“TutorHubBD”
An AI-Powered Tuition Marketplace Platform

Software Requirements Specification

Prepared by

Student ID	Name
24141215	Syed Ar Rafi
24141216	Adiba Islam Khan
23241080	Sidratul Muntaha

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of the requirements for TutorHub DHK, a web-based professional tuition marketplace. The primary objective of this platform is to create a transparent, efficient, and reliable ecosystem that connects guardians and students with qualified and verified tutors within the Dhaka metropolitan area.

This document outlines the functional and non-functional requirements, system architecture, technology stack, and project scope. It is intended to serve as a comprehensive guide for the development team, project stakeholders, and faculty members to ensure all requirements are met and the final product aligns with the project's goals.

1.2 Scope

The scope of this project encompasses the design, development, testing, and deployment of the "TutorHub DHK" web application. The platform will serve three primary user roles:

- **Guardians/Students:** Can post tuition requirements, search for tutors, review teacher profiles, shortlist candidates, confirm hiring, and leave verified reviews.

- **Teachers:** Can create detailed professional profiles, upload verification documents, browse and apply for available tuition jobs, and manage their application status.

- **Administrators:** Will have oversight of the entire platform, including user management, verification of teacher profiles, management of job postings, revenue tracking, and dispute resolution.

This SRS will detail the business logic for a commission-based revenue model, where the platform charges a percentage of the first month's salary from the teacher upon successful hiring. It also covers the integration of a simple AI module to provide intelligent teacher recommendations.

1.3 Definitions, Acronyms, and Abbreviations

Acronym/Term	Definition
ASP.NET Core	A cross-platform, high-performance, open-source framework for building modern, cloud-based, Internet-connected applications.
EF Core	Entity Framework Core. A modern object-database mapper for .NET. It enables developers to work with a database using .NET objects.
Flask	A lightweight WSGI web application framework in Python. It is designed to make getting started quick and easy.
AI	Artificial Intelligence. The theory and development of computer systems able to perform tasks that normally require human intelligence.
API	Application Programming Interface. A set of rules and protocols for building and interacting with software applications.
DHK	Abbreviation for Dhaka.
MVC	Model-View-Controller. A software architectural pattern for implementing user interfaces on computers.
OTP	One-Time Password. A password that is valid for only one login session or transaction.
RBAC	Role-Based Access Control. A method of restricting network access based on the roles of individual users within an enterprise.
SRS	Software Requirements Specification. A description of a software system to be developed.
UI/UX	User Interface / User Experience.
MySQL	An open-source relational database management system.

1.4 References

- ***ASP.NET Core Stack Documentation:***
    - ASP.NET Core
    - Entity Framework Core
    - C# Language Guide

- ***Python AI Stack Documentation:***
    - Python
    - Flask
    - scikit-learn

- ***General Methodologies & Inspirations:***
    - IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
    - Caretutors.com: An existing platform in the Bangladeshi market, analyzed for competitive features.

1.5 Overview

This SRS document is organized into subsequent sections to provide a clear and structured overview of the project. Section 2 will provide an overall description of the product, its user classes, and operating environment. Section 3 will detail the specific functional and non-functional requirements. Further sections will outline the system architecture, technology stack, and the proposed development plan. This structured approach ensures that all aspects of the TutorHub DHK platform are thoroughly defined, providing a solid foundation for the subsequent design and development phases.

2. Overall Description

2.1 Product Perspective

The "TutorHub DHK" web application is a standalone system built on the ASP.NET Core MVC framework for the backend and a Python Flask API for AI-driven services. It integrates with external APIs where necessary (e.g., payment gateways, SMS/email gateways). The application is designed to be a secure and trusted marketplace, ensuring data privacy and a reliable user experience for all participants.

2.2 Product Features

The platform is designed with a comprehensive set of features to facilitate a seamless connection between guardians and teachers. The core functionalities are broken down into the following 15+ features:

Marketplace & Hiring Features:

1. ***Guardian Tuition Job Posting:*** Guardians can create, publish, and manage detailed tuition job postings that include requirements such as subject, grade, location, salary, and preferred teacher characteristics.

