**Project Report**

**Learning Management System**

**Submitted by**

Muhammad Younas

BSIT-20-42

Session: 2020-24

**Supervised by**

Dr. Maruf Pasha



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**BAHAUDDIN ZAKARIYA UNIVERSITY MULTAN, PAKISTAN**

**FINAL APPROVAL**

This is to certify that we have read this project report titled “**Learning Management System**” submitted by **Muhammad Younas Roll No.BSIT-20-42**and it is our judgement that this report is of sufficient standard to warrant its acceptance by Bahauddin Zakariya University, Multan for the degree of Bachelor of Science in Information Technology(BS(IT)).

***Committee:***

1. **External Examiner** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
2. **Supervisor**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Maruf Pasha**

Professor

Department of Information and Communication Technology

Bahauddin Zakariya University, Multan

1. **Head Of Department**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Maruf Pasha**

Professor

Department of Information and Communication Technology

Bahauddin Zakariya University, Multan

**DEDICATION**

**To my Loving Parents, Supportive Family, Teachers, and Friends for their valuable suggestions, guidance, and support.**

**ACKNOWLEDGMENT**

In the completion of my project, I had to take the help and guidance of some respected persons, who deserve my greatest gratitude. The completion of the final project gave me much pleasure. I would like to show my gratitude to Dr. Maruf Pasha, Professor Department of Information Technology, BZU Multanfor giving me a good guideline for the project through his valuable & fruitful consultations. I would like to expand my deepest gratitude to all those who have directly and indirectly guided me in writing this project.

**Muhammad Younas**

**BSIT-20-42**

**PROJECT BRIEF**

|  |  |
| --- | --- |
| PROJECT NAME | Learning Management System |
| ORGANIZATION NAME | Bahauddin Zakariya University, Multan |
| UNDERTAKEN BY | Muhammad Younas |
| SUPERVISED BY | Dr. Maruf Pasha |
| STARTING DATE | January 10, 2024 |
| COMPLETION DATE | May 30, 2024 |

COMPUTER USED Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

Memory: 8.00 GB RAM

OPERATING SYSTEM Windows 10 64-bit

SOURCE LANGUAGE(S) Dart

DBMS USED Firebase

TOOLS/PACKAGES VS Code, LDPlayer, Android SDK, Flutter SDK

**ABSTRACT**

This project focuses on developing a LMS application to simplify academic and administrative tasks for students in educational institutions. The portal provides a central place where students can access important information and services like course registration, transport schedule, timetable management. Key features include a personalized dashboard, and access to academic resources such as fee vouchers and account statements. The portal also makes administrative tasks like fee payments easier, improving overall efficiency and convenience for students. With a user-friendly design, the application works on various devices, ensuring students can stay connected and informed anytime, anywhere. The LMS application ultimately aims to support students in their academic journey by providing a seamless and integrated digital solution.

# TABLE OF CONTENTS

**Contents Page No.**

[*Chapter 1*](#_Toc168990509)

[INTRODUCTION](#_Toc168990510)

[1.1. Project Introduction 1](#_Toc168990511)

[1.1.1. Main Theme 1](#_Toc168990512)

[1.1.2. Scope of the Project 1](#_Toc168990513)

[1.1.3. Objectives of the Project 2](#_Toc168990514)

[1.2. Introduction to Organization 2](#_Toc168990515)

[1.2.1. Required Project Feature 3](#_Toc168990516)

[*Chapter 2*](#_Toc168990517)

[System Analysis](#_Toc168990518)

[2.1 Feasibility Study 5](#_Toc168990519)

[2.1.1. Technical Feasibility 5](#_Toc168990520)

[2.1.2. Economic Feasibility 5](#_Toc168990521)

[2.1.3. Operational Feasibility 6](#_Toc168990522)

[2.2. Data Gathering 6](#_Toc168990523)

[2.2.1. Questionnaires 7](#_Toc168990524)

[2.2.2. Sampling and Observation 7](#_Toc168990525)

[2.3. Existing System 8](#_Toc168990526)

[2.3.1. Data Analysis 8](#_Toc168990527)

[2.3.2. Data Flow Diagrams 9](#_Toc168990528)

[*Chapter 3*](#_Toc168990529)

[SYSTEM DESIGN](#_Toc168990530)

[3.1 Introduction to System Design 13](#_Toc168990531)

[3.1.1. Preliminary or General Design 13](#_Toc168990532)

[3.1.2. Structured or Detailed Design 13](#_Toc168990533)

[3.2. Proposed System and its Features 14](#_Toc168990534)

[3.2.1. Features of Proposed System 14](#_Toc168990535)

[3.3 System Design using UML 15](#_Toc168990536)

[3.3.1 Use Case Diagram 15](#_Toc168990537)

[3.3.2. Sequence Diagram 17](#_Toc168990538)

[3.4. Database Design 19](#_Toc168990539)

[3.4.1. Collection and Document Structure 19](#_Toc168990540)

[3.4.2. Schema Design 19](#_Toc168990541)

[3.4.3. Data Relations 19](#_Toc168990542)

[3.5. Database Tables 21](#_Toc168990543)

[*Chapter 4*](#_Toc168990544)

[SYSTEM DEVELOPMENT](#_Toc168990545)

[4.1. Introduction 24](#_Toc168990546)

[4.2. Tool / Language / Technology Selection Criteria 24](#_Toc168990547)

[4.2.1. Microsoft VS Code: 24](#_Toc168990548)

[4.2.2. Back End Development with Dart: 25](#_Toc168990549)

[4.2.3. Development Environment 26](#_Toc168990550)

[4.2.4. Frontend Development with Dart: 28](#_Toc168990551)

[4.3. Implementation 28](#_Toc168990552)

[4.4. Design Patterns: 29](#_Toc168990553)

[4.4. Testing the Application 29](#_Toc168990554)

[*Chapter 5*](#_Toc168990555)

[USER’S GUIDE](#_Toc168990556)

[5.1. Login Screen 32](#_Toc168990557)

[5.1.1. Forget Password Screen 33](#_Toc168990558)

[5.2. Main Dashboard 34](#_Toc168990559)

[5.3. DashBoard entities 35](#_Toc168990560)

[5.3.1. Time Table 35](#_Toc168990561)

[5.3.2. Enrollments 36](#_Toc168990562)

[5.3.3. Study Plan 37](#_Toc168990563)

[5.3.4. Fee Voucher 38](#_Toc168990564)

[5.4. User 39](#_Toc168990565)

[5.4.1. View User Profile 39](#_Toc168990566)

[5.4.2. Edit User Profile 40](#_Toc168990567)

[5.5. Contact Us 41](#_Toc168990568)

[5.6. Admin Panel 42](#_Toc168990569)

[*Chapter 6*](#_Toc168990570)

[Conclusion](#_Toc168990571)

[6.1. Conclusion 44](#_Toc168990572)

# Chapter 1

# INTRODUCTION

## Project Introduction

Students need quick and easy access to a wide range of academic and administrative services. The Learning Management System application is designed to meet this need by providing a comprehensive, user-friendly platform for students in educational institutions. This application aims to simplify and enhance the student experience by centralizing essential functions in one convenient place.

