

YOUR LOGO

SHARE2TEACH

FUNCTIONAL & TECHNICAL SPECIFICATIONS

John Klerck

04 July 2024

Version 0.5

VERSION HISTORY				
VERSION	APPROVED BY	REVISION DATE	DESCRIPTION OF CHANGE	AUTHOR
0.1	Johan Venter	21/05/2024	Functional Draft	John Klerck
0.2	John Klerck	29/05/2024	Technical Draft	Johan Venter
0.3		11/06/2024		
0.4		02/07/2024		

Functional & Technical Specifications Document

Authorization Memorandum

I have carefully assessed the Functional & Technical Specifications Document for Share2Teach.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

_____ The document is accepted.

_____ The document is accepted pending the changes noted.

_____ The document is not accepted.

We fully accept the changes as needed improvements and authorize the initiation of work to proceed. Based on our authority and judgement, the continued operation of this system is authorized.

Name: _____

Project Manager

Date

Name: _____

Project Sponsor

Date

Name: _____

Project Sponsor

Date

Preface.....	5
Purpose.....	5
Intended Audience and Pertinent Sections.....	5
Project Scope.....	5
Document Conventions & Terminology.....	5
Introduction.....	6
Overview.....	6
Operating Environment.....	6
User Roles.....	6
Overview of Functional Requirements.....	7
Overview of Data Requirements.....	7
Overview of Product and Technical Requirements.....	8
Application.....	8
Data Storage.....	8
Security.....	8
Backend.....	8
Architectural Requirements.....	8
Constraints.....	8
Assumptions.....	9
Dependencies.....	9
Guidelines.....	9
User Documentation.....	9
Requirements.....	10
External Interface Requirements.....	10
User Interfaces.....	10
Hardware Interfaces.....	11
Software and Communications Interfaces.....	11
Functional Requirements.....	12
Non-Functional Requirements.....	14
Performance.....	14
Safety.....	14
Security.....	14
Software Quality.....	15
MoSCoW Breakdown	Bookmark missing
Product and Technical Requirements.....	16
Application.....	16
Data Storage.....	16
Security.....	16
Backend.....	17
Architectural Requirements.....	17
Constraints.....	17
Assumptions.....	17
Dependencies.....	17

Guidelines.....	18
User Documentation.....	18
Appendices.....	21
Appendix A: Glossary of Terms.....	22

Preface

Purpose

This Technical and Functional Specifications document serves to inform all project stakeholders of the agreed upon project scope, functionality expectations, technical expectations and responsibilities.

Intended Audience and Pertinent Sections

Project Sponsor - Preface, Introduction

Project Manager - Preface, Introduction, Requirements

Development Lead - Preface, Introduction, Requirements

Project Scope

This project aims to extend upon the Share2Teach platform by implementing the following features:

1. Account Creation and Secure Sign-in
 - a) User verification for Educator Role required
2. File Uploading & Storage
3. File Moderation (Gate keeping, reviewing, approving/denying of documents)
4. File Reporting
5. Pre-pending Watermark/License to files
6. Allow tagging of Documents when uploading
7. Allow Searching of Documents
8. Implement Analytics to monitor user engagement and behaviour
9. Facilitate document ratings by Users.
10. Create an FAQ page

Document Conventions & Terminology

Introduction

Overview

Share2Teach is introduced as a vibrant open educational resource (OER) project crafted to nurture a global community of learners and educators. At its core lies the principle that knowledge should be accessible, collaborative, and freely available. Share2Teach is a testament to the power of collective endeavour, co-crafted by students under the guidance of their facilitators.

The project was initiated by Dr. Chantelle Bosch, a dedicated lecturer and sub-area leader for Blended Learning to Enhance Self-Directed Learning within the Research Unit Self-Directed Learning at the North-West University (NWU). Alongside her, Prof. Dorothy Laubscher, the chair-holder of the UNESCO Chair on Multi-modal Learning and OER, has played a pivotal role in shaping the vision and trajectory of Share2Teach.

