

A project

TUITION MANAGEMENT SYSTEM



CSE 318: System Analysis and Design Lab

SUBMITTED BY

Name	ID	Intake
Md. Shahuidul Islam Parvez	22234103123	50
Noor Uddin Yousuf Shanto	22234103136	50
Akkas Uddin	22234103137	50
Md. Rahat Bin Israil	22234103139	50
Md. Saidul Islam	22234103142	50
Jannatul Ferdous Prity	22234103239	50
Md. Jahid Hossain	21224103155	48

SUPERVISED BY:

Md. Masudul Islam

*Assistant Professor,
Department of CSE*

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY
(BUBT)**

11th, December, 2024

Abstract

The Tutoring Management System is a comprehensive software solution designed to streamline the operations and management of tutoring services. This project aims to develop a user-friendly platform that enables tutors, students, and administrators to efficiently manage tutoring sessions, track student progress, and facilitate seamless communication.

The system will provide a centralized database to store and manage student and tutor information, including profiles, schedules, and session details to enhance the overall tutoring experience.

The key objectives of this project are to improve the efficiency of tutoring services, enhance the communication and collaboration between tutors and students, and provide a robust data management system to support the growth and development of the tutoring program.

Through the implementation of the Tutoring Management System, we aim to streamline the tutoring process, improve student outcomes, and contribute to the overall success of the educational institution.

List of Figures

Sl.No.	Figure Name	Page
1.	Agile Development Methodology Model	10
2.	System Architecture	12
3.	Context Level Diagram	12
4.	Use Case Diagram	13
5.	Data Flow Diagram	14
6.	System Flowchart	15

Table of Contents

Abstract	2
List of Figures	3
Chapter: 1 Introduction	5-6
1.1. Problem Specification/Statement	
1.2. Objectives	
1.3. Organization Of Project Report	
Chapter: 2 Background	7-9
2.1. Existing System Analysis	
2.2. Supporting Literatures	
Chapter: 3 System Analysis & Design	10-15
3.1. Technology & Tools	
3.2. Model & Diagram	
3.2.1. Model Agile	
3.2.2. System Architecture	
3.2.2. Context Level Diagram	
3.2.3. Use Case Diagram	
3.2.4. Data Flow Diagram	
3.2.5. System Flowchart	
Chapter: 4 Implementation	16-17
4.1. Modules	
Chapter: 5 User Manual	18-22
5.1. System Requirement	
5.1.1. Hardware Requirement	
5.1.2. Software Requirement	
5.2. User Interfaces/Prototypes	
5.2.1. Home Page	
5.2.2. Sign up Page	
5.2.3. Login Page	
5.2.4. Dashboard	
5.2.5. My Courses	
5.2.6. Enroll Courses	
5.2.7. Score	
5.2.8. Exit	
Chapter: 6 Conclusion	23
6.1. Conclusion	
6.2. Limitation	
6.3. Future Works	
References	24

Chapter 1

INTRODUCTION

1.1. Problem Specification

The system has several critical issues that need to be addressed. Firstly, it lacks the ability to provide different interfaces for students, tutors, and admins; it operates with a single interface that does not change based on user type during signup or sign in. This limitation means that the system currently only supports an admin-centric approach, which is not ideal as it does not cater to students and tutors effectively. Additionally, there is no payment system integrated, which means transactions cannot be processed within the app. The system also lacks any report analysis features, making it difficult to generate and view detailed reports on performance and other metrics. Furthermore, tutors are unable to offer courses directly, restricting their ability to manage and contribute to course content. These issues significantly impact the system's functionality and overall user experience.

1.2. Objectives

- i. Enhance Student and Course Management.
- ii. Improve Accessibility .
- iii. Modernize the User Interface.
- iv. Integrate with Learning Management Systems (LMS) .
- v. Introduce Virtual Tutoring Features.
- vi. Incorporate Data Analytics and Reporting.
- vii. Integrate Gamification Elements.
- viii. Implement a Feedback and Rating System.
- ix. Provide Continuous Training and Support.

1.3. Organization Of Project Report

The project report is organized into six chapters, providing a comprehensive overview of the development and implementation of the C# Desktop-based Tuition Management System. Below is a brief summary of each chapter's contents and outcomes:

Chapter 1: Introduction

- Defines the problem, objectives, and scope of the project, setting the foundation for addressing system limitations.

Chapter 2: Background

- Analyzes existing systems, discusses their pros and cons, and reviews the theoretical and technological principles applied.

Chapter 3: System Analysis & Design

- Explains the tools, technologies, and Agile methodology used, supported by system diagrams System Architecture, Context Level Diagram, Use Case Diagram, Data Flow Diagram and System Flowchart for better visualization.

Chapter 4: Implementation

- Details the developed modules and how they address the project objectives with functional system features.

Chapter 5: User Manual

- Provides system requirements, interface snapshots, and instructions for navigating the system's functionalities.

Chapter 6: Conclusion

- Summarizes project findings, highlights limitations, and proposes potential future enhancements.

Chapter 2

BACKGROUND

2.1.Existing System Analysis

To better understand the landscape of tuition management systems, three existing systems were studied: **Preply**, **Tutor.com**, and **Google Classroom**. These systems offer a variety of features for managing tutoring sessions and student interactions. Below is a brief discussion of their pros and cons.

2.1.1. Preply [1]

Description:

Preply is an online tutoring platform that connects students with tutors worldwide. It offers scheduling, payment integration, and video conferencing tools.

- **Pros:**
 - i. Global reach with access to diverse tutors.
 - ii. Built-in scheduling and payment processing.
 - iii. User-friendly interface for both tutors and students.
- **Cons:**
 - i. Focused primarily on one-on-one online tutoring, limiting its use for group classes.
 - ii. High commission fees for tutors.
 - iii. Limited offline functionality, making it unsuitable for centers without strong internet connectivity.

