ONLINE DOCUMENT (SPECIFIC TO NBA & NAAC) MAINTENANCE SYSTEM

Design Document

31.05.2021

Mentored By: DR. SMRITI KUMAR SINHA

Submitted By:

HRISHIKESH GOGOI

CSB19030

B. Tech CSE (6th Semester)

Abstract

In an information intensive project, the document maintenance and project maintenance are closely connected together. The project result is usually a number of documents describing the product to be developed, re-engineered or delivered. When the organization can define the output documents and necessary documents in different intermediate steps, the progress of the project can be tracked using the document maintenance system.

Specific tools and techniques for manipulating and communicating information on building projects have been developed, which are the core of this chapter. In the debate on the principles of document maintenance systems the concept of information handling will be explored.

Overview

The Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

1. Introduction

1.1 Purpose and Scope

The purpose of this document is to build a web-based application for university authority so as to maintain a common platform for sharing NAAC or NBA reports of various departments under them.

The objective of this process is as follows:

1) To develop a full stack website for university authority to share notifications to the departments and in response to it the departments will be able to share their reports.
2) Also enhance the feature for feedback by the deans for the departments.
3) The project should be very easy to use, enabling even a novice person to use it.

1.2 Project Executive Summary

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design has been prepared. This app is intended for university authority and department representatives are the main target audience for this project. This application would be created with the motivation to simplify the task of departments and university authority to interact at a common platform to share reports and notifications. The application will be led by a single administrator who will be in charge of every school of the deemed university. The admin will be able to create sub sectors under him such as the school of Dean for every school. Every school of the deemed university will be having one head who will be sharing the different reports to the higher authority and looking after the sub sector under him. The dean will be able to create heads of the different departments under him. The head will be sharing their annual NAAC and NBA reports to the dean which will be again shared to the university authority. The deans will be able to post notifications demanding reports to their affiliated departments and in response to it the departments will be able to share their reports in given specified places. Also enhance the feature for feedback by the university authority for the departments.

1.2.1 System Overview

This section describes the system in narrative form using non-technical terms. It provides a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams, if applicable, show interfaces to external systems.

1.2.2 Design Constraints

This section describes any constraints in the system design and includes any assumptions made by the project team in developing the system design. The system must be connected to the internet. Users must have a valid email id and an internet browser.

1.3 Document Organization

This section describes the organization of the Systems Design Document. The purpose of this document is to build a web-based application for university authority so as to maintain a common platform for sharing NAAC or NBA reports of various departments under them. The main issue with our physical document maintenance system is that we either did not find or lost the required file. When a user is finished with a document, it is stored in a physical document maintenance system. The majority of users disregard the organizational rules for filing documents in the records centre file rooms. Users have a tendency to hoard information once they have obtained the documents necessary for their activity. At the most, they will complete all of the records associated with a project at the end of the activity. There is no added value in request, receipt, and disposition systems for documents that are not accessible in file folders. Furthermore, the physical document maintenance system is paper-based, resulting in non-traceability, possible loss, and information inaccessibility. To address this issue, I plan to develop a web-based application that will allow me to store these documents digitally in a unified platform. This platform will be available to the University Authority as well as departments. The end product will be a full stack web page with all the essential features that can be accommodated in it.

1.4 Points of Contact

This section provides the organization code and title of the key points of contact (and alternates) for the information system development effort. No document conventions are being used at this time. The performance of the application will depend on the fact of uploading files by multiple clients at a time. This is tackled using the concepts of multi-threading in the application.

1.5 Project References

This section provides a bibliography of key project references and deliverables that have been produced before this point. Till now no reference has been used. References are subjected to future use if required.

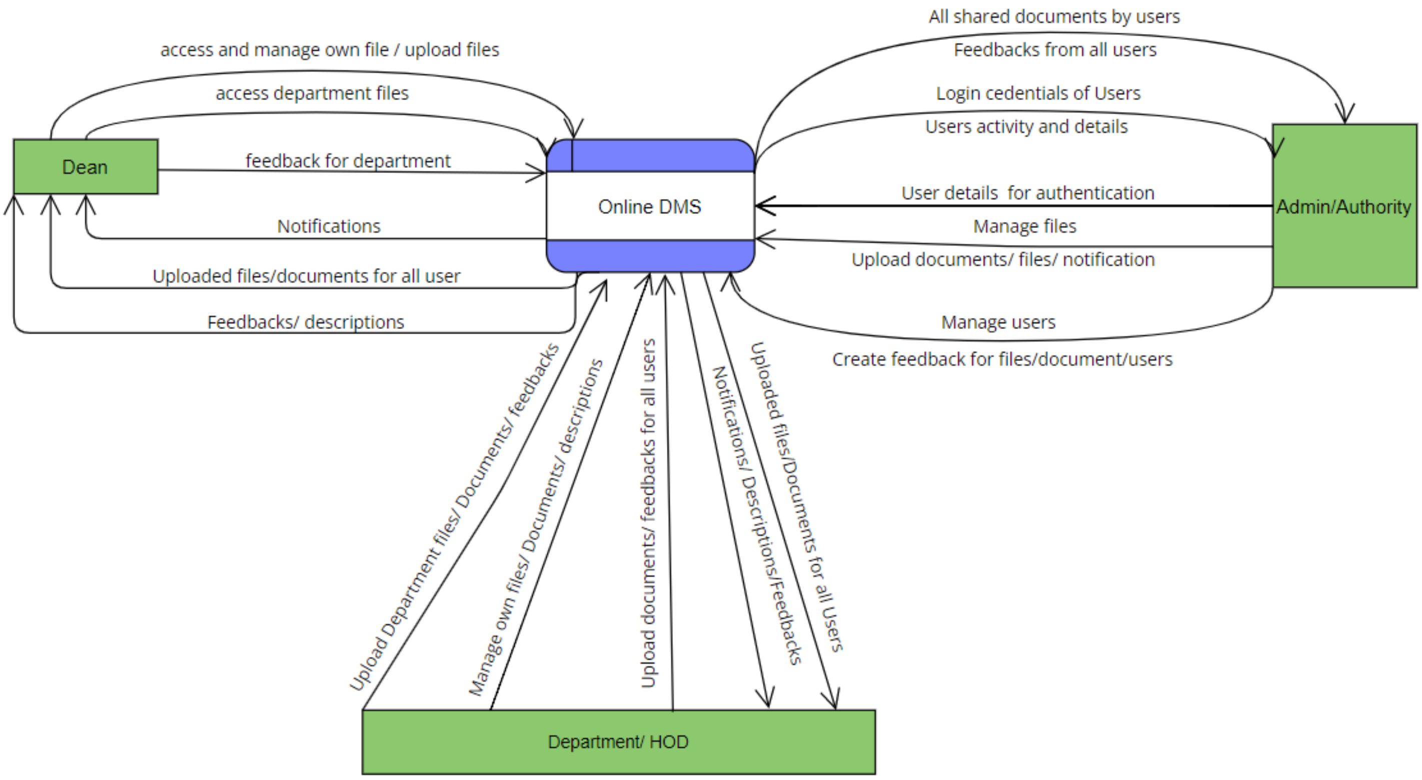
1.6 Glossary

Till now no abbreviations have been used. Abbreviations are subjected to future use if required.

