F24-050-D-Scholarly

Project Team

M.Umair Khalid	21I-0455
Ali Umer	21I-0380
Areeb Hayat	21I-0435

Session 2021-2025

Supervised by

Dr. Asif Muhammad

Co-Supervised by

Dr. Ahmad Raza Shahid

Department of Computer Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

June, 2025

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this project proposal document is to outline the development of “Scholarly”, an AI-driven Smart Learning app that helps students overcome the difficulty of staying focused while studying via textbooks or long video lectures. This document aims to provide a comprehensive overview of the problem, explain the motivation behind solving the identified problem, highlight the stakeholders, propose a solution, and define the scope of this project.

In today’s world the education system is rapidly evolving and the adoption of digital methods to learn and understand is also growing rapidly. Amongst these digital methods, learning via hour-long video lectures is quite popular. However, research shows that maintaining focus for a long period of time is extremely difficult and challenging, especially in a self-paced learning environment. Many students report the issue of losing focus whilst learning new concepts via hour-long lectures or YouTube tutorials. To address this issue, the proposed idea is to create an educational app that revolutionizes the way students learn by using Artificial Intelligence to turn textbooks into short and engaging video reels. This would not only make it easier for students to learn in small, manageable segments but automatically refresh their focus resulting in solving the issue of retaining focus while studying. Furthermore, students can also convert long textual explanations into short summaries and generate MCQ-based quizzes to test their knowledge and enhance their learning.

1.1 Existing Solutions

What Scholarly aims to do is to develop a mobile application that revolutionizes how students comprehend and interact with learning-based content. This app will not only enable it’s user to convert long textual content into summarized explanations and into short and engaging reels to effectively study without losing focus but also allow users to attempt an MCQ based quiz to enhance their knowledge and test their learning. Some research has been done in this domain in the past few years but after the mass introduction of Artificial Intelligence this topic has seemed to grab attention of numerous digital educational platforms.

NotebookLM: NotebookLM was initially launched in July 12, 2023 by Google and is an AI-driven platform that enables it's users to generate textual summaries based on their uploaded documents. Furthermore, NotebookLM can also be used to convert the generated response into audio format for easy access. This system also allows it's users to provide different prompts to the system to refine the generated responses based on user's uploaded documents.

Evernote: Evernote is an AI Drive platform that can be used to convert Images and Video to textual form. Evernote can also be used to convert long text descriptions to short text summarization for better understanding and clarity.

Already Existing AI Models: Various pre-trained AI models already exist in the market that can be used to generate text summaries based on given content or generate videos either on the basis of given prompt, script or text description. Some of the famous AI models to achieve these tasks include GPT-4, BERTSum, T5 for text summarization and Pictory AI,Text-to-Video,CogVideoX for video generation

Table 1.1: Comparison of Existing Solutions

System Name	**System Overview**	**System Limitations**
NotebookLM	NotebookLM can be used to generate notes and textual explanations from uploaded content via prompts. It can also be used to convert generated responses to audio format.	The issue of losing focus while studying still exists as NotebookLM does not include the option to generate short video reels. Also, it is difficult to visualize user progress due to reliance on user prompts rather than following a specific roadmap.
Evernote	Evernote can be used to generate transcripts based on uploaded videos. It can also be used to convert long text descriptions into short text summaries for better understanding and clarity.	Evernote lacks the ability to keep track of user progress and is also limited to currently uploaded content, meaning if the user logs out, the current data and progress is lost. Also, Evernote does not provide the functionality to generate short-form videos based on given textual content.

System Name	System Overview	System Limitations
Already Existing AI Models	After the mass introduction of Artificial Intelligence, various pre-trained AI models have been introduced to generate text summarizations or generate videos based on some given prompt. Some of the models that can be used to achieve these tasks include GPT-4, BERTSum, and T5 for text summarization, and Pictory AI, Text-to-Video, and CogVideoX for video generation.	There exist multiple models to achieve the tasks that Scholarly is intended to achieve, but all the existing models are general-purpose and do not specifically target students. Also, all these models work independently, so there does not exist a common platform to provide all the services under a single umbrella.

1.2 Problem Statement

According to research conducted by the American Physiology Journal [1], an average human being has an attention span of 15-20 minutes whilst watching an hour-long video but this drastically decreases to only 5-8 minutes when reading or learning via a book or any other form of textual content. In the modern educational era, students often struggle to maintain focus whilst watching long video lectures or learning via a textbook. The vast amount of information presented in lengthy video lectures can also lead to cognitive overload, making it harder for students to retain key concepts and stay engaged throughout the video lectures. This issue is worsened in subjects that require deep understanding, where consistent focus is necessary for understanding complex topics. Our research indicates that every 3 out of 4 students not only experience the issue of retaining focus while studying but are also in need of a better alternative solution that provides more concise and engaging learning materials to not only better understand complex topics with great speed and accuracy but also cater to students' attention spans.

1.3 Scope

The project aims to develop an innovative educational app to revolutionize how students comprehend and interact with learning-based content. The app will enable its users to upload a book and automatically generate a personalized learning roadmap based on the uploaded content. The generated roadmap will act as a study plan for the user and will allow the user to track their progress as they move forward. Inside the generated roadmap, the app will feature detailed explanations as well as summarized explanation for every topic the user selects to facilitate quick and efficient learning. Furthermore, the app will also enable the user to generate short videos (reels) for every topic in the generated roadmap, whose main objective will be to make complex topics easier to understand and learn them with great speed and accuracy. After a user has completed a topic, they can attempt a quiz of that topic to test their knowledge. The app also features a dedicated user settings and dashboard for personalization and user data management; enabling the users to view their analytics and recommendations for the generated learning roadmap. The main objective behind this project is to enhance students' learning experience by combining traditional study methods with cutting-edge AI technology, making education more interactive, personalized, and efficient.

1.4 Modules

1.4.1 User Management

This module will focus on the development of a prototype, enabling a secure login and registration process. After logging in, the personalized dashboard will allow users to monitor their learning progress, access uploaded materials, and engage with other app features.

1. User Authentication: To securely authenticate the user and provide access to the system
2. User Profile: A personalized user profile, to view and manage user's personal information
3. User Settings: Enable user to effectively customize, and update their user data
4. Personalized Dashboard: A dedicated interface to track progress, view insights and analytics about the uploaded content

1.4.2 Content Upload and Roadmap Generation

This module will add the functionality of Book Upload that will act as an input for the system and will automatically generate a personalized learning roadmap for the user. The generated roadmap will act as a structured study plan for the user and will visually present the topics and subtopics in a chronological order based on the uploaded content.

1. Book Upload: Allow the user to upload books in the supported format and enable the system to process and analyze the uploaded content
2. Generate Roadmap: Automatically generate a learning roadmap based on the uploaded book

3. View Roadmap: Enable the user to view the generated roadmap

1.4.3 Text Summarization and Reels Generation

This module will focus on the development of features such as text summarization, which will enable the user to generate short text explanations for the topics and subtopics in the generated roadmap. This module will also feature the development of short videos (reels) for the roadmap, allowing the user to understand the content in a visual representation.

1. Generate Topic Summary: Generate summarized textual explanation based on the selected topic
2. Generate Reels: Allow the user to convert textual explanations into engaging short-form video reels for better understanding.
3. View Reels: Enable the user to view the generated reels

1.4.4 Quiz Generation

This module will focus on the functionality of quiz generation, which will enable the user to generate a quiz based on their selected topic or subtopic in the generated roadmap. The user can then attempt the quiz to test their knowledge, and further enhance their learning by knowing their mistakes.

1. Generate Quiz: Generate MCQ based quiz given the selected topic
2. Attempt Quiz: Enable the user to attempt the quiz which will be evaluated in real-time.

1.5 Work Division

Table 1.2: Work Division

Name	Registration	Responsibility/ Module / Feature
M.Umair Khalid	21I-0455	(Module 1 - Feature 1) User Authentication (Module 2 - Feature 1) Book Upload
Ali Umer	21I-0380	(Module 1 - Feature 4) Personalized Dashboard (Module 2 - Feature 3) View Roadmap
Areeb Hayat	21I-0435	(Module 1 - Feature 2) User Profile (Module 1 - Feature 3) User Settings (Module 3 - Feature 3) View Reels
Team Work	- - - - - - -	(Module 3 - Feature 1) Generate Topic Summary (Module 2 - Feature 2) Generate Roadmap (Module 3 - Feature 2) Generate Reels (Extra Feature) Quiz Generation (Extra Feature) Attempt Quiz

Chapter 2

Project Requirements

2.1 Use-case/Event Response Table/Storyboarding

2.1.1 Use Case Diagram

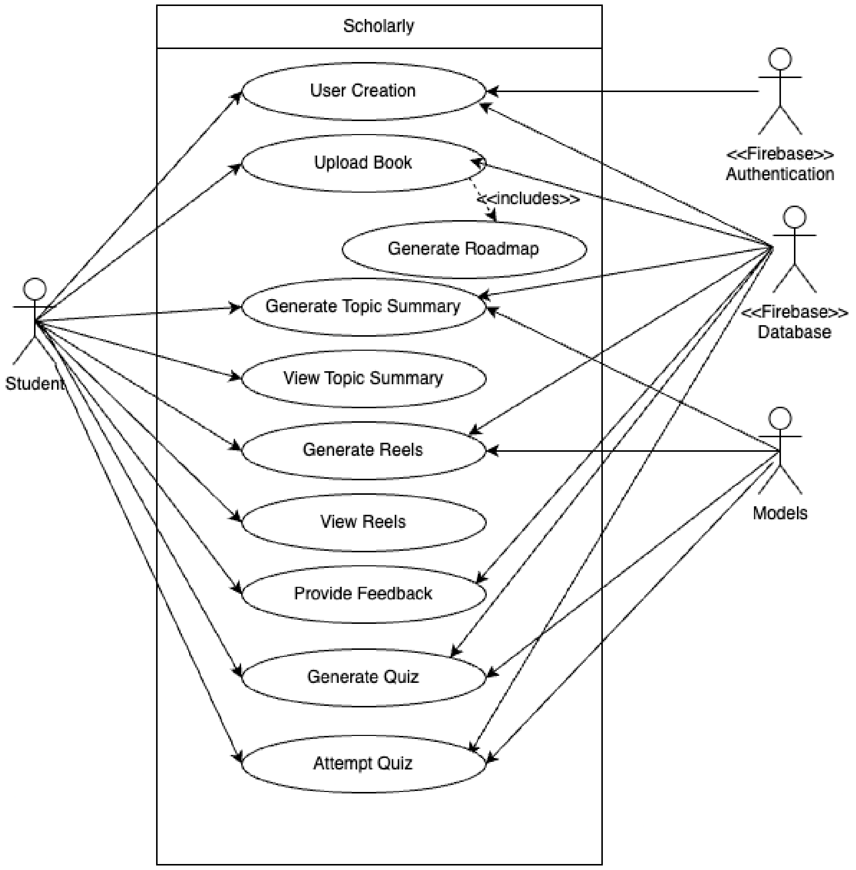


Figure 2.1: Use Case Diagram

2.1.2 User Creation

<table>
<tr><td><b>ID</b></td><td>UC001</td></tr>
<tr><td><b>Name</b></td><td>User Creation</td></tr>
<tr><td><b>Scope</b></td><td>Scholarly</td></tr>
<tr><td><b>Level</b></td><td>User Level</td></tr>
<tr><td><b>Description</b></td><td>This use case enables students to register by providing valid credentials. Upon verification, the system creates the account, updates the database, and logs the student in securely.</td></tr>
<tr><td><b>Primary Actor</b></td><td>Student</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>Student</b> – Interested in creating a new user account.
<b>System</b> – Interested in providing account creation and authentication service to the student.</td></tr>
<tr><td><b>Trigger</b></td><td>A student fills the user registration form and clicks the register button</td></tr>
<tr><td><b>Preconditions</b></td><td>Student does not have a user account before.
Student has all the valid information available such as Email and Password etc.</td></tr>
<tr><td><b>Postconditions</b></td><td>Student has successfully created a new user account and logged into the system.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Student desires to create a new user account.</td><td></td></tr>
<tr><td>Student provides the necessary information such as Email, Username, Password etc.</td><td></td></tr>
<tr><td></td><td>System checks if the information provided by the user is correct or not.</td></tr>
<tr><td></td><td>System creates a new user account and updates the database.</td></tr>
<tr><td></td><td>System logs in the new user.</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td>In case of any system error or any technical issue, the student may not be able to register an account.
Student does not have the necessary information available, or the user has submitted wrong information. In both cases, the user cannot have a registered account.</td></tr>
</table>

Frequency of Use	Once for unregistered user
Constraints and Special Requirement	None
Assumptions	None
Notes and Issues	None

Table 2.1: User Creation

2.1.3 Upload Book

ID	UC002
Name	Upload Book
Scope	Scholarly
Level	User Level
Description	This use case allows students to upload a book in pdf format that the system pre-processes it and stores it in the database.
Primary Actor	Student
Stakeholders and Interests	**Student** – Interested in uploading their course book to generate a roadmap. **System** – Interested in getting the uploaded course book and updating the database.
Preconditions	Student has a registered account and has logged in to the system.
Postconditions	The system updates the database and redirects to the “Roadmap Page”.