2.1.2. Tutor.com [2]

Description:

Tutor.com is a digital platform offering live tutoring sessions for various subjects, primarily targeting school and college students.

- **Pros:**
 - i. On-demand tutoring available 24/7.
 - ii. Wide range of subjects and professional tutors.
 - iii. Integrated tools for tracking student progress.
- **Cons:**
 - i. Subscription-based model may not be affordable for all.
 - ii. Focused on digital sessions, not accommodating for offline classes.
 - iii. Limited customization for specific tutoring center needs.

2.1.3. Google Classroom [3]

Description:

Google Classroom is a free platform for managing educational workflows, allowing teachers to assign tasks and interact with students.

- **Pros:**
 - i. Free and widely accessible.
 - ii. Seamless integration with other Google services (Docs, Drive, etc.).
 - iii. Supports collaborative learning.
- **Cons:**
 - i. Requires a Google account for access.
 - ii. Not specifically designed for tutoring; limited in tracking payments or managing sessions.
 - iii. Lack of advanced features for private tutoring centers, such as automated scheduling or attendance tracking.

While these systems are powerful in their respective domains, they lack certain features critical for managing a private tutoring center. For instance, offline functionality, customized payment tracking, and tailored student-teacher management are not adequately addressed. These gaps highlight the need for a C# desktop-based Tuition Management System, which will focus on providing a comprehensive solution tailored to the needs of a tutoring center.

2.2.Supporting Literatures

The development of the C# Desktop-based Tuition Management System is grounded in various theoretical, mathematical, methodological, and technological principles. These elements were carefully selected to ensure the system meets the requirements of a tutoring center effectively. Below is a brief discussion of the applied knowledge and tools:

2.2.1. Theoretical Knowledge

- i. **Object-Oriented Programming (OOP):** The project is built using C#, leveraging OOP principles such as encapsulation, inheritance, and polymorphism. This approach ensures modularity, scalability, and ease of maintenance.
- ii. **System Analysis and Design:** Understanding the workflow of a tutoring center helped in designing a system that addresses user needs, including student enrollment, session scheduling, and payment management.

2.2.2. Mathematical Knowledge

- i. **Data Calculations:** Mathematical concepts are applied for calculating total fees, generating monthly payment summaries, and tracking overdue payments.
- ii. **Database Queries:** SQL is used for performing mathematical operations, such as aggregations (e.g., SUM, AVG), to analyze student performance and fee trends.

2.2.3. Methodological Knowledge

- i. **Agile Development Model:** Agile methodology was chosen for its iterative approach, allowing continuous improvement and adaptation to user feedback. It enables

incremental delivery of functional features, such as student registration and attendance tracking.

- ii. **Unified Modeling Language (UML):** UML diagrams (e.g., use case, class, and sequence diagrams) were used to visualize system design and functionality.

2.2.4. Technological Knowledge

- i. **Programming Language:**
 - **C#:** Selected for its robustness, ease of integration with desktop applications, and extensive libraries.
- ii. **Database Management:**
 - **SQL Server:** Chosen for its reliability, scalability, and compatibility with C# for data storage and management.
- iii. **User Interface (UI):**
 - **Windows Forms:** Used to create a user-friendly interface for both tutors and administrators.
- iv. **Version Control:**
 - **Git:** Employed to track changes in code, ensuring collaboration and rollback capabilities.

2.2. 5. Tools and Techniques

- i. **Integrated Development Environment (IDE):**
 - **Visual Studio:** Selected for its powerful debugging tools, ease of use, and seamless integration with C#.
- ii. **Data Validation:** Applied techniques to ensure input accuracy, such as validating student records, session timings, and fee amounts.

Justification for Choices

- i. **C# and XAMPP Server:** These tools were chosen for their compatibility and ability to handle the complexity of managing a tutoring center's data.
- ii. **Agile Development Model:** Its flexibility allowed iterative development, ensuring the final product met evolving requirements.

Chapter 3

SYSTEM ANALYSIS & DESIGN

3.1. Technology & Tools

The development of the C# Desktop-based Tuition Management System involved the use of various software and hardware tools to ensure an efficient, reliable, and scalable system. Below is an overview of the technologies and tools used in this project:

- i. **User Interface Design:**
 - **Windows Forms:** Simplified development of desktop application interfaces.
- ii. **Modeling Tools:**
 - **Draw.io:** Used to create UML diagrams for system design.
- iii. **Testing Frameworks:**
 - **MSTest:** Utilized for unit testing the application logic.
- iv. **Libraries and APIs:**
 - Standard libraries included in the .NET framework, such as `System.Data.SqlClient` for database operations.
- v. **Error Logging and Debugging:**
 - Built-in Visual Studio debugging tools and logging mechanisms for error tracking.

