Hamdard University
Department of Computing
Final Year Project

StudyMate – AI Powered Advanced Quiz Creator (FYP-037/FL24)

Software Requirements Specifications

Submitted by
S. M Abubakar Zaidi (1600-2021)
Hanzala Siddiqe (2577-2021)

Supervisor(s)
Asst. Prof. Mr. Osama Ahmed Khan
Senior Lecturer Mr. Waqas Pasha

Spring 2025

Document Sign off Sheet

Document Information

Project Title	StudyMate – AI Powered Advanced Quiz Creator
Project Code	FYP-037/FL24
Document Name	Software Requirements Specifications
Document Version	3.1
Document Identifier	FYP-037/FL24-SRS
Document Status	Approved for Revision
Author(s)	Abubakar Hanzala
Approver(s)	Mr. Osama Ahmed Khan
Issue Date	14/08/2024

Name	Role	Signature	Date
S. M Abubakar Zaidi	Team Lead		
Hanzala Siddiqe	Team Member 2		
Mr. Osama Ahmed Khan	Supervisor		
Mr. Waqas Pasha	Co-Supervisor		
Mr. Prem Sagar	Project Coordinator		

Revision History

Date	Version	Description	Author
15/08/2024	1.0	Prepared draft of SRS	Hanzala
20/12/2024	1.1	Made changes related to Req.	Abubakar
25/12/2024	2.0	Add new features	Hanzala
09/01/2025	2.1	Corrected grammatical errors	Abubakar
16/01/2025	3.0	Add references	Hanzala
29/06/2025	3.1	Made changes related to updated work	Abubakar

Definition of Terms, Acronyms, and Abbreviations

Term	Description
Students and Teachers	Primary target users of the application
Version Updates	Future enhancements and additions planned for the application
OCR (Optical Character Recognition)	A technology used to extract text from images or scanned documents
Llama	A large language model optimized for text generation
Django	A high-level Python web framework used for building the backend of the application
User Authentication	A feature that enables secure login and registration of users
Spiral Methodology	A software development methodology emphasizing iterative development
Educational Technology (EdTech)	The application of technology in education to enhance teaching, learning
Prototype	The initial working version of the application
User Interface (UI)	The visual component of the application

Table of Contents

Contents

Document Information 2

Table of Contents 5

1. Introduction 7

1.1 Purpose of Document 7

1.2 Intended Audience 7

2. Overall System Description 8

2.1 Project Charter 8

2.2 Project Background 11

2.3 Problem Statement 11

2.4 Project Scope 11

2.5 Not In Scope 11

2.6 Project Objectives 11

2.7 Project Plan 11

2.8 Stakeholders & Affected Groups 14

2.9 Operating Environment 15

2.10 System Constraints 15

2.11 Assumptions & Dependencies 15

3. External Interface Requirements 16

3.1 Hardware Interfaces 16

3.2 Software Interfaces 17

3.3 Communications Interfaces 17

4. Risk Analysis 18

4.1 Risk Identification 17

4.2 Risk Mitigation Strategy 20

5. System Functions / Functional Requirements 22

5.1 System Functions 22

5.2 Use Cases 23

5.2.1 List of Actors 23

5.2.2 List of Use Cases 23

5.2.3 Use Case Diagram 24

5.2.4 Description of Use Cases 25

6. Non - Functional Requirements 28

6.1 Performance Requirements 28

6.2 Safety Requirements 28

6.3 Security Requirements 28

6.4 Reliability Requirements 28

6.5 Usability Requirements 29

6.6 Supportability Requirements 29

6.7 User Documentation 29

7. References 30

1. Introduction

1.1 Purpose of Document

This Software Requirements Specification (SRS) document is aimed at defining the Functional, non-functional, and system requirements for the AI-Powered StudyMate project. This is a reference point for the development team, stakeholders, and end-users, hence. A very clear understanding of the system's goals, features, and limitations. It details the scope, assumptions, and constraints of the project and guides development and subsequent iterations.

1.2 Intended Audience

This project is intended for the following audience:

- **Project Team:** Developers, designers, and testers who build and maintain the system.
- **Stakeholders:** Sponsors, project champions, and university supervisors overseeing the progress of the project.
- **End-Users:** Students and instructors who will actually use the system to generate quizzes and Q&A from handwritten notes.
- **Researchers:** Those interested in finding improvements in OCR and AI-based educational resources.

1.3 Abbreviations:

Abbreviation	Description
SRS	Software Requirements Specification
OCR	Optical Character Recognition
LLM	Large Language Model
AI	Artificial Intelligence
UI	User Interface
Q&A	Question and Answer
Llama 3.1	Large Language Model Version 3.1 by Meta
Django	High-level Python web framework
EdTech	Educational Technology
JSON	JavaScript Object Notation

2. Overall System Description

2.1 Project Charter

GENERAL PROJECT INFORMATION	
PROJECT NAME	StudyMate – AI-Powered Advanced Quiz Creator
PROJECT SPONSOR	Abubakar Zaidi Hanzala Siddiqe
PROJECT MANAGER	Abubakar Zaidi
STAKEHOLDERS	Students, Educators, Supervising Faculty
EXPECTED START DATE	7/30/2024
EXPECTED COMPLETION DATE	6/30/2025

PROJECT DETAILS	
EXECUTIVE SUMMARY	StudyMate is a learning platform that scans handwritten notes into organized digital content. Advanced OCR technology, AI-based grammar refinement, and quiz generation algorithms make the process of producing unique study materials highly efficient. This platform is ideal for students and instructors who wish to prepare for exams and classes with fewer efforts and much accuracy.
AUTHORIZATION	This project is authorized by Hamdard University, with the approval of the project supervisor and key evaluators. It is aligned with the objectives of the institution in utilizing technology for educational development.