2. ***Advanced Job Search & Discovery for Teachers:*** Teachers can browse a live job board with multi-faceted filtering (by location, subject, class, salary) and a keyword search to find relevant tuition opportunities.

3. ***Teacher Application System:*** Teachers can submit applications to job postings, which are then organized for the guardian's review. Teachers can track the status (Pending, Shortlisted, Hired) of all their applications.

4. ***Applicant Shortlisting & Management:*** Guardians can review a list of applicants for their job post and shortlist a limited number of candidates (e.g., up to 5) to proceed with interviews or trial classes.

5. ***Hiring Confirmation Workflow****:* A formal system for guardians to select and confirm the final teacher, which officially marks the job as "filled" and initiates the platform's commission process.

AI & Intelligence Features:

6. ***AI-Powered Teacher Recommendation Engine:*** An intelligent feature that allows guardians to use a natural language prompt (e.g., "I need a patient female math teacher for Class 5 in Mirpur") to receive a ranked list of the most suitable teacher profiles.

7. ***Profile-Based Job Alert System:*** An automated system that sends email or in-app notifications to teachers when new tuition jobs are posted that match their specific profile criteria (subjects, location, grade level).

Trust & Quality Features:

8. ***Teacher Profile Verification System:*** An admin-driven workflow to review and approve teachers' uploaded academic credentials and identity documents, awarding a "Verified" badge on their profile to build guardian trust.

9. ***Verified Review and Rating System:*** Guardians can submit a star rating and a detailed review for a teacher only after a hiring has been officially confirmed and completed through the platform, preventing fake reviews.

10. ***Teacher Profile Analytics:*** Teachers can view analytics on their own profile, such as the number of profile views, search appearances, and application success rate, helping them improve their profiles.

Financial & Administrative Features:

11. ***Automated Commission Calculation & Invoicing:*** The system automatically calculates the 40% first-month commission upon a successful hiring and generates an invoice for the teacher.

12. ***Integrated Payment Processing:*** Secure integration with local Bangladeshi payment gateways (e.g., bKash, Nagad) to allow teachers to pay their commission fees directly on the platform.

13. ***Comprehensive Admin Dashboard & Analytics:*** A master control panel for administrators to manage users, verify documents, moderate content, resolve disputes, and view key platform metrics like revenue, user growth, and popular subjects.

14. ***Dispute Resolution Module:*** A basic ticketing system for guardians or teachers to report issues with a hiring, which an administrator can then mediate.

User Experience & Communication Features:

15. ***Multi-channel Notification System:*** An automated system that sends critical updates (e.g., application shortlisted, payment due) to users via in-app alerts, email, and SMS.

16. ***Detailed User Dashboards:*** Separate, tailored dashboards for Guardians, Teachers, and Admins, providing each user with the specific information and tools they need to manage their activities on the platform.

2.3 User Classes and Characteristics

- **Guardians/Students:** Individuals seeking qualified tutors. Typically non-technical, requiring a simple, user-friendly interface to post jobs and hire teachers.

- **Teachers:** University students or professional educators seeking tuition jobs. They require a robust profile builder and an efficient way to find and apply for opportunities.

- **Administrators:** System overseers responsible for user management, teacher verification, financial tracking, and overall platform maintenance. They require full access to system settings and analytics.

2.4 Operating Environment

- ***Web Application:*** Accessible on modern web browsers (Chrome, Firefox, Edge, Safari).

- ***Mobile Compatibility:*** Responsive design for seamless use on smartphones and tablets.

- ***Server-Side:*** ASP.NET Core 8 and Python Flask running in a cloud environment (e.g., Microsoft Azure).

- ***Database:*** MySQL, preferably a cloud-managed instance like Azure SQL Database for scalability.

- ***Hosting:*** Deployment will be on a major cloud platform like Microsoft Azure.

2.5 Constraints

- ***Technology Stack:*** The project must be developed using ASP.NET Core MVC, Python Flask, and MySQL.

- ***Security:*** Must implement encryption for data at rest and in transit (HTTPS/SSL) and secure handling of user documents.

