



# **SOFTWARE ENGINEERING CONCEPTS SE LAB MANUAL**

## **TEAM 4**

### **TEAM MEMBERS**

YOGESH CV 220701327

THARUNASRI 220701303

VARSHA T 220701309

VENKAT A 220701315

VIKNESH J 220701321

SARWASHRI CR 220701511

# Software Concepts & Engineering - Lab Manual

## TEAM MEMBERS

YOGESH CV 220701327

THARUNASRI 220701303

VARSHA T 220701309 VENKAT A 220701315

VIKNESH J 220701321

SARWASHRI CR 220701511

Overview of the Project .....	3
Business Architecture Diagram .....	7
Requirements as user story .....	16
Architecture Diagram .....	20
Test Strategy .....	29

# Software Concepts & Engineering - Lab Manual

## Overview of the Project

The Question Paper Generator project aims to streamline and automate the creation of educational exams by leveraging modern web technologies and AI. The system allows educators to upload various content formats, including PDFs, Word documents, and images, along with the syllabus and question bank. It then utilizes ChatGPT to generate comprehensive, balanced question papers aligned with the specified syllabus and pattern. This automation reduces the time and effort required for manual question paper creation, ensures consistency and quality in exams, and provides a user-friendly interface for seamless file uploads and document generation. Ultimately, the project enhances efficiency and accuracy in educational assessments, benefiting both educators and students.

## What is the problem it is trying to resolve

**Manual Question Paper Creation:** Creating question papers manually is time-consuming and error-prone. Educators must sift through large volumes of content, syllabus guidelines, and question banks to assemble a balanced and comprehensive question paper.

**Inconsistent Question Papers:** Ensuring that each question paper aligns with the syllabus and difficulty level, and covers the necessary topics evenly, can be challenging without a standardized approach.

# Software Concepts & Engineering - Lab Manual

**Difficulty in Managing Multiple Formats:** Educators often receive or create question-related content in various formats, including PDFs, Word documents, and images. Integrating these different formats into a cohesive question paper can be cumbersome.

## With respect to data

### **Data Inputs:**

#### **Question Paper Pattern:**

Defines the structure of the question paper, such as sections, types of questions (multiple choice, short answer, etc.), and distribution of marks.

#### **Question Bank:**

A repository of questions categorized by topics, difficulty levels, and types.

#### **Syllabus:**

An outline of the topics and subtopics that need to be covered in the question paper.

**Content:** Additional resources or materials that can be used to generate or reference questions.

Data Processing:

### **Content Extraction:**

# Software Concepts & Engineering - Lab Manual

The system needs to parse and extract relevant data from different formats (PDFs, Word documents, images) to utilize them effectively.

**Pattern Matching:** Matching questions from the question bank to the defined question paper pattern.

## **Syllabus Alignment:**

Ensuring that the generated questions align with the syllabus, covering all required topics and adhering to the specified difficulty levels.

## **Data Output:**

**Generated Question Paper:** A document (PDF or text format) that includes the assembled questions following the specified pattern, aligned with the syllabus and drawn from the question bank and additional content.

# Software Concepts & Engineering - Lab Manual

## How it would help users when we implement the system

### **Efficiency:**

**Time-Saving:** Automating the question paper generation process significantly reduces the time educators spend on creating exams.

### **Streamlined Workflow:**

Users can upload various formats directly from their devices or cloud storage, making it easy to consolidate all necessary information in one place.

### **Consistency and Quality:**

#### **Standardization:**

Ensures that each question paper adheres to a consistent format and covers the required syllabus comprehensively.

**Balanced Difficulty:** The system can balance the difficulty levels of the questions, ensuring a fair assessment for all students.

### **User-Friendly Interface:**

#### **Easy Navigation:**

The web interface allows users to easily navigate through different sections and upload the necessary files without hassle.

# Software Concepts & Engineering - Lab Manual

## **Integration with GPT:**

Utilizing ChatGPT for generating and refining questions adds a layer of intelligence to the system, making the question paper creation process more robust and flexible.

## **Accessibility:**

Device and Cloud Integration: Users can upload files from local devices or directly from cloud storage services like Google Drive, offering flexibility and convenience.

Downloadable Outputs: Generated question papers can be downloaded in PDF format, making it easy for educators to distribute or print them.

Error Reduction:

## **Automated Checks:**

The system can include automated checks to ensure questions are not repeated and the syllabus coverage is complete, reducing human errors.

## Business Architecture Diagram

Components in the Diagram:

### **1. \*Users (Stakeholders)\*:**



# Software Concepts & Engineering - Lab Manual

- **\*Educators\***: Primary users who create and manage question papers.
- **\*Administrators\***: Users who oversee the system and manage user permissions.

## 2. **\*Front-End Interface\***:

- **\*Login/Sign-Up Page\***: User authentication interface.
- **\*Upload Page\***: Interface for uploading syllabus, question paper pattern, content, and question bank.
- **\*Result Page\***: Displays the generated question paper and provides options for downloading or copying.

## 3. **\*Back-End Services\***:

- **\*Authentication Service\***: Manages user login and registration.
- **\*File Upload Service\***: Handles the uploading and storage of files.
- **\*Question Paper Generation Service\***: Uses AI (ChatGPT) to generate question papers based on uploaded content.
- **\*Data Storage\***: Stores user data, uploaded files, and generated question papers.

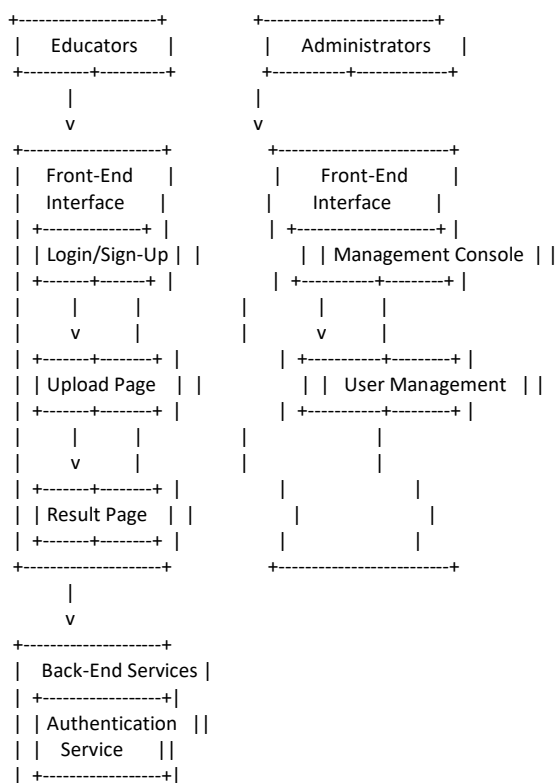
## 4. **\*External Services\***:

- **\*Cloud Storage Integration\***: Allows file uploads from cloud services like Google Drive.
- **\*AI Service\***: Integration with ChatGPT for generating question papers.

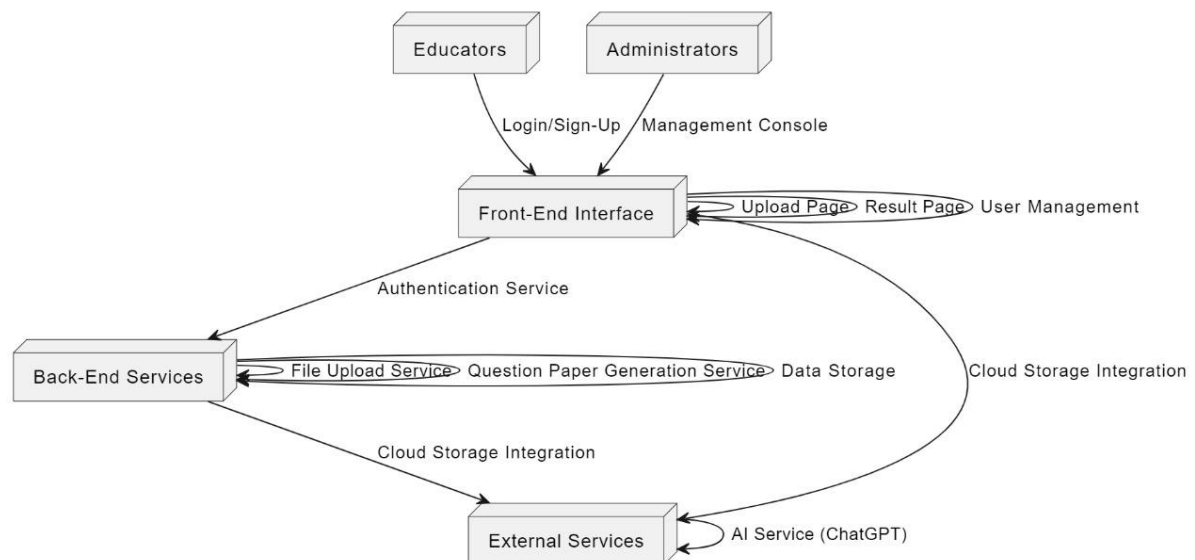
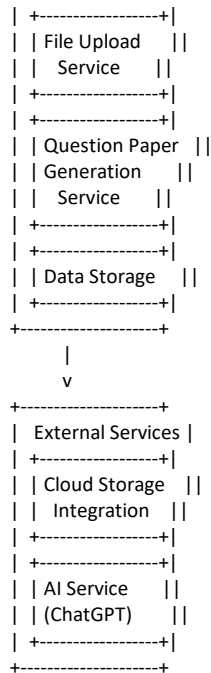
## 5. **\*Data Flows\***:

# Software Concepts & Engineering - Lab Manual

- \*User Input\*: Data provided by users (syllabus, patterns, content, question bank).
- \*Generated Question Paper\*: Output generated by the system for educators.



# Software Concepts & Engineering - Lab Manual



## Explanation

**Educators and Administrators:** The primary stakeholders who interact with the system through the front-end interface.

## Front-End Interface:

# Software Concepts & Engineering - Lab Manual

The user-facing part of the system, where educators log in, upload necessary files, and view the generated question papers.

## **Back-End Services:**

The server-side components that handle authentication, file uploads, and the question paper generation logic using AI.

## **External Services:**

Integrations with third-party services such as cloud storage (for file uploads) and ChatGPT (for AI-based question paper generation).

## **Data Flows:**

Represent the movement of data from user inputs to the generated question paper output.