3.2. Model & Diagram

3.2.1. Model (Agile) [4]

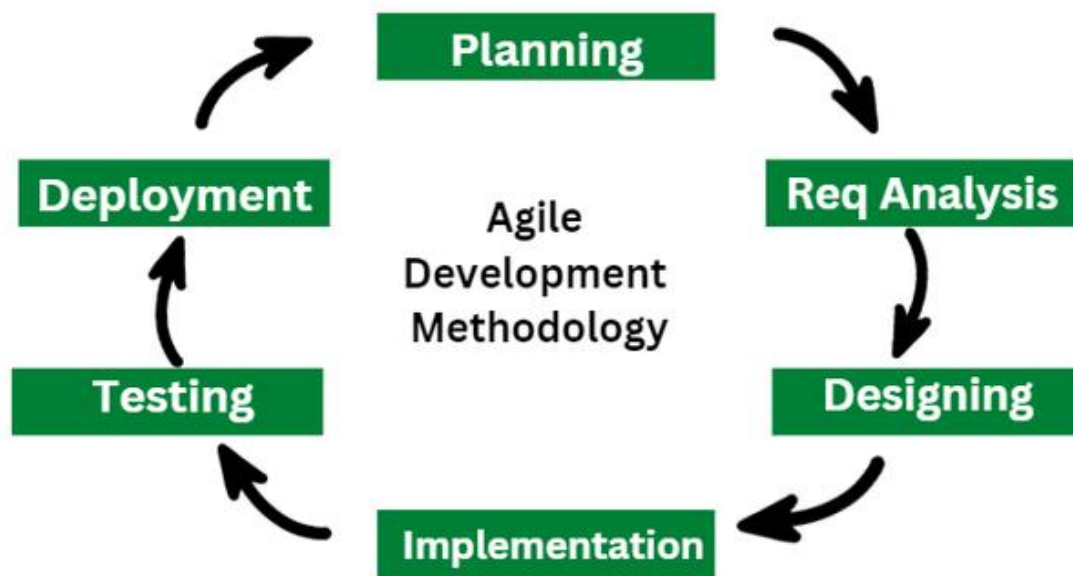


Fig 1 : Agile Development Methodology Model

The Agile Development Model was chosen for the development of the C# Desktop-based Tuition Management System. Agile is an iterative and incremental software development methodology that focuses on collaboration, flexibility, and continuous delivery.

Why Agile?

The project required frequent refinements based on evolving requirements, such as incorporating additional features like payment tracking or customized student reports. Agile allowed for adaptability and quick feedback integration throughout the development process.

Key Features of Agile Applied in This Project

1. **Iterative Development:**
 - The project was divided into multiple iterations or "sprints," each focusing on specific functionalities, such as student enrollment, session scheduling, and payment tracking.
2. **Continuous Feedback:**
 - Stakeholder feedback was gathered after every sprint to refine the system's features and address any issues promptly.
3. **Collaboration:**
 - Frequent communication among team members ensured alignment on goals and quick resolution of challenges.
4. **Incremental Delivery:**
 - Functional modules were delivered and tested incrementally, allowing early detection and resolution of errors.

Benefits of Agile in the Project

1. **Efficiency:**
 - The iterative approach ensured that the project stayed on track and allowed efficient use of resources. By breaking down the project into smaller tasks, development time was optimized, and each module was delivered with higher quality.
2. **Flexibility:**
 - Agile's adaptability made it easier to incorporate changes, such as modifying the database schema or enhancing UI elements based on user needs.
3. **Risk Reduction:**
 - Continuous testing and feedback at every stage minimized the risk of encountering major issues in the final product.
4. **User-Centric Design:**
 - The iterative nature ensured that the final system met the expectations of its users, such as tutors and administrators.
5. **Early Value Delivery:**
 - Functional components, like the student registration module, were delivered early in the project timeline, enabling partial system use while other features were developed.

