Scholar-Path Tailored Education for Everyone

MUNTAZIR MEHDI
01-134221-061
NAUMAN SAEED
01-134221-062

Supervisor: Ahtesham Noor

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

October 2025

Certificate

We take the piece of work found in the report entitled Scholar-Path Tailored Education for Everyone is a confirmation note by Muntazir Mehdi and Nauman Saeed to the necessity. standard on the partial completion of degree of Bachelor of Science in Computer.

Approved by . . . :

Supervisor: Ahtesham Noor

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator: Dr. Muhammad Asim

_____

Head of the Department: Dr. Khurram Ehsan

_____

Abstract

Scholar Path is an academic management system that is cross-platform based and designed to streamline regular school activities and optimize individual education. By bringing educators, students, administrators, and the administration together on one platform, the application does away with decentralized or manual operations and allows individual groups to access the resources they require. The system is supported by Firebase Firestore and is built using Flutter, and is secure in storing and updating data in real time, with easy access across various devices.

It allocates varying abilities to every user role on the platform. Administrators can systematize institutional data on students, instructors, classroom formats, and semester data, and see general activity using interactive dashboards. Instructors are able to post course content, automatically create quizzes using AI according to the material, and monitor student performance. Students look through class content, have the opportunity to create their own notes, generate AI-based practice quizzes immediately, control their study activities, and track their performance with a clean, responsive interface.

Scholar Path follows a three-layer architecture that is scalable and enhances efficiency and maintainability. The Cache-First Repository structure guarantees reliable access to data and depends on Dependency Injection, through Provider, to ensure that the components are modular and loosely coupled. It augments interaction between the three most important components: the mobile interface, AI microservice, and cloud database. The system is secure and stable because it consists of these components.

In general, Scholar Path reduces the amount of academic work done manually, eases collaboration, and makes the learning process easier and more interactive through the integration of intelligence in an academic management solution in harmony with present times.

Acknowledgments

We extend our sincere gratitude to Allah Almighty for providing us with the opportunity to embark on this project. We are confident that His help and guidance will pave the way for the successful completion of this endeavour. We are also deeply thankful to our esteemed supervisor, Ahtesham Noor, for his unwavering support, insightful vision, and continuous encouragement throughout the course of this project. His motivation and belief in our abilities have been instrumental in inspiring us to persevere and deliver our best work. Working under his supervision is truly an honour, and we are grateful for the valuable mentorship he has provided. Finally, we express our heartfelt appreciation to our parents for their prayers and unwavering support. Their blessings and encouragement have been a constant source of strength, propelling us forward in this project and beyond.

MUNTAZIR MEHDI
NAUMAN SAEED
Bahria University Islamabad, Pakistan

October 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
UI	User Interface
UX	User Experience
API	Application Programming Interface
DBMS	Database Management System
AWS	Amazon Web Services
NLP	Natural Language Processing
LMS	Learning Management System
FCM	Firebase Cloud Messaging
OOP	Object Oriented Programming
UML	Unified Modeling Language
JSON	JavaScript Object Notation

Chapter 1

Introduction

1.1 Project Background

Education is vital for personal development as well as for establishing professional success. With learning increasingly shifting to digital platforms, students often confront problems in organizing scattered study resources and keeping track of their learning requirements. Most of the online learning systems developed today are rigid in structure and offer little scope for customization. Thus, it is difficult for learners of varying pace, preference, and needs to stay organized and motivated.
Modern learning research insists that learners do better in adaptive environments, whereby the system acts upon progress and gives feedback that is always relevant to them. At the same time, only a handful of systems allow students to self-regulate their own content, provide actionable recommendations, or conduct assessments that adapt to strengths and weaknesses. With current technologies, it is possible to create innovative systems capable of personalizing learning pathways, constructing adaptive assessments, and responding with continually updating feedback.

1.2 Problem Description

The classic e-learning platforms are usually strictly based on one-size-fits-all strategy that provides with little individualization and fixed content that cannot be adjusted to the unique learning rate, style or needs. Learners find it hard to systematize the disorganized study resources, monitor their development, and get prompt and feedback-specific feedback. Teachers do not have effective and evidence-based instruments to trace the development of students and detect unacquired knowledge in a short period of time. Current systems also enable the user to upload and process the personal academic materials (notes, textbooks, lectures) to create the specific assessments, which leads to the discontinuous learning processes, loss of motivation, and poor academic performance.

1.2.1 Lack of Individualized Learning

Traditional learning systems do not usually favor individualized learning. Students struggle to systematize various study materials, monitor their academic performance, and identify areas requiring improvement due to the lack of dynamism in these platforms.

1.2.2 Limited Educator Support

Teachers do not have current and data-driven tools to track student performance. This creates challenges in identifying learning gaps and providing suitable and timely feedback to students.

1.2.3 Institutional Inefficiency

Learning institutions experience difficulty in handling intricate academic processes within a centralized digital space. Current systems lack flexibility, making it challenging to optimize processes and integrate cloud-supported, scalable solutions.

1.2.4 Static and Non-Adaptive Platforms

Existing learning platforms are usually non-adaptive and static in nature. This limits the ability to create individualized learning paths and provide dynamic feedback tailored to individual learner needs.

1.3 Project Objectives

SScholar-Path aims to facilitate self-directed learning through a robust content architecture and AI-enabled individualization. The key objectives of the project are outlined as follows:

1.3.1 Content Management

Goal: Provide a secure, centralized storage system for planning, managing, and searching academic resources.
Steps:

1. Facilitate secure document storage using cloud storage.
2. Apply systematic classification to enable efficient resource retrieval.
3. Offer a clear interface for personal and course-related materials.

1.3.2 Intelligent Assessment & Personalization

Purpose: Use Artificial Intelligence (AI) to automate the development of study assessments and provide individualized learning reinforcement.
Steps:

1. Automatically generate context-aware quizzes on user-uploaded documents with the help of Large Language Models (LLMs).
2. Provide immediate and objective responses to assessments to evaluate student comprehension.
3. Personalize the learning process by focusing on specific knowledge gaps during quiz attempts.

1.3.3 Data-Driven Progress Tracking

Purpose: Implement a longitudinal tracking system that visualizes academic performance through actionable analytics.
Steps:

1. Create real-time dashboards to display mastery scores and attempt history.
2. Present performance trends to help users identify strengths and weaknesses.
3. Provide evidence-based insights to support effective study planning.

1.3.4 Accessibility and Scalability

Goal: Ensure a consistent and user-friendly experience across all user roles and devices.
Steps:

1. Develop a responsive interface for both mobile and web platforms.
2. Implement Role-Based Access Control (RBAC) to support distinct workflows for administrators, students, and teachers.
3. Architect the system to scale efficiently using cloud-native infrastructure.

1.4 Project Scope

Scholar-Path involves the development of an end-to-end, AI-powered Learning Management System (LMS) that is scalable and adaptive. The project focuses on transforming static learning material into an interactive learning environment. The major functional areas and stakeholders of the system are defined as follows:

1.4.1 Personalized Content Management

This feature provides an individualized content management system integrated into the platform. It defines a centralized repository of academic resources that extends beyond simple file storage. Users can access study materials in multiple formats (PDF, DOCX) securely stored in the cloud.
Scope Detail: This module manages metadata association by linking files to specific classes or users and enables efficient retrieval through structured indexing. It transforms disorganized local files into a synchronized cloud-based library accessible to both students and educators.

1.4.2 Intelligent Assessment & AI Generation

This module is the main innovation of the platform, Instead of using pre-built questions banks, the system generates tests dynamically using the input of the user-specific content.
Scope Detail: The FastAI-based microservice is integrated with Large Language Models (LLMs). The system analyzes the uploaded documents to identify important concepts and creates context-sensitive Multiple Choice Questions (MCQs). This offers real-time, objective feedback to the learner, confirming their comprehension of the particular material they are studying.

1.4.3 Data-Driven Progress Monitoring

The project will have a full suite of an analytics engine that will visualize academic performance over time.
Scope Detail: The system is able to track each attempt of a quiz and compute a cumulative Mastery Score for each student. To teachers, it gives aggregated data to find out the class-wide knowledge gaps. This shifts assessment from a periodic event to a continuous and data-driven feedback loop represented in real-time interactive dashboards.

1.4.4 Scalable Multi-User Architecture

The platform will be scaled to accommodate a hierarchy of academic roles that have varying privileges, providing security and integrity of workflow.
Scope Detail: Role-Based Access Control (RBAC) is used in the system to distinguish among Administrators (system management), Teachers (content creation), and Students (content consumption). In addition, a cross-platform Flutter frontend secures uniformity of functionality across a wide variety of hardware, including mobile gadgets and web browsers.

1.5 Out of Scope

The following should be removed to ensure that the main goals of automation and personalization are kept in mind. The current version does not include the following features:

1. **Manual Grading:** It is concentrated on objective, automated evaluation (MCQs) to maximize efficiency.
2. **Multi-Language Support:** The NLP models are now adjusted to English-language academic data, making them highly accurate in question generation.
3. **Offline AI Processing:** LLMs need processing power to conduct quizzes, which is why they cannot run offline. Quiz generation presupposes an active internet connection and on-device local AI inference is not currently supported.

1.6 Feasibility Study

1.6.1 Risks Involved

1.6.1.1 Technical Risks

- **LLM Dependency:** The system is dependent on third-party AI providers like OpenAI and Gemini to generate quizzes. This brings with it reliance on outside availability, price adjustments, and API rate limits.
- **Data Synchronization:** Control of real-time synchronization in thousands of parallel students may be difficult. Optimized data and proper indexing of Firestore structure designs have to be such that they provide quick and stable synchronization.

1.6.1.2 Non-Technical Risks

- **Dissuaders to Adoption:** There are obstacles that may be observed on the part of some educators and institutions to adobt transitioning paper-based workflows to a digital learning management system.
- **Content Quality:** The quality of AI-generated quizzes will greatly rely on the clarity and format of the documents posted by the users.

1.6.2 Resource Requirements

1.6.2.1 Development Environment

Installed workstations with Android Studio or VS Code and mobile emulators. At least 16GB RAM is required to make the development easy.

1.6.2.2 Cloud Infrastructure

A Firebase application with a Blaze plan of the Cloud Functions and Cloud Storage. The Python environment (like Google Cloud Run or AWS EC2) must be hosted, deploying the Fastapi-based AI microservice.

1.6.3 Team Roles and Responsibilities

- **Frontend Engineer:** Builds Flutter responsive interfaces, executes the MVVM architecture, and controls application state with Provider.
- **Backend/Cloud Engineer:** Security rule configuration of Firebase, Firestore management. indexing, and creates the FastAPI-based AI microservice.
- **QA and Integration Specialist:** Unit Testing, API Integration verification, and provides cross-platform compatibility between Android and iOS.

1.7 Solution Application Areas

- **Academic Institutions:** Academic Institution used to digitize assessment, distribute learning content and academic workflow.
- **Corporate Training:** Allows corporations to store training materials and create quizzes automatically to conduct onboarding and compliance.
- **Self-Directed Learners:** Allows the individuals to post notes or study material to create customized quizzes on practice and monitor their learning.

1.8 Tools and Technology

1.8.1 Hardware

- **Development:** Core i5/i7 computers that have at least 16GB of RAM to operate Android. and iOS emulators are also efficient.
- **Proposed Deployment Architecture:** The proposed deployment architecture is supposed to be deployed on a cloud native architecture with. Google Cloud Platform, which does not require physical servers.

1.8.2 Software Stack

- **Frontend:** Cross-platform mobile and web development, Flutter (Dart).

- **Backend (BaaS):** Firebase Authentication, Firestore (NoSQL database), and Firebase Storage.
- **AI Microservice:** Python with FastAPI, which exploits LangChain and the text processing and quiz generation APIs using OpenAI.
- **Development Tools:** GitHub as the version control tool, PI can be tested using Postman, and Database and application management using firebase console.

Chapter 2

Literature Review

The chapter is a review of the available research, tools, and technologies, which are the basis of Scholar-Path. It talks about how the e-learning platforms have evolved, how they have come to use adaptive learning systems, quiz generation methods and recommendation models in the educational setting. The gaps in the existing systems are also identified in the chapter. It describes how Scholar-Path tries to overcome these constraints by being more flexible, learner-focused design.

2.1 Overview

The shift to the digital learning environments has essentially transformed the manner in which the students gain access to the educational resources. Platforms such as Coursera [1], Udemy, Kahoot!, and Khan Academy [2] gives quality education by accessible on video lessons, designed courses, and online tests. These platforms have made education democratic because learners in various geographical areas, socioeconomic backgrounds, and skill levels can access the education by high-quality materials that used to be confined to physical classrooms or prestigious schools. The convenience of self-directed learning, along with multimedia information, e.g. training videos,interactive activities, and discussion board has greatly improved the level of interaction and understanding among various students [3].
In spite of these achievements, most of the platforms continue using a one-size-fits-all approach, where the same information is given to every learner irrespective of their background, previous knowledge,or preferred learning style. Though organized courses are consistent, they tend to fail to take into consideration the personal variations that may be present in the student,e.g., the speed at which he/she may comprehend. This constraint may cause disengagement, frustration, or poor learning to the students who might need it. Moreover, the platforms do offer assessment instruments such as quizzes or assignments. They tend to be pre-built and constructed and provide little information to a learner about

his or her developing. Feedback that is tailored to the individual is very limited and the learners have little control in the creation of content that is specifically relevant to the particular learning materials, including personal notes, institutional textbooks or lecture outlines. This lack of flexibility tends to lock out learners in trying to capitalise their individual study approaches or bringing their own materials in the learning process.
Scholar-Path tries to fill this void by providing individual content management, dynamic quiz. generation, and customized learning paths, permitting them to advance in a manner that best suits their personal needs and learning styles. Through processing user supplied resources and creating quizzes, recommendations, and studying schedules according to real-time performance data, Scholar-Path will make sure that every learner will have a dynamic, learner-oriented experience. [4]. This is not only effective in enhancing understanding and retention but at the same time it promotes self directed learning.