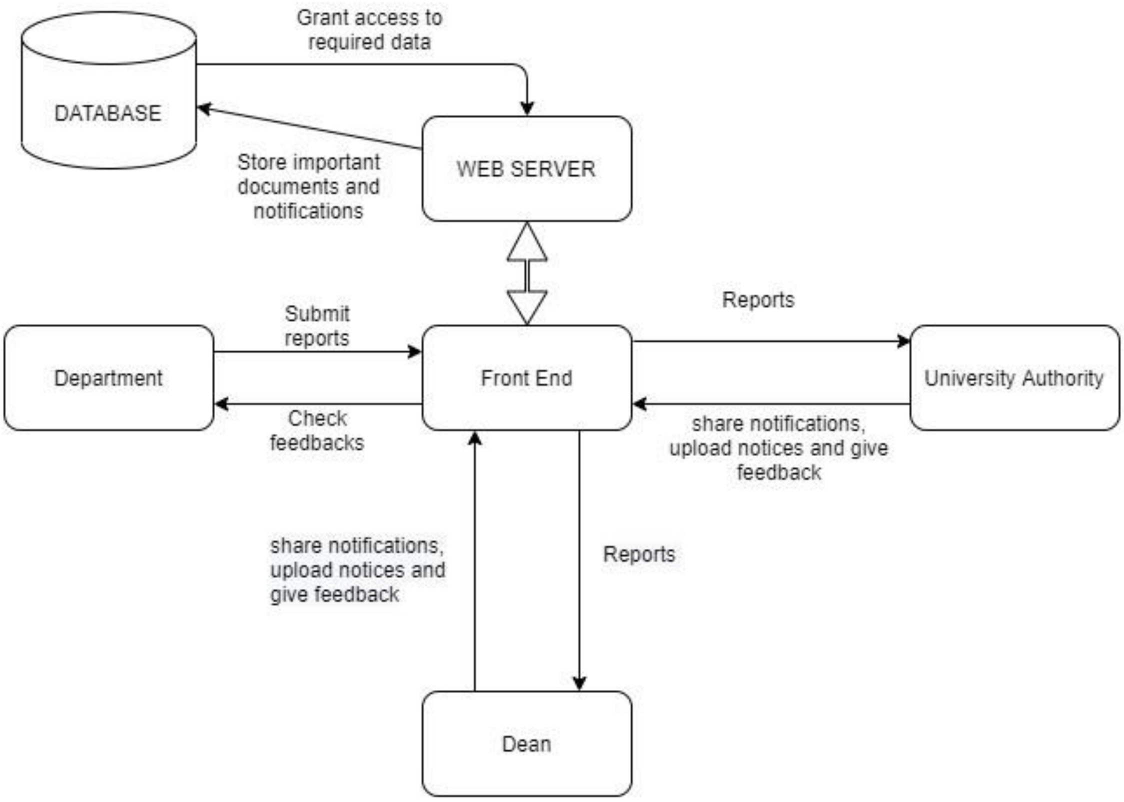
2. Context diagram and system architecture

In this section, the system and/or subsystem(s) architecture for the project is described. The implementation of the software should follow a similar structure, the tech stack that shall be used in each module is listed in the *tech stack requirement* section.

Context Diagram


Dean
Online DMS
Admin/Authority
Department/ HOD
access and manage own file / upload files
access department files
feedback for department
Notifications
Uploaded files/documents for all user
Feedbacks/ descriptions
All shared documents by users
Feedbacks from all users
Login cedentials of Users
Users activity and details
User details for authentication
Manage files
Upload documents/ files/ notification
Manage users
Create feedback for files/document/users
Upload Department files/ Documents/ feedbacks
Manage own files/ Documents/ descriptions
Upload documents/ feedbacks for all users
Notifications/ Descriptions/Feedbacks
Uploaded files/Documents for all Users

System Architecture


DATABASE
WEB SERVER
Grant access to required data
Store important documents and notifications
Department
Front End
University Authority
Dean
Submit reports
Check feedbacks
Reports
share notifications, upload notices and give feedback
share notifications, upload notices and give feedback
Reports

2.1 System Hardware Architecture

In this section, describe the overall system hardware and organization.
During the development phase the hardware used:

Operating System: Windows

Browser: By default, the generated project supports all modern browsers. Support for Internet Explorer 9, 10, and 11 requires polypills. Using Chrome and Edge for debugging.

Internet Connectivity: Yes, essential requirement for transfer of data

2.2 System Software Architecture

In this section, describe the overall system software and organization. This mostly includes a list of software modules (this could include functions, subroutines, or classes), computer languages, and programming computer-aided software engineering tools.

Most of the functions are analysed as below:

2.2.1 Login Admin Accounts

This Feature will enable the overall admin to login to their own account so as to create the structure of the system.

2.2.2 Creating structure

This Feature will enable the admin to create their subsector admins, i.e. deans after they have logged in using valid email and password. After this admin can assign sub sectors admins.

2.2.3 Dean Login

This Feature will enable the sub sector admins whose access was granted by the overall admin to create their accounts so that they can take charge of the sub sectors dean. The dean will have to sign up with their valid email id and a valid as per format password.

2.2.4 DEPARTMENT Login

This Feature will enable the affiliated departments to create their accounts in the specified domain sectors.

2.2.5 Sharing Notification

This Feature will enable both the university authority and deans to share notifications as per their sectors.

2.2.6 Submitting Reports

This Feature will enable the affiliated departments to upload documents assigned to them by the officials.

2.2.7 Checking Reports and Sharing Feedback

This Feature will enable both the university authority as well as deans to download and check the reports and share feedback.

