

# **GU Course Folder Management System**



**Bachelor of Science in Information Technology**

**Muhammad Safiullah**

**2020-GU-0164**

**Department of Computer Science & IT  
Ghazi University, Dera Ghazi Khan, Pakistan.**

**2020-2024**

# **GU Course Folder Management System**



This project report is submitted to Department of Computer Science & IT, Ghazi University, Dera Ghazi Khan in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology (BSIT).

**Submitted by:** Muhammad Safiullah

**Registration No.** 2020-GU-0164

**Supervised by:** Dr. Muhammad Abubakar Siddique

**Department Computer Science & IT**

**Faculty of Sciences**

**Ghazi University, Dera Ghazi Khan, Pakistan.**

**September, 2024**

## **STUDENT'S DECLARATION**

I **Muhammad Safiullah** having registration no. **2020-GU-0164** hereby state that this project report entitled "***GU Course Folder Management System***" is my own work and has not been submitted previously for any degree from Ghazi University, Dera Ghazi Khan, or elsewhere in the country/world. If my statement is found incorrect even after graduation, the university has the right to take action against me as per approved rules and regulations of the University.

---

Signature of the candidate

## **FINAL APPROVAL**

This is to certify that we have read this report submitted by **Muhammad Safiullah** and it is my judgment that this report is of sufficient standard to warrant its acceptance by Ghazi University, Dera Ghazi Khan for the degree of **BS (Information Technology)**.

### **Committee:**

#### **1. External Examiner** \_\_\_\_\_

Mr. Professor

Department of CS & IT

Ghazi University, D.G. Khan

#### **2. Supervisor** \_\_\_\_\_

Dr. Muhammad Abubakar Siddique

Associate Professor

Ghazi University, D.G. Khan

#### **3. Head of Department** \_\_\_\_\_

Dr. Hafiz Gulam Ahmad Umar

Associate Professor

Ghazi University DG Khan

## **ACKNOWLEDGMENTS**

I would like to express my deepest gratitude to my supervisor, **Dr. Muhammad Abubakar Siddique**, for his invaluable guidance, support, and encouragement throughout the research and composition of this thesis. His expertise and insights have been instrumental in shaping this work.

I would also like to thank my teachers and mentors at the Department of Computer Science & IT, Ghazi University, Dera Ghazi Khan, for their continuous support and for providing a strong foundation in my field of study.

A special thanks to my colleagues and friends for their unwavering support, encouragement, and constructive feedback throughout this journey.

Lastly, I am profoundly grateful to my family, especially my parents, for their endless support, patience, and belief in my abilities, which have motivated me to persevere and complete this work.

**Muhammad Safiullah**

## ABSTRACT

The **GU Course Folder Management System** is an advanced web-based application crafted with the Laravel framework to transform and enhance the administrative and academic processes within educational institutions. This system is designed to address the inefficiencies and challenges associated with traditional course management methods, which often rely on manual tracking and extensive paperwork. By offering a centralized digital platform, the system streamlines various academic tasks and processes, providing significant improvements in efficiency and accuracy.

The system is divided into two primary panels: the Admin Panel and the Teacher Panel. The **Admin Panel** equips administrators with comprehensive tools to manage essential academic entities, including the ability to add, delete, and edit records for administrators, teachers, students, classes, and subjects. It also facilitates the assignment of classes and subjects to teachers, thus simplifying the administrative burden and ensuring an organized academic structure.

On the other hand, the **Teacher Panel** is tailored to support teachers in their daily responsibilities. It allows teachers to view their assigned classes and subjects, mark daily attendance with ease, and upload critical academic resources such as semester assignments, quizzes, past papers, and their answer sheets. This functionality not only aids in efficient classroom management but also enhances the accessibility and organization of academic materials.

By integrating these features into a single platform, the GU Course Folder Management System aims to eliminate the challenges posed by manual processes, improve the accuracy of academic records, and enhance communication between administrators and teachers. The system's design emphasizes user-friendliness, scalability, and adaptability, making it a valuable tool for any educational institution seeking to modernize and optimize its academic management practices.

**Keywords:** GU Course Folder Management System, Laravel framework, web-based application, administrative processes, academic processes, educational institutions, centralized digital platform, Admin Panel, Teacher Panel.

## TABLE OF CONTENTS

STUDENT'S DECLARATION .....	i
FINAL APPROVAL .....	ii
ACKNOWLEDGMENTS .....	iii
ABSTRACT .....	iv
LIST OF FIGURES .....	viii
<b>CHAPTER 1.....</b>	<b>1</b>
1.1.    Introduction.....	2
1.2.    Project Statement .....	2
1.3.    Objectives .....	3
1.4.    Project Scope .....	4
1.5.    Advantages of Proposed Solution.....	5
1.6.    Chapter Summary .....	6
<b>CHAPTER 2.....</b>	<b>7</b>
2.1.    Existing System.....	8
2.2.    Drawbacks of Existing Systems.....	9
2.3.    Proposed System.....	12
2.4.    Benefits of the Proposed System .....	14
2.5.    Need to replace Existing System.....	15
2.6.    Chapter Summary .....	18
<b>CHAPTER 3.....</b>	<b>19</b>
3.1.    Introduction.....	20
3.2.    Understanding the system.....	20
3.3.    Advantages of the Proposed System.....	22

3.4.	Hurdles to Optimize the Current System .....	24
3.5.	Chapter Summary .....	26
<b>CHAPTER 4.....</b>		<b>28</b>
4.1.	Software Process Model.....	29
4.2.	Benefits of Selected Model.....	32
4.3.	Limitations of Selected Model.....	34
4.4.	Design .....	36
4.4.1.	Data Flow Diagram .....	39
4.4.2.	Entity Relationship Diagram.....	41
4.4.3.	Class Diagram .....	42
4.4.4.	UML Diagram.....	44
4.4.5.	Sequence Diagram.....	45
4.5.	Chapter Summary .....	46
<b>CHAPTER 5.....</b>		<b>48</b>
5.1.	Introduction to Database .....	49
5.2.	Why SQL? .....	51
5.3.	Structure .....	53
5.4.	Functional vs. Non-Functional Requirements.....	53
5.5.	Chapter Summary .....	56
<b>CHAPTER 6.....</b>		<b>58</b>
6.1.	Introduction.....	59
6.2.	Setting Up the Development Environment .....	60
6.3.	Implementation of Key Features.....	64
<b>CHAPTER 7.....</b>		<b>73</b>
7.1.	Introduction.....	74
7.2.	Testing Strategies.....	74

7.2.1.	Unit Testing.....	74
7.2.2.	Integration Testing .....	74
7.2.3.	System Testing.....	74
7.3.	Acceptance Testing.....	75
7.4.	Test Cases .....	75
7.4.1.	User Management Test Cases .....	75
7.4.2.	Class and Subject Management Test Cases .....	76
7.4.3.	Attendance Management Test Cases.....	76
7.5.	Testing Tools .....	77
7.6.	Testing Results.....	77
7.7.	Chapter Summary .....	78
<b>CHAPTER 8.....</b>		<b>79</b>
8.1.	Introduction.....	80
8.2.	Achievement of Objectives.....	80
8.3.	Improved Administrative Efficiency.....	80
8.4.	Enhanced Academic Management.....	80
8.5.	Effective Communication .....	80
8.6.	Benefits to Users .....	80
8.6.1.	Administrators.....	81
8.6.2.	Teachers .....	81
8.7.	Performance Evaluation.....	81
8.8.	Limitations and Areas for Improvement.....	82
8.9.	Chapter Summary .....	82
<b>REFERENCES .....</b>		<b>84</b>

## LIST OF FIGURES

<b>Figure 4.1:</b> Software Development Models .....	32
<b>Figure 4.2:</b> Data Flow Diagram .....	40
<b>Figure 4.3:</b> ER Diagram.....	42
<b>Figure 4.4:</b> Class Diagram .....	43
<b>Figure 4.5:</b> USECASE Diagram .....	44
<b>Figure 4.6:</b> Activity Diagram.....	45
<b>Figure 4.7:</b> Sequence Diagram.....	46
<b>Figure 5.1:</b> Database Structure.....	53
<b>Figure 6.1:</b> Login Page.....	64
<b>Figure 6.2:</b> Admins .....	65
<b>Figure 6.3:</b> Create New Admin.....	65
<b>Figure 6.4:</b> Teachers .....	66
<b>Figure 6.5:</b> Add New Teacher.....	66
<b>Figure 6.6:</b> Classes.....	67
<b>Figure 6.7:</b> Add New Classes.....	67
<b>Figure 6.8:</b> Subjects .....	68
<b>Figure 6.9:</b> Add New Subjects.....	68
<b>Figure 6.10:</b> Assigned Teacher To Classes With Subjects .....	69
<b>Figure 6.11:</b> Assign New Teacher To classes.....	69
<b>Figure 6.12:</b> Attendance Marks Sheet.....	70
<b>Figure 6.13:</b> Attendance Reports .....	70
<b>Figure 6.14:</b> Assignment Uploading Form (Teacher).....	71
<b>Figure 6.15:</b> Quiz Uploading Form (Teacher) .....	71
<b>Figure 6.16:</b> Mid Exam Answer Sheet Uploading (Teacher) .....	72
<b>Figure 6. 17:</b> Final Exam Answer Sheet Uploading (Teacher).....	72

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1. Introduction

The **GU Course Folder Management System** is a web-based application developed using the Laravel framework, aimed at streamlining the management of academic activities and resources within educational institutions. The system is designed to address the inefficiencies and challenges posed by traditional course management methods, which often rely heavily on manual processes and extensive paperwork. By providing a centralized digital platform, the system facilitates efficient management of courses, student attendance, assignments, quizzes, and other academic resources.

This project encompasses two main components: the Admin Panel and the Teacher Panel. The Admin Panel is intended for use by administrators to manage key academic entities such as administrators, teachers, students, classes, and subjects. The Teacher Panel is designed for teachers to manage their assigned classes and subjects, mark attendance, and upload academic materials. The overarching goal of the GU Course Folder Management System is to improve the efficiency, accuracy, and organization of academic management processes.

## 1.2. Project Statement

The traditional methods of managing academic activities and resources in educational institutions often involve significant manual effort, leading to inefficiencies, errors, and administrative burdens. The **GU Course Folder Management System** aims to transform these traditional practices by providing an innovative, digital solution that automates and streamlines course management tasks. The project focuses on creating a user-friendly platform that supports the following:

- Simplified management of administrators, teachers, students, classes, and subjects.
- Efficient assignment of classes and subjects to teachers.
- Easy access to academic resources and information for teachers.
- Accurate tracking of student attendance and academic performance.
- Secure storage and retrieval of academic materials, including assignments, quizzes, and past papers.

By leveraging the capabilities of the Laravel framework, the system ensures scalability, flexibility, and robustness, making it suitable for a wide range of academic institutions.

### **1.3. Objectives**

The **GU Course Folder Management System** is developed with several key objectives in mind:

#### **Streamline Administrative Tasks:**

Enable administrators to efficiently manage all aspects of course administration, including the addition, deletion, and editing of administrators, teachers, students, classes, and subjects. The system also allows for the assignment of classes and subjects to teachers, simplifying the overall management process.

#### **Facilitate Teacher Tasks:**

Provide teachers with the tools they need to manage their assigned classes and subjects effectively. This includes functionalities for marking daily attendance, uploading semester assignments, quizzes, past papers, and their corresponding answer sheets.

#### **Enhance Communication:**

Improve the communication between administrators and teachers through a centralized platform, ensuring that all relevant information is easily accessible and up-to-date. This fosters better coordination and collaboration, ultimately enhancing the educational experience for students.

#### **Reduce Paperwork:**

Minimize the reliance on physical documents by digitizing course management processes. This not only reduces the administrative burden but also contributes to a more sustainable environment by lowering paper consumption.

#### **Improve Accuracy:**

Ensure accurate tracking and management of academic activities and resources by leveraging the automation capabilities of the system. This reduces the likelihood of errors associated with manual tracking and enhances the overall reliability of the data.

## 1.4. Project Scope

The scope of the **GU Course Folder Management System** includes the development and implementation of two primary components, each with specific features and functionalities:

### Admin Panel:

The Admin Panel is designed to provide administrators with comprehensive control over the academic resources and activities within the institution. Key features of the Admin Panel include:

- **User Management:** Add, delete, and edit administrators, teachers, and students. This functionality ensures that the right personnel have access to the system and can perform their respective tasks.
- **Class and Subject Management:** Add, delete, and edit classes and subjects. Administrators can also assign classes and subjects to specific teachers, ensuring that each teacher has a well-defined set of responsibilities.
- **Administrative Control:** Manage various administrative tasks, including generating reports, monitoring system usage, and ensuring that the academic structure is organized and efficiently maintained.

### Teacher Panel:

The Teacher Panel focuses on facilitating the daily tasks and responsibilities of teachers. Key features of the Teacher Panel include:

- **Class and Subject Viewing:** View assigned classes and subjects. This functionality ensures that teachers have easy access to information about their academic responsibilities, including the subjects they need to teach and the classes they are responsible for.
- **Attendance Management:** Mark daily attendance for each class. This feature simplifies the attendance tracking process, ensuring accurate and up-to-date records of student attendance.
- **Resource Upload:** Upload semester assignments, quizzes, past papers, and their answer sheets. This functionality provides students with easy access to academic resources, supporting their learning and preparation for exams.

## 1.5. Advantages of Proposed Solution

The **GU Course Folder Management System** offers several significant advantages over traditional methods of course management:

### Efficiency

By automating and streamlining administrative and teaching tasks, the system significantly reduces the time and effort required for managing academic activities. This allows administrators and teachers to focus more on their core responsibilities, such as improving educational quality and student engagement.

### Accuracy

The digital management of academic resources and activities minimizes errors associated with manual tracking and paperwork. This enhances the reliability and accuracy of the data, ensuring that all academic records are up-to-date and error-free.

### Accessibility

Being a web-based platform, the system allows administrators and teachers to access the necessary information and functionalities from anywhere, providing flexibility and convenience. This is particularly beneficial in situations where remote access is required, such as during off-campus activities or in cases of distance learning.