<table>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Student enters relevant information to log in.</td><td></td></tr>
<tr><td></td><td>System authenticates the information and grants access to the student.</td></tr>
<tr><td>Student clicks on the “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student clicks on the “Upload New Book” button.</td><td></td></tr>
<tr><td></td><td>System displays an upload pop-up.</td></tr>
<tr><td>Student uploads the book.</td><td></td></tr>
<tr><td></td><td>System preprocesses the book and updates the database.</td></tr>
<tr><td></td><td>System redirects the student to the “Roadmap Page”.</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td>In case of any system error or technical issue, the student may not be able to upload their course book.
In case of any system error or technical issue, the system may not be able to update the database or redirect to the “Roadmap Page”.
The student does not remember their login credentials.</td></tr>
<tr><td><b>Includes</b></td><td>Generate Roadmap</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Every time a student wants to upload a book (maximum of 6 books per user).</td></tr>
<tr><td><b>Constraints and Special Requirement</b></td><td>None</td></tr>
<tr><td><b>Assumptions</b></td><td>None</td></tr>
<tr><td><b>Notes and Issues</b></td><td>None</td></tr>
</table>

Table 2.2: Upload Book

2.1.4 Generate Roadmap

<table>
<tr><td><b>ID</b></td><td>UC003</td></tr>
<tr><td><b>Name</b></td><td>Generate Roadmap</td></tr>
<tr><td><b>Scope</b></td><td>Scholarly</td></tr>
<tr><td><b>Level</b></td><td>System Level</td></tr>
<tr><td><b>Description</b></td><td>This use case enables the system to create a personalized study roadmap based on the uploaded course book.</td></tr>
<tr><td><b>Primary Actor</b></td><td>System</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>Student</b> – Interested in generating a roadmap for the uploaded course book.
<b>System</b> – Interested in providing a roadmap to the student for the uploaded course book.</td></tr>
<tr><td><b>Preconditions</b></td><td>Student has a uploaded their course book.</td></tr>
<tr><td><b>Postconditions</b></td><td>Student can view the generated roadmap for their course book.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Student clicks on “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student select a book and clicks on the upload book button.</td><td></td></tr>
<tr><td></td><td>System retrieves the book contents from the database.</td></tr>
<tr><td></td><td>System preproccess the book and generates a personalized roadmap</td></tr>
<tr><td></td><td>System displays the roadmap to the student.</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td>In case of any system error or technical issue, the system may not be able to generate and display the roadmap.
In case of any system error or technical issue, the student may not be able to view their generated roadmap.
The student does not remember their login credentials.</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Every time a student uploads a book (maximum of 6 books allowed).</td></tr>
</table>

Constraints and Special Requirement	None
Assumptions	None
Notes and Issues	None

Table 2.3: Generate Roadmap

2.1.5 Generate Topic Summary

<table>
<tr><td><b>ID</b></td><td>UC004</td></tr>
<tr><td><b>Name</b></td><td>Generate Topic Summary</td></tr>
<tr><td><b>Scope</b></td><td>Scholarly</td></tr>
<tr><td><b>Level</b></td><td>System Level</td></tr>
<tr><td><b>Description</b></td><td>This use case allows the student to generate a textual summary based on a selected topic.</td></tr>
<tr><td><b>Primary Actor</b></td><td>System</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td>System – Interested in generating a text summary for the student based on a selected topic.</td></tr>
<tr><td><b>Preconditions</b></td><td>System has generated a roadmap and a topic has been selected by the student.</td></tr>
<tr><td><b>Postconditions</b></td><td>System updates the database and stores the generated topic summary.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Student selects a topic from the generated roadmap.</td><td></td></tr>
<tr><td></td><td>System fetches the description from the database based on the selected topic.</td></tr>
<tr><td></td><td>System generates a summary based on the topic description.</td></tr>
<tr><td></td><td>System updates the database and stores the generated summary.</td></tr>
</table></td></tr>
<tr><td><b>Extensions</b></td><td>In case of any system error or technical issue, the system might not be able to generate a summary based on a selected topic.</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Every time a student selects a topic to generate a topic summary.</td></tr>
</table>

Constraints and Special Requirement	None
Assumptions	None
Notes and Issues	None

Table 2.4: Generate Topic Summary

2.1.6 View Topic Summary

ID	UC005
Name	View Topic Summary
Scope	Scholarly
Level	User Level
Description	This use case allows the student to view the generated topic summary based on a selected topic.
Primary Actor	System
Primary Actor	Student
Stakeholders and Interests	**Student** – Interested in viewing a summary of their selected topic. **System** – Interested in providing a text summary to the student based on a selected topic.
Preconditions	Student has a registered account and has logged in to the system. Student has uploaded their course book and a roadmap has been generated.
Postconditions	Student can view the topic summary of their selected topic.

<table>
<tr><td rowspan="9"><b>Main Success Scenario</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td>Student enters relevant information to log in.</td><td></td></tr>
<tr><td></td><td>System authenticates information and grants access to the student.</td></tr>
<tr><td>Student clicks on “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student clicks on the roadmap.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Roadmap Page”.</td></tr>
<tr><td>Student selects a topic and clicks on “View Textual Explanation” button.</td><td></td></tr>
<tr><td></td><td>System displays the topic summary to the student.</td></tr>
<tr><td><b>Extensions</b></td><td colspan="2">In case of any system error or technical issue, the student may not be able to view the topic summary.
The student does not remember their login credentials.</td></tr>
<tr><td><b>Frequency of Use</b></td><td colspan="2">Every time a student clicks on view summary button.</td></tr>
<tr><td><b>Constraints and Special Requirement</b></td><td colspan="2">None</td></tr>
<tr><td><b>Assumptions</b></td><td colspan="2">None</td></tr>
<tr><td><b>Notes and Issues</b></td><td colspan="2">None</td></tr>
</table>

Table 2.5: View Topic Summary

2.1.7 Generate Reels

<table>
<tr><td><b>ID</b></td><td colspan="2">UC006</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Generate Reels</td></tr>
<tr><td><b>Scope</b></td><td colspan="2">Scholarly</td></tr>
<tr><td><b>Level</b></td><td colspan="2">User Level</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This use case allows the student to generate a short form video (reel) for a selected topic.</td></tr>
<tr><td><b>Primary Actor</b></td><td colspan="2">Student</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td colspan="2"><b>Student –</b> Interested in generating reels of their selected topic.
<b>System –</b> Interested in displaying the generated reels to the student based on a selected topic.</td></tr>
<tr><td><b>Preconditions</b></td><td colspan="2">Student has a registered account and has logged in to the system.
Student has uploaded their course book and a roadmap has been generated.</td></tr>
<tr><td><b>Postconditions</b></td><td colspan="2">The system updates the database and stores the generated reels.</td></tr>
<tr><td rowspan="10"><b>Main Success Scenario</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td>Student logs into the system.</td><td></td></tr>
<tr><td>Student clicks on “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student clicks on the roadmap.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Roadmap Page”.</td></tr>
<tr><td>Student selects a topic and clicks on the “Generate Reels” button.</td><td></td></tr>
<tr><td></td><td>System redirects to the reels page.</td></tr>
<tr><td></td><td>System fetches the contents from the database and generates reels based on the selected topic and description.</td></tr>
<tr><td></td><td>System redirects to the “View Reels” page.</td></tr>
</table>

Extensions	In case of any system error or technical issue, the system may not be able to generate and display the reels. The student does not remember their login credentials.
Frequency of Use	Every time a student selects a topic and clicks on the generate reel button.
Constraints and Special Requirement	None
Assumptions	None
Notes and Issues	None

Table 2.6: Generate Reels

2.1.8 View Reels

ID	UC007
Name	View Reels
Scope	Scholarly
Level	User Level
Description	This use case allows the student to view the generated short form video (reel) based on a selected topic.
Primary Actor	Student
Stakeholders and Interests	**Student** – Interested in viewing reels of their selected topic. **System** – Interested in displaying reels to the student based on their selected topic.
Preconditions	Student has a registered account and has logged in to the system. Student has uploaded their course book and a roadmap has been generated.
Postconditions	System displays the reel to the student.

<table>
<tr><td rowspan="9"><b>Main Success Scenario</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td>Student enters relevant information to log in.</td><td></td></tr>
<tr><td></td><td>System authenticates the information and grants access to the student.</td></tr>
<tr><td>Student clicks on “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student clicks on the roadmap.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Roadmap Page”.</td></tr>
<tr><td>Student selects a topic and clicks on the “View Reels” button.</td><td></td></tr>
<tr><td></td><td>System fetches the contents from the database and displays the reels to the student.</td></tr>
<tr><td><b>Extensions</b></td><td colspan="2">In case of any system error or technical issue, the student may not be able to view the reels.
The student does not remember their login credentials.</td></tr>
<tr><td><b>Frequency of Use</b></td><td colspan="2">Every time a student clicks on view reel button.</td></tr>
<tr><td><b>Constraints and Special Requirement</b></td><td colspan="2">None</td></tr>
<tr><td><b>Assumptions</b></td><td colspan="2">None</td></tr>
<tr><td><b>Notes and Issues</b></td><td colspan="2">None</td></tr>
</table>

Table 2.7: View Reels

2.1.9 Provide Feedback

ID	UC008
Name	Provide Feedback
Scope	Scholarly
Level	User Level
Description	This use case allows the student to provide a feedback based on a view short form video (reel)
Primary Actor	Student
Stakeholders and Interests	**Student** – Interested in providing feedback related to the generated reel to better optimize the system performance. **System** – Interested in recording the feedback and storing it in the database, and performing further actions accordingly.
Preconditions	Student has a registered account and has logged in to the system. System has generated the reel for the particular topic. Student has watched the reel of a particular topic for a fixed time.
Postconditions	The system updates the database and records the student's feedback.

Main Success Scenario	**Actor Action**	**System Response**
	Student enters relevant information to log in.	
		System authenticates the information and grants access to the student.
	Student clicks on "Books page".	
		System redirects the student to the "Books page".
	Student clicks on the roadmap.	
		System redirects the student to the "Roadmap Page".
	Student selects a topic and clicks on the "View Reels" button.	
		System redirects to the reels page and displays the reels based on the selected topic.
	Student watches a fixed portion of the reel.	
		System tracks the student's interaction with the reel.
	Student clicks on the "Feedback Button" and provides feedback.	
		System stores the feedback and updates the database.
Extensions	In case of any system error or technical issue, the student may not be able to provide feedback for a viewed reel. In case of any system error or technical issue, the system may not be able to update the database and store the provided feedback. The student does not remember their login credentials.	
Frequency of Use	Every time a student watches a short form video (reel) for a selected topic	
Constraints and Special Requirement	None	

Assumptions	None
Notes and Issues	None

Table 2.8: Provide Feedback

2.1.10 Generate Quiz

ID	UC009
Name	Generate Quiz
Scope	Scholarly
Level	User Level
Description	This use case allows the student to generate an MCQ based quiz for a selected topic.
Primary Actor	Student
Stakeholders and Interests	**Student** – Interested in generating quizzes of their selected topic. **System** – Interested in displaying the generated quiz to the student based on a selected topic.
Preconditions	Student has a registered account and has logged in to the system. Student has uploaded their course book and a roadmap has been generated. Student has selected a topic and the summary of the selected topic has been generated by the system.
Postconditions	The system updates the database and stores the generated quiz.

<table>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Student logs into the system.</td><td></td></tr>
<tr><td>Student clicks on “Books page”.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Books page”.</td></tr>
<tr><td>Student clicks on the roadmap.</td><td></td></tr>
<tr><td></td><td>System redirects the student to the “Roadmap Page”.</td></tr>
<tr><td>Student selects a topic and clicks on the “View Summary” button.</td><td></td></tr>
<tr><td></td><td>System redirects to the summary page.</td></tr>
<tr><td>Student clicks on the “Generate Quiz” button.</td><td></td></tr>
<tr><td></td><td>System redirects to the quiz page.</td></tr>
<tr><td></td><td>System fetches the contents from the database and generates quiz based on the selected topic and description.</td></tr>
<tr><td></td><td>System redirects to the “Take Quiz” page.</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td>In case of any system error or technical issue, the system may not be able to generate and display the quiz.
The student does not remember their login credentials.</td></tr>
<tr><td><b>Frequency of Use</b></td><td>Every time a student selects a topic and clicks on the generate quiz button.</td></tr>
<tr><td><b>Constraints and Special Requirement</b></td><td>None</td></tr>
<tr><td><b>Assumptions</b></td><td>None</td></tr>
<tr><td><b>Notes and Issues</b></td><td>None</td></tr>
</table>

Table 2.9: Generate Quiz

2.1.11 Attempt Quiz

ID	UC010
Name	View Quiz
Scope	Scholarly
Level	User Level
Description	This use case allows the student to view and attempt the generated quiz based on a selected topic.
Primary Actor	Student
Stakeholders and Interests	**Student –** Interested in attempting quizzes of their selected topic. **System –** Interested in displaying quizzes to the student based on their selected topic.
Preconditions	Student has a registered account and has logged in to the system. Student has uploaded their course book and a roadmap has been generated.
Postconditions	System displays the quiz to the student.