## **Explain the Business Need of the project**

**The current process (either manual or automatic). How the current process is working**

How the Current Process is Working

# Software Concepts & Engineering - Lab Manual

## 1. **\*Syllabus and Pattern Analysis\*:**

- **\*Educators\*:** Manually review the syllabus and exam pattern to understand the requirements for the question paper.
- **\*Time-consuming\*:** This step involves a thorough reading of the syllabus documents and understanding the examination guidelines, which can be quite extensive.

## 2. **\*Question Selection\*:**

- **\*Question Bank\*:** Educators typically maintain a question bank with various questions categorized by topic, difficulty, and type.
- **\*Manual Selection\*:** They manually select appropriate questions from this bank that align with the syllabus and pattern.

## 3. **\*Question Paper Assembly\*:**

- **\*Formatting\*:** Educators manually format the selected questions into a coherent question paper, ensuring it meets the pattern requirements.
- **\*Consistency Checks\*:** They need to ensure that the paper is balanced in terms of difficulty and coverage of the syllabus.

## 4. **\*Review and Approval\*:**

- **\*Peer Review\*:** Often, the question paper is reviewed by peers or supervisors to ensure quality and adherence to standards.
- **\*Revisions\*:** Any feedback received during the review process requires manual revisions.

# Software Concepts & Engineering - Lab Manual

## 5. \*Distribution\*:

- \*Printing or Digital Distribution\*: The finalized question paper is either printed or distributed digitally to students.

## Explain the same for the different personas the system would be used

### Personas and Their Use of the System

#### Educators

##### \*Current Process\*:

- Manually review the syllabus, select questions, format the question paper, and ensure all requirements are met.
- Involved in multiple iterative revisions based on peer reviews and feedback.

##### \*With the System\*:

- Upload syllabus, pattern, content, and question bank files.
- Utilize the automated system to generate a question paper that aligns with the uploaded criteria.
- Review the generated paper and make minor adjustments if necessary.

#### Administrators

##### \*Current Process\*:

# Software Concepts & Engineering - Lab Manual

- Oversee the question paper creation process, ensure standards are met, and manage resource allocation for paper creation and review.

**\*With the System\*:**

- Monitor the automated system's performance, manage user permissions, and ensure smooth operation.
- Handle escalations and intervene in case of any system issues or required manual overrides.

## **Students**

**\*Current Process\*:**

- Indirectly affected by the quality and consistency of the manually created question papers.

**\*With the System\*:**

- Benefit from more consistent, balanced, and error-free question papers.
- Experience a more standardized assessment process.

## **Explain the business problems**

### **Business Problems**

1. **\*Inefficiency\*:**

- The current manual process is highly inefficient, consuming significant amounts of time and resources, which could be better spent on instructional activities and student engagement.

# Software Concepts & Engineering - Lab Manual

## 2. \*Inconsistency and Quality Control\*:

- Manual creation leads to inconsistent question papers, which can affect the fairness and quality of assessments.
- Lack of standardization in question selection and paper formatting can lead to variability in exam difficulty and coverage.

## 3. \*Human Error\*:

- High susceptibility to human error in question selection, formatting, and alignment with the syllabus.
- Errors in the question paper can lead to student confusion and affect exam outcomes.

## 4. \*Scalability\*:

- The manual process does not scale well with an increasing number of exams or growing question banks, making it difficult for institutions to manage larger volumes of assessments.

## 5. \*Resource Allocation\*:

- Significant human resources are required for creating and reviewing question papers, which can strain educational institutions, especially during peak exam seasons



## Requirements as user story

- A minimum of 10 user stories excluding the login user story in the same as told in the theory class

### User Stories

#### 1. \*Upload Syllabus\*

- As an educator, I want to upload the syllabus file so that the system can generate questions relevant to the syllabus.

- \*Estimate\*: 3

#### 2. \*Upload Question Paper Pattern\*

- As an educator, I want to upload the question paper pattern file so that the system can format the question paper accordingly.

- \*Estimate\*: 3

#### 3. \*Upload Content\*

- As an educator, I want to upload additional content files so that the system can use these as references for generating questions.

- \*Estimate\*: 3

#### 4. \*Upload Question Bank\*

- As an educator, I want to upload the question bank file so that the system can select questions from this repository.

- \*Estimate\*: 5

# Software Concepts & Engineering - Lab Manual

## 5. \*Generate Question Paper\*

- As an educator, I want the system to generate a question paper based on the uploaded syllabus, pattern, content, and question bank so that I can save time on manual creation.

- \*Estimate\*: 8

## 6. \*Review Generated Question Paper\*

- As an educator, I want to review the generated question paper before finalizing it to ensure it meets my requirements.

- \*Estimate\*: 5

## 7. \*Download Question Paper as PDF\*

- As an educator, I want to download the generated question paper as a PDF so that I can easily distribute it to students.

- \*Estimate\*: 3

## 8. \*Copy Question Paper Text\*

- As an educator, I want to copy the text of the generated question paper so that I can make any necessary edits or use it in different formats.

- \*Estimate\*: 2

## 9. \*Upload Files from Google Drive\*

- As an educator, I want to upload files from my Google Drive so that I can easily access my materials stored in the cloud.

- \*Estimate\*: 5

## 10. \*Notifications for Upload Status\*

# Software Concepts & Engineering - Lab Manual

- As an educator, I want to receive notifications about the status of my file uploads so that I know when the upload is complete and if there were any issues.

- Estimates using the Poker planning methodology for above

## Poker Planning Methodology

Poker planning involves assigning estimates based on relative sizes of user stories using Fibonacci sequence numbers (1, 2, 3, 5, 8, 13, etc.). Each participant privately selects an estimate card, and all cards are revealed simultaneously. The process iterates until consensus is reached.

## Estimates Summary

- \*Upload Syllabus\*: 3
- \*Upload Question Paper Pattern\*: 3
- \*Upload Content\*: 3
- \*Upload Question Bank\*: 5
- \*Generate Question Paper\*: 8
- \*Review Generated Question Paper\*: 5
- \*Download Question Paper as PDF\*: 3
- \*Copy Question Paper Text\*: 2
- \*Upload Files from Google Drive\*: 5
- \*Notifications for Upload Status\*: 3

# Software Concepts & Engineering - Lab Manual

The estimates represent the relative effort required to implement each user story. They help in planning and prioritizing work in sprints. The NFRs ensure that the system meets critical performance, security, and usability standards.

- A minimum of 3 NFR's

## Non-Functional Requirements (NFRs)

### 1. **\*Performance\***

- The system should generate a question paper within 30 seconds after all files are uploaded and the generate button is clicked.

### 2. **\*Security\***

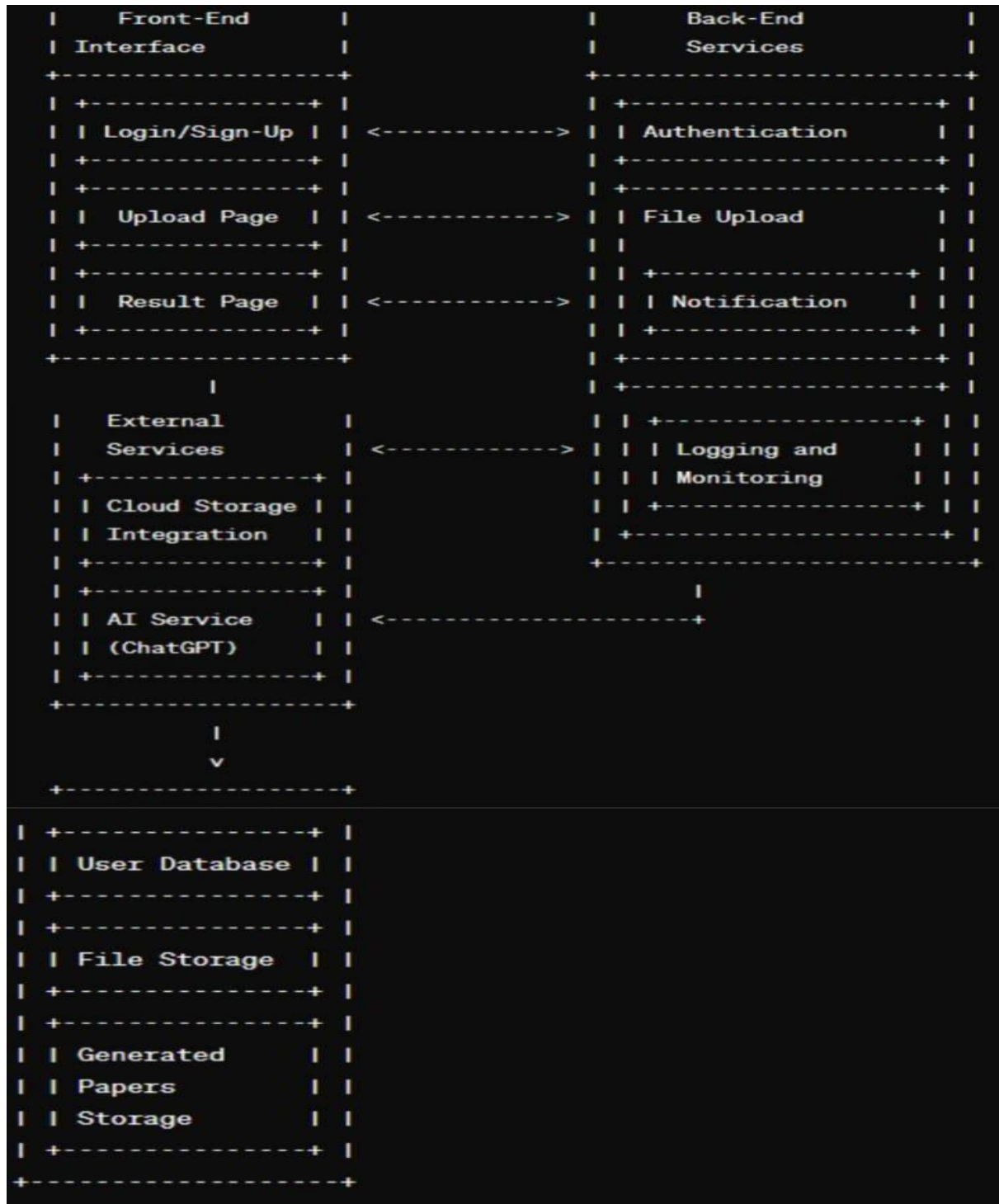
- The system must ensure that all uploaded files and generated question papers are securely stored and accessible only to authenticated users.