### Organization

The centralized platform ensures that all academic resources and activities are well-organized and easily accessible, leading to better overall management. This improves the efficiency of administrative processes and ensures that all relevant information is readily available when needed.

### Communication

The system enhances communication between administrators and teachers, facilitating better coordination and collaboration. This helps to ensure that all parties are informed about important updates and changes, reducing the likelihood of misunderstandings and errors.

## Sustainability

By reducing the reliance on physical documents, the system contributes to a more sustainable and eco-friendly approach to course management. This not only lowers paper consumption but also reduces the environmental impact of the institution's administrative activities.

### 1.6. Chapter Summary

In this chapter, we introduced the **GU Course Folder Management System**, outlining its purpose and significance in enhancing the management of academic activities within an institution. We discussed the project statement, highlighting the challenges it aims to address, and outlined the primary objectives and scope of the project. Furthermore, we explored the advantages of the proposed solution, emphasizing its efficiency, accuracy, and organizational benefits. This foundation sets the stage for a detailed exploration of the system's design, implementation, and testing in the subsequent chapters.

## **CHAPTER 2**

## **EXISTING SYSTEM**

## 2.1. Existing System

Traditional course management systems in educational institutions often rely on a combination of manual processes and basic digital tools to handle various administrative and academic tasks. These systems typically involve paper-based records, spreadsheets, and basic database applications to manage course-related activities. While some institutions have adopted partial digital solutions, many continue to use outdated methods that present several challenges.

### Manual Processes

In many institutions, course management involves extensive manual procedures, including:

- **Paper-Based Records:** Administrative tasks such as maintaining student records, tracking attendance, and managing assignments are often handled using physical documents. This method is prone to errors, loss of documents, and inefficiencies in retrieving and updating information.
- **Manual Attendance Tracking:** Attendance is usually recorded on paper or in spreadsheets, leading to difficulties in tracking and managing student attendance over time. This approach can result in inaccurate records and increased administrative workload.
- **Assignment and Resource Management:** Assignments, quizzes, and other academic resources are often distributed and collected manually. Teachers may use physical copies or unorganized digital files, which complicates the management and retrieval of academic materials.

### Digital Solutions and Limitations

Some institutions have adopted digital solutions to address specific aspects of course management, such as:

- **Spreadsheet Applications:** Tools like Microsoft Excel or Google Sheets are used to track student information, attendance, and academic performance. While these tools offer basic functionalities, they lack integration and can become unwieldy as data volumes grow.

- **Basic Database Systems:** Some institutions use simple database systems to store and manage course information. However, these systems often require manual data entry and lack advanced features for automating and streamlining processes.
- **Learning Management Systems (LMS):** Existing LMS platforms provide functionalities for managing course content, assignments, and communication. However, many of these systems are not fully integrated with institutional administrative functions, leading to fragmented workflows and additional administrative overhead.

### Challenges of Existing Systems

The existing systems present several challenges that hinder the efficiency and effectiveness of course management:

- **Inefficiency:** Manual processes and fragmented digital tools result in time-consuming and error-prone administrative tasks. Managing large volumes of data and documents becomes increasingly complex and inefficient.
- **Accuracy Issues:** Paper-based records and manual data entry are prone to errors, leading to inaccuracies in student records, attendance tracking, and academic performance.
- **Limited Integration:** Many existing systems lack integration between different administrative functions, such as student management, attendance tracking, and resource management. This fragmentation leads to duplicated efforts and a lack of cohesive data.
- **Accessibility and Flexibility:** Traditional methods and basic digital tools often restrict access to information and resources, particularly for remote or off-campus users. This limits the flexibility and responsiveness of administrative and teaching tasks.

## 2.2. Drawbacks of Existing Systems

Existing systems for managing academic activities and resources in educational institutions present several significant drawbacks that impact efficiency, accuracy, and overall effectiveness. These drawbacks highlight the need for a more advanced and integrated solution, such as the **GU Course Folder Management System**. Here's a detailed look at the limitations of traditional and partially digital systems:

## 1. Inefficiency

- **Manual Data Entry:** Many traditional systems rely on manual data entry for maintaining student records, attendance, and academic materials. This process is time-consuming and labor-intensive, leading to increased administrative workload and slower response times.
- **Paper-Based Records:** Institutions that use paper-based records face challenges in organizing, storing, and retrieving documents. This method is not only inefficient but also prone to physical damage, loss, or misplacement of important documents.
- **Fragmented Tools:** The use of disparate tools, such as spreadsheets and basic databases, for different administrative tasks creates inefficiencies. Users often need to switch between multiple applications, leading to duplication of efforts and difficulties in maintaining consistent and accurate data.

## 2. Accuracy Issues

- **Human Error:** Manual processes and data entry are highly susceptible to human error. Mistakes in recording attendance, grading assignments, or updating student records can lead to inaccuracies that affect the integrity of academic information.
- **Data Duplication:** When using multiple systems or tools, there is a higher risk of data duplication and inconsistencies. For example, student records may be updated in one system but not in others, leading to discrepancies and confusion.

## 3. Limited Integration

- **Disjointed Systems:** Many existing solutions do not integrate seamlessly with other administrative functions. For instance, learning management systems (LMS) may not be fully integrated with student information systems (SIS), resulting in a fragmented approach to managing academic activities.
- **Lack of Automation:** Existing systems often lack automation features for routine tasks, such as generating reports, tracking attendance, or notifying stakeholders. This manual approach increases the workload for administrators and teachers and delays critical processes.

#### 4. Accessibility and Flexibility

- **Restricted Access:** Paper-based records and basic digital tools often restrict access to information and resources. Access may be limited to specific locations or require physical presence, which can be inconvenient for remote or off-campus users.
- **Limited Remote Functionality:** Many existing systems do not provide adequate support for remote access or online collaboration. This limitation affects the flexibility of teaching and administrative tasks, especially in cases where remote or hybrid learning is required.

#### 5. Security and Compliance

- **Data Security:** Paper-based records and basic digital tools may lack robust security measures to protect sensitive academic information. This vulnerability increases the risk of unauthorized access, data breaches, and loss of confidential information.
- **Compliance Challenges:** Institutions must comply with various regulations and standards regarding data privacy and security. Existing systems may not always meet these compliance requirements, leading to potential legal and regulatory issues.

#### 6. Usability and User Experience

- **Complex Interfaces:** Many existing systems suffer from complex and non-intuitive interfaces that make it challenging for users to navigate and perform tasks efficiently. This can lead to a steep learning curve and increased user frustration.
- **Inadequate Support:** Existing solutions may lack adequate support and training resources for users, leading to difficulties in fully utilizing the system's features and capabilities.

#### 7. Scalability and Adaptability

- **Limited Scalability:** Traditional and basic digital tools may not scale effectively to accommodate growing volumes of data or increasing numbers of users. This limitation can hinder the ability of institutions to expand and adapt to changing needs.
- **Lack of Customization:** Existing systems may offer limited customization options, making it difficult for institutions to tailor the system to their specific requirements and workflows.

## 2.3. Proposed System

The **GU Course Folder Management System** is designed to address the limitations of existing course management systems by providing a comprehensive, digital solution for educational institutions. Developed using the Laravel framework, the system aims to streamline and enhance the management of academic activities through its robust features and functionalities. Below is a detailed description of the proposed system, including its architecture, features, and benefits.

### System Architecture

The GU Course Folder Management System is built on a modern web-based architecture that ensures scalability, reliability, and ease of use. The system consists of two primary panels: the Admin Panel and the Teacher Panel. Both panels are designed to provide tailored functionalities for administrators and teachers, respectively.

- **Frontend:** The user interface (UI) is designed to be intuitive and user-friendly, ensuring that users can easily navigate and perform tasks. The frontend is developed using HTML, CSS, and JavaScript, providing a responsive design that is accessible across various devices and screen sizes.
- **Backend:** The backend is built using the Laravel framework, which provides a robust and secure environment for managing data and application logic. Laravel's features, such as routing, middleware, and Eloquent ORM, are utilized to create a scalable and efficient backend system.
- **Database:** A relational database management system (RDBMS) is used to store and manage data. Laravel's Eloquent ORM facilitates seamless interaction with the database, ensuring data integrity and consistency.

### Key Features

#### 1. Admin Panel

The Admin Panel is designed to provide comprehensive control over the academic and administrative functions of the institution. Key features include:

- **User Management:**
  - **Add, Edit, Delete Users:** Administrators can manage records for other administrators, teachers, and students, ensuring that user information is up-to-date and accurate.
  - **Role Management:** Assign roles and permissions to different users, controlling their access to various functionalities within the system.
- **Class and Subject Management:**
  - **Manage Classes and Subjects:** Add, edit, and delete classes and subjects, allowing for dynamic updates to the academic curriculum.
  - **Assign Classes to Teachers:** Assign specific classes and subjects to teachers, ensuring that each teacher has a defined set of responsibilities.
- **Administrative Control:**
  - **Generate Reports:** Produce reports on various aspects of academic management, including attendance, academic performance, and resource utilization.
  - **Monitor System Usage:** Track user activity and system performance to ensure smooth operation and identify areas for improvement.

## 2. Teacher Panel

The Teacher Panel is designed to support teachers in managing their academic responsibilities. Key features include:

- **Class and Subject Management:**
  - **View Assigned Classes and Subjects:** Teachers can access information about the classes and subjects they are responsible for, including schedules and student lists.
- **Attendance Management:**
  - **Mark Daily Attendance:** Teachers can record and manage student attendance for each class, ensuring accurate tracking of student participation.

- **Resource Management:**

- **Upload Academic Materials:** Teachers can upload semester assignments, quizzes, past papers, and their answer sheets, providing students with easy access to essential academic resources.
- **Organize Resources:** Categorize and manage uploaded resources, making it easier for students to find and access relevant materials.

## 2.4. Benefits of the Proposed System

### 1. Efficiency and Automation

- **Streamlined Processes:** Automates routine administrative tasks, reducing the time and effort required for managing academic activities.
- **Centralized Management:** Provides a single platform for managing all aspects of course administration, eliminating the need for multiple tools and systems.

### 2. Accuracy and Reliability

- **Error Reduction:** Digital management minimizes the risk of errors associated with manual data entry and paperwork.
- **Real-Time Updates:** Ensures that information is updated in real-time, providing accurate and current data for decision-making.

### 3. Enhanced Accessibility

- **Web-Based Access:** Allows users to access the system from any location with an internet connection, providing flexibility for remote and off-campus users.
- **Responsive Design:** Ensures that the system is accessible across various devices, including desktops, tablets, and smartphones.

### 4. Improved Communication and Collaboration

- **Integrated Communication:** Facilitates better communication between administrators and teachers through a unified platform, ensuring that all relevant information is easily accessible.

- **Collaboration Tools:** Supports collaboration by providing shared access to academic resources and information.

## 5. Security and Compliance

- **Robust Security:** Implements security measures to protect sensitive academic data, including user authentication, data encryption, and secure access controls.
- **Compliance:** Ensures that the system meets regulatory and compliance requirements for data privacy and security.

## 6. Scalability and Flexibility

- **Scalable Architecture:** Designed to handle increasing volumes of data and users, making it suitable for institutions of varying sizes.
- **Customizable Features:** Provides customization options to tailor the system to the specific needs and workflows of different institutions.

### 2.5. Need to replace Existing System

The **GU Course Folder Management System** is designed to replace traditional and partially digital course management methods, addressing the limitations and inefficiencies of existing systems. This replacement aims to modernize and enhance the management of academic activities through a comprehensive, integrated digital platform. Below is a detailed explanation of how the proposed system improves upon and replaces the existing methods.

#### Challenges of Existing Systems

Existing course management methods often involve a combination of manual processes and basic digital tools, which present several significant challenges:

- **Manual Processes:** Reliance on paper-based records and manual data entry results in inefficiencies, increased administrative workload, and a higher risk of errors.
- **Fragmented Tools:** Use of disparate tools, such as spreadsheets and basic databases, leads to fragmented workflows and difficulties in maintaining consistent and accurate data.

- **Limited Integration:** Existing systems often lack integration between various administrative functions, leading to disjointed processes and additional administrative overhead.
- **Restricted Access:** Many systems restrict access to information and resources, particularly for remote or off-campus users, limiting flexibility and responsiveness.
- **Security and Compliance Issues:** Traditional methods may not provide robust security measures or meet compliance requirements for data privacy and protection.

## Proposed System Improvements

The **GU Course Folder Management System** is designed to address these challenges through several key improvements:

### 1. Automation and Efficiency

- **Eliminating Manual Data Entry:** The system automates routine administrative tasks, such as user management, attendance tracking, and resource uploads. This reduces the time and effort required for managing academic activities and minimizes the risk of errors associated with manual data entry.
- **Centralized Platform:** Provides a single, integrated platform for managing all aspects of course administration, including user roles, class assignments, and academic resources. This eliminates the need for multiple tools and streamlines workflows.

### 2. Integration and Consistency

- **Unified System:** Integrates various administrative functions into a cohesive system, ensuring that all data and processes are connected. This integration improves consistency and accuracy, as all information is updated in real-time and managed through a single interface.
- **Automated Reporting:** Generates automated reports on academic activities, performance, and resource usage, providing administrators with actionable insights and reducing the need for manual report generation.

### 3. Enhanced Accessibility and Flexibility

- **Web-Based Access:** Offers web-based access to the system from any location with an internet connection, allowing users to manage academic tasks and access resources remotely. This is particularly beneficial for remote or hybrid learning environments.
- **Responsive Design:** Features a responsive design that ensures accessibility across various devices, including desktops, tablets, and smartphones. This enhances user experience and allows for flexible access to the system.

### 4. Improved Security and Compliance

- **Robust Security Measures:** Implements advanced security features, such as user authentication, data encryption, and secure access controls, to protect sensitive academic information. This addresses the security vulnerabilities of traditional methods and ensures that data is safeguarded.
- **Compliance with Regulations:** Ensures that the system meets regulatory and compliance requirements for data privacy and protection, addressing potential legal and regulatory issues associated with existing systems.

### 5. Enhanced Communication and Collaboration

- **Integrated Communication:** Facilitates better communication between administrators and teachers through a unified platform, providing a centralized location for sharing information and updates. This reduces the likelihood of misunderstandings and improves coordination.
- **Collaboration Features:** Supports collaboration by providing shared access to academic resources and information, enhancing the ability of teachers and administrators to work together effectively.