- ***Local Payment Gateways:*** Integration is limited to Bangladeshi payment providers like bKash and Nagad, which dictates the payment workflow.

- ***Timeline:*** The project is to be delivered within the academic semester. (10 Weeks)

- ***Team Size:*** The project will be developed by a team of three members. This constrains the total amount of work that can be accomplished within the timeline.

2.6 Assumptions and Dependencies

- Users are assumed to have a stable internet connection.

- The reliability of third-party services (payment, SMS, email) is crucial for platform functionality.

- Users are assumed to provide truthful information, with the verification process serving as a trust-building measure.

- The AI recommendation is an assistive tool; the final hiring decision is the guardian's responsibility.

.

3. System Requirements

3.1 Functional Requirements

3.1.1 Authentication & Authorization

- OTP Authentication:
    - FR-1: The system shall generate and send a time-bound One-Time Password (OTP) to the user's mobile number for registration and password resets.
    - FR-2: The system shall invalidate OTPs after a configurable time period (e.g., 10 minutes) or after a single successful use.
- Role-Based Access Control (RBAC):
    - FR-3: The system shall provide different dashboards and functionalities for Guardian, Teacher, and Administrator roles.
    - FR-4: The system shall restrict access to features based on the user's role (e.g., a teacher cannot post a tuition job; a guardian cannot apply for one).

3.1.2 Marketplace and Hiring Workflow

- Tuition Job Management:
    - FR-5: Guardians shall be able to create, publish, edit, and close tuition job postings with detailed requirements.
    - FR-6: The system shall display all "Open" tuition jobs on a central job board for teachers.
    - FR-7: Teachers shall be able to search for jobs using filters (location, subject, grade) and apply to them.

- Hiring Process:
    - FR-8: Guardians shall be able to view a list of all teacher applicants for a specific job.
    - FR-9: The system shall allow guardians to shortlist a limited number of applicants (e.g., 5) for further communication.
    - FR-10: The system shall provide a mechanism for the guardian to confirm the final hiring of one teacher, which changes the job status to "Filled".

3.1.3 User Profiles and Trust System

- Teacher Profile Management:
    - FR-11: Teachers shall be able to create and manage a detailed professional profile, including educational qualifications, experience, preferred subjects, and locations.
    - FR-12: The system shall allow teachers to upload digital copies of their academic certificates and identity documents for verification.

- Verification and Ratings:
    - FR-13: Administrators shall have an interface to review uploaded documents and mark a teacher's profile as "Verified".
    - FR-14: Guardians shall be able to submit a star rating and a written review for a teacher only after a hiring is confirmed and completed through the platform.
    - FR-15: The system shall calculate and display an average rating on each teacher's public profile.

3.1.4 AI, Payments, and Administrative Features

- AI-Powered Recommendations:
    - FR-16: The system shall provide an interface for guardians to input a natural language prompt describing their teacher requirements.
    - FR-17: The system shall send this prompt to a dedicated Python Flask API. This service will use Natural Language Processing (NLP) techniques to extract key entities (e.g., subject, location, grade) and then use a scoring algorithm to return a ranked list of the most relevant teacher profiles.

- Commission & Payments:
    - FR-18: Upon hiring confirmation, the system shall automatically calculate the platform's commission (e.g., 40% of the first month's salary) and generate an invoice for the teacher.
    - FR-19: The system shall integrate with at least one major Bangladeshi payment gateway (e.g., bKash, Nagad) to facilitate the teacher's payment of the commission.

- Notifications & Administration:
    - FR-20: The system shall send automated notifications (email, SMS, in-app) to users for key events like new applications, shortlisting, and payment reminders.
    - FR-21: Administrators shall have access to a dashboard to manage users, moderate content, and view financial reports.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- NFR-1: Core pages (Homepage, Job Board, Teacher Profile) shall have a server response time of under 500ms and a client-side load time (Largest Contentful Paint) of under 2.5 seconds on a standard broadband connection.

- NFR-2: The database connection pool shall be configured to handle up to 100 concurrent connections.

3.2.2 Security Requirements

- NFR-3: All communication between the client and server shall occur over HTTPS (SSL/TLS).