With this system, users can easily create and manage their accounts. Secure login and authentication ensure that user data remains protected. In case users forget their passwords, they can reset them with ease.

This portal allows students to manage their academic life efficiently. They can register for courses and view their class schedules all from their personalized dashboard. The portal also provides access to important academic resources, such as fee vouchers and their account statements, making it easier for students to stay on top of their studies..

The application streamlines administrative tasks like fee payments saving students time and reducing the hassle of navigating multiple systems. The responsive design allows students to stay connected whether they are on campus or off.

By integrating these features into a single platform, the Student Portal application not only enhances convenience but also supports students in achieving their academic goals.

### Main Theme

The main theme of the LMS application is to make student life easier and more organized. This user-friendly platform offers essential features like timetable management, transport information, current enrollments, study plans, fee vouchers, email, and account statements. By streamlining these tasks, the portal enhances accessibility and provides valuable insights into academic and administrative details, helping students navigate their educational journey smoothly and efficiently.

### Scope of the Project

The scope of the Learning Management System project includes the following key features and functionalities:

* **User Account Management:**
* Registration and profile management
* Secure login and authentication
* Password reset functionality for account security
* **Academic Management:**
* Adding new courses to the curriculum
* Updating course details and requirements
* Removing courses from the system
* Viewing detailed course information
* **Student Progress Tracking:**
* Monitoring current enrollments and course progress
* Identifying courses with available seats
* Tracking academic performance and grades
* **Financial Management:**
* Generating fee vouchers and managing fee payments
* Providing access to account statements for financial tracking
* **User Interface and Accessibility:**
* Designing a user-friendly interface for easy navigation
* Ensuring accessibility features for all users.

This project aims to streamline and improve the academic journey for students by providing easy access to essential tools and information, ultimately enhancing efficiency and accuracy in academic management.

### Objectives of the Project

The main goal of the LMS application is to establish a user-friendly and efficient platform that empowers students to manage their academic journey seamlessly.

## Introduction to Organization

Bahauddin Zakariya University is located in Multan, Punjab, Pakistan. It is the Largest university of South Punjab. Bahauddin Zakariya University was formerly known as Multan University. It was renamed in honour of Hazrat Baha-ud-din Zakariya (RA). Bahauddin Zakariya University has **10** Faculties, **2** Sub-Campuses (Lodhran & Vehari) and **80** Departments and it is offering **94** Undergraudate Programs, **69** Graduate(MS) Programs, **48** Graduate (PhD) Programs, **36** BS(5th Semester) Programs, **5** ADP Programs, **13** Diploma Programs and **13** Short Courses. There are total **646** faculty members, in which **494** faculty members are PhD degree holders.

### Required Project Feature

1. **Provide user managament**
2. The system shall allow the users to create their accounts with email.
3. The system shall allow the user to securely login to their accounts.
4. The system shall allow users to reset their passwords in case they forget it.
5. The system shall allow users to update their passwords if they want
6. **Course Management**
7. Users can add different courses to the system.
8. Users can update or modify information for selected courses.
9. Users can remove outdated courses from the system.
10. **User Profile Management**
11. The system shall maintain a user profile and display all necessary information.
12. The system shall allow the user to update his/her profile.
13. **Mobile-Based Application**
14. The application is accessible through mobile devices with internet connectivity.
15. Basic mobile device usage skills are required to navigate the application. The interface is designed to be user-friendly, minimizing the learning curve for users.
16. **Fee Management**
17. System should Maintain detailed records of each student's fee payments, including due dates, amounts paid, and outstanding balances.
18. System should allow students to pay fees online through secure payment gateways, making the payment process convenient and efficient.
19. System should easily set up and manage different fee structures for various courses, programs, or student categories, providing flexibility and accuracy.

# Chapter 2

# System Analysis

## Feasibility Study

A feasibility report is a document that assesses the practicality and viability of a proposed project or system. It provides detailed analysis and evaluation to determine whether the project is worth pursuing and helps decision-makers understand the potential benefits, risks, costs, and impacts associated with it. System feasibility is accessed in three principal ways:

* Technical
* Economical
* Operational

### Technical Feasibility

The purpose of this technical feasibility report is to evaluate the practicality of developing and implementing a Learning Management System (LMS). This system aims to streamline educational processes, enhance user account management, and provide real-time tracking of course progress and student performance.

The Learning Management System will include features such as user account management, secure authentication, password reset functionality, course management (add, update, delete, view), progress tracking, and performance analytics. The system will leverage modern mobile technologies to ensure a user-friendly interface and robust functionality.

Technical feasibility is often the most challenging area to assess because the objectives and performance requirements can be somewhat unclear. The analysis process must be conducted in parallel with the assessment of technical feasibility. Fortunately, the necessary equipment and software tools are currently available in the market.

### Economic Feasibility

A Learning Management System (LMS) is economically feasible if its benefits outweigh the costs of development and operation. Despite the initial investment, the LMS will bring significant financial advantages through increased efficiency, improved accuracy, and better decision-making in education. Although the return on investment may take time, the long-term gains in streamlined administration and enhanced learning experiences make it a worthwhile investment.

### Operational Feasibility

The operational feasibility of the Learning Management System (LMS) is strong, as it will integrate seamlessly with existing educational processes at Bahauddin Zakariya University (BZU). The system's user-friendly interface ensures that students, teachers, and administrative staff can quickly learn and adapt to using it, minimizing the need for extensive training. By automating key academic tasks, the LMS will streamline workflows, enhance accuracy, and improve overall efficiency in managing courses, assignments, and communication. Technical support and regular maintenance plans will ensure smooth operation and minimal downtime, making the system a practical and effective solution for the university.

## Data Gathering

“Data gathering is the foundation of knowledge; it transforms raw information into actionable insights.”.

The approach and manner in which information is gathered require a person with sensitivity, common-sense, and knowledge of what and when to gather and what channel to use in securing the information. Before one determines where to go for information or whare to look, the first requirement is to check out what information to gather.

* **Information about Firm**

Understanding an organization's rules, what it aims to achieve, how it's set up, and how it runs helps us get a clear picture of what it's like to work there. Rules tell us how things are done, goals show us what the organization wants to achieve, and its structure tells us how it's organized. We can learn about these things from things like documentation review, Interview and surveys, observatios and collaboration with university stakeholders.

#### Origin of Information

We collect information from two main places: stuff that's inside the organization and things from outside it. Inside, we look at stuff like:.

The primary internal sources are:

* University Records
* Faculty and Staff
* Student feedback
* University commitees

Outside the organization, we might talk to the vendors who work with the organization.