### 6. Scalability and Customization

- **Scalable Architecture:** Designed to handle increasing volumes of data and users, making it suitable for institutions of varying sizes. The system can scale to accommodate growth and evolving needs.

- **Customizable Features:** Offers customization options to tailor the system to the specific requirements and workflows of different institutions. This ensures that the system can be adapted to meet the unique needs of each organization.

## 2.6. Chapter Summary

The **GU Course Folder Management System** aims to replace traditional and partially digital course management methods that rely heavily on manual processes, paper-based records, and fragmented tools like spreadsheets and basic databases. These existing systems are inefficient, error-prone, and lack integration, leading to disjointed workflows and increased administrative workload. They also face challenges in accessibility, security, and compliance. The proposed system, developed using the Laravel framework, automates and centralizes administrative tasks, ensuring real-time updates, robust security, and regulatory compliance. It enhances efficiency, accuracy, and accessibility with features such as web-based access, responsive design, and integrated communication tools. This transition to a modern, scalable, and customizable digital platform addresses the limitations of existing systems, supporting better management of academic activities and resources.

## **CHAPTER 3**

## **PROPOSED SYSTEM**

### 3.1. Introduction

The **GU Course Folder Management System** is a comprehensive digital solution designed to replace traditional and partially digital course management methods, addressing their inefficiencies, inaccuracies, and fragmentation. Developed using the Laravel framework, the system features two main panels: the Admin Panel and the Teacher Panel. The Admin Panel allows administrators to manage users, classes, subjects, and administrative tasks, such as generating reports and monitoring system usage, with capabilities for adding, editing, and deleting records. The Teacher Panel enables teachers to view their assigned classes and subjects, mark daily attendance, and upload academic materials like assignments, quizzes, and past papers. The system offers numerous benefits, including streamlined processes, reduced manual errors, real-time data updates, and centralized management of academic activities. It enhances accessibility with web-based access and a responsive design, improves communication and collaboration through integrated tools, and ensures robust security and compliance with regulatory standards. Additionally, its scalable and customizable architecture makes it suitable for institutions of varying sizes, allowing for adaptation to specific needs and workflows. Overall, the GU Course Folder Management System provides an efficient, reliable, and secure platform for modernizing and optimizing course management in educational institutions.

### 3.2. Understanding the system

The **GU Course Folder Management System** is a sophisticated digital platform designed to streamline and enhance the management of academic activities within educational institutions. Developed using the Laravel framework, this system is divided into two main components: the Admin Panel and the Teacher Panel. Each panel is tailored to the specific needs of its users, providing a comprehensive solution for managing courses, users, and academic resources.

#### Admin Panel

The Admin Panel is the control center for administrators, offering a range of features to manage the institution's academic and administrative operations efficiently:

- **User Management:** Administrators can add, edit, and delete records for other administrators, teachers, and students. This ensures that user information is always accurate and up-to-date. Additionally, role management features allow administrators to assign

specific permissions to users, controlling their access to different functionalities within the system.

- **Class and Subject Management:** This feature allows administrators to manage the creation, modification, and deletion of classes and subjects. Administrators can dynamically update the academic curriculum and assign specific classes and subjects to teachers, ensuring that each teacher's responsibilities are clearly defined.
- **Administrative Control:** The system provides tools for generating comprehensive reports on various aspects of academic management, such as attendance, academic performance, and resource utilization. It also includes features for monitoring system usage, tracking user activity, and ensuring smooth operation.

## Teacher Panel

The Teacher Panel is designed to assist teachers in their daily academic responsibilities, providing tools and features that facilitate efficient classroom management:

- **Class and Subject Management:** Teachers can easily view the classes and subjects assigned to them, along with relevant details such as schedules and student lists. This helps teachers stay organized and focused on their teaching duties.
- **Attendance Management:** Teachers can mark daily attendance for each class directly within the system, ensuring accurate and timely tracking of student participation. This feature simplifies the process of attendance management and reduces the likelihood of errors.
- **Resource Management:** Teachers can upload and organize various academic materials, including semester assignments, quizzes, past papers, and their answer sheets. These resources are made accessible to students, providing them with the materials they need to succeed in their studies.

## Benefits

The **GU Course Folder Management System** offers numerous advantages over traditional and partially digital course management methods:

- **Efficiency and Automation:** By automating routine administrative tasks and centralizing management, the system reduces the time and effort required for managing academic activities.
- **Accuracy and Reliability:** Digital management minimizes the risk of errors associated with manual data entry and paperwork, while real-time updates ensure that information is always current.
- **Enhanced Accessibility:** Web-based access and a responsive design allow users to access the system from any location with an internet connection, providing flexibility for remote and off-campus users.
- **Improved Communication and Collaboration:** Integrated communication tools facilitate better coordination between administrators and teachers, while collaboration features support the sharing and management of academic resources.
- **Security and Compliance:** The system implements robust security measures, including user authentication, data encryption, and secure access controls, to protect sensitive academic information and ensure compliance with regulatory standards.
- **Scalability and Customization:** The system's scalable architecture can handle increasing volumes of data and users, making it suitable for institutions of varying sizes. Customizable features allow the system to be tailored to the specific needs and workflows of different institutions.

### 3.3. Advantages of the Proposed System

The **GU Course Folder Management System** offers numerous advantages that address the shortcomings of traditional and partially digital course management methods. Developed using the Laravel framework, this modern digital platform enhances efficiency, accuracy, accessibility, and security in the management of academic activities. Here are the key advantages of the proposed system:

## 1. Efficiency and Automation

- **Streamlined Processes:** Automates routine administrative tasks such as user management, attendance tracking, and resource uploads. This reduces the time and effort required for managing academic activities.
- **Centralized Management:** Provides a single platform for managing all aspects of course administration, eliminating the need for multiple tools and systems and thereby streamlining workflows.

## 2. Accuracy and Reliability

- **Error Reduction:** Digital management minimizes the risk of errors associated with manual data entry and paperwork, ensuring more accurate and reliable information.
- **Real-Time Updates:** Ensures that information is updated in real-time, providing administrators and teachers with accurate and current data for decision-making.

## 3. Enhanced Accessibility

- **Web-Based Access:** Allows users to access the system from any location with an internet connection, providing flexibility for remote and off-campus users.
- **Responsive Design:** Ensures that the system is accessible across various devices, including desktops, tablets, and smartphones, enhancing the user experience and allowing for flexible access.

## 4. Improved Communication and Collaboration

- **Integrated Communication:** Facilitates better communication between administrators and teachers through a unified platform, ensuring that all relevant information is easily accessible.
- **Collaboration Tools:** Supports collaboration by providing shared access to academic resources and information, enhancing the ability of teachers and administrators to work together effectively.

## 5. Security and Compliance

- **Robust Security:** Implements advanced security measures to protect sensitive academic data, including user authentication, data encryption, and secure access controls. This addresses the security vulnerabilities of traditional methods and ensures that data is safeguarded.
- **Compliance with Regulations:** Ensures that the system meets regulatory and compliance requirements for data privacy and protection, addressing potential legal and regulatory issues associated with existing systems.

## 6. Scalability and Flexibility

- **Scalable Architecture:** Designed to handle increasing volumes of data and users, making it suitable for institutions of varying sizes. The system can scale to accommodate growth and evolving needs.
- **Customizable Features:** Provides customization options to tailor the system to the specific requirements and workflows of different institutions. This ensures that the system can be adapted to meet the unique needs of each organization.

## 7. Enhanced Resource Management

- **Organized Resource Uploads:** Allows teachers to upload and categorize academic materials such as assignments, quizzes, and past papers, making it easier for students to find and access relevant resources.
- **Efficient Resource Sharing:** Facilitates the sharing of academic resources among teachers and students, enhancing the overall educational experience.

### 3.4. Hurdles to Optimize the Current System

Optimizing the current course management system to transition to the **GU Course Folder Management System** involves addressing several hurdles. These challenges must be overcome to ensure a smooth and effective implementation of the new system. Here are the key hurdles:

## 1. Resistance to Change

- **User Adaptation:** Teachers, administrators, and students may be resistant to adopting a new system, especially if they are accustomed to traditional methods. Overcoming this resistance requires effective communication, training, and support to help users understand the benefits and functionality of the new system.
- **Cultural Shift:** Transitioning from a paper-based or partially digital system to a fully digital platform involves a cultural shift within the institution. This requires buy-in from all stakeholders and a commitment to embracing new technologies and workflows.

## 2. Technical Challenges

- **Legacy System Integration:** Existing data and processes may be entrenched in legacy systems that are incompatible with the new platform. Migrating data and integrating these systems can be complex and time-consuming.
- **Infrastructure Requirements:** The new system may require upgrades to the institution's IT infrastructure, including servers, networks, and security protocols. Ensuring that the infrastructure can support the new system is crucial for its successful implementation.

## 3. Data Migration and Management

- **Data Integrity:** Migrating data from old systems to the new platform must be done carefully to ensure data integrity and accuracy. Any errors during migration can lead to significant issues in the new system.
- **Data Standardization:** Existing data may be in various formats and structures, requiring standardization before it can be effectively used in the new system. This process can be labor-intensive and require significant attention to detail.

## 4. Training and Support

- **Comprehensive Training:** Users need thorough training to understand and utilize the new system effectively. This includes training sessions, user manuals, and ongoing support to address any issues that arise.

- **Technical Support:** Providing robust technical support during and after the transition is essential to address any technical issues and ensure that users can rely on the system without disruptions.

## 5. Cost and Resource Allocation

- **Budget Constraints:** Implementing a new system involves significant costs, including software development, infrastructure upgrades, training, and support. Securing the necessary budget and resources can be challenging.
- **Resource Management:** Allocating sufficient resources, including personnel and time, to manage the transition while maintaining ongoing operations is critical. This may involve hiring additional staff or reallocating existing resources.

## 6. Security and Compliance

1. **Data Security:** Ensuring that the new system meets all security requirements to protect sensitive academic data is paramount. This includes implementing robust security measures and regular audits to prevent data breaches.
2. **Regulatory Compliance:** The system must comply with all relevant regulations and standards related to data privacy and protection. This requires a thorough understanding of regulatory requirements and the ability to implement necessary measures.

## 7. User Experience and Acceptance

- **System Usability:** The new system must be user-friendly and intuitive to encourage adoption and regular use. If the system is difficult to navigate or use, users may revert to old methods.
- **Feedback Mechanisms:** Establishing channels for user feedback and continuously improving the system based on this feedback is essential to ensure that it meets the needs and expectations of all users.

### 3.5. Chapter Summary

In this chapter, we have comprehensively explored the **GU Course Folder Management System**, a modern digital platform designed to enhance the management of academic activities. The

**Detailed Description of the Proposed System** highlighted the system's architecture, featuring an Admin Panel for user, class, and subject management, and a Teacher Panel for managing attendance and academic resources. We delved into **Understanding the System**, emphasizing its user-friendly interface, centralized management, and the roles of administrators and teachers within the system. The **Advantages of the Proposed System** were outlined, showcasing its ability to streamline processes, improve accuracy, enhance accessibility, foster better communication and collaboration, and ensure robust security and compliance. Finally, we addressed the **Hurdles to Optimize the Current System**, identifying challenges such as resistance to change, technical integration, data migration, training needs, budget constraints, and security concerns. This chapter provides a comprehensive understanding of the proposed system, its benefits, and the challenges to be addressed for successful implementation.

# **CHAPTER 4**

## **SOFTWARE PROCESS MODEL**

## 4.1. Software Process Model

The development of the **GU Course Folder Management System** can be approached using various software process models. Selecting an appropriate model is crucial for managing the project efficiently and ensuring the successful delivery of the system. Here, we explore three widely-used software process models: The Waterfall Model, the Agile Model, and the Iterative Model, and discuss their applicability to the development of the GU Course Folder Management System.

### Waterfall Model:

The Waterfall Model is a linear and sequential approach to software development, where each phase must be completed before the next phase begins. This model is characterized by its structured and systematic progression through distinct phases.

- **Phases:**
  - **Requirements Gathering:** Collecting detailed requirements from stakeholders.
  - **System Design:** Designing the system architecture and components based on the gathered requirements.
  - **Implementation:** Coding and integrating the system components.
  - **Testing:** Verifying and validating the system to ensure it meets the specified requirements.
  - **Deployment:** Releasing the system to the end-users.
  - **Maintenance:** Ongoing support and updates to the system.
- **Advantages:**
  - Clear and well-defined phases.
  - Easy to manage due to its rigidity.
  - Well-suited for projects with well-understood requirements.

- **Disadvantages:**
  - Inflexible to changes in requirements.
  - Late discovery of issues, which can be costly to fix.
  - Not ideal for projects where requirements may evolve.
- **Applicability:** The Waterfall Model can be used for the GU Course Folder Management System if the project requirements are well-understood and unlikely to change. However, it may not be the best choice if flexibility is needed to accommodate evolving requirements.

### **Agile Model:**

The Agile Model is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer feedback. Agile methodologies, such as Scrum and Kanban, focus on delivering small, functional pieces of the system in short development cycles called sprints.

- **Phases:**
  - **Planning:** Defining the project vision, roadmap, and sprint goals.
  - **Development:** Coding and building small, incremental features in sprints.
  - **Testing:** Continuous testing of features during each sprint.
  - **Review:** Reviewing and demonstrating the completed features to stakeholders.
  - **Retrospective:** Reflecting on the sprint to identify improvements for the next cycle.
- **Advantages:**
  - High flexibility and adaptability to changes.
  - Continuous delivery of functional software.
  - Strong collaboration between developers and stakeholders.
  - Early detection and resolution of issues.

- **Disadvantages:**
  - Requires active involvement from stakeholders.
  - Can be challenging to predict timelines and budgets.
  - May lead to scope creep if not managed properly.
- **Applicability:** The Agile Model is well-suited for the GU Course Folder Management System, especially if the project requirements are expected to evolve or if stakeholder feedback is critical. Agile's iterative nature allows for continuous improvement and adaptation.

