# Mekelle University

**EIT-M**

**DEPARTMENT: SOFTWARE ENGINEERING**

**COURSE NAME: SOFTWARE TESTING AND QUALITY ASSURANCE**

**ASSESEMENT TYPE: INDIVIDUAL**

**PREPARED BY: KAPITAL GEBREHIWET**

**ID NUMBER: EITM/UR178681/12**

**SUBMITTED TO: INSTRUCTOR MESELE**

**Submission date:9/05/2017**

# SRS (software requirement specification)

# Table of content

1, introduction

- 1.1 purpose----------3
- 1.2 scope---------- 3
- 1.3 definition, acronyms and abbreviations----------3
- 1.4 references----------4

2, overall description

- 2.1 product perspective----------4
- 2.2 product functions----------4
- 2.3 user characteristics----------5
- 2.4 constraints----------5
- 2.5 assumption and dependency----------5

3 specific requirements
3.1 user authentication----------6
3.2 course management----------6
3.3 payment processing----------7
4 external interface requirement
4.1 user interface----------7
4.2 software interface----------7
5 other non-functional requirements
5.1 performance requirement----------7
5.2 security requirement----------8
5.3 maintenance requirement----------8

## 1 Introduction

# 1.1 purpose

The purpose of this document is to provide a detailed and comprehensive description of the requirements for the School System. This document serves as a guide for the development team to design, implement, and test the system. It outlines the functionality, performance, and constraints of the system, ensuring that all stakeholders have a clear understanding of the project's objectives. The end goal is to create a user-friendly platform that facilitates efficient course management and seamless payment processing.

# 1.2 scope

The School System is designed to be a web-based application that offers a platform for students to enroll in courses provided by the school. The system will include features such as user authentication, course management, and payment processing via Stripe. It will allow users to sign up, sign in, add courses, view available courses, and make payments for courses. The scope of this document is limited to the functional and non-functional requirements necessary to achieve these goals.

## 1.3 Definitions, acronyms and abbreviations

**SRS:** Software Requirement Specification

**UI**: User Interface

**DB**: Database

**Stripe**: A payment processing platform used for handling transactions within the system.

**Dark Mode**: A user interface theme that reduces the amount of light emitted by the screen, using a darker color palette.

**Search Functionality**: A feature that allows users to search for specific courses within the system.

**Sort by:** An option that enables users to arrange courses based on specific criteria, such as name, price, or date.

## 1.4 references

*stripe documentation

*clerk documentation

*mongo dB documentation

*next js documentation

*stack overflow (for fixing errors)

*Microsoft co-pilot

## 2, overall description

## 2.1 product perspective

The School System is envisioned as a website that will serve the needs of students and course administrators. It will interact with a backend database (MongoDB) for storing user credentials, course information, and payment details. The system will leverage Clerk for user authentication and Stripe for secure payment processing.

The UI, built with Next.js, will be designed to be intuitive and user-friendly, ensuring a seamless experience for all users.

## 2.2 product functions

The primary functions of the School System include:

*__User Authentication__: Enabling users to sign up and sign in securely using Clerk.

*__Course Management__: Allowing users to add, view, search, and sort courses.

*__Dark Mode__: Providing an optional dark theme for the UI.

*__Payment Processing__: Facilitating payments for courses through Stripe.

## 2.3 user characteristics

The target users of the School System are students who wish to enroll in courses offered by the school. These users are expected to have basic computer literacy and access to the internet. Additionally, there will be course administrators who will manage the course offerings and handle administrative tasks related to the system..

## 2.4 constraints

The following constraints must be considered during the development of the system:

*The system must be accessible on various devices, including desktops, laptops, and mobile devices.

*Payment processing must comply with Stripe’s security and compliance guidelines.

*User data must be stored securely to protect privacy and prevent unauthorized access.

## 2.5 assumptions and dependencies

*Users will have a stable internet connection to access the system.

*The school will provide the necessary course content and details for the system.

*The development team will have access to Clerk's, Next js's, and MongoDB's API and documentation for integration.

# 3 specific requirements

## 3.1 user authentication

**Sign up**

*__Fields__: Username, Password, Email.

*Validation for unique email addresses and password strength to ensure security.

*Managed by Clerk to streamline the sign-up process and handle user sessions.

**Sign in**

* **Fields**: Username.

*Authentication managed by Clerk against stored credentials in MongoDB.

*Option to reset the password if the user forgets it, facilitated by Clerk.

## 3.2 course management

**Add Course**:

*Fields: Course Name, Course Description, Image URL, Price (Course Fee).

*Connection to MongoDB to store and retrieve course details.

*Button: Add Course to save the course information in the database.

**View Courses:**

*Display a list of all available courses.

*Features: Dark Mode toggle, Search bar to find specific courses, Sort by options (Name, Price, Date).

## 3.3 payment processing

**Buy Course**:

*Fields: Card Number, Expiry Date, CVC.

*Integration with Stripe's API to handle payment transactions.

*Alert message to confirm successful payment.

*Store payment details in MongoDB for record-keeping and future reference.

# 4 external interface requirements

## 4.1 user interface

*Clean and intuitive UI design using Next.js to enhance user experience.

* Navigation bar with links to Sign Up, Sign In, Add Course, and View Courses.

*Dark Mode toggle switch for user preference.

## 4.2 software interface

*Integration with Clerk's API for user authentication.

*Integration with Stripe's API for payment processing.

*Connection to the backend MongoDB database for storing user, course, and payment details.

# 5 other non functional requirements

## 5.1 performance requirements

*The system should be able to handle multiple users accessing it simultaneously without significant performance degradation.

*Page load times should not exceed 2 seconds to ensure a smooth user experience.

## 5.2 security requirements

*All data transfers between the client and server must be encrypted using SSL/TLS.

* User passwords must be hashed before storing them in the database to protect against unauthorized access.

## 5.3 maintenance requirement

*Regular updates and maintenance to ensure compatibility with new browser versions and technologies.

*Scheduled backups of the MongoDB database to prevent data loss