OBJECTIVES	• Automate the digitization and structuring of handwritten notes. • Provide an intuitive platform for personalized quiz and Q&A generation. • Ensure scalability and reliability for diverse educational needs.
EXPECTED BENEFITS	• For Students: Simplified study preparation and efficient revision tools. • For Educators: Rapid generation of customized learning materials. • For Institutions: Enhanced student outcome and learning process.
SCOPE	• Handwritten note digitization and error correction • Quiz and Q&A generation based on user inputs • User-friendly web-based interface with login and registration.
MILESTONES	• Initial Requirements Gathering - August 20, 2024 • Prototype Development December - 05, 2024 • Backend and Frontend Integration - January 10, 2025 • Final Testing and Deployment - April 15, 2025

SUCCESS METRICS	• 90% accuracy in OCR text extraction. • 85% user satisfaction rate during testing. • Generation of quizzes in less than 10 seconds per input.
ESTIMATED COST & RESOURCES	• Budget – Rs 30,000 • Human Resources - 2 developers • Tools - Google Vision OCR, AI APIs, Django framework • Materials - Cloud hosting, laptops, and supporting hardware
DATE	August 25, 2024

2.2 Project Background

The AI-Powered StudyMate project makes the entire process of preparing for exams more manageable by using Optical Character Recognition and Large Language Models. The application takes text out of handwritten notes and converts them into structured content, providing quizzes and Q&A for students and teachers to learn and teach effectively. This project addresses the difficulties of manual processing of notes for exams and preparing for them by including tools such as Google Vision OCR, Llama 3.1, and provides a web interface using Django.

2.3 Problem Statement

Students find it difficult to study voluminous hand-written notes effectively when examinations are near. Similarly, teachers take a lot of time preparing quizzes and exam papers manually. Inefficiency in these processes reduces productivity and impacts performance. There is a need for a tool that can automate such processes, which would enable proper time management and focus on core tasks.

2.4 Project Scope

The project will provide a web-based application that:

- Students and teachers can upload handwritten notes in different formats such as PDF, JPG, PNG.
- Extraction and text cleaning using Google Vision OCR and Llama 3.1.
- Generates quizzes and Q&A content based on user preferences using "meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo."
- Offers user-friendly templates for Home, Login, Register, and Quiz/Q&A generation pages.
- User authentication is provided with login/register functionalities along with proper storage of notes and generated materials in google cloud storage.

2.5 Not In Scope

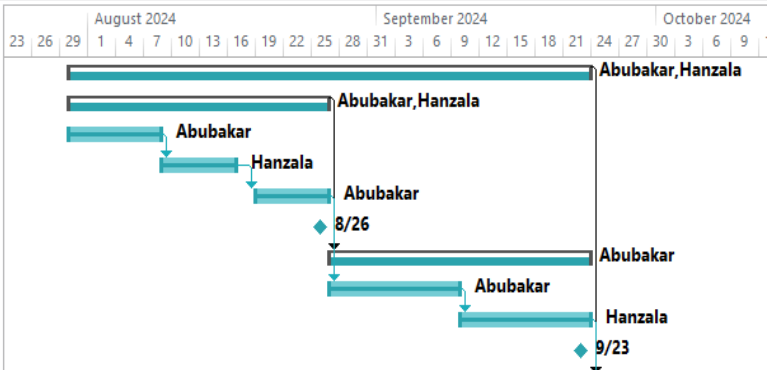
- Advanced models for handwriting recognition from scratch.
- Offline text extraction and content generation.
- Analytics and reporting for the content generated by users.
- Language support other than English in the first release.

2.6 Project Objectives

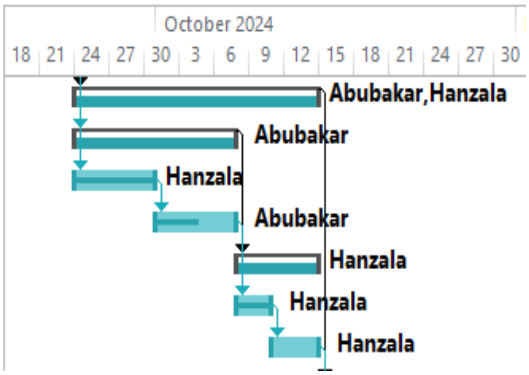
- Text extraction from handwritten notes to be automated.
- Improve text structure and correctness for usability.
- Provide tools for quiz and Q&A generation in an efficient manner.
- Create a user-friendly and scalable web-based platform. Ensure seamless integration of OCR, LLM, and web technologies.