### **Iterative Model:**

The Iterative Model focuses on developing the system through repeated cycles (iterations) of planning, development, and refinement. Each iteration builds upon the previous one, gradually enhancing the system's functionality and quality.

- **Phases:**
  - **Initial Planning:** Defining the project's scope and initial requirements.
  - **Iteration Planning:** Planning the objectives and deliverables for each iteration.
  - **Development:** Coding and implementing the planned features for the iteration.
  - **Testing:** Verifying and validating the implemented features.
  - **Review and Feedback:** Assessing the iteration's outcomes and incorporating feedback for the next cycle.
- **Advantages:**
  - Allows for incremental improvement and refinement.
  - Early identification and resolution of issues.
  - Flexibility to accommodate changes in requirements.

- **Disadvantages:**
  - Requires careful planning and management of iterations.
  - Can be resource-intensive due to repeated cycles of development and testing.
  - Potential for iteration scope creep if not controlled.
- **Applicability:** The Iterative Model is suitable for the GU Course Folder Management System, particularly if the project involves complex requirements that may need refinement over time. Its incremental approach ensures continuous improvement and stakeholder involvement.

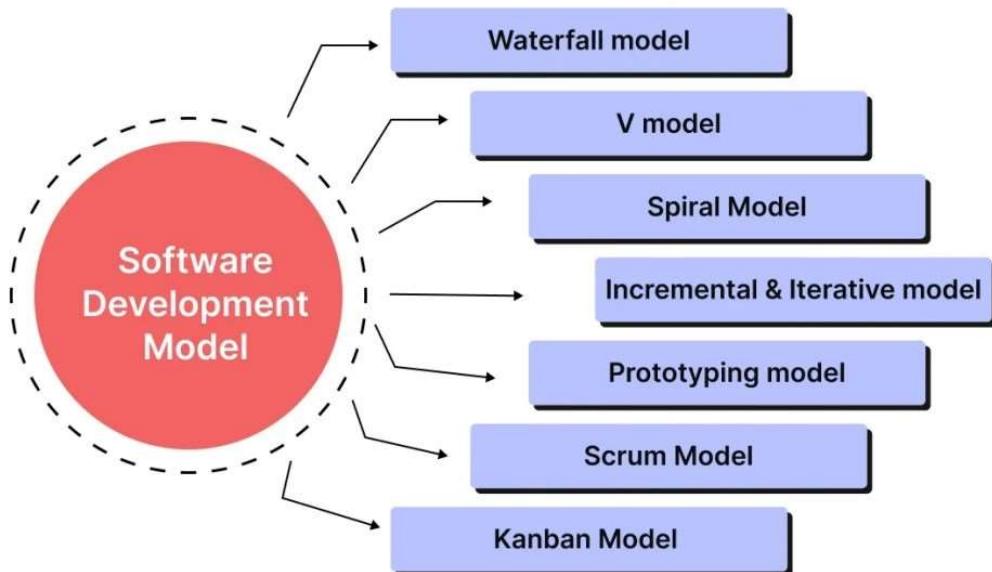


Figure 4.1: Software Development Models

## 4.2. Benefits of Selected Model

After considering various software process models, the **Agile Model** is selected as the best approach for developing the **GU Course Folder Management System**. The Agile Model's iterative and flexible nature makes it well-suited for handling evolving requirements and fostering collaboration between developers and stakeholders.

## Benefits of the Agile Model

### 1. Flexibility and Adaptability:

- **Responsive to Change:** Agile allows for changes in requirements even late in the development process. This is crucial for the GU Course Folder Management System, where feedback from teachers and administrators can lead to refinements and improvements.
- **Iterative Development:** The project is broken down into small, manageable sprints, allowing the team to focus on delivering functional increments of the system regularly.

### 2. Continuous Improvement:

- **Regular Feedback Loops:** Each sprint concludes with a review and retrospective, ensuring that stakeholder feedback is continuously incorporated into the development process. This leads to a product that better meets user needs.
- **Incremental Enhancements:** Features are developed, tested, and improved incrementally, leading to a more robust and refined system over time.

### 3. Enhanced Collaboration:

- **Cross-Functional Teams:** Agile promotes collaboration among developers, testers, and stakeholders, ensuring that all perspectives are considered and integrated into the project.
- **Stakeholder Involvement:** Continuous engagement with stakeholders ensures that the project aligns with their expectations and requirements, leading to higher satisfaction and acceptance.

### 4. Early and Predictable Delivery:

- **Regular Releases:** The Agile Model enables regular delivery of functional software, allowing stakeholders to see progress and provide feedback early and often. This reduces the risk of large-scale project failures and ensures that issues are identified and resolved promptly.

- **Improved Project Visibility:** Frequent updates and demonstrations provide stakeholders with clear visibility into the project's progress, fostering trust and transparency.

## 5. Risk Mitigation:

- **Early Issue Detection:** Continuous testing and integration help identify and address issues early in the development cycle, reducing the likelihood of major problems later on.
- **Adaptation to Change:** The Agile Model's inherent flexibility allows the project team to adapt to changes in requirements or priorities without significant disruption.

## 6. Quality Assurance:

- **Continuous Testing:** Testing is integrated into every sprint, ensuring that quality is maintained throughout the development process. This leads to a more reliable and robust system.
- **User-Centric Development:** Frequent user feedback ensures that the system is developed with the end-user in mind, resulting in a more intuitive and user-friendly application.

## 7. Efficient Resource Utilization:

- **Focused Sprints:** Each sprint focuses on specific features or functionalities, ensuring that resources are efficiently allocated and managed.
- **Adaptable Planning:** Agile allows for dynamic planning, enabling the project team to re-prioritize tasks based on current needs and available resources.

### 4.3. Limitations of Selected Model

While the Agile Model offers numerous benefits for the development of the **GU Course Folder Management System**, it also comes with certain limitations. These limitations need to be considered to ensure that potential challenges are effectively managed throughout the project.

## 1. Dependency on Active Stakeholder Involvement

- **Continuous Collaboration Required:** Agile relies heavily on regular feedback and active participation from stakeholders. If stakeholders are not consistently available or engaged, it can hinder the progress and effectiveness of the development process.
- **Time Commitment:** Stakeholders must dedicate significant time to attend meetings, reviews, and provide feedback, which may be challenging in a busy academic environment.

## 2. Scope Creep

- **Frequent Changes:** The flexibility to accommodate changing requirements can lead to scope creep if not carefully managed. This can result in extended timelines and increased costs if additional features and modifications are continually added.
- **Prioritization Challenges:** Determining which features and changes to prioritize can be difficult, especially when there are conflicting demands from different stakeholders.

## 3. Difficulty in Predicting Timelines and Costs

- **Uncertain Estimates:** The iterative nature of Agile makes it challenging to predict the overall project timeline and cost accurately. Initial estimates can change significantly as the project evolves and new requirements emerge.
- **Budget Management:** Managing the project budget can be complicated due to the ongoing changes and adjustments, requiring careful financial oversight.

## 4. Requires Experienced Team Members

- **Agile Expertise:** Successful implementation of Agile requires team members who are experienced in Agile methodologies. Inadequate understanding or experience can lead to improper implementation and reduced effectiveness.
- **Training Needs:** Team members and stakeholders may require training in Agile practices, which can be time-consuming and require additional resources.

## 5. Documentation Challenges

- **Less Formal Documentation:** Agile emphasizes working software over comprehensive documentation. This can lead to insufficient documentation, which may be problematic for future maintenance, onboarding new team members, or meeting regulatory requirements.
- **Balancing Documentation:** Finding the right balance between coding and documenting can be challenging, as overemphasis on minimal documentation might overlook the importance of detailed records.

## 6. Risk of Miscommunication

- **Frequent Meetings:** Agile involves frequent meetings and updates, which can sometimes lead to information overload or miscommunication among team members and stakeholders.
- **Consistency in Communication:** Ensuring that all team members are consistently on the same page can be difficult, especially in larger teams or when dealing with remote stakeholders.

## 7. Project Complexity

- **Managing Complexity:** Agile can become challenging to manage for very complex projects with many interdependent components. Coordinating across multiple sprints and ensuring integration can be difficult.
- **Integration Issues:** Regular integration and continuous delivery require robust testing and integration practices, which can be challenging to maintain.

## 4.4. Design

The design phase of the **GU Course Folder Management System** focuses on creating a detailed architecture and blueprint that outlines how the system's components will interact to achieve the desired functionalities. The design process includes system architecture, user interface design, database design, and security considerations.

## System Architecture

The system architecture of the GU Course Folder Management System is based on the MVC (Model-View-Controller) pattern, which is a common design pattern for web applications, especially those built with the Laravel framework.

- **Model:** Manages the data and business logic of the application. It defines the database schema and handles data manipulation.
- **View:** Represents the user interface of the application. It includes HTML, CSS, and JavaScript components to create a responsive and user-friendly interface.
- **Controller:** Handles the user requests and interactions. It processes input from the user, interacts with the model, and returns the appropriate view.

## User Interface Design

The user interface (UI) design focuses on creating an intuitive and user-friendly experience for both administrators and teachers. Key elements of the UI design include:

- **Admin Dashboard:**
  - **User Management:** Interfaces for adding, editing, and deleting administrators, teachers, and students.
  - **Class and Subject Management:** Interfaces for managing classes and subjects, and assigning them to teachers.
  - **Analytics and Reports:** Dashboards for viewing attendance statistics, assignment submissions, and other relevant data.
- **Teacher Dashboard:**
  - **Class and Subject Overview:** A summary of the classes and subjects assigned to the teacher.
  - **Attendance Management:** Tools for marking daily attendance and viewing attendance history.

- **Resource Upload:** Interfaces for uploading semester assignments, quizzes, past papers, and answer sheets.

## Database Design

The database design involves defining the structure of the database that will store the application's data. This includes creating tables, establishing relationships between tables, and defining constraints to ensure data integrity.

- **Tables:**
  - **Users:** Stores information about administrators, teachers, and students.
  - **Classes:** Stores information about different classes.
  - **Subjects:** Stores information about various subjects.
  - **Assignments:** Stores details of assignments, quizzes, past papers, and their answer sheets.
  - **Attendance:** Records daily attendance for each class.
- **Relationships:**
  - **One-to-Many:** One user can be assigned to many classes and subjects.
  - **Many-to-Many:** Teachers can be associated with multiple classes and subjects, and vice versa.

## Security Considerations

Security is a critical aspect of the system design to protect sensitive academic data and ensure only authorized users have access to specific functionalities.

- **Authentication:** Implementing robust user authentication mechanisms to ensure that only authorized users can access the system.
- **Authorization:** Defining user roles and permissions to restrict access to certain functionalities based on the user's role (e.g., admin, teacher).

- **Data Encryption:** Encrypting sensitive data, both in transit and at rest, to protect against unauthorized access.
- **Regular Audits:** Implementing logging and monitoring to detect and respond to any suspicious activities or breaches.

## Design Patterns and Best Practices

Adopting design patterns and best practices ensures that the system is scalable, maintainable, and efficient.

- **MVC Pattern:** Ensuring a clear separation of concerns, making the system easier to maintain and extend.
- **RESTful APIs:** Designing APIs that follow REST principles for easy integration with other systems and services.
- **Responsive Design:** Ensuring that the user interface is responsive and works well on various devices, including desktops, tablets, and smartphones.
- **Modular Design:** Breaking down the system into modular components to improve reusability and maintainability.

### 4.4.1. Data Flow Diagram

The Data Flow Diagram (DFD) illustrates how data moves through the GU Course Folder Management System, detailing the inputs, processes, and outputs at various levels.

#### Level 0: Context Diagram

- **Admin Panel:** Manages users, classes, and subjects.
- **Teacher Panel:** Manages attendance and academic resources.
- **Database:** Central repository for all data.

## Level 1: Detailed Diagram

### 1. Admin Panel:

- **Manage Users:** Admins can add, edit, and delete users (administrators, teachers, students).
- **Manage Classes and Subjects:** Admins can create and assign classes and subjects to teachers.

### 2. Teacher Panel:

- **Attendance Management:** Teachers can mark daily attendance.
- **Resource Management:** Teachers can upload assignments, quizzes, past papers, and answer sheets.

### 3. Database:

- Stores all user, class, subject, attendance, and resource data.

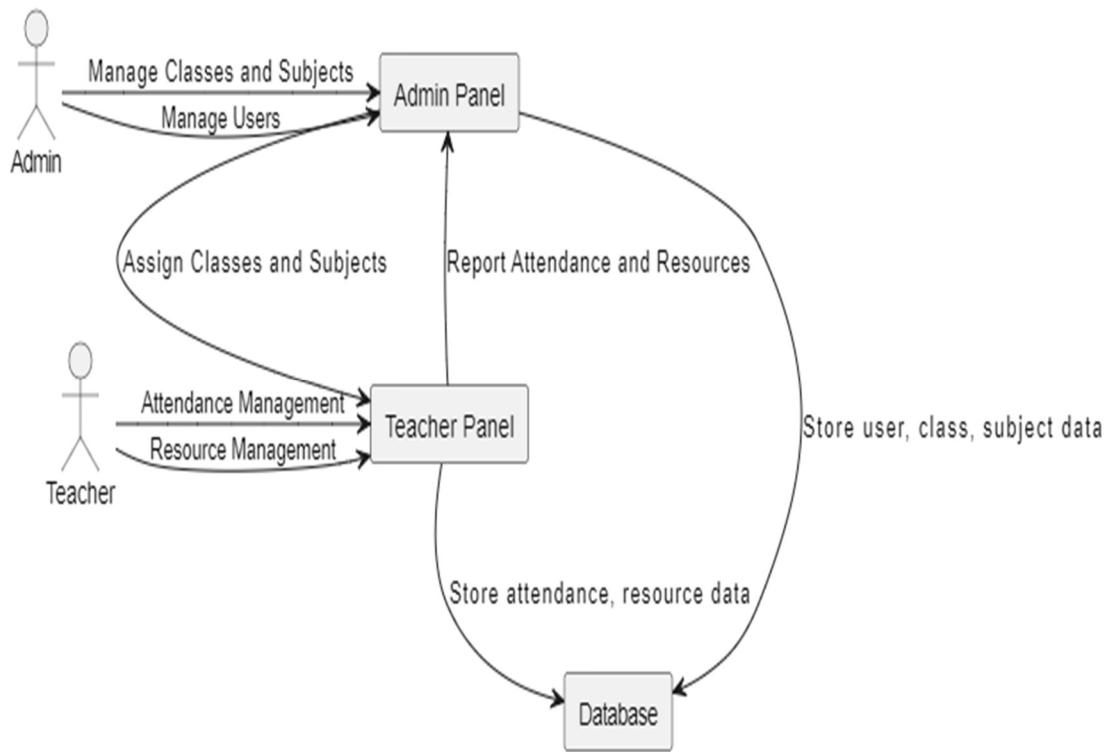


Figure 4.2: Data Flow Diagram

#### 4.4.2. Entity Relationship Diagram

The **Entity Relationship Diagram (ERD)** shows the relationships between different entities in the GU Course Folder Management System.

