



Assignment
On
Java GUI Development

Topic: SRS on Cooking Course Management System

Submitted to:
Professor Fuad Ahmed
Head of the
Department of Software Engineering

Submitted by:
Uzma Yakub (222-134-001)
Tahmina Tanni (241-134-004)
Abida Sultana (241- 134- 029)
SWE 6th Batch
Department of Software Engineering

Software Requirements Specification (SRS) for Flavorista (Cooking Course Management System)

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
 2. Overall Description
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 User Classes and Characteristics
 - 2.4 Operating Environment
 - 2.5 Design and Implementation Constraints
 - 2.6 Assumptions and Dependencies
 3. Specific Requirements
 - 3.1 External Interfaces
 - 3.2 Functional Requirements
 - 3.3 Non-Functional Requirements
 - 3.4 Future Advancements: AI Integration
 - 3.5 Logical Database Requirements
 4. Supporting Information
 - 4.1 Use Case Diagrams
 - 4.2 Data Flow Diagrams
 - 4.3 Entity-Relationship Model
- Appendices
- Appendix A: Glossary
- Appendix B: Sample Certificate Format

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for an online learning platform specialized in cooking courses. This cooking course management system, inspired by 10 Minute School, enables users to access structured video-based cooking lessons from home. It includes features for user authentication, course management, video streaming, progress tracking, notifications, and certification. The document serves as a reference for developers, testers, and stakeholders to ensure the system meets user needs and supports future AI enhancements for personalization and engagement.

1.2 Scope

The system provides an end-to-end solution for online cooking education, covering user registration, course enrollment (free and paid), secure payments, on-demand video streaming, progress monitoring, certification issuance, and email notifications. It supports multiple devices and platforms, with scalability for up to 10,000 concurrent users. Future scope includes AI-driven features like personalized recommendations, interactive assistants, adaptive learning, and content generation.

The system does not include physical cooking tools, live instructor interactions (beyond pre-recorded videos), or in-person events. It focuses on video-based, self-paced learning with thematic courses (e.g., Baking Course with sequential lessons).

1.3 Definitions, Acronyms, and Abbreviations

- **API:** Application Programming Interface
- **CDN:** Content Delivery Network
- **GDPR:** General Data Protection Regulation
- **NLP:** Natural Language Processing
- **PWA:** Progressive Web App
- **SRS:** Software Requirements Specification
- **UI/UX:** User Interface/User Experience
- **Course:** A thematic collection of sequential video lessons (e.g., Baking Course).
- **Lesson:** An individual video unit within a course (e.g., Basics of Dough).
- **Progress:** Percentage of lessons completed in a course.
- **Certification:** Digital PDF document issued upon 100% course completion.

1.4 References

- IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.
- 10 Minute School platform documentation (for inspiration on structured video lessons).
- Stripe/PayPal API documentation for payment integration.
- TensorFlow and GPT model guidelines for AI features.
- GDPR compliance guidelines.

1.5 Overview

This SRS is organized as follows: Section 2 provides an overall description of the product. Section 3 details specific requirements, including functional, non-functional, and future AI integrations. Section 4 includes supporting diagrams and models for clarity. Appendices provide additional details.

2. Overall Description

2.1 Product Perspective

This platform builds on the model of 10 Minute School by specializing in cooking education, addressing the growing demand for accessible, home-based skill development. It differentiates through thematic, sequential video courses tailored to cooking, with integrated payments and certifications. It aims to democratize cooking education, making it available to hobbyists and professionals alike, while planning for AI to enhance personalization in future iterations.

2.2 Product Functions

The system shall:

- Authenticate and register users.
- Manage course catalogs, enrollments, and payments.
- Stream videos with playback controls and serial progression.
- Track and display user progress.
- Issue downloadable certificates upon completion.
- Send email notifications for key events.
- Support future AI for recommendations, chat assistance, analytics, and content generation.

2.3 User Classes and Characteristics

- **Learners (Primary Users):** Individuals interested in cooking, aged 18+, with basic digital literacy. They access courses via mobile or web, expecting intuitive navigation and high-quality videos.
- **Instructors/Administrators:** Content creators who upload courses (assumed to have technical support access). They require tools for course management.
- **Guests:** Unregistered users who browse catalogs but cannot enroll or access full content.

2.4 Operating Environment

- **Hardware:** Smartphones, tablets, laptops, desktops.
- **Software:** Android (v10+), iOS (v14+), Windows (v10+), macOS (v11+). Web version compatible with Chrome, Safari, Firefox.
- **Network:** Internet connection for streaming; offline support for downloaded content.
- **Backend:** Cloud services like AWS or Firebase for scalability.