2.2 E-Learning Platforms

The online learning platforms have become important in the contemporary learning environment. Coursera and Udemy offer professional crafted courses in an extremely broad variety of topics. [1]. On the other hand Khan Academy concentrates on free academic work underpinned by interactive learning. [2]. Despite the fact that these platforms have millions of users, the main role of these platforms is to serve. Individualization is low and the learners are usually unable to adapt the system to their own study objectives, which has been identified as a weakness in various research works on individualised systems and adaptive learning.
Scholar-Path improves this idea by allowing the learners to cooperate with their own study materials. Users post notes, documents and textbooks that are processed by the system. It generates the quizzes and offers guidance on study depending on the precise information that the student must know. This brings about a more pertinent, effective and individualized learning experience.

2.3 Adaptive Learning Systems

The solution to the drawbacks of the conventional one directional models of teaching was the development of the adaptive learning technologies Carnegie Learning and Smart Sparrow are some of the platforms that vary the levels of difficulty based on the learner performance by giving them different levels of difficulty based on their progress. Although these systems are a form of expression of equal significance, their flexibility can be frequently associated with pre-existing collections of materials, or at least programmed courses, impeding the degree of genuine personalization.
Scholar-Path expands on these concepts with the addition of flexibility on student-created

content. The system appraises performance in regards to automatically generated quizzes and analyzes the results to distinguish between those aspects of learners that need additional practice. This enables the learners to devote their time on the weaker concepts which will eventually build their understanding by way of constant feedback and self-directed mastery.

2.4 Quiz Creation Techniques

Tests are very crucial in solidifying ideas, understanding and gauging the understanding,leading the learners to excellence. Conventional resources like Quizlet enable one to manually develop flashcards and tests, which provides an opportunity in the structure of the learning materials and review [5]. Nevertheless, such tools usually demand a lot of practical work on the part of students or teachers to get important points, create questions, and align the content with the targeted learning outcomes. Consequently, this usually determines the quality of assessments depending on the time, experience, and acquaintance of the user with the content of the matter at hand.
Contemporary learning systems are trying to streamline this procedure, but even now a lot of them are still extremely dependent on premeditated libraries of content or predetermined question banks. Such methods restrict individualization and are not always representative of the study materials that the students come across on a regular basis, e.g. lecture notes, handouts, presentations or textbook passages.
Scholar-Path handles such limitations, the AI-based quiz generation process is introduced as a new tool that directly depends on the uploaded study materials by the learner. Using natural language processing, the system determines important concepts, gathers learning outcomes, and produces significant practice questions, such as MCQs. This reduces the manual burden learners and educators, and allows them to give more attention to interpretation and getting instead of formulating questions.
Scholar-Path facilitates students by transforming learning materials into organized tests in the active process of working with the material, monitoring their development and identifying those areas that need improvement further review. The adaptive feedback loop of the system assists the students to have a clearer sense of their strong and weak points, and causing a more profound, systematic, and extensive effective learning. This strategy helps not only improve the learning process but also makes sure that the evaluation is in close relation with what students are actually studying.

2.5 Recommendation Models in Education

Recommendation systems are highly embedded on online platforms where Netflix suggests movies and Amazon recommends products based on user behavior information. However,

this has not been the case in the education sector where the use of personalized techniques of recommendations has been slower. Most of the existing e-learning platforms continue to offer generic content recommendations that do not take into account the performance trends and engagement track record or the particular area of weaknesses of learners. This causes learners to get suggestions that are usually not in tandem with their development or learning requirements.

The latest technology in AI has stimulated the integration of personalization into educational sites based on data. Coursera, Udemy and LinkedIn Learning have adapted AI models to provide users with skill and course recommendations, as well as learning paths, according to their interactions and performance indicators. In the same manner, the adaptive learning studies are characterized by the focus on the content customization in terms of mastery, background knowledge, and the ongoing assessment outcomes.

Scholar-Path takes it a step further and realizes these ideas by introducing a recommendation model that is more academically optimized. The system compares test scores of quizzes, study behavior and subject-based performance to provide customized advice. In case a learner has difficulties in working with a certain chapter, Scholar-Path will propose specific reading resources, more practice questions, or adapted study timetables. This strategy is consistent with the educational analytics study, where the effectiveness of adaptive feedback and targeted content in enhancing learning results has been noted.

2.6 Analysis of Existing Systems

2.6.1 Coursera

Coursera has both structured courses that include the work of professionals, and in most cases, graded assignments, peer-reviewed projects, and certificates of completion. This platform will offer the learners access to the quality academic material offered in major universities around the world giving them the opportunities to follow formal learning programs on a variety of subjects. It also involves discussion forums, interaction with instructors, and quizzes in order to reinforce concepts.

Limitations: The course structure of Coursera is highly inflexible, which does not provide a lot of flexibility to learners who would need to move at their own speed or customize the course content to their needs. The assessments are pre-established and do not frequently conform to personal strengths and learning deficiencies. Moreover, the students are not able to add their own resources, including their personal notes or institution-related textbooks, to the learning process.

Scholar-Path Improvement: To overcome these drawbacks, Scholar-Path has the option to enable users to upload their own study data including notes or PDFs, which get processed

by the system to create quizzes and individual study suggestions. This guarantees a more personalised and context-oriented learning.

2.6.2 Khan Academy

Khan Academy provides free educational material in disciplines such as mathematics, science and humanities. The platform offers educational videos, practices, and simple progress tracking, which assist learners in developing their initial knowledge at their own speed [2].

Limitations: Despite a well-organized curriculum in Khan Academy, it is not very personalized. Workouts do not comprehensively fit the individual performance beyond the mastery cues, and the application cannot create personalized tests basing on the student-posted materials. The users are also not allowed to make their own quizzes based on their own notes using the complete personalization features.

Scholar-Path Improvement: Scholar-Path builds a better learning experience using the analysis of user performance and creating adaptive quizzes and suggestions based on the uploads of the particular student. The system concentrates on the weak points and offers a customized practice, and this guarantees a more specific and personalized learning process.

2.6.3 Quizlet

Quizlet enables the learner to hand create flashcards, practice quizzes, and study games. The platform encourages active recall and spaced repetition and allows the learners to create study set with peers to share learning materials [5].

Weaknesses: Quizlet is manual in nature and thus requires users to take a lot of time to develop or formatting study content. Questions are subject to the skill of the user, and therefore questions are not always equally good. Moreover, its adaptive learning capabilities are restricted and fail to give profound insights into conceptual flaws or correspondence with extensive learning materials like textbook books or lecture notes.

Scholar-Path Improvement: Scholar-Path does not require manual creation of questions as it will automatically create meaningful quizzes based on uploaded documents. The analytics engine it uses gives continuous feedback and adjusts the future quizzes to improve the performance of the learners with time.

2.6.4 Kahoot!

Kahoot! represents a popular learning system in the game format that can be used to facilitate interactive learning in the classroom. Teachers are able to develop interactive quizzes which students can take part in at the same time thus making the learning process enjoyable and competitive.

Drawbacks: Kahoot! is not a one-on-one learning program although it has a high engagement value as it is mostly a real-time classroom assessment tool. Question sets are either created by hand or chosen in public libraries and may not be in line with particular needs of a study. It is oriented towards speed, which could lead to memorizations instead of profound knowledge, and it does not provide long-term monitoring of progress.
Scholar-Path Improvement: Scholar-Path removes these limitations by automatically creating quizzes based on individual study materials. As opposed to Kahoot!, it encourages personalized learning by offering adaptive question difficulty, performance feedback, and personalized study suggestions to ensure meaningful learning within the context of more than classroom activity in the short term.

Table 2.1: Comparison of Existing E-Learning Systems and Scholar-Path

Platform	**Key Features**	**Limitations**	**Scholar-Path Improvement**
Coursera	Structured courses, assignments, certificates	Rigid content, limited personalization	User-uploaded content, adaptive quizzes
Khan Academy	Free lessons, videos, exercises	Uniform content, minimal adaptation	Personalized paths, targeted exercises
Quizlet	Flashcards, practice quizzes, collaborative	Manual creation, limited adaptation	Auto quiz generation, adaptive learning
Kahoot!	Game-based quizzes, engagement	One-to-many, focus on recall	Personalized quizzes, progress tracking

2.7 Summary

The current platforms are proven to be very advanced in digital learning. They are outstanding in the content delivery, interaction, and accessibility, as was observed in platforms like Coursera [1], Khan Academy [2], Quizlet [5], and Kahoot!. However, Nevertheless, even with these developments, a majority of platforms are still not really personalized and flexible. Adaptive learning systems and AI-based tools have appeared to facilitate learning, and they are usually restricted to pre-defined sets of data or involve a lot of manual intervention by users. Similarly, there are automatic question generation methods, but most of them are based on text corpora and do not rely on the specifics of the learner. These issues are solved by Scholar-Path, which consists of individualized processing of study-material, adaptive assessment tools, recommendations based on AI, and a convenient interface. Through these features, Scholar-Path offers a very personalized and accommodating learning platform, which will essentially close the gaps that are still present in the current educational technologies.

Chapter 3

Requirement Specifications

3.1 Existing System

Although quite functional in terms of the simple administrative work, traditional academic management and e-learning systems are usually characterized by the serious constraints, which obstruct the student interaction and the efficiency of the working process. The majority of available platforms have a one-size-fits-all approach, whereby the same static content is presented to all learners in the same manner disregarding their learning style, prior knowledge, or individual pace.
The major weaknesses of existing systems are:

1. **Absence of Personalization:** The traditional Learning Management Systems (LMS) are mainly file repositories. They are not intelligent enough to adjust to the performance of a student and do not suggest particular remedial measures or other study directions to the weak points.

2. **Manual labor on the side of Educators:** The teachers are overwhelmed with the manual labor of development of quiz and tests. Transforming lecture notes to objective questions is time consuming and has the tendency of being subject to human error, which decreases the amount of time educators have on actual mentorship.

3. **Fragmentation of Content:** Students must go through several fragmented interfaces, one to see the documents, one to do the quizzes, and one to do tracking grades. Such discontinuity interferes with learning and complicates tracking the progress in a comprehensive manner.

4. **Static Assessment:** Existing systems have a not-evolving quiz which is usually static. After a quiz is set it is permanent unlike dynamic systems which may generate new questions depending on the concrete situation of uploaded study material.

Such inefficiencies give rise to a passive learning system in which students are mere consumers of knowledge as opposed to emergency learners and teachers are bombarded with administrative overhead.

3.2 Proposed System

Scholar Path is suggested as a new generation, AI-based Academic Management System planned to mitigate the limitations of the conventional platforms via automation and personalization. A combination of a Flutter-built cross-platform frontend and a strong backend is integrated.The system is built using firebase backend and Python-based AI microservices, which build a unified system. smart learning environment.
The innovations presented in the proposed system include:

- **AI-Based Content Generation:** Compared to the conventional systems, the Scholar Path is automated the assessment process. The system makes use of Large Language Models (LLMs) to de-parse uploaded study material (PDFs, DOCX) and automatically create context-aware quizzes. This minimizes significantly the work of teachers and students are able to produce their self-study evaluations are immediately evaluated.
- **Centralized & Role-Based Architecture:** The system consolidates all the academic processes into one application.
    1. **Admins**do institutional data and user providing.
    2. **Teachers** control classes and track compiled performance data.
    3. **Students** get individualized dashboard, study planners and materials under a single unit.
- **Data-Based Progress Tracking:** The proposed system will not rely on the outdated gradebooks and adopts live Mastery Scores. It is a longitudinal performance tracking tool which visualizes student performance and enables learners to understand their strengths and weaknesses dynamically.
- **Scalable Infrastructure:** A Cache-First-based infrastructure built upon. The system has Hive and Firestore where data is available even when the network is in poor condition. The serverless backend enables scalability to support with ease.Thousands of parallel users and no loss of performance.

In short, Scholar Path transforms the traditional model of consumption of passive content to individualized and active learning, using modern software engineering practices to build an individualized and personalized learning environment.

3.3 Requirement Specifications

The chapter enumerates the non- functional and the functional requirements of the Scholar-Path platform. The system has three major roles Student, Educator, and Admin and offers content upload and management, computer generated quizzes with the help of AI.

3.3.1 Functional Requirements

3.3.1.1 Authentication and User Management

- FR-01: Student Registration
- FR-02: Educator and Admin Provisioning
- FR-03: Role-Based Login
- FR-04: Password Management
- FR-05: Logout

3.3.1.2 Academic Structure Management (Admin Only)

- FR-06: Semester Management
- FR-07: Class Management
- FR-08: Student Enrollment

3.3.1.3 Content Management Module

- FR-09: Material Upload
- FR-10: Material Viewing
- FR-11: Class Resources

3.3.1.4 AI Assessment & Quiz Module

- FR-12: AI Quiz Generation (Teacher)
- FR-13: AI Quiz Generation (Student)
- FR-14: Quiz Attempt
- FR-15: Quiz History

3.3.1.5 Progress Tracking & Analytics

- FR-16: Student Dashboard Analytics
- FR-17: Teacher Performance Monitoring
- FR-18: Real-Time Notification Feed

3.3.1.6 Task Planner (Student)

- FR-19: Study Planner

3.3.1.7 System Settings

- FR-20: Profile Management
- FR-21: Theme Personalization

3.3.2 Non-Functional Requirements

3.3.3 Performance Requirements

- **NFR-01: Response Time (Read Operations)** Key screens (Dashboards, Lists) shall return less than 200 ms with an cache-first data retrieval model with Hive preceding network calls.
- **NFR-02: AI Processing Latency** The FastAPI microservice will be able to process uploaded allot 15 seconds to files less than 5 MB documents and turn in a quiz.
- **NFR-03: App Launch Time** The application will be launched to the Login or the Dashboard screen in less than 3 seconds on mid-range machines.

3.3.4 Reliability and Availability Requirements

- **NFR-04: Offline Capability** Previously loaded material (Classes, Materials, Quiz Results) will be offline available via local persistence.
- **NFR-05: Service Availability** The system will be based on the infrastructure of Firebase to register a minimum of 99.9
- **NFR-06: Data Consistency** The system will ensure eventual consistency between local caches and Firestore using background synchronization systems.

3.3.5 Security Requirements

- **NFR-07: Data Encryption** Communications between the Flutter client and Firebase application, and the FastAPI server is going to be secured with TLS/HTTPS.
- **NFR-08: Access Control** Role-Based Access Control (RBAC) based access control will be implemented both in Firestore security rules and routing logic to avoid unauthorized access of data.
- **NFR-09: Authentication Integrity** The system will be based on Firebase only Authentication ID Tokens to verify the identity of the user; no passwords will be stored outside Firebase Auth.

