the logic and data layer. It executes user's requests, by performing query based on the request and return response to the user on the front-end.
3. The created web application must contain the following features
   1. A login system
      1. This is a security mechanism to restrict application access to authorized users. This will be created using HTML 5 forms, CSS and Bootstrap on the front-end and PHP/MySQL on the backend.
   2. More than one user role
      1. This is an identity management and authorization measure to. The application will be able to support users with different roles assigned vary degrees of permissions and privileges in what a user is capable of doing. However, no one user can have one role at a time.
   3. Some type of file upload system; and
      1. This will facilitate the uploading of research papers on the server for storing and download at request. This will be implemented using HTML 5 file upload facility on the front-end and PHP for executing upload to remote server on the backend.
   4. A system for user to input data that is stored and then recalled from a database.
      1. The system will support data storage (documents in editable format) and retrieval using PHP as the application engine and MySQL as the database server.

3.0 SYSTEM REQUIREMENTS SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

These requirements help clearly define the set of functions the application will perform.

|  |  |  |
| --- | --- | --- |
| ID | REQUIREMENT STATEMENT | PRIORITY RANKING (MoSCoW) |
| R1 | The system shall be structured into front-end and back-end. The front-end is the presentation layer which the user sees and interacts with, while the back-end is the logic and data layer which performs the behind the scene operation and data storage. | Must have |
| R2 | The front-end shall present the users with user-friendly GUI to interact with. | Must have |
| R3 | Each user of the system shall be required to login to use the system. | Must have |
| R4 | Each user shall belong to a role either as an Administrator, Student team leader or Student which determines the task the user can perform. | Must have |
| R5 | The Administrator shall be able to create projects, users and assign users to projects. | Must have |
| R6 | The Administrator shall add every created user to one or more project group | Must have |
| R7 | The Administrator shall assign the role of the Student Team leader. | Must have |
| R8 | Student team leader shall be able to upload research paper and assign paper to students for peer review. | Must have |
| \*R9 | Students shall be able to upload papers which will be submitted to the student team leader, who shall then allocate paper to any number of students for review | Should have |
| R10 | The Student team leader specifies the duration in days for the review, which counts down to the deadline. | Should have |
| R11 | Students will be able to download papers, upload reviewed papers with annotations and add comments for the Student team leader and other students to view. | Should have |
| \*R12 | The system shall notify and alert the student three(3) days before the deadline on the platform and on their email. | Could have |
| R13 | The system shall help the student clearly see the research papers he/she has reviewed in the past and the one(s) currently being reviewed, with the dates of allotment and submission. | Could have |
| R14 | The system shall be able to store records of users, projects, uploaded papers, papers assigned to users, reviewed papers and comments made by users on papers. | Must have |
| \*R15 | Other members of the project group shall be able to see the reviewed paper and comment on it. | Should have |
| \*R16 | Student team leader shall be notified on the platform and email when a new research paper is uploaded, when a reviewed paper is submitted. | Could have |
| \*R17 | Students shall be notified on the platform and email when he/she is assigned a research paper to review. | Could have |
| \*R18 | Students are expected to within the stipulated period review the assigned document, upload the reviewed document with annotations, give additional notes (optional) in a submission to the team leader. | Must have |
| R19 | Reviewed papers shall be accessible to all members of a project group. | Should have |
| R20 | The system shall keep track of the papers the user has to review at any given period of time. | Could have |
| R21 | Uploaded papers shall be stored on the server, and make available when requested. | Must have |
| \*R22 | The system shall authenticates users using the university email address and choice password. | Should have |
| 23 | Student team leader shall not be able to create project group | Won't have |

3.2 NON-FUNCTIONAL REQUIREMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| ID | REQUIREMENT CATEGORY | REQUIREMENT STATEMENT | PRIORITY RANKING (MoSCoW) |
| R1 | Accessibility | The application shall be easily accessible to all users within their granted privileges. | Must have |
| R2 | File Upload limit | The size of files to be uploaded will not be more than 10MB | Should have |
| R3 | Security | All users shall be authorized to use the system. Functions to be performed by user are restricted to role. | Must have |
| R4 | Performance | The application shall be efficient in output delivery. | Should have |
| R5 | Backup and recovery | The system shall be resilient to disaster and be recovery in cases of disaster. | Must have |
| R6 | User friendliness | The application shall be intuitive, easy to navigate with good user experience. | Should have |
| R7 | Portability | The application shall render and function uniformly on different operating system and devices. | Could have |
| R8 | Scalability | The application shall be able to handle, cope and accommodate the increasing need of users and resources. | Should have |
| R9 | Response time | The application shall be light and load fast. | Could have |
| R10 | Reliability | The application shall perform to specification and expectation. | Must have |
| R11 | Privacy | The confidentiality of all users shall be maintained. | Should have |
| R12 | Usability | The system shall be useable in meeting the needs of the different categories of users. | Should have |
| R13 | Manageability | The application shall be flexible in managing usage, upgrade and maintenance. | Could have |
| R14 | Data Integrity | The data on the application shall be consistent, accurate and reliable at all times. | Must have |
| R15 | Capacity | The system shall have the storage space to accommodate the growing size of users and resources. | Should have |
| R16 | Availability | The system shall be available to be use at all times. | Could have |
| R17 | Interoperability | The application shall be able to function with other hardware and software features and facilities. | Could have |

3.3 WORKFLOW

**LOGIN**

**Administrator**  
*App Management*Setup project, create users, assign roles

**Student Team Leader**  
*Group Management*Upload paper, assign reviewer, publish

**Student***Reviews*Upload papers, review, comments

**Upload Research Paper**Upload paper to be reviewed

**Assign Paper to Reviewer**Assign paper to student reviewer, set deadline

**Review Paper**Review paper, comment on reviewed paper

**Submit Reviewed Paper**Complete the review

**Publish Paper**Close review and publish to public archieve

**Project Group Visibility**Members can view, comment on reviewed research paper