3.2.2. System Architecture [5]

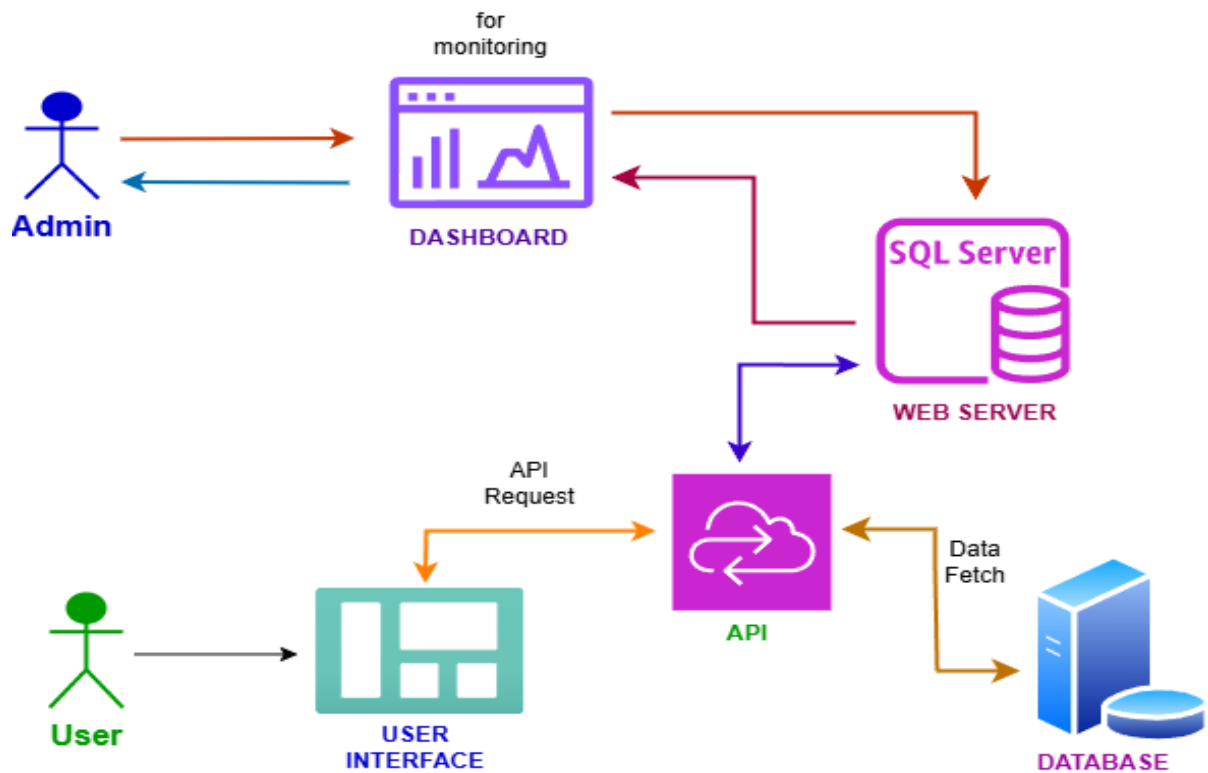


Fig 2 : System Architecture

3.2.3. Context Level Diagram [6]

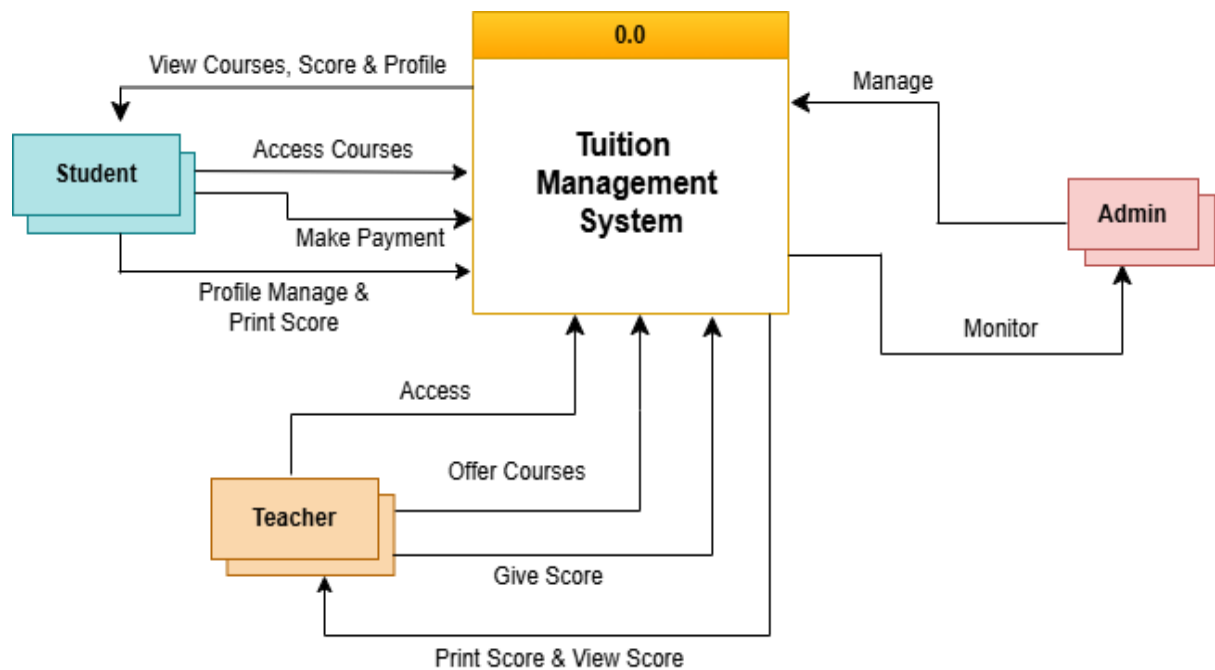


Fig 3 : Context Level Diagram

3.2.4. Use Case Diagram [7]

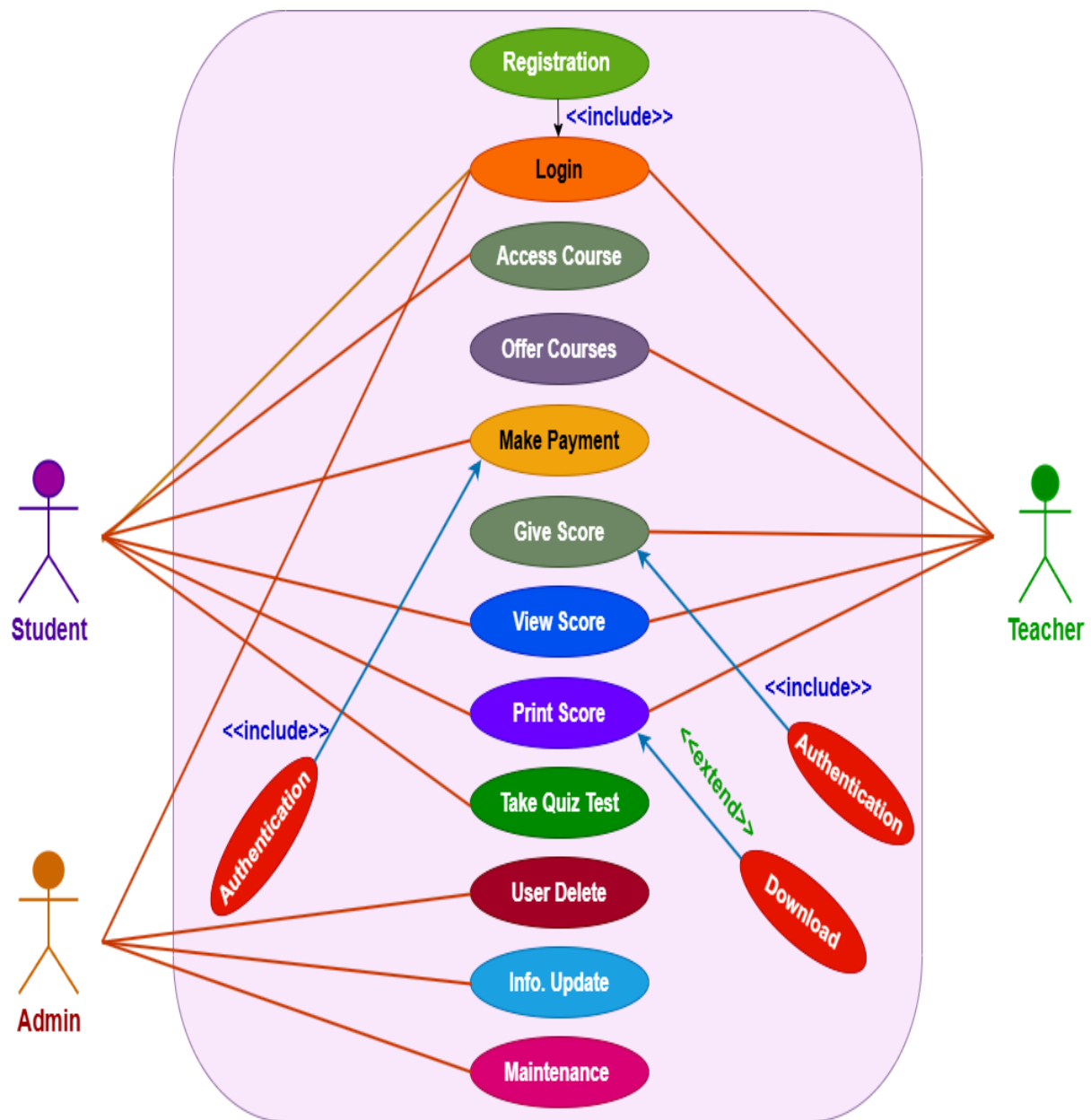


Fig 4 : Use Case Diagram

3.2.5. Data Flow Diagram [8]

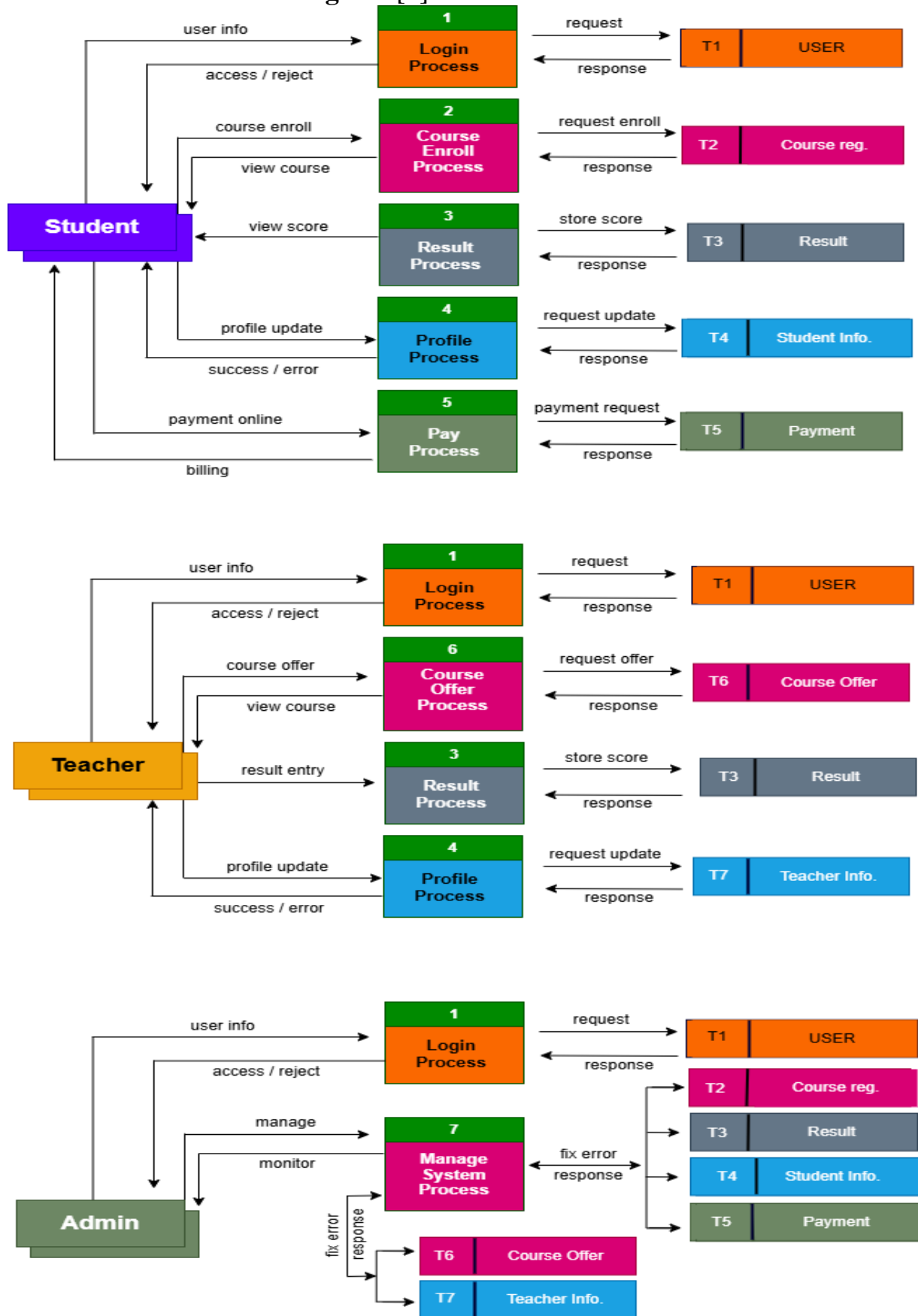


Fig 5 : Data Flow Diagram

3.2.6. System Flowchart [9]

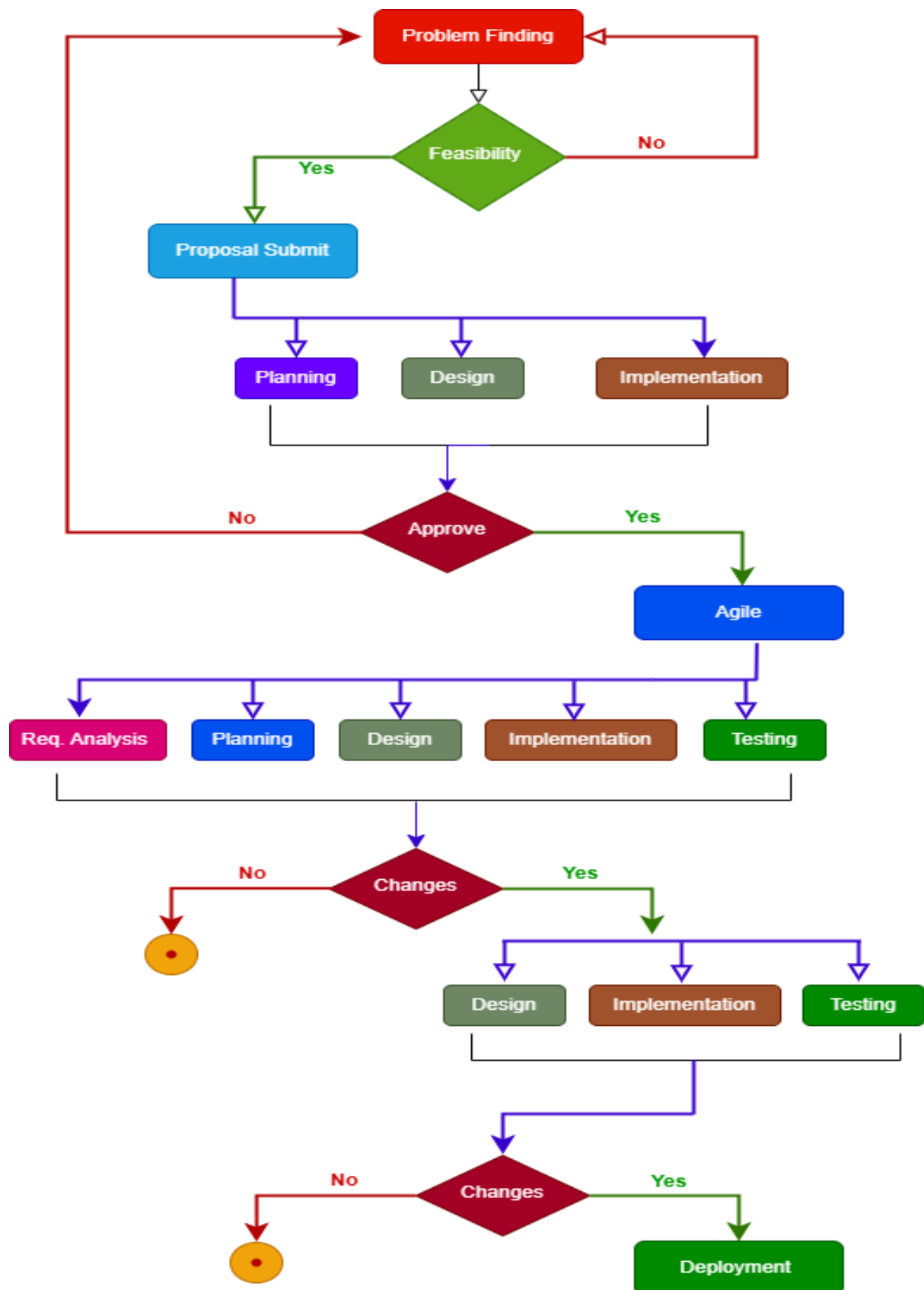


Fig 6 : System Flowchart

Chapter 4

IMPLEMENTATION

4.1. Modules

The C# Desktop-based Tuition Management System is structured with distinct modules to cater to the needs of students, teachers, and administrators. Each module is designed to provide specific functionalities, ensuring an organized and user-friendly experience. Below is a detailed description of the implemented modules:

4.1.1. Student Management Module

Description:

This module allows students to manage their profiles, access courses, take assessments, and view academic progress.

Features:

- i. **Course Access:** Students can enroll in courses after completing the payment process.