### 3. **\*Usability\***

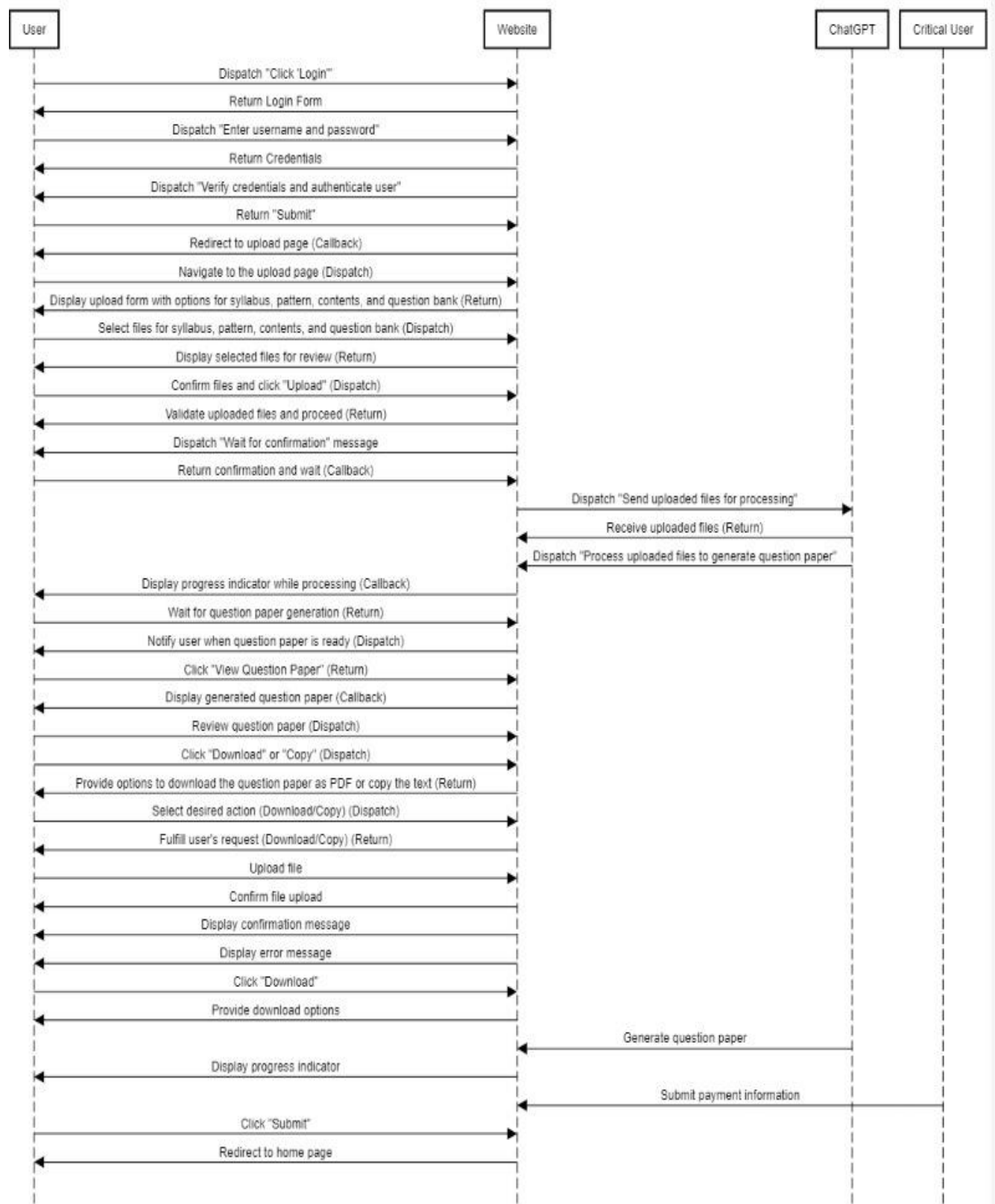
- The system should have an intuitive user interface that allows users to easily navigate and complete tasks without extensive training or support.

# Software Concepts & Engineering - Lab Manual

## Architecture Diagram

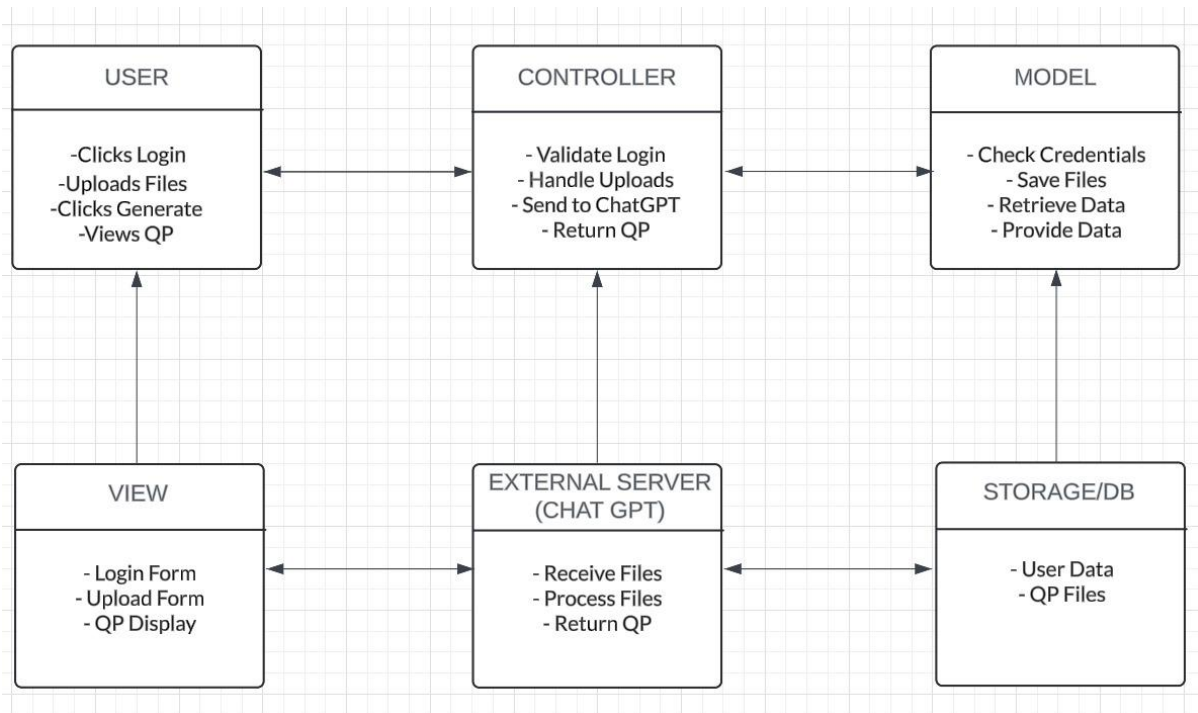


# Software Concepts & Engineering - Lab Manual



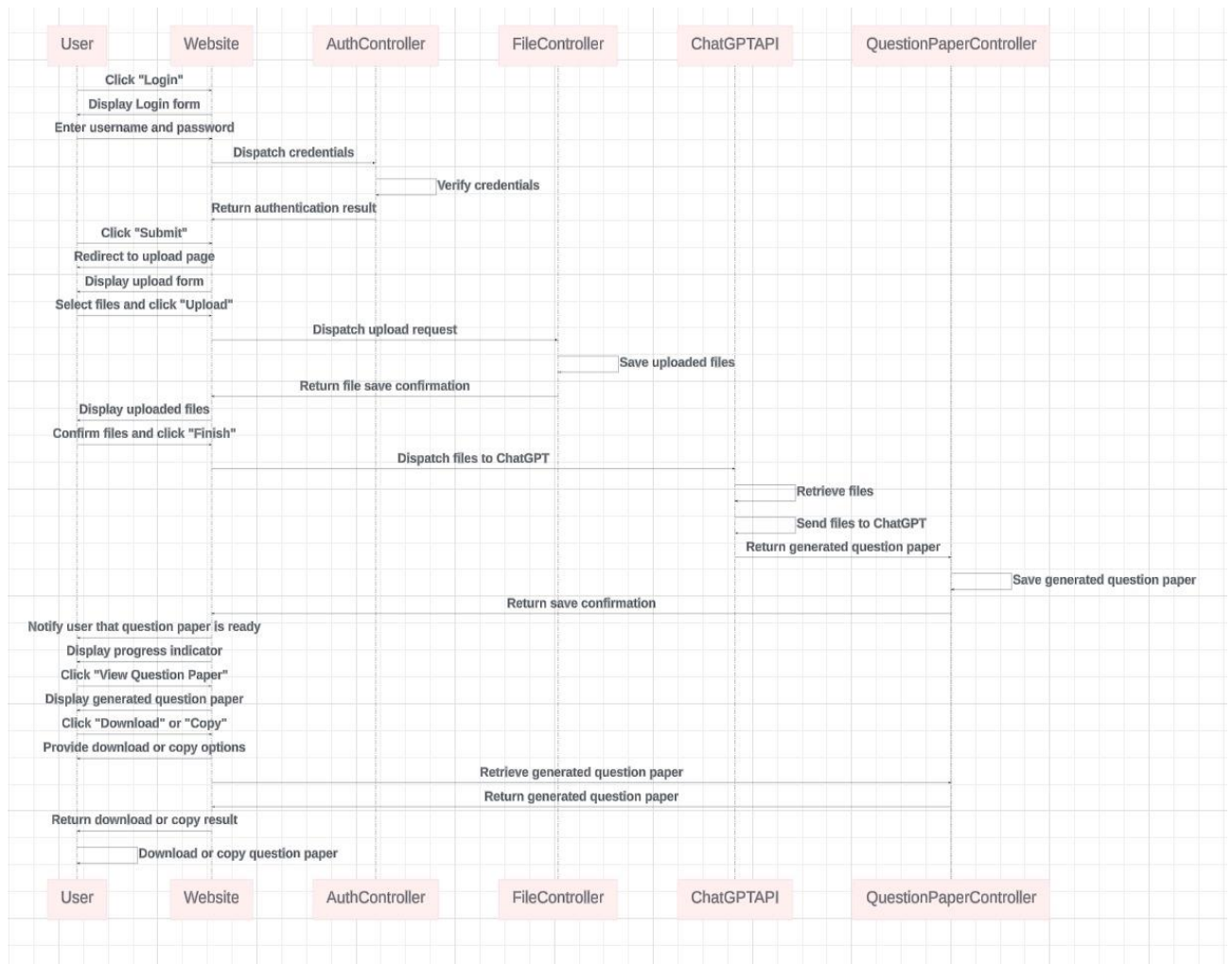
# Software Concepts & Engineering - Lab Manual

## MODEL VIEW CONTROLLER (MVC) DIAGRAM:



## SEQUENCE DIAGRAM:

# Software Concepts & Engineering - Lab Manual



- Architecture pattern used and why

## Microservices Architecture



# Software Concepts & Engineering - Lab Manual

## Reason for Using Microservices:

Scalability: Each service can be scaled independently based on demand, ensuring efficient use of resources.

