**Software Requirement**

**Specifications**

**Of**

**Web-Based Quiz Platform**

**Prepared by**

**Manoj Pentapalli – 240850120075**

**Mudit Kumar – 240850120079**

**Sudhanshu Patel – 240850120143**

**Utkarsh Tyagi – 240850120153**

**Vashu Chowdary – 240850120157**

**Course: DAC,**

**C-DAC Bangaluru.**

**Date:**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reviewers**

|  |
| --- |
| **Reviewer Name** |
|  |
|  |
|  |

**Table of Contents**

# 1. Introduction 1

## 1.1 Purpose 1

## 1.2 Scope 1

## 1.3 Definitions, Acronyms, and Abbreviations 1

## 1.4 References 1

**2. Functional Requirements 1**

**2.1 Admin Use Case Diagram 1**

**2.2 User Use Case Diagram 2**

# 3. Overall Description 3

## 3.1 Product Perspective 3

## 3.2 Product Features 3

## 3.3 User Classes and Characteristics 4

## 3.4 Operating Environment 4

## 3.5 Design and Implementation Constraints 4

## 3.6 Assumptions and Dependencies 4

# 4. Specific Requirements 4

## 4.1 Functional Requirements 4

## 4.2 Non-Functional Requirements 5

## 4.3 Database Requirements 5

## 4.4 Interface Requirements 5

# 5. System Design 5

## 5.1 Architecture Diagram 5

## 5.2 Flow Diagram 5

# 6. Appendices 5

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to provide a comprehensive description of the functional, non-functional, and system requirements for the Web-Based Quiz Platform. The platform will allow educators to manage quizzes efficiently and provide students with an engaging quiz experience.

## 1.2 Scope

The Web-Based Quiz Platform will:  
- Enable educators to create and manage question banks.  
- Allow students to attempt quizzes and get their results immediately.  
- Generate randomized quizzes from predefined question banks.  
- Provide analytics for educators to evaluate student performance.  
Technologies used include Java, Spring Boot, MySQL, HTML, CSS, and JavaScript.

## 1.3 Definitions, Acronyms, and Abbreviations

- Quiz: A test or assessment designed for students.  
- Spring Boot: A Java-based framework for developing backend applications.  
- MySQL: A relational database management system for data storage.

## 1.4 References

- Spring Boot Documentation: https://spring.io/projects/spring-boot  
- MySQL Documentation: <https://dev.mysql.com/doc/>

**2. Functional Requirements**

**2.1 Admin Use Case Diagram**

|  |  |
| --- | --- |
| Actor | Description |
| New Admin | A person registering as an administrator. |
| Registered Admin | An admin with an existing account who manages quizzes. |

# 

**2.2 User Use Case Diagram**

|  |  |
| --- | --- |
| Actor | Description |
| New User | A person registering as User. |
| Registered User | A User with an existing account who takes quizzes. |

# 

# 3. Overall Description

## 3.1 Product Perspective

The platform will be a standalone web application accessible via browsers. It is designed for educational institutions, online learning platforms, and organizations conducting assessments.

## 3.2 Product Features

- Educators can:  
 - Create, edit, and manage question banks.  
 - Configure quiz settings (duration, pass criteria).  
 - View and download performance reports.  
- Students can:  
 - Attempt quizzes with a timer.  
 - View immediate feedback and scores.  
- The system supports secure authentication, quiz generation, and data storage in MySQL.

## 3.3 User Classes and Characteristics

- Educators: Technically proficient users who manage quizzes.  
- Students: End-users who take quizzes; require a user-friendly interface.

## 3.4 Operating Environment

The platform will use:  
- Front-End Technologies: Java, HTML, CSS, JavaScript.  
- Back-End Framework: Spring Boot.  
- Database: MySQL.  
- Deployment: Web server (e.g., Apache Tomcat).

## 3.5 Design and Implementation Constraints

- The platform can handle at least 50 concurrent users.  
- Data integrity must be maintained during quiz generation.

## 3.6 Assumptions and Dependencies

- Users will have access to devices with a modern web browser.  
- MySQL database is configured and operational.

# 4. Specific Requirements

## 4.1 Functional Requirements

- User Management:  
 - Secure login and registration for educators and students.  
 - Password recovery functionality.  
- Quiz Management:  
 - Create, edit, and delete quizzes.  
 - Tag questions by difficulty or category.  
 - Generate random quizzes based on selected tags or categories.  
- Question Bank:  
 - Upload bulk questions via CSV.  
 - Support multiple-choice.  
- Quiz Taking:  
 - Countdown timer for quizzes.  
 - Automatic submission upon timer expiration.  
- Scoring and Feedback:  
 - Immediate feedback for objective questions.  
 - Manual grading for subjective questions by educators.  
- Reports and Analytics:  
 - Generate performance reports by student, quiz, or category.  
 - Download reports in CSV or PDF formats.

## 4.2 Non-Functional Requirements

- Performance: The platform can support 50 concurrent users with minimal latency.  
- Usability: The interface should be intuitive for educators and students.  
- Scalability: The system should allow future integration with mobile applications.   
- Availability: The platform must have 99.9% uptime.

## 4.3 Database Requirements

- Tables for users, questions, quizzes, quiz attempts, and reports.  
- Efficient indexing to speed up queries.

## 4.4 Interface Requirements

- User Interface: Responsive web pages for desktops, tablets, and mobile devices.  
- API: RESTful APIs for backend services.

# 5. System Design

## 5.1 Architecture Diagram

- Frontend: HTML, CSS, JavaScript for UI.  
- Backend: Spring Boot handles business logic and REST APIs.  
- Database: MySQL stores user, quiz, and performance data.

## 5.2 Flow Diagram

Educator Workflow: Login → Manage Question Bank → Create Quiz → Publish Quiz → View Reports.  
Student Workflow: Login → Attempt Quiz → Submit Quiz → View Results.

# 6. Appendices

- Technology Stack: Java, Spring Boot, MySQL, HTML, CSS, JavaScript.  
- Tools: Eclipse, Spring Tools Suite, MySQL Workbench, Postman.  
- Standards: OWASP Security Guidelines, REST API conventions.