- ii. **Profile Management:** Students can update personal details such as contact information and preferences.
- iii. **Assessment Participation:** Students can take quizzes and exams scheduled by their teachers.
- iv. **Result Viewing:** Students can view subject-wise results, print or download scorecards for future reference.
- v. **Signup Process:** Students register with a unique ID, ensuring secure and personalized access to the system.

4.1.2. Teacher Management Module

Description:

This module provides teachers with tools to manage their courses, grade assessments, and monitor student performance.

Features:

- i. **System Access:** Teachers can sign up or log in using their unique ID.
- ii. **Course Management:** Teachers can offer new courses and manage the curriculum.
- iii. **Exam Evaluation:** Teachers can evaluate quizzes and exams, update marks, and provide feedback.
- iv. **Performance Tracking:** Teachers can view scores for specific students or groups, and print or download score summaries for their records.

4.1.3. Admin Management Module

Description:

The admin module is responsible for overseeing the entire system, ensuring smooth operation, and maintaining control over all users.

Features:

- i. **System Monitoring:** The administrator has access to all data within the system, including student and teacher activities.
- ii. **User Management:** Admins can manage user accounts, ensuring proper access permissions.
- iii. **System Maintenance:** Admins handle any necessary updates, backups, and troubleshooting to keep the system functional.

4.1.4. Authentication and Interface Module

Description:

This module ensures secure and seamless access for all users, providing separate interfaces for students and teachers.

Features:

- i. **Unique E-mail or Phone -Based Signup:** Both students and teachers register with unique E-mail & Phone for secure access.
- ii. **Role-Based Interfaces:**
 - o **Student Interface:** Tailored for course enrollment, profile management, assessments, and result viewing.
 - o **Teacher Interface:** Designed for course creation, grading, and performance monitoring.
- iii. **Secure Login:** Implements password encryption and role-based authentication to protect user data.

The modular design ensures that each user type has access to functionalities specific to their role, providing a streamlined experience while maintaining system integrity.