#### Information Gathering Tools

The analyst needs to choose the right tools to collect information and decide how to use them. There aren't strict rules about which tools to use, but the most important thing is to get accurate information. Here are some tools they can use:

* Interviews and Questionnaires.
* On-site Observation.
* Collaborative Workshop

### Questionnaires

A questionnaire is a form with a bunch of questions, usually given to a lot of people to get information for a survey. Unlike interviews, questionnaires are more stress-free and easy to get done. By its nature it has the following advantages:

* They're cheaper and easier for the person running the survey compared to interviews.
* While interviews talk to one person at a time, questionnaires can be given to many people at once.

Questions in a questionnaire are asked in the same way and same order and they are answered in that same order, making them uniform and consistent. In contrast to the interview, things go different from interview to another.

### Sampling and Observation

Another way to gather information in system studies is through observation. Observation means watching and paying attention to people, things, and events to gather information. The main goals of observation is to understand the real system being studied as closely as possible. That's why it's important for the system analyst to know a lot about how the system

works and what it does. Through observation, system analyst can get systematically correct information with any kind of mix up

* **Natural or Contrived**

In natural observation, the observer watches people in their usual environment like at their workplace. In contrived observation, the observer creates a situation, like in a lab. Observations can also be obtrusive or unobtrusive. Obtrusive means the person being watched knows they're being observed, while unobtrusive means they might not know, like when the observer is behind a one-way mirror.

* **Direct or Indirect**

In a direct observation, the analyst watches the subject or system in action. In an indirect observation, the analyst uses tools like cameras and videotapes to capture information instead of watching directly.

## Existing System

The current system relies on manual processes for managing learning resources, which presents several challenges. These include missing data, and errors in aggregated results. Moreover, there's a lack of real-time tracking and reporting on key statistics related to course enrollment, progress tracking, and student performance.

### Data Analysis

Data analysis is the process of examining raw data to make it organized and useful. It's like cleaning up a messy room to find what you need. By sorting through the data, we can extract valuable information that helps us understand trends, make decisions, and solve problems. This organized information gives us insights into how things are working and what changes might be needed to improve efficiency or effectiveness.

In businesses, data analysis is crucial for evaluating whether systems are doing their job effectively. It helps determine if data is being protected properly, if operations are running smoothly, and if overall goals are being achieved. By analyzing data, organizations can identify areas for improvement and make changes to enhance performance and meet objectives.

Several steps are involved in the analysis stage while designing a database. Some of these steps are given below:

* Feasible Study
* Requirement Analysis
* Project Planning
* Data Analysis

### **Data Flow Diagrams**

The Data Flow diagram is a visual way to show how data moves through a computer system. It's like drawing a map of how information travels from one place to another. This diagram helps us understand the process of the system and is often the first step in planning how to build it. Later, we can add more details to the diagram to get a clearer picture of how everything works together.

**Types of DFD:**

1. Context DFD
2. Level 1 DFD
3. Detail DFD

#### Context DFD

In the context of a Data Flow diagram, we focus on how the system interacts with its surroundings. The entire system is shown as a bubble, while external interactions are represented by squares where data comes in and goes out. This type of diagram doesn't show the inner workings of the system itself; it's more about how the system connects with the world around it.

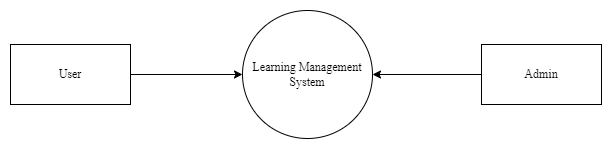


Fig 2. 1 Context DFD

#### Level 1 DFD

#### Level 1 DFD is a picture of the system itself. It shows the main processes, external parts, where data is kept, and how data moves around.

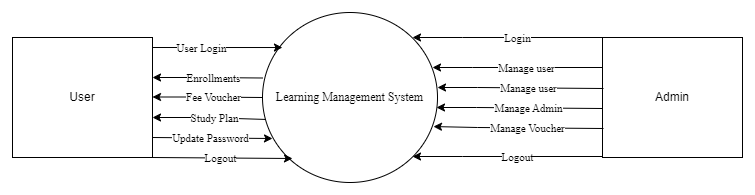


Fig 2. 2 Level 1 DFD

#### Detail DFD

A detailed Data Flow Diagram (DFD) is like a detailed map of a system. It shows all the little parts of the system, like every step of a process, where data comes from and goes to, and how everything connects together. It's like zooming in on a big picture to see all the small details of how things work.

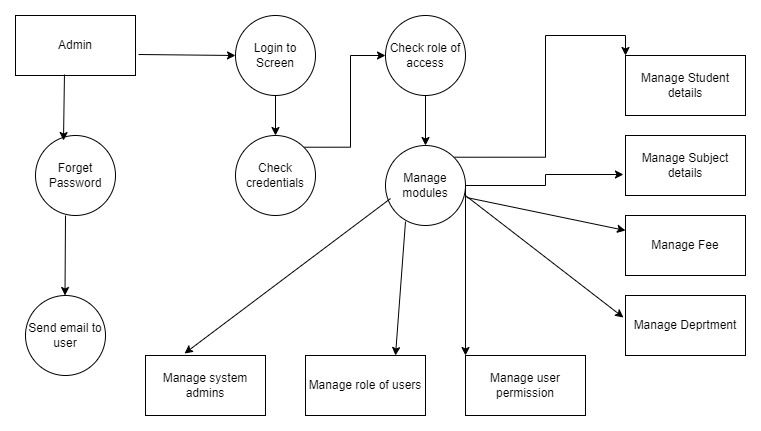


Fig 2. 3 Detail DFD

# Chapter 3

# SYSTEM DESIGN

## Introduction to System Design

System design plays a crucial role in developing an effective Learning Management System (LMS). It involves creating a blueprint for how the system will operate, including its structure, components, and interactions. This ensures that all parts of the system work together seamlessly to meet the needs of users.

In our project, system design focuses on key areas such as data management, user interfaces, and real-time account statements. By carefully planning each aspect, we aim to create a system that helps users easily manage courses, track progress, and enhance educational experiences.

Data management is fundamental to our LMS. We'll design a robust database to store information about courses, including titles, descriptions, instructors, schedules, and enrollment details. This database ensures that data is well-organized and accessible, enabling accurate tracking and reporting.

Our user interfaces are designed to be intuitive and user-friendly. They allow users to enroll in courses, track their progress, and interact with course materials.

By focusing on these core aspects, our system design aims to create a comprehensive and user-friendly Learning Management System that enhances the educational journey for students and faculty alike.

Normally, the design proceeds in two stages.

* + - Preliminary or General Design
    - Structured or Detailed Design

### Preliminary or General Design

In the initial design phase, we outline the features of the new system. We also estimate the costs of implementing these features and the benefits they will provide. If the project seems feasible, we then proceed to the detailed design stage.