Flexibility: Allows for the use of different technologies and frameworks for different services based on their specific needs.

Resilience: Failures in one service do not impact the entire system, improving overall system robustness.

Maintainability: Services can be developed, deployed, and maintained independently, making the system easier to manage and update.

- Design principles used and why

## Single Responsibility Principle (SRP):

Why: Each service handles a single responsibility (e.g., authentication, file upload, question paper generation), making the system modular and easier to maintain.

# Software Concepts & Engineering - Lab Manual

## Separation of Concerns:

Why: Different concerns (e.g., user interface, business logic, data storage) are separated into different layers and services, promoting clear boundaries and reducing complexity.

## Loose Coupling:

Why: Services communicate via well-defined interfaces, reducing dependencies and allowing for independent development and deployment.

## High Cohesion:

Why: Related functionalities are grouped together within the same service, improving maintainability and understandability.

## Scalability:

Why: The system is designed to handle growing amounts of work by distributing the load across multiple services and scaling them as needed.

## Resilience:

Why: The system is designed to handle failures gracefully, with error handling and logging services ensuring that issues are tracked and managed without impacting the user experience.

## Error Handling and Logging

## Error Handling:

# Software Concepts & Engineering - Lab Manual

Each service includes mechanisms to catch and handle errors gracefully, providing meaningful feedback to users and triggering notifications if needed.

Example: If file upload fails, the user is notified and prompted to retry.

## Logging:

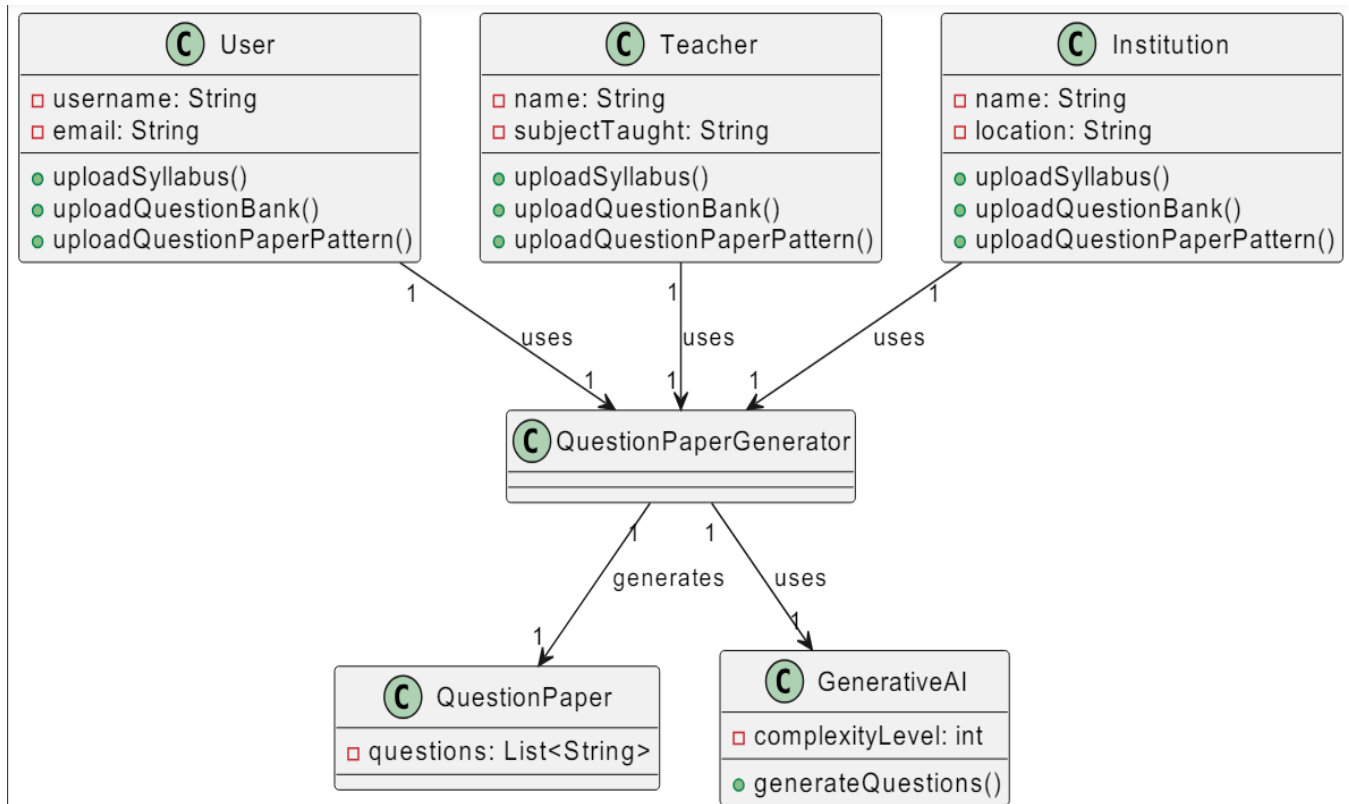
The Logging and Monitoring Service tracks all actions, errors, and system performance metrics.

Logs are stored centrally and can be analyzed to troubleshoot issues, optimize performance, and ensure compliance with security standards.

This architecture ensures a robust, scalable, and maintainable system for generating question papers, leveraging modern design principles and architectural patterns to meet the business needs efficiently.

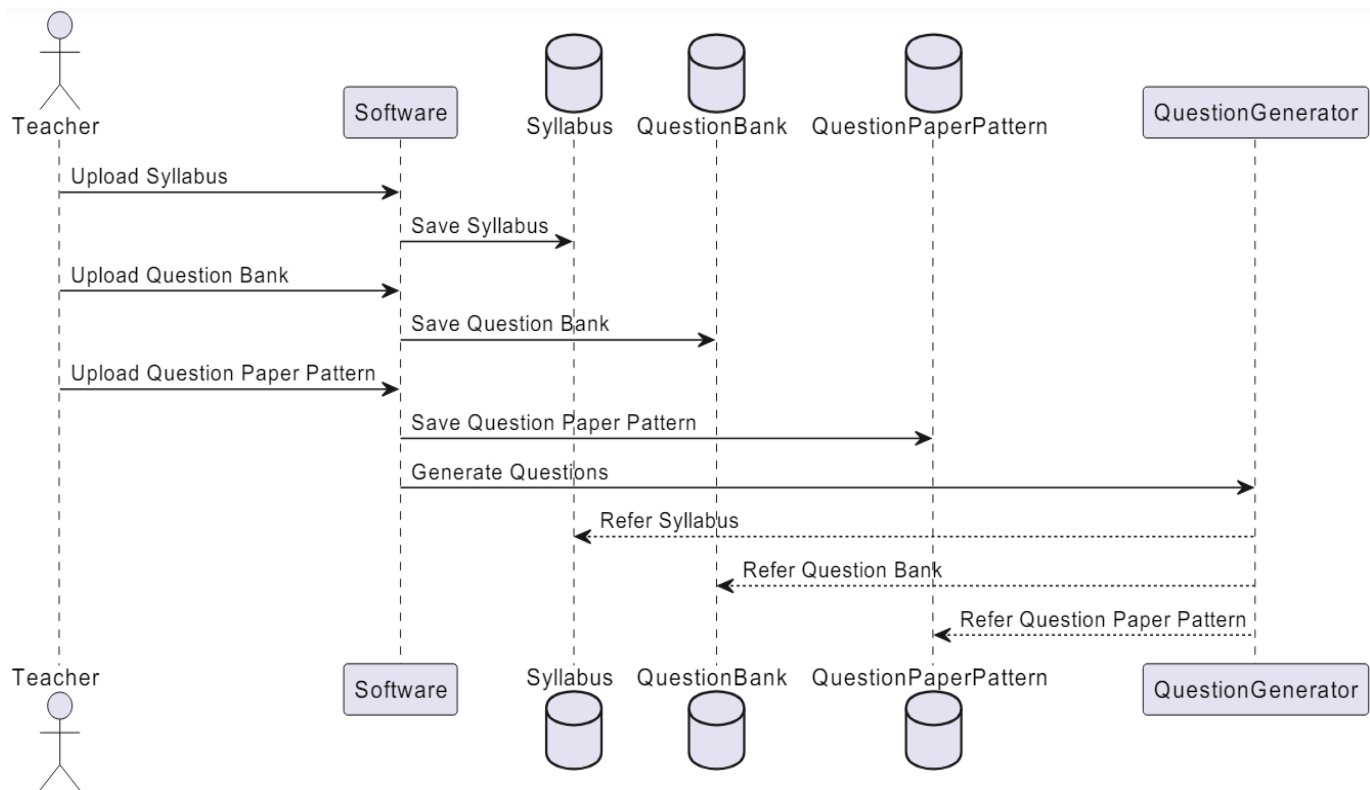
- Class diagrams
  - For all entities described in the Business Architecture Diagram, their relationships with other entities
  - Should depict the different attributes and its methods

# Software Concepts & Engineering - Lab Manual



- Sequence diagram

# Software Concepts & Engineering - Lab Manual



# Software Concepts & Engineering - Lab Manual

## Test Strategy

- Document the Test Plans

The test plans aim to ensure that all functionalities of the Question Paper Generator are working as expected. This includes verifying user interactions, file uploads, question paper generation, and error handling. The testing will cover the following types:

**Unit Testing:** Verify the functionality of individual components or services.

**Integration Testing:** Ensure that different services work together correctly.

**Functional Testing:** Validate that the system meets the specified requirements.

**Performance Testing:** Check the system's performance under load.

**Security Testing:** Ensure the system is secure from vulnerabilities.

**Usability Testing:** Validate the user experience and interface.

- Test cases for atleast 5 user stories showcasing the Happy Path and the Error Scenarios

### 1. Upload Syllabus

# Software Concepts & Engineering - Lab Manual

Happy Path:

Test Case: Verify that the user can upload a syllabus file from their device.

