Software Requirements Specification

for

Fluentify

Version 1.0 approved

Prepared by Pawan , Sharv , Bishesh and Vedant

InovaTech

April 5,2024

Table of Contents

Revision History

Name	Date	Release Description	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and nonfunctional requirements of the interactive online language learning platform. It serves as a guide for the development team, stakeholders, and users to understand the features, capabilities, and constraints of the software.

1.2 Document Conventions

Formatting: This document follows the IEEE standard format for Software Requirements Specifications.
Terminology: Technical terms and acronyms are defined in the Glossary section (Section 8) of this document.
References: All external sources, standards, and documents referenced are listed in the References section (Section 1.5).

1.3 Intended Audience and Reading Suggestions

Development Team: To understand the requirements for implementing the software.
Project Stakeholders: To review and approve the proposed features and functionalities.
Quality Assurance Team: To create test cases based on the specified requirements.
Users: To gain insight into the capabilities and usage of the software.

1.4 Product Scope

The interactive online language learning platform aims to facilitate personalized language learning experiences by connecting learners with qualified tutors. The scope of the product includes but is not limited to:

- User registration and authentication for tutors and students.
- Availability management for tutors to set teaching slots.
- Booking system for students to schedule lessons with tutors.
- Flashcard creation and review functionality for language practice.
- One-to-one video call feature for conducting online lessons.
- Integration of Stripe API for secure payment processing.

1.5 References

The following references have been used in the preparation of this Software Requirements Specification (SRS) document:

- IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998)
- AgoraRTC API documentation for video call integration
- Stripe API documentation for payment integration

2. Overall Description

2.1 Product Perspective

The interactive online language learning platform is designed to operate within a larger educational ecosystem, providing a specialized service for language learners and tutors. It interfaces with external systems for payment processing (Stripe API) and real-time communication (AgoraRTC API) while maintaining its unique features and functionality.

2.2 Product Functions

The interactive online language learning platform offers a range of core functions to facilitate effective language learning experiences. Users, both tutors, and students, begin by registering and authenticating their accounts, ensuring a secure and personalized learning environment. Tutors have the flexibility to manage their availability by setting teaching slots, allowing students to schedule lessons at their convenience. Furthermore, students can utilize the booking system to browse tutors and book lessons based on their preferences and the availability of tutors. To aid in language practice, the platform offers comprehensive flashcard functionality, enabling users to create, edit, and review flashcards for vocabulary retention and comprehension. Additionally, the platform facilitates one-to-one video calls, enabling interactive and personalized lessons between tutors and students. Integration with the Stripe API ensures secure payment processing, simplifying transactions for booking lessons or purchasing additional learning materials. Together, these core functions combine to create a seamless and enriching language learning experience for users of the platform.

2.3 User Classes and Characteristics

The platform caters to two primary user classes:

1. Tutors:
    - Have expertise in specific languages.
    - Set their availability for teaching slots.
    - Can manage their pricing and payment preferences.

2. Students:
    - Seek language learning opportunities.
    - Schedule lessons with available tutors.
    - Create and review flashcards for language practice.

2.4 Operating Environment

The interactive online language learning platform operates in a web-based environment, accessible through modern web browsers such as Google Chrome, Mozilla Firefox, and Safari. It requires a stable internet connection for video calls and data synchronization. The platform is compatible with desktop and mobile devices for flexible usage.

2.5 Design and Implementation Constraints

The design and implementation of the platform are subject to the following constraints:

- User interface design must be intuitive and accessible to users of all levels.
- Integration with AgoraRTC API for real-time video communication.
- Implementation of Stripe API for secure payment processing.
- Compliance with data protection regulations (e.g., GDPR, CCPA).

2.6 User Documentation

User documentation will be provided to guide users on using the platform effectively. It will include:

- User guides for tutors and students on registration, scheduling, and using platform features.
- Technical documentation for developers on system architecture, APIs, and deployment procedures.

2.7 Assumptions and Dependencies

The development and operation of the interactive online language learning platform are based on the following assumptions and dependencies:

- Availability of stable internet connections for seamless video calls.
- Proper functioning of third-party APIs (AgoraRTC, Stripe) for video communication and payment processing.
- Compliance with legal and regulatory requirements related to online education and payment processing.
- Availability of sufficient server resources and infrastructure for platform scalability.