### Structured or Detailed Design

In the detailed design stage, the system's design becomes more organized. This stage involves creating a structured blueprint of the computer system, which includes the same components and relationships as the original problem. Detailed specifications for input, output, forms, and processing are developed.

## Proposed System and its Features

The proposed Learning Management System (LMS) aims to streamline the management of courses, making it easier and more efficient for users. It includes a database to store comprehensive information about each course, including titles, descriptions, instructors, schedules, and enrollment details. The system features an intuitive interface, allowing users to easily enroll in courses, track their progress, and interact with course materials. This system ensures that all aspects of course management are handled smoothly and effectively, enhancing the educational experience for students and instructors alike.

### Features of Proposed System

The proposed system includes many features, such as:

* **Efficiency**

Efficiency means making the best use of resources to achieve a goal. The new system will be more efficient and easier to use than the current system.

* **Data Security**

The data needed for decision-making is very sensitive. The proposed system will provide secure and reliable services to protect this data.

* **Accuracy**

The system will deliver precise and accurate information necessary for decision-making, ensuring efficient and accurate record-keeping.

* **Reliability**

Due to its security and accuracy, the new system will be more reliable than the existing one, allowing for timely decision-making.

* **Minimize Redundancy**

The new system is designed to minimize data duplication, reducing redundancy in files.

* **User Friendly**

The system will be easy to use, allowing users to interact with it effortlessly without needing specialized training.

* **Comprehensive Database**

The proposed system will feature a comprehensive database, enabling users to insert, modify, and delete data easily.

## 3.3 System Design using UML

System design using UML (Unified Modeling Language) is like drawing a picture of how a system works. UML uses different symbols and diagrams to represent different parts of the system, like its structure, behavior, and interactions. It helps developers plan and communicate about how the system will be built and how it will work. UML diagrams show things like what the system will do, how different parts will work together, and what it will look like. Overall, UML is a helpful tool for designing and understanding complex systems.

UML allows us to develop several diagrams to demonstrate the overall structure of our software. Such as:

* Use Case Diagram
* Sequence Diagram

### 3.3.1 Use Case Diagram

Use case diagrams are like maps that show how users interact with a system. They use simple symbols to represent different actions or tasks that users can do. Each action is called a "use case." These diagrams help developers understand what users need from the system and how it should respond to their actions. By using use case diagrams, developers can plan and design systems that meet users' needs effectively.

There will be two types of actors in Learning Management System.

* User
* Admin

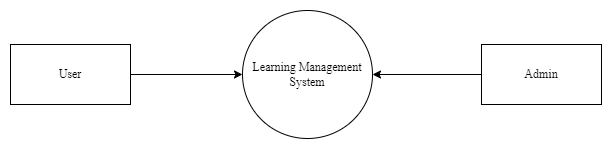


Fig 3. 1 Interaction of Learning Management System

#### 3.3.1.1 Use Case Diagram:

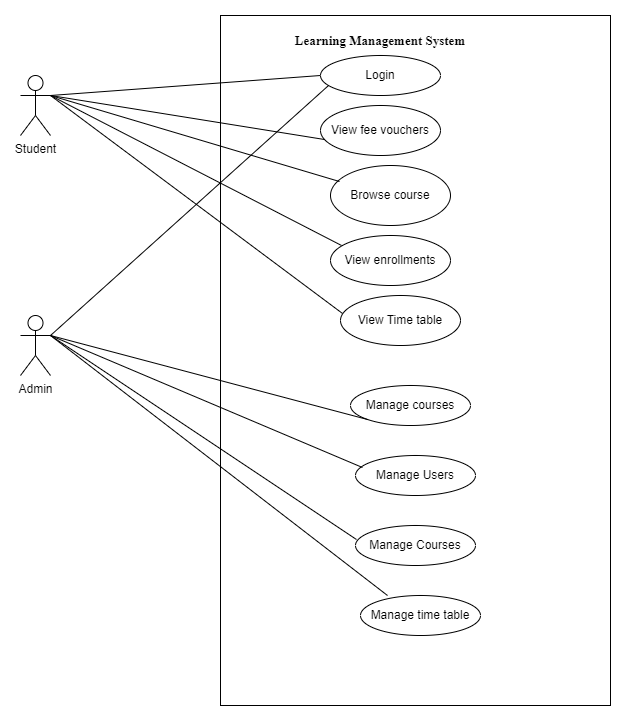


Fig 3. 2 Use Case Diagram

### Sequence Diagram

Sequence diagrams are a type of UML (Unified Modeling Language) diagram that show how objects interact with each other in a specific sequence to achieve a particular goal. They are used to visualize the flow of messages between different parts of a system over time. Each participant in the sequence diagram is represented by a vertical lifeline, and the interactions between these participants are shown as horizontal arrows. These arrows are labeled with the messages or actions being performed.