Steps:

Navigate to the Upload Page.

Click on the "Upload from your device" button under the Syllabus section.

Select a valid syllabus file and upload.

Verify that the file upload completes successfully and the file name is displayed.

Expected Result: The syllabus file is uploaded and displayed.

Error Scenario:

Test Case: Verify that an error message is displayed if the uploaded syllabus file is not in a valid format.

Steps:

Navigate to the Upload Page.

Click on the "Upload from your device" button under the Syllabus section.

Select an invalid file format (e.g., .exe) and upload.

Verify that an error message is displayed indicating an invalid file format.

Expected Result: The system displays an error message.

## 2. Upload Question Paper Pattern

Happy Path:

# Software Concepts & Engineering - Lab Manual

Test Case: Verify that the user can upload a question paper pattern file from Google Drive.

Steps:

Navigate to the Upload Page.

Click on the "Upload from drive" button under the Pattern section.

Select and authorize Google Drive access.

Choose a valid pattern file and upload.

Verify that the file upload completes successfully and the file name is displayed.

Expected Result: The pattern file is uploaded and displayed.

Error Scenario:

Test Case: Verify that an error message is displayed if the user cancels the Google Drive authorization.

Steps:

Navigate to the Upload Page.

Click on the "Upload from drive" button under the Pattern section.

Cancel the Google Drive authorization.

Verify that an error message is displayed indicating the authorization was canceled.

Expected Result: The system displays an error message.

## 3. Generate Question Paper

Happy Path:



# Software Concepts & Engineering - Lab Manual

Test Case: Verify that the system generates a question paper after uploading all necessary files.

Steps:

Upload a syllabus file.

Upload a pattern file.

Upload a content file.

Upload a question bank file.

Click on the "Finish uploading" button.

Verify that the system generates a question paper and displays it.

Expected Result: The question paper is generated and displayed.

Error Scenario:

Test Case: Verify that an error message is displayed if one of the required files is not uploaded.

Steps:

Upload a syllabus file.

Upload a pattern file.

Do not upload a content file.

Upload a question bank file.

Click on the "Finish uploading" button.

Verify that an error message is displayed indicating a missing file.

Expected Result: The system displays an error message.

## 4. Download Question Paper as PDF

Happy Path:

# Software Concepts & Engineering - Lab Manual

Test Case: Verify that the user can download the generated question paper as a PDF.

Steps:

Generate a question paper.

Click on the "Download as PDF" button.

Verify that the PDF is downloaded successfully.

Expected Result: The PDF file is downloaded.

Error Scenario:

Test Case: Verify that an error message is displayed if the PDF generation fails.

Steps:

Generate a question paper.

Simulate a failure in the PDF generation process (e.g., server issue).

Click on the "Download as PDF" button.

Verify that an error message is displayed indicating the failure.

Expected Result: The system displays an error message.

## 5. Notifications for Upload Status

Happy Path:

Test Case: Verify that the user receives a notification when a file upload is complete.

Steps:

Upload a file.

Wait for the upload to complete.

Verify that a notification is displayed indicating the upload is complete.

# Software Concepts & Engineering - Lab Manual

Expected Result: A notification is displayed.

Error Scenario:

Test Case: Verify that the user receives an error notification if the file upload fails.

Steps:

Upload a file.

Simulate a failure in the file upload process (e.g., network issue).

Verify that a notification is displayed indicating the upload failed.

Expected Result: An error notification is displayed.

# Software Concepts & Engineering - Lab Manual

- A view of the github repository showcasing the Project structure, their naming conventions

## GitHub Repository Structure

### RUST

```
question-paper-generator/  
├── backend/  
│   ├── auth/  
│   │   ├── AuthController.js  
│   │   ├── AuthService.js  
│   │   └── AuthRoutes.js  
│   ├── file-upload/  
│   │   ├── FileUploadController.js  
│   │   ├── FileUploadService.js  
│   │   └── FileUploadRoutes.js  
│   ├── question-paper/  
│   │   ├── QuestionPaperController.js  
│   │   ├── QuestionPaperService.js  
│   │   └── QuestionPaperRoutes.js  
│   ├── notification/  
│   │   ├── NotificationService.js  
│   │   └── NotificationController.js  
│   ├── logging/  
│   │   └── Logger.js  
│   └── app.js  
├── frontend/  
│   ├── components/  
│   │   ├── Login.js  
│   │   ├── Upload.js  
│   │   └── Result.js  
│   ├── assets/  
│   ├── styles/  
│   │   └── styles.css  
│   └── index.js  
├── tests/  
│   ├── backend/  
│   └── frontend/  
└── README.md
```

# Software Concepts & Engineering - Lab Manual

- A view of their DevOps Architecture for their respective project and the associated tools used in Azure

## DevOps Architecture

### Overview

The DevOps architecture for the Question Paper Generator project is designed to automate the deployment, testing, and monitoring of the application using Azure DevOps tools. It includes CI/CD pipelines, automated testing, and monitoring.

## Tools Used in Azure

### Azure DevOps:

#### Azure Repos:

For version control.

#### Azure Pipelines:

For CI/CD pipelines.

#### Azure Boards:

For project management and tracking user stories.

### Azure App Service:

Hosting the web application (front-end and back-end).

Azure Storage:

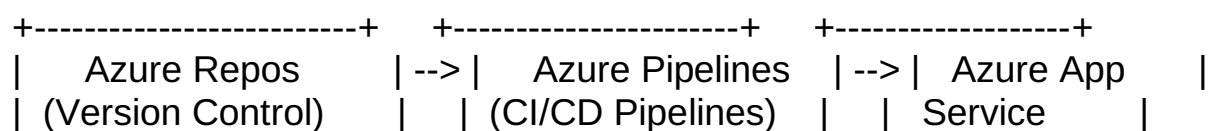
# Software Concepts & Engineering - Lab Manual

For storing uploaded files and generated question papers.

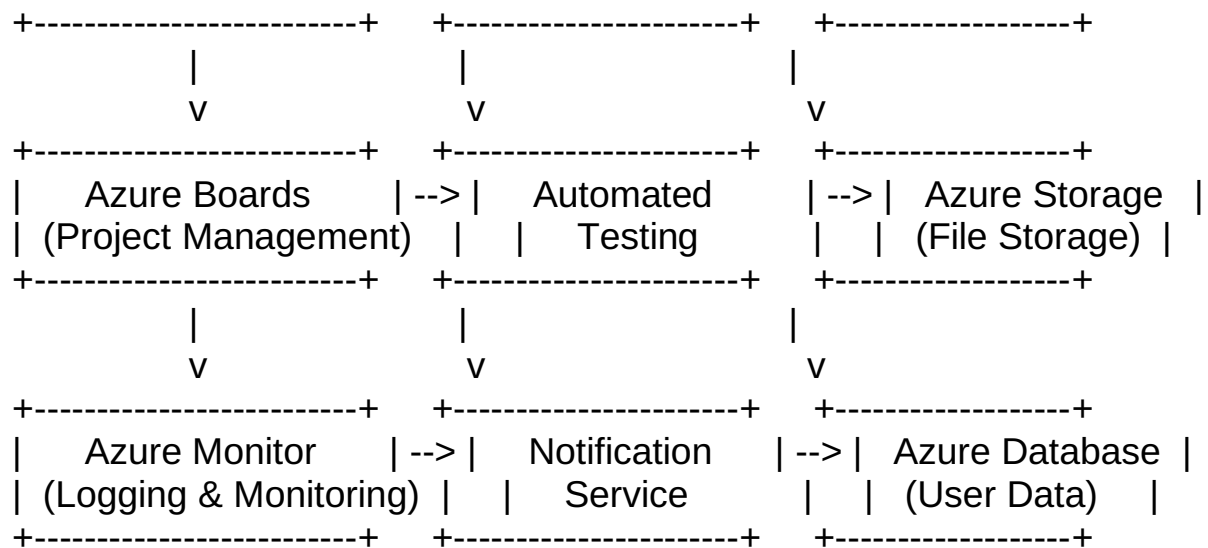
## Azure Monitor:

For logging and monitoring application performance and errors.

## DevOps Architecture Diagram



# Software Concepts & Engineering - Lab Manual



## Explanation of the DevOps Architecture

### Azure Repos:

Hosts the source code and supports branching, merging, and pull requests.

Azure Pipelines:

### CI Pipeline:

Automatically builds and tests the code upon each commit.

### CD Pipeline:

Deploys the application to Azure App Service upon successful builds.

# Software Concepts & Engineering - Lab Manual

## **Azure Boards:**

Manages the project, user stories, tasks, and bugs.

## **Azure App Service:**

Hosts the web application, ensuring it is always available to users.

## **Azure Storage:**

Stores uploaded files (syllabus, patterns, content, question banks) and generated question papers.

## **Azure Monitor:**

Tracks application performance, logs errors, and monitors the health of the application.

## **Automated Testing:**

Runs automated tests to ensure new changes do not break existing functionality.

