1. Problems

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.

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.

3. Existing System Analysis

To further identify limitations in popular tutoring and educational platforms, this analysis examines Preply, Tutor.com, and Google Classroom. Each system has its strengths but also suffers from similar shortcomings, particularly in areas like user experience, engagement, analytics, and payment integration.

3.1. Preply [Ref: preply.com]

Description: Preply is an international online tutoring platform primarily focused on language learning. It connects students with tutors for personalized lessons via video chat, allowing for flexible scheduling.

Shortcomings: Unified Interface: Preply uses the same interface for students, tutors, and admins, which hinders role-specific functionality and personalization.

No Gamification: The platform lacks interactive games or exercises, which could enhance learning experiences and motivate students.

Limited Reporting and Analytics: The system provides only basic progress reports, making it difficult to track tutor effectiveness or student improvement.

No In-App Payment Gateway: Payment is processed externally, which may complicate transactions and lead to security issues.

3.2. Tutor.com [Ref: tutor.com]

Description: Tutor.com offers on-demand tutoring services with a focus on real-time academic assistance across various subjects. It supports scheduled and instant tutoring sessions, primarily via text and chat.

Shortcomings: Single Interface for All Roles: Similar to Preply, Tutor.com uses a generalized interface for students, tutors, and administrators, reducing the user experience's efficiency and personalization.

No Gamification: There are no gamified or interactive learning elements, which can make the learning process feel less engaging.

Basic Reporting: The platform provides limited data on student progress and tutor performance, restricting the ability to assess outcomes effectively.

External Payment Handling: Payment is managed externally, which can reduce the smoothness of transactions and pose potential security risks.

3.3. Google Classroom [Ref: sites.google.com/view/classroom-workspace]

Description: Google Classroom is an education-focused platform designed to help teachers and students manage coursework. It integrates with other Google tools like Docs and Sheets, simplifying assignments, grading, and communication.

Shortcomings: No Specialized Tutoring Features: Google Classroom is not optimized for one-on-one tutoring. It lacks essential features like virtual whiteboards and tools for live tutoring sessions.

Lack of Advanced Analytics: While it tracks basic assignment completion and grading, it lacks advanced reporting tools to provide deeper insights into student performance or tutoring effectiveness.

No Gamification: The platform does not include gamification features, missing an opportunity to make learning more engaging and interactive for students.

The analysis of these systems shows that they all suffer from similar shortcomings:

1.Unified Interface Issues: Preply and Tutor.com use the same interface for all user roles, reducing usability and efficiency for students, tutors, and administrators.

2.Lack of Engagement Tools: None of the platforms include gamification or other interactive learning features, missing opportunities to boost student engagement and motivation.

3.Basic Reporting Features: Reporting tools across these platforms are rudimentary, limiting the ability to track student progress or tutor performance in detail.

4.External Payment Processing: Both Preply and Tutor.com rely on external payment systems, which could lead to transactional inefficiencies and security concerns.

4. Features

4.1 Student Management:

- 4.1.1 Registration: Allow the addition of new students with personal details, ensuring secure storage of student records.
- 4.1.2 Management: Enable updating of student information and provide search and filter options based on criteria like name and address.
- 4.1.3 Reporting: Generate and print student information reports.

4.2 Course and Teacher Management:

- 4.2.1 Course Registration: Add new courses with detailed information and assign tutors to specific courses. Associate students with courses.
- 4.2.2 Course Management: Update course details, search and filter courses, and manage course information.
- 4.2.3 Course Reporting: Generate and print course reports, including enrolled students and assigned tutors.

4.3 Score Management:

- 4.3.1 Recording Scores: Record and associate student scores with their IDs and course details, including score descriptions.
- 4.3.2 Score Management: Update score details, search, and filter scores by name or course.
- 4.3.3 Score Reporting: Generate comprehensive student result reports.

4.4 Dashboard:

- 4.4.1 Student Statistics: Display total student enrollment and gender distribution.
- 4.4.2 Course Enrollment Trends: Track and display student enrollment trends across courses.