#### Login Sequence Diagram

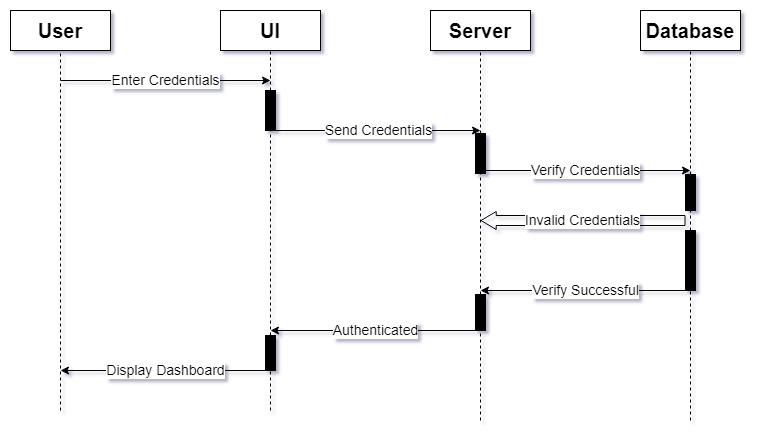


Fig 3. 3 Sequence Diagram for Login

#### Detail Sequence Diagram

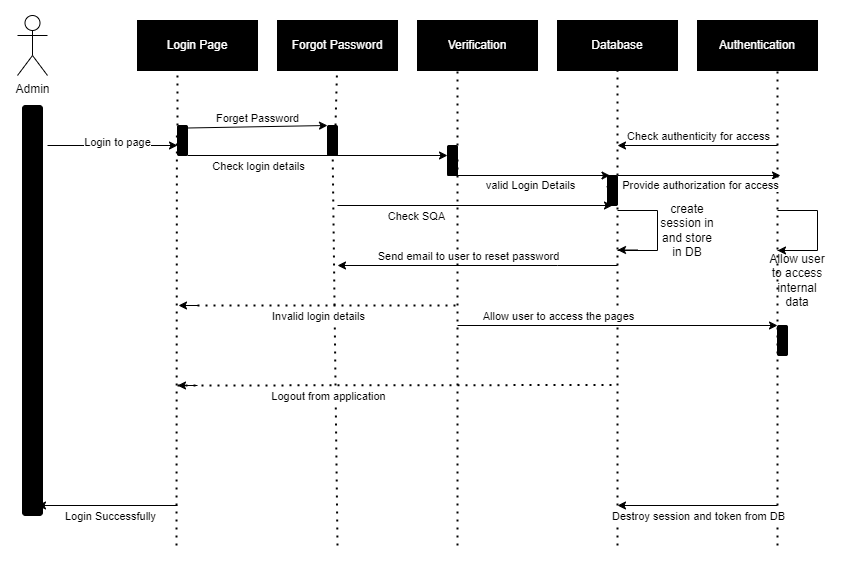


Fig 3. 4 Detail Sequence Diagram

## Database Design

Database design is a critical phase in crafting a successful Learning Management System (LMS). It entails structuring the database to efficiently and securely store, organize, and manage data. For our project, we've opted for Firebase Realtime Database, a NoSQL database renowned for its versatility, scalability, and user-friendliness. Unlike traditional relational databases, Firebase stores data in JSON format, facilitating dynamic and adaptable schema design. This flexibility proves invaluable for managing diverse data types and accommodating evolving data needs without extensive database restructuring. Leveraging Firebase Realtime Database ensures our LMS is robust and adaptable, adept at efficiently managing the varied data associated with courses, users, and interactions.

### Collection and Document Structure

In Firebase Realtime Database, data is organized into collections and documents. Each collection acts as a container for related data, similar to tables in traditional databases but more flexible. Documents within collections are JSON-like objects with key-value pairs, allowing for dynamic data modeling. For our Learning Management System (LMS), we'll have collections for courses, users, and interactions. Each course document will contain fields like ID, title, instructor, schedule, and enrollment details. This setup simplifies data retrieval and allows for easy modification of fields as needed, without major schema changes.

### Schema Design

Even though Firebase Realtime Database is schema-less, establishing a structured design is vital for maintaining data consistency and integrity within our Learning Management System (LMS). In this, we'll define structured collections for entities such as courses, users, and interactions. For example, the course collection will include documents with fields like course ID (unique identifier), title, instructor (an object containing name and email fields), schedule (date and time), and enrollment details. Similarly, the user collection will contain documents with fields such as user ID (unique identifier), name, email, role, and enrolled courses. This schema ensures that data is organized consistently, facilitating efficient data management and retrieval within the LMS.

### Data Relations

In Firebase Realtime Database, data relations play a crucial role in managing interactions between users, courses, and instructors. Users can enroll in multiple courses, forming a simple connection where each course document stores a list of enrolled users. Additionally, instructors can teach multiple courses, establishing a straightforward link between courses and instructors. This structured approach ensures smooth navigation and efficient data management within our LMS, enhancing the overall user experience.

* **Entity Relation Diagram**

An Entity-Relationship Diagram (ERD) serves as a visual guide to showcase the connections between different elements within a database system. It aids in crafting the database structure by illustrating how entities such as courses, users, and enrollments are interlinked. Each entity is depicted as a rectangle, containing its attributes, while relationships are represented by lines connecting these rectangles. ERDs are invaluable for grasping the data model, ensuring data consistency, and streamlining database design. In our Learning Management System (LMS) project, the ERD sheds light on the relationships between users, courses, and enrollments, providing a comprehensive framework for the database schema.

The elements of ERD are:

1. Entities
2. Relationship
3. Attributes

Steps involved in creating an ERD include:

1. Determining all interaction between entities.
2. Analyzing the nature of interaction/ determine the cardinality of the relationship.
3. Creating the ERD.
4. Identifying and defining the entities

Here’s the basic structure of ER diagram:

**User**

* userId: String
* name: String
* email: String
* role: String
* profilePicture: String
* courses: [courseId]

**Course**

* courseId: String
* title: String
* description: String
* enrolledStudents: [userId]

## Database Tables

In our Learning management system, data is stored as JSON-like objects in a hierarchical structure, where each object is represented by a key-value pair. These objects are stored in a single, flat JSON tree, rather than separate collections. Each key represents a unique path in the database, allowing for efficient data retrieval and manipulation.

There are following documents in the database of my project:

**Document: Users**

|  |  |  |
| --- | --- | --- |
| **Key** | **Type** | **Constraints** |
| User | String | required |
| Name | String | Required |
| Blood group | String | required |
| Birth Place | String | required |
| CNIC | String | required |
| Gender | String | required |
| image | Object |  |

**Document**: **Admin**

|  |  |  |
| --- | --- | --- |
| **Key** | **Type** | **Constraints** |
| AdminId | String | required |
| Name | String | required |
| Number | String | required |
| Password | String | required |

**Document**: **Voucher**

|  |  |  |
| --- | --- | --- |
| **Key** | **Type** | **Constraints** |
| feeID | String | required |
| Title | String | required |
| Challan No | String | Required |
| Amount | String | required |
| Date | String | required |

# Chapter 4

# SYSTEM DEVELOPMENT

## Introduction

The Learning Management System project is developed using the FLUTTER, which includes FirebaseDB, Dart and Java. This system is designed to help efficient management of educational resources, student data, and course materials. Once the system is proposed and the analysis and design phase has been accomplished, the designer moves towards the development phase of the software in accordance with the proposed system and design phase specifications because we are now ready to transform the theoretical concepts into a tangible and functional solution.

The project includes a robust backend that handles data storage, API endpoints, as well as a user-friendly frontend that provides an intuitive interface for users to interact with the system. Throughout the development process, various tools and libraries were used to ensure the application is responsive, secure, and easy to use. This chapter details the system development process, highlighting the tools, technologies, and methodologies employed to create the Learning Management System.

## Tool / Language / Technology Selection Criteria

The selection of suitable tools is paramount for the effective development of the "Learning Management System." To ensure an informed decision, various criterias were evaluated to assess the appropriateness of the tools, including Firebase for the database, Java for the backend server. These selections were made based on factors such as scalability, performance, community support, and alignment with project requirements.