2.5 Design and Implementation Constraints

- Use Kotlin for multiplatform development (Android, iOS via Kotlin Multiplatform, web via Kotlin/JS or PWA).
- Integrate third-party services: Stripe/PayPal for payments, CDN for video streaming.
- Comply with accessibility standards (e.g., WCAG 2.1).
- Initial language: English, with extensibility for others.

2.6 Assumptions and Dependencies

- Users have reliable internet for initial access; offline features depend on device storage.
- Third-party APIs (e.g., payment gateways, email services) are available and reliable.
- Video content is pre-produced and uploaded by administrators.
- AI features depend on external libraries like TensorFlow or cloud AI services.

3. Specific Requirements

3.1 External Interfaces

- **User Interfaces:** Responsive UI with navigation menus, search bars, video players, and dashboards. Supports touch and mouse inputs.
- **Hardware Interfaces:** None (software-only).
- **Software Interfaces:** APIs for payment (Stripe/PayPal), email (e.g., SendGrid), video streaming (CDN like CloudFront), and future AI services (e.g., Google Cloud AI).
- **Communication Interfaces:** HTTPS for secure data transfer; SMTP for emails.

3.2 Functional Requirements

Functional requirements are grouped by feature.

3.2.1 User Authentication and Registration

- The system shall allow users to sign up using email/password or social logins (Google, Apple).
- The system shall provide secure sign-in with password reset via email
- Upon sign-up, the system shall send an automated email confirmation.

3.2.2 Course Management

- The system shall display a catalog of courses with details: name, description, duration, instructor, preview videos.
- The system shall allow enrolled users to register in free or paid courses.
- For paid courses, the system shall integrate a secure payment gateway and send email confirmation upon success.

3.2.3 Video Streaming and Learning

- The system shall provide on-demand streaming of high-quality videos with controls (play/pause, seek, speed).
- Lessons shall be organized serially, with automatic progression to the next upon completion.
- Videos shall stream without full download, using adaptive bitrate.

3.2.4 Progress Tracking

- The system shall track and display progress (e.g., percentage completed, lessons viewed).
- The system shall allow resuming from the last viewed point.

3.2.5 Certification

- Upon 100% completion, the system shall automatically issue a downloadable PDF certificate.
- Certificates shall include user name, course details, completion date, and be shareable.

3.2.6 Notifications

- The system shall send emails for sign-up, payment, completion, and optional reminders for unfinished lessons.

3.3 Non-Functional Requirements

3.3.1 Platform Compatibility and Responsiveness

- The app shall be responsive across devices: Android/iOS mobiles/tablets, Windows/macOS desktops.
- Web version shall support major browsers.
- Videos shall load with <5s buffering; support offline downloads.

3.3.2 Security

- Data shall be encrypted (e.g., AES for user info, SSL for payments).
- Comply with GDPR; protect against SQL injection, XSS.

3.3.3 Usability

- Intuitive UI with search, subtitles, screen reader support.
- Localization for English initially, extensible to others.

3.3.4 Scalability and Performance

- Handle 10,000 concurrent users via cloud backend (Firebase/AWS).
- Use CDN for video delivery.

3.3.5 Performance Requirements

- Response time: <2s for page loads; <1s for API calls.
- Availability: 99.9% uptime.

3.4 Future Advancements: AI Integration

AI features shall be backend-integrated, with Kotlin handling frontend API calls.

3.4.1 Personalized Recommendations

- Use ML (e.g., collaborative filtering via TensorFlow) to analyze history/preferences and recommend courses (e.g., "If you liked baking, try Pastry"). Delivered via in-app notifications.

3.4.2 Interactive Cooking Assistant

- AI chatbot (GPT-like NLP) for real-time tips during playback, answering queries (e.g., egg substitutes) or generating recipes based on inputs (e.g., dietary restrictions).

3.4.3 Progress Analytics and Adaptive Learning

- Monitor performance (e.g., via quizzes) using reinforcement learning to adapt difficulty, suggesting reviews if stalled.