4.5 Enhanced User Interface:

- 4.5.1 Update the interface to be more intuitive and user-friendly, incorporating modern design principles.

4.6 Mobile Application:

- 4.6.1 Develop a mobile version of the system to allow access on-the-go.

4.7 Integration with LMS:

- 4.7.1 Integrate the system with popular LMS platforms for seamless data sharing and progress tracking.

4.8 AI-Powered Recommendations:

- 4.8.1 Implement AI algorithms to provide personalized tutor recommendations based on student preferences and performance data.

4.9 Virtual Tutoring Features:

- 4.9.1 Introduce virtual tutoring capabilities, including video conferencing and virtual whiteboards.

4.10 Data Analytics and Reporting:

- 4.10.1 Incorporate advanced analytics tools to generate insights on tutoring effectiveness and student progress.

4.11 Feedback and Rating System:

4.11.1 Implement a system for students to provide feedback on tutor performance and session quality.

4.12 Integration with Payment Gateways:

4.12.1 Enable secure payment processing for tutoring services within the system.

4.13 Gamification Elements:

4.13.1 Educational Games: Introduce subject-based educational games to engage students and reinforce their learning.

4.13.2 Tutor Contributions: Enable tutors to create questions or provide ideas for these games, tailored to specific subjects or courses.

4.13.3 Progress Tracking and Rewards: Implement a system to track students' progress in the games and award points or badges for their achievements.

4.14 Teacher & Student Sign up:

4.14.1 Allow teachers and students to sign up separately and manage their own information and courses

5. Development Model

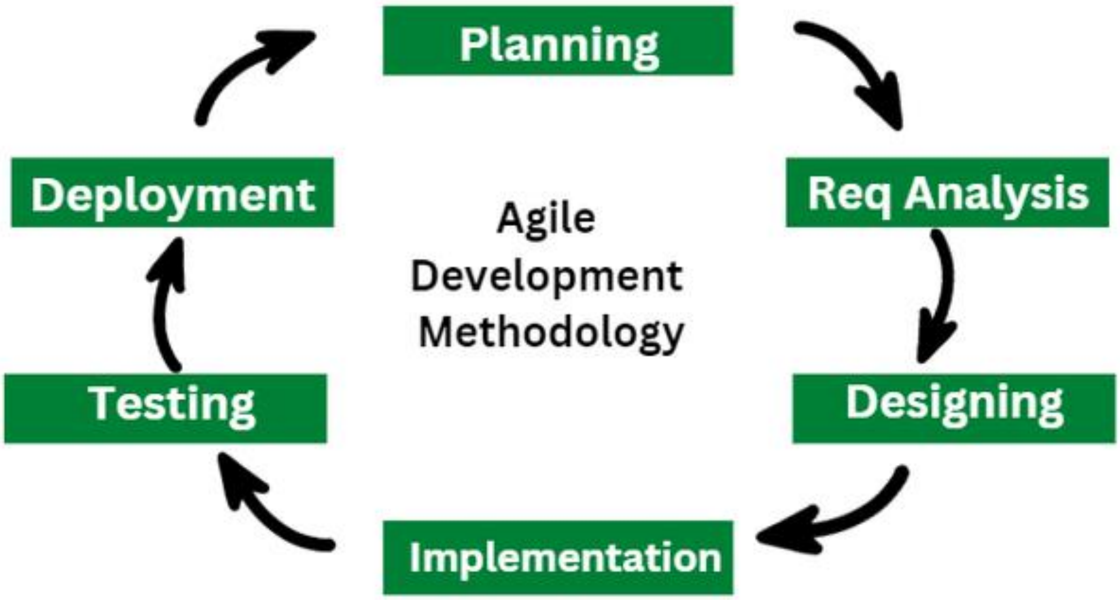

Fig 1 : Agile Development Methodology Model

For our Tuition Management System, the most suitable development model would be Agile, which is highly adaptive for projects with evolving requirements, like educational systems. For this C# desktop based application, an Agile Model [Ref: javatpoint.com/software-engineering-agile-model] remains a suitable choice due to its flexibility and iterative nature.

