**[Dof3aa]**

**By  
Mohamed Badawy Sayed**

**Mahmoud Ahmed Khalil Ibrahim**

**Ahmed Nageh Abbas  
Mostafa Ahmed Hasan El-Gelany**

**[](https://github.com/Mohamed-badawy-sayed/CMS)**

[**GitHub Repo**](https://github.com/Mohamed-badawy-sayed/CMS)

**Dr.**

**Mohamed Ramadan Saady**

**Mohamed Ahmed El-Kholy**

**Table of Contents**

**Requirements 2**

**Functional requirements** 2

**Non-Functional requirements** 4

**Use Case Diagram** 5

**Use Case Scenario** 6

**ERD Diagram 15**

**Class Diagram 16**

Sequence Diagram **17**

**Introduction**  
Dof3aa is a web application designed to foster a cohesive and organized community among faculty’s students. It serves as a central hub for all faculty-related updates, materials, and assignments, ensuring that students remain fully informed and engaged with their academic environment.

This comprehensive documentation offers a detailed insight into the system, covering its scope, requirements, dependencies, and architecture.

**Scope**

The dof3aa Web Application will include features for both admins and students. Admins will be able to create and manage courses, upload course materials, and add announcements. Students will be able to enroll in courses, and access course materials.

**Requirements**

**Functional requirements**

**System Features: Authentication and Authorization**

General Authentication:

* **User Registration & Login**: Users can register for an account and log in. The login process is secured, ensuring that user credentials are protected.

Role-Based Access Control:

* **Role Differentiation**: The system distinguishes between two primary roles: Admin and Student. Each role has unique permissions and access rights within the system.

**Admin Role Capabilities:**

* **Course Management**: Admins have the ability to create new courses within the system or delete courses.
* **Content Management**: Admins can upload, update, and remove course materials as needed.
* **Announcements**: Admins have the authority to create, edit, and delete announcements for courses.
* **Enrollment Oversight**: Admins manage student enrollments, including adding or removing students from courses.
* **Role Assignment**: Admins can change the roles of users, promoting students to admins or demoting admins to students, as necessary.
* **Assignment Management**: Admins can create, modify, and delete assignments within courses and add deadlines to tasks.
* **User Management**: Admins have the power to remove students or other admins from the system, with the exception of the original course creator.

**Student Role Capabilities:**

* **Course Enrollment**: Students can enroll in available courses of their interest.
* **Resource Access**: Students have the ability to browse through course materials, announcements, and assignments.
* **Material Downloads**: Students can download course materials for offline study.
* **Course Withdrawal**: Students have the option to unenroll from courses if they choose.

**Non-functional requirements:**

**1. Security:**

* **Data Encryption:** All personal information and user credentials must be securely encrypted to protect against unauthorized access.
* **Access Control**: Implement strict access control measures to ensure that only authenticated students can access their respective courses, materials, and announcements.

**2. Performance:**

* **Quick Load Times:** The platform should load content and respond to user requests within 2 seconds under typical usage conditions.
* **Scalability:** Capable of scaling dynamically to accommodate increases in user numbers and data volume, supporting at least 10,000 simultaneous student logins.

**3. Usability:**

* **Intuitive Design:** The platform must feature an intuitive, straightforward design enabling students to find and use features without extensive guidance.
* **Support Resources:** Provide detailed help documentation and resources, allowing students to resolve common issues independently and understand platform functionalities.

**4. Reliability:**

* **High Availability:** Achieve 99.9% uptime, not including planned maintenance windows, to ensure students can always access their materials and announcements.
* **Consistent Data Integrity:** Employ mechanisms to ensure that uploaded materials and entered data are consistently validated, preventing errors or corruption.
* **Robust Backup and Recovery:** Regularly back up data with a reliable recovery strategy in place to quickly restore service in case of system failure or data loss.

**5. Maintainability:**

* **Modular Architecture:** Design the system with a modular approach to simplify updates, maintenance, and the rollout of new features.
* **Comprehensive Documentation:** Maintain detailed documentation of the system’s architecture, codebase, and APIs to facilitate efficient maintenance and future enhancements.
* **Future-proof Technologies:** Choose widely supported and contemporary technologies for development to ensure long-term viability and community support.

**Dependencies**

The application may depend on third-party services for features such as authentication.

**System architecture**

The dof3aa Web Application will be built using a client-server architecture. The front end will be developed using modern web technologies such as HTML5, CSS3, and JavaScript. The back end will be implemented using a server-side framework such as Node.js, with a not only relational database (e.g., mongo db) for data storage.