3.4.4 Content Generation

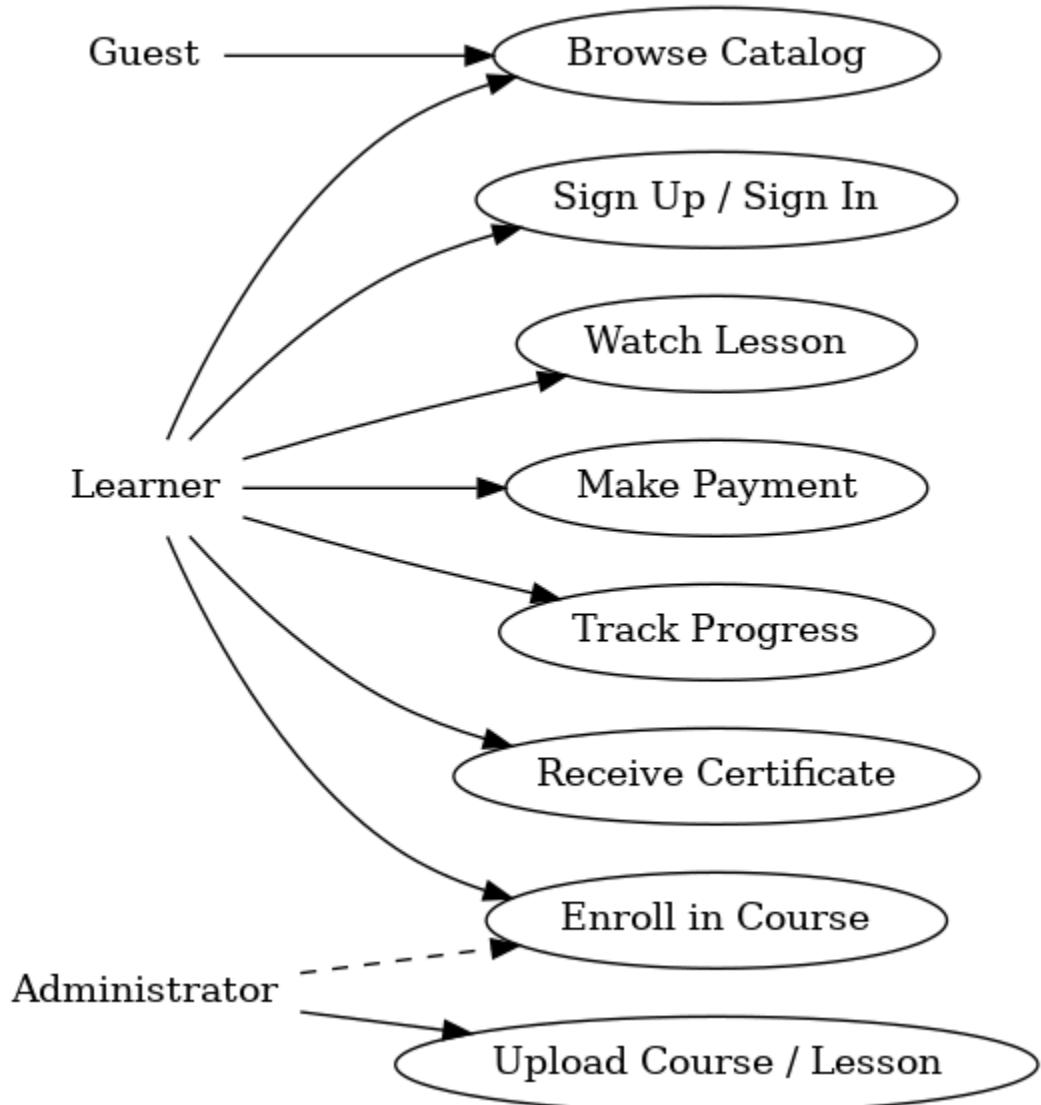
- Auto-generate subtitles, summaries, quizzes using speech-to-text and NLP.
- Ethical considerations: Mitigate bias in recommendations; ensure data privacy.
-