##### Entities and Relationships:

- **User:** (UserID, Name, Role, Email, Password)
  - Relationships:
    - One-to-Many with **Class**: A user (teacher/admin) can manage multiple classes.
    - Many-to-Many with **Subject**: Teachers can be associated with multiple subjects, and subjects can be taught by multiple teachers.
- **Class:** (ClassID, ClassName, Description)
  - Relationships:
    - One-to-Many with **User** (admin/teacher).
    - One-to-Many with **Attendance**.
- **Subject:** (SubjectID, SubjectName, Description)
  - Relationships:
    - Many-to-Many with **User** (teacher).
    - One-to-Many with **Assignment**.
- **Attendance:** (AttendanceID, Date, Status, ClassID, StudentID)
  - Relationships:
    - Many-to-One with **Class**.
    - Many-to-One with **User** (student).
- **Assignment:** (AssignmentID, Title, Description, FilePath, SubjectID, TeacherID)
  - Relationships:
    - Many-to-One with **Subject**.
    - Many-to-One with **User** (teacher).

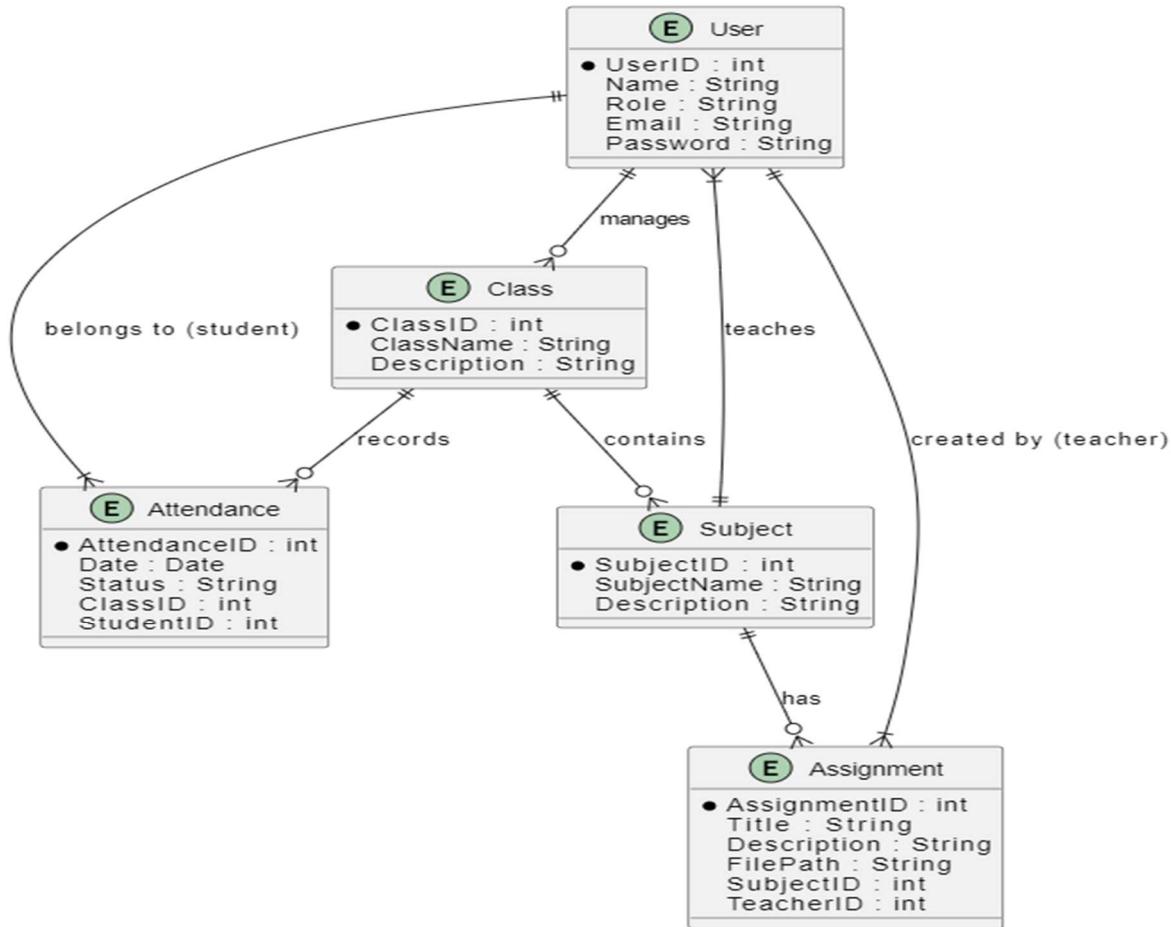


Figure 4.3: ER Diagram

#### 4.4.3. Class Diagram

The **Class Diagram** represents the structure of the system by showing the system's classes, their attributes, and the relationships between the classes.

#### Classes and Attributes:

- **User:**
  - Attributes: UserID, Name, Role, Email, Password
  - Methods: addUser(), editUser(), deleteUser(), assignClass(), assignSubject()
- **Class:**
  - Attributes: ClassID, ClassName, Description
  - Methods: addClass(), editClass(), deleteClass(), assignTeacher()
  -

- **Subject:**
  - Attributes: SubjectID, SubjectName, Description
  - Methods: addSubject(), editSubject(), deleteSubject(), assignTeacher()
- **Attendance:**
  - Attributes: AttendanceID, Date, Status, ClassID, StudentID
  - Methods: markAttendance(), viewAttendance()
- **Assignment:**
  - Attributes: AssignmentID, Title, Description, FilePath, SubjectID, TeacherID
  -

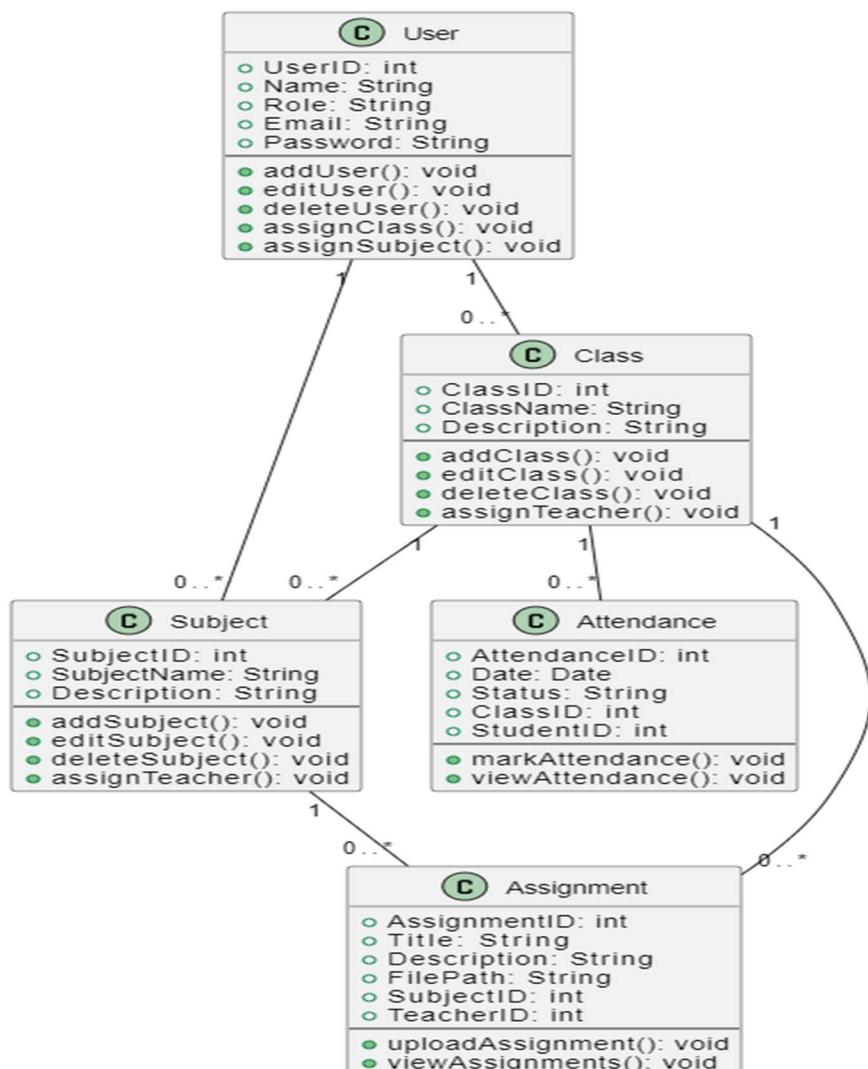


Figure 4.4: Class Diagram

- Methods: uploadAssignment(), viewAssignments()

#### 4.4.4. UML Diagram

The **Unified Modeling Language (UML) Diagram** combines various diagrams

- **Use Case Diagram:**
  - **Actors:** Admin, Teacher, Student
  - **Use Cases:**
    - **Admin:** Manage Users, Manage Classes, Manage Subjects
    - **Teacher:** Mark Attendance, Upload Resources (Assignments, Quizzes, Past Papers)

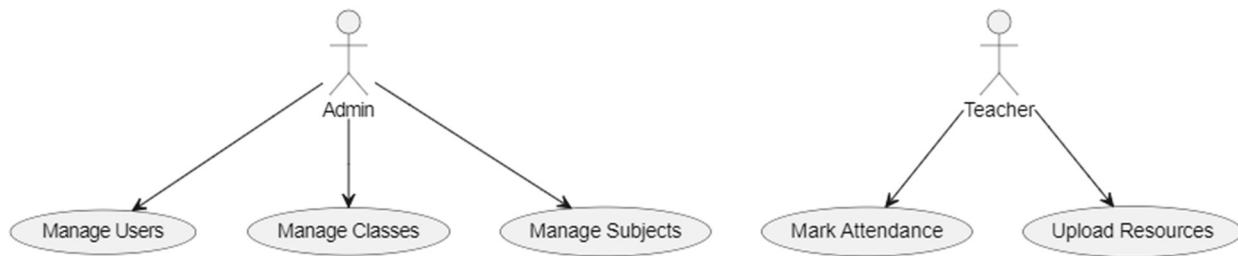


Figure 4.5: USECASE Diagram

#### ACTIVITY DIAGRAM

- **Activity Diagram:**
  - **Processes:**
    - Admin: Login -> Manage Users -> Manage Classes -> Manage Subjects
    - Teacher: Login -> View Classes -> Mark Attendance -> Upload Resources

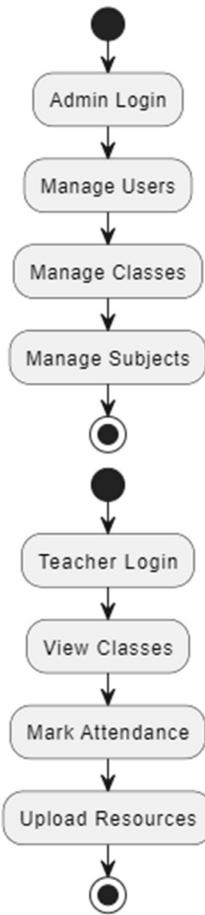


Figure 4.6: Activity Diagram

#### 4.4.5. Sequence Diagram

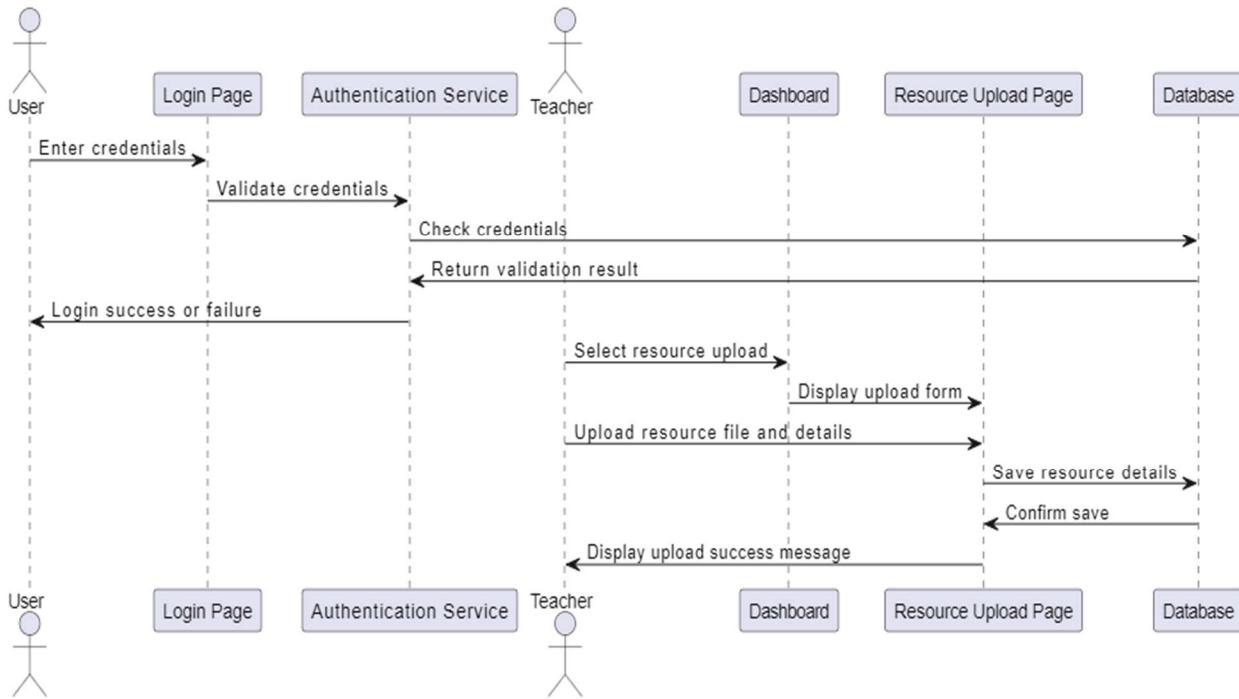
The **Sequence Diagram** shows the interactions between objects in a particular sequence of time. Here, we focus on two primary sequences: user login and resource upload.