2.7 Project Plan

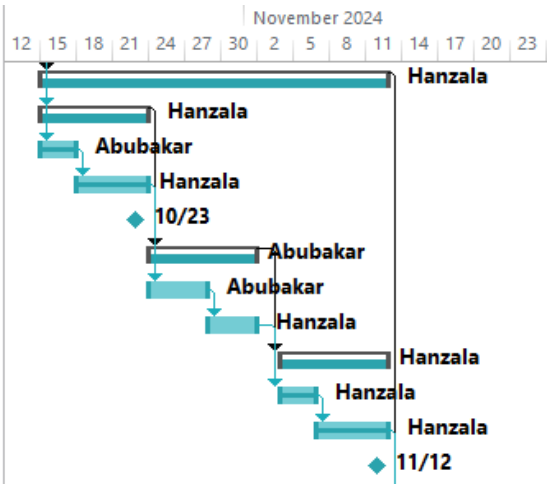
Task Mode	Task Name	Duration
	1.1 Project Management	**40 days**
✓	**1.1.1 Project Planning**	**20 days**
✓	1.1.1.1 Define Project Scope	8 days
✓	1.1.1.2 Develop Project Timeline	6 days
✓	1.1.1.3 Resource Allocation	6 days
✓	1.1.2 <New Milestone>	0 days
✓	**1.1.3 Risk Management**	**20 days**
✓	1.1.3.1 Identify Project Risks	10 days
✓	1.1.3.2 Develop Risk Mitigation Plan	10 days
✓	1.1.4 <New Milestone>	0 days



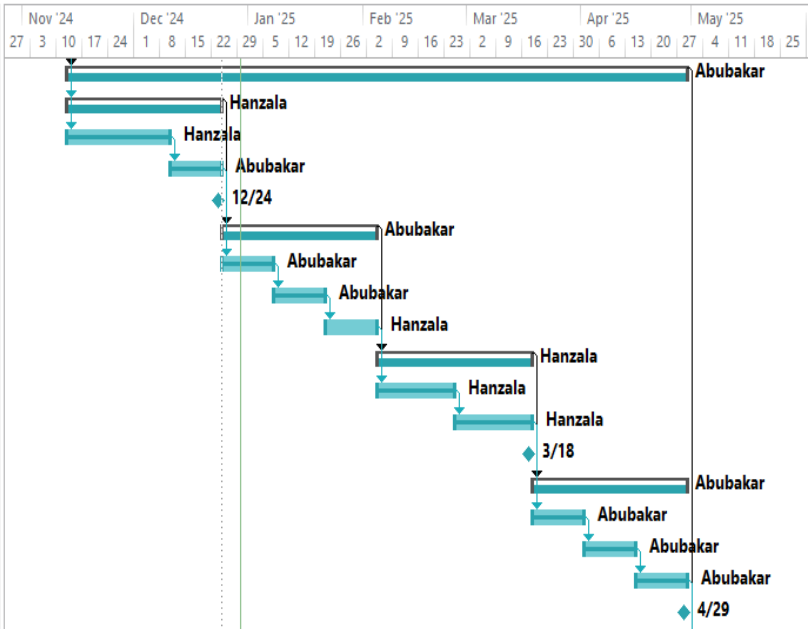
Task Mode	Task Name	Duration
	1.2 Requirement Analysis	**15 days**
	1.2.1 Requirement Gathering	**10 days**
✓	1.2.1.1 Student & Teacher Requirements	5 days
	1.2.1.2 Technical Requirements	5 days
	1.2.2 Requirement Documentation	**5 days**
✓	1.2.2.1 Create Requirement Specification Document	3 days
	1.2.2.2 Finalize Requirement Approval	2 days



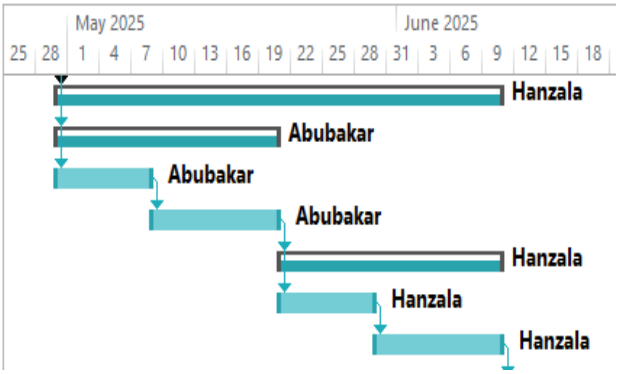
Task Mode	Task Name	Duration
	1.3 System Design	**21 days**
✓	**1.3.1 Architecture Design**	**7 days**
✓	1.3.1.1 Backend System Design	3 days
✓	1.3.1.2 Frontend System Design	4 days
✓	1.3.2 <New Milestone>	0 days
	1.3.3 Database Design	**7 days**
	1.3.3.1 Define Data Schema	3 days
	1.3.3.2 Design Entity Relationship Diagrams (ERD)	4 days
✓	**1.3.4 UI/UX Design**	**7 days**
✓	1.3.4.1 Wireframing	3 days
✓	1.3.4.2 Prototyping	4 days
✓	1.3.5 <New Milestone>	0 days