Together, a platform has been cultivated where diverse educational resources are brought to life, crafted by students for students. From comprehensive semester planning documents to topic-specific insights, Share2Teach offers a wide array of materials tailored to enhance self-directed learning through cooperative learning and project-based teaching strategies.

Share2Teach serves as a beacon for educational innovation, extending an invitation to educators and learners worldwide to contribute, explore, and evolve within this open, inclusive community. Joining this journey means participating in the endeavour to transform learning into a shared adventure, dismantling barriers and erecting bridges toward a more knowledgeable and interconnected world.

Operating Environment

The system will be web-based, and only allow access through a web browser using the internet. This system will make use of technologies supported by most of the latest desktop and desktop-like browsers and should allow for a seamless user experience. Mobile compatibility might be supported, but the system will not be optimized for mobile usage.

User Roles

Four user groups have been identified. They are presented as follows:

1. Admin
 - a) This user role represents the project owner and/or sponsor and the developers maintaining the system. This user has unrestricted access to all components of the system. The only unique component that this user has access to is the analytics component. Access to this user role is tightly controlled.
2. Moderator
 - a) This user role represents a subset of subject experts as selected by the project owner and/or sponsor. This user has access to the following features:
 - i. Document Searching
 - ii. Document Viewing
 - iii. Document Contribution

- iv. Document Rating
 - v. Using the FAQ
 - vi. Moderating Documents
- 3. Educator
 - a) This user role represents any user who registers for an educator's account. This user has access to the following functionality:
 - i. Document Searching
 - ii. Document Viewing
 - iii. Document Contribution
 - iv. Document Rating
 - v. Using the FAQ
- 4. Open Access User
 - a) This user role represents any user that accesses the site and does not sign in. This user has access to the following functionality:
 - i. Document Searching
 - ii. Document Viewing
 - iii. Document Rating
 - iv. Using the FAQ

Overview of Functional Requirements

The following Functional Requirements have been stipulated by the client:

- Account Creation and Secure Sign-in - (User verification for Educator Role required)
- File Uploading & Storage
- File Moderation (Gate keeping, reviewing, approving/denying of documents)
- File Reporting
- Pre-pending Watermark/License to files
- Tagging of Documents when uploading (Metadata)
- Searching of Documents
- Analytics to monitor user engagement and behaviour
- Document Ratings
- FAQ page

Overview of Data Requirements

The following Data Requirements have been stipulated by the client:

- File Storage
- File Metadata
- File Ownership
- Moderation History
- User Data - Account Creation
- User Analytics

Overview of Product and Technical Requirements

This section is meant to outline the technical requirements for the development of a web application designed to provide free access to educational resources for students from various backgrounds. Roles including educators, moderators, open access users and admins will manage and interact with the content, based on the assigned permissions. It details the technologies, architecture, and implementation strategies to be used by the development team.

Application

This project will deliver an application that users can interact with in the form of a web app. This application will extend an existing Google pages application that has a familiar look and feel. This application will form the main way in which users will interface with the system.

Data Storage

The system needs to store different types of data:

Document Storage: Uploaded documents will be kept in a file storage system.

Document Metadata: Information like subject, grade, and keywords, along with document ratings and storage locations, will be stored in an SQL database.

User Data: User login details, roles, and activity logs for analytics and audits will be securely stored.

FAQ: Frequently asked questions and answers will be managed in a database.

Security

1. **Verification and Validation:** This system will make use of HTTP-only cookies, JWTs and encrypted passwords for making sure that users are only able to access and modify the content that they are authorized to, and to ensure that user accounts are secured.

Backend

1. **API:** This system will provide a server listening for requests from the application. Request behaviour will be moderated using an API that will abstract away the complex interactions from the application with the data.

Architectural Requirements

1. **Hosting:** This application will have to be hosted on a web hosting platform that could either be an in-house bare metal server or a 3rd party service such as AWS, Microsoft Azure, Google cloud.