##### User Login Sequence:

1. **User -> Login Page:** Enter credentials
2. **Login Page -> Authentication Service:** Validate credentials
3. **Authentication Service -> Database:** Check credentials
4. **Database -> Authentication Service:** Return validation result
5. **Authentication Service -> User:** Login success or failure

### Resource Upload Sequence (Teacher):

1. Teacher -> Dashboard: Select resource upload
2. Dashboard -> Resource Upload Page: Display upload form
3. Teacher -> Resource Upload Page: Upload resource file and details
4. Resource Upload Page -> Database: Save resource details
5. Database -> Resource Upload Page: Confirm save
6. Resource Upload Page -> Teacher: Display upload success message



**Figure 4.7: Sequence Diagram**

## 4.5. Chapter Summary

This chapter has provided a comprehensive overview of the software process models considered for the development of the **GU Course Folder Management System**, highlighting the benefits and limitations of the selected Agile Model, followed by a detailed design of the system. Various software process models, including Waterfall, V-Model, and Agile, were analyzed for their suitability. The Agile Model was chosen due to its flexibility, iterative development, and ability to incorporate continuous feedback, making it ideal for the dynamic requirements of the GU Course

Folder Management System. The Agile Model offers numerous advantages, including flexibility and adaptability to change, continuous improvement through regular feedback loops, enhanced collaboration among cross-functional teams, early and predictable delivery of functional software, effective risk mitigation, and a strong focus on quality assurance and efficient resource utilization. These benefits ensure that the system can evolve with user needs and deliver a high-quality product. Despite its advantages, the Agile Model also has limitations, such as dependency on active stakeholder involvement, risk of scope creep, challenges in predicting timelines and costs, the need for experienced team members, potential documentation challenges, risk of miscommunication, and difficulties in managing complex projects. These limitations require careful management to ensure the success of the project.

# **CHAPTER 5**

## **INTRODUCTION TO DATABASE**

## 5.1. Introduction to Database

The **GU Course Folder Management System** employs a structured database designed to efficiently store, manage, and retrieve data related to the administration and academic activities within an educational institution. Utilizing SQL within the XAMPP server environment, the database plays a crucial role in ensuring the integrity, security, and accessibility of the system's data.

### Overview of Database Management System

A **Database Management System (DBMS)** is essential for handling the large volumes of data generated by the GU Course Folder Management System. The DBMS used in this project is MySQL, a robust and widely-used relational database management system (RDBMS) that offers excellent performance and reliability. The choice of MySQL within the XAMPP server setup ensures seamless integration with the Laravel framework, facilitating efficient data operations and management.

### Key Components of the Database

#### 1. User Management:

- **Admins, Teachers, and Students:** The database maintains comprehensive records for each user type, storing essential information such as user ID, name, role, email, and password. This facilitates user authentication, role-based access control, and personalized interactions within the system.

#### 2. Class and Subject Management:

- **Classes:** The database holds details of various classes, including class ID, name, and description. This allows for organized management and assignment of classes to students and teachers.
- **Subjects:** Similar to classes, subjects are stored with relevant details, enabling their assignment to classes and teachers.

### 3. Attendance Tracking:

- **Daily Attendance:** The database records daily attendance for each class, capturing the date, attendance status, and associated class and student IDs. This ensures accurate attendance tracking and reporting.

### 4. Resource Management:

- **Assignments, Quizzes, and Past Papers:** The database stores information about academic resources uploaded by teachers, including titles, descriptions, file paths, and associated subject and teacher IDs. This facilitates easy access and management of educational materials.

## Database Structure

The database is structured using tables that represent different entities in the system. Each table is designed with fields that capture specific attributes of the entity, and relationships between tables are established to ensure data integrity and coherence. Key tables in the database include:

- **Users:** Stores user information (UserID, Name, Role, Email, Password).
- **Classes:** Contains class details (ClassID, ClassName, Description).
- **Subjects:** Holds subject details (SubjectID, SubjectName, Description).
- **Attendance:** Records attendance data (AttendanceID, Date, Status, ClassID, StudentID).
- **Assignments:** Stores assignment details (AssignmentID, Title, Description, FilePath, SubjectID, TeacherID).

## Benefits of the Database Approach

1. **Data Integrity:** The relational structure ensures that data is consistently accurate and reliable through the use of constraints and relationships.
2. **Efficiency:** SQL queries enable efficient data retrieval and manipulation, supporting the system's performance needs.
3. **Scalability:** The database design can accommodate the growth of the institution, handling increasing volumes of data without compromising performance.
4. **Security:** Role-based access control and data encryption ensure that sensitive information is protected from unauthorized access.

## 5.2. Why SQL?

The **GU Course Folder Management System** relies on SQL (Structured Query Language) for its database management due to several compelling reasons. SQL is a powerful and versatile language that provides the necessary tools and functionalities to handle the complex data requirements of the system efficiently.

### Relational Database Management

- **Structured Data:** SQL is designed to manage structured data within a relational database. The GU Course Folder Management System involves various interrelated entities such as users, classes, subjects, attendance records, and assignments, making a relational model ideal for this purpose.
- **Data Integrity:** SQL ensures data integrity through the use of constraints (e.g., primary keys, foreign keys), which enforce relationships between tables and maintain the consistency and accuracy of the data.

### Efficient Data Retrieval and Manipulation

- **Querying Capability:** SQL's powerful querying capabilities allow for efficient retrieval and manipulation of data. Complex queries can be executed to extract specific information, generate reports, and support decision-making processes.
- **Performance:** SQL databases, like MySQL, are optimized for performance, handling large volumes of data with high efficiency. This is crucial for the GU Course Folder Management System, which must manage and retrieve data quickly and reliably.

### Standardization and Compatibility

- **Industry Standard:** SQL is an industry-standard language used by numerous database management systems. Its widespread adoption ensures compatibility and interoperability across different platforms and applications.

- **Cross-Platform Support:** The use of SQL within the MySQL database, which is supported by the XAMPP server, ensures that the system can be deployed and accessed across various operating systems, enhancing its versatility and usability.

## Data Security

- **Access Control:** SQL provides robust mechanisms for access control, allowing administrators to define user roles and permissions. This ensures that sensitive data is accessible only to authorized users, enhancing the security of the system.
- **Data Protection:** Features such as encryption and secure connections (e.g., SSL/TLS) can be implemented within SQL databases to protect data from unauthorized access and breaches.

## Scalability and Flexibility

- **Scalability:** SQL databases are highly scalable, capable of growing with the system's needs. As the GU Course Folder Management System expands, the database can handle increasing amounts of data and user interactions without compromising performance.
- **Flexibility:** SQL allows for modifications and updates to the database schema without significant disruptions. This flexibility is essential for accommodating evolving requirements and enhancements to the system.

## Robust Support and Community

- **Documentation and Resources:** SQL and MySQL have extensive documentation, tutorials, and resources available, making it easier for developers to find solutions and best practices.
- **Active Community:** A large and active community of SQL and MySQL users contributes to continuous improvements, support, and troubleshooting, ensuring the technology remains up-to-date and reliable.

## Integration with Laravel and XAMPP

- **Seamless Integration:** SQL integrates seamlessly with the Laravel framework, which is used for developing the GU Course Folder Management System. Laravel's Eloquent ORM (Object-Relational Mapping) simplifies database interactions using SQL.
- **XAMPP Compatibility:** The XAMPP server package includes MySQL, providing a convenient and reliable environment for development and deployment. This integration streamlines the setup and management of the database.

### 5.3. Structure

Table	Action	Rows	Type	Collation	Size	Overhead
assignments	Browse  Structure  Search  Insert  Empty  Drop	15	InnoDB	utf8mb4_general_ci	16.0 KiB	-
assign_subjects	Browse  Structure  Search  Insert  Empty  Drop	7	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
class	Browse  Structure  Search  Insert  Empty  Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
class_teacher	Browse  Structure  Search  Insert  Empty  Drop	13	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
finalterms	Browse  Structure  Search  Insert  Empty  Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
midterms	Browse  Structure  Search  Insert  Empty  Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse  Structure  Search  Insert  Empty  Drop	20	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
papers	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
quizzes	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
student_attendance	Browse  Structure  Search  Insert  Empty  Drop	57	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
subjects	Browse  Structure  Search  Insert  Empty  Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
teacher_subjects	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse  Structure  Search  Insert  Empty  Drop	28	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
Console	les	Sum	162	InnoDB	utf8mb4 general ci	320.0
						0 E

Figure 5.1: Database Structure

### 5.4. Functional vs. Non-Functional Requirements

The **GU Course Folder Management System** is designed to streamline academic management processes within an educational institution. To achieve this, it is essential to clearly define both functional and non-functional requirements. Functional requirements specify what the system should do, while non-functional requirements describe how the system performs its functions.

## Functional Requirements

Functional requirements detail the specific behaviors and functionalities that the system must exhibit. For the GU Course Folder Management System, these include:

### 1. User Management:

- **Admin:**
  - Add, edit, and delete administrators, teachers, and students.
  - Assign roles to users.
- **Teacher:**
  - View assigned classes and subjects.
  - Mark Attendance
  - Upload Quizzes and Assignments
  - Upload Subject Past papers

### 2. Class and Subject Management:

- Create, edit, and delete classes and subjects.
- Assign subjects to specific classes.
- Allocate classes and subjects to teachers.

### 3. Attendance Management:

- Teachers can mark daily attendance for each class.
- View and edit attendance records.

### 4. Resource Management:

- Teachers can upload semester assignments, quizzes, past papers, and their answer sheets.
- Organize and categorize uploaded resources.

**5. Reporting and Analytics:**

- Generate reports on attendance, assignment submissions, and academic performance.
- Provide statistical insights for administrative decision-making.

**6. Authentication and Authorization:**

- Secure login and logout for all users.
- Role-based access control to ensure that only authorized users can access specific functionalities.

**Non-Functional Requirements**

Non-functional requirements define the system's quality attributes, focusing on how the system operates rather than specific behaviors. These include:

**1. Performance:**

- The system should handle multiple concurrent users without significant performance degradation.
- Data retrieval and submission operations should be completed within a few seconds to ensure a smooth user experience.

**2. Usability:**

- The user interface should be intuitive and easy to navigate for all users, including administrators, teachers, and students.
- Provide clear instructions and feedback for user actions.

**3. Reliability:**

- The system should have a high uptime, ensuring it is available when needed.
- Implement robust error handling to manage unexpected situations gracefully.

#### 4. Scalability:

- The system should be able to scale to accommodate an increasing number of users and data without compromising performance.
- Support for adding new features and modules as the institution's needs grow.

#### 5. Security:

- Protect sensitive data through encryption, both in transit and at rest.
- Implement strong authentication mechanisms to prevent unauthorized access.
- Regular security audits and updates to address vulnerabilities.

#### 6. Maintainability:

- The system's codebase should be modular and well-documented to facilitate easy maintenance and updates.
- Use version control to manage changes and track the history of modifications.

#### 7. Compatibility:

- Ensure compatibility with various web browsers and devices, including desktops, tablets, and smartphones.
- Seamless integration with other institutional systems if needed.

#### 8. Backup and Recovery:

- Regular backups of the database to prevent data loss.
- Implement a recovery plan to restore data and system functionality in case of failure

### 5.5. Chapter Summary

This chapter provided an in-depth look into the database aspect of the **GU Course Folder Management System**, highlighting the importance of using SQL within the XAMPP server for its robust data management capabilities. The rationale behind choosing SQL was discussed, emphasizing its strengths in handling structured data, ensuring data integrity, offering efficient querying and manipulation, and providing security and scalability. Additionally, the chapter

detailed the system's functional and non-functional requirements, outlining the specific functionalities the system must perform, such as user management, attendance tracking, and resource management, as well as the quality attributes it must exhibit, including performance, usability, reliability, and security. Together, these elements form a comprehensive foundation for developing a robust, efficient, and user-friendly academic management system.

# **CHAPTER 6**

## **DEVELOPMENT**

## 6.1. Introduction

The development phase is the heart of the software engineering process, where the system is constructed based on the designs and requirements defined in the earlier phases. This chapter provides a detailed overview of the development process for the GU Course Folder Management System, covering the tools and technologies used, the implementation of key features, and the testing strategies employed to ensure a reliable and functional system.

## Tools and Technologies

The development of the GU Course Folder Management System leveraged a variety of tools and technologies to ensure a robust and scalable solution.

## Programming Languages and Frameworks

- **PHP:** The core programming language used for server-side scripting.
- **Laravel:** A PHP framework used for developing the system, providing a clean and elegant syntax, and facilitating rapid application development.
- **HTML/CSS:** Used for designing the user interface.
- **JavaScript:** Used for enhancing user interactions and functionality on the client side.

## Database

- **MySQL:** The relational database management system used to store and manage the application's data.

## Development Environment

- **XAMPP:** A local server environment used to develop and test the application.
- **phpMyAdmin:** A web-based interface used to manage the MySQL database.

## 6.2. Setting Up the Development Environment

Setting up the development environment for the **GU Course Folder Management System** involves several key steps to ensure a smooth and efficient development process. The environment needs to be configured correctly to support the Laravel framework, MySQL database, and XAMPP server, which are essential components of the system.

### Step 1: Install XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting of Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

- **Download and Install:**

- Download the XAMPP installer from the official website (<https://www.apachefriends.org/>).
- Run the installer and follow the on-screen instructions to complete the installation.
- Choose the components to install (ensure Apache, MySQL, and PHP are selected).
- Once installed, launch the XAMPP Control Panel and start the Apache and MySQL services.

### Step 2: Install Composer

Composer is a dependency management tool for PHP that allows you to manage your project's libraries and packages.