3.5 Logical Database Requirements

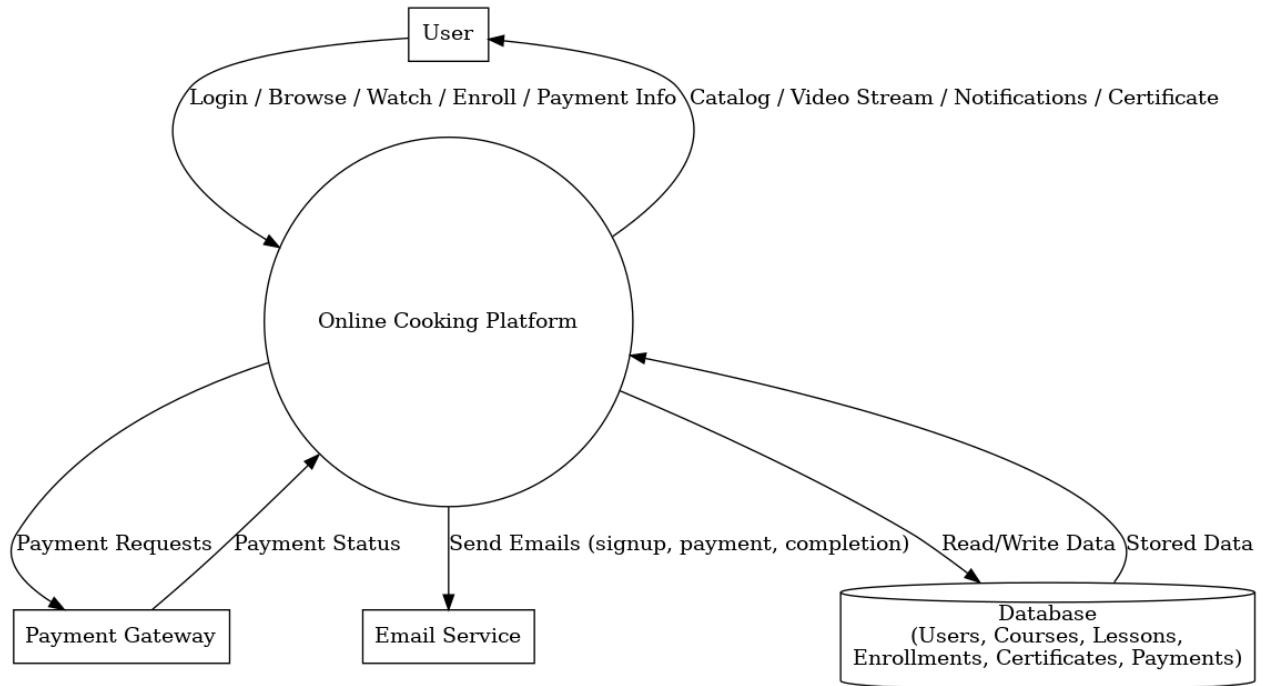
- The system shall use a relational database (e.g., PostgreSQL) with tables for Users (ID, email, password hash), Courses (ID, name, description), Enrollments (userID, courseID, progress), Lessons (ID, courseID, videoURL), Certificates (ID, userID, courseID, issueDate).