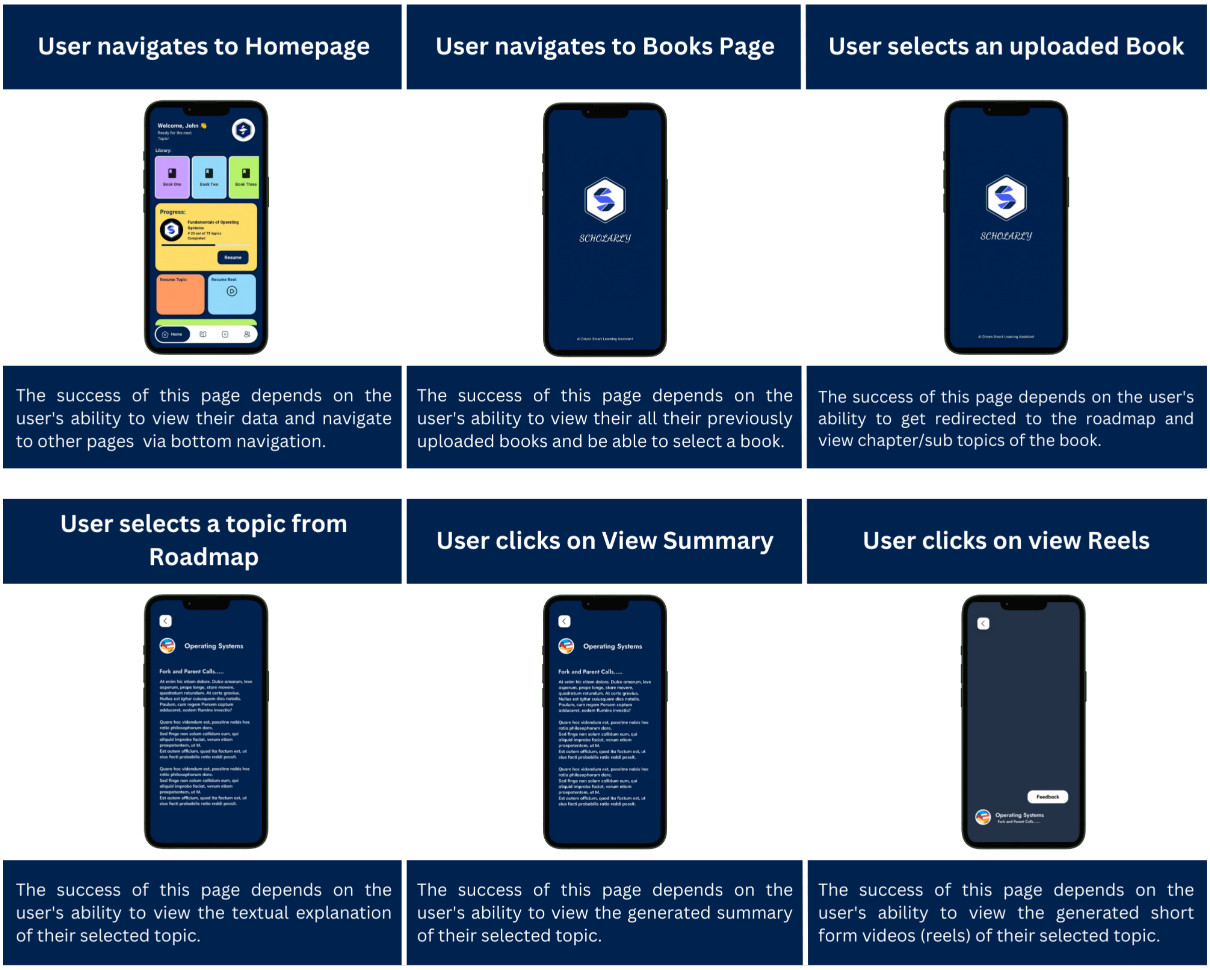
Main Success Scenario	**Actor Action**	**System Response**
	Student enters relevant information to log in.	
		System authenticates the information and grants access to the student.
	Student clicks on “Books page”.	
		System redirects the student to the “Books page”.
	Student clicks on the roadmap.	
		System redirects the student to the “Roadmap Page”.
	Student selects a topic and clicks on the “View Summary” button.	
		System redirects to the summary page.
	Student clicks on the “Take Quiz” button.	
		System fetches the contents from the database and displays the quiz to the student.
Extensions	In case of any system error or technical issue, the student may not be able to view the quiz. The student does not remember their login credentials.	
Frequency of Use	Every time a student clicks on Attempt Quiz button.	
Constraints and Special Requirement	None	
Assumptions	None	
Notes and Issues	None	

Table 2.10: Attempt Quiz

2.2 Storyboarding

Storyboarding can be used to illustrate the sequence of steps that a user takes to navigate throughout the system. The use of storyboards proves to be quite helpful in capturing the user experience and making sure that the design will behave as it should.

Below is an example storyboard for the Scholarly app, showcasing how the user interacts with the system:



User navigates to Homepage

The success of this page depends on the user's ability to view their data and navigate to other pages via bottom navigation.

User navigates to Books Page

The success of this page depends on the user's ability to view their all their previously uploaded books and be able to select a book.

User selects an uploaded Book

The success of this page depends on the user's ability to get redirected to the roadmap and view chapter/sub topics of the book.

User selects a topic from Roadmap

The success of this page depends on the user's ability to view the textual explanation of their selected topic.

User clicks on View Summary

The success of this page depends on the user's ability to view the generated summary of their selected topic.

User clicks on view Reels

The success of this page depends on the user's ability to view the generated short form videos (reels) of their selected topic.

Figure 2.2: Example Storyboard for User Interaction in Scholarly App

Explanation of Storyboard Steps:

1. **User Navigates to Home Page:** When a user logs in and the system redirects the user to the homepage, the uer will be able to view the dashboard consisting of their data and a bottom navigation. The user will be able to navigate the system using the bottom navigation.

2. **User Navigates to Books Page:** When user clicks on Books Page in the bottom navigation, the user is redirected to the Books page and from there the user can view all the previously uploaded book by the user from which the user can select and learn from the selected book.

3. **User selects an uploaded book:** When user selects a book the user is redirected to the roadmap page where the user can view all the chapters, modules and subtopic of their selected book.

4. **User selects a topic from the roadmap:** When user selects a topic from the roadmap the user will be able to view the textual explanation of that topic which is basically the portion that was written in the uploaded book.

5. **User clicks on View Summary:** After selecting a topic from the roadmap the user will be able to generate a summarized explanation of their selected topic to speed up their learning process and effectively understand their selected topic.

6. **User clicks on View Reel** After selecting a topic from the roadmap the user will also be able to generate short form videos (reels) to visualize and engage with the content more effectively.

7. **User clicks on Attempt Quiz** After selecting a topic from the roadmap the user will also be able be generate a quiz to test their knowledge and enhance their learning.

2.3 Functional Requirements

2.3.1 User Management

- FR001: The student should be able to create a new account by providing essential data such as email, username, password, and personal details.
- FR002: The system shall allow students to log in and authenticate them as users.
- FR003: The system shall offer an interface for users to view their profile.
- FR004: The user should be able to update their personal details, including username, date of birth, password etc.
- FR005: The system shall provide a user dashboard.

2.3.2 Roadmap Generation

- FR001: The system shall allow users to upload a book in a pdf format.
- FR002: The user should be able to see a loading screen when waiting for book upload completion.
- FR003: The system shall analyse the uploaded content and clean it.
- FR004: The system shall be able to identify main topics in the book for roadmap generation.
- FR005: The system shall visualize the roadmap.
- FR006: The user should be able to view the roadmap generated from the book.
- FR007: The user shall be able to select a topic from the roadmap.
- FR008: The user should be able to view the progress of their generated roadmap.

2.3.3 Reels Generation

- FR001: The system shall be able to summarize the text in the topic selected by the user.
- FR002: The system shall present the user with an option to select a textual summary or reels.
- FR003: The user should be able to select either a textual summary or reels.

- FR004: The system shall be able to generate short video reels from the topic selected by the user.
- FR005: The user should be able to see generated reels based on the selected topic.
- FR006: The user should be able to provide feedback on the viewed reels.
- FR007: The system shall store the feedback.

2.3.4 Quiz Generation

- FR001: The system shall be able to summarize the text in the topic selected by the user.
- FR002: The system shall present the user with the option to select a textual summary or attempt a quiz.
- FR003: The user should be able to select a textual summary or attempt a quiz.
- FR004: The system shall be able to generate a quiz from the topic selected by the user.
- FR005: The user should be able to attempt the quiz generated based on the selected topic.

2.4 Non-Functional Requirements

2.4.1 Reliability

1. The system should have a Mean Time Between Failures (MTBF) of at least 500 hours during normal usage to ensure minimal interruptions.
2. In case of a failure of a subsystem (e.g., text summarization or reels generation), the entire system should not fail, and fallback mechanisms should be in place to notify the users of a failure/error
3. For a system failure, the system will log necessary errors and critical issues for debugging and bug/error resolution.

2.4.2 Usability

1. **USE-1:** The user interface should be intuitive enough for first-time users to easily navigate the key features with minimal assistance.

2. **USE-2:** The system shall allow users to upload a book in utmost two interactions (e.g., selecting the file and confirming the upload).
3. **USE-3:** The system shall provide progress bars for all lengthy processes such as uploading a book or analyzing content.
4. **USE-4:** The system shall provide an interactive roadmap allowing users to navigate through the roadmap to view topics and sub-topics.
5. **USE-5:** The system shall enable users to recover from errors by providing error messages with clear description.

2.4.3 Performance

- **PER-1:** The system shall allow users to upload a book within 30 seconds, for a size up to 50 MB.
- **PER-2:** The system shall generate a personalized roadmap, within 15 seconds after course book has been successfully uploaded.
- **PER-3:** The user dashboard shall load within 3 seconds after successful user login.
- **PER-4:** The system shall display the topics and subtopics within 3 sec after clicking on the generated roadmap.
- **PER-5:** The system should be able to handle at-least 100 concurrent users without any performance issues.
- **PER-6:** The average response time for any standard operations should not be more than 3 seconds.

2.4.4 Security

- **SEC-1:** The system shall ensure secure authorized login for all registered users.
- **SEC-2:** The system must guarantee data integrity for all users, preventing any unauthorized data access.
- **SEC-3:** The system shall implement security measures that require a minimum of two levels of authentication for accessing sensitive user data.

Chapter 3

System Overview

Scholarly is an AI Driven Smart Learning Assistant designed and developed to boost student's learning process through concise and topic-specific short reels instead of long video lectures. The primary objective of this system is to solve the problem of students being unable to retain focus while learning from traditional textbooks. Scholarly mainly functions by allowing its users to upload a book and automatically generate a roadmap to act as a study plan for the user allowing them to keep a track of their progress as they move forward. The generated roadmap will feature textual summaries, short video reels, and MCQ based quizzes categorized into topics and subtopics based on the uploaded content. The design concept of this application is based on three key principles: user-friendliness, accessibility, and efficiency; mainly focusing on providing a simple yet intuitive experience for the user. The integration of AI algorithms will ensure that the system adapts to the needs of every student to offer personalized roadmaps and recommendations based on user performance and preferences.

3.1 Architectural Design

The Scholarly mobile application is designed using a three-tiered architecture, which includes the Presentation Tier, Application Logic Tier, and Data Tier. This three-tiered architecture ensures that all the functionalities and responsibilities are distributed amongst different tiers, resulting in the development of a scalable, maintainable and efficient mobile application.

1. **Presentation Tier:** The responsibilities of this tier include handling the user interface where students can upload and view content of the mobile application.

2. **Application Logic Tier:** This tier contains the core logic and functionality of the application. It includes modules for user management, content upload and extraction, roadmap generation, textual summaries generation, reels generation, and quiz generation.

3. **Data Tier:** This Data tier of this application handles the storage, retrieval, and data management operations. It includes the database server where all user data, generated roadmaps, generated summarized text, generated quiz, user feedback, and other information will be stored.

3.1.1 Relationships Between Tiers

1. **Presentation Tier and Application Logic Tier:** The Presentation tier containing the mobile app interfaces communicate with the application logic tier via APIs. The interaction between both the tiers allows users to securely log in, view personal information, upload books, generate textual summaries and roadmaps etc.

2. **Application Logic Tier and Data Tier:** The Application tier contains the business logic modules of this application and can interact with the data tier to store and retrieve user data using JSON. This interaction enable the user the efficiently manage and update their data.

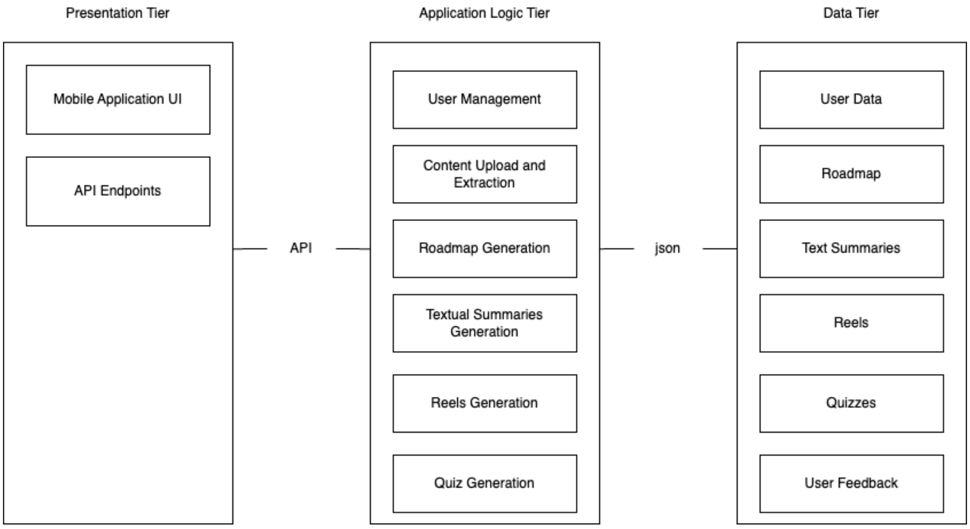


Figure 3.1: Architectural Design

3.2 Design Models

3.2.1 Activity Diagram

3.2.1.1 User Management

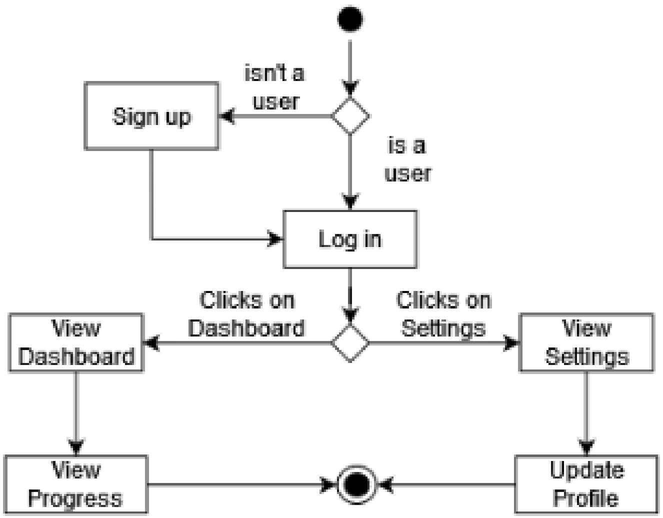


Figure 3.2: Activity Diagram

This activity diagram showcases the user creation and authentication sequence and the user settings and dashboard where if a student is already a user then they can simply log in, otherwise they can sign up and login. Afterward, if they select the dashboard, then they will view their personalized user dashboard and view their progress. If they select the settings, then they will view the settings page and can update their profile as well.