3.3.6 Scalability Requirements

- **NFR-10: Horizontal Scalability** The stateless Serverless Firebase backend and Automatic scaling will be supported by FastAPI service to meet the peak usage times.
- **NFR-11: Storage Scalability** Cloud storage shall be able to support uploads with a limit of 50 MB nfile without harming Firestore performance.

3.3.7 Maintainability and Portability Requirements

- **NFR-12: Code Modularity** The codebase will be based on the MVVM architecture with Dependency Injection in order to be maintainable and testable.
- **NFR-13: Cross-Platform Compatibility** At least 95 percent of the codebase will be flutter crossplatform Android and iOS.
- **NFR-14: Theming Support** The system will provide dynamic switching between Light and Dark, preferences are stored locally.

3.3.8 Usability Requirements

- **NFR-15: User Feedback** TThe system should have visual feedback (e.g., loading, indicators, snackbars) 500 ms after any user action.
- **NFR-16: Interface Consistency** All interfaces shall be grouped with Material Design 3 instructions to give a homogenous and user-intuitive experience.

3.4 Use Cases

3.4.1 Use Case Diagram

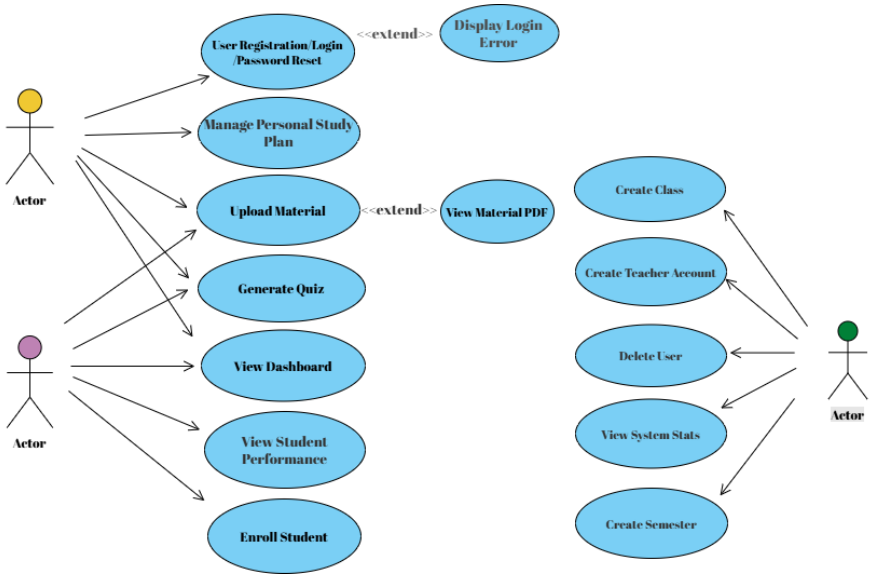

Figure 3.1: Use Case Diagram

This use case diagram represents an educational management system in which the students are in control of the study plans and quizzes, and the teachers are in control of the students in terms of the student enrollment and tracking of student performance. To facilitate the smooth running of the system, an administrator is provided to undertake high level activities such as account creation, semester organization as well as maintenance of the system.

3.4.2 Tabular Description of Use Cases

3.4.2.1 Student Registration

Table 3.1: Student Registration

Use Case ID	UC1
Use Case Name	Student Registration
Actors	Student
Preconditions	User must have a valid email.
Main Flow	1. Student opens the application and chooses Create Account. 2. Student will be asked to include name, email, password and profile details. 3. System verifies and generates the student account. 4. System takes the student to the dashboard.
Postconditions	Student account successfully created.
Exceptions	Invalid email, weak password, network error.

3.4.2.2 Instructor Registration

Table 3.2: Instructor Registration

Use Case ID	UC2
Use Case Name	Instructor Registration
Actors	Instructor
Preconditions	Instructor must provide valid institution information.
Main Flow	1. Instructor clicks on Register as Instructor. 2. Instructor enters verification information and profile. 3. System authenticates identities and generates the account. 4. System sends confirmation and redirects to dashboard.
Postconditions	Instructor account activated.
Exceptions	Invalid data, missing credentials, network error.

3.4.2.3 User Login

Table 3.3: User Login

Use Case ID	UC3
Use Case Name	User Login
Actors	Student, Instructor, Admin
Preconditions	User must have a registered account.
Main Flow	1. User enters email and password. 2. System verifies credentials. 3. System grants access and shows the dashboard.
Postconditions	User logged in successfully.
Exceptions	Incorrect password, unregistered email, account disabled.

3.4.2.4 View Enrolled Classes

Table 3.4: View Enrolled Classes

Use Case ID	UC4
Use Case Name	View Enrolled Classes
Actors	Student
Preconditions	Student must be logged in.
Main Flow	1. Student opens the Classes tab. 2. System fetches enrolled classes. 3. System displays class list with schedule and instructor info.
Postconditions	Student views list successfully.
Exceptions	No classes found, network failure.

3.4.2.5 Create Class

Table 3.5: Instructor Creates Class

Use Case ID	UC5
Use Case Name	Create Class
Actors	Instructor
Preconditions	Instructor must be authenticated.
Main Flow	1. Instructor clicks “Create Class”. 2. Instructor enters class name, semester, and description. 3. System saves the class to Firestore. 4. Class becomes visible to students.
Postconditions	New class created.
Exceptions	Missing data, Firestore write error.

3.4.2.6 Instructor Dashboard Access

Table 3.6: Instructor Dashboard Access

Use Case ID	UC-07
Use Case Name	Instructor Dashboard Access
Actors	Instructor
Preconditions	Instructor must be logged in.
Main Flow	1. Instructor opens the application menu. 2. Instructor selects the dashboard option. 3. System displays the dashboard with class stats, materials, and student progress.
Postconditions	Dashboard data displayed successfully.
Exceptions	Network error, data retrieval failure.

3.4.2.7 Upload Study Material

Table 3.7: Upload Study Material

Use Case ID	UC-08
Use Case Name	Upload Study Material
Actors	Instructor
Preconditions	Instructor must be logged in and assigned to a class.
Main Flow	1. Instructor selects a class. 2. Instructor chooses "Upload Material." 3. Instructor selects PDF or file from device. 4. System uploads file and stores metadata.
Postconditions	Material successfully uploaded.
Exceptions	File upload failure, unsupported file format, network error.

3.4.2.8 Generate AI-Based Quiz

Table 3.8: Generate AI-Based Quiz

Use Case ID	UC-09
Use Case Name	Generate AI-Based Quiz
Actors	Instructor, AI Microservice
Preconditions	PDF material must already be uploaded.
Main Flow	1. Instructor selects material. 2. Instructor clicks "Generate Quiz." 3. System sends material to AI microservice. 4. AI generates quiz questions. 5. System saves quiz in the database and associates it with the class.
Postconditions	Quiz stored and available for students.
Exceptions	AI service unresponsive, invalid file content, network failure.

3.4.2.9 Publish Quiz

Table 3.9: Publish Quiz to Students

Use Case ID	UC-10
Use Case Name	Publish Quiz to Students
Actors	Instructor
Preconditions	Quiz must exist in the system.
Main Flow	1. Instructor selects a quiz. 2. Instructor clicks "Publish." 3. System notifies students and makes quiz visible to them.
Postconditions	Students can now attempt the quiz.
Exceptions	Quiz not found, network issue.

3.4.2.10 Attempt Quiz

Table 3.10: Attempt Quiz

Use Case ID	UC-11
Use Case Name	Attempt Quiz
Actors	Student
Preconditions	Quiz must be published and visible to the student.
Main Flow	1. Student opens the quiz. 2. Student answers each question. 3. Student submits the quiz.
Postconditions	Quiz submission recorded and stored.
Exceptions	Network failure, quiz expired, submission error.

3.4.2.11 View Quiz Results

Table 3.11: View Quiz Results

Use Case ID	UC-12
Use Case Name	View Quiz Results
Actors	Student, Instructor
Preconditions	Quiz must have been submitted.
Main Flow	1. Student or instructor opens quiz results page. 2. System retrieves score and answer details. 3. System displays correct/incorrect responses.
Postconditions	Results displayed successfully.
Exceptions	Results not found, database read error.

3.4.2.12 Track Student Progress

Table 3.12: Track Student Progress

Use Case ID	UC-13
Use Case Name	Track Student Progress
Actors	Student, Instructor
Preconditions	System must have stored quiz results and progress data.
Main Flow	1. User opens progress tracking module. 2. System aggregates quiz history, material usage, and progress metrics. 3. Progress charts and analytics are displayed.
Postconditions	Progress insights shown successfully.
Exceptions	Data retrieval issues, insufficient historical data.

3.4.2.13 Manage Class Information

Table 3.13: Manage Class Information

Use Case ID	UC-14
Use Case Name	Manage Class Information
Actors	Instructor, Admin
Preconditions	User must be authenticated and authorized.
Main Flow	1. User opens class management page. 2. User views list of classes. 3. User updates class details (name, semester, assigned teacher). 4. System saves changes.
Postconditions	Updated class information stored.
Exceptions	Invalid data, unauthorized access, network failure.

3.4.2.14 Admin Manage Users

Table 3.14: Admin Manage Users

Use Case ID	UC-15
Use Case Name	Admin Manage Users
Actors	Admin
Preconditions	Admin must be logged in.
Main Flow	1. Admin opens user management panel. 2. Admin views list of users. 3. Admin will create, update or delete user accounts. 4. Changes are validated and saved in system.
Postconditions	User accounts updated successfully.
Exceptions	Validation errors, unauthorized operation, data sync failure.

3.4.2.15 Admin Manage Classes

Table 3.15: Admin Manage Classes

Use Case ID	UC-16
Use Case Name	Admin Manage Classes
Actors	Admin
Preconditions	Admin must be authenticated.
Main Flow	1. Admin opens class management page. 2. Admin views all classes. 3. Admin creates, edits, or deletes classes. 4. System updates Firestore records accordingly.
Postconditions	Class data updated successfully.
Exceptions	Invalid class data, unauthorized access, network issues.

3.4.2.16 Admin Manage Semesters

Table 3.16: Admin Manage Semesters

Use Case ID	UC-17
Use Case Name	Admin Manage Semesters
Actors	Admin
Preconditions	Admin must be logged in.
Main Flow	1. Admin opens semester management section. 2. Admin creates, modifies, or deletes semesters. 3. System stores or updates semester information.
Postconditions	Semester information updated.
Exceptions	Data validation failure, network error.

3.4.2.17 Admin View Platform Metrics

Table 3.17: Admin View Platform Metrics

Use Case ID	UC-18
Use Case Name	Admin View Platform Metrics
Actors	Admin
Preconditions	Admin must be authenticated.
Main Flow	1. Admin navigates to analytics dashboard. 2. System retrieves metrics (user count, quizzes, materials, classes). 3. Dashboard visualizes statistics.
Postconditions	Metrics displayed successfully.
Exceptions	Data retrieval issues, insufficient permissions.

3.4.2.18 AI Recommendation of Study Materials

Table 3.18: AI Recommendation of Study Materials

Use Case ID	UC-19
Use Case Name	AI Recommendation of Study Materials
Actors	AI Microservice, Student
Preconditions	Student must have quiz history or learning data.
Main Flow	1. Student clicks suggested materials. 2. Students are fed on AI engine through system. 3. AI interprets the trends of performance. 4. AI gives back recommended material. 5. Student recommendations are exhibited by the system.
Postconditions	Personalized recommendations displayed.
Exceptions	AI service timeout, insufficient data.

3.4.2.19 AI Suggest Improvement Areas

Table 3.19: AI Suggest Improvement Areas

Use Case ID	UC-20
Use Case Name	AI Suggest Improvement Areas
Actors	AI Microservice, Student
Preconditions	Student must have attempted at least one quiz.
Main Flow	1. Student opens develops understanding. 2. Quiz results are sent to AI. 3. AI identifies weak areas. 4. AI provides feedback with specific improvement recommendations. 5. Student was proposed topics on the system displays.
Postconditions	Improvement suggestions displayed successfully.
Exceptions	AI unavailable, incomplete student data, network issues.

3.4.2.20 Text Extraction and Question Generation

Table 3.20: Text Extraction and Question Generation Use Case

Use Case ID	UC-21
Use Case Name	Text Extraction and Question Generation
Actors	AI Microservice
Preconditions	Material (PDF/DOCX) uploaded and accessible.
Main Flow	1. System receives Material ID. 2. Downloads file from storage. 3. Parses text content. 4. Communicates text to LLM to generate questions. 5. Questions are provided in a structured format by AI. 6. Questions are saved on Firestore by the system.
Postconditions	Quiz questions generated and saved successfully.
Exceptions	File not found, AI service unavailable, parsing error.

Chapter 4

Design

4.1 System Architecture

Scholar Path is based on a recent, distributed architecture that is aimed at being scaled and real-time performance. The system is developed like a cloud-based application in which the frontend interacts with the services on the cloud directly with core operations, and offloads the complex intelligence operations to an Application-Specific microservice.
The major elements of the system architecture are:

1. **Frontend (Mobile Application):** The frontend was developed through Flutter and offers a integrated Android and iOS interface. It has a rigid architectural structure of Model-ServiceViewModel (MVVM). This guarantees a clean division of concerns, in which reactive data streams by the business logic are listened to by the UI layer, to provide a glitch free user experience.

2. **Core Backend (BaaS):** The application uses Google Firebase as the major one backend-as-a-service. Firebase authentication manages secure user sessions and role management (Student, Teacher, Admin). With this serverless solution, one can assume scalability within seconds without having to maintain your servers manually.

3. **Primary Database:** Firebase Firestore is the major central NoSQL database. It stores all organized information such as user records, hierarchies of classes, material records and quiz results in a document-oriented form. This makes it possible to have realtime data synchronization so that a teacher can post a quiz, and it displays immediately on the student's dashboard.

4. **Quiz Creation Module:** The teachers create their quizzes manually on the basis of uploaded materials. The system offers Multiple Choice Question (MCQs) building tools making sure that quizzes are in line with the content and learning objectives. This approach offloads the main application, yet allows proper evaluation.