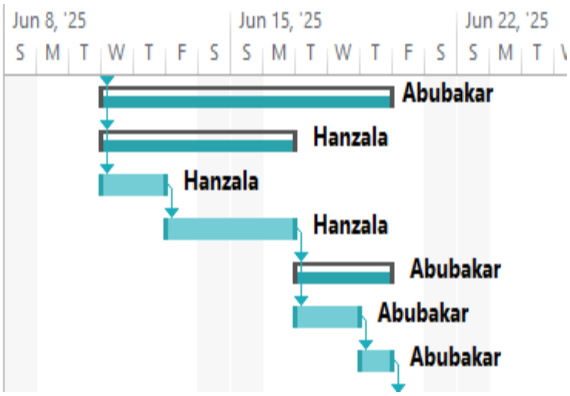
Task Mode	Task Name	Duration
	1.4 Development	**120 days**
✓	**1.4.1 Frontend Development**	**30 days**
✓	1.4.1.1 Develop User Interface	20 days
✓	1.4.1.2 Integrate Frontend with Backend APIs	10 days
✓	1.4.2 <New Milestone>	0 days
	1.4.3 Backend Development	**30 days**
✓	1.4.3.1 Develop APIs	10 days
✓	1.4.3.2 Implement Business Logic	10 days
	1.4.3.3 Set Up Database (PostgreSQL)	10 days
✓	**1.4.4 OCR Module Development**	**30 days**
✓	1.4.4.1 Research and Select OCR Tools	15 days
✓	1.4.4.2 Implement Text Extraction	15 days
✓	1.4.5 <New Milestone>	0 days
✓	**1.4.6 Quiz/QnA Module Development**	**30 days**
✓	1.4.6.1 Research NLP Models	10 days
✓	1.4.6.2 Implement Keyword Extraction	10 days
✓	1.4.6.3 Implement Question Generation Algorithm	10 days
✓	1.4.7 <New Milestone>	0 days

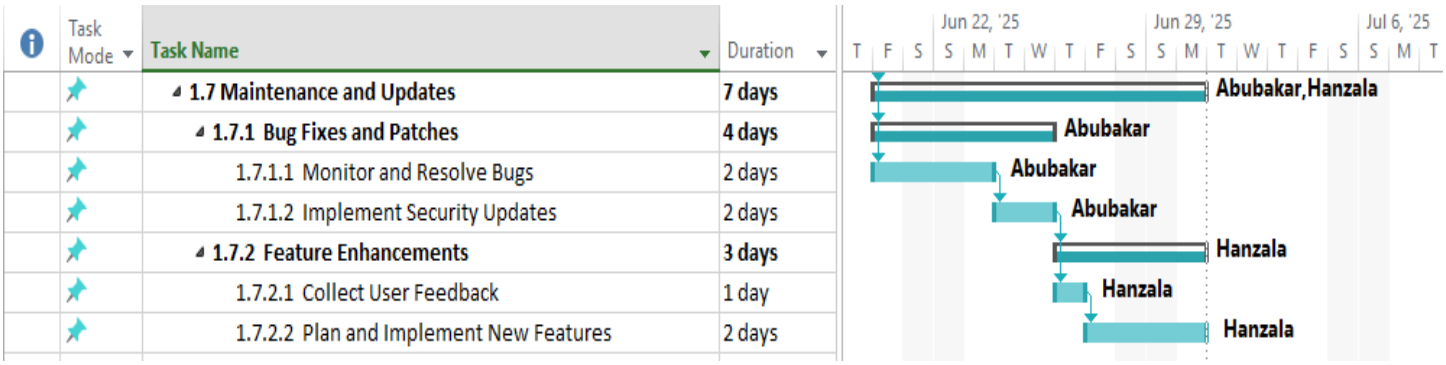


Task Mode	Task Name	Duration
	1.5 Testing and QA	**30 days**
	1.5.1 Integration Testing	**15 days**
	1.5.1.1 Test Frontend-Backend Integration	7 days
	1.5.1.2 Test OCR and NLP Modules	8 days
	1.5.2 User Acceptance Testing (UAT)	**15 days**
	1.5.2.1 Conduct Testing with Students & Teachers	7 days
	1.5.2.2 Collect Feedback and Make Revisions	8 days



Task Mode	Task Name	Duration
	1.6 Deployment	**7 days**
	1.6.1 Deployment Planning	**4 days**
	1.6.1.1 Choose Cloud Infrastructure (Google Cloud)	2 days
	1.6.1.2 Set Up Deployment Pipeline	2 days
	1.6.2 Deployment Execution	**3 days**
	1.6.2.1 Deploy Backend and Database	2 days
	1.6.2.2 Deploy Frontend and Integrate Services	1 day





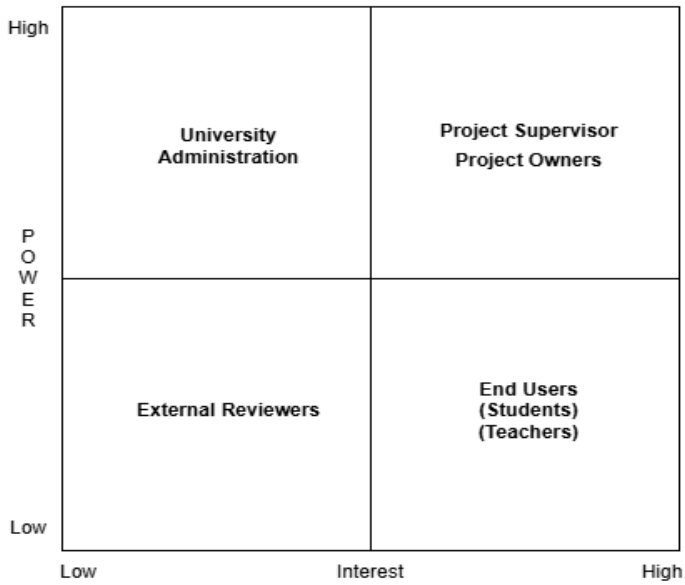
Task Mode	Task Name	Duration
	1.7 Maintenance and Updates	7 days
	1.7.1 Bug Fixes and Patches	4 days
	1.7.1.1 Monitor and Resolve Bugs	2 days
	1.7.1.2 Implement Security Updates	2 days
	1.7.2 Feature Enhancements	3 days
	1.7.2.1 Collect User Feedback	1 day
	1.7.2.2 Plan and Implement New Features	2 days