3.2.1.2 Roadmap Generation

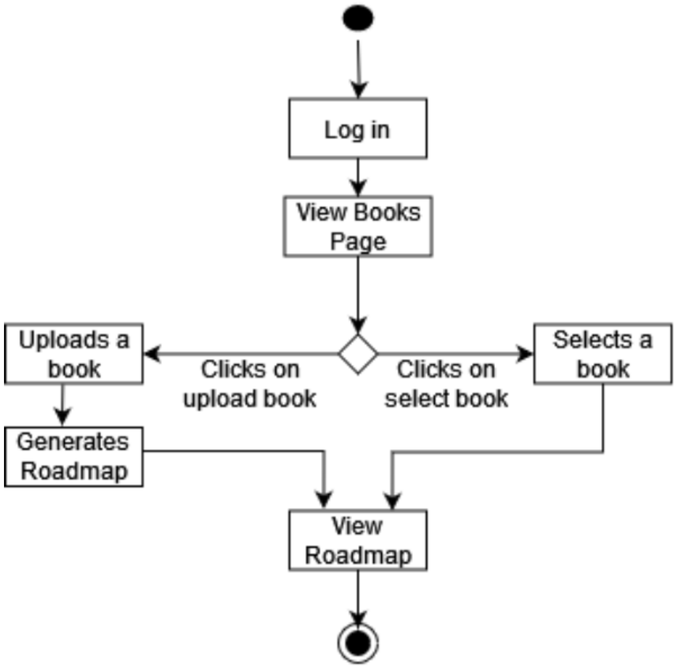


Figure 3.3: Activity Diagram

This activity diagram showcases the upload book and view/generate roadmap sequence. When a user logs in, they can view their books page and have the option to either upload a book or select an already uploaded book. If they select a book, then they will see the upload book page where they will upload a book and then a loading screen where a roadmap is being generated. Lastly, a roadmap will be displayed. If they select an already uploaded book then they will directly view the roadmap.

3.2.1.3 Summary, Reel, and Quiz Generation

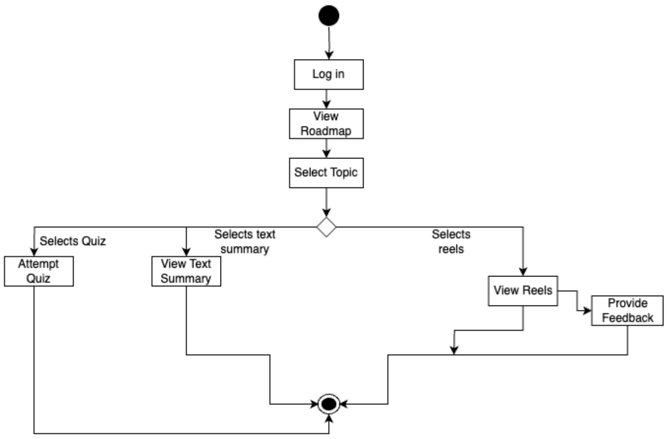


Figure 3.4: Activity Diagram

This activity diagram showcases the summary generation, reel generation, and quiz generation sequence. When a user logs in and has a book uploaded, they can view their generated roadmap. The user then selects a topic and based on the selected topic, they can either choose to view textual summary of the topic, view reels of the topic, or attempt a quiz from the topic.

3.2.2 Class Diagram

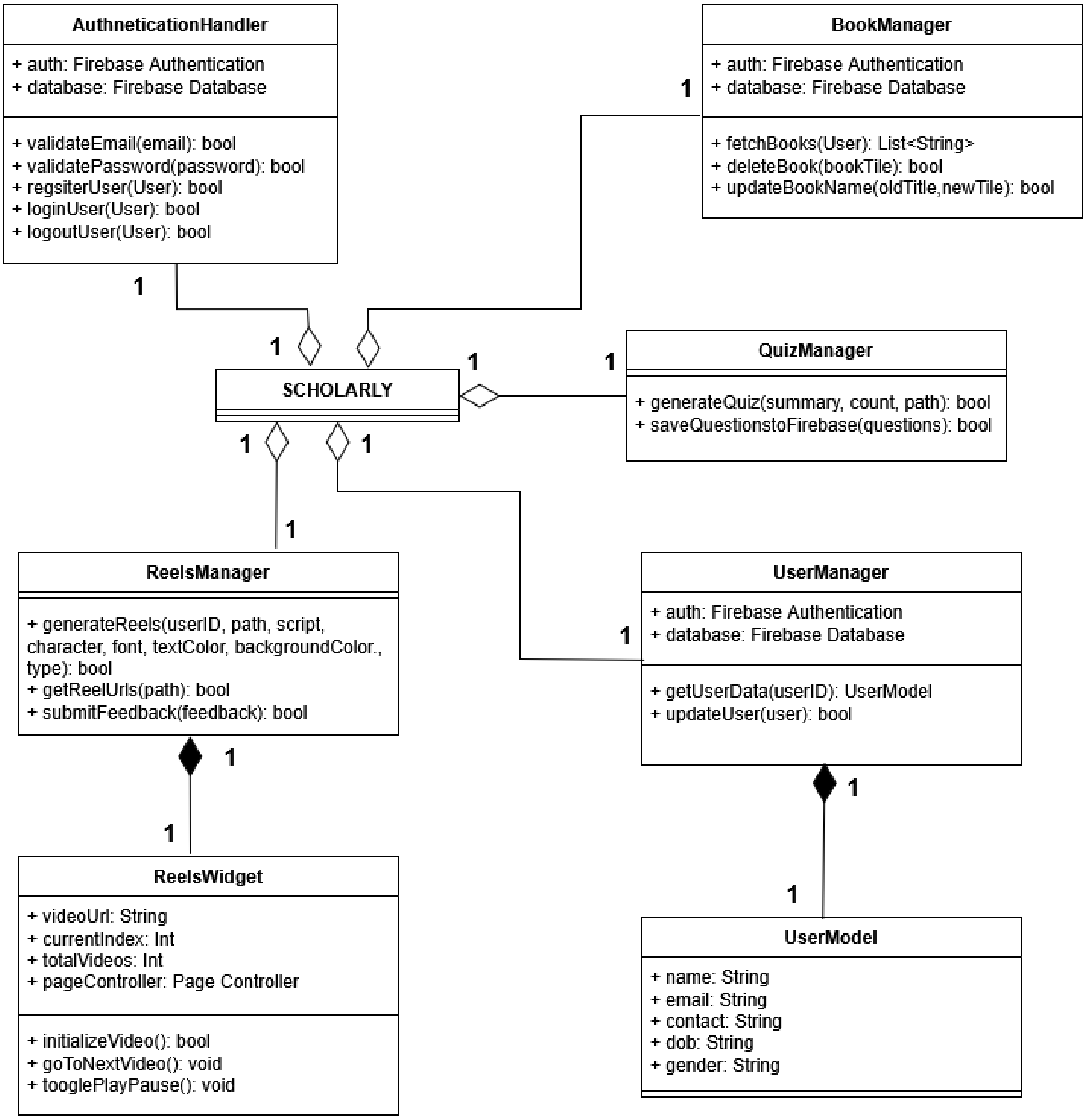

AuthneticationHandler
+ auth: Firebase Authentication
+ database: Firebase Database
+ validateEmail(email): bool
+ validatePassword(password): bool
+ regsiterUser(User): bool
+ loginUser(User): bool
+ logoutUser(User): bool
BookManager
+ auth: Firebase Authentication
+ database: Firebase Database
+ fetchBooks(User): List<String>
+ deleteBook(bookTile): bool
+ updateBookName(oldTitle,newTile): bool
SCHOLARLY
QuizManager
+ generateQuiz(summary, count, path): bool
+ saveQuestionstoFirebase(questions): bool
ReelsManager
+ generateReels(userID, path, script, character, font, textColor, backgroundColor., type): bool
+ getReelUrls(path): bool
+ submitFeedback(feedback): bool
UserManager
+ auth: Firebase Authentication
+ database: Firebase Database
+ getUserData(userID): UserModel
+ updateUser(user): bool
ReelsWidget
+ videoUrl: String
+ currentIndex: Int
+ totalVideos: Int
+ pageController: Page Controller
+ initializeVideo(): bool
+ goToNextVideo(): void
+ tooglePlayPause(): void
UserModel
+ name: String
+ email: String
+ contact: String
+ dob: String
+ gender: String
1
1
1
1
1
1
1
1
1
1
1
1
1
1

Figure 3.5: Class Diagram

The class diagram of the Scholarly app demonstrates how different classes group together to form the business logic of the system; ensuring the complete implementation of the core functionalities for the Scholalry app. The main classes of this class diagram are **Scholarly, User, Authentication Handler, Reels Manager, Quiz Manager, Book Manager, User Model** etc.

- **Scholarly** acts as the main system that coordinates all the actions. It communicates with all the other classes to ensure and manage the core functionality of the app.

- **Authentication Handler** class handles operations related to secure authentication of the user. This class holds the functionality related to register, login and logout user securely from the scholarly application.
- **User Manager** class acts a manager class for the main User and is responsible to maintain, get and update User data.
- **User Model** class stores all the personal information of the user such as name, password, email, contact number, gender etc.
- **Reels Manager** class acts a manager class for the Reels and is responsible for the generations of reels. This class also contains the functionality to get Reels and submit feedback.
- **Reel Widget** is a class that displays the fetched reels to the user. This class contains the Video URLs, current index, total video count etc. Methods of this class include initialize video, go to next video and toggle Pause/Play.
- **Book Manager** class ensure the app's functionality related to book handling operations. This class contains the login to fetch, delete and update user books.
- **Quiz Manager** class is responsible for handling operations related to quiz generation and storing the generated quiz in the database.

3.2.3 Class-level Sequence Diagram

3.2.3.1 User Creation

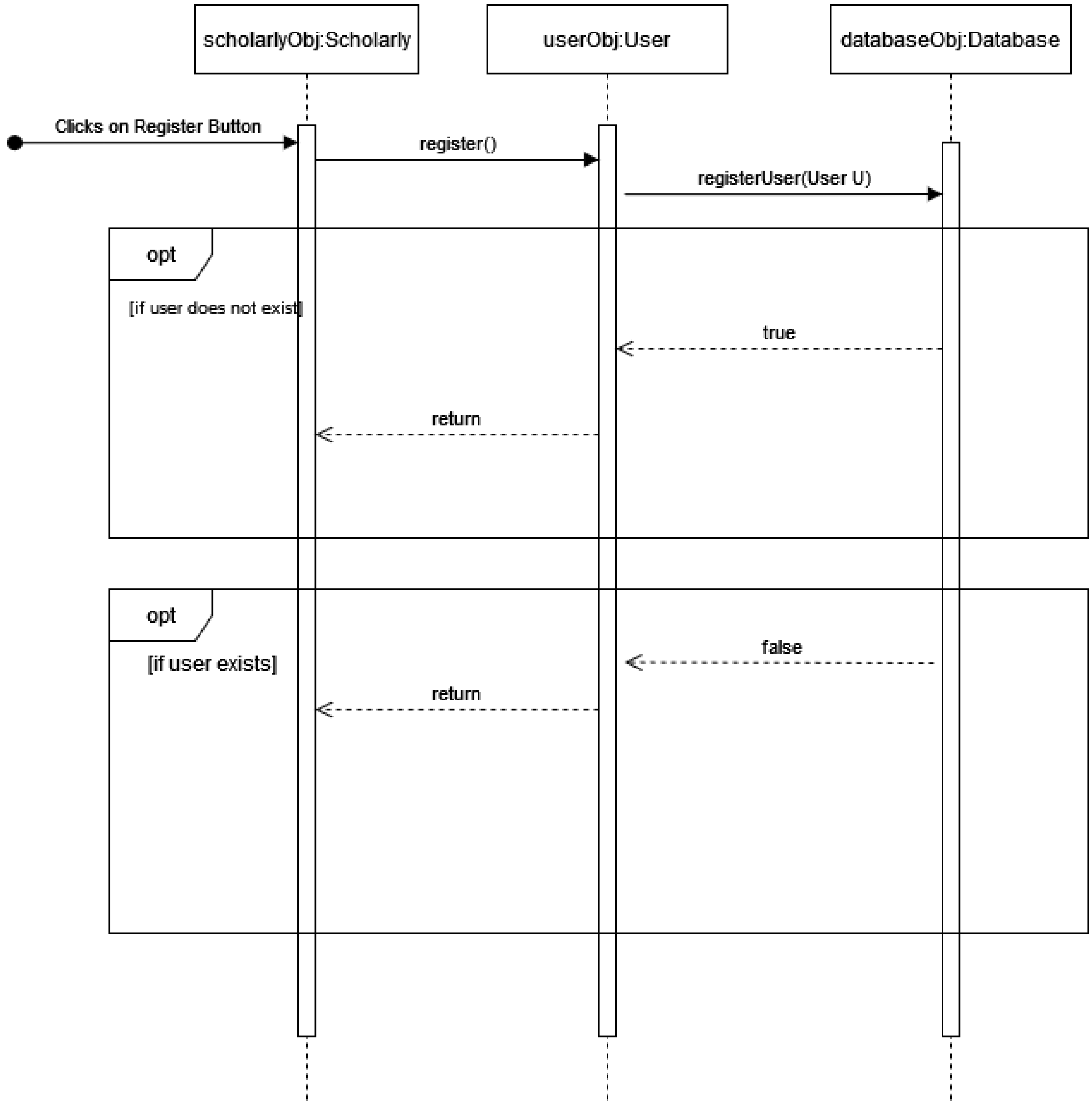


Figure 3.6: SD User Creation

This sequence diagram showcases the registration process. When the user clicks on the register button, the system class; Scholarly, calls the registerUser() function from the User class which requests to add a new user in Database. Given the user information, if the user does not exist then the Database adds a new user and returns true but if a user does exist, the Database returns false.