5. **Progress Tracking Module:** This is an implementation in the logic layer of the application module summarizes quiz activity and study activities. It calculates metrics such as mastery score in a real-time way which gives students immediate visual feedback. on their academic standing.

6. **Quiz Management Engine:** This is a pipeline between the quiz and the frontend creation tools. It enables educators to create, post, and maintain quizzes to the students, such as grading and feedback, which is a well-structured and smooth evaluation workflow.

4.1.1 System Architecture Diagram

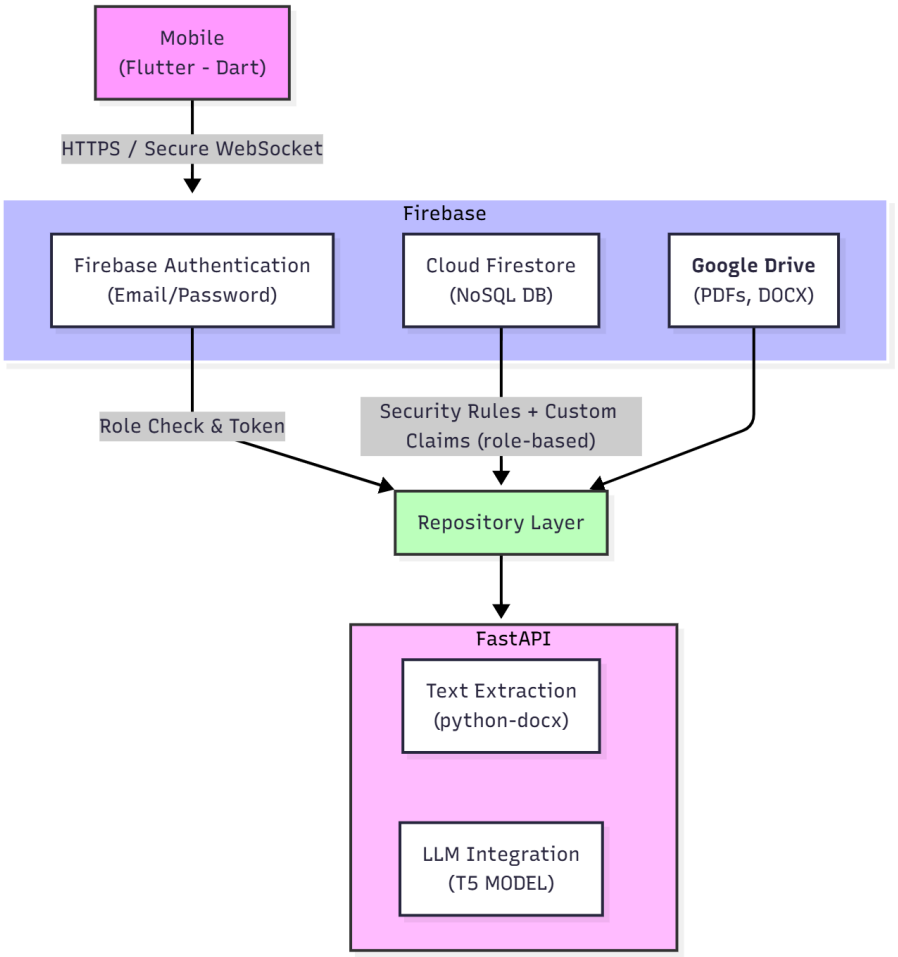

Figure 4.1: Use Case Diagram

4.2 Limitations on Design

1. **Target Demographic:** The original version is aimed mostly at university students.

2. **Content Quality:** The teacher is supposed to deliver clear and well-organized quiz and learning suggestions materials.
3. **Internet Connection:** Stable Internet connection is needed in Recommendation and scheduling modules.
4. **Hardware Constraints:** Real-time content viewing and attempting a quiz demand moderate performance on a device.

4.3 Methodology

Scholar Path was developed in a process of software engineering built to deal with the complexity of a distributed and multi-role system. The project adopted the Agile Methodology since it enables flexibility and refining features in an iterative manner as AI-based quiz generation and role-based access control.

4.3.1 Agile Methodology

The project life cycle was split into several sprints, and it guaranteed the uninterrupted delivery and fast feedback incorporation.

4.3.1.1 Iterative Sprints

TThe development process was planned as follows 2-week iterations:

1. **Phase 1:** Installation of the basic architecture (Provider, AutoRoute) and deployment authentication.
2. **Phase 2:** Implementation of role-based dashboards and CRUDs operations for Admin, Teacher and Students.
3. **Phase 3:** Adding the AI microservice that generates quizzes to the FastAPI-generated one.
4. **Phase 4:** User interface fine tuning, performance tuning (including caching), and comprehensive testing.

4.3.1.2 Continuous Integration

Constant unit and widget testing meant that other new features like the File Viewer did not interfere with the current functionality such as authentication.

4.3.2 Application Framework & Technology Stack

In contrast to other web-only solutions, Scholar Path is designed as a cross-platform solution. founded on a three-tier architecture.

4.3.2.1 Frontend (Mobile)

Flutter (Dart) is used to develop the front-end. Flutter was selected to facilitate one codebase compatible with Android and iOS. It is based on the MVVM(Model-View-ViewModel) pattern with Provider as the state management and with a responsive and reactive user interface.

4.3.2.2 Backend (BaaS & Microservice)

The system takes a hybrid backend architecture:

- **Firebase (Backend as a Service):** Provides the following vital services Authentication, Google Drive.
- **FastAPI (Python):** A microservice that handles computational intensive workloads like document content processing and large language communication models to generate quizzes.

4.3.2.3 Database (Real-Time NoSQL)

The main database is Cloud Firestore that provides real-time synchronization for all users. As well, the Hive is being used as a local, NoSQL cache on the device, which allows it. Intensive load speed and offline support by a so-called cache-first scheme.
Through the morphism of these particular functional areas, Scholar-Path provides a solution to the limiting severe shortcomings of conventional e-learning tools by offering an integrated solution, which is both technically balanced. viability of high educational effect.

4.4 High-Level Design

4.4.1 Conceptual View

- **AI-Based Personalization:** It analyze user performance and materials to create quizzes and prescribe subjects
- **Quiz Generation and Evaluation:** This generates individualized quizzes out of uploaded content and past results.
- **User Management and Learning Analytics:** Registration and Authentication gauges, and monitors progress of users.

4.4.2 Process View

- **AI-Based Recommendation:** Gathers interaction, quiz results, as well as materials to analyze and suggest study content.
- **Quiz Generation and Evaluation:** Derives major concepts and creates quizzes with instant feedback.
- **Progress Tracking and Analytics:** Measures performance and suggests areas for improvement.
- **User Authentication:** Dasherboards, secure logins and role-based.

4.4.3 Physical View

- **Backend:** Firebase Cloud Services for database, authentication, and storage.
- **Frontend:** Flutter deployed on Android/iOS/web.
- **Database:** Firebase Firestore securely stores all user, material, and quiz data.

4.4.4 Module View

- **User Management Module:** Registration, authentication, and role-based access.
- **Content Management Module:** Upload and organization of study materials.
- **Quiz and Evaluation Module:** Generates quizzes, evaluates responses, tracks performance.
- **Recommendation and Analytics Module:** Suggests personalized learning paths and monitors progress.

4.4.5 Security View

- **Encryption:** Data that is sensitive that is encrypted prior to storage.
- **Secure Communication:** HTTPS implemented in all client-server communications.
- **Authentication and Authorization:** Firebase Auth is employed to verify identity, enabling access control using tokens as well as role based access control (RBAC).

4.5 Low-Level Design

4.5.1 Recommendation Module

- **Data Preprocessing:** Transforms the uploaded materials and quiz results into embeddings.
- **Model Inference:** Makes personalized, trained deep learning model suggestions.
- **Feedback Integration:** Tunes AI model via the interaction with users.

4.5.2 Progress Tracking Module

- **Data Collection:** Records interactions, quiz results and documents.
- **Performance Analysis:** Strengths and Weaknesses.
- **AI-Based Feedback:** Use of individualized feedback.
- **Progress Visualization:** Displays progress reports through dashboards.

4.5.3 Quiz Creation Module

- **Content Analysis:** Teachers learn the major concepts of uploaded materials.
- **Question Creation:** Educators hand create MCQs.
- **Result Evaluation:** Student responses and performance displayed in system summaries.

4.5.4 User Management Module

- **User Registration:** Dealing with the student, teacher, and the admin account.
- **Authentication:** Provides credentials validation based on Firebase Auth tokens, which uses the authentication option, strengthy session management and token verification system.
- **Role-Based Control:** Access to dashboards and features is granted based on role.

4.6 Database Architecture

The system is based on Firebase Firestore and the data is conveniently managed in collections:

- **Users:** name, email, role, encrypted credentials.

- **Resources:** Metadata study materials.
- **Quizzes:** Data, attempts, and scores are stored in the form of quizzes.
- **Recommendations:** Stores AI-generated recommendation and user feedback.

4.7 Graphical User Interface

- **Registration Page:** This is where the user can create accounts.
- **Login Page:** Student, teacher and admin secure login.
- **Dashboard (Student/Instructor):** Displays the learning progress, quizzes, schedules, and materials.
- **Quiz Page:** The student does quizzes and gets the feedback right away.
- **Admin Dashboard:** Governing, tracking activities and managing materials.

4.8 Sequence Diagrams

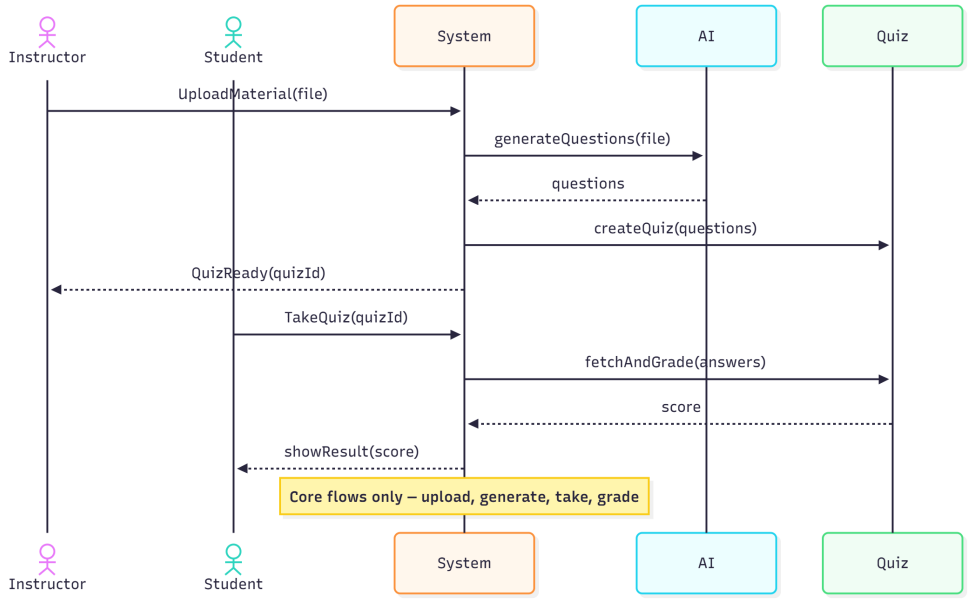

Figure 4.2: Sequence Diagram

This sequence diagram shows the central automated quiz process, in which uploaded material by an instructor is analyzed by an artificial intelligence element to create questions and form a quiz. After preparing the quiz, a student is able to take it via the system and the answers are automatically graded by the system, and the resultant mark is presented.

4.9 Activity Diagrams

The diagram presented below describes the authentication process of the application, in which a session check is performed at the launch of the application and redirects logged-in users or initiates the process of onboarding and role selection in new users. Once a user has logged in through Firebase or registered, the system conducts a role verification to make sure that he/she is properly redirected to either the Student, Teacher, or Admin dashboard.