Chapter 5

USER MANUAL

5.1. System Requirement

5.1.1. Hardware Requirement

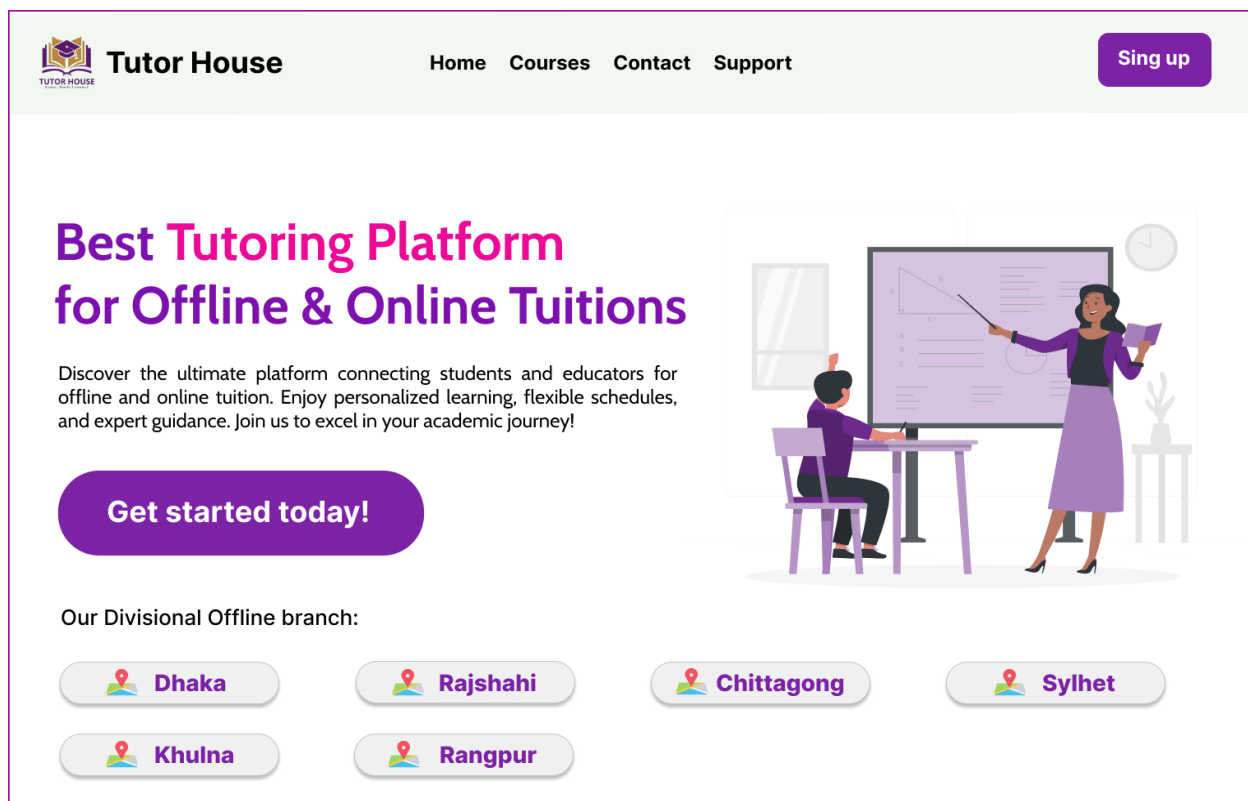
- i. CPU: Core i3 4th Generation or equivalent (or better).
- ii. Disk Space: 200 MB or more.
- iii. RAM: 1 GB or better.
- iv. Monitor: 15” VGA Color or better.

5.1.2. Software Requirement


- i. Operating System: Windows 7/8/10/11.
- ii. Development Environment: Microsoft Visual Studio (for C# development).
- iii. Backend: XAMPP, SQL Server (or any SQL-based server compatible with the application).
- iv. Technology: ASP.NET (for web functionalities), C# (for application logic).
- v. Database Management: XAMPP, SQL Server.

5.2. User Interfaces [10]

5.2.1. Home Page

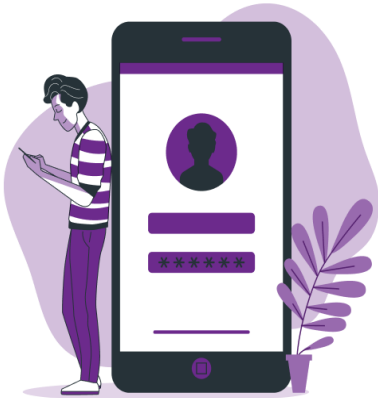


5.2.2. Sing up Page


**Tutor House**


Home Courses Contact Support

Log inSing up



Sing up

☐ Teacher

☒ Student

Name *

Phone *

Email *


Location

Password *

Re-type Password *


☐ I agree to Tutor House [Terms of Service](#) and [Privacy Policy](#).

Sing up

 Continue with Google


Already have an account? [Sing in](#)

5.2.3. Login Page


**Tutor House**


Home Courses Contact Support

Log inSing up



Sing in


☐ Teacher

☒ Student

Email or Phone *


Password *


Sing in


 Continue with Google


Don't have an account? [Sing up](#)


5.2.4. Dashboard


**Tutor House**


English  RAHAT ▾


**Rahat Bin Israil**
rahmat.edu.cse@gmail.com
Student ID: 139


 Dashboard


 Status


 My Courses


 Quiz


 Score

 Enroll Courses

 Payment History


 Settings





 Log Out


 Exit


Notice Board
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fuga, asperiores ratione tempore voluptatem aliquid dolore aperiam quod illum vel temporibus! voluptatem aliquid Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
11 DEC 2024 [See more](#)

Tutor Hub
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fuga, asperiores ratione tempore voluptatem aliquid dolore aperiam quod illum vel temporibus! voluptatem aliquid Ut enim ad minim veniam.
[See →](#)


**70%**
Your Profile setup 70% done.
Complete the full setup.
[Update →](#)


Contact us



Join a Course

[Join Now →](#)