3.2.3.2 Upload Book

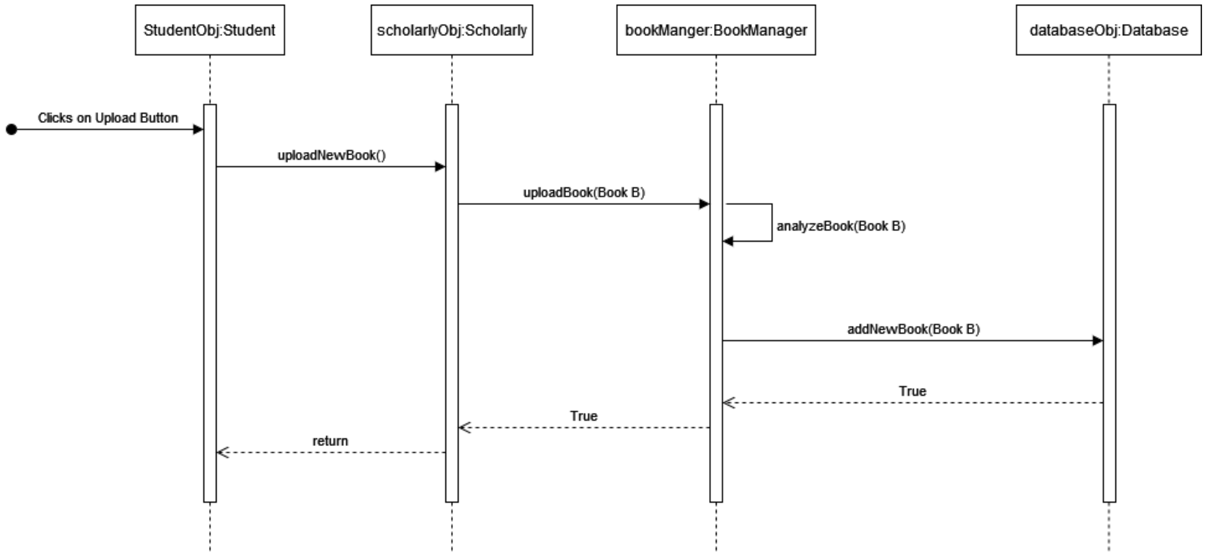


Figure 3.7: SD Upload Book

This sequence diagram showcases the book upload process. When the user clicks on the upload book button, the student class; Student, calls the uploadBook() function in system class; Scholarly. The system class calls a similar function in BookManager class which preprocess the book and adds the book into the Database.

3.2.3.3 Roadmap Generation

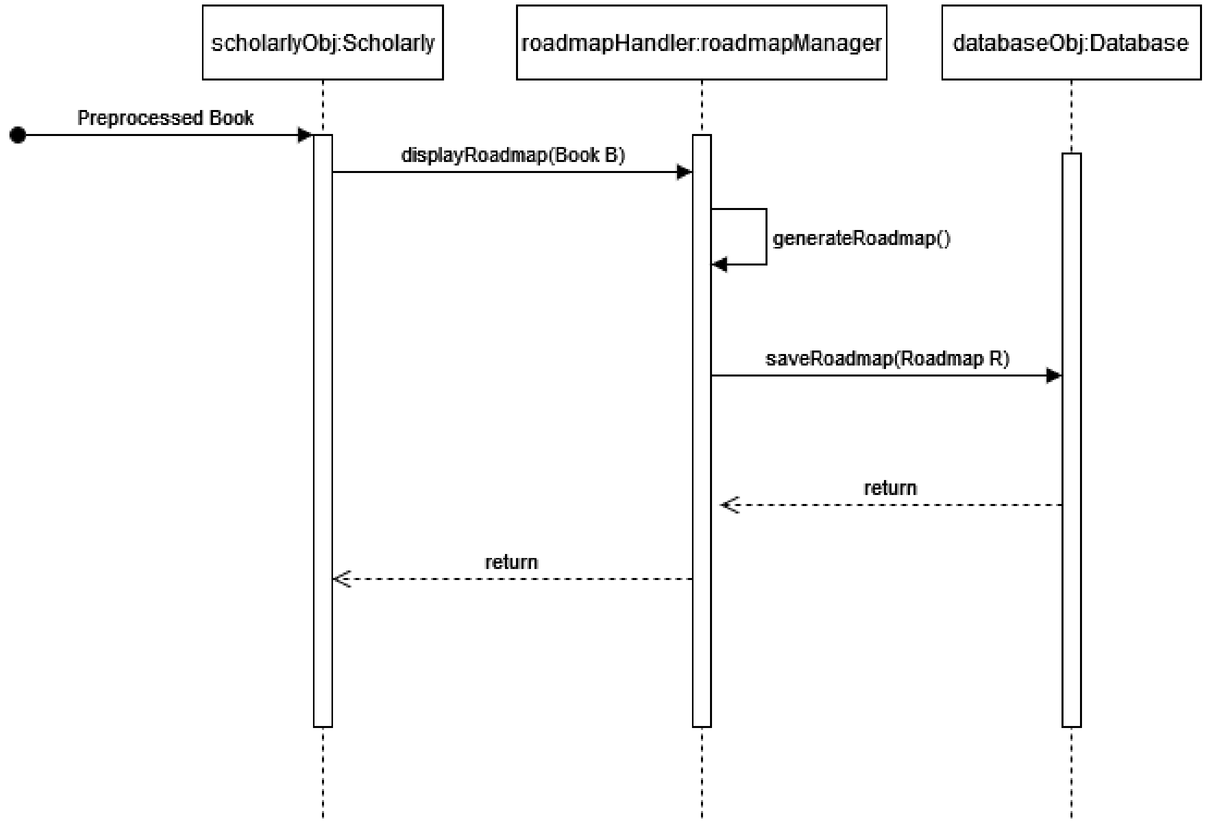


Figure 3.8: SD Roadmap Generation

This sequence diagram showcases the roadmap generation process. When the book is uploaded, the preprocessed book is sent to the system class; Scholarly, which calls the displayRoadmap() function in the roadmapManager class. This class then generates a roadmap and loads it into the Database and returns it to the system class.

3.2.3.4 Generate Topic Summary

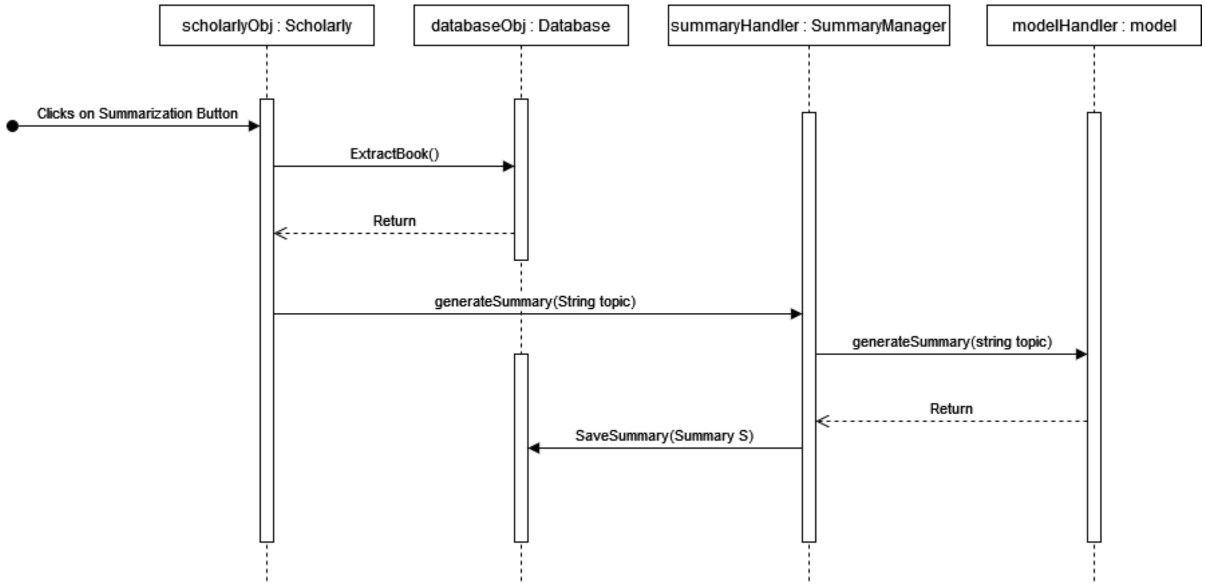

scholarlyObj : Scholarly
databaseObj : Database
summaryHandler : SummaryManager
modelHandler : model
Clicks on Summarization Button
ExtractBook()
Return
generateSummary(String topic)
generateSummary(string topic)
Return
SaveSummary(Summary S)

Figure 3.9: SD Generate Topic Summary

This sequence diagram showcases the text summarization process. When the user clicks on the summerize button, the system class; Scholarly, extracts the book from the Database. It sends this book to the SummaryHandler which interacts with the Model to generate a summary and then saves the summary into the Database.

3.2.3.5 View Topic Summary

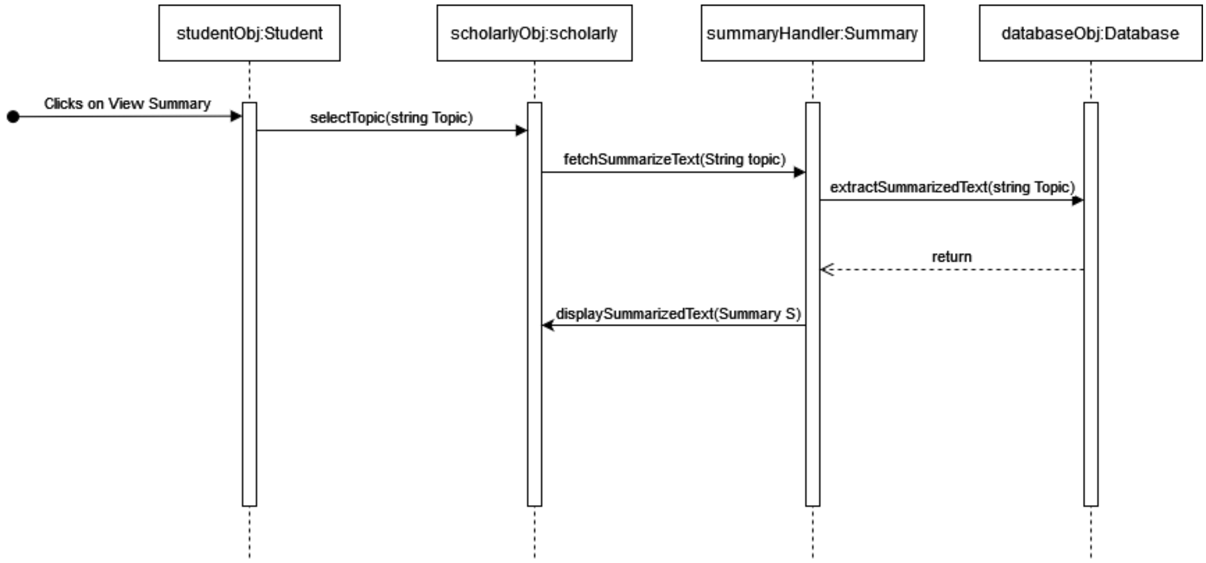

studentObj:Student
scholarlyObj:scholarly
summaryHandler:Summary
databaseObj:Database
Clicks on View Summary
selectTopic(string Topic)
fetchSummarizeText(String topic)
extractSummarizedText(string Topic)
return
displaySummarizedText(Summary S)

Figure 3.10: SD View Topic Summary

This sequence diagram showcases the view summary process. When the user clicks on view summary and they select the topic, the system class; Scholarly, fetches the summarized text from the Summary class. The Summary class extracts the summarized text from the Database and returns it. The system class then displays the summary.