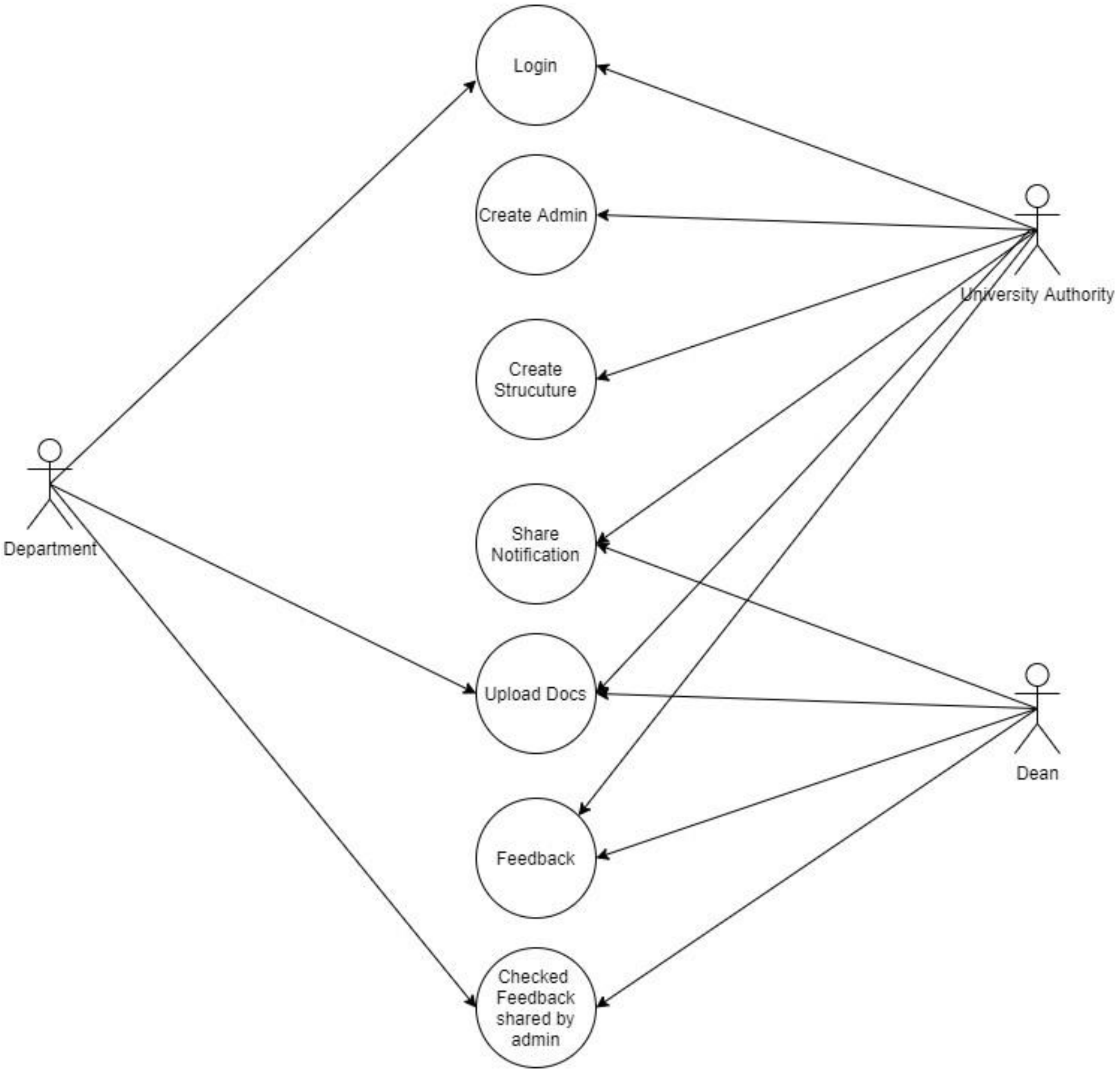
2.2.8 Viewing Feedback

This Feature will enable affiliated departments to check the feedback shared with them.

2.2.9 Maintaining Reports

This Feature will enable the university authority to maintain the reports of all the sectors under him/her. Each sector admin will be able to maintain reports of their particular domain keeping privacy among the domains. At the end, each department can check their respective reports and feedback shared with them.