Constraints

Compatibility: The application must be a web app, accessible via modern browsers.

Open-source 3rd party libraries: All technologies and libraries used for functional applications (such as architectural, framework, and library technologies) with the exception of the hosting costs must be fully free, open-source.

Scalability: (Infrastructure Architecture) Docker Swarm

Modularity, Extensibility, Expandability: The constituents of the code architecture must be modular and loosely coupled in support of high loads and future expansion.

Security: The system must adhere to common security standards for secure data storage and

transmission.

Budget: ~~The budget allocated will determine the hosting platform for the project.~~

Time: The current iteration of the project has one semester allocated for completion, which will conclude on the 30th of October 2024.

Version Control: The software will use git and GitHub for version control and collaboration. Each member needs to prove a similar level of contribution to the project.

Assumptions

- The client will be available for regular demonstrations and feedback.
- The requirements outlined in this document dictate the scope of the project that will not undergo significant change.
- The existing web platform will be extended, but not replaced.
- There will exist at least one stable branch that will always be ready for the production environment.

Dependencies

3rd party libraries: This project will make use of a vast array of 3rd party software for each part of the system.

Hosting platform: The project will require servers for hosting the application, the data stores, and the API gateway. The addition of load balancers may become required.

Stakeholder Input: Regular feedback on stakeholder requirements will direct the project throughout its lifecycle.

Guidelines

Coding Standards: Code must follow language-specific conventions.

Commenting and Documentation: Concise, descriptive comments are expected in the code. All functionalities must be documented thoroughly.

Version Control: GitHub will be used to host the project source code. One branch will be named something like 'stable' that can be easily identified and will always be production ready.

Unit Testing: Write unit tests for non-trivial source code.

User Documentation

User Manuals: The user will be provided with a detailed guide on how to use the application that covers all the previously discussed features.

FAQ: The main site will contain an FAQ section that will provide users with access to frequently asked questions and their suggested answers.

Video Demonstration: The stakeholders and users will be provided with a video that demonstrates how to use the system and its features.

Developer Documentation: The Stakeholders will be provided with comprehensive developer documentation to aid future developers in understanding the system and the design decisions that were made. (Hons Only)

Requirements

External Interface Requirements

User Interfaces

Describe product / user interface characteristics, including standards, style guides, constraints, functionality, and sample screens if applicable.

The following table contains a preliminary list of the identified User Interfaces required in the system. This list is non-exhaustive, and subject to change. The "flow" through the system as described in the "Links to" column is considered to be the optimal flow through the system as currently described. As Developers develop the system, more interfaces may be identified, which could lead to changes in the flow through interfaces. It is also important to note that the "flow" is only 'positive' and does not account for when a user may encounter errors or issues with the system ('negative' flow). It is up to the developers to account for this 'negative' flow.

No	User Interface Name	Currently Exists (Y/N/P)	Description	Links to
1	Landing	Y	The main page the user sees. Presents the user with hyperlinks to the majority of the screens listed below.	All
2	Subject View	P	This page is subject specific. It allows further searching/viewing of documents within a subject.	3,4,13,
3	Search (Results)	P	This page allows for searching of documents based on tags created when the documents are uploaded. It provides optional filters along with the search results.	4,13
4	Contribute	N	This page allows a user to contribute documents and to tag them while doing so.	13,
5	Other useful OER's	Y	This page directs the user to other OER resource pages.	All
6	Contributors	Y	This page displays all the contributors to the project. This page needs to be updated to be dynamic.	All
7	About Us	Y	This page provides information about the project, the project website, and the project founders.	
8	Self-Directed Learning	Y	This page provides links to Self-Directed Learning resources available to users.	
9	Moderate	N	This page will present a user with all the files uploaded by users, that need to be moderated.	
	Account			

10	Creation	N	This page allows users to create accounts.	
11	Password Reset	N	This page allows users to reset their passwords should they forget them.	
12	Analytics	N	This page allows administrators to view the analytics information gathered on the website.	
13	FAQ	N	This page provides users with answers to some frequently asked questions.	
14	User Management	N	This page allows an administrator to change the access levels of users, as well as view all the users on the system	

Hardware Interfaces

The system requires the following Hardware interfaces.