3. System Features

3.1 User Registration

The user registration feature allows individuals to create accounts on the platform. Users provide necessary information such as name, email address, password, and language preferences during registration. The system verifies the provided information and creates a unique user profile for each registered user.

3.2 Tutor Availability

Tutors can set their availability for teaching sessions using the tutor availability feature. They specify their preferred time slots, days of the week, and duration of each session. Students can view the available slots and schedule lessons accordingly.

3.3 Student Booking

The student booking feature enables learners to book lessons with available tutors based on their availability. Students can browse through the list of tutors, view their profiles, check available slots, and book lessons according to their preferred schedule.

3.4 Flashcard Creation

Users, specifically students, can create and manage flashcards for language practice using the flashcard creation feature. They can add new flashcards, categorize them based on topics or difficulty levels, and review them for learning and revision purposes.

3.5 Video Call Feature

The video call feature facilitates one-to-one online lessons between tutors and students. It uses real-time communication technology to establish video calls with high-quality audio and video. During the video call, users can interact, share screens, and collaborate on learning materials.

3.6 Payment Integration

Payment integration allows for secure and seamless transactions between students and tutors. The system integrates with a payment gateway (e.g., Stripe API) to process payments for lessons. Students can make payments for booked sessions, and tutors receive payments for their services through the platform.

4. External Interface Requirements

4.1 User Interface

The user interfaces of the interactive online language learning platform are designed to be intuitive, user-friendly, and accessible across devices. The following user interfaces are included:

- **Registration Interface**: Allows users to register as tutors or students by providing necessary information such as name, email, password, and language preferences.
- **Dashboard Interface**: Provides a personalized dashboard for each user role (tutor or student), displaying relevant information such as upcoming lessons, available slots, payment history, and flashcard management options.
- **Tutor Profile Interface**: Displays detailed profiles of tutors, including their bio, expertise, teaching style, availability, pricing, and student reviews.
- **Booking Interface**: Enables students to browse tutors, view available slots, select preferred lesson times, and book sessions based on tutor availability.
- **Flashcard Creation Interface**: Allows students to create, edit, categorize, and review flashcards for language practice.
- **Video Call Interface**: Facilitates one-to-one video calls between tutors and students during scheduled lessons, providing features for video, audio, screen sharing.

4.2 Software Interface

The platform integrates with the following software interfaces:

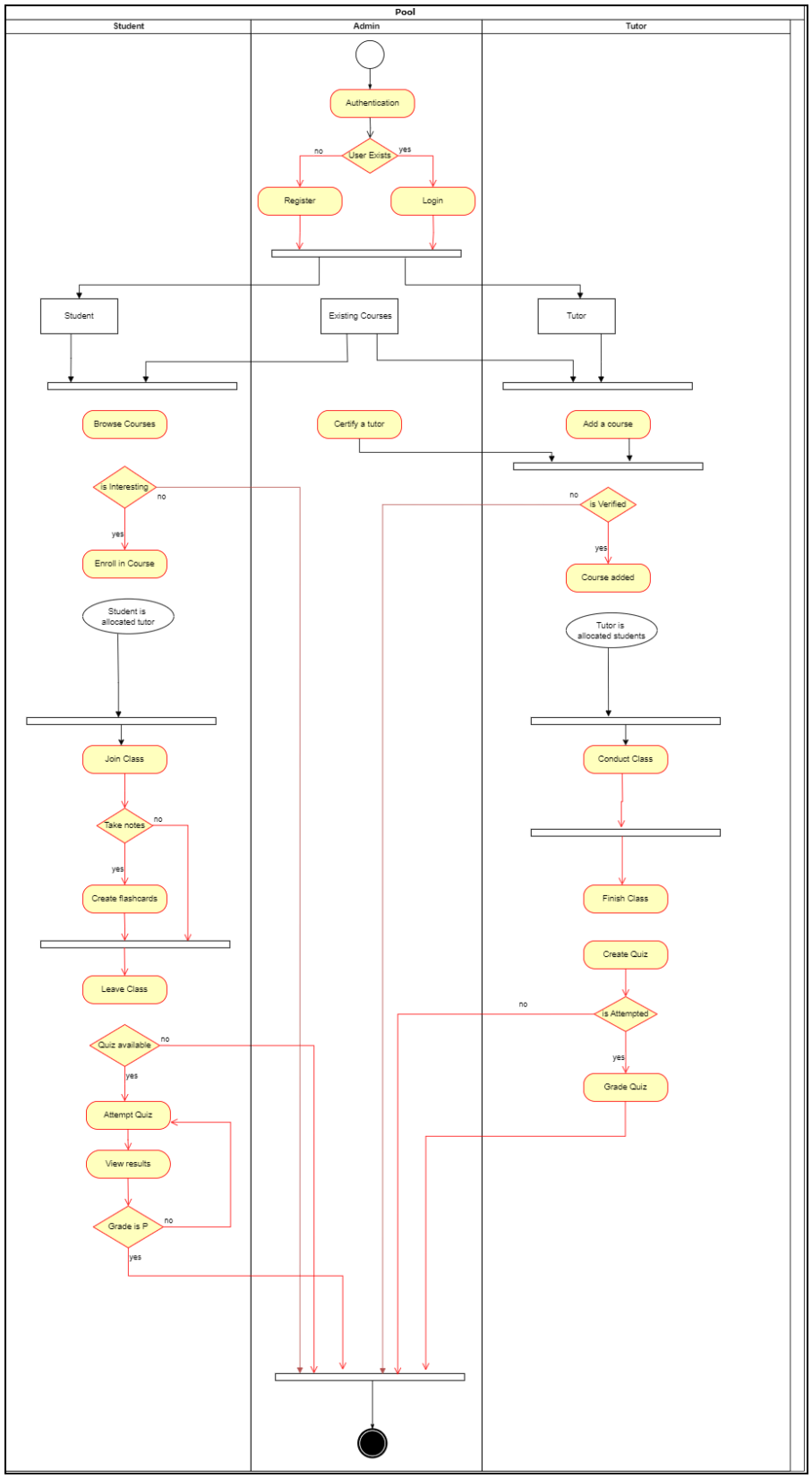
- **AgoraRTC API**: Used for real-time communication features, including video calls, audio calls, screen sharing, and interactive collaboration tools.
- **Stripe API**: Integrated for secure payment processing, allowing students to make payments for lessons and tutors to receive payments for their services.