## **Notification Service:**

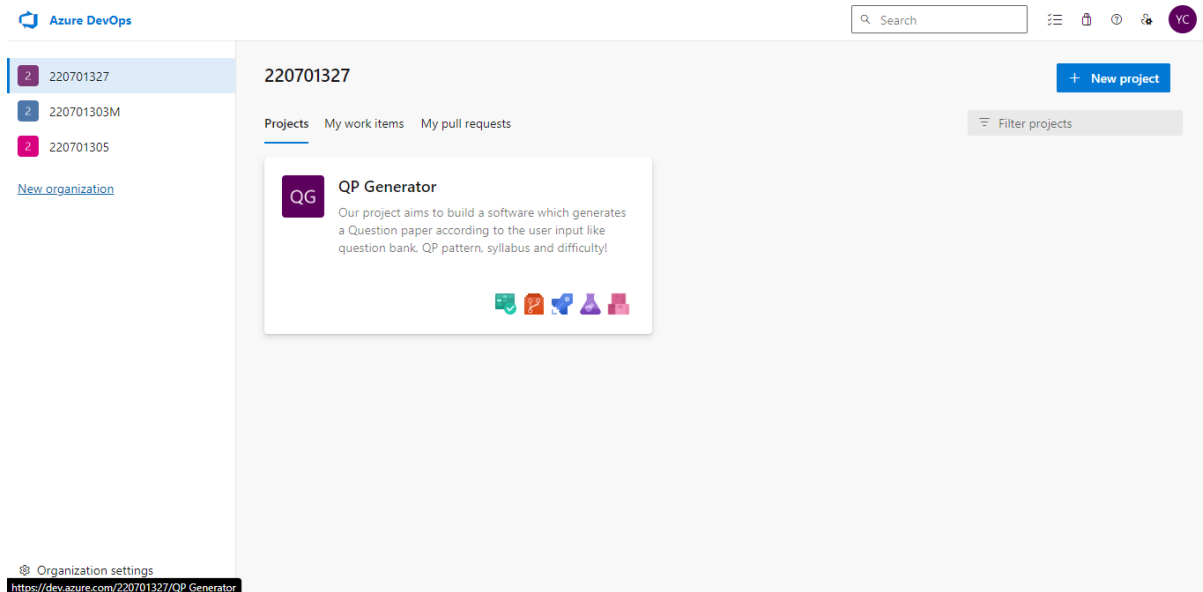
Sends notifications to users about the status of their uploads and other important events.

## **PROJECT FLOW:**

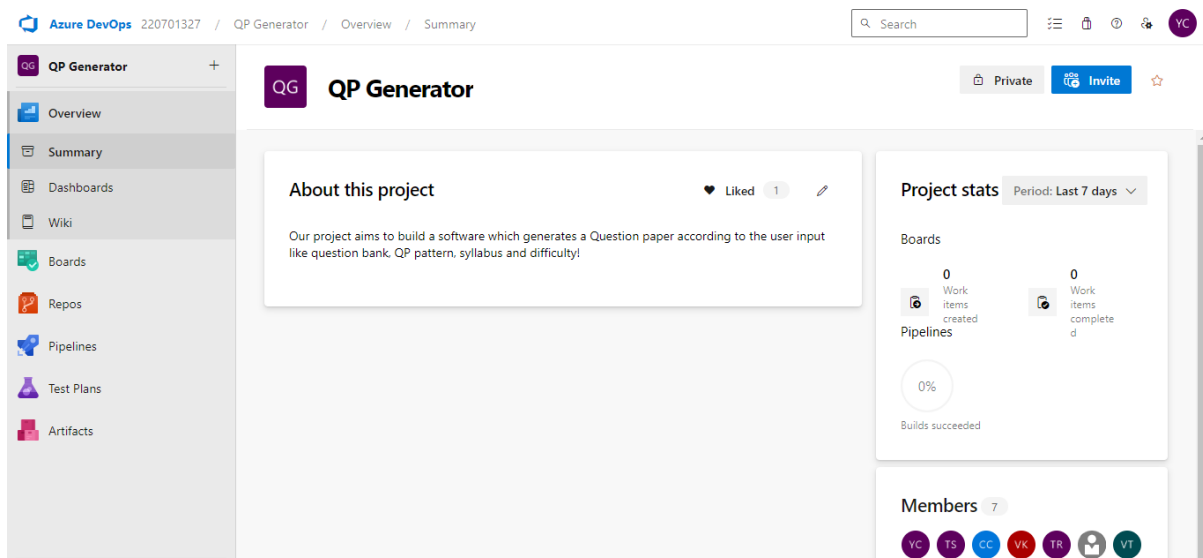
### **1. PROJECT CREATION**



# Software Concepts & Engineering - Lab Manual



## 2.PROJECT DESCRIPTION



# Software Concepts & Engineering - Lab Manual

## 3.ADDING TEAM MEMBERS:

The screenshot shows the 'QP Generator Team' settings page in Jira. The left sidebar contains navigation options: Project Settings, General, Overview, Teams (selected), Permissions, Notifications, Service hooks, Dashboards, Boards, Project configuration, Team configuration, GitHub connections, Pipelines, Agent pools, and Parallel jobs. The main content area shows the team's members. A search bar at the top right allows searching for users and groups. The members table lists six users, including an admin.

Total	6	Direct Members	Add
	Name	Type	Username, Scope, or App Id
<input type="checkbox"/>	220701303@rajalakshmi.edu.in 220701303@rajalakshmi.edu.in	user	220701303@rajalakshmi.edu.in
<input type="checkbox"/>	220701309@rajalakshmi.edu.in 220701309@rajalakshmi.edu.in	user	220701309@rajalakshmi.edu.in
<input type="checkbox"/>	220701321@rajalakshmi.edu.in 220701321@rajalakshmi.edu.in	user	220701321@rajalakshmi.edu.in
<input type="checkbox"/>	C R Sarwashri Chinnadurai 220701511@rajalakshmi.edu.in	user	220701511@rajalakshmi.edu.in
<input type="checkbox"/>	Venkata Hema Kumar 220701315@rajalakshmi.edu.in	user	220701315@rajalakshmi.edu.in
<input type="checkbox"/>	Yogesh CV Admin 220701327@rajalakshmi.edu.in	user	220701327@rajalakshmi.edu.in

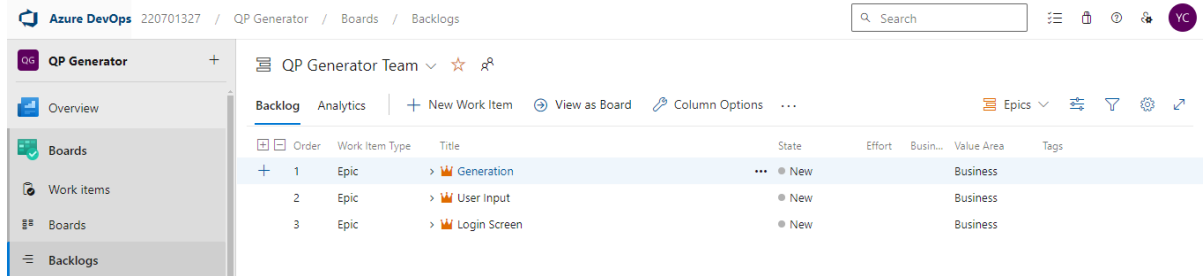
## 4.CREATED SPRINTS:

The screenshot shows the 'QP Generator' project settings page in Jira, specifically the 'Iterations' section. The left sidebar is the same as in the previous screenshot. The main content area shows a message about using the Agile process and a link to the process customization page. Below this, there is a table of iterations for the 'QP Generator' project.

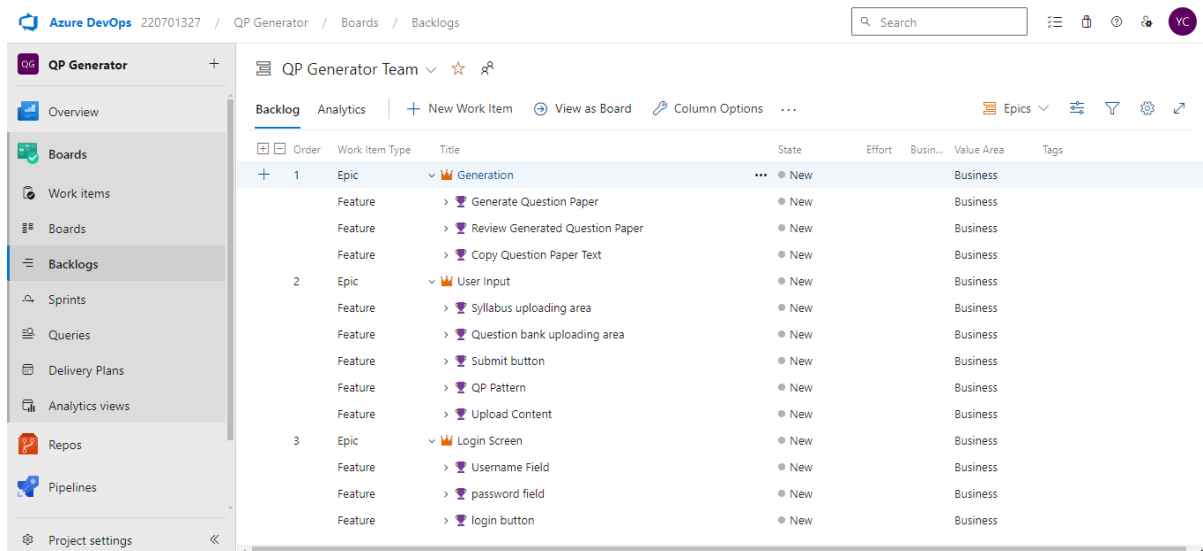
Iterations	Start Date	End Date
sprint 1	4/25/2024	5/5/2024
sprint 2	5/6/2024	5/16/2024
sprint 3	5/17/2024	5/27/2024