Type	Description	Interactions
Keyboard-like HID	A HID used to input keystroke data into the website's text fields and interact with buttons.	
Mouse-like HID	A HID used to select buttons, fields and files on the website.	Selecting (Focusing) on buttons and fields.

Software and Communications Interfaces

The system requires the following types of Software interfaces. This list is non-specific, non-exhaustive, and subject to change as the development process proceeds.

Type	Description	Interactions
Database	Relational or non-relational databases used to store documents, user data, transaction logging etc.	Read/Write, backup and restore
File Storage System	The primary file storage for any documents uploaded to the system	Read/Write, backup and restore
API	RESTful API: Primary access to services and features of the system will be accessed through this API gateway.	The frontend application will make requests and receive responses in JSON. OAuth2 is suggested for security
Hosting Operating System	Ubuntu LTS release: The underlying operating system for the application and related services.	The Operating System will be responsible for providing a runtime for the entire application and servers.
Framework	A web application framework for building the frontend user interface	UI rendering and user input for enabling communication with the API gateway
Tool	Docker: A platform for developing, building and shipping and running software in containers.	Builds in CI/CD pipeline, Auto-scaling for load balancing

Authentication	Users will need to be authenticated for access to the system	Users will provide their login credentials in the frontend application for authentication.
Version Control	The git version control platform will be used on GitHub to facilitate version control and developer collaboration.	Branching/Push and pull requests will be made via a shell or on GitHub's website.

Functional Requirements

This list contains the functional requirements for the system to be developed during the projects execution. This list is non-exhaustive and should be expanded upon as the developers identify a need.

Purpose/Description	Account Creation
Inputs	FName, LName, Email, Password, Affiliation (opt), Credentials (Opt)
Processing	Required Field check, Email-type check, password match check
Outputs	Failure: Required Fields empty, invalid email type, passwords don't match.
	Success: Account creation successful

Purpose/Description	Sign In
Inputs	Email, Password
Processing	Required Field Check, email type check, user exist check, password match check
Outputs	Failure: Email or password does not match, required fields empty
	Success: sign in successful.

Purpose/Description	File Uploading & Tagging
Inputs	User File, File type, file name, subject, grade, date created,
Processing	Required field check, file size check, file type check,
Outputs	Failure: File too large, incorrect file type, required fields empty, file upload failed,
	Success: file upload successful.

Purpose/Description	File Storage
Inputs	
Processing	

Outputs	Failure:
	Success:

Purpose/Description	File Moderation
Inputs	File approval/disapproval, comments
Processing	updating file approval status
Outputs	Failure: Could not complete action
	Success: Action Complete

Purpose/Description	File Rating
Inputs	0-5 star rating for document
Processing	Rating entry made in database
Outputs	Rating visible to user

Purpose/Description	File Reporting
Inputs	Report button selected, reason selected
Processing	Report entry made in database
Outputs	"Report submitted" message

Purpose/Description	Watermark/License adding
Inputs	File uploaded
Processing	Watermark/License is Prepended
Outputs	File with watermark/license is saved in file storage

Purpose/Description	Document Search
Inputs	Keywords entered by user
Processing	Database tags searched for keywords.
Outputs	Files presented to user.

Purpose/Description	User Analytics
Inputs	User actions on system
Processing	Data relating to actions written to database.
Outputs	none

Purpose/Description	FAQ
Inputs	none
Processing	none
Outputs	List of FAQ's with answers provided

Purpose/Description	Password Reset
Inputs	Password Reset request, email address, verification cookie/session/token, new password
Processing	Email verification, token verification, password match verification
Outputs	Failure: Failure message.
	Success: Email Verification Message to generate token, success Message

Non-Functional Requirements

Performance

Scalability: The system should be able to scale horizontally to handle a growing number of users and increased data load without performance degradation. This includes the ability to add more instances of the web server and database as needed.

