**Software Requirements Specification**

**for**

TestRight

**Version 1.0 approved**

**Prepared by Triangle Corporation**

**DA-IICT**

**Table of Contents**

**Table of Contents 2**

**1. Introduction 3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

**2. Overall Description 3**

2.1 Product Perspective 3

2.2 Product Functions 4

2.3 User Classes and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements 5**

3.1 User Interface 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 5

**4. System Features 6**

**5. Other Nonfunctional Requirements 7**

5.1 Performance Requirements 7

5.2 Security Requirements 7

5.3 Software Quality Attributes 7

# **Introduction**

## **Purpose**

The purpose of the document is to cover the features of our application, i.e TestRight - Exam tools, along with discussing how it will work, under what constraints.It is intended for users and potential developers of the application*.*

## **Document Conventions**

This Document is created based on the IEEE template for System Requirement Specification Documents.

## **Intended Audience and Reading Suggestions**

* Students who want to give tests from a digital platform, whether it be open tests like entrance exams or closed ones like class or tuition tests.
* Teachers who want to conduct tests to evaluate performance of students on different subjects.
* For project managers who want to test expertise of team members.
* Recruiters who want to hire potential employees based on their skills.

## **Product Scope**

TestRight is a system for conducting all kind of exams and assessments online with

security measures to prevent cheating and other exam related illegal activities.

TestRight will eliminate the need of traditional pen - paper exams and requirement of creation of separate portals for different exams. It aims to centralize the examination for coming generation.

## **References**

*IEEE STD 730-2002, IEEE Standard*

# **Overall Description**

## **Product Perspective**

*There are many test-taking applications like hackerrank,our own college’s placement test-taking app.,TCS Eon etc.Our application takes inspiration from these but it stands out as it’s created such as it supports taking small-scale tests for tuitions and classes as wells as tests on a wide scale like entrance exams,organisation recruitment tests,college tests and the like.And it can be used for non-tech fields also which enables the application to target a wider audience.*

## **Product Functions**

* Create test
* View results
* Edit test details
* Edit profile
* Enroll for test
* Generate unique test-ID

## **User Classes and Characteristics**

* Students who want to give tests from a digital platform, whether it be open tests like entrance exams or closed ones like class or tuition tests.
* Teachers who want to conduct tests to evaluate performance of students on different subjects.
* For project managers who want to test expertise of team members.
* Recruiters who want to hire potential employees based on their skills.

## **Operating Environment**

* Software requirements:
  + Windows operating system 7 or above
  + NoSQL server
* Internet connectivity
* Hardware requirements:

## **Design and Implementation Constraints**

* Time:
  + The project needs to be completed within the time duration stated in project schedule.
* Scope:
  + The scope of our application is wide if used on a frequent basis and used for the right purpose.

## **User Documentation**

*User manual is available as a guide for users to use the system correctly.*

## **Assumptions and Dependencies**

* Users will adapt our system.
* Internet connection will be stable so that end-user can use the application.
* There won’t be any undefined problem occuring at server side.
* The technology we are using won’t become obsolete in the future.

# **External Interface Requirements**

## **User Interfaces**

* Refer Wireframe files to get an idea of the layout used for all web pages.
* We’ve made our website responsive to meet industry standards.
* Use case:



## **Hardware Interfaces**

* Any hardware setup that supports Google Chrome/Mozilla Firefox.
* AWS server(once the application becomes scalable and will require to be deployed on a paid server)

## **Software Interfaces**

* MongoDB database
* Node.js,Express.js
* Visual Studio Code

## **Communications Interfaces**

* Our project is a web application so it requires internet connection.(with HTTP implemented)

# **System Features**

This section highlights the features of TestRight that can be used by the users of the application.

Types of Users:

1)Examiner

2)Student

**Functions/privileges to examiner:**

* Examiner can create tests by specifying test details and question bank.After that, a unique test ID related to the test will be generated, which the students can use to enroll for the test.
* Examiner can also view/edit their profile information.
* It is also possible for the examiner to edit test details before the start time of the exam.
* Examiner can view the results of students after the test gets over.

**Functions/privileges to student:**

* Students can enroll for a test by entering it’s test ID specified by the examiner.
* Students can also prepare for tests by practising from an ever-growing “open question-bank” of already conducted tests.They can search for specific category tests by providing tags.

* Students are also able to view/edit their profile information.

* They can give tests when the test becomes active.And to prevent any cheating, the screen automatically enters full-screen mode when the student starts the test.
* Students can view their test result after submitting their responses.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

1. Technical team needs to be active for recovery in the event of

server crash.

2. Development team needs to be active to troubleshoot any bugs that can occur at

any point when the application is live.

3.System must be scalable to handle any future growth.

## **Security Requirements**

* Only database admin can view user’s data.
* All the exam related processing happens at back-end, so it’s not possible for anyone to inspect or tamper with the system.

## **Software Quality Attributes**

* The application should be easy-to-use by the users.
* It should be efficient and fast so that the students giving tests are able to smoothly give their responses in real-time.
* It should be scalable so that it is able to easily accommodate any additional features in the future.