3.2.3.6 Reel generation

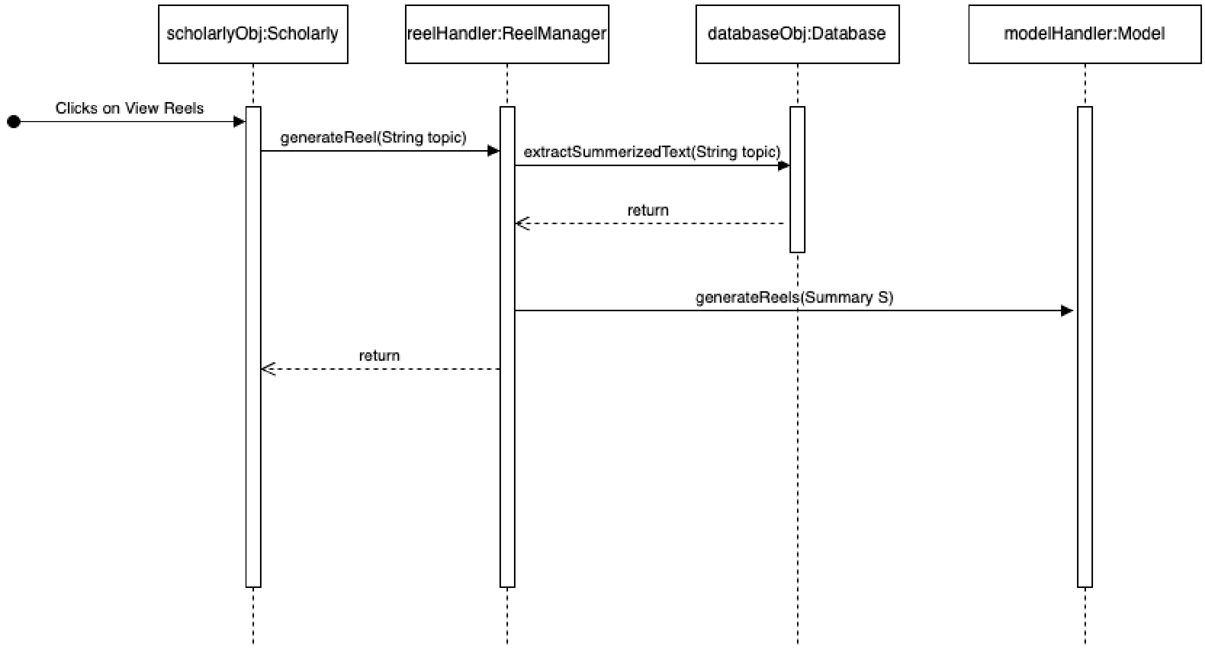


Figure 3.11: SD Generate Reels

This sequence diagram showcases the reel generation process. When the user clicks on the generate reel button, the system class; Scholarly, extracts the summary from the database. It sends this summary to the ReelHandler which interacts with the Model to generate short video reels and then saves them into the Database.

3.2.3.7 View Reels

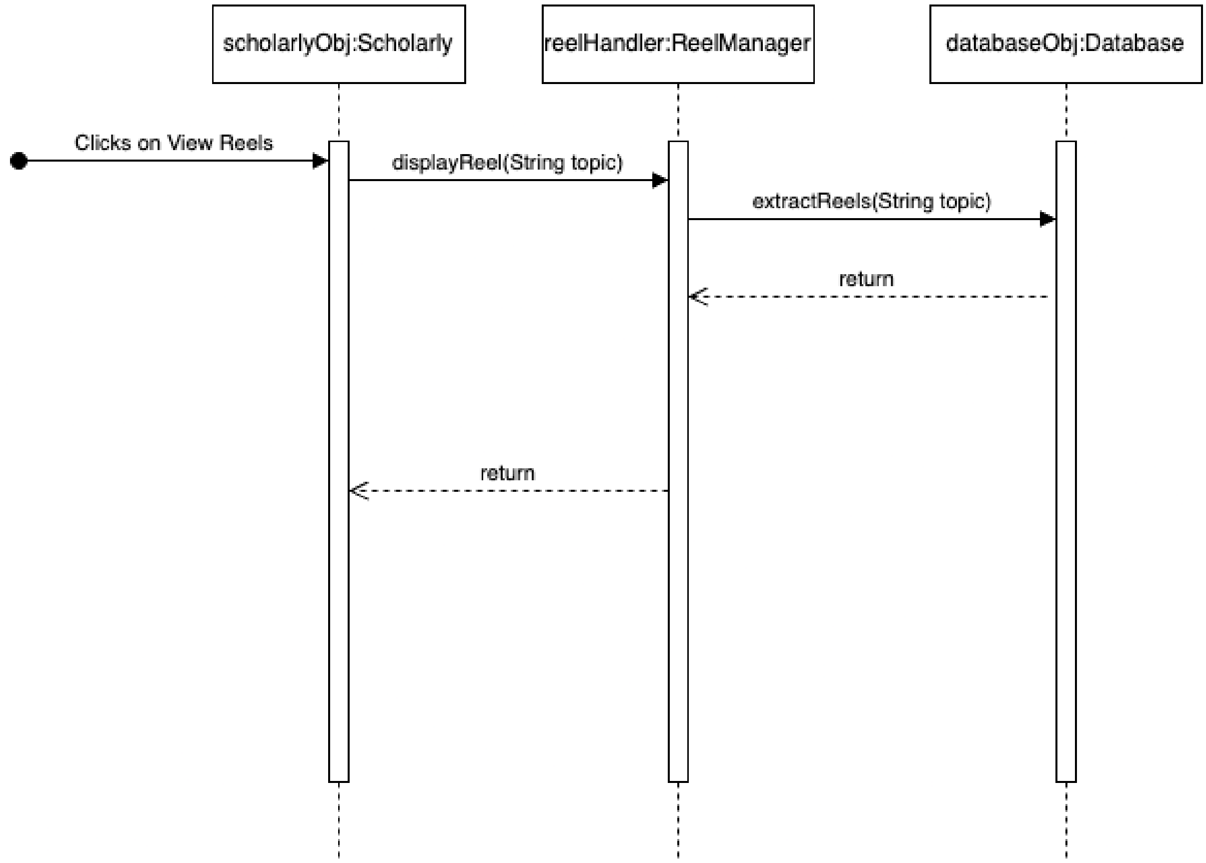


Figure 3.12: SD View Reels

This sequence diagram showcases the view reels process. When the user clicks on view reels after they select the topic, the system class; Scholarly, fetches the short video reels from the Reels class. The Reel class extracts the short video reels from the Database and returns it. The system class then displays the reels.

3.2.3.8 Quiz generation

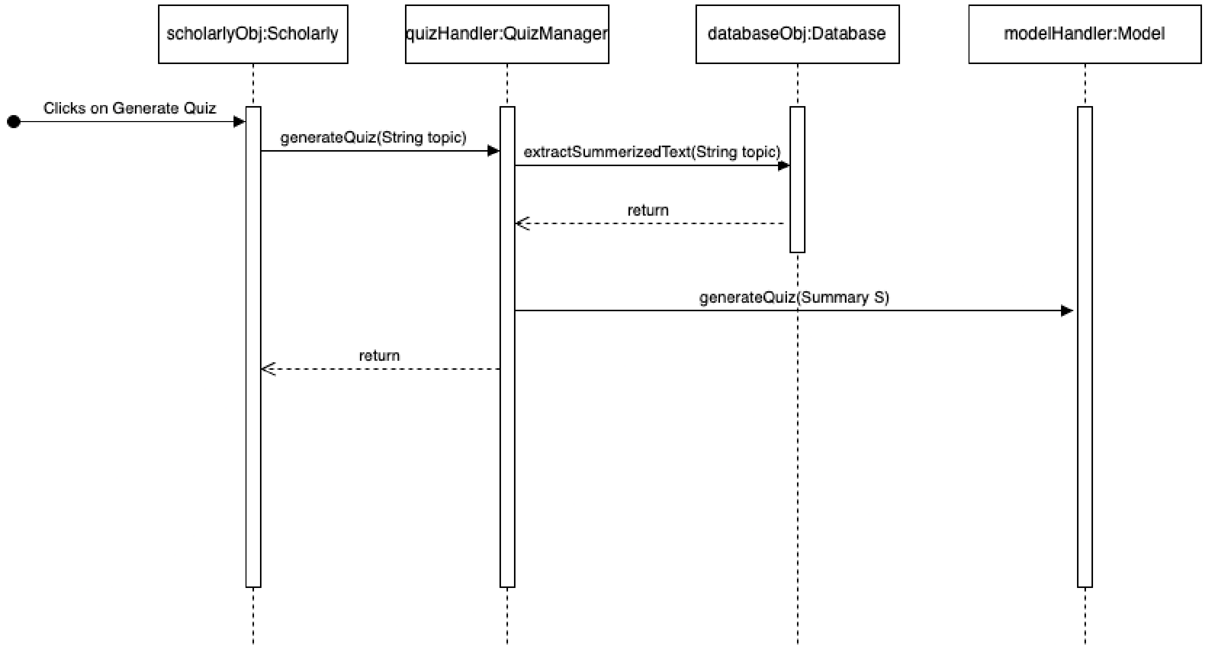


Figure 3.13: SD Generate Quiz

This sequence diagram showcases the quiz generation process. When the user clicks on the generate quiz button, the system class; Scholarly, extracts the summary from the database. It sends this summary to the QuizHandler which interacts with the Model to generate quiz and then saves it into the Database.

3.2.3.9 Attempt Quiz

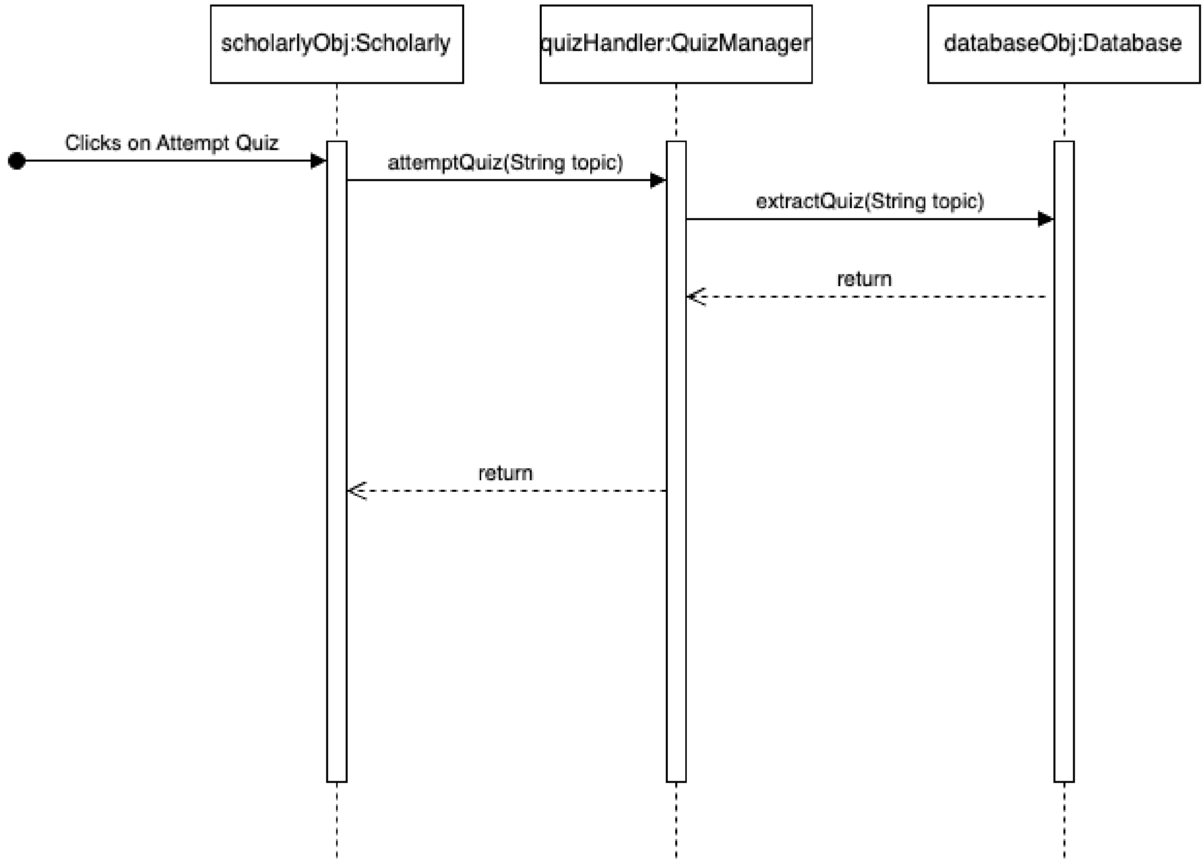


Figure 3.14: SD Attempt Quiz

This sequence diagram showcases the attempt quiz process. When the user clicks on attempt quiz button after they select the topic, the system class; Scholarly, fetches the generated quiz from the Quiz class. The Quiz class extracts the quiz from the Database and returns it. The system class then displays the quiz.