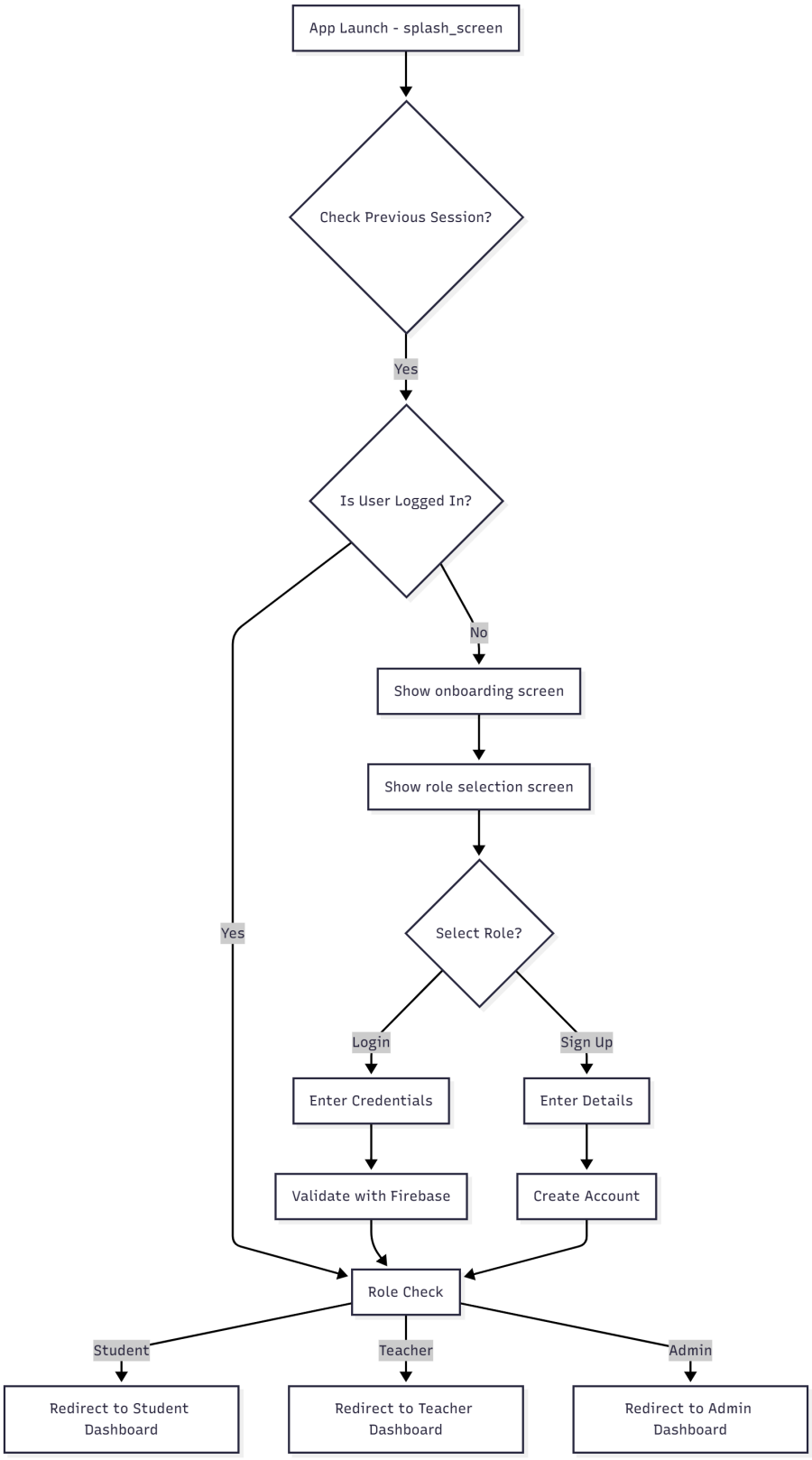

Figure 4.3: Authentication & Role Redirection Activity Diagram

4.9.1 Student Learning Flow Activity Diagram

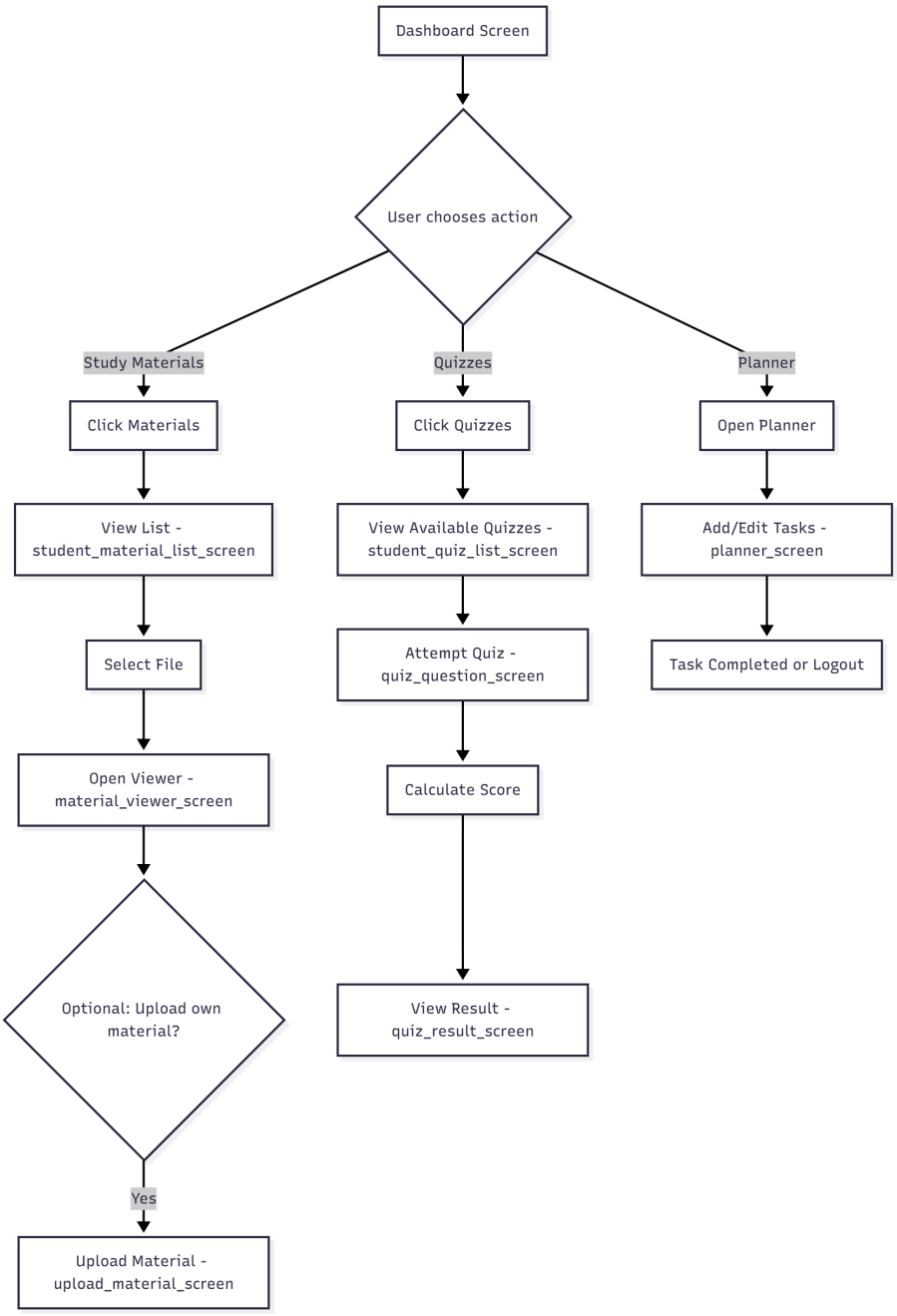

Figure 4.4: Student Learning Flow Activity Diagram

The following activity diagram describes the workflow of the student, as he or she will be able to select between studying material, taking exams, or using their academic planner. Students are given a chance to peruse and optionally post materials, take electronic timed quizzes that are automatically graded, and plan activities in order to keep track of what is completed or in progress.

4.9.2 Teacher Administration Flow Activity Diagram

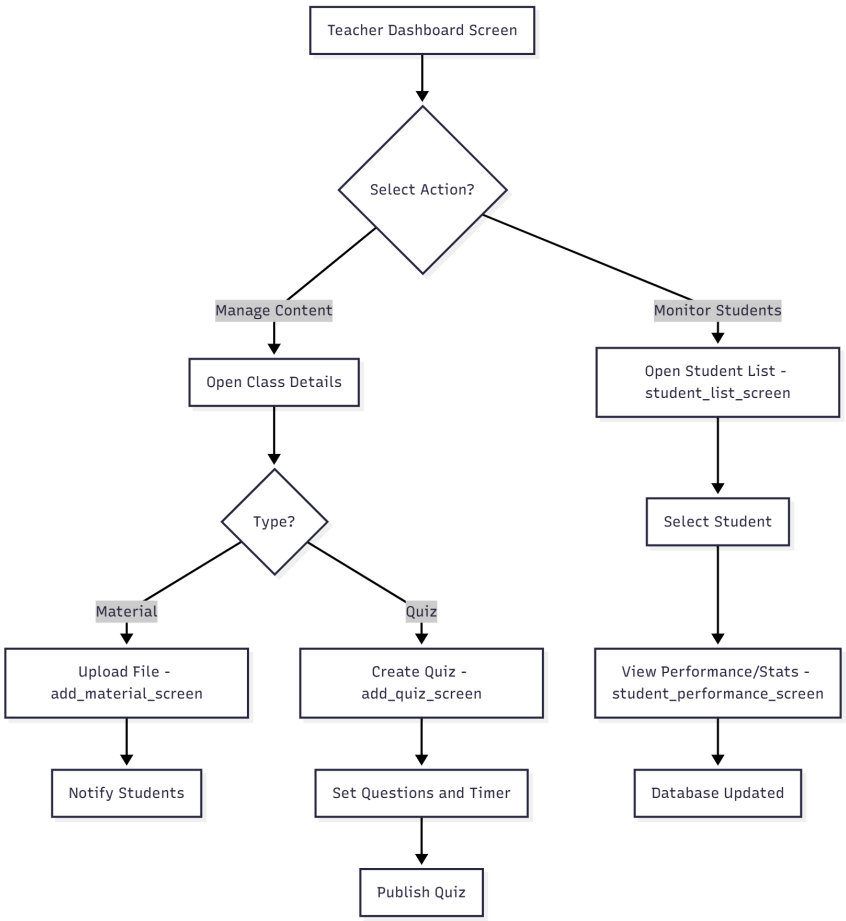

Figure 4.5: Teacher Administration Flow Activity Diagram

The activity diagram above will show the workflow of the teacher that will split to either handle the content of the classes by uploading the materials and creating quizzes or examining the performance of the students. This process implies that once the actions have been fulfilled, the system automatically leads an update of the database and provides notifications to the concerned students.

4.9.3 Admin System Management Flow Activity Diagram

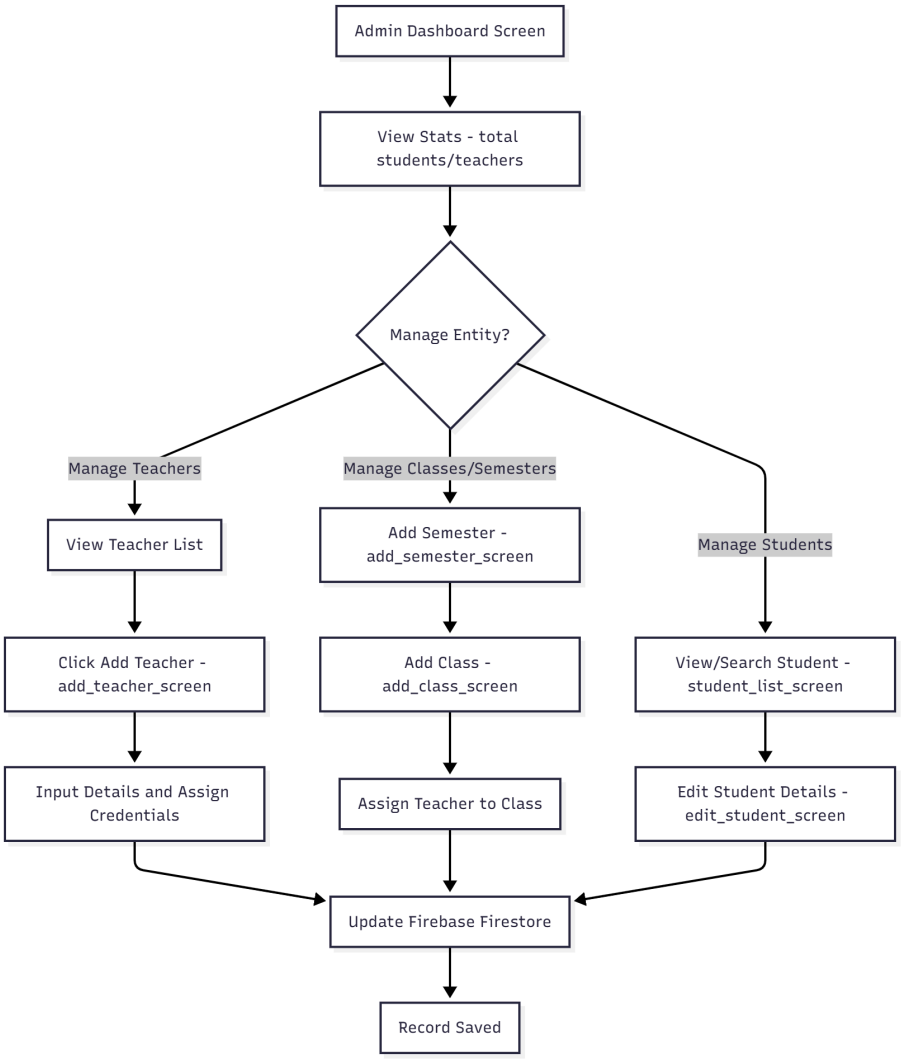

Figure 4.6: Admin System Management Flow Activity Diagram

The workflow of the administrator in the management of the core entities of the platform, which are the teachers, the students, and the academic structures such as semesters and the classes, have been outlined in this activity diagram. The admin is now able to do specific tasks after viewing the system-wide statistics like assigning teacher credentials or updating the details of students, and any of the changes are automatically reflected in the Firebase Firestore database.

4.10 Summary

This Chapter outlines the system design in general, its key functional modules, database design, and user interface. The design comprises progress visualization in the form of

dashboards, a quiz creation module that will be returned to the educators, and a user management system that will have secure authentication and role-based access control. Firebase Firestore is applied to store and manage the data in an effective way and handle users, learning materials, quizzes, and AI-based recommendations. The user interface will serve students, teachers and administrators with special dashboards and interactive quiz pages. The sequence and activity diagrams are used to depict system behaviour and workflows, including authentication and learning processes alongside administrative processes.

Chapter 5

System Implementation

The translation of the architectural blueprint into a implementation phase is known as Scholar Path the operating and production quality application. This step was focused on developing a distributed, three-level system where the various user type: Students, Teachers and Administrators are integrated into a unified system.
It was developed in a very straightforward separation of concerns: Flutter was the cross-platform client, Firebase provided real-time persistence and identity, and a specific Python client was used. AI-related processing was done by microservice. This chapter explains how these elements were implemented in code, the engineering patterns used (e.g. Dependency, Injection and Repository) patterns and the database layout that is intended to implement scalable academic workflows.

5.1 System Architecture Overview

Scholar Path is an application that is built as a distributed cloud application. To provide high availability and a seamless experience under different network conditions. The system is a combination of local persistence using cloud synchronization in a hybrid data strategy.

Five components of the architecture are:

- **Flutter Client-Side Application:** This is the main user interface that is compiled to run on Android and iOS. It is reactive in its approach with views that change automatically to state changes.
- **Firebase Authentication (Identity Layer):** Manages authentication of secure sessions and is the source of authoritative user identity (UID) on which the basis of Role-Based Access Control (RBAC) of platform is laid.
- **Firebase Firestore (Data Layer):** No SQL document store that is the primary database for user-profile, class-relationship (enrollments) and content metadata.

- **Google Cloud Storage (Asset Layer):** This contains binary files like PDFs; Firestore keeps reference URLs to these files.
- **FastAPI Microservice (AI Processing Layer):** Intensive functionality such as text extraction and this Python service is assigned NLP, to ensure the lightweightness of the mobile client. It relates the raw study materials with the LLM through which a quiz would be created.

5.2 Front-End Implementation

On the front side, Flutter is utilized to ensure that there is only one codebase across platforms. The application is based on a Model-Repository-Service-Viewmodel-UI (MVVM) architecture to ensure that business logic is not mixed with presentation to enhance testability and maintainability.

