

Contents

[1. Introduction 3](#_Toc180000374)

[2. System Architecture 3](#_Toc180000375)

[3. User Roles and Responsibilities 3](#_Toc180000376)

[4. Functional Requirements 4](#_Toc180000377)

[5. Security and Authentication 4](#_Toc180000378)

[6. Data Management and Storage 5](#_Toc180000379)

[7. Wireframes and UI Design 5](#_Toc180000380)

[8. Deployment and Build Process 5](#_Toc180000381)

[9. Future Enhancements 6](#_Toc180000382)

[10. MoSCoW Analysis 6](#_Toc180000383)

[Conclusion 7](#_Toc180000384)

# 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
* Bootstrap for UI and UX enhancements

# 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

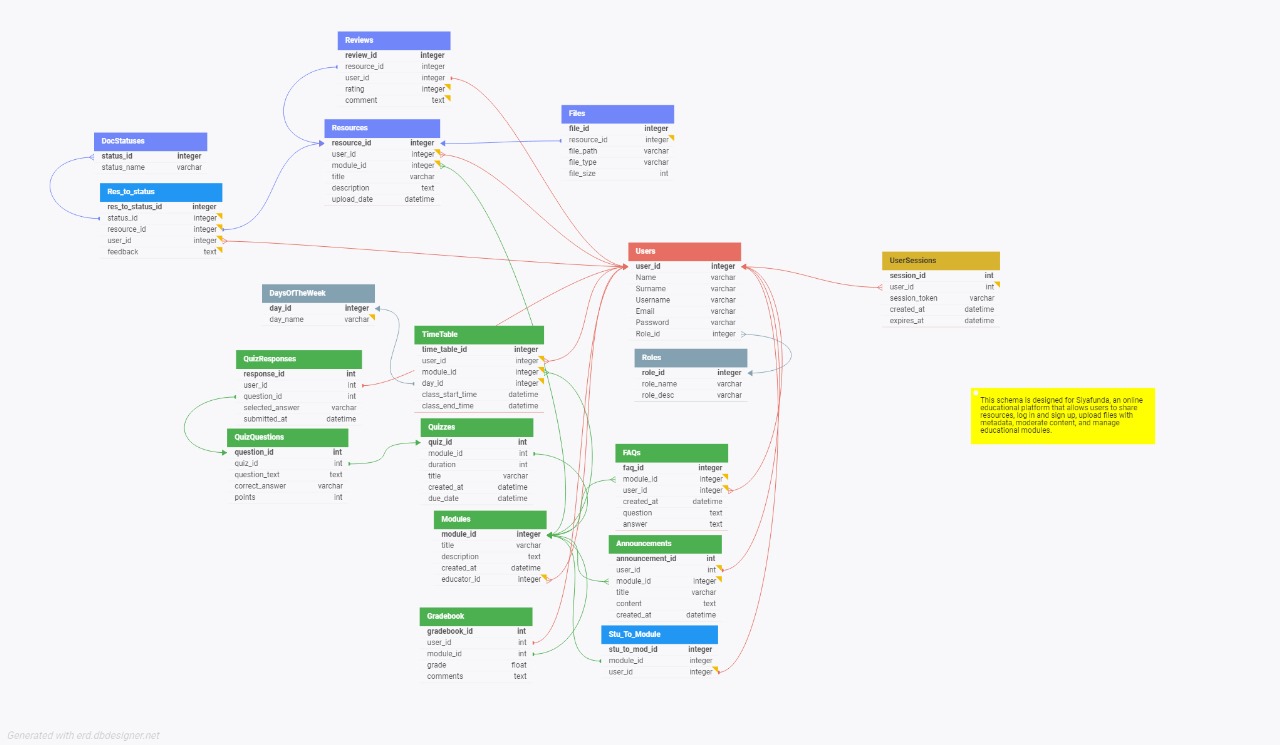
The Siyafunda platform provides a range of functionalities including:

* Account creation and secure sign-in
* File uploading and storage
* Document moderation and reporting
* Quiz creation and management (MCQ, fill-in-the-blank and long-form)
* 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. The validity of the user is constantly checked by validating their ID and role to ensure constant appropriate access. Further, the system is configured for HTTPS to ensure that the site is always securely accessed.

# 6. Data Management and Storage

The platform relies on a T-SQL database to manage user data, file metadata, and quiz results. Die database is critical to the successful CRUD operations of our APIs, as the functionality and security of the system are heavily dependent on this. Each uploaded document is tagged with metadata, allowing for efficient search and retrieval. Our T-SQL-based database is structured 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 to potentially include capabilities for Assignments, enhancements of the grade book functionality

# 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 | ✓ |
| Fully Traversable UI | ✓ |
| **Should Have** | Pre-pending Watermark/License to Files | ✓ |
| Analytics | ✓ |
| Announcements Page | ✓ |
| FAQ Page | ✓ |
| **Could Have** | Announcements Feature | ✓ |
| Timetables for Students | ✓ |
| Video Implementation | ✗ |
| **Won't Have** | Mobile Application | ✗ |
| Assignment Submissions | ✗ |
| Advanced Personalization | ✗ |
| Document Ratings by Users | ✗ |

# Conclusion

With the Siyafunda mobile application being ready for deployment, the system now enters the post-launch phase of development. Our first priority now is to collect user feedback, utilize user analytics and combine this information to determine what the next steps are for continuous improvement and support. We aim to also focus on fixing bugs that were missed during development. After these steps, we plan to continue developing new features for the system such as assignment capabilities and video player implementation.