## 5.CREATED EPICS:

# Software Concepts & Engineering - Lab Manual



## 6.CREATED FEATURES FOR EACH EPICS:



## 7.CREATED USER STORIES FOR EACH EPICS:

# Software Concepts & Engineering - Lab Manual

## USER STORY 1:

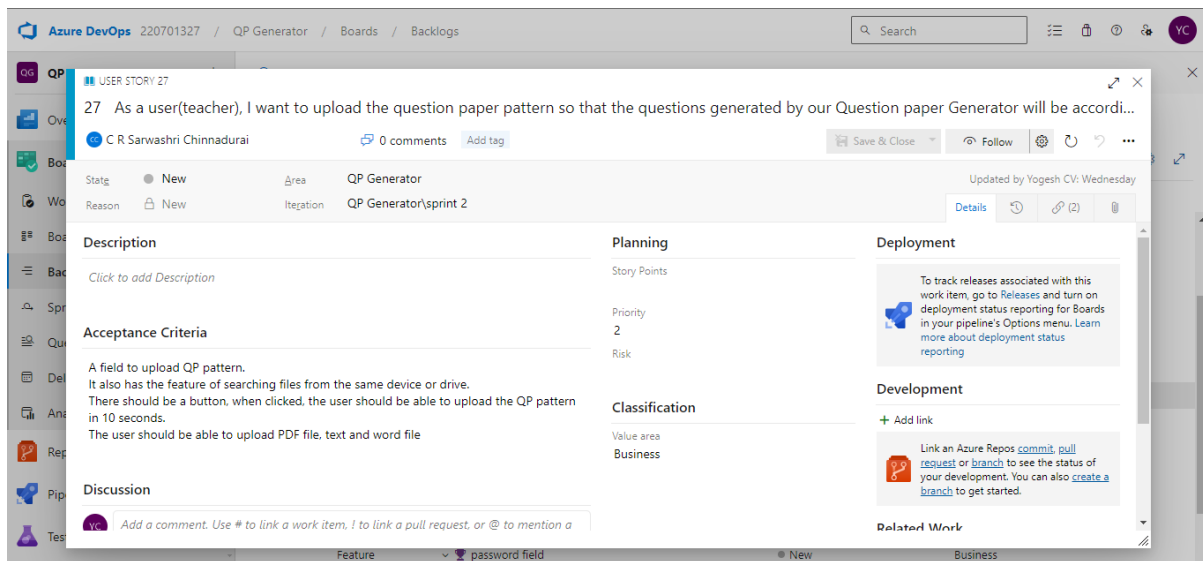
The screenshot displays the Azure DevOps interface for User Story 32. The story is titled "As a user, I want to login to my account so I can access this website" and is assigned to the user 220701309@rajalakshmi.edu.in. It is currently in the "New" state. The "Area" is set to "QP Generator" and the "Iteration" is "QP Generator\sprint 1". The "Description" field is empty, with a placeholder "Click to add Description". The "Acceptance Criteria" section contains the text: "The username must match with the password when the user clicks this button it should login within 5 seconds". The "Discussion" section is empty, with a placeholder "Add a comment. Use # to link a work item, I to link a pull request, or @ to mention a person." The "Planning" section shows "Story Points" as empty, "Priority" as 2, and "Risk" as empty. The "Classification" section shows "Value area" as empty and "Business" as selected. The "Deployment" section contains a link to "Releases" and a link to "Options menu". The "Development" section contains a link to "Add link" and a link to "Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started." The "Related Work" section is empty.

## USER STORY 2:

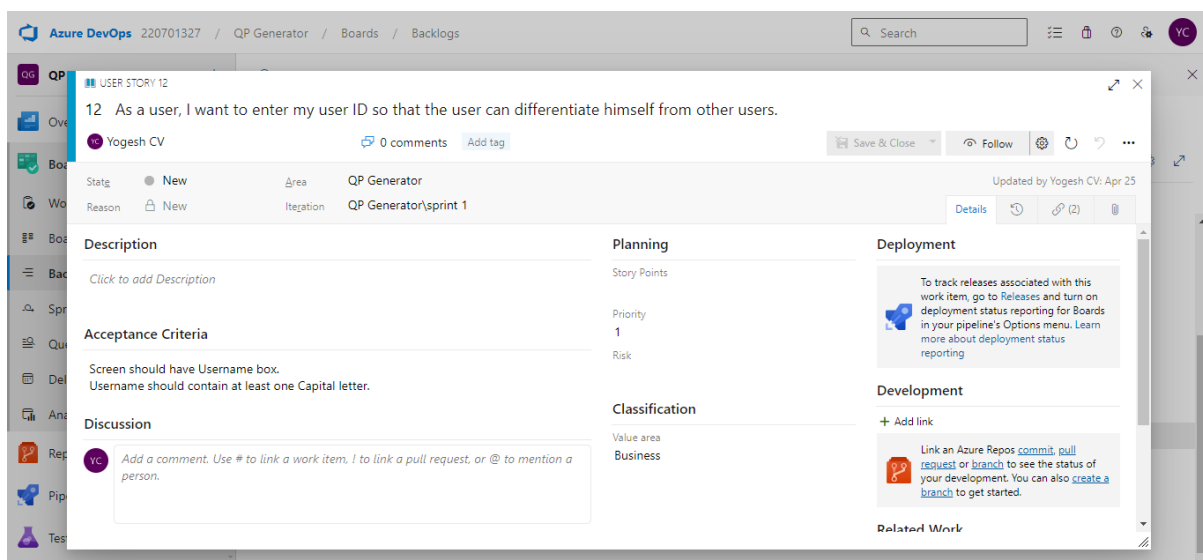
The screenshot displays the Azure DevOps interface for User Story 23. The story is titled "As a user(teacher), I want to click the submit button so that I can complete uploading my documents necessary for the QP" and is assigned to the user "Unassigned". It is currently in the "New" state. The "Area" is set to "QP Generator" and the "Iteration" is "QP Generator\sprint 3". The "Description" field is empty, with a placeholder "Click to add Description". The "Acceptance Criteria" section contains the text: "Button which has written Submit on it. On clicking it, should give the files uploaded by the user to the Generative AI which generates the QP. On clicking it, User could complete uploading the documents in 20 seconds". The "Discussion" section is empty, with a placeholder "Add a comment. Use # to link a work item, I to link a pull request, or @ to mention a person." The "Planning" section shows "Story Points" as empty, "Priority" as 2, and "Risk" as empty. The "Classification" section shows "Value area" as empty and "Business" as selected. The "Deployment" section contains a link to "Releases" and a link to "Options menu". The "Development" section contains a link to "Add link" and a link to "Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started." The "Related Work" section is empty.