2.8 Stakeholders & Affected Groups

- **Stakeholder Register**

Stakeholder	Role	Interest	Influence
Students	End-User	Easy quiz and Q&A preparation	High
Teachers	End-User	Efficient exam preparation tools	Medium
Project Supervisor	Advisor	Project alignment and progress	Medium
Development Team	Implementation	Successful delivery of the project	High
University	Evaluator	Sets standards and guidelines	Low

- **Stakeholder Analysis Matrix**



- **Stakeholder Communication Plan**

Stakeholder	Communication Medium	Frequency	Engagement Strategy
Students	Email, User Surveys	Bi-Weekly	Regular feedback and usability testing
Teachers	Email, Presentations	Bi-Weekly	Regular updates on progress and demos
Project Supervisor	Meetings, Reports	Weekly	Weekly updates and progress reports
Development Team	Daily Standups	Daily	Collaborative sprints and daily standups
University	Reports, Presentations	Bi-Annually	Bi-Annually reports and presentations

2.9 Operating Environment

The application will operate as a web-based solution, accessible through modern web browsers (e.g., Chrome, Firefox, Safari) on desktops, laptops, and mobile devices. It will use a Django backend integrated with templates for frontend functionality. The application will rely on cloud services for OCR (Google Vision API) and AI processing (Llama 3.1).

2.10 System Constraints

- Limited to processing English handwritten notes in the initial release.
- Requires an active internet connection for OCR and LLM operations.
- Dependent on third-party APIs (Google Vision, Meta Llama).
- Constrained by the allocated project timeline and resources.

2.11 Assumptions & Dependencies

- Users will upload clear and legible handwritten notes.
- Third-party APIs (Google Vision, Llama 3.1) will remain functional and accessible.
- The application will be hosted on a reliable server with adequate resources.
- The methodology used (Spiral) will adapt to project needs dynamically.

3. External Interface Requirements

3.1 Hardware Interfaces

The hardware requirements for StudyMate are minimal, as it primarily operates as a web application. Key hardware interfaces include:

- **Client Devices:**

Devices such as laptops, desktops, tablets, and smartphones that will access the StudyMate web application via a browser.

Minimum Requirements:

- 2 GB RAM
- Dual-core processor
- Screen resolution of at least 1024x768 pixels

- **Server Infrastructure *(Future Feature)*:**

Servers hosting the Django backend and database for StudyMate.

Recommended Configuration:

- 16 GB RAM
- Quad-core processor
- SSD storage for fast data retrieval
- Reliable internet connectivity with a minimum bandwidth of 100 Mbps

3.2 Software Interfaces

StudyMate interacts with several software components and libraries to function seamlessly:

- Frontend:
  - Browser compatibility with Chrome, Firefox, Safari, and Edge (latest versions).
  - HTML, CSS, and JavaScript frameworks (integrated into Django templates).
- Backend:
  - Django Framework: Handles application logic, user authentication, and data processing.
  - Google Vision OCR API: Used for extracting text from uploaded handwritten notes.
  - LLaMA 3.1 Model API: Processes and structures extracted text, and generates quizzes and Q&A.
- Database:
  - SQLite (prototype stage) and PostgreSQL (for deployment) for storing user-generated content and application data.

3.3 Communications Interfaces

StudyMate requires robust communication interfaces to ensure efficient data exchange:

- User Interaction:
  - HTTP/HTTPS protocols for secure data transfer between the client and the server.
  - RESTful APIs for handling backend communication.
- External API Integration:
  - Google Vision API: Secure HTTPS requests for OCR processing.
  - Meta-Llama 3.1 API: JSON-based requests and responses for quiz and Q&A generation.
- Email Notifications:
  - Integration with an SMTP service to allow visiting users to contact developers for any collaboration or inquiry.

4. Risk Analysis

4.1 Risk Identification

Methods Used for Identifying Risks:

The identification of potential risks in the StudyMate project was carried out using the following methods:

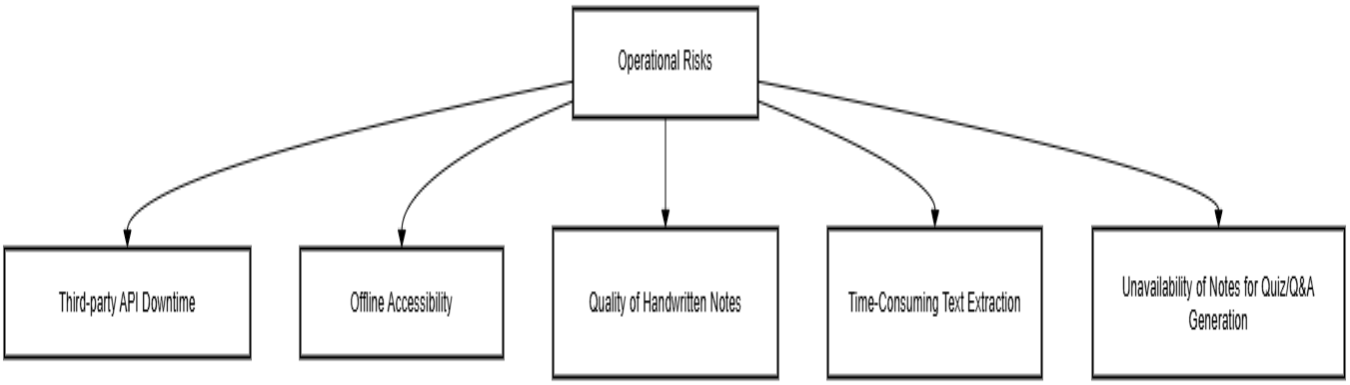
- **Brainstorming Sessions:** Involves team discussions to identify possible risks that could hinder the project.
- **Expert Interviews:** Feedback was obtained from domain experts and experienced developers to anticipate challenges.
- **Review of Similar Projects:** Examining existing projects using similar technologies helped identify common risks faced during development.
- **SWOT Analysis:** A systematic review of Strengths, Weaknesses, Opportunities, and Threats (SWOT) was used to identify project vulnerabilities.

List of Identified Risks:

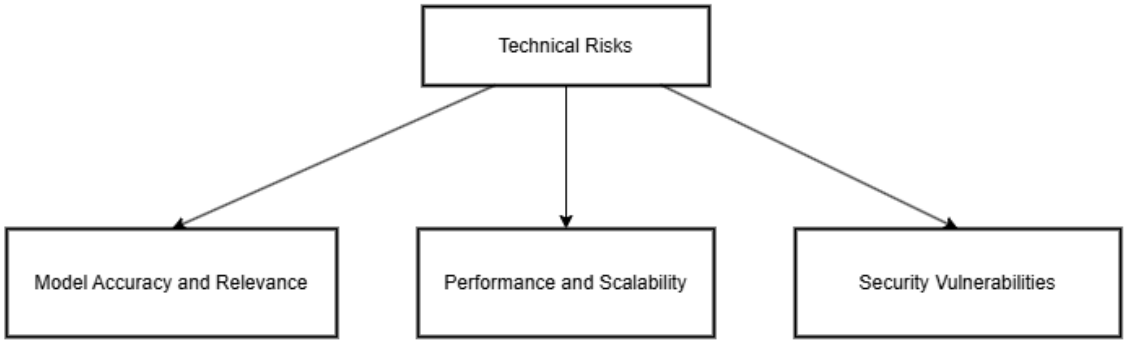
- The application is dependent on third-party APIs, such as Google Vision OCR and LLaMA 3.1 Model. If these APIs are down, the entire application might be impacted.
- The application requires an active internet connection because it uses external APIs, which makes it less usable when offline.
- Handwritten notes' quality will affect the text extraction accuracy. If the handwriting quality is low, then there will be a problem in the extraction of text.
- If there are no notes, the system cannot produce quizzes or Q&As.
- The LLaMA 3.1 Model can produce irrelevant and inaccurate quizzes and Q&As, leading to user dissatisfaction.
- It may suffer from performance issues and not scale with the number of users.
- The generated quizzes and answers containing sensitive information of the user could be exposed to unauthorized access.

Risk Breakdown Structure (RBS):

Operational Risks



Technical Risks



4.2 Risk Mitigation Strategy

To ensure the successful execution of the StudyMate project, a robust risk mitigation strategy has been devised. This strategy focuses on proactive identification, monitoring, and resolution of potential risks. Below are the primary steps in the strategy:

- **Risk Assessment:** Continuously monitor and assess risks during the project lifecycle to determine their probability and impact.
- **Mitigation Planning:** Develop contingency plans and fallback mechanisms for high-priority risks.
- **Communication:** Maintain transparent communication with stakeholders about identified risks and mitigation measures.
- **Responsibility Assignment:** Assign a responsible individual or team to monitor and resolve specific risks.
- **Periodic Review:** Conduct regular risk reviews to ensure new risks are identified and existing ones are managed effectively.

Risk Register

Risk ID	Risk Description	Impact	Probability	Priority	Mitigation Plan	Responsible Team
R1	Third-party API downtime	High	Medium	Critical	Implement fallback mechanisms; monitor third-party API status regularly.	Backend Development Team
R2	Offline accessibility	High	Medium	High	Investigate offline functionality options; use local caching for essential features.	Backend Development Team
R3	Poor quality of handwritten notes	Medium	High	High	Inform users to upload clear images; use pre-processing techniques to enhance image quality.	OCR Development Team
R4	Time-consuming text extraction for large volumes	Medium	Medium	Medium	Implement batch processing and queue systems to optimize processing time.	OCR Development Team

R5	Unavailability of notes for quiz/Q&A generation	High	Medium	High	Include input validation checks to ensure users provide notes before proceeding.	Frontend Development Team
R6	Model accuracy and relevance	High	Low	High	Fine-tune the LLaMA 3.1 model using feedback; enable manual review of generated content.	AI Development Team
R7	Performance and scalability issues	High	Medium	Critical	Optimize backend processes; consider cloud-based infrastructure to handle growing user base.	DevOps Team
R8	Security vulnerabilities	High	Low	Critical	Implement encryption for data storage and transmission; ensure secure user authentication methods.	Security Team

5. System Functions / Functional Requirements

5.1 System Functions

The system functions of StudyMate are defined below, grouped by their logical functionalities, with their attributes and boundary constraints described.