Throughput: The system should be able to support multiple concurrent users without a significant drop in performance.

Caching: caching mechanisms should be implemented to reduce server load and improve response times.

Safety

Data Integrity: Ensure all **requests** are processed accurately and data is consistently maintained across the system. Implement database constraints and transactional integrity checks.

Fault Tolerance: The system should be able to recover from hardware or software failures. This includes automatic failover mechanisms and redundancy in critical components.

Error Handling: Implement comprehensive error handling throughout the application. All errors should be logged, and appropriate user-friendly messages should be displayed to the users.

Backups: Regular backups of all critical data should be performed and stored securely. Ensure that a disaster recovery plan is in place and tested periodically.

Security

Authentication and Authorization: All users must be authenticated using a secure authentication mechanism (e.g., OAuth2, JWT). Implement role-based access control to ensure users only have access to the resources they are authorized to use.

Data Encryption: All sensitive data, both at rest and in transit, must be encrypted using industry-standard encryption protocols).

Security Audits: Security audits and penetration testing can be used to identify and mitigate vulnerabilities. Follow best practices and comply with relevant security standards and regulations.

Software Quality

Code Quality: Ensure high code quality by following coding standards and best practices. Conduct regular code reviews.

Testing: Implement a comprehensive testing strategy that includes unit tests, integration tests, system tests, and user acceptance tests. Aim for high test coverage to ensure the reliability of the codebase.

Continuous Integration/Continuous Deployment (CI/CD): Utilize CI/CD pipelines to automate the build, testing, and deployment processes. This helps in detecting and fixing issues early and ensures that changes are delivered quickly and reliably.

Documentation: Maintain up-to-date and comprehensive documentation for the entire system, including code comments, API documentation, user manuals, and operation guides.

Usability: Design the application with a user-centric approach, ensuring that it is intuitive and easy to use. Conduct usability testing to gather feedback and make necessary improvements.

Maintainability: Ensure the system is easy to maintain by modularizing the code, following design patterns, and keeping dependencies up to date. Document the architecture and design decisions to aid future maintenance efforts.

Product and Technical Requirements

This section is meant to outline the technical requirements for the development of a web application designed to provide free access to educational resources for students from various backgrounds. Roles including educators, moderators, open access users and admins will manage and interact with the content, based on the assigned permissions. It details the technologies, architecture, and implementation strategies to be used by the development team.

The system will provide a web application that the primary users will interact with for downloading, uploading and approving documents. These documents can be tagged with descriptions and other relevant information that will be used to index and identify the documents in a central data store.

The system will implement a user access and upload policy that will allow only authenticated users to approve document upload requests. In this way, document quality can be ensured. This will require users to sign up to the system and apply for moderator privileges. This user data will be stored in the data store and any sensitive user information will be encrypted.

For all data processing and preparation, a backend server will be hosted that will be responsible for all data processing and aggregation. This backend server will provide access to data in the form of an API gateway.

The entire system will need to be hosted on a platform that will give all the users access to the system. Some options are available for in-house hosting as well as cloud providers.

Application

This project will deliver an application that users can interact with in the form of a web app. This application will extend an existing Google pages application that has a familiar look and feel. This application will form the main way in which users will interface with the system.

Data Storage

The system will require persistent storage for the following:

1. **Document Storage:** All uploaded documents will need to be stored in a file storage system.
2. **Document Metadata:** When a user is in the process of uploading a document, they will add some metadata about the document such as the subject group, grade, key words etc. This data, the document ratings and a reference to the location of the actual document in the file storage system will need an entry in a database. The database paradigm suggested for this is SQL, since we are dealing with highly structured data, where data aggregation, sorting and grouping will be very important when users search and filter for documents.
3. **User Data:** All user-related content, such as their login information and the associated roles need to be stored for user verification and identification. For analytics and auditing purposes, all user actions will be recorded in a database with timestamps and actions.
4. **FAQ:** Frequently asked questions and answers to these will be stored in a database.