4.3 Communication Interface

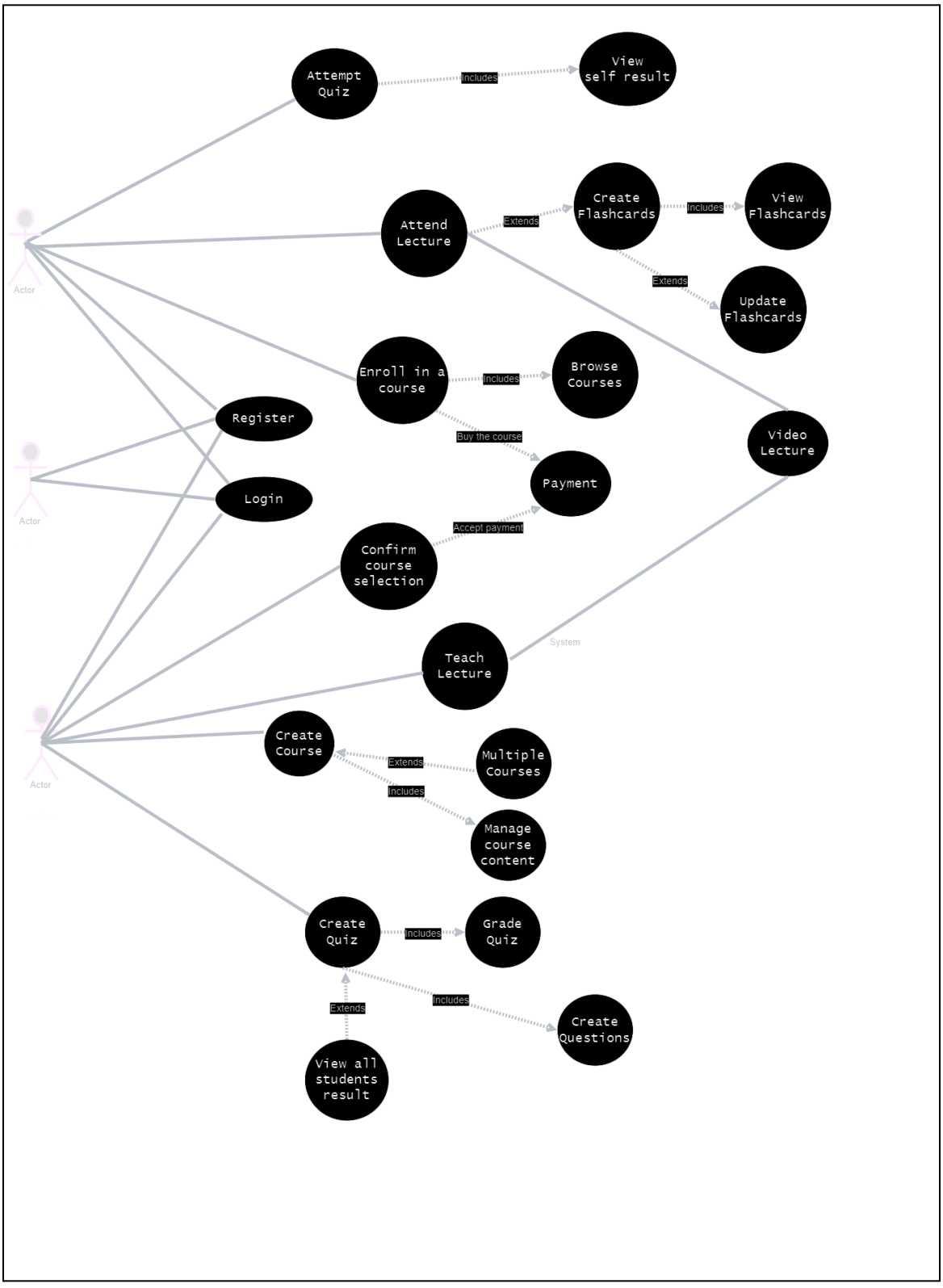
Communication interfaces within the platform include:

- **Email Notifications**: Sends automated notifications to users for account verification, lesson reminders, booking confirmations, payment receipts, and other relevant updates.
- **In-App Messaging**: Provides an internal messaging system for tutors and students to communicate, discuss lesson details, share learning materials, and ask questions outside of scheduled lessons.

ACTIVITY DIAGRAM


Pool
Student
Admin
Tutor
Authentication
User Exists
no
yes
Register
Login
Student
Existing Courses
Tutor
Browse Courses
Certify a tutor
Add a course
is Interesting
no
yes
Enroll in Course
Student is allocated tutor
Join Class
Take notes
no
yes
Create flashcards
Leave Class
Quiz available
no
yes
Attempt Quiz
View results
Grade is P
no
yes
is Verified
no
yes
Course added
Tutor is allocated students
Conduct Class
Finish Class
Create Quiz
is Attempted
no
yes
Grade Quiz

USE CASE DIAGRAM



5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system shall meet the following performance requirements:

- **Response Time**:
  - The system shall respond to user interactions within 1 second for 90% of requests.
- **Scalability**:
  - The system should be able to handle a concurrent user load of at least 1000 users without significant degradation in performance.
- **Resource Utilization**:
  - The system should optimize resource utilization to ensure efficient use of CPU, memory, and network bandwidth.

5.2 Security Requirements

The system shall adhere to the following security requirements:

- **Data Encryption**:
  - All sensitive data transmitted over the network shall be encrypted using HTTPS/TLS.
- **Data Protection**:
  - User data shall be stored securely with measures such as encryption at rest and regular backups.
- **Authentication and Authorization**:
  - The system shall support multi-factor authentication (MFA) for user authentication and fine-grained authorization policies.

5.3 Software Quality Attributes

5.3.1 Usability

The system shall exhibit the following usability characteristics:

- **Intuitive User Interface**:
  - The user interface shall be designed to be intuitive, with clear navigation and user-friendly controls.
- **Accessibility**:
  - The system shall adhere to accessibility standards (e.g., WCAG) to ensure usability for users with disabilities.
- **Multilingual Support**:

    - The system shall support multiple languages to accommodate users from diverse linguistic backgrounds.
- **Help and Documentation**:
    - The system shall provide contextual help and comprehensive documentation to assist users in using the platform effectively.

5.3.2 Robustness

The system shall demonstrate robustness in the following aspects:

- **Error Handling**:
    - The system shall handle errors gracefully, providing meaningful error messages and recovery options.
- **Fault Tolerance**:
    - The system shall be designed with fault-tolerant mechanisms to minimize downtime and disruptions in service.
- **Data Integrity**:
    - The system shall ensure data integrity through validation checks, data redundancy, and error correction mechanisms.

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: User Flow Models