2.2.10 Overall view

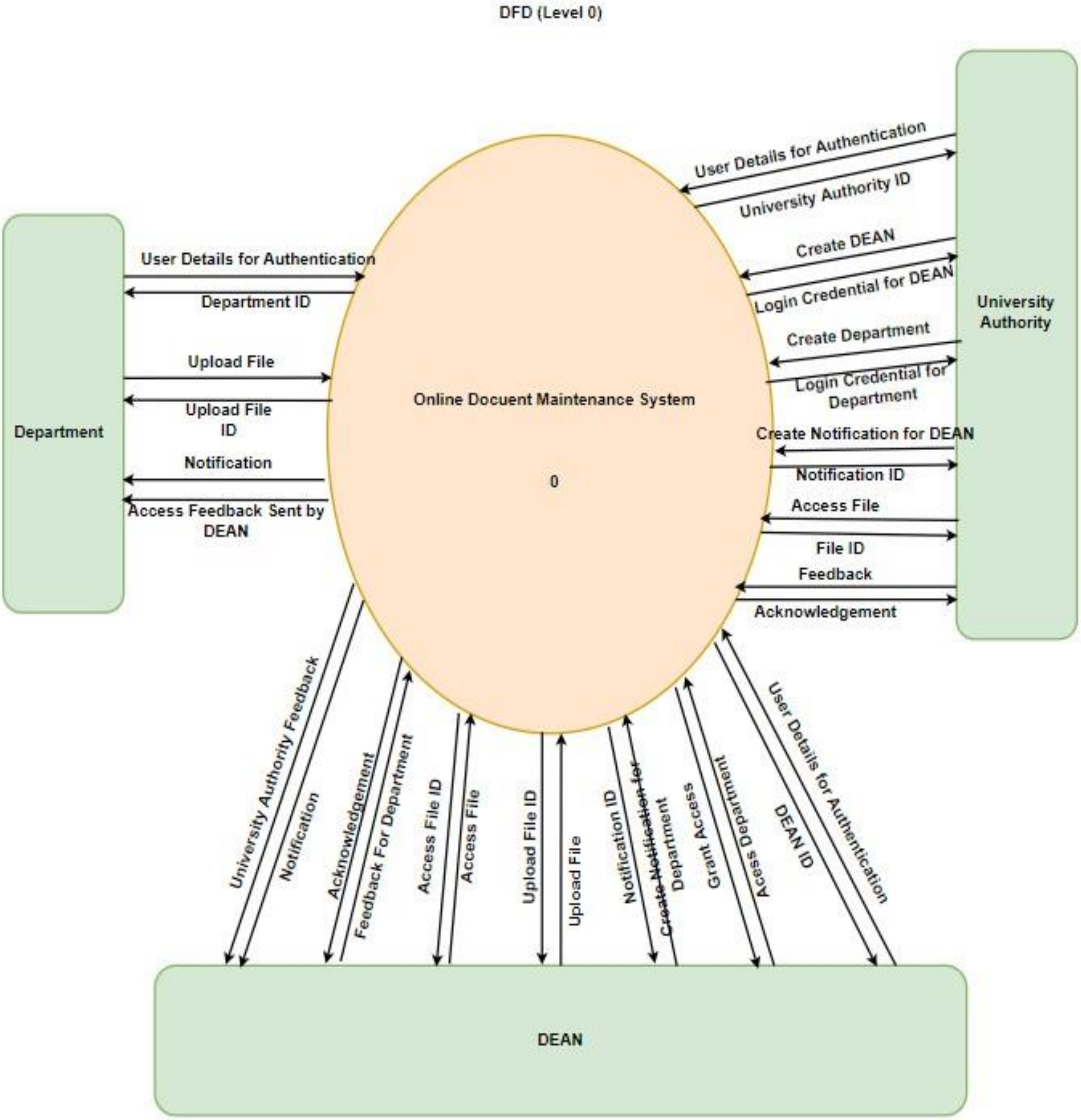


2.3 Internal Communication Architecture

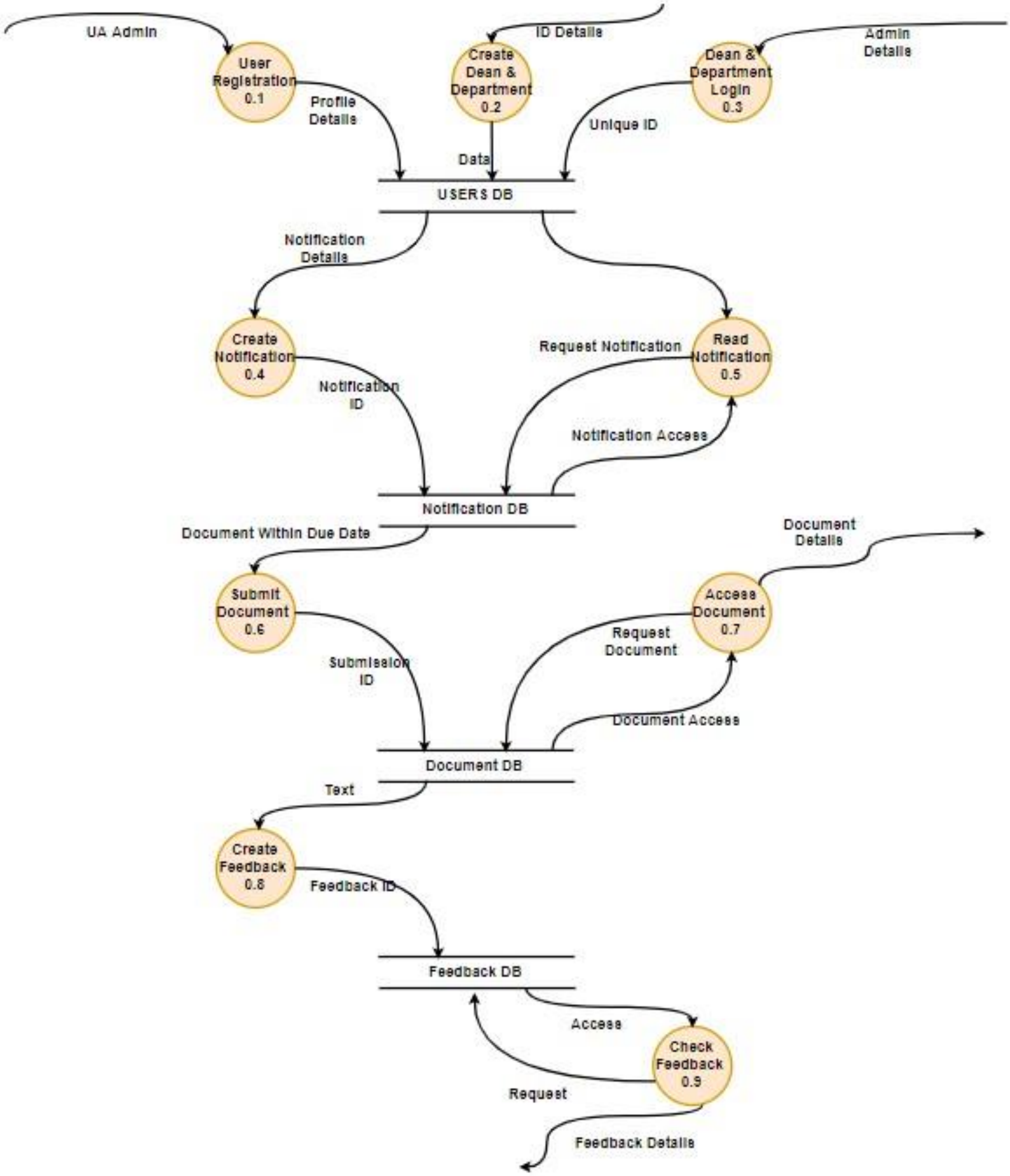
In this section, describe the overall communications within the system; for example, LANs, buses, etc. As of now, no such communications have been used.

3. DATA FLOW DIAGRAM(DFD) AND ER DIAGRAM

3.1 Data Flow Diagram (DFD)

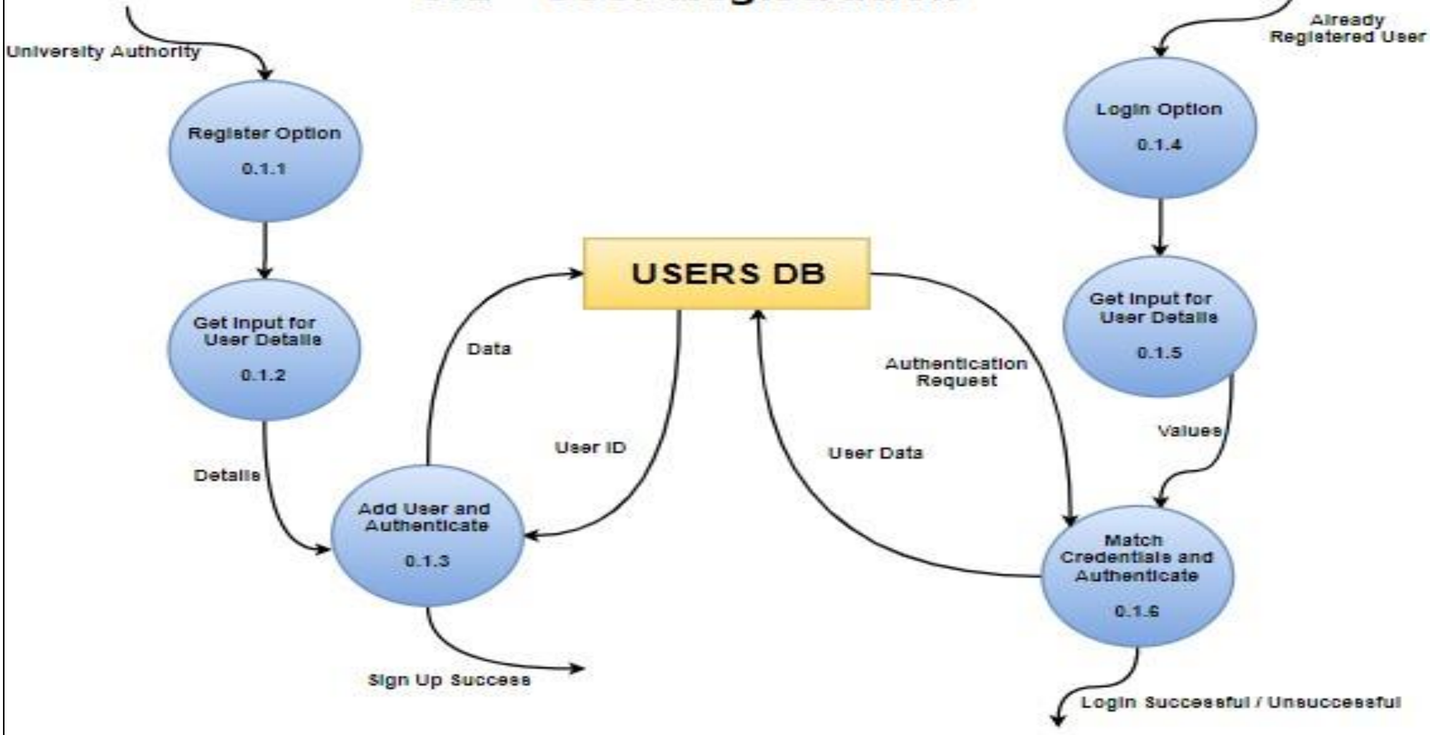


DFD (Level 1)