- NFR-4: All user passwords must be securely hashed using a modern algorithm (e.g., PBKDF2) and never stored in plain text.

- NFR-5: The system must implement protections against common web vulnerabilities such as SQL Injection and Cross-Site Scripting (XSS).

3.2.3 Reliability & Availability

- NFR-6: The system shall aim for 99.5% uptime, excluding scheduled maintenance windows.

- NFR-7: Regular database backups shall be performed (e.g., daily) to prevent data loss in case of system failure.

3.2.4 Maintainability

- NFR-8: The codebase shall follow the standard ASP.NET Core conventions and the Repository design pattern to ensure it is easy to understand and modify in the future.

- NFR-9: Documentation for the REST API connecting the web app and the Python AI service shall be maintained and kept current.

3.2.5 Scalability

- NFR-10: The system architecture shall allow for the independent scaling of the web application and the Python AI service to handle increased load.

3.3 External Interface Requirements

3.3.1 User Interfaces

- UI-1: A responsive web interface built with HTML5, CSS3, and JavaScript that adapts cleanly to desktop, tablet, and mobile browsers.

- UI-2: Intuitive navigation and separate, purpose-built dashboards for each of the three user roles (Guardian, Teacher, Admin).

3.3.2 Hardware Interfaces

- HI-1: None. The system is a web application and does not directly interface with any specialized hardware. It will run on standard server infrastructure in a cloud environment.

3.3.3 Software Interfaces

- SI-1: Payment Gateway API: The system will interface with a local Bangladeshi payment gateway (e.g., bKash, Nagad) to process commission payments.

- SI-2: SMS Gateway API: The system will use an SMS provider's API (e.g., Twilio, or a local provider like SSL Wireless) to send OTPs and notifications.
- SI-3: Email Service API: The system will interface with an email service provider (e.g., SendGrid) for sending transactional emails.

3.3.4 Communication Interfaces

- CI-1: The primary communication protocol will be HTTPS.

- CI-2: A RESTful API will be used for internal communication between the main ASP.NET Core application and the Python AI microservice. Data will be exchanged in JSON format.

- CI-3: SignalR will be used to facilitate real-time, low-latency communication for in-app notifications.

4. Technology Stack & Architectural Overview

4.1 Technology Stack

The TutorHubBD platform utilizes a hybrid technology stack to leverage the robustness of the .NET ecosystem for core application logic and the flexibility of Python for AI/ML capabilities.

- **ASP.NET Core (v8.0):** The primary framework for the web application, handling the MVC (Model-View-Controller) structure, user authentication (Identity), and business logic.
- **MySQL:** Relational database management system for storing structured data such as User Profiles, Tuition Offers, Booking Requests, and Transactions.
- **Entity Framework Core:** The Object-Relational Mapper (ORM) used to interact with the database using .NET objects, ensuring type safety and faster development.
- **Python (Flask):** A lightweight microservice framework used specifically to host the AI Teacher Recommendation Engine.
- **HTML5, CSS3 (Bootstrap), & JavaScript:** The frontend technologies used to build the responsive user interface (Razor Views).
- **Git & GitHub:** Version control system used for collaborative development and source code management.

4.2 High-Level Architecture

The system follows a modular Monolithic architecture with a decoupled AI microservice.