## USER STORY 3:

# Software Concepts & Engineering - Lab Manual

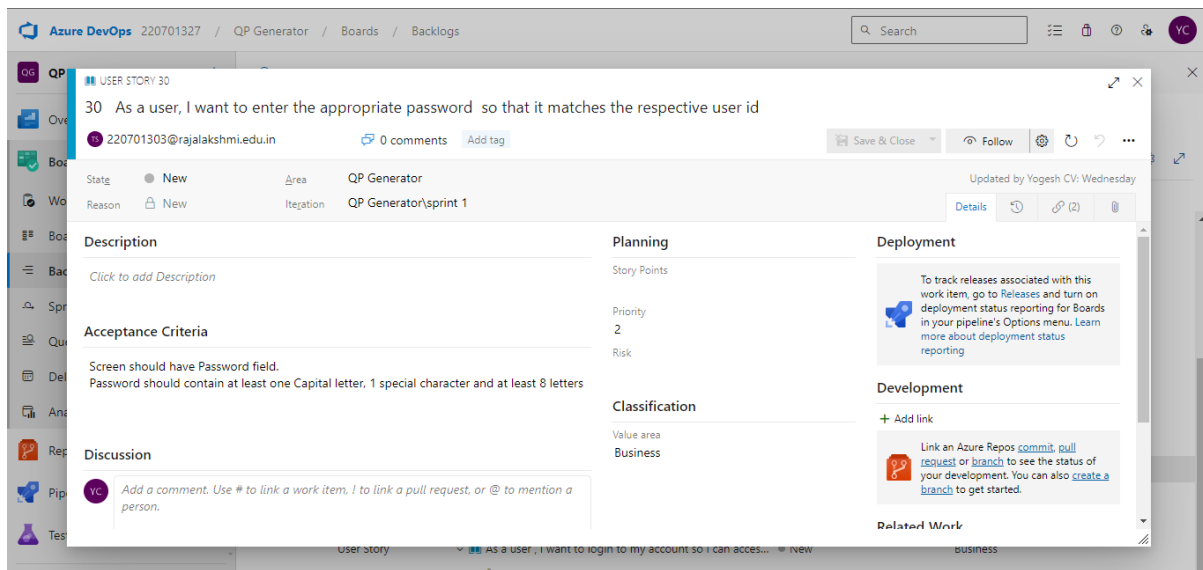


## USER STORY 4:

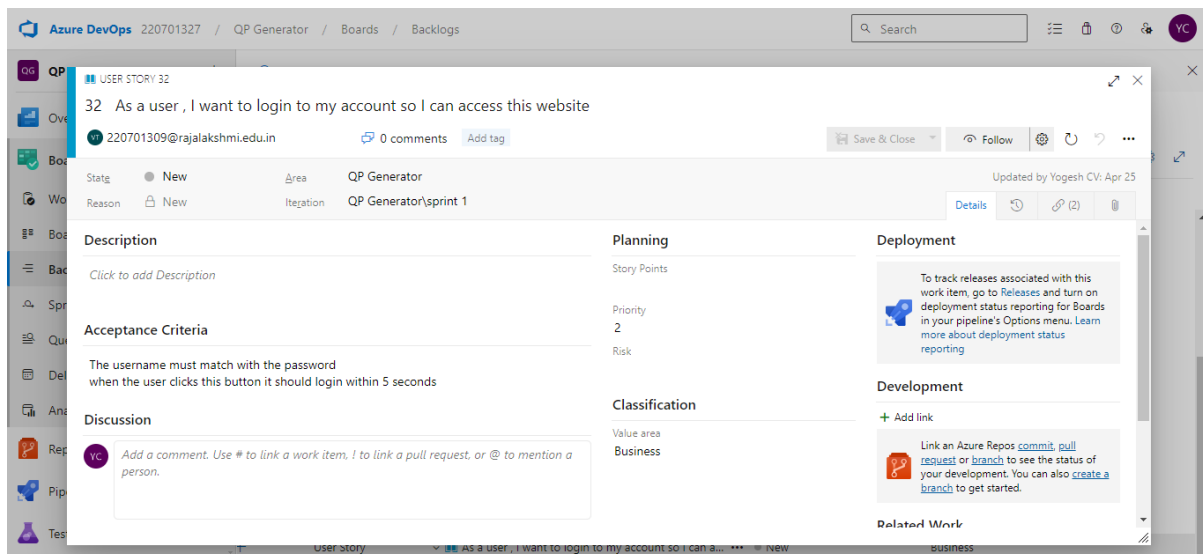


## USER STORY 5:

# Software Concepts & Engineering - Lab Manual

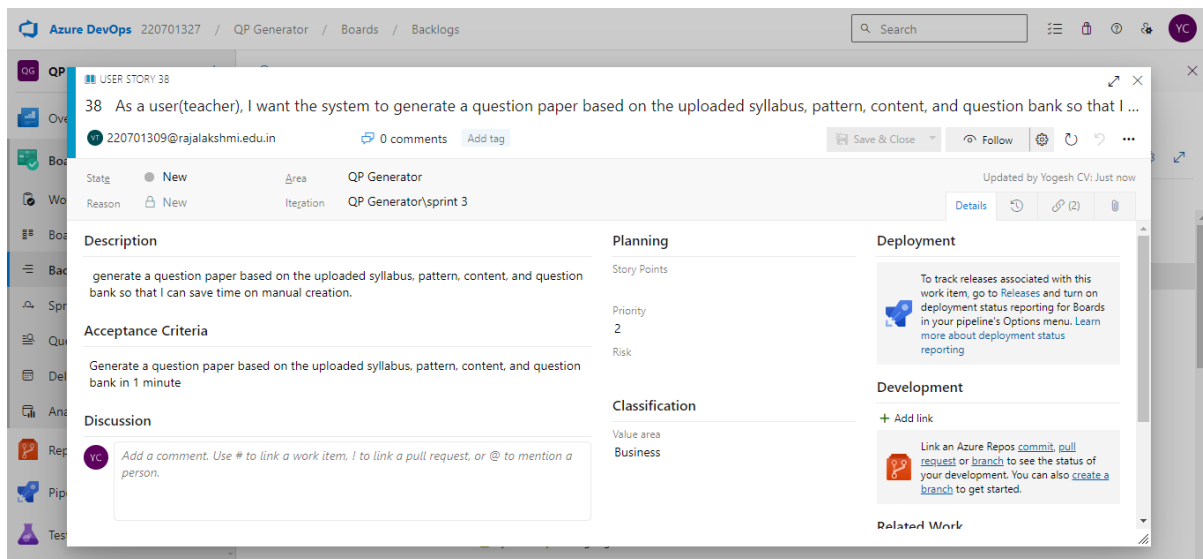


## USER STORY 6:

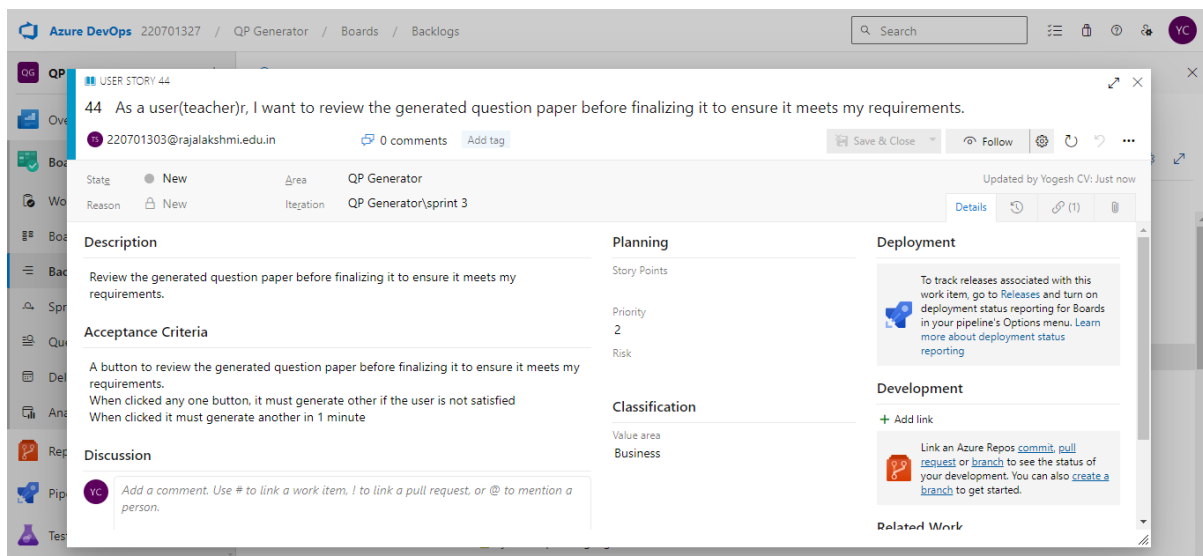


## USER STORY 7:

# Software Concepts & Engineering - Lab Manual

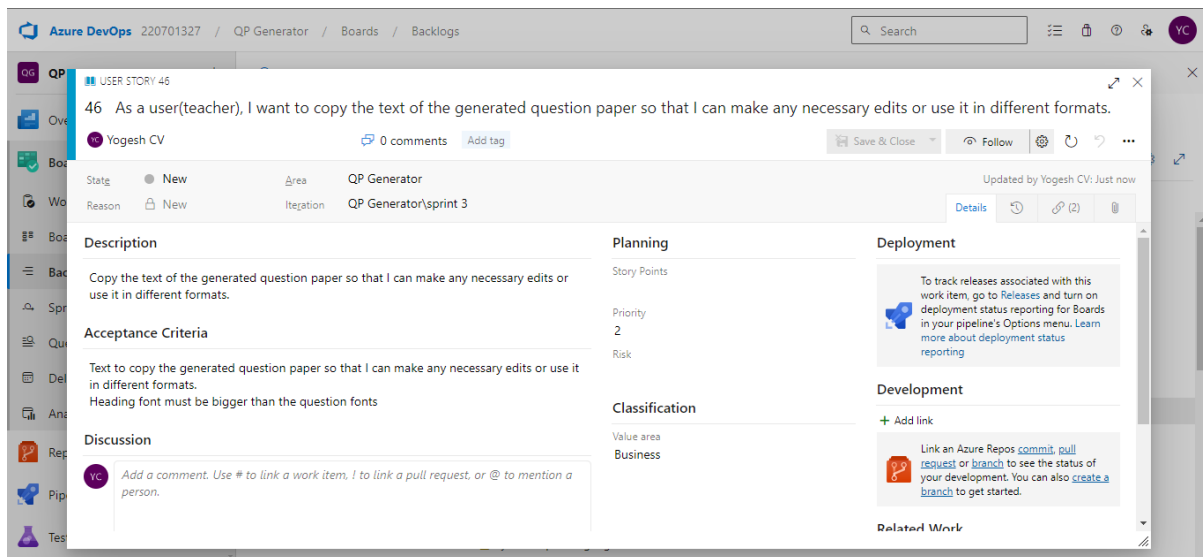


## USER STORY 8:

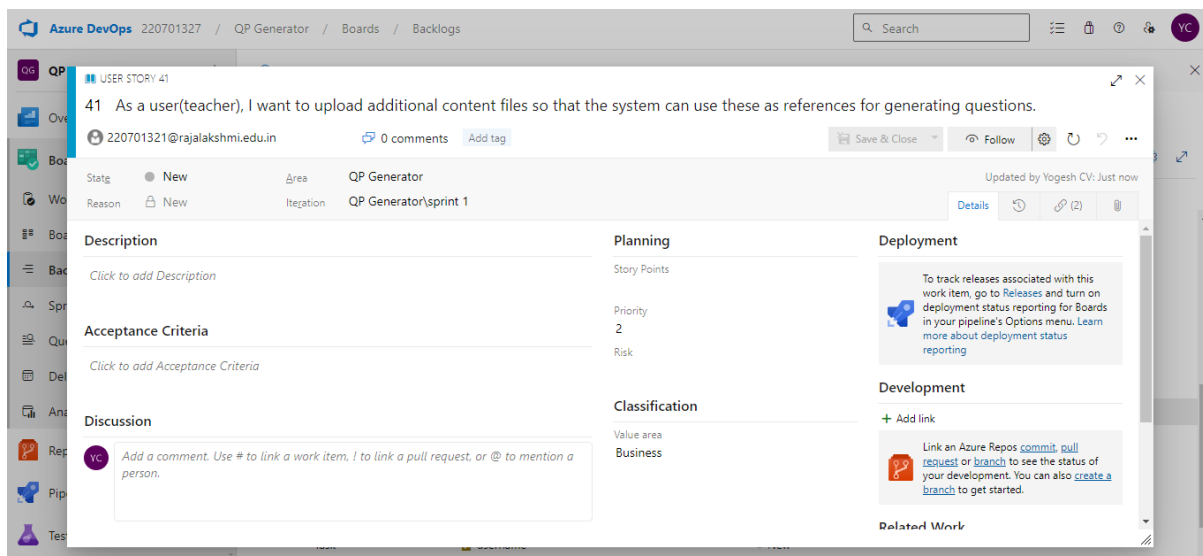


## USER STORY 9:

# Software Concepts & Engineering - Lab Manual



## USER STORY 10:



## CREATED TASKS FOR EACH USER STORIES:



# Software Concepts & Engineering - Lab Manual

The image displays three screenshots of the Azure DevOps Backlog view for the 'QP Generator Team'. Each screenshot shows a list of work items in a table format, with columns for Order, Work Item Type, Title, State, Effort, Business Area, Value Area, and Tags. The work items are categorized by type (Task, Feature, User Story, Epic) and include details like 'As a user(teacher), I want to upload additional content' and 'Generate Question Paper'.

**Screenshot 1:** Shows a backlog with items like 'QP pattern', 'Upload Content', 'Content upload', 'Login Screen', 'Username Field', 'username', 'password field', 'password', 'login button', and 'Login button'. The 'User Story' 'As a user(teacher), I want to upload additional content' is highlighted.

**Screenshot 2:** Shows a backlog with items like 'Generation', 'Generate Question Paper', 'Generate QP', 'Review Generated Question Paper', 'Copy Question Paper Text', and 'Generate text of QP'. The 'Epic' 'Generation' is highlighted.

**Screenshot 3:** Shows a backlog with items like 'User Input', 'Syllabus uploading area', 'Question bank uploading area', 'Question Bank', 'Submit button', 'QP Pattern', and 'Upload Content'. The 'Epic' 'User Input' is highlighted.

## ASSIGNED TASKS ACCORDING TO SPRINTS:

# Software Concepts & Engineering - Lab Manual

The image displays three screenshots of the Azure DevOps Sprints board for the 'QP Generator Team'. Each screenshot shows a Kanban board with columns for 'New', 'Active', 'Resolved', and 'Closed'. The board is populated with work items, each represented by a card with a title, description, and assignee.

**Screenshot 1 (Top):** Shows work items 23, 24, 38, 39, 44, and 46. The 'New' column contains items 23, 38, 44, and 46. The 'Active' column contains items 24 and 39. The 'Resolved' and 'Closed' columns are empty.

**Screenshot 2 (Middle):** Shows work items 12, 30, 32, and 41. The 'New' column contains items 12, 30, 32, and 41. The 'Active' column contains items 33, 34, 35, and 42. The 'Resolved' and 'Closed' columns are empty.

**Screenshot 3 (Bottom):** Shows work items 19, 20, and 27. The 'New' column contains items 19, 20, and 27. The 'Active' column contains items 21, 25, and 28. The 'Resolved' and 'Closed' columns are empty.