3.2.4 State Transition Diagrams

3.2.4.1 Login

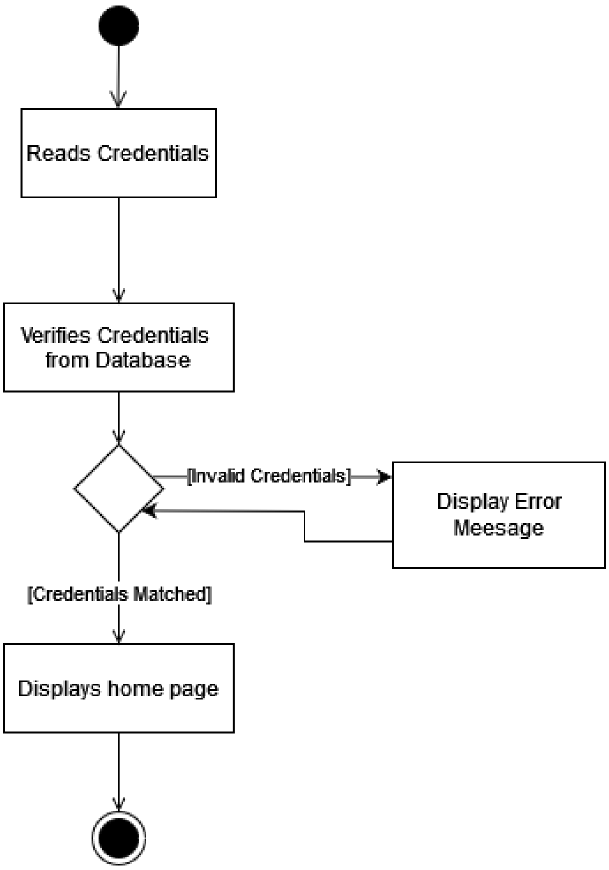


Figure 3.15: State Transition User Login

This state transition diagram showcases the login process. When the user tries to login to the system using their credentials, the system fetches the entered credentials, validates them and if valid then redirects the user to the home page.

3.2.4.2 Upload Book

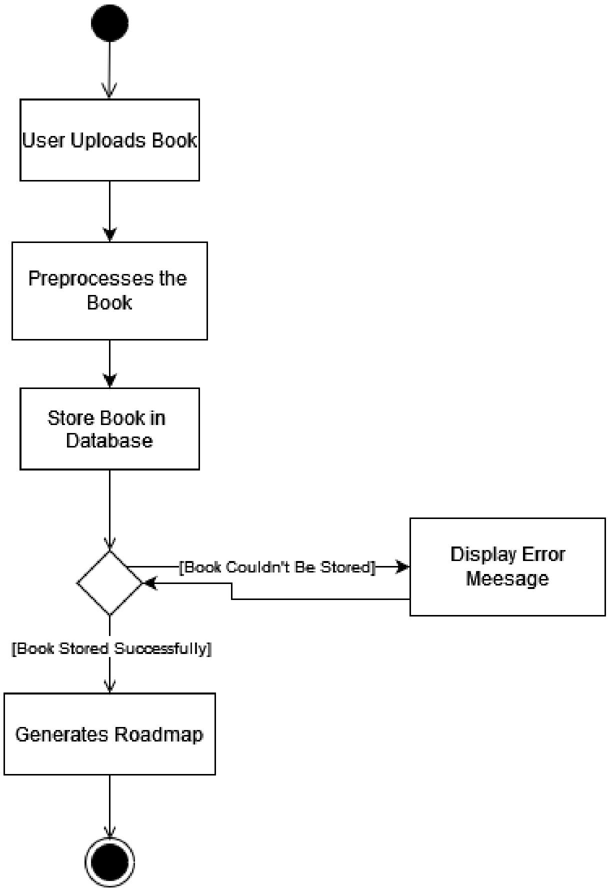


Figure 3.16: State Transition Upload Book

This state transition diagram showcases the book upload process. When the user selects a book and tries to upload it, the system initially preprocesses the book and then stores the book inside the database. If book is successfully stored in the database then the system automatically generates a roadmap based on the uploaded book.

3.2.4.3 Generate Summary

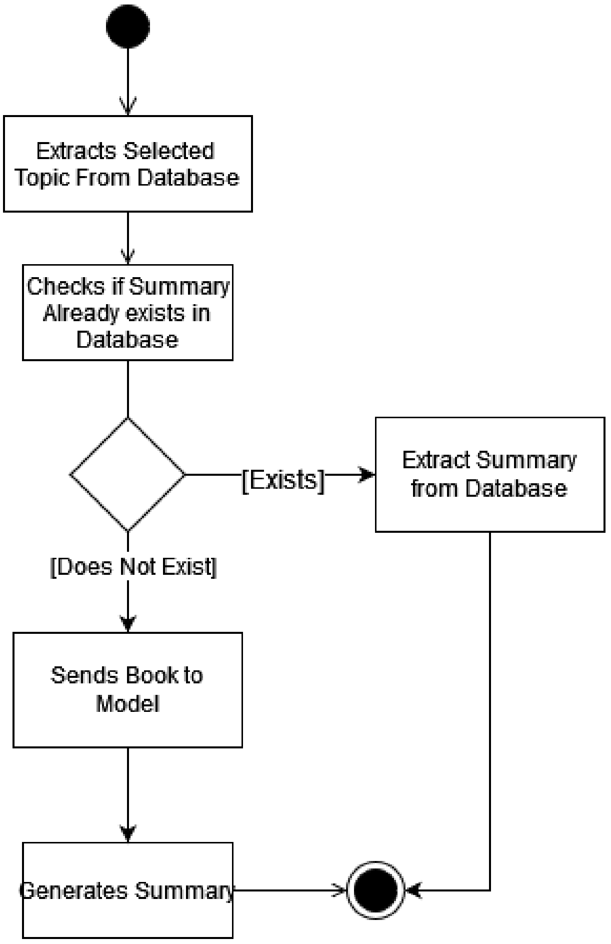

Extracts Selected Topic From Database
Checks if Summary Already exists in Database
[Exists]
Extract Summary from Database
[Does Not Exist]
Sends Book to Model
Generates Summary

Figure 3.17: State Transition Generate Summary

This state transition diagram showcases the summary generation process. When the book has been uploaded by the user and user selects a topic to generate summary, the system extracts the selected topic from the database and check if a summary has been previously generated for the topic or not. If summary has not been previously generated then the system sends the extracted text from the database to the model to generate a textual summary of the selected topic.

3.2.4.4 Generate Reels

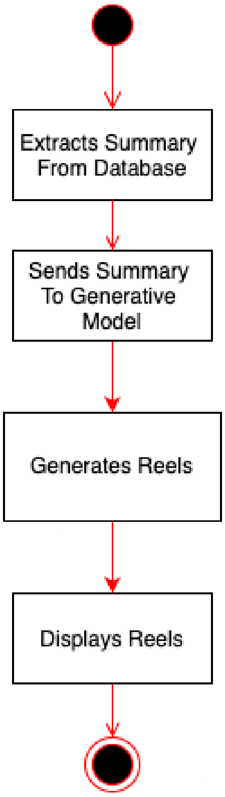

Extracts Summary From Database
Sends Summary To Generative Model
Generates Reels
Displays Reels

Figure 3.18: State Transition Generate Reels

This state transition diagram showcases the reel generation process. When user clicks on generate reels button, the system extracts the summary of the selected topic and sends it to the model to generate reels. The generated reels are then displayed to the user according to the topic selected.

3.2.4.5 Generate Quiz

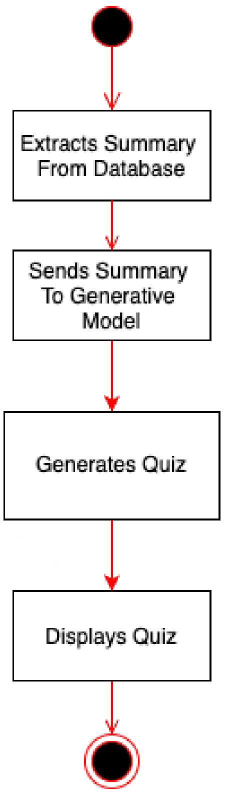

Extracts Summary From Database
Sends Summary To Generative Model
Generates Quiz
Displays Quiz

Figure 3.19: State Transition Generate Quiz

This state transition diagram showcases the quiz generation process. When user clicks on generate quiz button, the system extracts the summary of the selected topic and sends it to the model to generate quiz. The generated quiz is then displayed to the user according to the topic selected and the user can then attempt the quiz.

3.3 Data Design

Note: The above Data Design Diagram is for a No SQL database, that is the reason the diagram does not follow standard Data Design rule. Also, as most objects are stored as Objects in other Firebase services that's why major object classes such as Quiz and Reels are only shown as a variable in the Data Design Diagram.

The data design of the Scholarly app translates the system's information into well-structured data models which are stored and managed using a database system to support core functionalities such as user management, content processing, feedback, summaries, roadmaps, quizzes,

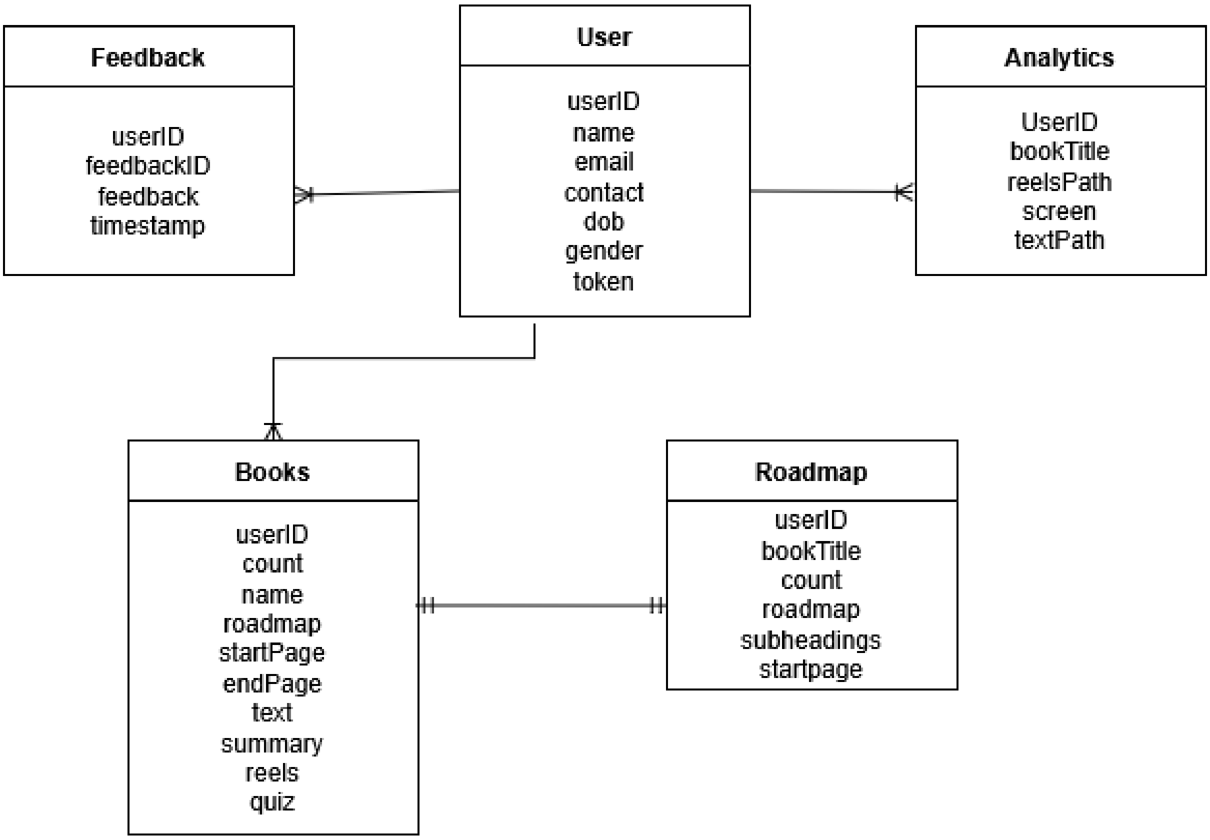


Figure 3.20: Data Design Diagram

and reels.

3.3.1 Data Structures

3.3.1.1 User

- **Attributes:** userID, Name, Email, Contact, DOB, Gender, Token.
- **Description:** Stores essential student identifer and personal information.

3.3.1.2 Books

- **Attributes:** userID, Count, Name, Roadmap, StartPage, EndPage, Text, Summary, Reels,Quiz
- **Description:** Stores book information for a book uploaded by the user.

3.3.1.3 Roadmap

- **Attributes:** userID, BookTile, Count, Roadmap, SubHeadings, StartPage.

- **Description:** Stores the generated roadmap for the each uploaded book.

3.3.1.4 Feedback

- **Attributes:** userID, FeedbackID, Feedback, Timestamp.
- **Description:** Contains the feedback provided by students for a specific reels

3.3.1.5 Analytics

- **Attributes:** userID, BookTitle, ReelsPath, Screen, TextPath.
- **Description:** Contains the analytics regarding the users of the app.