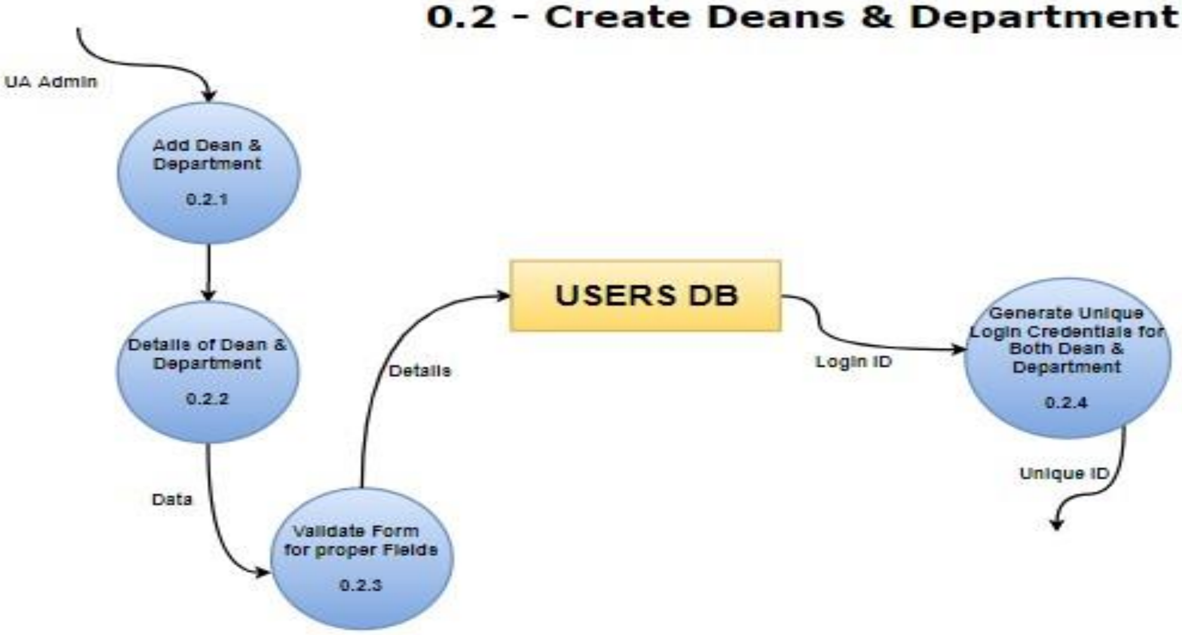


UA Admin
User Registration 0.1
ID Details
Create Dean & Department 0.2
Admin Details
Dean & Department Login 0.3
Profile Details
Data
Unique ID
USERS DB
Notification Details
Create Notification 0.4
Request Notification
Read Notification 0.5
Notification ID
Notification Access
Notification DB
Document Within Due Date
Document Details
Submit Document 0.6
Access Document 0.7
Request Document
Submission ID
Document Access
Document DB
Text
Create Feedback 0.8
Feedback ID
Feedback DB
Access
Check Feedback 0.9
Request
Feedback Details

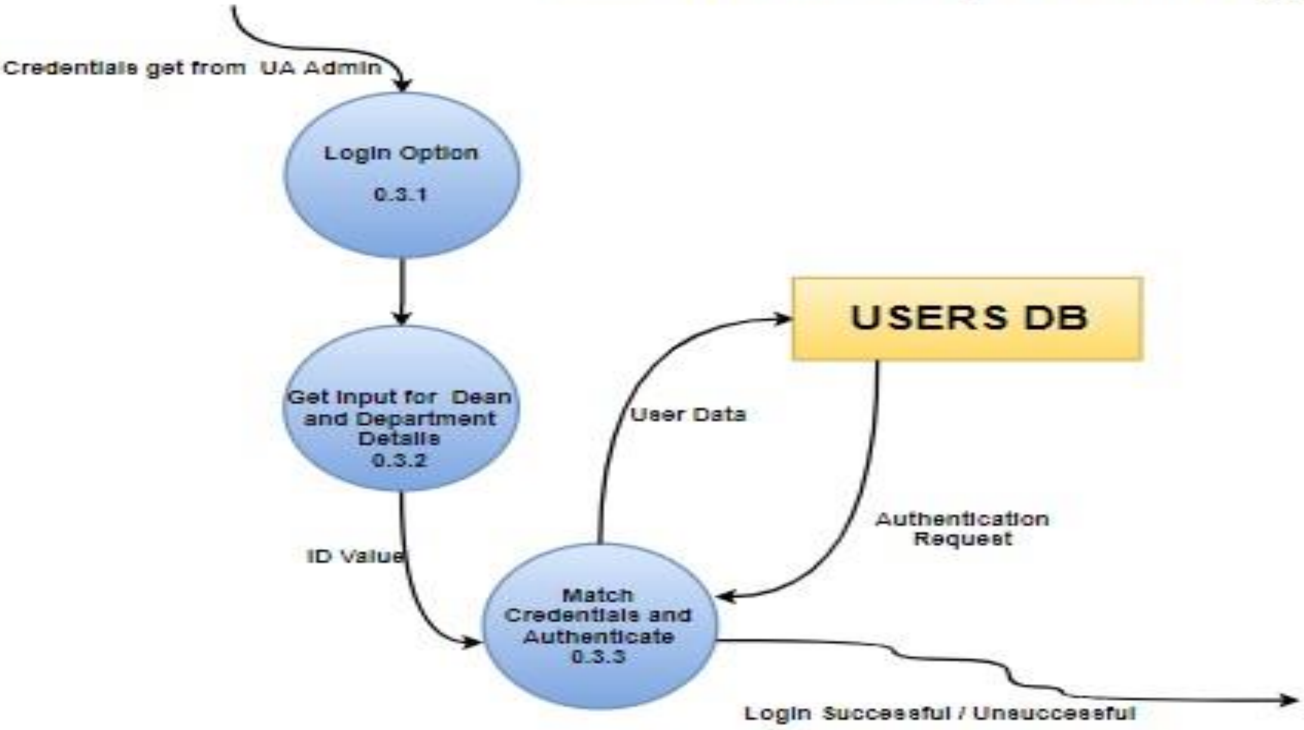
DFD (LEVEL 2)
0.1 - User Registration



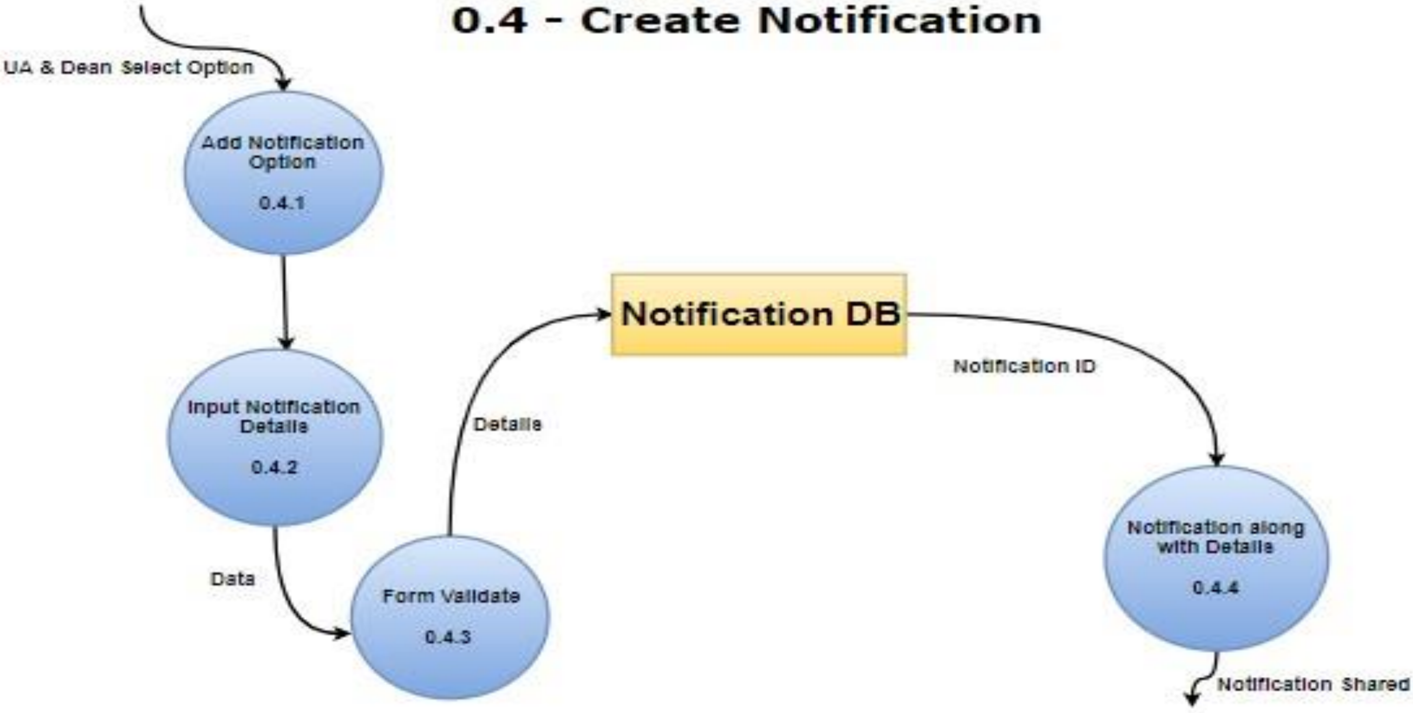
DFD (LEVEL 2)
0.2 - Create Deans & Department