 **Zaya AI**


5.2.5. My Courses


**Tutor House**


English  RAHAT ▾


**Rahat Bin Israil**
rahmat.edu.cse@gmail.com
Student ID: 139


 Dashboard


 Status


 My Courses


 Quiz


 Score

 Enroll Courses

 Payment History

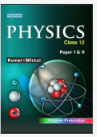
 Settings

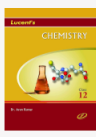
 Log Out

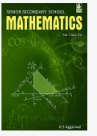
 Exit

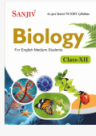
Categories ▾

Search 🔍


**Physics**
Progress **50%**
[Continue →](#)

**Chemistry**
Progress **70%**
[Continue →](#)


**Math**
Progress **20%**
[Continue →](#)


**Biology**
Progress **07%**
[Continue →](#)


[See More →](#)

 **Zaya AI**

5.2.6. Enroll Courses



Tutor House


English

RAHAT



Rahat Bin Israil
rahata.edu.cse@gmail.com
Student ID: 139

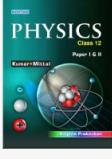
Categories

Search here
Search


ICT
● 50+ Live Class
Price: 3000 ₳
Details Enroll



Higher Math
● 90+ Live Class
Price: 5500 ₳
Enroll



English
● 80+ Live Class
Price: 5000 ₳
Details Enroll



Physics
● 90+ Live Class
Price: 5500 ₳
Continue

See More →

5.2.7. Score


Tutor House

English

RAHAT


Rahat Bin Israil
rahata.edu.cse@gmail.com
Student ID: 139

Search here
Search

ID	First Name	Course Name	Score	Grade
139	Rahat	Physics	80	A+
139	Rahat	Chemistry	75	A
139	Rahat	Math	70	A-
139	Rahat	Biology	90	A+

Download
Print

5.2.8. Exit

The screenshot displays the Tutor House web application interface. At the top, the 'Tutor House' logo is on the left, and a language selector set to 'English' and a user profile for 'RAHAT' are on the right. A search bar is located below the language selector. On the left side, a vertical sidebar menu contains the following items: Dashboard, Status, My Courses, Quiz, Score, Enroll Courses, Payment History, Settings, Log Out, and Exit (which is highlighted in purple). The main content area is partially obscured by a white modal dialog box in the center. The dialog box contains the text 'Are you sure you want to Exit ?' and two buttons: a green 'Stay X' button and a red 'Exit →' button. To the right of the dialog, a table with a single column titled 'Grade' is visible, containing four rows with the values 'A+', 'A', 'A-', and 'A+'. At the bottom right of the main content area, there are two buttons: 'Download' with a download icon and 'Print' with a printer icon.

Tutor House

English RAHAT

Search here Search

Rahat Bin Israil
rahat.edu.cse@gmail.com
Student ID: 139

Dashboard
Status
My Courses
Quiz
Score
Enroll Courses
Payment History
Settings
Log Out
Exit

Are you sure you want to Exit ?

Stay X Exit →

Grade
A+
A
A-
A+

Download Print

Chapter 6

CONCLUSION

6.1. Conclusion

The C# Desktop-based Tuition Management System was developed as a robust solution for tutoring centers, addressing challenges like student management, session scheduling, and payment tracking using C#, SQL Server, and Windows Forms. The Agile Development Model ensured flexibility, iterative improvements, and user satisfaction. Its modular design and scalability allow for future enhancements, such as advanced reporting and online platform integration. Overall, the project effectively applies theoretical and technological knowledge, offering a practical solution for efficient tutoring operations.

6.2. Limitations

- i. **Limited Advanced Features:** Since the focus is on low-cost development, some advanced features like AI-driven recommendations or real-time analytics may not be included in the initial version.
- ii. **Basic User Interface:** The user interface might be simpler and less polished, given the budget and limited resources.
- iii. **Database Limitations:** Handling large datasets may lead to performance bottlenecks, particularly as the system scales.

6.3. Future Works

- i. **Enhanced Reporting and Analytics:** Future updates could focus on deeper insights into student performance, tutor effectiveness, and course analytics.
- ii. **Mobile App Development:** Extend the system to mobile platforms with additional features like push notifications and mobile payments.
- iii. **Cloud Migration:** If the user base grows, migrate the system to a scalable cloud infrastructure to handle larger data volumes more efficiently.
- iv. **Gamification:**
 - **Badges and Achievements:** Rewarding students for completing courses, achieving high scores, or maintaining consistent attendance.
 - **Leaderboards:** Encouraging healthy competition by displaying top-performing students in quizzes, exams, or overall performance.
 - **Interactive Quizzes:** Adding timed challenges and rewards for quick and accurate responses.
 - **Progress Tracking:** Allowing students to visualize their learning journey with milestone-based progress indicators.

References

- [1] Preply "Online Learning Platform." Retrieve from: <https://preply.com>
- [2] Tutor.com "Online Tutoring Services." Retrieve from: <https://www.tutor.com>
- [3] Google Classroom "Google Classroom Workspace." Retrieve from: <https://sites.google.com/view/classroom-workspace>
- [4] Javatpoint "Agile Model in Software Engineering." Retrieve from: <https://www.javatpoint.com/software-engineering-agile-model>
- [5] The System Architecture was designed using diagrams created in draw.io ([diagrams.net](https://draw.io)).
- [6] The Context Level Diagram was designed using diagrams created in draw.io ([diagrams.net](https://draw.io)).
- [7] The Use Case Diagram was designed using diagrams created in draw.io ([diagrams.net](https://draw.io)).
- [8] The Data Flow Diagram was designed using diagrams created in draw.io ([diagrams.net](https://draw.io)).
- [9] The System Flowchart was designed using diagrams created in draw.io ([diagrams.net](https://draw.io)).
- [10] All User Interfaces were created using [figma.com](https://www.figma.com)