5.2.1 State Management and Dependency Injection

Dependency Injection is possible with the Provider package, which also controls application state. (DI) service(s) (e.g. AuthService or StudentService) is provided to the widget tree at the root.
The database is not accessed by the UI widgets. Rather, they see View Models(e.g., DashboardViewModel). Changes in the backend data (e.g. when a new quiz is published by teacher), the View Model informs listeners and the subscribing widgets, the View Model notifies listeners and the subscribing widgets rebuild automatically, without complete page reloads.

5.2.2 Navigation and Routing

AutoRoute library is used to implement routing via a RootStackRouter. It allows the use of Navigation Guards that intercept route requests to do authentication role of the state and user and then grant access to the secured screens (e.g., blocking a student is not able to reach Teacher Dashboard).

5.2.3 User Authentication Flow

The flow of logins contains extra verifications of the credentials:

- Credentials in the app are verified by Firebase Auth.
- When authentication is successful, a background process in AuthService makes a query to Firestore to identify the position of the user (Student, Teacher, or Admin).

- The first role look up bypasses the local cache and uses a network-first fetch to avoid problems with logging in due to the presence of outdated information in the cache.

5.2.4 Material and Quiz Interfaces

Material management screens allow one to operate with uploaded files. The viewer strategy is hybrid:

- **PDFs:** PDFs can be shown in-app through the use of a PDF view controller to provide a smooth reading experience.
- **Other Formats:** Other files such as Word documents or videos are accessed in the platform through native handlers (through $\text{url}_{l}auncher$)$sothebestpossibleexternalapplication.They$

5.3 Back-End and Logic Implementation

The serverless Firebase functions and one dedicated compute service (FastAPI) share backend logic.

5.3.1 Repository Pattern and Caching Strategy

Scholar Path has a Cache-First to enhance responsiveness and reduce network utilization.Pattern based on Hive (a lightweight local NoSQL store) or repository.
In cases where the UI demands data (e.g. a student list), the Repository first checks local Hive box and shows data instantly provided. At the same time, a background fetch data from firestore synchs the local cache. This is to make sure that the app can be used offline or online on weak networks as they eventually connect with the server.

5.3.2 Quiz Creation Workflow

The system tools are used by educators to create the quiz. The workflow is as follows:

1. Flutter client enables the educator to choose a materialID on which a quiz should be chosen to create.
2. The system retrieves the file URL related to that file of Firestore and grants access to the document.
3. Teachers go through the material and pick out important points that can be assessed.
4. Using quiz creation tools available on the platform, teachers came up with several multiple-choice questions manually.

5. After it has been completed, the quiz will be stored under the Firestore quizzes collection and will be saved as such students can have access to through real-time streams.

5.3.3 Data Integrity and Race Condition Resolution

One implementation difficulty was a race condition in the signup process: the authentication token was emitted a bit earlier than the profile document of the user was written to Firestore. This caused unsuccessful role checks and automatic logout.The solution optimized AuthService by relying on network-first strategy in the initial session initialization. Besides, the use of retry logic was provided to poll the database a bit and profile document is not located immediately, enhancing the strength in account creation.

5.4 Database Design

The database is based on Firebase Firestore and has a schema that minimizes read operations and enforce security.

- **Users Collections (students, teachers, admins):** The users are grouped into distinct ones rather than one series. This eases security policies (such as only teachers collection documents can write to Class documents). Each profile metadata and Firebase UID are the primary key in the document.
- **Classes Collection:** The main reference point where arrays of IDs (e.g., `studentIds`, `quizIds`, `materialIds`). This denormalized structure enables retrieval of a class document, and having direct knowledge of its corresponding material without extravagant complexity.
- **Materials Collection:** It also contains metadata of files, containing fileUrl and uploaderId and allowing the checks of ownership and enforcement of permissions.
- **Quizzes and Results:** Divided into two groups: quizzes stores immutable. question/answer sets, and quiz results avoid redundant student attempts with fields like studentID, quizID, score and completedAt. This separation helps to do the analytics (such as the calculation of an average score of a student) efficiently retrieving the complete quiz payload.

5.5 System Interface Snapshots

The figures below show the key functional modules and user interfaces that were created in the Scholar-Path application.

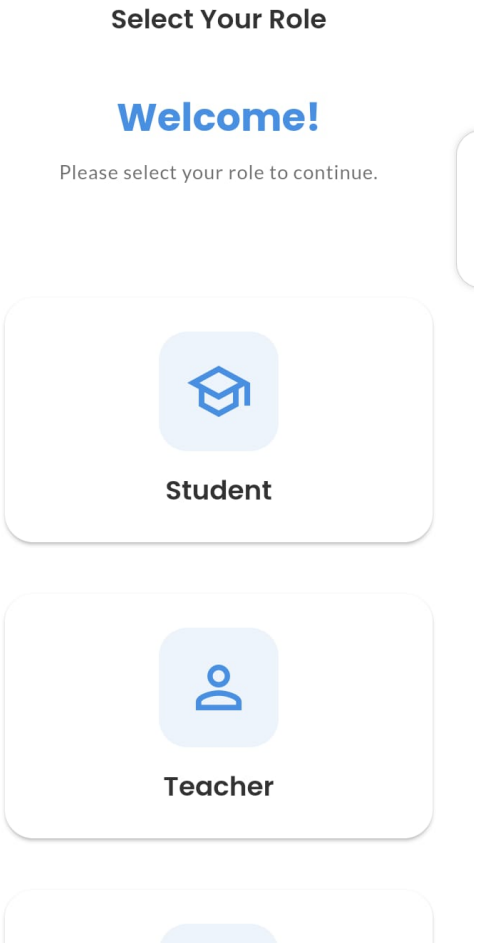

Figure 5.1: Role Selection Page: Displays the initial welcome screen prompting the user to select their role (Student, Teacher, or Admin) to proceed with the application.

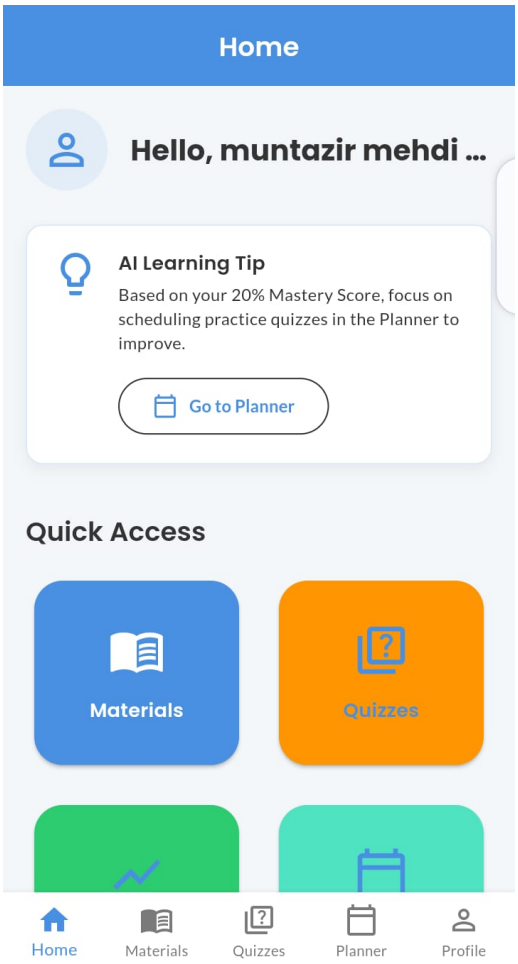

Figure 5.2: User Home Page: Displays the personalized landing screen, including the user's name, an AI Learning Tip, and Quick Access navigation to Materials and Quizzes.

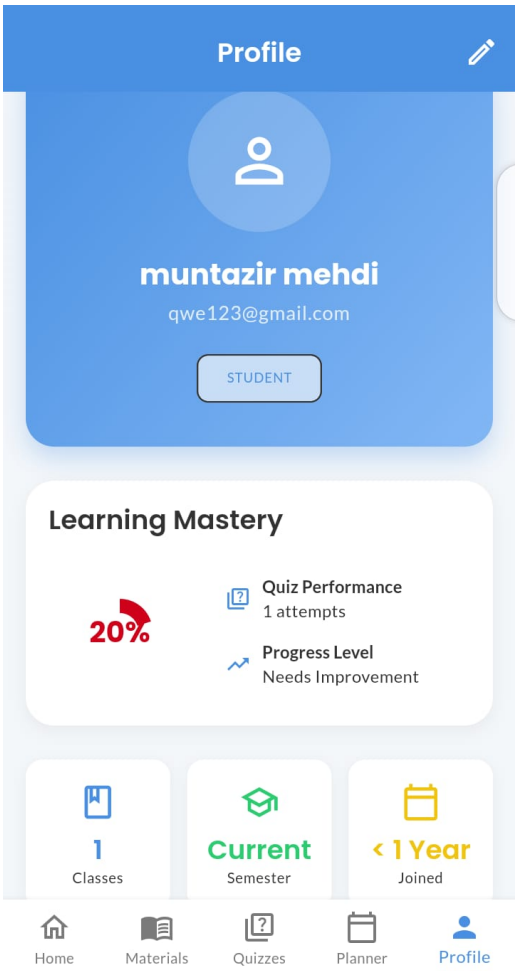

Figure 5.3: User Profile and Learning Mastery: Displays the user's personal details, registered role (Student), and key performance metrics, including the calculated Learning Mastery score (20%), Quiz Performance, and Progress Level.

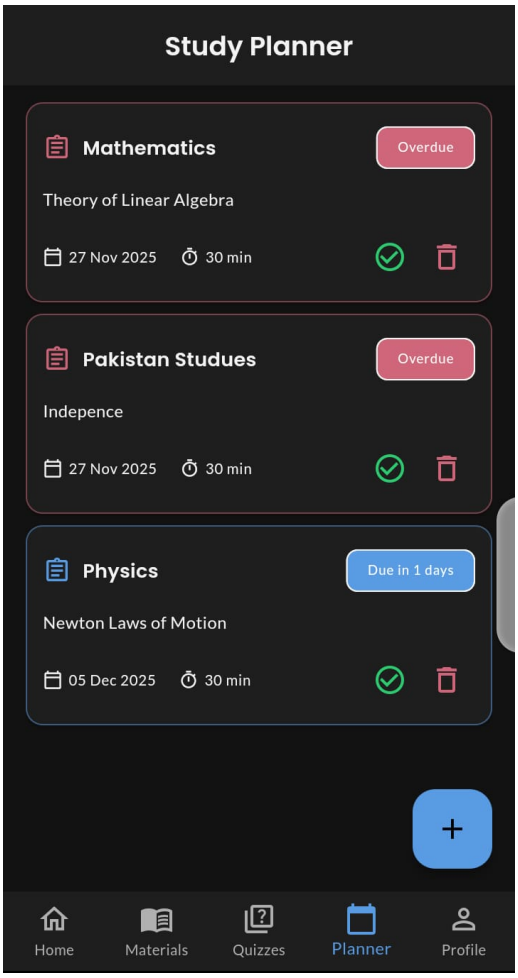

Figure 5.4: Study Planner Interface (Dark Mode): Displays the user's scheduled academic tasks, including subjects, topics, dates, and estimated time.

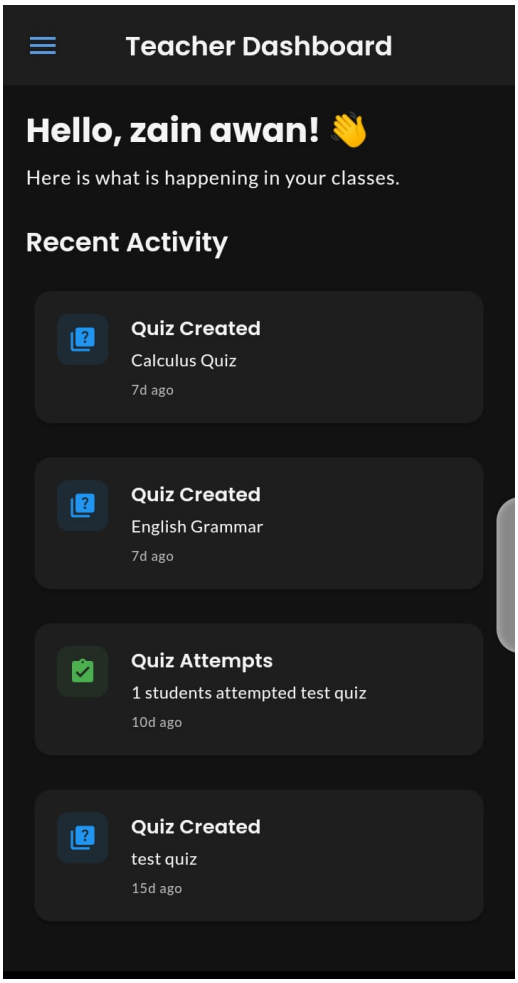

Figure 5.5: Teacher Dashboard: Displays the main landing screen for the teacher user, featuring a summary of Recent Activity in their classes, including newly created Quizzes and records of Quiz Attempts by students.

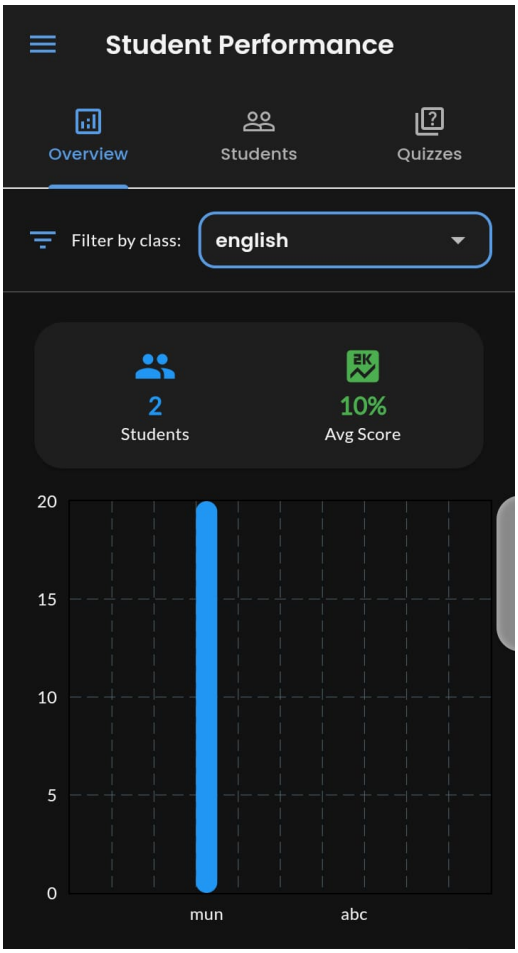

Figure 5.6: Student Performance Overview: Shows a student's progress through Quiz Score Trend graph and categorized performance metrics in Learning Mastery and Average Score.