Here's why Agile fits in our project:

i. **Iteration and Flexibility**: C# desktop applications, especially those with complex features like student management, score tracking, and analytics, benefit from Agile's iterative approach. Each feature (e.g., user interface updates, integration with databases, or course management) can be developed and tested in sprints. Allowing continuous improvement and quick adaptation to feedback.

ii. **User Feedback**: Agile allows ongoing user feedback, ensuring that key components such as the tutor, student, and admin interfaces are continually refined based on usability testing.

iii. **Feature Prioritization**: Features like score management, reporting, and AI recommendations can be developed in stages, with basic functionality rolled out first and more advanced features added iteratively.

iv. **Collaboration and Communication**: Agile emphasizes regular communication between stakeholders, including students, tutors, and administrators. This ensures that the development process aligns with their needs and expectations, addressing issues as they arise and integrating their feedback into the development process.

6. Feasibility Analysis & Possible Risks

Given that the system is desktop based and developed in C#, the feasibility analysis changes slightly, focusing on factors such as desktop compatibility, user experience, and database integration.

1. **Technical Feasibility**:
   - i. The application will be built using C# and .NET Framework. These technologies are well-suited for desktop applications with robust support for UI design and database integration.
   - ii. **Database Integration**: SQL Server or MySQL also XAMPP will be used for managing data. Integrating with a relational database from C# via ADO.NET is technically feasible and common in desktop application development.
   - iii. **Potential Risks**: The major challenge is ensuring smooth communication between the desktop application and the database, especially when handling large datasets (e.g., student records, course information, teacher information etc). Performance issues could arise if the database is not optimized properly.

2. **Operational Feasibility**:
   - i. Desktop applications require careful attention to UI design and user experience. The system needs to be highly user-friendly. Ensuring that students, tutors, and admins can navigate the system with ease is critical.
   - ii. **Potential Risks**: A poor user interface could reduce the adoption rate of the system. Continuous feedback will be needed to ensure that the interface is user-friendly and meets the needs of all user groups.

3. **Economic Feasibility**:
   - i. **Cost of Development :** The major expenses are related to development tools software licenses (e.g., Visual Studio, database hosting) and hardware requirements

(server or cloud storage for database). If you introduce more advanced features such as AI-powered recommendations, the cost of development and maintenance may rise.

ii. **Potential Risks**: If advanced features like AI recommendations or LMS integration are introduced, the cost of development could increase due to the need for additional expertise and tools.

Possible risks include technical issues related to database integration, UI design challenges, or missing deadlines due to unexpected technical hurdles.

7. Requirements

7.1 Hardware Requirements:

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.

7.2 Software Requirements:

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.

7.3 Personnel:

i. **Developers:** C# developers experienced with ASP.NET and SQL Server.
ii. **UI/UX Designers:** To modernize the user interface and enhance user experience.
iii. **QA Testers:** To test the system for bugs and ensure functionality across different scenarios.
iv. **Project Manager:** To oversee project progress and ensure adherence to timelines and budgets.
v. **Support Staff:** For user training and ongoing support post-implementation.

7.4 Tools & Technology:

i. **Development Tools:** Microsoft Visual Studio, SQL Server Management Studio.
ii. **Project Management Tools:** JIRA or Trello for tracking progress and managing tasks.
iii. **Version Control:** GitHub or Bitbucket for source code management.
iv. **Communication Tools:** Slack or Microsoft Teams for team collaboration.

7.5 Languages:

i. **Programming Language:** C#.
ii. **Scripting Language:** ASP.NET (for web components).
iii. **Database Language:** SQL (for database management and queries).