Ref #	**Functions**	**Category**	**Attribute**	**Details & Boundary Constraints**
R1.1	Extract text from uploaded handwritten notes	Evident	System Accuracy	OCR accuracy ≥ 95% for clear handwriting; completion time within 15 seconds for a single page
R1.2	Process extracted text for grammar corrections	Hidden	Processing Speed	Response time ≤ 5 seconds for refining text
R1.3	Generate quizzes based on processed text	Evident	AI Model Performance	Generation time ≤ 15 seconds; model accuracy ≥ 90% for structured quizzes
R1.4	Generate Q&A content based on user preferences	Evident	AI Model Performance	Same as R1.3
R1.5	Authenticate users during login/register	Evident	Security	Password hashing compliant with industry standards
R1.6	Save generated quizzes and Q&As to database	Hidden	Database Write Performance	Transaction completion within 2 seconds
R1.7	Provide user-friendly web interface	Evident	User Experience	Intuitive navigation; page load time ≤ 3 seconds
R1.8	Allow users to manage their content (CRUD ops)	Evident	Usability	CRUD actions performed with a response time ≤ 2 seconds

System Attributes / Nonfunctional Requirements

Attribute	**Details & Boundary Constraints**	**Category**
Response Time	Max response time of 5 seconds for quiz/Q&A generation.	Mandatory
Concurrent User Load	Minimum support for 20 users simultaneously.	Mandatory
Interface Metaphor	Web-based, responsive, intuitive browser interface.	Mandatory
System Security	Data encrypted at rest and in transit; compliant with GDPR/CCPA.	Mandatory
Scalability	System scales horizontally to support up to 100 users.	Optional

5.2 Use Cases

5.2.1 List of Actors

- **User**

  The primary actor who interacts with the system to upload notes, generate quizzes/Q&A, and manage their content.

- **Administrator**

  Responsible for managing system maintenance, handling data backups, and monitoring system performance.

5.2.2 List of Use Cases

Use Case ID	Use Case Name	Description
UC1	Upload Handwritten Notes	User uploads handwritten notes in image format for text extraction.
UC2	Extract Text	The system uses Google Vision OCR to extract text from uploaded notes.
UC3	Refine Extracted Text	LLaMA 3.1 processes the text for grammar corrections, spelling fixes, and better structuring.
UC4	Generate Quiz	The system generates a quiz based on the processed text.
UC5	Generate Q&A	The system creates Q&A content tailored to user preferences.
UC6	Register New User	A new user creates an account with a username, email, and password.
UC7	Login Existing User	An existing user logs in using their credentials.
UC8	Save Generated Content	User saves generated quizzes or Q&A content to their account.
UC9	Manage Saved Content	User retrieves, edits, or deletes their saved content.
UC10	Monitor System Performance	Administrator monitors system logs, performance metrics, and handles maintenance.
UC11	Secure User Data	The system encrypts user data and ensures secure communication channels.
UC12	Generate Error Logs	System generates error logs for debugging and troubleshooting.

5.2.3 Use Case Diagram

5.2.4 Description of Use Cases

Section: Main	
Name:	*Quiz and Q&A Generation*
Actors:	*User*
Purpose:	*Enable users to generate quizzes or Q&A content based on uploaded handwritten notes.*
Description:	*A user uploads handwritten notes for text extraction and quiz/Q&A generation. The system extracts text, refines it using LLaMA 3.1 for grammar and structure improvement, and generates quiz or Q&A content as per user preference.*
Cross References:	*Functions: F1.1, F1.2, F1.3* *Use Cases: The user must be logged into the system.*
Pre-Conditions	*The user must have uploaded clear and readable handwritten notes.*
Successful Post-Conditions	*The system generates and displays a quiz or Q&A based on the processed notes.*
Failure Post-Conditions	*The system displays an error if the notes are unreadable or the text extraction fails.*

Typical Course of Events	
Actor Action	**System Response**
1. The user logs in and selects the "Generate Quiz/Q&A" option.	*2. Displays the upload page for handwritten notes.*
3. The user uploads a handwritten note.	*4. Processes the note using Google Vision OCR for text extraction.*
5. The user chooses between "Generate Quiz" or "Generate Q&A."	*6. Refines text using LLaMA 3.1 and creates the selected output.*
7. The user downloads or reviews the generated content.	*8. Saves the content in the user's session or displays it on the page.*

Alternative Course	
Step 3:	*User uploads an unreadable or unsupported file. Displays an error and prompts reupload.*
Step 7:	*User does not select an option. Prompts the user to choose a generation mode.*

Section: Upload Notes	
Name:	*Upload Handwritten Notes*
Actors:	*User*
Purpose:	*Allow users to upload handwritten notes for processing.*
Description:	*Users can upload clear handwritten notes to initiate text extraction and processing for content generation.*
Cross References:	*Functions: F1.1, F1.2* *Use Cases: The user must be logged in and authenticated.*
Pre-Conditions	*The user has clear handwritten notes in a supported format.*
Successful Post-Conditions	*The system successfully extracts text from the notes.*
Failure Post-Conditions	*Displays an error if the file is unsupported or unreadable.*