Security

2. **Verification and Validation:** This system will make use of HTTP-only cookies, JWTs and encrypted passwords for making sure that users are only able to access and modify the content that they are authorized to, and to ensure that user accounts are

secured.

Backend

2. **API Gateway:** This system will provide a server listening for requests from the application. Request behaviour will be moderated using an API gateway that will abstract away the complex interactions from the application with the data.

Architectural Requirements

2. **Hosting:** This application will have to be hosted on a web hosting platform that could either be an in-house bare metal server or a 3rd party service such as AWS, Microsoft Azure, Google cloud.

Constraints

Compatibility: The application must be a web app, accessible via modern browsers.

Open-source 3rd party libraries: All technologies and libraries used for functional applications (such as architectural, framework, and library technologies) with the exception of the hosting costs must be fully free, open-source.

Scalability: The constituents of the architecture must be modular and loosely coupled in support of high loads and future expansion.

Security: The system must adhere to common security standards for secure data storage and transmission.

Budget: The budget allocated will determine the hosting platform for the project.

Time: The current iteration of the project has one semester allocated for completion, which will conclude on the 30th of October 2024.

Version Control: The software will use git and GitHub for version control and collaboration.

Assumptions

- The client will be available for regular demonstrations and feedback.
- The requirements outlined in this document dictate the scope of the project that will not undergo significant change.
- The existing web platform will be extended, but not replaced.
- There will exist at least one stable branch that will always be ready for the production environment.

Dependencies

3rd party libraries: This project will make use of a vast array of 3rd party software for each part of the system.

Hosting platform: The project will require servers for hosting the application, the data stores and the API gateway. The addition of load balancers may become required.

Stakeholder Input: Regular feedback on the stakeholder requirements will direct the project throughout its lifecycle.

Guidelines

Coding Standards: Code must follow language-specific conventions.

Commenting and Documentation: Concise, descriptive comments are expected in the code. All functionalities must be documented thoroughly.

Version Control: GitHub will be used to host the project source code. One branch will be named something like 'stable' that can be easily identified and will always be production ready.

Unit Testing: Write unit tests for non-trivial source code.

User Documentation

User Manuals: The user will be provided with a detailed guide on how to use the application that covers all the previously discussed features.

FAQ: The main site will contain an FAQ section that will provide users with access to frequently asked questions and their suggested answers.

Video Demonstration: The stakeholders and users will be provided with a video that demonstrates how to use the system and its features.

MoSCoW Breakdown

Categories	Item	Category Weight
Must Have	User Verification File Uploading File Storage Document Reporting File Metadata capturing Document Searching Document ratings FAQ Page	40%
Should Have	File Moderation Automatic Metadata generation pdf conversion Pre pending Licence	35%
Could Have	Docker Google Analytics integration	25%
Won't Have	Firebase/Superbase Backend	(<100%)

Suggested Technologies

Front End

Back End

Database

Data Store

Appendices

Appendix Number	Document Name	Description	Location
A	Glossary of Terms	A Glossary containing all unique terms and acronyms used within this document that pertains to this project.	
B	Issue List	A list of outstanding issues within this document, to be rectified before the next revision.	
C	Analysis Models	A document containing various diagrams generated during the systems analysis and design phases	

Appendix A: Glossary of Terms

Term	Definition
HID	Human Interface Device. A method by which a human interacts with an electronic information system either by inputting data or receiving output.

C1 "Preface" in Page 5

Johan Venter on Wednesday, May 29, 2024, 07:16 AM

Ek verstaan die eerste section hier as n disclosure vir die "readers" of dit nou die kliente of stakeholders is waar ons hulle bietjie meer konteks oor die doc gee. Soos konvensies en sections of interest vir die relevante audience so ek het dit verander na Preface, as jy n beter title het gooi