8. Project Breakdown with time estimation (in weeks)

Here's the breakdown of the tasks with time estimates, expressed in weeks:

#	Stages/Tasks	Efforts (man-hours)	Time Estimate (Weeks)
Stage 1	**Analysis and Design**		
1.1	Requirements analysis and mockups	20	1 week
1.2	Work plan and task assignments	10	0.5 week
Stage 2	**Development**		
2.1	Database creation and testing	50	2 week
2.2	Backend programming (ASP.NET)	60	3 week
2.3	Frontend design and UI development	45	2 week
2.4	Integration with payment gateway	25	1 week
2.5	Virtual tutoring and course management features	40	2 week
Stage 3	**Testing & Quality Assurance**		
3.1	Testing and bug fixing	40	1.5 week
Stage 4	**Deployment**		
4.1	Deployment and documentation	15	0.5 week
Stage 5	**Support & Maintenance**		
5.1	Ongoing support and feedback	Continuous	0.5 week
Total		**305 (man-hours)**	**14 week** **(3.3 months)**

9. Budget

To significantly reduce costs, we can make use of free tools and basic hardware/software, avoiding premium licenses or external consultants.

Project Item	Amount	Unit Price(TK)	Total(TK)
Software License	Free (Open-source)	N/A	N/A
Server License	Use free tier cloud services (AWS/Heroku)	N/A	N/A
Database License	Free (XAMPP, MySQL)	N/A	N/A
Development Consultant	Internal team	7,000	7,000
Project Management Software	(Trello, Jira premium)	2,000	2,000
UI/UX Design Tools	Premium	2,000	2,000
Hardware (If needed)	Reuse existing hardware	Minimal	5,000
Training & Support	(Team-led)	3,000	3,000
Total			19,000

10. Development Team & Stakeholders

ID	Name	Designation	Responsibilities
22234103139	Md. Rahat Bin Israil	Team Leader	Team managing, Co-coordinating among members, UI/UX Enhancements & Gamification Features, Backend Support.
22234103123	Md. Shahidul Islam Parvez	Co-Leader	Assisting the team leader, helping coordinate tasks and decisions, Frontend Developer, UI Developer.
22234103137 22234103136	Akkas Uddin Noor Uddin Yousuf Shanto	Database Administrator	Managing the MySQL/ XAMPP database,
22234103142	Md. Saidul Islam	Payment Integration	Payment gateway, Ensure smooth processing of payments within the application
22234103239 21224103155	Jannatul Ferdous Prity Md. Jahid Hossain	QA Tester & Basic Bug Fixer	Testing the system for bugs, ensuring functionality, and providing feedback for improvements.

Stakeholders:

The following are the key stakeholders identified for the **Tuition Management System** project:

i. **Tutoring Center Administration**: The management or owners responsible for overseeing the operations of the tutoring center, including student enrollment, tutor management, and payment tracking.

ii. **Tutors**: Educators who will use the system to manage their schedules, monitor student progress, and receive payments for their services.

iii. **Students**: Learners who will interact with the system to access class schedules, study materials, and manage payments.

iv. **Technical Support Team**: Personnel responsible for maintaining, updating, and providing technical assistance for the system.

v. **System Developers**: The development team tasked with designing and implementing the system, ensuring that it meets the functional requirements of the stakeholders.

11. Limitations & Future Works

11.1 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.

11.2 Future Works:

i. **AI-Powered Recommendations**: Introduce advanced machine learning features for personalized tutor recommendations and course suggestions based on student progress.

ii. **Enhanced Reporting and Analytics**: Future updates could focus on deeper insights into student performance, tutor effectiveness, and course analytics.

iii. **Mobile App Development**: Extend the system to mobile platforms with additional features like push notifications and mobile payments.

iv. **Cloud Migration**: If the user base grows, migrate the system to a scalable cloud infrastructure to handle larger data volumes more efficiently.

12. References

[1] Preply "Online Learning Platform." Retrieve from: https://preply.com [Ref: Section 3.1].

[2] Tutor.com "Online Tutoring Services." Retrieve from: https://www.tutor.com [Ref: Section 3.2].

[3] Google Classroom "Google Classroom Workspace." Retrieve from: https://sites.google.com/view/classroom-workspace [Ref: Section 3.3].

[4] Javatpoint "Agile Model in Software Engineering." Retrieve from: https://www.javatpoint.com/software-engineering-agile-model [Ref: Section 5].