Software Requirements Specification for Test_School Competency Assessment Platform

EForgeIT Development Team

August 8, 2025

Contents

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) delineates the functional and non-functional requirements for the Test_School Competency Assessment Platform, a web-based application designed to assess and certify users' digital competencies through a structured, multi-stage evaluation process aligned with the Test_School framework (levels A1 to C2).

1.2 Product Scope

The platform facilitates a three-step assessment process to evaluate digital skills, automates certification based on performance, enforces test integrity through timers and optional secure browser controls, and provides robust user management. It aims to deliver an accurate, secure, and user-friendly experience for assessing digital competencies.

1.3 Definitions, Acronyms, and Abbreviations

- Test_School: A framework for assessing digital competencies with levels ranging from A1 to C2.
- SRS: Software Requirements Specification.
- JWT: JSON Web Token, used for secure authentication.
- OTP: One-Time Password, for additional authentication security.
- SEB: Safe Exam Browser, for secure test environments.
- CRUD: Create, Read, Update, Delete operations.
- UI: User Interface.
- API: Application Programming Interface.

1.4 References

- Asana: Write a Software Requirement Document, https://asana.com/resources/software-requirement-document-template.
- Perforce: How to Write a Software Requirements Specification (SRS) Document, https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-documen
- GitHub: SRS-Template, https://github.com/jam01/SRS-Template.

1.5 Overview

This document is structured as follows:

- Section 2 provides an overall description, including product perspective, functions, user classes, operating environment, constraints, and assumptions.

- Section 3 details specific requirements, encompassing functional, external interface, and non-functional requirements.
- Section 4 includes appendices with additional information.

2 Overall Description

2.1 Product Perspective

The Test_School Competency Assessment Platform is a standalone web application accessible via standard web browsers. It operates on a client-server architecture, interacting with a backend server for data storage and processing, and may integrate with third-party services for email and SMS notifications.

2.2 Product Functions

The platform provides the following core functionalities:

- User registration and authentication with secure token management.
- Management of user roles: Admin, Student, and Supervisor.
- Creation and management of assessment questions categorized by competency and level.
- Conducting a three-step assessment process with timed tests.
- Automatic scoring and certification based on assessment results.
- Generation and optional delivery of digital certificates.
- Optional secure exam environment with browser lockdown and video monitoring.

2.3 User Classes and Characteristics

- Students: Primary users who take assessments. They require basic digital literacy to navigate the platform.
- Admins: Manage users, questions, and view reports. They need administrative privileges and system training.
- Supervisors: Oversee students' progress and results, with specific roles to be defined during development.

2.4 Operating Environment

The platform operates as a web-based application, accessible via modern web browsers (e.g., Chrome, Firefox, Safari, Edge) on devices with a stable internet connection. The backend requires a server with sufficient resources to handle the expected user load.

2.5 Design and Implementation Constraints

- The frontend must be developed using React.js, TypeScript, Redux, RTK Query, Axios, and Tailwind CSS.
- The backend must utilize Node.js, Express, TypeScript, Mongoose, and MongoDB.
- Authentication must employ JWT for secure token management.
- Email and SMS services must use Nodemailer or Twilio.
- The system must adhere to industry best practices for code organization, security, and user experience.

2.6 Assumptions and Dependencies

- Users have access to a stable internet connection and compatible web browsers.
- Third-party services (e.g., Nodemailer, Twilio) are available and configured correctly.
- The Test_School framework and its levels (A1 to C2) are predefined and stable during development.

3 System Features and Requirements

3.1 Functional Requirements

3.1.1 User Authentication and Management

- The system shall allow users to register with an email address and password.
- Upon registration, the system shall send a verification email with a link or code.
- The system shall allow verified users to log in using their email and password.
- Upon successful login, the system shall issue a JWT access token (valid for approximately 15 minutes) and a refresh token (valid for approximately 7 days).
- The system shall enable token refresh using the refresh token without requiring re-login.
- The system shall provide OTP verification via email or SMS, with resend functionality.
- The system shall offer a forgot password feature, enabling password reset via email.
- Passwords shall be hashed using bcrypt before storage.
- The system shall support three user roles (Admin, Student, Supervisor) with distinct permissions.

3.1.2 Assessment Management

- The system shall provide a three-step assessment process for evaluating digital competencies.