UA Admin
Add Dean & Department
0.2.1
Details of Dean & Department
0.2.2
Data
Validate Form for proper Fields
0.2.3
Details
USERS DB
Login ID
Generate Unique Login Credentials for Both Dean & Department
0.2.4
Unique ID

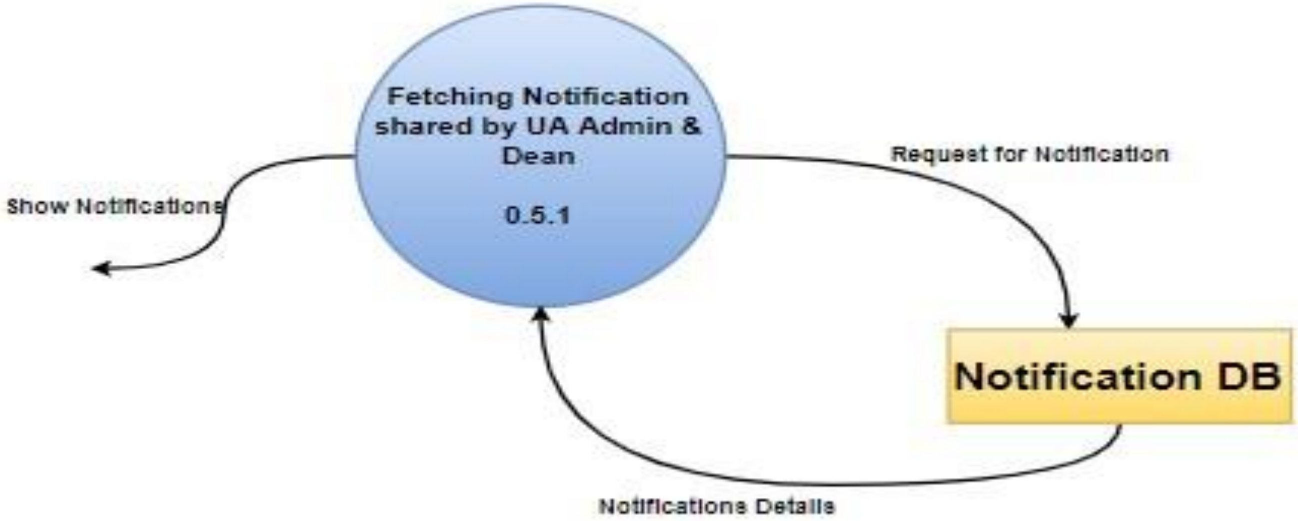
DFD (LEVEL 2)
0.3-Dean & Department Login


Credentials get from UA Admin
Login Option
0.3.1
Get Input for Dean and Department Details
0.3.2
ID Value
USERS DB
User Data
Authentication Request
Match Credentials and Authenticate
0.3.3
Login Successful / Unsuccessful