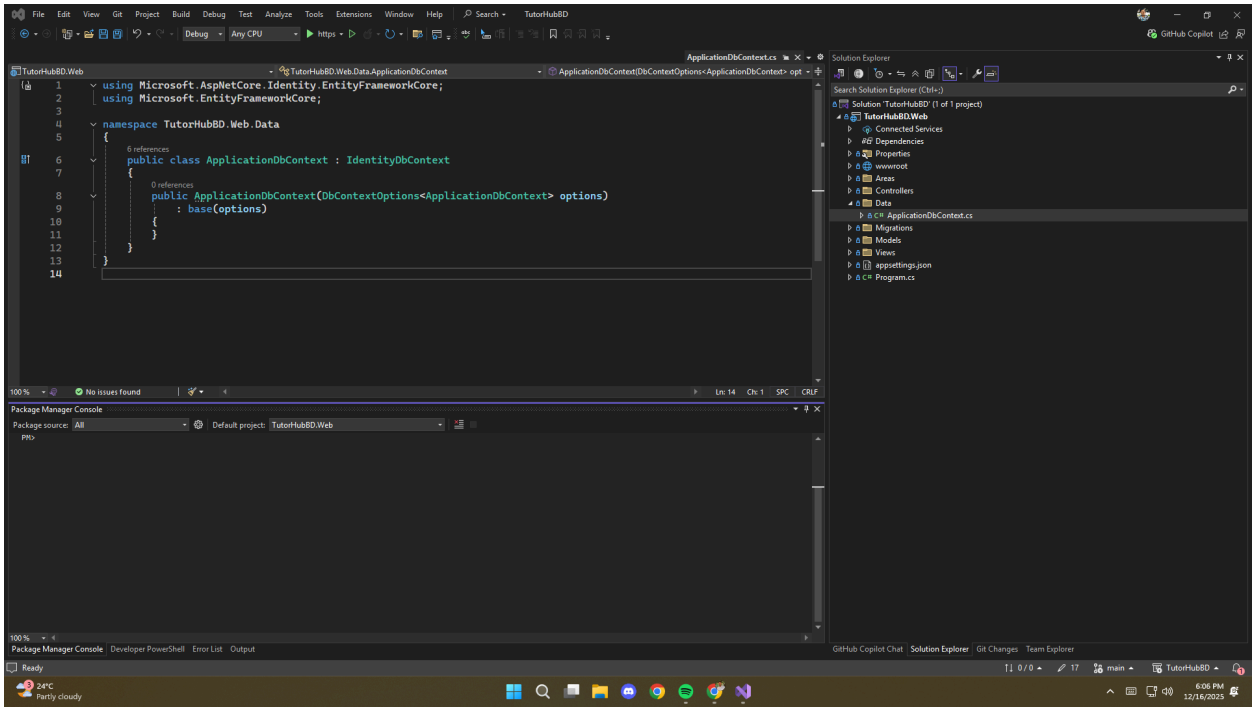
- **Presentation Layer (ASP.NET Core MVC):** Handles user interactions, form submissions, and dynamic UI rendering via Razor Views.
- **Application Layer:** Manages core business logic and facilitates communication with the Python AI API.
- **AI Microservice (Python Flask):** A standalone REST API that executes NLP algorithms for teacher recommendations.
- **Data Layer (SQL Database):** Centralized repository for persistent data, accessed primarily via Entity Framework Core.

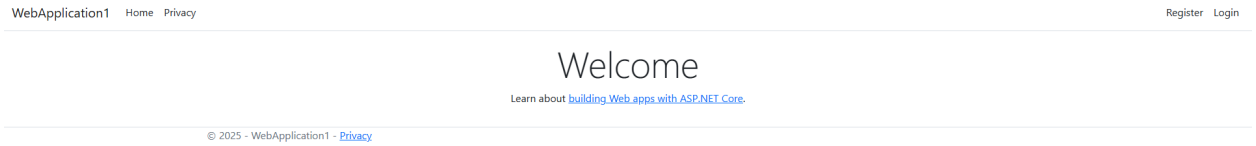
6. Conclusion

This SRS provides a detailed outline for the design and development of "TutorHubBD"—an innovative AI-powered tuition marketplace built using ASP.NET Core and Python Flask. It ensures a trust-based ecosystem by integrating core functionalities—job posting, secure hiring, tutor verification, and AI-driven teacher matching—while maintaining strict data privacy and security standards. By following the outlined Agile sprints, the project aims to deliver a robust, scalable, and efficient solution that creates genuine value for both students and educators within the designated timeline.

Sprint-1

In Sprint 1, we successfully established the foundational architecture for TutorHubBD. This involved initializing the solution with a dual-project structure (ASP.NET Core for the main web app and Python Flask for the AI service), implementing the core database schema with Entity Framework, and physically creating the SQL database with user identity support.





Additionally, we configured the frontend branding, set up the project repository with proper version control, and finalized the essential SRS and class diagram documentation to guide future development.