4. Supporting Information

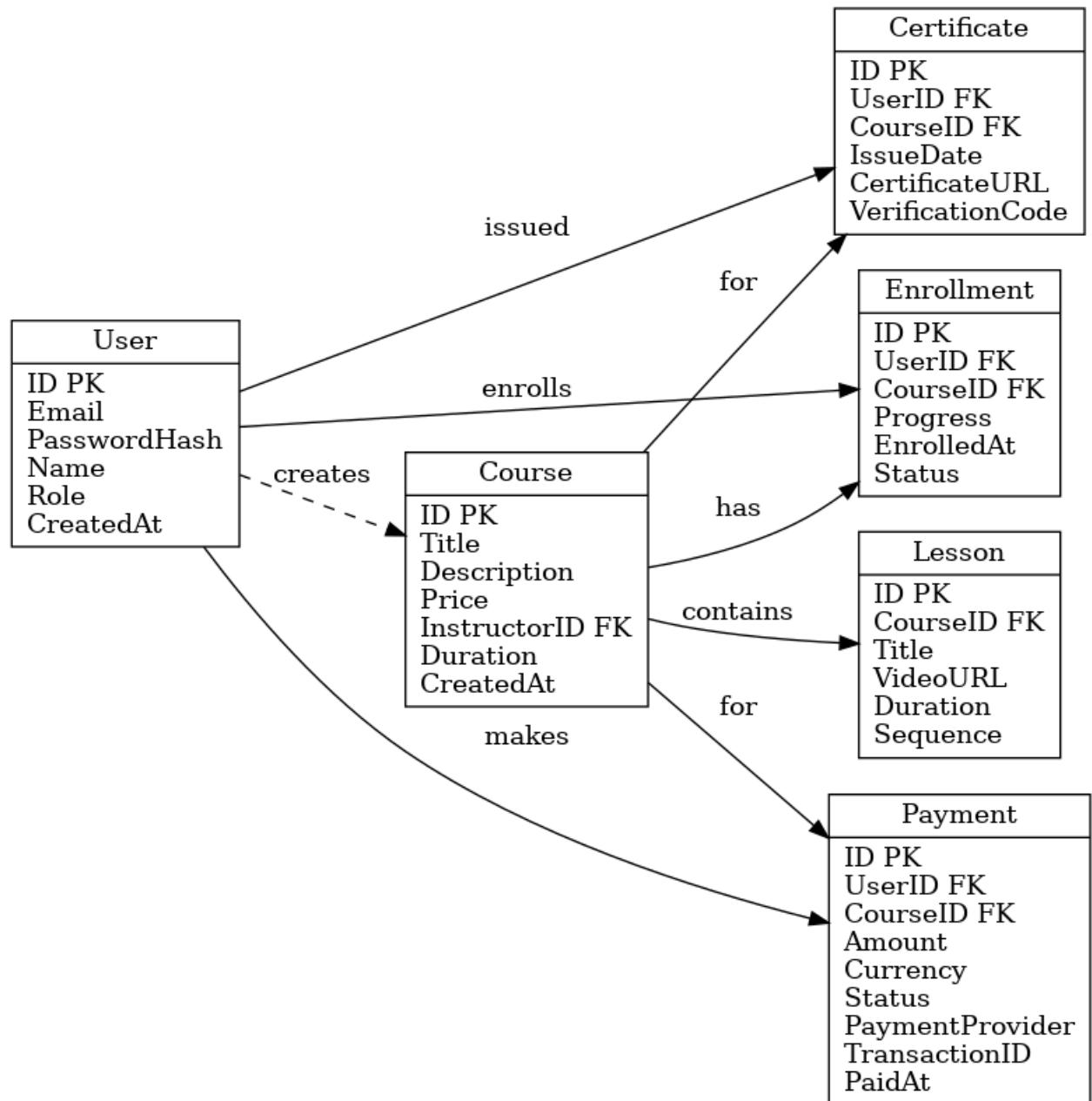
4.1 Use Case Diagrams



4.2 Data Flow Diagrams



4.3 Entity-Relationship Model



Schema

User (

ID INT PRIMARY KEY,

```
Email VARCHAR(255) UNIQUE NOT NULL,  
  
PasswordHash VARCHAR(255) NOT NULL,  
  
Name VARCHAR(255),  
  
Role VARCHAR(50),  
  
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP  
  
);  
  
Course (  
  
ID INT PRIMARY KEY,  
  
Title VARCHAR(255) NOT NULL,  
  
Description TEXT,  
  
Price DECIMAL(10,2),  
  
InstructorID INT,  
  
Duration INT,  
  
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,  
  
FOREIGN KEY (InstructorID) REFERENCES User(ID)  
  
);  
  
Lesson (
```

```
ID INT PRIMARY KEY,  
  
CourseID INT,  
  
Title VARCHAR(255),  
  
VideoURL VARCHAR(255),  
  
Duration INT,  
  
Sequence INT,  
  
FOREIGN KEY (CourseID) REFERENCES Course(ID)  
);
```

```
Enrollment (  
  
ID INT PRIMARY KEY,  
  
UserID INT,  
  
CourseID INT,  
  
Progress DECIMAL(5,2),  
  
EnrolledAt DATETIME DEFAULT CURRENT_TIMESTAMP,  
  
Status VARCHAR(50),  
  
FOREIGN KEY (UserID) REFERENCES User(ID),  
  
FOREIGN KEY (CourseID) REFERENCES Course(ID)
```

);

Certificate (

ID INT PRIMARY KEY,

UserID INT,

CourseID INT,

IssueDate DATETIME,

CertificateURL VARCHAR(255),

VerificationCode VARCHAR(100),

FOREIGN KEY (UserID) REFERENCES User(ID),

FOREIGN KEY (CourseID) REFERENCES Course(ID)

);

Payment (

ID INT PRIMARY KEY,

UserID INT,

CourseID INT,

Amount DECIMAL(10,2),

Currency VARCHAR(10),

```
Status VARCHAR(50),  
  
PaymentProvider VARCHAR(100),  
  
TransactionID VARCHAR(255),  
  
PaidAt DATETIME,  
  
FOREIGN KEY (UserID) REFERENCES User(ID),  
  
FOREIGN KEY (CourseID) REFERENCES Course(ID)  
);
```

Appendices

Appendix A: Glossary

- **Adaptive Learning:** AI-adjusted lesson difficulty based on user performance.
- **Collaborative Filtering:** ML technique for recommendations based on similar users.

Appendix B: Sample Certificate Format



CERTIFICATE

of completion

Presented to :



Olivia D'Souza

For successfully completing the Professional Baking
Course

PRESENTED BY:

Afrin Etu

COURSE INSTRUCTOR