Typical Course of Events

Actor Action	System Response
1. User logs in and navigates to the upload page.	*2. Displays the file upload interface.*
3. User selects a handwritten note for upload.	*4. Validates the file format and readability.*
5. System confirms a successful upload.	*6. Initiates text extraction using Google Vision OCR.*

Alternative Course	
Step 3:	*User uploads an unsupported file format. Displays an error and prompts re upload.*

Section: User Authentication	
Name:	*Login/Register*
Actors:	*User*
Purpose:	*Authenticate users to access system functionalities.*
Description:	*Users log in with credentials or register for a new account to access StudyMate features.*
Cross References:	*Functions: F1.4* *Use Cases: The user must register for the first-time login.*
Pre-Conditions	*The user must provide valid credentials or required registration details.*
Successful Post-Conditions	*The user is granted access to the application.*
Failure Post-Conditions	*The system denies access and prompts error messages.*

Typical Course of Events	
Actor Action	**System Response**
1. User navigates to the login/register page.	*2. Displays login and registration options.*
3. User provides credentials or registration details.	*4. Validates the input and processes the request.*
5. System grants or denies access based on validation.	*6. Redirects authenticated users to the home page.*

Alternative Course	
Step 3:	*User enters incorrect credentials. Prompts error message and allows retry.*
Step 5:	*Registration fails due to missing information. Prompts user to complete the form.*

6. Non - Functional Requirements

6.1 Performance Requirements

- The system should answer any quiz/Q&A within 10 seconds for a maximum number of concurrent users being 10.
- Text extraction from handwritten notes to be done within 60 seconds for a normal 5-page note.
- It should be able to handle 100 concurrent users with almost no performance degradation.
- API response for LLaMA 3.1 operations in normal load cannot exceed 5 seconds.

6.2 Safety Requirements

- The system should encrypt all data uploaded by users when transmitted and also store it securely.
- The Google Vision OCR should be resistant to data corruption and inform users of upload failures.
- Unauthorized access that may cause possible data breach or improper text extraction should be prevented.
- In the event of an application crash, no unsaved data should be lost and should be recoverable when the system restarts.

6.3 Security Requirements

- Hash and store the user credentials securely according to the OWASP security standards.
- All data exchanged between the client and server MUST be encrypted using SSL/TLS.
- Users must have role-based access to functions, for example, Admin versus Regular User.
- The system must enforce strong password policies during registration.
- Must maintain and track logs of all the critical operations that include login attempts and file uploads.

6.4 Reliability Requirements

- The system needs to be available with 99.5% uptime.
- The application should automatically retry failed connections to third-party APIs up to three times before it notifies the user.
- The failure in quiz or Q&A generation should not affect the entire system; it should give an appropriate error message.
- All user-generated content should be persisted and recoverable in case of server failure or crash.

6.5 Usability Requirements

- The user interface must be simple, intuitive, and consistent across all devices.
- Key operations, such as uploading notes and generating content, must require no more than three clicks.
- The system must provide appropriate tooltips, user guides, and error messages for novice users.
- The application must support responsive design to ensure usability on desktops, tablets, and smartphones.

6.6 Supportability Requirements

- The application codebase must be modular to allow easy updates or feature additions.
- Integration with third-party APIs like Google Vision and LLaMA 3.1 must have configurable endpoints to facilitate future updates.
- The system should maintain detailed logs for debugging and support purposes, accessible by administrators.
- Documentation for developers must be provided, including API usage, system architecture, and deployment instructions.

6.7 User Documentation

The following user documentation will be provided:

- **User Guide:** Step-by-step instructions for account creation, uploading handwritten notes, and generating quizzes or Q&A content.
- **FAQs:** Answers to common questions such as file format requirements, troubleshooting upload issues, and understanding output content.
- **Tutorial Videos:** Demonstrating the core functionalities of the application.
- **Contact Information:** Channels for reaching technical support in case of unresolved issues.

7. References

Google Vision OCR Documentation

- **URL:** https://cloud.google.com/vision/docs
- **Description:** Official documentation for Google Vision OCR, used for extracting text from handwritten notes.

LLaMA 3.1 Documentation

- **URL:** https://www.llama.com/docs/get-started/
- **Description:** Meta's official documentation for the LLaMA 3.1 model, employed for text improvement and content generation.

Django Framework Documentation

- **URL:** https://docs.djangoproject.com
- **Description:** Official Django framework documentation, used for backend development and template integration.

OWASP Security Standards

- **URL:** https://owasp.org
- **Description:** Guidelines for secure application development, followed for implementing encryption and secure user authentication.

Spiral Development Methodology Overview

- **URL:** https://www.geeksforgeeks.org/software-engineering-spiral-model/
- **Description:** Comprehensive overview of the Spiral model, considered for iterative and risk-focused project development.

Web Accessibility Guidelines

- **URL:** https://www.w3.org/WAI/standards-guidelines/
- **Description:** W3C’s web accessibility standards, followed for ensuring usability across devices and user groups.

GitHub Repository for StudyMate

- **Repository:** https://github.com/abzaidi/studyMate
- **Description:** Source code repository for tracking development progress and version control of the StudyMate project.

IEEE Standards for Software Development

- **URL:** https://standards.ieee.org/
- **Description:** Guidelines for ensuring software reliability, safety, and performance in the StudyMate application.