3.4 Domain Model

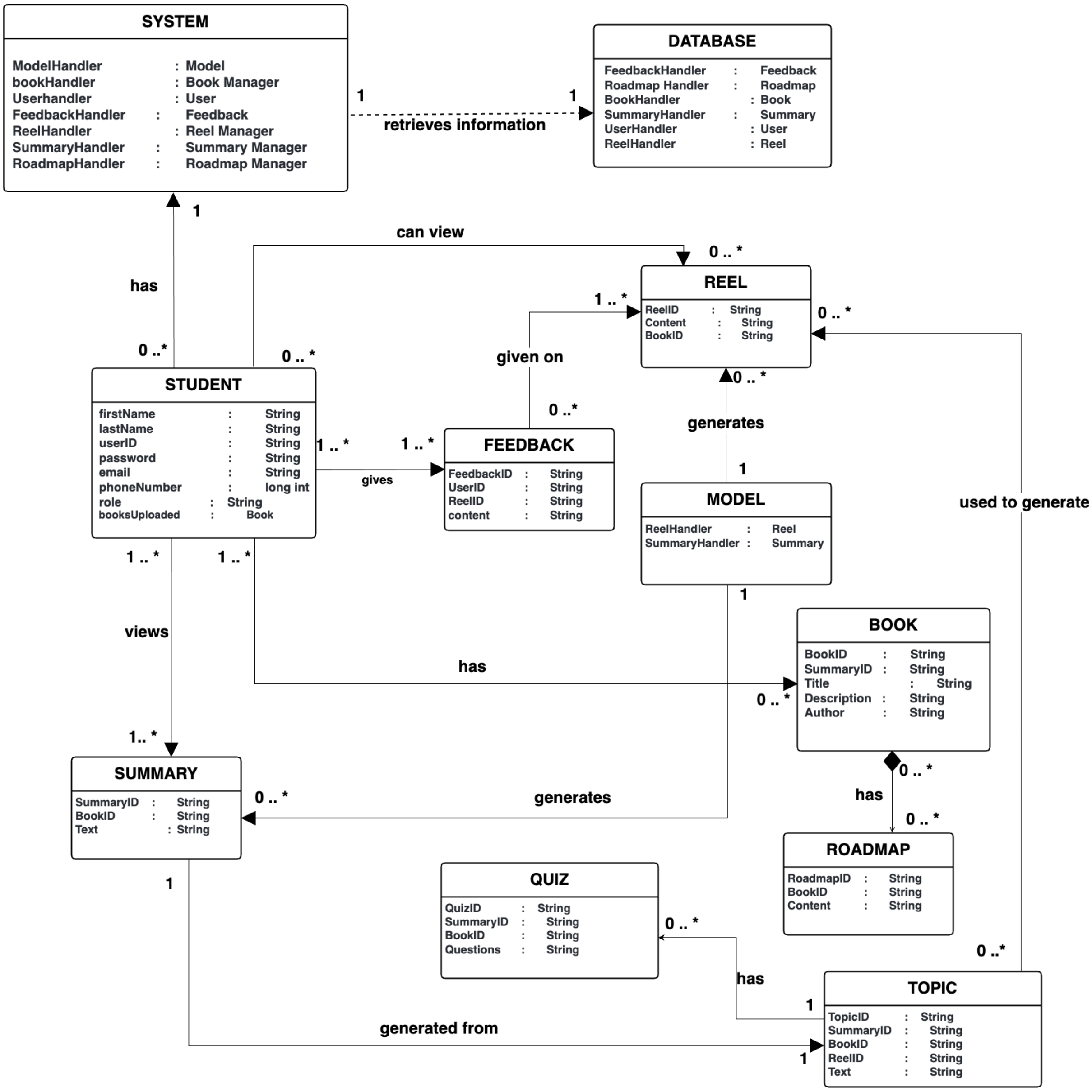


Figure 3.21: Domain Model

Chapter 4

Implementation and Testing

Scholarly is an AI-powered Smart Learning Assistant designed to improve students learning process by replacing the need to study from long textual content or hours of long video lectures with topic-focused short video reels. Scholarly addresses the issue that students face while studying; which is the inability to maintain focus while learning from textbooks. Scholarly facilitates students by allowing them to upload a book from which the system generates a personalized roadmap that acts as a study plan for the student. This roadmap comprises text summaries, short video reels, and quizzes arranged in topics and subtopics obtained from the uploaded content. Scholarly is designed to follow three core principles: friendliness to the user, easy accessibility, and learning efficiency. Scholarly aims to combine traditional study methods with cutting-edge AI technology to make education more interactive, personalized, and efficient.

4.1 Algorithm Design

Below are the algorithms that will be used for Book Upload, Generate Text Summarization, Reel Generation, and Quiz Generation based on a specific topic selected by the user.

4.1.1 Book Upload

Algorithm 1 BookUpload

Input: User, BookFile (PDF format)

Output: Organized TOC and text stored in the database

```
Navigate to the Book Upload screen
User clicks on the "Upload Book" button
if (BookFile is not in PDF format) then
    Display "Invalid file format"
    Exit
else
    Use PyMuPDF to extract Table of Contents (TOC) from BookFile
    if (TOC extraction fails) then
        Display "Failed to extract TOC"
        Exit
    else
        Store TOC in the database
        for (each Chapter, Topic, Subtopic in TOC) do
            Extract corresponding text using PyMuPDF
            Clean and preprocess the extracted text
            Store the processed text in the database
        end for
        Create Book node in the database and store contents
        Display "Book uploaded successfully"
    end if
end if
```

4.1.2 Generate Summary

Algorithm 2 GenerateTextSummary

```
Input: User, Book Name, Chapter, Topic, SubTopic
Output: Summarized text displayed to the user
Navigate to the Books page
User selects a Book
Fetch roadmap for the selected Book from the database
User selects Chapter, then Topic, optionally SubTopic
Fetch text explanation for the selected Topic/SubTopic
User clicks "Generate Summary" button
if (summary exists in the database) then
    Fetch and display summary
    Exit
else
    Initialize ModelHandler
    Send fetched text explanation to the model via ModelHandler
    Receive summarized text from model
    Store summarized text in the database
    Display summarized text to the user
end if
```

4.1.3 Generate Reels

Algorithm 3 GenerateReels

```
Input: User, Book Name, Chapter, Topic, SubTopic,Summary
Output: Short Video Reels displayed to the user
Navigate to the Books page
User selects a Book
Fetch roadmap of the selected Book from the database
User selects Chapter, then Topic, optionally SubTopic
Fetch summary for the selected Topic/SubTopic
User clicks "Generate Reels" button
if (summary exists in the database) then
    Fetch summary, generate reels and display to the user
    Exit
else
    Initialize ModelHandler
    Send fetched text explanation to the model via ModelHandler
    Receive summarized text from model
    Send the summary to the Modelhandler for Reel Generation
    Display generated reels to the user
end if
```

4.1.4 Quiz Generation

Algorithm 4 GenerateQuiz

```
Input: User, Chapter, Topic, SubTopic,Summary
Output: Quiz displayed to the user which they can attempt
Navigate to the Books page
User selects a Book
Fetch roadmap of selected Book from the database
User selects Chapter, then Topic, optionally SubTopic
Fetch summary for the selected Topic/SubTopic
User clicks "Generate Quiz" button
if (summary exists in the database) then
    Fetch summary, generate quiz and display to the user
    Exit
else
    Initialize ModelHandler
    Send fetched text explanation to the model via ModelHandler
    Receive summarized text from model
    Send the summary to the Modelhandler for Quiz Generation
    Display generated quiz to the user
end if
```

4.2 External APIs/SDKs

The Scholarly app uses multiple third-party APIs and SDKs to provide core functionalities to its users. Below are some of the APIs and SDKs that are currently used by Scholarly:

SDK/Libraries	Description	Purpose of Usage	Functions
Firebase Core (v3.8.1)	Core Firebase functionality	Used to Configure and Initializes Firebase Services	Firebase.initializeApp()
Firebase Auth (v5.4.0)	Firebase Authentication for Flutter	To allow users to register, sign in, and manage sessions securely	.createUserWithEmailAndPassword(), signInWithEmailAndPassword()
Firebase Database (v11.3.0)	No SQL Realtime Database	To store and retrieve user data, roadmaps, and other information	FirebaseDatabase.instance .ref(), DatabaseReference.set(), DatabaseReference.get()

Http (v1.2.2)	HTTP client for Dart	To perform network calls to backend Flask APIs for invoking Python functions and comunicating with models	http.post(), http.get(), http.Response
Video Player (v2.9.3)	Flutter video playback SDK	For View Reels Functionality	VideoPlayerController.asset(), VideoPlayerController.network(), VideoPlayerController.play()

Table 4.1: External APIs/SDKs Used in Scholarly App Development

4.3 Testing Details

4.3.1 Unit Testing

Each unit test in the following section has been designed to test a specific function or method independently. These test cases not only ensure a smooth functionality of the mobile application but also help to identify any issues directly related to the tested functionality.

Following are the test cases for Scholarly app functionalities:

Test Case ID	Test Objective	Pre-conditions	Steps	Test Data	Expected Result	Post-Condition	Actual Result	Pass/Fail
TC001	Verify user registration functionality.	User should not have an already existing account.	Click on the "Register" button. Enter name, email, and password, contact, gender, DOB. Clicks on the Register buttom.	Name: Test User Email: test@example.com Password: Password123 Contact: 12345678 Gender: Male DOB: 31 October, 2002	System registers the user in the database and redirects to the Homepage.	User is redirected to the Homepage.	As, Expected	Pass

TC002	Verify user login with email and password.	User must have a registered account.	Click on the Login Menu. Enter valid email and password. Click on the "Login" button.	Email: text@example.com Password: Password123	System logs the user in and redirects to the homepage.	User is redirected to the homepage.	As, Expected	Pass
TC003	Verify viewing user profile.	User must be logged in to the system.	Click on the "Profile" button in the bottom navigation of homepage.	N/A	System redirects the user to their profile screen displaying user details.	User details are displayed on the profile screen.	As, Expected	Pass
TC004	Verify updating the user data.	User must be on the profile screen.	User enters a new username. Click on the "Update Profile" button. User is redirected to the Homepage. User data gets updated.	New username: John Doe	System updates the username and redirects to the Homepage.	The updated username is displayed on the Homepage.	As, Expected	Pass

TC005	Verify book upload functionality.	User must be logged in.	Navigate to the "Upload Book" page. Select a book file in PDF format. Click on the "Upload" button.	Book file: book.pdf	System pre-processes the uploaded book and updates the database	Book is successfully uploaded and the user is redirected to the Books page.	As, Expected	Pass
TC006	Verify roadmap generation for an uploaded book.	User must have uploaded a book.	Wait for the system to pre-process the uploaded book. View the generated roadmap.	Uploaded book: book.pdf	System generates a roadmap containing chapters, modules, and topics.	The roadmap is displayed to the user.	As, Expected	Pass
TC007	Verify view roadmap functionality.	User must have uploaded a book.	Navigate to the "Books" page. Select an uploaded book from the list. System fetches the data from the database. View the generated roadmap.	N/A	System displays a roadmap containing chapters, modules, and topics.	The roadmap is displayed to the user.	As, Expected	Pass

TC008	Verify forgot password functionality.	User must have a registered account with a valid email address.	Click on the "Forgot Password" button. Enter the registered email address. Submit the form. Enter the received OTP/code via email. Set a new password.	Email: user@example.com New password: NewPass123	System resets the password and displays a confirmation message.	The user is able to log in with the new password.	As, Expected	Pass
TC009	Verify viewing text-based content for a selected topic.	User must have a generated roadmap.	Navigate to the roadmap. Select a specific topic from the roadmap. Click on the "View Text" button.	Topic: Chapter 1 Subtopic: Overview	System displays the text-based content related to the selected topic.	Text content for the selected topic is displayed to the user.	As, Expected	Pass
TC010	Verify view topic summary functionality.	A topic must exist in the roadmap.	Navigate to the roadmap page. Select a topic from the roadmap. Click on the "Generate Summary" button.	Topic: Chapter 2 Subtopic: Key Concepts	System generates and displays a summary of the selected topic.	The summary for the selected topic is displayed to the user.	As, Expected	Pass

TC011	Verify view reels functionality.	A topic must exist in the roadmap.	Navigate to the roadmap page. Select a topic from the roadmap. Click on the "Generate Reels" button.	Topic: Chapter 2 Subtopic: Key Concepts	System generates and displays reels of the selected topic.	The reels for the selected topic are displayed to the user.	As, Expected	Pass
TC012	Verify attempt quiz functionality.	A topic must exist in the roadmap.	Navigate to the roadmap page. Select a topic from the roadmap. Click on the "Attempt quiz" button.	Topic: Chapter 2 Subtopic: Key Concepts	System generates and displays a quiz of the selected topic.	The quiz for the selected topic are displayed to the user.	As, Expected	Pass

Table 4.2: Unit Test Cases for Scholarly App

Chapter 5

Conclusions and Future Work

5.1 Conclusion

The main purpose of this project was to develop an innovative educational app that caters to the problem of students being unable to maintain focus while studying for long hours. So, by utilizing modern AI, Scholarly aimed to transform course content from long textual explanation to short and engaging video reels. The current version of this app contains key features such as secure user authentication, book upload, personalized learning roadmap generation, summary generation, short video reel generation, quiz generation and furthermore. These features have been carefully implemented with Firebase services, with a custom backend built using Python and Flask to provide an intelligent and seamless learning experience.

5.2 Future Work

After thorough development and testing, of the current version of Scholarly app, the main purpose behind this project has been achieved. However, few features and improvement that are recommended to be developed in the upcoming versions of this app include:

- Integration of multiple types of questions including True/False and Short Questions
- Dedicated Mentor and Classroom system
- Development of a website version of the app
- Multilingual support

Bibliography

[1] Neil A. Bradbury. Attention span during lectures: 8 seconds, 10 minutes, or more? *Advances in Physiology Education*, 41(4):509–513, 2016.