### Microsoft VS Code:

Features of Visual Studio Code for Application Development:

1. **Integrated Development Environment (IDE):** Visual Studio Code is a helpful tool for writing code. It has many features that make coding easier. It has built-in support for many different languages.
2. **Language Support:** Visual Studio Code works with many programming languages like Dart, Java, and more. It helps by highlighting code and offering helpful suggestions while typing. It also provide thoughtful code snippets.
3. **Extensions Marketplace:** Visual Studio Code has many add-ons called extensions. These extensions add extra features to the editor, making it more useful for different tasks.
4. **Crystal Integrated Terminal:** Visual Studio Code has a built-in terminal. This lets you run commands hassle free and do tasks without leaving the editor.
5. **Version Control Integration:** Visual Studio Code works well with version control systems like Git. It has built-in support for Git so developers can easily host their code on Github right from VS Code. This helps you keep track of changes in your code.
6. **Debugging Capabilities:** Visual Studio Code helps you find and fix mistakes in your code. It has tools for checking your code while you're writing it in real-time.
7. **Task Automation:** Visual Studio Code can do tasks for you, like running tests or deploying your code.
8. **Live Share Collaboration:** Visual Studio Code lets you work with others in real-time. You can share your code and edit it together.
9. **Lightweight and Customizable:** Visual Studio Code is easy to use and can be modified to fit your needs. You can change things like the theme and shortcuts to make it work better for you.
10. **Hot Reload:** Hot reload quickly see the changes you make in your code reflected in your app without restarting the app, speeding up the development process.
11. **Hot Restart:** Hot restart reloads the entire application, losing the app's current state. It’s useful when you make changes to the main structure of your app or need to reset its state. Unlike a full restart, it doesn't require stopping and starting the app from scratch, making the development process faster.

These advantages collectively make Visual Studio Code an excellent choice for application development across various programming languages and frameworks. With its user-friendly interface, extensive language support, and rich ecosystem of extensions, Visual Studio Code offers developers a comprehensive development environment. Its seamless integration with version control systems and debugging tools enhances productivity and streamlines the development process. Additionally, features like Live Share collaboration and task automation contribute to efficient teamwork and code management. Visual Studio Code's lightweight nature and customizable settings further enhance its appeal, making it an indispensable tool for developers seeking a versatile and efficient coding experience.

### Back End Development with Dart:

Dart is selected for the project due to the following advantages:

1. **Object-Oriented:** Dart is an object-oriented language, meaning it supports principles like encapsulation, inheritance, and polymorphism. This allows developers to create modular and reusable code structures.
2. **Strongly Typed:** Dart is statically typed, which means variables must have their types explicitly declared. This helps catch type-related errors at compile-time, improving code reliability and readability.
3. **Just-In-Time (JIT) Compilation:** Dart supports both JIT and Ahead-of-Time (AOT) compilation. JIT compilation allows for fast development and iteration cycles by compiling code on-the-fly during runtime, enabling features like hot reload in Flutter.
4. **Garbage Collection:** Dart includes automatic memory management through garbage collection, which helps manage memory usage and prevents memory leaks by deallocating unused objects.
5. **Asynchronous Programming:** Dart provides built-in support for asynchronous programming through features like async/await and Futures.
6. **Cross-Platform Compatibility:** Dart is designed to be platform-independent, allowing developers to write code that runs on various platforms, including web browsers, mobile devices (via Flutter), and servers (via Dart VM).
7. **Community and Ecosystem:** Dart has a growing community of developers and a vibrant ecosystem of libraries, packages, and frameworks. This provides developers with resources and support to build robust and scalable applications.
8. **Tooling Support:** Dart has excellent tooling support with features like an integrated development environment (IDE) with debugging, code completion, and testing tools that enhance developer productivity.

Dart's combination of features, including strong typing, asynchronous programming support, and cross-platform compatibility, makes it a versatile and efficient language for building a wide range of applications, particularly for Flutter development.

### Development Environment

The development environment plays a crucial role in the creation of any software project, including the Learning Management System. In this section, we'll explore the various components and configurations of the development environment used for building and testing the application.

1. **Visual Studio Code:** Android Studio serves as the primary integrated development environment (IDE) for coding the Learning Management System. Android Studio provides a comprehensive set of features and tools to support the entire Android app development lifecycle, from initial design to testing and deployment, making it the preferred choice for millions of developers worldwide.
2. **Version Control: Git and GitHub:** Git, along with GitHub for remote hosting, is used for version control in the development process. Git enables collaborative development by allowing multiple developers to work on the project simultaneously, track changes, and manage code revisions effectively. GitHub serves as a centralized repository for storing and sharing the project's source code, providing features like pull requests, issue tracking, and collaboration tools.
3. **Dart:** Dart is combination of features, including strong typing, asynchronous programming support, and cross-platform compatibility, makes it a versatile and efficient language for building a wide range of applications, particularly for Flutter development.
4. **FirebaseDB:** Firebase Realtime Database is the database technology used for storing and managing the application's data. As a NoSQL database, Firebase offers flexibility, scalability, and real-time synchronization advantages over traditional relational databases. It is well-suited for handling JSON data structures, making it an ideal choice for the dynamic data requirements of the Learning Management System.
5. **Flutter SDK:** Flutter SDK provides a rich set of features for building cross-platform mobile applications. With its hot reload capability, developers can quickly iterate on app designs, enhancing productivity. Flutter's widget-based architecture and customizable UI components enable the creation of beautiful and responsive user interfaces. Additionally, Flutter offers native performance and access to platform-specific APIs, ensuring high-quality app experiences across iOS and Android devices.
6. **Android SDK:** The Android SDK offers comprehensive tools and resources for building native Android applications. Developers can access a wide range of APIs and libraries for tasks like UI design, data storage, and networking. With Android Studio as the official IDE, developers benefit from powerful debugging and testing tools, accelerating app development.

By leveraging these components and configurations in the development environment, developers can effectively collaborate, write code, manage dependencies, interact with databases, build user interfaces, implement backend logic, and test the Learning Management System, ultimately leading to the successful development of a robust and functional application.

### Frontend Development with Dart:

For front-end development of Flutter applications, Dart language is used. Dart is known for its simplicity and efficiency, making it suitable for building user interfaces in Flutter. Its features include a hot reload capability for quick iterations, strong typing for catching errors early, and support for asynchronous programming for handling tasks like network requests seamlessly. Dart also offers a widget-based architecture, allowing developers to create custom UI components and build beautiful, responsive interfaces. Overall, Dart makes front-end development in Flutter intuitive and productive, empowering developers to create engaging user experiences for mobile applications.

## Implementation

The implementation phase of the Learning Management System involved turning the design and plans into a working software solution. This phase included setting up the development environment, writing code for the backend and frontend, integrating different parts of the system, and thoroughly testing everything to make sure it worked as expected.

1. **Backend & Frontend Development with Dart:**

We built the backend & frontend of the Learning Management System using Dart and Firebase as the database. Firebase helped us store user profiles, courses, and more, and kept everything synchronized in real-time. . Each feature, such as course displays, search functionalities, and user authentication, was developed as a separate widget. We set up user authentication and organized data neatly in Firestore. With Cloud Functions, we managed tasks like course management and notifications. We made sure everything was secure and protected sensitive data with rules. Firebase's powerful features made sure our system ran smoothly and responded quickly to user actions.

1. **Styling:**

Flutter provides a wide range of built-in widgets with customizable properties like color, font size, padding, margin, and more. You can use these properties to style individual widgets according to your design requirements.

## Design Patterns:

I have used MVC design pattern for this application. MVC stands for Model-View-Controller, which is a design pattern used in software engineering to separate the concerns of data (model), user interface (view), and the control logic (controller). This separation facilitates modularity and enables the independent development, testing, and maintenance of each component.