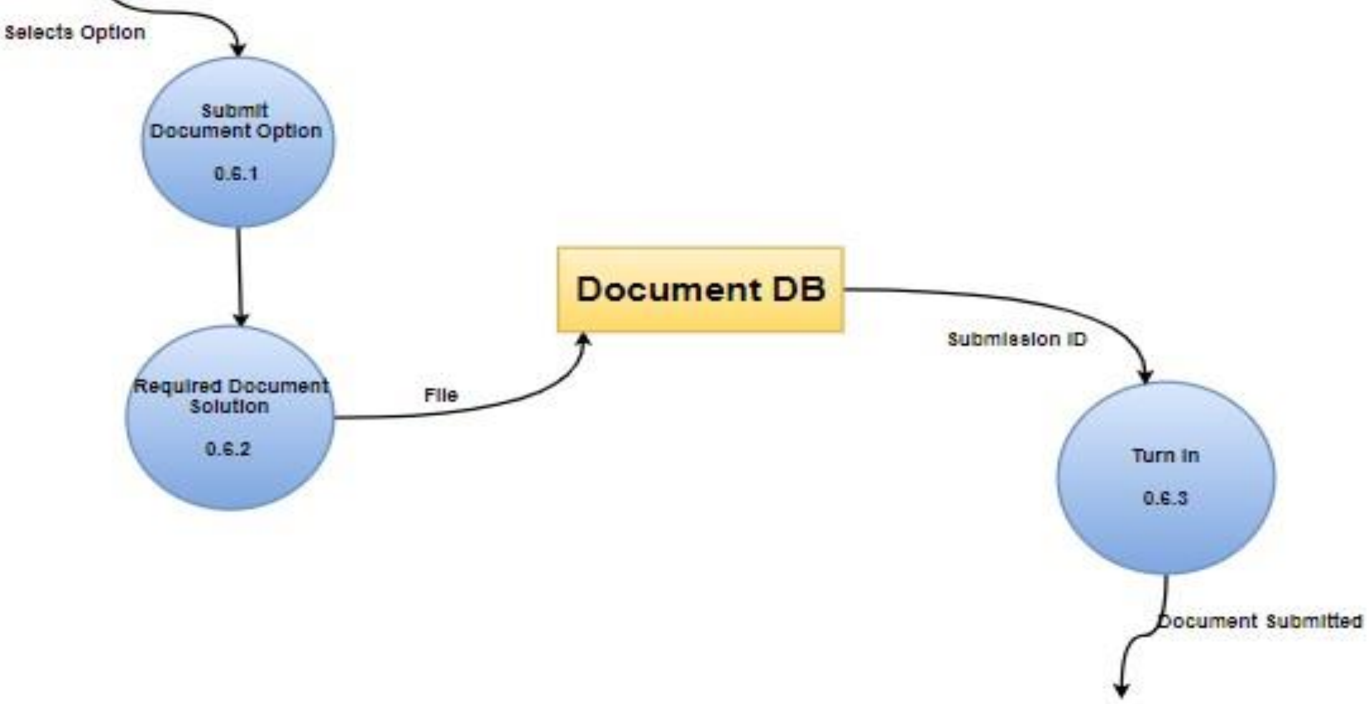
DFD (LEVEL 2)
0.4 - Create Notification



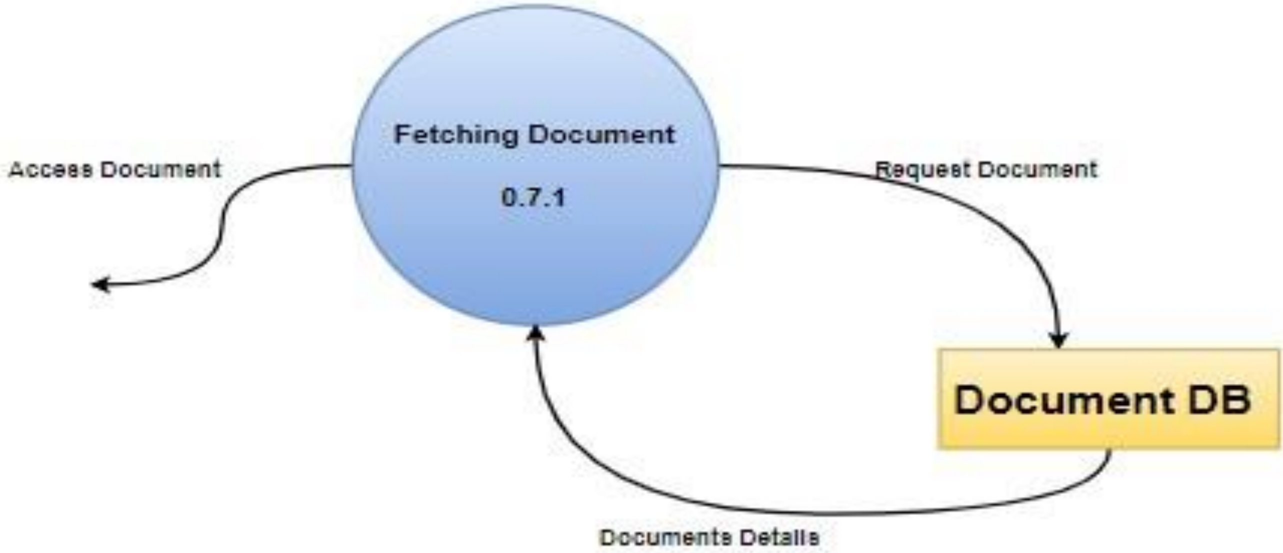
DFD (LEVEL 2)
0.5 - Read Notification



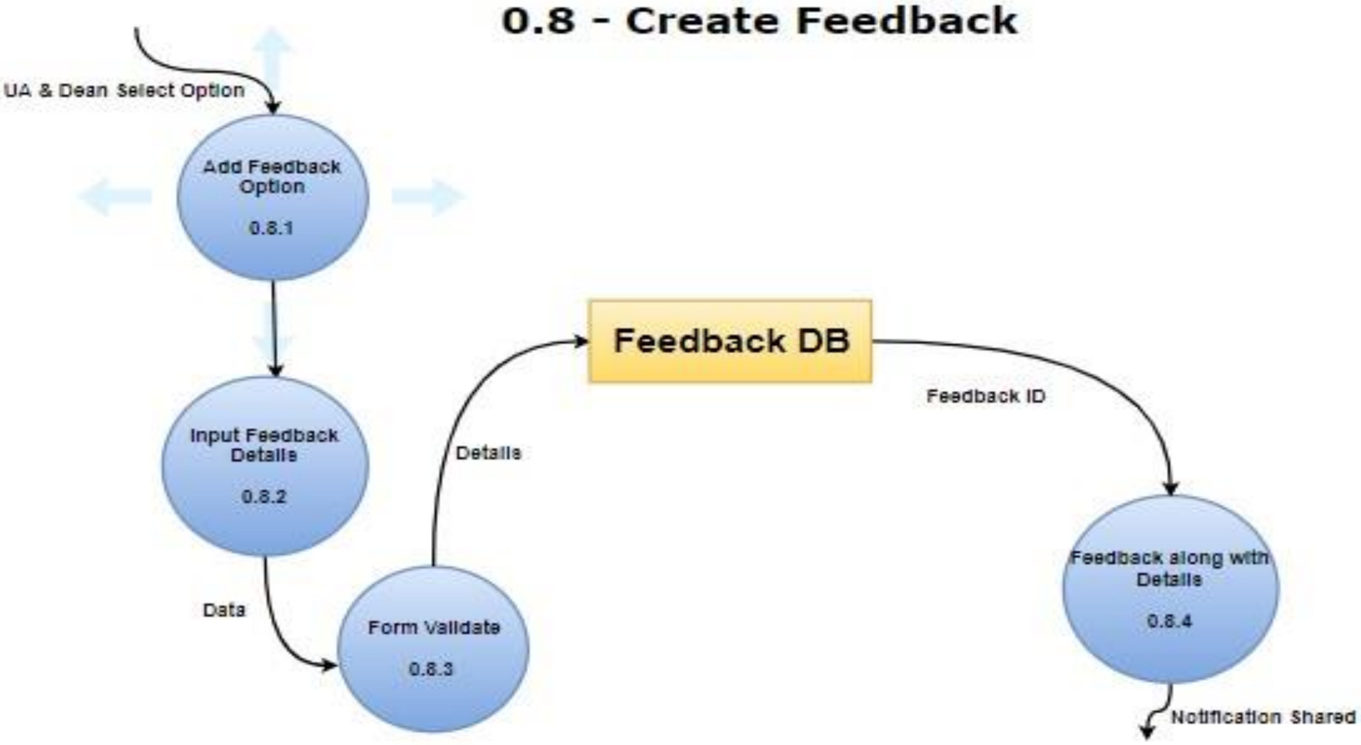
DFD (LEVEL 2)
0.6 - Submit Documents



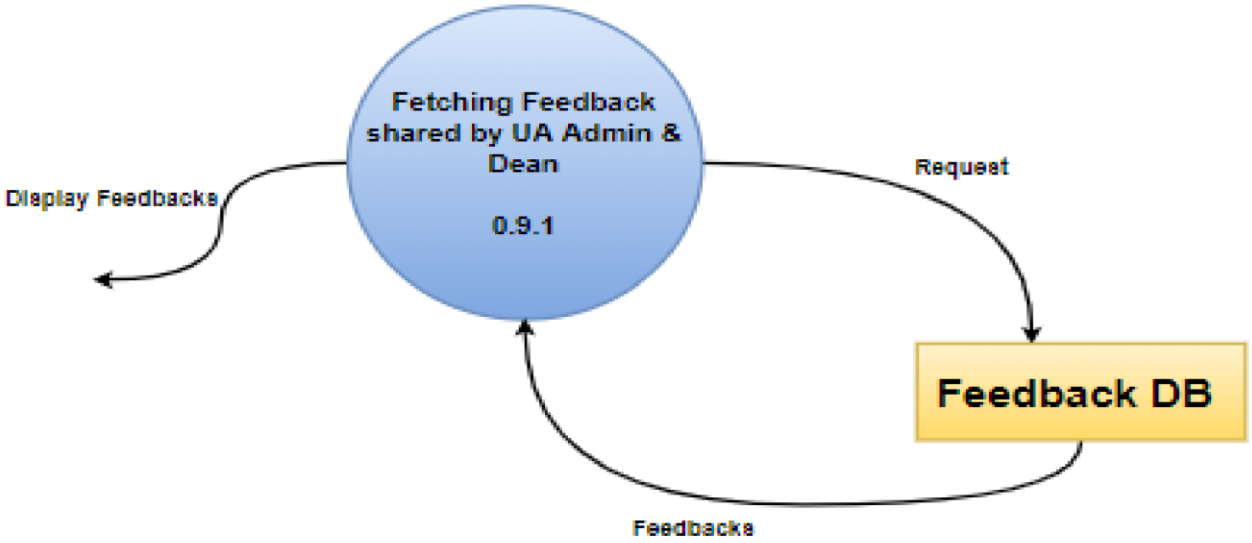
DFD (LEVEL 2)
0.7 - Access Document



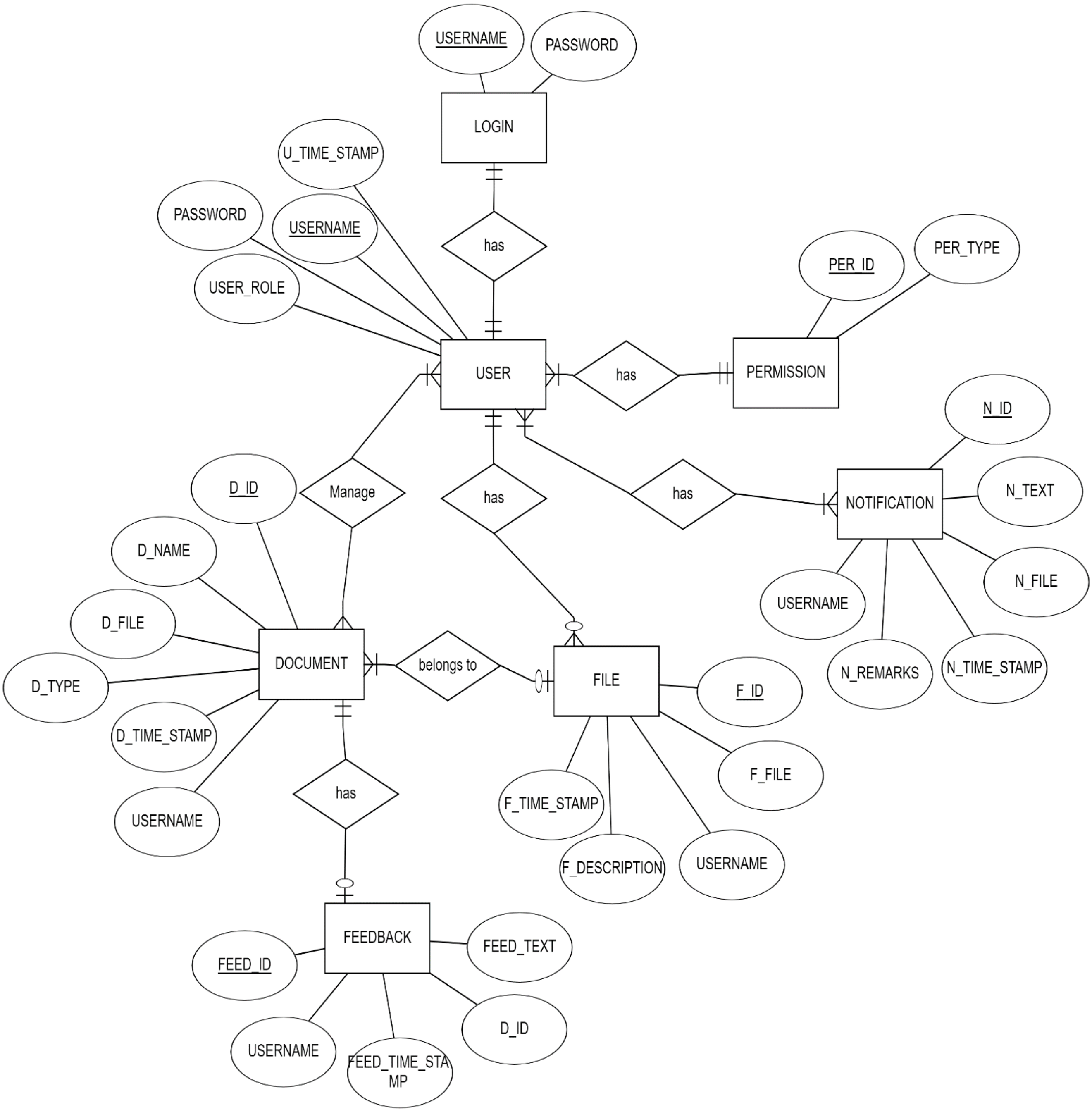
DFD (LEVEL 2)
0.8 - Create Feedback



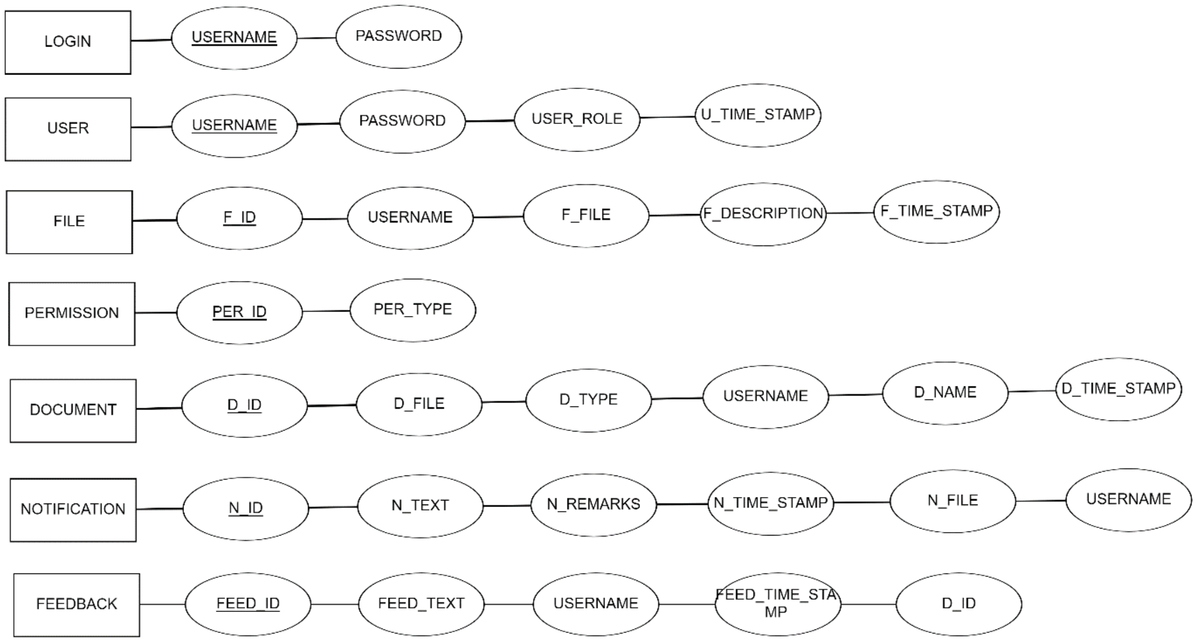
DFD (LEVEL 2)
0.9 - Check Feedback



3.2 ER Diagram



4. DATABASE MANAGEMENT SYSTEM FILES



5. Tech Stack Requirements

I plan to use the following technologies in my project:

5.1 HTML/CSS/JS

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. I intend to use it for the frontend purpose.

5.2 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. User Interface interacts with the smart contracts through bootstrap.

5.3 SQL

SQL stands for Structured Query Language. It is designed for managing data in a relational database management system (RDBMS). It is pronounced as S-Q-L or sometimes See-Qwell. SQL is a database language, it is used for database creation, deletion, fetching rows, and modifying rows, etc. Details and other files would be stored in the database.

5.4 PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP 7 is the latest stable release. I intend to use PHP for my API calls.