- **Download and Install:**

- Download the Composer installer from the official website (<https://getcomposer.org/>).
- Run the installer and follow the on-screen instructions.
- After installation, open a command prompt and run `composer -V` to verify the installation.

### Step 3: Install Laravel

Laravel is a PHP framework for web artisans that provides an expressive and elegant syntax. It simplifies common tasks such as routing, sessions, caching, and authentication.

- **Install Laravel via Composer:**

- Open a command prompt and navigate to the htdocs directory in your XAMPP installation (e.g., cd C:\xampp\htdocs).
- Run the command: composer create-project --prefer-dist laravel/laravel gu\_course\_folder\_management\_system.
- This will create a new Laravel project in a directory named gu\_course\_folder\_management\_system.

### Step 4: Configure the .env File

The .env file is used to configure environment-specific settings for your Laravel application.

- **Set Database Configuration:**

- Open the .env file in the root of your Laravel project.
- Update the database settings to match your MySQL configuration:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=gu_course_folder_management_system
DB_USERNAME=root
DB_PASSWORD=
```

- Save the .env file.

## Step 5: Create the MySQL Database

- **Using phpMyAdmin:**
  - Open phpMyAdmin by navigating to <http://localhost/phpmyadmin> in your web browser.
  - Click on the "New" button in the left-hand sidebar to create a new database.
  - Name the database `gu_course_folder_management_system` and click "Create".

## Step 6: Migrate Database and Seed Data

Laravel uses migrations to manage database schema changes and seeders to populate the database with initial data.

- **Run Migrations:**
  - Open a command prompt and navigate to your Laravel project directory.
  - Run the command: `php artisan migrate`.
  - This will create the necessary tables in your database.
- **Seed the Database (optional):**
  - If you have seed data, run the command: `php artisan db:seed`.

## Step 7: Set Up a Virtual Host (Optional)

Setting up a virtual host can make it easier to access your project.

- **Configure Apache:**
  - Open the Apache configuration file (`httpd-vhosts.conf`) located in `C:\xampp\apache\conf\extra\`.
  - Add the following configuration:

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/gu_course_folder_management_system/public"
    ServerName gucourse.local
    <Directory "C:/xampp/htdocs/gu_course_folder_management_system">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

- Save the file and restart the Apache server.
- **Update Hosts File:**
  - Open the hosts file located at `C:\Windows\System32\drivers\etc\hosts` in a text editor with administrative privileges.
  - Add the following line:

```
127.0.0.1 gucourse.local
```
  - Save the file.

## Step 8: Run the Application

- **Access the Application:**
  - Open your web browser and navigate to `http://localhost/gu_course_folder_management_system/public` or `http://gucourse.local` if you set up a virtual host.
  - You should see the Laravel welcome page, indicating that the environment is set up correctly.

By following these steps, you will have a fully configured development environment for the **GU Course Folder Management System**, enabling you to proceed with development, testing, and deployment of the application.

### 6.3. Implementation of Key Features

The GU Course Folder Management System is designed to streamline the management of academic processes within an educational institution. The key features include user management, class and subject management, attendance tracking, resource management, and reporting. Here is a detailed overview of the implementation of these key features:

#### 1. User Management

##### Admin Panel:

##### Login:

Here the user; admin and teacher can login using their credentials:

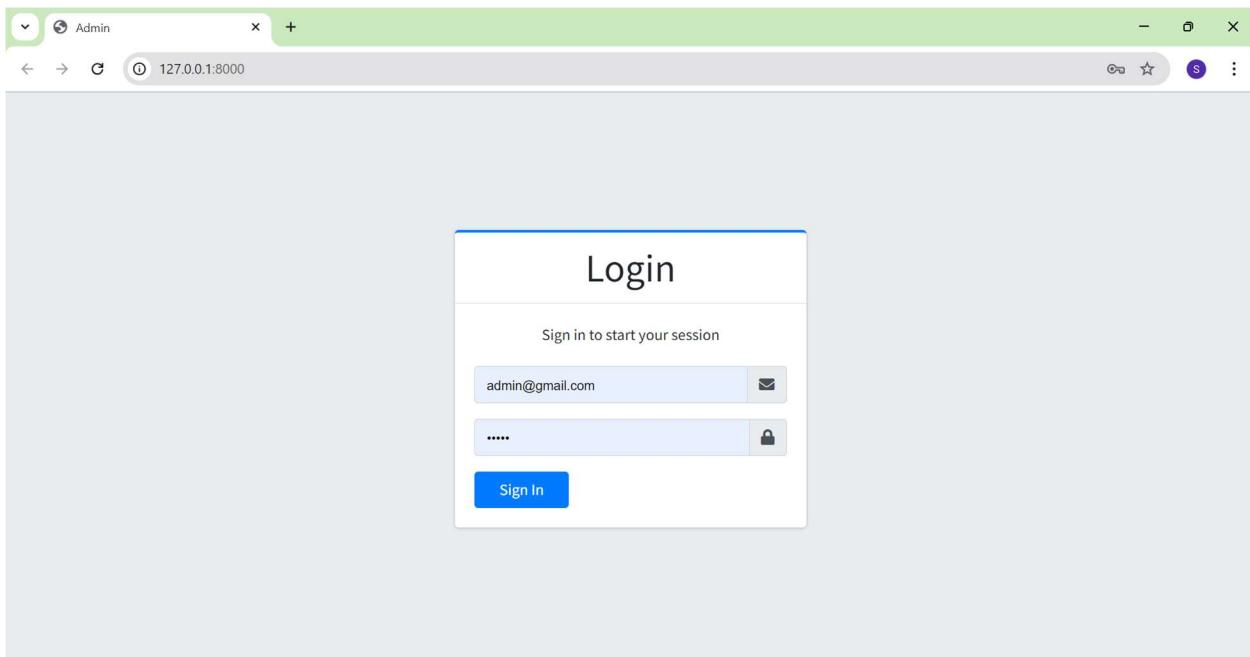


Figure 6.1: Login Page

##### Add, Edit, and Delete Users:

**Implementation:** Using Laravel's built-in authentication system, user roles (admin, teacher, student) are defined. Admins can add new users through a form that collects necessary information such as name, email, role, and password. The UserController handles the logic for creating, updating, and deleting users.

**View Admins:**

The screenshot shows a web application interface for managing course folders. On the left is a dark sidebar with a user icon, the text 'GU Course Folder', and a search bar. Below the search bar are links for 'Dashboard', 'Admin' (which is highlighted in blue), 'Academics', 'Students', 'Teachers', 'Attendance', 'Change Password', and 'Logout'. The main content area has a header 'Search Admin' with fields for 'Name' and 'Email', and buttons for 'Search' and 'Reset'. To the right of the search area is a message 'Total Admins: 1'. Below this is a table titled 'Admins List' with columns 'ID', 'Name', 'Email', and 'Actions'. A single row is shown for an admin named 'Admin' with email 'admin@gmail.com', with 'Edit' and 'Delete' buttons in the Actions column. At the bottom of the page, there is a copyright notice 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and a version note 'Version 3.2.0'.

**Figure 6.2: Admins****Add New Admin:**

The screenshot shows a form titled 'Admin New Admin'. It contains three input fields: 'Name\*' with value 'admin2', 'Email\*' with value 'admin2@gmail.com', and 'Password\*' with value '\*\*\*\*\*'. Below the password field is a large empty text area. At the bottom of the form is a blue button labeled 'Add Admin'.

**Figure 6.3: Create New Admin**

**View Teachers:**

The screenshot shows a user interface for managing teachers. On the left is a dark sidebar with a user profile picture for 'Admin'. Below it are navigation links: 'Dashboard', 'Admin', 'Academics', 'Students', 'Teachers' (which is highlighted in blue), 'Attendance', 'Change Password', and 'Logout'. The main content area has a header 'Search Teacher' with input fields for 'Name' and 'Email', and buttons for 'Search' and 'Reset'. Below this is a 'Teacher List' table with columns: ID, Name, Email, Status, and Actions (with 'Edit' and 'Delete' buttons). Two teacher entries are listed: one with ID 13, Name Azam, Email azam@gmail.com, and another with ID 2, Name Teacher, Email teacher@gmail.com, both marked as 'Active'. At the bottom of the page are copyright and version information: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

**Figure 6 4: Teachers****Add new Teacher:**

The screenshot shows a 'Add New Teacher' form. The sidebar on the left is identical to Figure 6.4. The main form has fields for 'Name\*' (containing 'New Teacher'), 'Email\*' (containing 'teacher2@gmail.com'), 'Status' (set to 'Active'), 'Password\*' (containing '.....'), and a 'Add Teacher' button. The bottom of the page includes the standard copyright and version information.

**Figure 6.5: Add New Teacher**

## 2. Class and Subject Management

- **Create, Edit, and Delete Classes and Subjects:**

- **Implementation:** Admins can manage classes and subjects through forms and views. The ClassController and SubjectController manage these operations.

### Classes

ID	Name	Status	Creator	Actions
4	BSIT 4th Morning	Active	Admin	<button>Edit</button> <button>Delete</button>
3	BSIT 6Th Evening	Active	Admin	<button>Edit</button> <button>Delete</button>
2	BSIT 8th Morning	Active	Admin	<button>Edit</button> <button>Delete</button>
1	BSIT 5th Evening	Active	Admin	<button>Edit</button> <button>Delete</button>

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Figure 6.6: Classes

Add New Class

Name\*

Active

Add Class

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Figure 6.7: Add New Classes

## Subjects:

The screenshot shows a web-based course management system. On the left is a dark sidebar with user profile information (Admin), a search bar, and a navigation menu. The menu items include Dashboard, Admin, Academics (which is currently selected), Class, Subject (selected), Assign Class Teacher, Students, Teachers, and Attendance. The main content area has a header "Search Subject" with fields for "Subject Name" and "Subject Type\*" (with a dropdown menu showing "Select Type"), and buttons for "Search" and "Reset". Below this is a table titled "Subjects List" with columns: ID, Name, Type, Status, Creator, Created Date, and Actions. The table contains three rows of data:

ID	Name	Type	Status	Creator	Created Date	Actions
6	JAVA	Practical	Active	Admin	04-06-2024 11:13 AM	<button>Edit</button> <button>Delete</button>
5	Web	Practical	Active	Admin	16-02-2024 17:12 PM	<button>Edit</button> <button>Delete</button>
4	C++	Practical	Active	Admin	16-02-2024 17:08 PM	<button>Edit</button> <button>Delete</button>

At the bottom of the page, there is a copyright notice "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and a version number "Version 3.2.0".

Figure 6.8: Subjects

This screenshot shows the "Admin New Subject" form. The sidebar is identical to Figure 6.8. The main form has a title "Admin New Subject" and contains the following fields:

- Subject Name\***: A text input field containing the value "Project".
- Subject Type\***: A dropdown menu showing "Practical".
- Status\***: A dropdown menu showing "Active".

At the bottom of the form is a blue "Add Subject" button. The footer of the page includes the copyright notice "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and the version number "Version 3.2.0".

Figure 6.9: Add New Subjects

- **Assign Subjects to Classes and Teachers:**

- **Implementation:** The relationship between classes, subjects, and teachers is managed through pivot tables. Laravel's Eloquent ORM makes it easy to define and interact with these relationships.

The screenshot shows a web-based course management system interface. On the left is a dark sidebar with the title 'GU Course Folder' and a user profile picture for 'Admin'. Below the profile are navigation links: 'Dashboard', 'Admin' (selected), 'Academics' (selected), 'Class', 'Subject', 'Assign Class Teacher' (selected), 'Students', 'Teachers', and 'Attendance'. A search bar with a magnifying glass icon is also present. The main content area has a header 'Assign Class Teacher List' with a blue 'Assign Class Teacher' button. Below the header is a table titled 'Class Teacher List' with columns: ID, Class Name, Teacher Name, Subject Name, Status, Creator, Created Date, and Actions (Edit and Delete buttons). There are four rows of data in the table. At the bottom of the page, a copyright notice reads 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and a version number 'Version 3.2.0'.

ID	Class Name	Teacher Name	Subject Name	Status	Creator	Created Date	Actions
7	BSIT 6Th Evening	Teacher	Network And Security	Active	Admin	27-02-2024 12:13 PM	<button>Edit</button> <button>Delete</button>
6	BSIT 8th Morning	Azam	C++	Active	Admin	27-02-2024 12:12 PM	<button>Edit</button> <button>Delete</button>
3	BSIT 4th Morning	Azam	Datamining	Active	Admin	27-02-2024 12:09 PM	<button>Edit</button> <button>Delete</button>
2	BSIT 6Th Evening	Azam	Network And Security	Active	Admin	27-02-2024 12:09 PM	<button>Edit</button> <button>Delete</button>

**Figure 6.10: Assigned Teacher To Classes With Subjects**

The screenshot shows a 'Add new Class Teacher' form. The sidebar on the left is identical to Figure 6.10. The main form has a header 'Add new Class Teacher'. It contains four input fields with dropdown menus: 'Class Name\*' (BSIT 4th Morning), 'Teacher Name\*' (Teacher), 'Subject Name\*' (JAVA), and 'Select Status' (Active). At the bottom is a blue 'Add Class Teacher' button.

**Figure 6.11: Assign New Teacher To classes**

### 3. Attendance Tracking

- **Mark Daily Attendance:**
  - **Implementation:** Teachers can mark attendance for their classes through an intuitive interface. Attendance data is stored in the attendances table, linked to the relevant class and student records also a teacher can upload leave file.

The screenshot shows a 'Mark Attendance' form. On the left is a sidebar with navigation links: GU Course Folder, Teacher, Search, Dashboard, Class & Subjects, My Students, Attendance (selected), Assignments, Quizzes, Past Papers, Change Password, and Logout. The main area has 'Select Date:' (22-Aug-2024), 'Class:' (BSIT 6Th Evening), and 'Subject:' (Network And Security) dropdowns. A 'Filter' button is below them. A table lists students Javed and Latif with columns: Student Name, Status (Present dropdown), and Leave File (Choose File button). Both rows show 'No file chosen'.

**Figure 6.12:** Attendance Marks Sheet

Attendance Report is shown as:

The screenshot shows an 'Attendance Reports' page. The sidebar is identical to Figure 6.12. The main area has 'Select Date:' (18-Aug-2024), 'Class:' (BSIT 6Th Evening), and 'Subject:' (Network And Security) dropdowns, along with 'Filter' and 'Export' buttons. A table lists student attendance with columns: Student, Class Name, Subject Name, Date, Status, and Leave File. Javed is present on 2024-08-18, and Latif is absent on the same date. Copyright information and Version 3.2.0 are at the bottom.

**Figure 6.13:** Attendance Reports

## 4. Resource Management

- **Upload Assignments, Quizzes, Past Papers:**

- **Implementation:** Teachers can upload various academic resources. The files are stored in the public storage, and metadata is stored in the database.

The screenshot shows a web-based course management system. On the left is a dark sidebar menu with the following items:

- GU Course Folder
- User profile picture of Azam
- Search bar
- Dashboard
- Class & Subjects
- My Students
- Attendance
- Assignments (highlighted in blue)
- Quizzes
- Past Papers
- Change Password
- Logout

The main content area has a blue header "Upload Assignments". It contains several form fields:

- Class:** Select class (dropdown menu)
- Subject:** Select Subject (dropdown menu)
- No of Assignment\***: Select Assignment No (dropdown menu)
- Best Assignment\***: Choose File (button) - No file chosen
- Average Assignment\***: Choose File (button) - No file chosen
- Worst Assignment\***: Choose File (button) - No file chosen

The URL in the browser address bar is 127.0.0.1:8000/teacher/assignment/assignment/addTeacher.

Figure 6.14: Assignment Uploading Form (Teacher)

The screenshot shows the same course management system interface as Figure 6.14, but the sidebar menu item "Quizzes" is highlighted in blue. The main content area has a blue header "Upload Quiz". The form fields are identical to Figure 6.14:

- Class:** Select class (dropdown menu)
- Subject:** Select Subject (dropdown menu)
- No of Assignment\***: Select Quiz No (dropdown menu)
- Best Quiz\***: Choose File (button) - No file chosen
- Average Quiz\***: Choose File (button) - No file chosen
- Worst Quiz\***: Choose File (button) - No file chosen

Figure 6.15: Quiz Uploading Form (Teacher)

The screenshot shows a web-based course management system interface. On the left is a dark sidebar with user information ('GU Course Folder' and profile picture 'Azam') and a search bar. Below these are navigation links: Dashboard, Class & Subjects, My Students, Attendance, Assignments, Quizzes, Past Papers, and Change Password, with 'Past Papers' having a dropdown arrow. At the bottom of the sidebar are Logout and Copyright information ('Copyright © 2014-2021 AdminLTE.io. All rights reserved.'). The main content area has a blue header 'Mid Term'. It contains a 'Paper' section with a table header ('Class', 'Subject', 'Paper Date', 'Paper', 'Best Ans', 'Average Ans', 'Worst Ans', 'Description', 'Action') and a message 'No data available in table'. A search bar and an 'Upload' button are also present.

Figure 6.16: Mid Exam Answer Sheet Uploading (Teacher)

This screenshot shows the same course management system interface, but the header is 'Final Term'. The main content area is identical to Figure 6.16, featuring a 'Paper' section with a table header ('Class', 'Subject', 'Paper Date', 'Paper', 'Best Ans', 'Average Ans', 'Worst Ans', 'Description', 'Action') and a message 'No data available in table'. A search bar and an 'Upload' button are also present.

Figure 6.17: Final Exam Answer Sheet Uploading (Teacher)

# **CHAPTER 7**

## **TESTING**

## 7.1. Introduction

Testing is a critical phase in the software development lifecycle, aimed at identifying and resolving defects and ensuring that the system meets its specified requirements. This chapter provides an overview of the testing strategies employed for the GU Course Folder Management System, including the different types of testing conducted, the tools used, and the results obtained.

## 7.2. Testing Strategies

The testing phase for the GU Course Folder Management System was divided into several key strategies to ensure comprehensive coverage and identification of issues.

### 7.2.1. Unit Testing

Unit testing focuses on verifying the functionality of individual components or units of the system. Each unit is tested in isolation to ensure it performs as expected.

- **Tools Used:** PHPUnit (a testing framework for PHP)
- **Scope:** Testing individual functions and methods within models, controllers, and other components.

### 7.2.2. Integration Testing

Integration testing verifies that different components of the system work together as expected. This type of testing focuses on the interactions between units and ensures that they integrate correctly.

- **Tools Used:** PHPUnit, Laravel's built-in testing tools
- **Scope:** Testing interactions between models, controllers, views, and the database.

### 7.2.3. System Testing

System testing evaluates the complete and integrated system to ensure it meets the specified requirements. This type of testing is conducted in an environment that closely mirrors the production environment.

- **Tools Used:** Manual testing, Selenium (for automated browser testing)
- **Scope:** Testing the entire application, including all features and functionalities.

### 7.3. Acceptance Testing

Acceptance testing is conducted to determine if the system meets the acceptance criteria and is ready for deployment. This testing is often performed by the end-users or stakeholders.

- **Tools Used:** Manual testing, user feedback
- **Scope:** Testing the system against the requirements and user stories defined during the requirement gathering phase.

### 7.4. Test Cases

This section outlines some of the key test cases used to validate the functionality of the GU Course Folder Management System.

#### 7.4.1. User Management Test Cases

- **Test Case 1:** Create a new user
  - **Description:** Verify that a new user can be created successfully.
  - **Steps:**
    1. Navigate to the "Add User" page.
    2. Enter valid user details.
    3. Submit the form.
  - **Expected Result:** The user is created, and a success message is displayed.
- **Test Case 2:** Edit an existing user
  - **Description:** Verify that an existing user's details can be updated.
  - **Steps:**
    1. Navigate to the "Edit User" page for a specific user.
    2. Update the user details.
    3. Submit the form.

- **Expected Result:** The user's details are updated, and a success message is displayed.

#### 7.4.2. Class and Subject Management Test Cases

- **Test Case 1:** Create a new class
  - **Description:** Verify that a new class can be created successfully.
  - **Steps:**
    1. Navigate to the "Add Class" page.
    2. Enter valid class details.
    3. Submit the form.
  - **Expected Result:** The class is created, and a success message is displayed.
- **Test Case 2:** Assign a subject to a class
  - **Description:** Verify that a subject can be assigned to a class.
  - **Steps:**
    1. Navigate to the "Assign Subject" page.
    2. Select a class and a subject.
    3. Submit the form.
  - **Expected Result:** The subject is assigned to the class, and a success message is displayed.

#### 7.4.3. Attendance Management Test Cases

- **Test Case 1:** Mark attendance for a class
  - **Description:** Verify that attendance can be marked for a class.
  - **Steps:**
    1. Navigate to the "Mark Attendance" page.
    2. Select a class and date.

3. Mark attendance for each student.
  4. Submit the form.
- **Expected Result:** Attendance is marked, and a success message is displayed.
- **Test Case 2:** View attendance records
    - **Description:** Verify that attendance records can be viewed.
    - **Steps:**
      1. Navigate to the "View Attendance" page.
      2. Select a class and date range.
      3. View the attendance records.
    - **Expected Result:** The attendance records are displayed correctly.

## 7.5. Testing Tools

The following tools were utilized during the testing phase of the GU Course Folder Management System:

- **PHPUnit:** Used for unit and integration testing of PHP code.
- **Laravel's Built-in Testing Tools:** Utilized for both unit and integration testing within the Laravel framework.
- **Selenium:** Used for automated system testing, particularly for testing the user interface in different browsers.
- **Manual Testing:** Conducted by the development team and end-users for acceptance testing and usability testing.

## 7.6. Testing Results

The testing phase yielded the following results:

- **Unit Testing:** All unit tests passed successfully, indicating that individual components are functioning correctly.

- **Integration Testing:** Integration tests were successful, confirming that components work together as expected.
- **System Testing:** System tests revealed a few minor issues, which were promptly addressed. The system is now stable and performs well under various scenarios.
- **Acceptance Testing:** End-users and stakeholders conducted acceptance testing and provided positive feedback. The system met all acceptance criteria and is ready for deployment.

## 7.7. Chapter Summary

Testing is a vital part of the software development process, ensuring that the GU Course Folder Management System is reliable, functional, and ready for use. By employing a combination of unit, integration, system, and acceptance testing, we were able to identify and resolve issues, resulting in a robust and high-quality application. The thorough testing process guarantees that the system meets the needs of administrators, teachers, and students, providing a seamless and efficient course management experience.

# **CHAPTER 8**

## **RESULTS**

## 8.1. Introduction

This chapter summarizes the outcomes of developing and deploying the GU Course Folder Management System. It evaluates the system against the initial goals and requirements, presents the benefits realized by its users, and discusses any limitations or areas for future improvement.

## 8.2. Achievement of Objectives

The primary objective of the GU Course Folder Management System was to streamline and enhance the academic and administrative processes within the educational institution. The following sections detail how the system met these objectives:

### 8.3. Improved Administrative Efficiency

- **Centralized Management:** The system provided a centralized platform for managing user records, classes, subjects, and attendance, significantly reducing the administrative burden.
- **Automated Processes:** Automated processes replaced manual tasks, such as tracking attendance and managing academic resources, leading to time savings and reduced errors.

### 8.4. Enhanced Academic Management

- **Resource Accessibility:** Teachers could easily upload and organize academic resources, and students could access these materials anytime, facilitating better learning experiences.
- **Attendance Tracking:** The system enabled accurate and efficient attendance tracking, with instant generation of attendance reports for administrators and teachers.

### 8.5. Effective Communication

- **Notifications:** Automated notifications ensured timely communication between administrators, teachers, and students regarding updates on assignments, quizzes, and other academic activities.
- **User Interface:** A user-friendly interface made it easy for all users to navigate and utilize the system's features effectively.

### 8.6. Benefits to Users

The GU Course Folder Management System delivered several benefits to its primary users, including administrators, teachers, and students:

### 8.6.1. Administrators

- **Streamlined Operations:** Reduced paperwork and manual tracking, leading to more efficient administrative operations.
- **Better Oversight:** Enhanced visibility and control over academic and administrative activities.

### 8.6.2. Teachers

- **Simplified Management:** Easy management of classes, subjects, and academic resources.
- **Focus on Teaching:** Reduced administrative workload allowed teachers to focus more on teaching and student engagement.
- **Timely Updates:** Notifications kept students informed about important academic activities and deadlines.

## 8.7. Performance Evaluation

The performance of the GU Course Folder Management System was evaluated based on several criteria, including functionality, usability, and reliability.

### Functionality

- **Comprehensive Features:** The system included all the essential features required for effective course and folder management.
- **Meeting Requirements:** All functional requirements, such as user management, class and subject management, attendance tracking, and resource management, were successfully implemented and tested.

### Usability

- **User-Friendly Interface:** Positive feedback from users indicated that the system was intuitive and easy to use.
- **Minimal Training:** The system required minimal training for users to become proficient in its use.

## Reliability

- **Stable Performance:** The system demonstrated stable performance during testing and deployment, with no major issues reported.
- **High Availability:** The system achieved the target uptime, ensuring it was available to users whenever needed.

### 8.8. Limitations and Areas for Improvement

While the GU Course Folder Management System successfully achieved its objectives, certain limitations were identified, and areas for future improvement were noted:

#### Limitations

- **Scalability Constraints:** While the system was scalable, further optimization might be required to handle significantly larger user bases and data volumes.
- **Feature Limitations:** Certain advanced features, such as detailed analytics and custom reporting, were not included in the initial version.

#### Areas for Improvement

- **Enhanced Analytics:** Future versions of the system could include advanced analytics features to provide deeper insights into academic performance and administrative efficiency.
- **Mobile Compatibility:** Developing a mobile application or optimizing the web application for mobile devices would enhance accessibility for all users.
- **User Feedback Integration:** Continuously gathering and integrating user feedback could help identify new features and improvements to enhance user satisfaction.

### 8.9. Chapter Summary

The GU Course Folder Management System successfully transformed the traditional, manual processes of course and folder management into a streamlined, efficient, and user-friendly digital platform. By achieving its objectives and delivering significant benefits to its users, the system demonstrated its value as a modern tool for educational institutions. The insights gained and

feedback received during this project provide a strong foundation for future enhancements, ensuring the system continues to meet the evolving needs of its users.

# REFERENCES

## 1. Laravel Documentation

- Laravel. (n.d.). The PHP Framework for Web Artisans. Retrieved from <https://laravel.com/docs>

## 2. PHP: Hypertext Preprocessor

- PHP Group. (n.d.). PHP Manual. Retrieved from <https://www.php.net/manual/en/>

## 3. MySQL Documentation

- Oracle Corporation. (n.d.). MySQL Reference Manual. Retrieved from <https://dev.mysql.com/doc/>

## 4. PHPUnit

- Sebastian Bergmann. (n.d.). PHPUnit Manual. Retrieved from <https://phpunit.de/manual/current/en/>

## 5. XAMPP Documentation

- Apache Friends. (n.d.). XAMPP. Retrieved from <https://www.apachefriends.org/index.html>

## 6. Selenium Documentation

- SeleniumHQ. (n.d.). Selenium WebDriver. Retrieved from <https://www.selenium.dev/documentation/en/>

## 7. phpMyAdmin Documentation

- phpMyAdmin. (n.d.). phpMyAdmin Documentation. Retrieved from <https://www.phpmyadmin.net/docs/>

## 8. HTML and CSS Standards

## References

---

- World Wide Web Consortium (W3C). (n.d.). HTML & CSS. Retrieved from <https://www.w3.org/standards/webdesign/htmlcss>

### **9. JavaScript Documentation**

- Mozilla Developer Network (MDN). (n.d.). JavaScript Guide. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

### **10. Software Engineering: A Practitioner's Approach**

- Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

### **11. Database Management Systems**

- Ramakrishnan, R., & Gehrke, J. (2002). Database Management Systems (3rd ed.). McGraw-Hill.

### **12. Agile Software Development with Scrum**

- Schwaber, K., & Beedle, M. (2001). Agile Software Development with Scrum. Prentice Hall.

### **13. Design Patterns: Elements of Reusable Object-Oriented Software**

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.