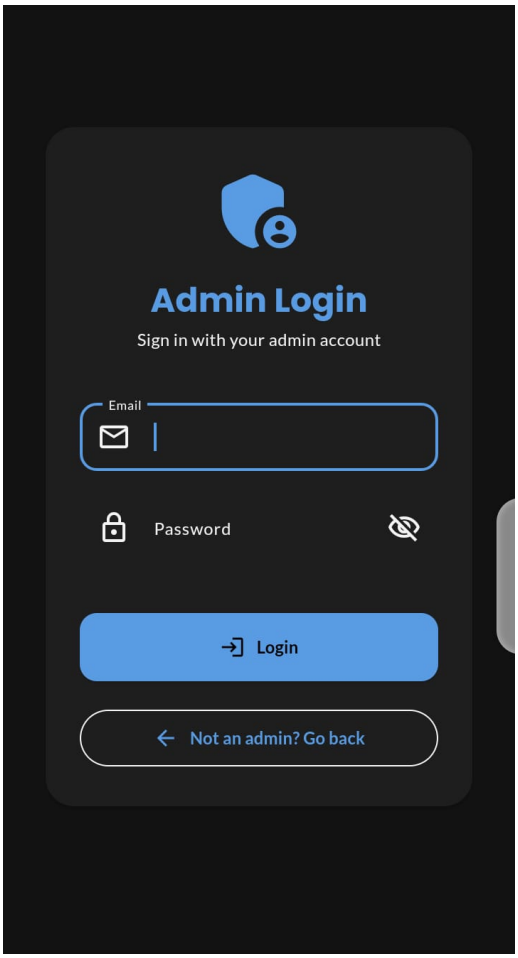

Figure 5.7: Administrative Login Interface: Displays the login panel for admin users requiring Email and Password for secure system access.

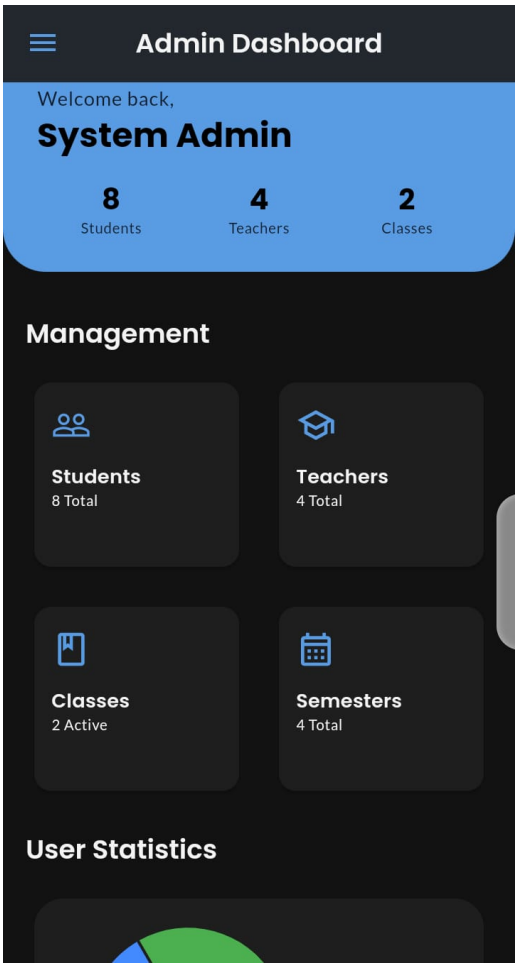

Figure 5.8: System Administrator Dashboard: Shows system-wide statistics (Students, Teachers, Classes) and provides management controls for Semesters and User Groups.

5.6 Summary

The figures above show the key functional modules and user interfaces that were created in the Scholar-Path application. The implementation of the Scholar Path connects the modern mobile development practices and stable cloud architecture. The transfer of the intensive AI processing to a back-end microservice and the usage of a powerful caching pattern makes the client responsive and scalable. The latter system offers a digital workflow, beginning with secure content delivery to tomputer-based intelligent exams.

Chapter 6

System Testing and Evaluation

The testing is also a necessity in order to establish that the system works, is reliable, is secure, and is performing as intended. In the testing process, Scholar Path was analysed to be stable,commoditiness, and the lack of faulty elements. A mix of methods of tests was used to check the system within a multi-module setting and vary environments.
In this chapter, the testing and evaluation of the Scholar Path Smart Academic Management System is presented. It is meant to ensure that the system is as required, is user-friendly, and works very well within different conditions. The Graphical User Interface (GUI) Testing, Usability Testing, Security Testing , Exception Handling, Compatibility Testing, Performance Testing, and Installation Testing were included in the testing activities.

6.1 Testing Methodology

The testing methods that were used to prove the strength and capabilities of Scholar Path were as follows.

- **Unit Testing**: Each separate module (like login, user management, material upload, and quiz management etc.) is tested separately to ensure that they behave correctly in isolation.
- **Integration Testing**: Checked the communication of Flutter front-end with back-end services (Firebase Firestore, Firebase Auth and firebase storage).
- **System Testing**: Tested the whole system in conditions that simulated real-life use to verify that all modules work well and the system achieves project objectives.
- **User Acceptance Testing (UAT)**: The system was tested by the end users (students, teachers, and administrators) who determined its usability, accessibility, and effectiveness in relation to academic workflows.

6.2 Graphical User Interface (GUI) Testing

Testing focus on GUI was based on visual layout, responsiveness and functional behavior of the interface. It was aimed to ensure the visual consistency of Scholar Path, easy to use, and similar among device types.
The tests covered:

- **Layout Consistency**: Verified alignment and placement of buttons, input fields, forms, and dashboard components on various screen sizes.
- **Navigation Testing**: Make sure that every page has a smooth transition (Dashboard, Classes, Without any errors Quizzes Materials)
- **Interactive Elements**: Verified that there is dynamic behavior and appropriate behavior of form inputs, file uploads and quiz submissions.
- **Error Handling**: Verified that meaningful error messages are displayed on invalid permeable measures or illegal acts.

The GUI was found to have been responsive, up to date and up to the expectations in terms of usability.

6.2.1 Student Registration

Table 6.1: Test registration functionality for students.

Test Case ID	TC-001
Description	Description Test registration functionality for students.
Preconditions	User must be on the registration page and select the “Student” role.
Test Steps	1. Navigate to the Student Registration page. 2. Fill in required fields (name, email, password, department, etc.). 3. Click “Sign Up.”
Expected Result	The student account is created successfully, and a confirmation message is displayed.
Actual Result	Actual Result The student account is created successfully, and a confirmation message is displayed.
Test Status	Passed

6.2.2 Instructor Registration

Table 6.2: Test registration functionality for instructors.

Test Case ID	TC-002
Description	Test registration functionality for instructors.
Preconditions	ser must be on the registration page and select the “Instructor” role.
Test Steps	1. Navigate to the Instructor Registration page. 2. Enter details such as name, email, password, and area of expertise. 3. Click “Sign Up.”
Expected Result	he instructor account is created successfully, and a confirmation message is displayed.
Actual Result	The instructor account is created successfully, and a confirmation message is displayed.
Test Status	Passed

6.2.3 Student Login

Table 6.3: Test login functionality for students.

Test Case ID	TC-003
Description	Test login functionality for students.
Preconditions	The student account must already exist.
Test Steps	1. Navigate to the Login page. 2. Enter valid credentials. 3. Click “Login.”
Expected Result	The student is redirected to the Student Dashboard.
Actual Result	The student is redirected to the Student Dashboard.
Test Status	Passed

6.2.4 Instructor Login

Table 6.4: Test login functionality for instructors.

Test Case ID	TC-004
Description	Test login functionality for instructors..
Preconditions	The instructor account must already exist.
Test Steps	1. Navigate to the Login page. 2. Enter valid credentials. 3. Click “Login.”
Expected Result	The instructor is redirected to the Instructor Dashboard.
Actual Result	The instructor is redirected to the Instructor Dashboard.
Test Status	Passed

6.2.5 Admin Login

Table 6.5: Test login functionality for Admin

Test Case ID	TC-005
Description	Test login functionality for admin users
Preconditions	The admin account must already exist.
Test Steps	1. Navigate to the Admin Login page. 2. Enter valid admin credentials. 3. Click "Login."
Expected Result	The admin is redirected to the "Admin Dashboard."
Actual Result	The admin is redirected to the "Admin Dashboard."
Test Status	Passed

6.3 Usability Testing

Usability testing assessed how intuitive and efficient the platform is for real users. The tests measured whether users could complete common tasks and how the interface supported those tasks.
Findings indicated that the platform is easy to use, visually consistent, and engaging for both students and instructors.

6.3.1 Usability Test Cases

Table 6.6: Test dashboard usability.

Test Case ID	TC-006-1
Description	Verify that the dashboard loads properly for students and instructors.
Preconditions	User must be logged in as a student or instructor.
Test Steps	1. Log in to ScholarPath. 2. Navigate to the dashboard. 3. Observe the UI elements.
Expected Result	All UI elements appear correctly.
Actual Result	All UI elements appeared correctly.
Test Status	Passed

Table 6.7: Test font readability on different devices.

Test Case ID	TC-006-2
Description	Check font readability across desktop, tablet, and mobile devices.
Preconditions	User must be logged in.
Test Steps	1. Open ScholarPath on desktop, tablet, and mobile. 2. Observe font clarity and readability.
Expected Result	Fonts are clear and readable on all devices.
Actual Result	Fonts were clear and readable.
Test Status	Passed

Table 6.8: Test navigation between pages.

Test Case ID	TC-006-3

Description	Verify smooth navigation between Dashboard, Courses, Quizzes, and Profile.
Preconditions	User must be logged in.
Test Steps	1. Navigate through Dashboard, Courses, Quizzes, and Profile pages. 2. Observe transitions and page loads.
Expected Result	Navigation is smooth without errors.
Actual Result	Navigation was smooth without errors.
Test Status	Passed

Table 6.9: Test upload of materials and taking quizzes.

Test Case ID	TC-006-4
Description	Verify that users can upload materials or take quizzes successfully.
Preconditions	User must be logged in.
Test Steps	1. Upload a material (teacher) or select a quiz (student). 2. Complete the action and observe behavior.
Expected Result	Actions complete successfully without errors.
Actual Result	Actions completed successfully.
Test Status	Passed

Table 6.10: Test display of error and success messages.

Test Case ID	TC-006-5
Description	Validate that error and success messages are clear and descriptive.
Preconditions	User must be logged in.
Test Steps	1. Perform actions that trigger success or error messages (e.g., incorrect login, successful upload). 2. Observe the messages displayed.
Expected Result	Messages are clear and descriptive.
Actual Result	Messages were clear and descriptive.
Test Status	Passed

Table 6.11: Test responsiveness on mobile and desktop devices.

Test Case ID	TC-006-6
Description	Check that layout adapts properly to different screen sizes.
Preconditions	User must be logged in.
Test Steps	1. Open ScholarPath on desktop and mobile devices. 2. Observe layout and UI adjustments.
Expected Result	Layout adapts properly without broken elements.
Actual Result	Layout adapted properly.
Test Status	Passed

6.4 Performance Testing

Tests of performance were taken on responsiveness, stability and scalability at diverse loads conditions. Some of the metrics followed were response times, resource consumption, and the ability to handle concurrent users.

- **Response Time**: Response Time: Response times are typically common (e.g. login, quiz submission) completed in under 3 seconds. The average loading of adaptive content was 4-5 seconds, which suffice in the context of nearly real time interactions.
- **Scalability**: The concurrent usage was simulated with a maximum of 100 users; the system continued to operate reliable and crash-free facile and laggardly untroubled.
- **Resource Usage**: CPU and memory usage remained satisfactory at times when the server is busy such as quiz generation and uploading of large files.

6.4.1 Performance Test Cases

Table 6.12: Test login performance for each user role.

Test Case ID	TC-007-1
Description	Test the time taken for the login process for each user role (Admin, Teacher, Student).
Preconditions	The Scholar Path application is deployed and the user has valid credentials.
Test Steps	1. Navigate to the Login page. 2. Enter valid credentials for each user role. 3. Click "Login."
Expected Result	Login should complete within 2–3 seconds.
Actual Result	Login completed within 2–3 seconds.

Test Status	Passed

Table 6.13: Test dashboard loading time after login.

Test Case ID	TC-007-2
Description	Test the time taken to load the dashboard after login.
Preconditions	User is logged in to the Scholar Path application.
Test Steps	1. Complete login process. 2. Observe the dashboard loading time.
Expected Result	Dashboard should appear and data should load within 3 seconds.
Actual Result	Dashboard loaded within 3 seconds.
Test Status	Passed

Table 6.14: Test material upload speed by teachers.

Test Case ID	TC-007-3
Description	Test the upload speed for materials (PDFs) by teachers.
Preconditions	Teacher is logged in and has access to upload materials.
Test Steps	1. Navigate to the materials upload section. 2. Upload a file (PDF or image). 3. Observe the upload time.
Expected Result	Upload should complete within 5–7 seconds depending on file size.
Actual Result	Upload completed within 5 seconds.
Test Status	Passed

Table 6.15: Test fetching class data and materials for students.

Test Case ID	TC-007-4
Description	Test the response time for fetching class data and materials from Firestore for students.
Preconditions	Student is logged in. Classes and materials exist in Firestore.
Test Steps	1. Navigate to the class materials section. 2. Load the list of classes and materials.
Expected Result	Materials and class lists should load within 2–4 seconds.
Actual Result	Loaded within 3 seconds.

Test Status	Passed

Table 6.16: Test quiz loading time for students.

