

Contents

[1. Introduction 3](#_Toc178373374)

[2. System Architecture 3](#_Toc178373375)

[3. User Roles and Responsibilities 3](#_Toc178373376)

[4. Functional Requirements 4](#_Toc178373377)

[5. Security and Authentication 4](#_Toc178373378)

[6. Data Management and Storage 4](#_Toc178373379)

[7. Wireframes and UI Design 4](#_Toc178373380)

[8. Deployment and Build Process 4](#_Toc178373381)

[9. Future Enhancements 5](#_Toc178373382)

# 1. Introduction

The Siyafunda project is an online platform facilitating resource sharing between educators and students. This technical documentation aims to provide a detailed description of the system's architecture, key functionalities, user roles, data management, and security protocols.

# 2. System Architecture

The Siyafunda platform is built as a web-based application, accessible on both desktop and mobile platforms. It uses an ASP.NET framework with C# for server-side logic and SQL for backend data storage. The application is hosted on Microsoft Azure, making use of cloud services for storage and deployment.

Key Components:

* ASP.NET for page rendering and user interface
* C# for business logic and server-side processing
* SQL Database for data storage
* Azure cloud hosting for deployment
* React for front-end integration

# 3. User Roles and Responsibilities

The system supports various user roles, each with specific functionalities:

* System Admin: Full access to all functionalities, including managing user roles and monitoring data.
* System Developer: Responsible for maintaining the system and deploying updates.
* Moderator: Handles document moderation and user management for institutions.
* Educator: Responsible for uploading resources, creating quizzes, and managing class data.
* Student: Can access shared resources, receive announcements, create timetables and participate in quizzes or assignments.

# 4. Functional Requirements

The Siyafunda platform provides a range of functionalities including:

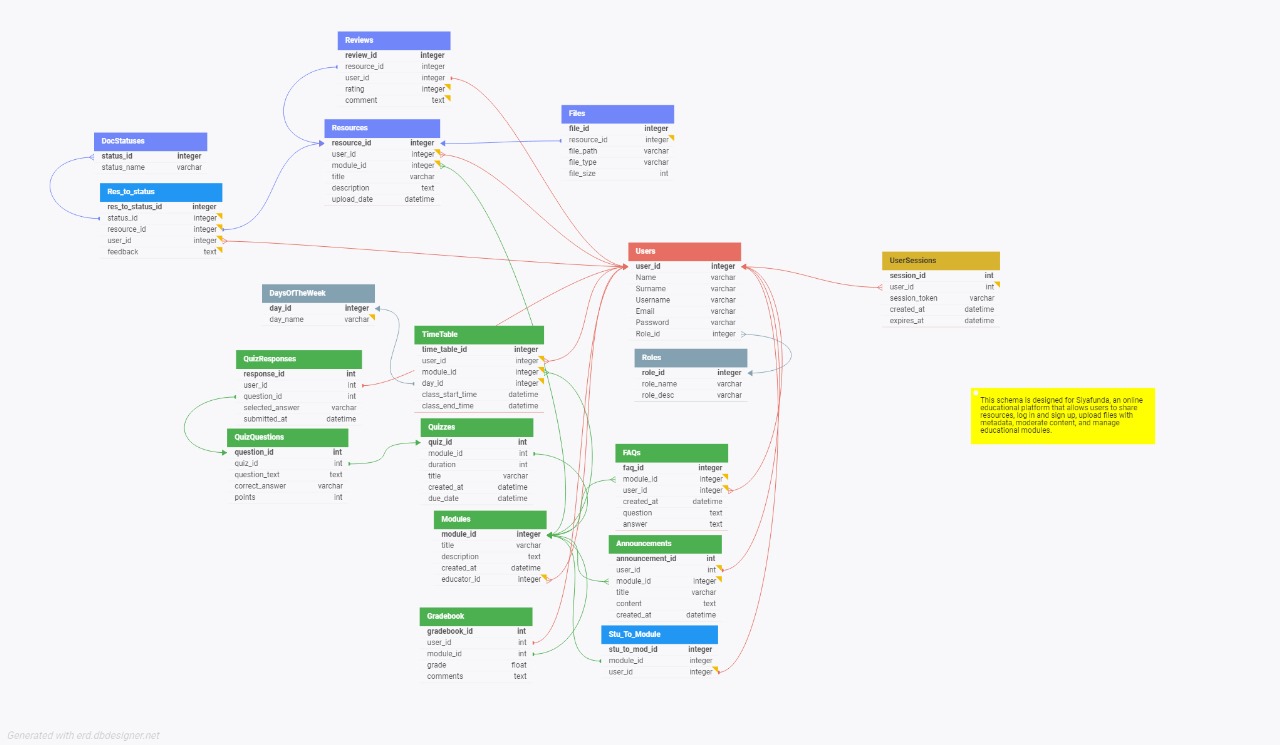
* Account creation and secure sign-in
* File uploading and storage
* Document moderation and reporting
* Quiz creation and management (MCQ, fill-in-the-blank quizzes)
* Document tagging and search functionality
* Analytics for user engagement and file interaction

# 5. Security and Authentication

The system implements secure authentication by utilizing Azure’s built-in App Service security thus reducing the load on the security developers during development. The `Web.config` file outlines the security protocols used for authentication and session management. Each user role is granted access rights based on their privileges, ensuring that sensitive operations are limited to authorized users and their session is vulnerable to timeouts after 30 minutes of inactivity to reduce server load and increase security.

# 6. Data Management and Storage

The platform relies on an SQL database to manage user data, file metadata, and quiz results. Each uploaded document is tagged with metadata, allowing for efficient search and retrieval. The database schema includes tables for users, documents, quizzes, and analytics.

Our SQL based database is structures as follows:

# 7. Wireframes and UI Design

The [wireframes provided](https://github.com/JaundreSmit/Siyafunda/blob/main/Documentation/Wireframes.pdf) offer a clear outline of the user interface, including the sign-in pages, file management interface, and quiz pages. The visual design is kept simple to ensure ease of use across both desktop and mobile platforms.

# 8. Deployment and Build Process

The system is deployed using Azure cloud services with a GitHub workflow. Once the project is ready for deployment, the workflow file will automatically build an artefact after every subsequent update and deploy it to the website. Continuous integration is handled through GitHub, with regular updates pushed to the repository.

# 9. Future Enhancements

Potential future improvements, as outlined in the MoSCoW Analysis, include the further development of the system, UI integration with React, and advanced analytics features to monitor user behaviour in real-time.

# 10. MoSCoW Analysis

|  |  |  |
| --- | --- | --- |
| **MoSCoW Analysis** | | |
| **Categories** | **Item** | **Completion Progress** |
| **Must-Have** | Account Creation & Secure Sign-in | ✓ |
| File Uploading, Storage, and Downloading | ✓ |
| File Moderation | ✓ |
| Document Searching & Tagging | ✓ |
| Security | ✓ |
| Backend Data Storage | ✓ |
| **Should Have** | Pre-pending Watermark/License to Files | ✓ |
| Analytics | ✗ |
| Document Ratings by Users | ✗ |
| FAQ Page | ✓ |
| **Could Have** | Announcements Feature | ✓ |
| Timetables for Students | ✓ |
| Assignment Submissions | ✗ |
| **Won't Have** | Mobile Application | ✗ |
| Video Integration | ✗ |
| Advanced Personalization | ✗ |