1. **Model:** Represents the data and the business logic of the application. It directly manages the data, logic, and rules of the application. The model notifies the view of any data changes so that the display can be updated. Following are the models we have used in our application:
2. Admin model
3. Campus model
4. Contact us model
5. Course model
6. Department model
7. Fee model
8. **View:** Represents the user interface of the application. It displays the data from the model to the user and sends user commands to the controller. The view is only concerned with the presentation of the data. There are two views in our application
9. User view
10. Admin view
11. **Controller:** Acts as an intermediary between the model and the view. It receives user input from the view, processes it (possibly updating the model), and returns the results to the view. The controller is responsible for handling the application's logic and user interactions.

## Testing the Application

Testing is a crucial part of the software development cycle. To ensure our Learning Management System works properly, we conducted testing throughout the development process. We tested the software on different operating systems to ensure it performed consistently across various platforms.

We carried out functional testing to check if all the features of the application worked as intended. This include testing various functionalities, such as course management, user authentication, and content access control. We carefully examined features like creating, updating, and deleting courses, ensuring seamless user authentication and authorization processes.By simulating different user actions, we ensured the system behaved correctly in all scenarios.

Unit testing was performed to validate individual components of the application. Each part, such as the backend , frontend components, and database operations, was tested separately. This helped us catch and fix bugs early in the development process, ensuring each component functioned correctly on its own.

Integration testing was done to ensure that different parts of the application worked well together. We tested the interaction between the frontend and backend, ensuring that data was correctly fetched, displayed, and updated. This was essential for verifying that the system provided a seamless user experience.

By thoroughly testing the Learning Management System, we ensured it was functional, reliable, and user-friendly, providing a positive experience for all users.