Test Case ID	TC-007-5
Description	Test the loading time of a teacher-created quiz for students.
Preconditions	Student is logged in and quizzes created by educators are available.
Test Steps	1. Select a quiz to attempt. 2. Load the quiz interface.
Expected Result	Quiz should appear within 3 seconds after selection.
Actual Result	Quiz loaded within 3 seconds.
Test Status	Passed

Table 6.17: Test system performance under concurrent access.

Test Case ID	TC-007-6
Description	Test system performance when multiple users access classes or upload data simultaneously.
Preconditions	At least 30 users are logged in.
Test Steps	1. Simultaneously access classes or upload materials with multiple users. 2. Monitor system response and stability.
Expected Result	The system should handle concurrent operations smoothly without slowdown or crash.
Actual Result	System handled concurrent access without issues.
Test Status	Passed

Table 6.18: Test data synchronization speed between teacher and student.

Test Case ID	TC-007-7
Description	Test the speed at which teacher uploads reflect on student devices.
Preconditions	Teacher has uploaded new materials; student is logged in.
Test Steps	1. Teacher uploads new materials. 2. Observe the time it takes to appear on student devices.
Expected Result	Updates should reflect on all devices within 3–5 seconds.

Actual Result	Updates reflected within 4 seconds.
Test Status	Passed

Table 6.19: Test response time of admin dashboard analytics.

Test Case ID	TC-007-8
Description	Test response time for loading admin dashboard metrics (charts, user counts).
Preconditions	Admin is logged in and dashboard contains data.
Test Steps	1. Navigate to the admin dashboard. 2. Observe loading time of analytics and charts.
Expected Result	Dashboard metrics should load within 3 seconds.
Actual Result	Loaded within 3 seconds.
Test Status	Passed

6.4.2 Compatibility Testing

The compatibility test was done to make sure that the Scholar Path operates well with the major systems and it works there.

- **Device Compatibility**: Enabled on laptops, tablets and smartphones; layouts adapted appropriately.
- **Operating System Compatibility**: Tested as stable on Android, iOS. Windows, and macOS.

6.5 Security Testing

Security testing was aimed at securing user information and avoidance of unauthorized test. The main checks included:

- **Data Encryption**: Confirmed all the communication between the Firebase and the app backend use HTTPS. Credential and academic records are sensitive and they should not be lost and secured during transmission to avoid leaking.
- **User Authentication**: Checked Authentication flows of Firebase Authentication to make sure that only authorized flows(Admin module, teacher module, student module) Session and token validation and handling were also checked.
- **Input Validation**: Verified that form fields (registration, quiz answers, uploads) are valid and verified to fight malformed or malicious data.

- **Access Control**: Authenticated access control: The role-based access control prevents users to perform exclusively those actions that they are permitted to do (e.g., students are not allowed to change anything of teacher/admin data).

Due to security verification, it was proven that Scholar Path adheres to the good practices of encryption, role-based access, authentication and input validation.

6.6 Iterations and Bug Fixes

A number of development cycles were conducted to fix the defects and enhance the product. Notable problems identified and addressed were:

- **Data Sync Delays**: Instances of delays in the syncing of uploaded content to student devices were met by making Firestore listeners faster at real-time propagation.
- **UI and Navigation Improvements**: Response to usability testing resulted in UI reddesigns. Some of the UI elements dashboard layouts, navigation tabs and controls have been made better for smoother interaction.
- **Authentication Handling**: The issues with session persistence and logout were resolved with an improved Firebase Auth state management.
- **Performance Optimization**: Less eliminated redundant database calls, new caching where necessary, and turned blocking operations into calls of a asynchronous nature for smoother user experience.

These iterations did improve performance, reliability and user experience hence yielding better polished and fixed application.

6.7 Summary

SScholar Path has undergone intense testing in terms of functional, usability, performance and so on. To guarantee reliability and performance, compatibility, and security aspects. Iterative releases and performance work were made to enhance performance. The delivered system provides a reliable academic management service, which allows effective cooperation with students, teachers, and administrators and addressing the primary objectives of the project of availability, broadness, and efficient educational assistance.

Chapter 7

Conclusions

The creation of ScholarPath is a significant move towards the combination of modern educational with artificial intelligence. The initiative aimed at developing a adaptive learning platform through which the students will be able to customize their study patterns by using the AI given recommendations, monitor progress, and rehearse by interactive quizzes administered through a user-friendly interface. There were a number of fundamental objectives that were achieved in the project:

- **Intelligent Learning Personalization**: The built-in AI feature is doing well screened user-posted content and innovative learning procedure and generated customized content suggestions. The platform adapts to every one through machine learning applications progress of the learner and assists in putting efforts on the areas that require improvement.
- **Easy to use and Responsive Design**: The interface of ScholarPath was introduced to deliver an efficient, responsive experience with Flutter. Both students and educators is able to navigate the platform and make personalized dashboards, easily, and interact with Quizzes generated by AI without any technical skills.
- **Comprehensive Learning Management**: ScholarPath combines a number of educational capabilities: uploading content, content analysis on an AI-based, and automated evaluation into one environment. This integration bridges the dissimilarity among traditional class room processes and distance learning through provisions of an end-to-end adaptive solution.
- **System Reliability and Testing**: The system was highly tested to ascertain functionality, stability, and security. Evaluation of the performance, usability, and reliability showed that ScholarPath is reliable as per its behavior in various conditions. Ongoing improvements on the basis of user feedback were used to make them consistent and reliable operations.

In general, the project is a manifestation of the way AI can result in the improvement of e-learning by making it smarter, more experiential, and learner-centered experience. ScholarPath can be made better to access, interaction and learning in online education.

7.1 Limitations

The ScholarPath system has not been without its share of limitations despite the success of the system and development plan:

- **Dataset Diversity**: The recommendation engine was trained and assessed on an assumptions about the generalizability to other populations because its set of study materials is relatively diverse academic subjects. It would improve the quality of recommendations to expand the corpus of the training data by adding more, domain specific data.
- **Internet Dependency**: ScholarPath is a cloud-based application, which relies on a stable network connectivity to be fully functional. Customers in intermittent or slow regions can undergo disruptions in availing personalized resources.
- **Limited Scope of Features**: Models on the personalized content, the platform has a restricted number of features delivery, auto quizzes and it is not yet an advanced collaboration functionality and more visual analytics that would enhance even more engagement.
- **User Familiarity**: There are those who will need first time use with a simple interface, despite its simplicity advice how to become AI-driven. The tutorials and onboarding would be added for help within the app would facilitate adoption.

7.2 Future Work

The ScholarPath system shows a number of potential opportunities to be extended and improved:

- **Improved AI Model**: Future developments should see the AI enhanced by being trained on larger,more varied, and documentation data. The use of advanced architectures. The model or methods of reinforcement learning, transformer based would increase the accuracy of recommendation and question generation.
- **Integration with Learning Management Systems (LMS)**: Integrating ScholarPath would enable using an existing LMS (Moodle or Google Classroom) system. Integration of assignment, grades, and progress information and thus simplifying it for teachers to embrace the system.

- **Gamification and Analytics**: This involves addition of gamified components (badges, leaderboards), more analytics dashboards may make learners become more motivated and offer more in-depth understandings in performance trends.
- **Interactive Learning Tools**: Future releases may have interactive flashcards, even more advanced AI-based question generation of uploaded notes and voice interactions to make the learning more interactive.

7.3 Summary

To sum up, ScholarPath project is a good base of personalized foundation for education demonstrating the ways AI can be implemented in the learning process. The system coholics meetings achieve its primary objectives adaptive learning, interactivity and operational efficiency and providing open possibilities of improvement.
Despite the opportunities available to expand and refine the architecture and implementation, the following offered are confident grounds to begin with the work in intelligent educational in the future systems. With the support of the limitations and the attempt at implementing the proposed improvements, ScholarPath might become a more flexible, user-friendly, and constructive learning platform to serve students worldwide.

Appendix A

User Manual

A.1 User Manual and System Documentation

ScholarPath is an artificial-intelligence adaptive learning system that is intended to improve students' learning experience through customized content suggestions, progress tracking, and interactive quizzes. The system enables students and teachers to interact in a unified environment that supports content uploading, intelligent recommendations, and automated evaluation.

A.2 System Requirements

To use the ScholarPath mobile application, the following requirements must be met:

- **Equipment:** Android or iOS smartphone or tablet.
- **Operating System:** Android 8.0 (Oreo) or above, iOS 13 or above.
- **Internet Connection:** An active internet connection is required for data synchronization, file uploads/downloads, and communication with AI microservices.

A.3 Registration and Login of an Account

A.3.1 Student Registration

1. Download the ScholarPath mobile application and select **Sign Up**.
2. Choose **Student** as the user role.
3. Enter personal details such as name, email, and password.
4. Click on **Create Account** to complete registration.
5. Verify the account using the confirmation email.

A.3.2 Teacher Registration

Educators cannot self-register. Teacher accounts are created by the system administrator to ensure security and proper access control.

1. Request an account from the system administrator.
2. The administrator creates the account and assigns appropriate permissions.
3. Login credentials are shared via email.

A.3.3 Login

Registered users can log in using their email and password to access their respective dashboards.

A.4 Learning, Quizzes, and User Roles

A.4.1 Uploading Study Materials

Teachers can upload PDFs, notes, and slides through the Materials List interface. Students can access these resources through the **Study Materials** section.

A.4.2 Learning and Quiz Features

Students can attempt quizzes created by instructors and receive immediate feedback. Teachers can manually create quizzes based on uploaded study materials.

A.4.3 User Roles and Dashboards

A.4.3.1 Student Dashboard

- Learning Materials
- Quizzes and Feedback
- Personalized Recommendations
- Progress Reports

A.4.3.2 Teacher Dashboard

- Material Management
- Quiz Creation
- Student Performance Analytics

A.4.3.3 Admin Dashboard

- User Management
- Access Control
- System Monitoring

A.5 Support and Troubleshooting

Users can seek support through the in-application help system or by contacting system administrators. It is also recommended to keep the application updated to the latest version.

References

[1] Coursera. About coursera, 2025. [Online]. Available: https://www.coursera.org/about. [Accessed: Oct. 2025]. Cited on pp. 8, 9, and 13.

[2] Khan Academy. Our mission, 2025. [Online]. Available: https://www.khanacademy.org/about. [Accessed: Oct. 2025]. Cited on pp. 8, 9, 12, and 13.

[3] Rose Luckin, Wayne Holmes, Mark Griffiths, and Laurie B Forcier. *Intelligence Unleashed: An Argument for AI in Education*. Pearson, London, 2016. Cited on p. 8.

[4] Duolingo Research. How duolingo uses ai to personalize learning. *Duolingo Blog*, 2023. Online. Cited on p. 9.

[5] Quizlet. How quizlet works, 2025. [Online]. Available: https://quizlet.com/features. [Accessed: Oct. 2025]. Cited on pp. 10, 12, and 13.

[6] Benjamin S Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, 1984. No Citations.

[7] S P P Nayak and C N Sowmyarani. A review on nosql databases and its types. In *2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, pages 119–124, Bangalore, India, 2021. IEEE. No Citations.

[8] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994. No Citations.

[9] Sebastián Ramirez. Fastapi: High performance, easy to learn, fast to code, ready for production, 2025. [Online]. Available: https://fastapi.tiangolo.com. [Accessed: Oct. 2025]. No Citations.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. No Citations.

[11] John Anderson and Cynthia Boyle. Adaptive learning technologies: From principles to practice. *Journal of Learning Analytics*, 1(2):5–20, 2014. No Citations.

[12] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1):87–110, 2001. No Citations.

[13] Beverly Woolf. Building intelligent tutoring systems for education. *AI Magazine*, 31(1):1–10, 2010. No Citations.

[14] Knewton. The adaptive learning revolution. *White Paper*, 2013. No Citations.

[15] Seung Yang and Chen Li. Ai for education: hype or reality? *EDUCAUSE Review*, 2021. No Citations.

[16] Li Chen and Hao Wang. A survey of recommendation systems for learning. *IEEE Access*, 8:140352–140367, 2020. No Citations.

[17] Kyung Lee and Soo Kim. Automatic question generation for educational content: A survey. In *Proceedings of the ACL*, pages 1–14, 2019. No Citations.

[18] Yulia Susanti and Shu Kurita. Neural question generation for reading comprehension. *Computational Linguistics*, 2020. No Citations.

[19] Michael Heilman and Noah Smith. Good question! statistical ranking for question generation. In *Human Language Technologies*, pages 609–617, 2010. No Citations.

[20] Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. No Citations.

[21] Alec Radford, Jeffrey Wu, et al. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019. No Citations.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. *MIT Press*, 2016. No Citations.

[23] Luis Perez and Mei Liu. Evaluating large language models for educational quiz generation. *Journal of Educational Data Mining*, 2021. No Citations.

[24] Ming Feng and Neil Heffernan. Survey on personalized learning platforms. *Educational Technology Review*, 2019. No Citations.

[25] Olaf Zawacki-Richter and Victoria Marín. Systematic review of ai in higher education. *International Journal of Educational Technology in HE*, 2019. No Citations.

[26] Ryan Baker. Learning analytics and educational data mining. *Journal of Learning Analytics*, 2014. No Citations.

[27] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. No Citations.

[28] Thomas Adewale and Yusuf Musa. Mobile learning applications and student performance. *Education and Information Technologies*, 2020. No Citations.

[29] Dilani Gamage and Aruni Wijesooriya. Online learning during covid–19: A review. *Journal of Applied Learning & Teaching*, 2020. No Citations.

[30] Young Park. Ux considerations for mobile learning platforms. *Computers & Education*, 2019. No Citations.

[31] Carlos Martinez and Rohan Shah. Deep learning for adaptive educational systems. *IEEE Transactions on Learning Technologies*, 2021. No Citations.

[32] Lasse Rouhiainen. Artificial intelligence: The next digital frontier. In *Digital Innovation Conference*, 2019. No Citations.

[33] UNESCO. Ai and the future of learning, 2022. Online. No Citations.

[34] Dale Schunk. Learning theories: Educational perspective. *Pearson*, 2015. No Citations.

[35] Savita Mahar. Student engagement in adaptive e-learning. *International Journal of Emerging Technologies in Learning*, 2021. No Citations.

[36] Lennart Tack and Aaron Singh. Automated pdf text extraction and preprocessing for education. In *International Conference on Document Analysis*, 2022. No Citations.