- Step 1: Levels A1 and A2
  - Present 44 questions (22 from A1, 22 from A2, one per competency per level).
  - Calculate score as the percentage of correct answers.
  - If score $< 25\%$, the user fails and is barred from further assessments.
  - If $25\% \leq$ score $< 50\%$, certify user at level A1.
  - If $50\% \leq$ score $< 75\%$, certify user at level A2.
  - If score $\geq 75\%$, certify user at level A2 and allow progression to Step 2.
- Step 2: Levels B1 and B2
  - Present 44 questions (22 from B1, 22 from B2).
  - Calculate score as the percentage of correct answers.
  - If score $< 25\%$, user retains A2 certification.
  - If $25\% \leq$ score $< 50\%$, certify user at level B1.
  - If $50\% \leq$ score $< 75\%$, certify user at level B2.
  - If score $\geq 75\%$, certify user at level B2 and allow progression to Step 3.
- Step 3: Levels C1 and C2
  - Present 44 questions (22 from C1, 22 from C2).
  - Calculate score as the percentage of correct answers.
  - If score $< 25\%$, user retains B2 certification.
  - If $25\% \leq$ score $< 50\%$, certify user at level C1.
  - If score $\geq 50\%$, certify user at level C2.
- Each assessment step shall have a configurable time limit, defaulting to 44 minutes.
- The system shall display a countdown timer during assessments.
- Upon time expiration, the system shall automatically submit the assessment.
- Each assessment step shall be taken only once, except as specified.
- Users failing Step 1 (score $< 25\%$) shall be prevented from further assessments.

3.1.3 Question Pool Management

- The system shall maintain a pool of 132 questions, categorized by 22 competencies and 6 levels (A1 to C2).
- Administrators shall be able to add, edit, and delete questions.
- Each question shall be associated with one competency and one level.
- Questions shall be multiple-choice with one correct answer.

- For each assessment step, the system shall randomly select one question per competency per relevant level.

3.1.4 Certification Generation

- Upon assessment completion, the system shall generate a digital certificate based on the highest level achieved.
- The certificate shall include the user's name, certification level, and date.
- Optionally, the system shall allow certificate download as a PDF and email delivery.

3.1.5 Timer System

- Each assessment step shall have a configurable time limit, defaulting to 44 minutes (1 minute per question for 44 questions).
- Administrators shall be able to set the time limit for each step.
- The system shall display a countdown timer during assessments.
- Upon time expiration, the system shall submit the assessment and calculate the score.

3.1.6 Secure Exam Environment (Optional)

- The system may integrate with Safe Exam Browser (SEB) to restrict navigation and input methods.
- The system may enable live video recording during exams to ensure test integrity.

3.2 External Interface Requirements

3.2.1 User Interfaces

- The system shall provide a web-based UI accessible via standard browsers.
- The UI shall be responsive and optimized for desktop and mobile devices.
- The UI shall include pages for registration, login, assessment, results, and certificate download.
- Administrators shall access a dashboard for managing users and questions.

3.2.2 Hardware Interfaces

- No specific hardware interfaces are required beyond standard web server and client hardware.

3.2.3 Software Interfaces

- The system shall interact with a MongoDB database using Mongoose.
- The system shall use Nodemailer or Twilio for email and SMS services.
- Optionally, the system may integrate with SEB and video recording services.

3.2.4 Communication Interfaces

- The system shall use HTTP/HTTPS for client-server communication.
- Email and SMS shall be used for notifications and OTP delivery.

3.3 Non-Functional Requirements

3.3.1 Security Requirements

- User passwords shall be hashed using bcrypt.
- Authentication shall use JWT with secure token management.
- The system shall implement measures to prevent unauthorized access and data breaches.
- Optionally, a secure exam environment shall include browser lockdown and video monitoring.

3.3.2 Performance Requirements

- The system shall handle at least 100 concurrent users without significant performance degradation.
- Response times for user actions shall be less than 2 seconds under normal conditions.
- The assessment timer shall be accurate to within 1 second.

3.3.3 Usability Requirements

- The UI shall be intuitive for users with basic digital literacy.
- The system shall provide clear instructions and feedback during assessments.
- The platform shall be accessible on desktops, tablets, and smartphones.
- Pagination shall be implemented on admin dashboard tables and product pages.

3.3.4 Reliability Requirements

- The system shall achieve an uptime of at least 99.9%.
- The system shall recover gracefully from failures, with comprehensive error handling.

3.3.5 Scalability Requirements

- The system shall scale to accommodate increasing users and data.
- The architecture shall support horizontal scaling if necessary.

3.3.6 Maintainability Requirements

- The code shall be well-structured, adhering to industry standards.
- Components shall be reusable where possible.
- The system shall include meaningful comments and documentation.
- All variables shall use TypeScript types.
- State management shall utilize Redux.

4 Future Considerations

- Develop an admin dashboard for enhanced user management, reports, and analytics.
- Implement advanced test analytics for per-competency and performance trends.
- Enable email notifications for results and certifications.
- Further optimize the UI for mobile devices.

5 Appendices

5.1 Assessment Flow Summary

Step	Levels	Scoring and Certification
Step 1	A1, A2	$< 25\%$: Fail, no retake 25–49.99%: A1 certified 50–74.99%: A2 certified $\geq 75\%$: A2 certified, proceed to Step 2
Step 2	B1, B2	$< 25\%$: Remain at A2 25–49.99%: B1 certified 50–74.99%: B2 certified $\geq 75\%$: B2 certified, proceed to Step 3
Step 3	C1, C2	$< 25\%$: Remain at B2 25–49.99%: C1 certified $\geq 50\%$: C2 certified

Table 1: Assessment Flow and Certification Criteria