# Chapter 5

# USER’S GUIDE

## Login Screen

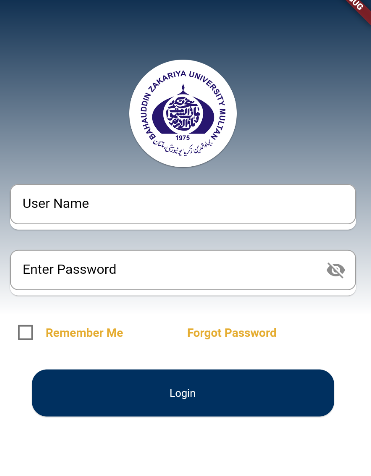
****

Fig 5. 1 Login Screen

### Forget Password Screen

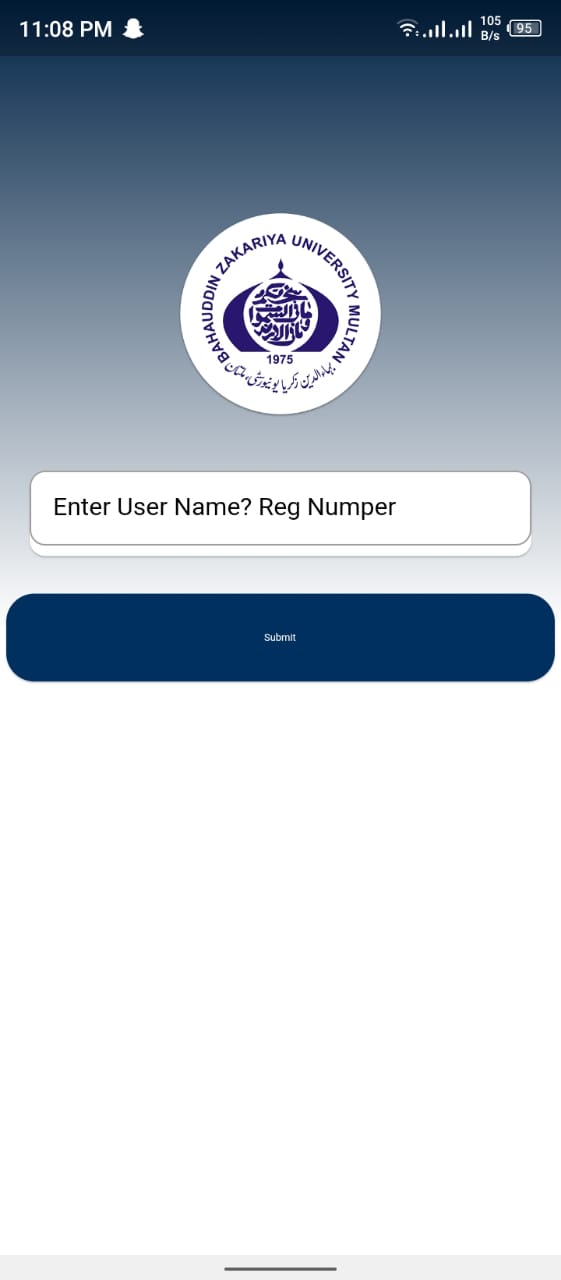


Fig 5. 2 Forget Password

## Main Dashboard

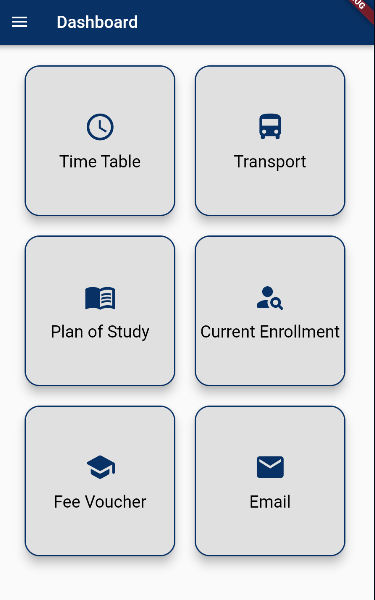


Fig 5. 3 Main Dashboard

## DashBoard entities

### Time Table

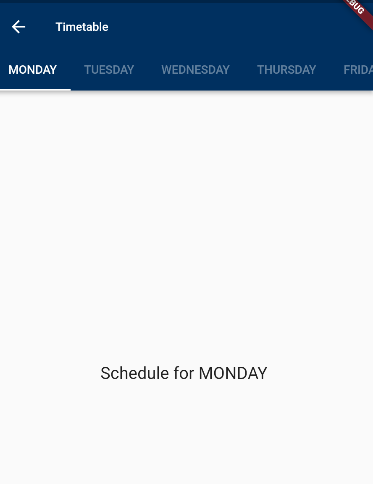


Fig 5. 4 Time Table

### Enrollments

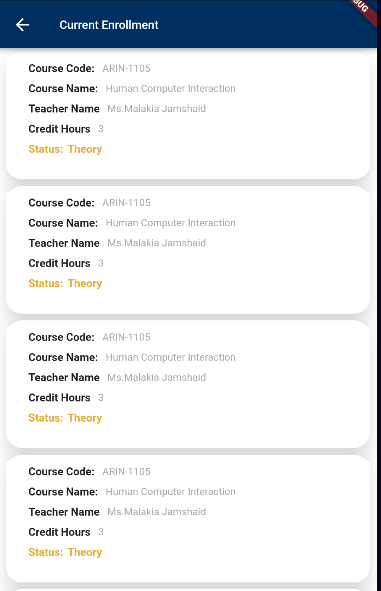


Fig 5. 5 Current Enrollments

### Study Plan

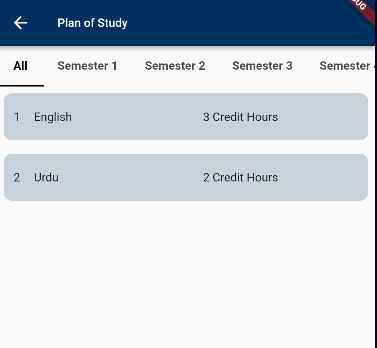


Fig 5. 6 Study Plan

### Fee Voucher

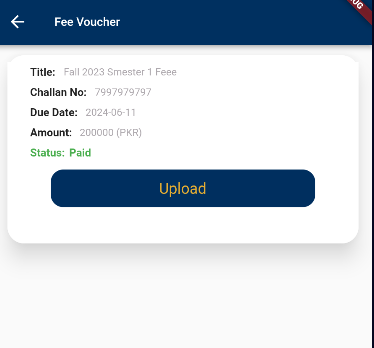


Fig 5. 7 Fee voucher

## User

### View User Profile

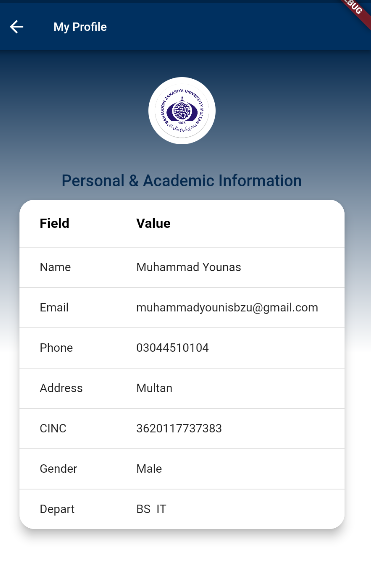


Fig 5. 8 User Profile

### Edit User Profile

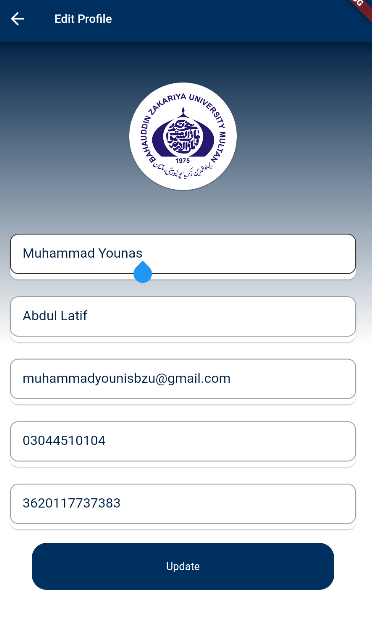


Fig 5. 9 Edit User Profile

### Contact Us

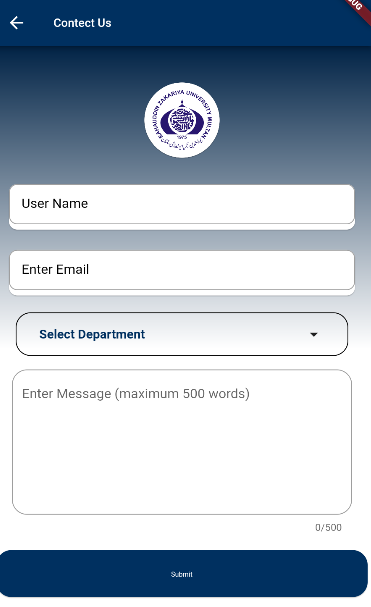


Fig 5. 10 Contact Form

### 5.6. Admin Panel

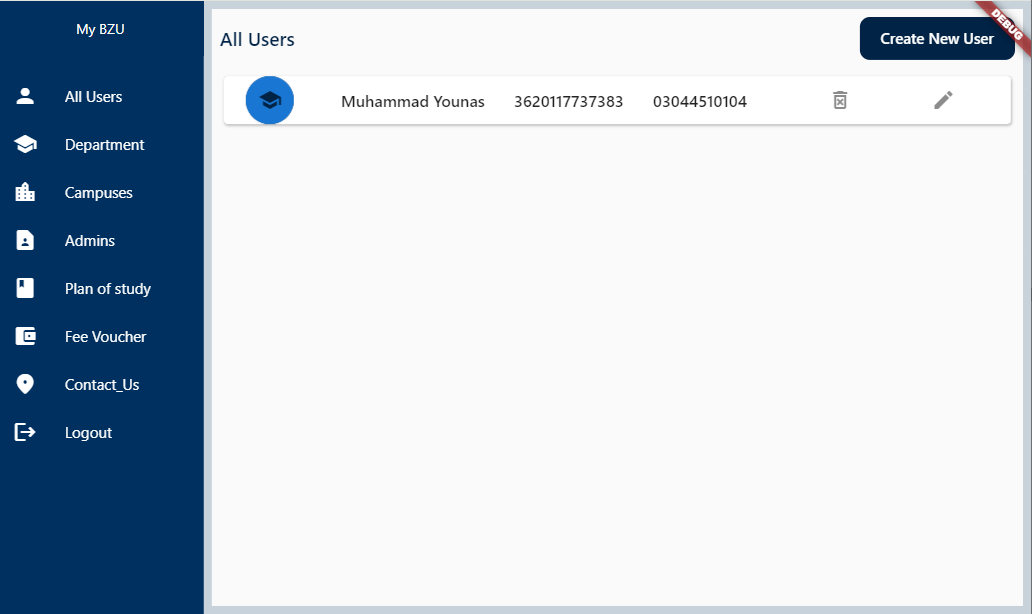


Fig 5. 11 Admin Panel

# Chapter 6

# Conclusion

## Conclusion

In conclusion, the Learning Management System offers a user-friendly solution for managing educational resources and facilitating effective learning experiences. This comprehensive platform, developed using modern technologies like Flutter, provides essential features such as course management, user authentication, and content access control. It empowers educators to create engaging courses and enables learners to access educational materials easily.

With functionalities like course creation, user enrollment, and content organization, the system simplifies the process of delivering educational content. The intuitive interface and administrative dashboard allow educators to monitor and manage course activities efficiently, ensuring a smooth learning experience for all users.

Utilizing innovative tools and methodologies, the Learning Management System demonstrates the transformative power of technology in education. By enhancing accessibility, improving engagement, and fostering collaboration, this project paves the way for future advancements in online learning, driving positive outcomes for both educators and learners alike.