# Quiz Manager
# Functional Specification Document

BY

YOGITHASATYASAI PANTHAM

FUNDAMENTAL JAVA PROJECT -- QUIZ

## Purpose:

- The purpose of this document is to give a brief description about the end to end specifications of the project.
- The main aim of this project is to develop an application (API Oriented , Web-based) that deals with quiz management.

## Introduction:

This project helps to do CRUD operations and get the questions based on the difficulty level. This project is a console application and execution.

## Overview:

In the project user can login as a student and take the test. User can get their results at the end of the exam. In addition to this user can add questions.

## Scope of the project:

- ➔ It gives automatic quiz with the help of open questions and mcq questions.
- ➔ It gives grades automatically for the quiz.
- ➔ It creates data access with CRUD methods.
- ➔ It creates a configurable file for application.
- ➔ Exports quiz to plain text.

## Definitions:

To run the program, the following steps are used:

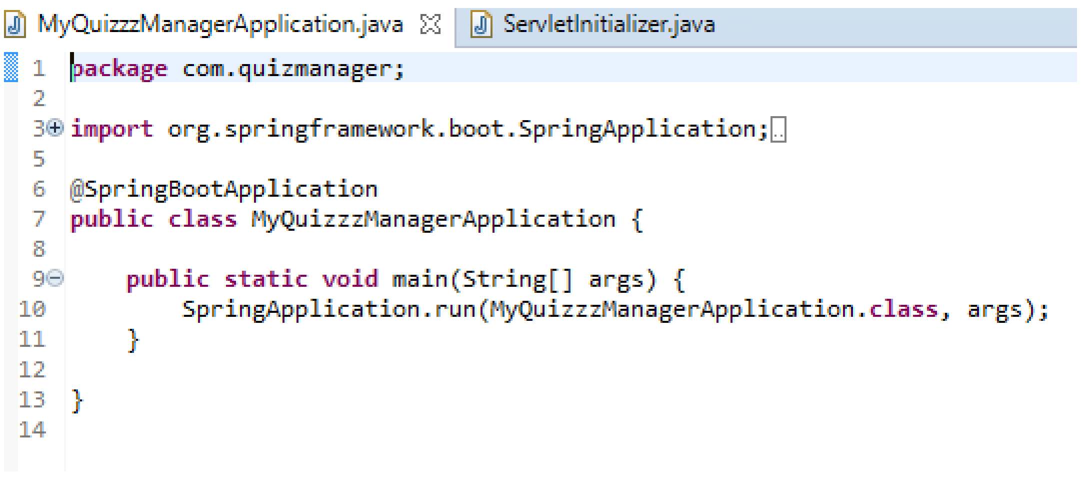
- ➔ Install h2
- ➔ Install java 8
- ➔ Install html
- ➔ Install spring boot
- ➔ Create tables in database
- ➔ Add questions
- ➔ Export questions in a text file

## Software requirements:

| | |
|---|---|
| FRONT END | Css , HTML , JSP |
| BACKEND | Java 8, spring-boot, TOMCAT |
| DATABASE | H2 |
| VERSION | GIT |

## System Requirements:

→ Main source code:



MyQuizzzManagerApplication.java ServletInitializer.java

```java
package com.quizmanager;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class MyQuizzzManagerApplication {

    public static void main(String[] args) {
        SpringApplication.run(MyQuizzzManagerApplication.class, args);
    }

}
```

→ List of topics:

For example,

```java
public Exam getExam() {
    List<Questions> q = new ArrayList<Questions>();
    Exam exam = new Exam();
    exam.setQuestionHeading("Information Techonogy Quiz");

List<Questions> q1 = new ArrayList<Questions>();
Exam exam11 = new Exam();
exam11.setQuestionHeading("Aptitude Test");

List<Questions> q2 = new ArrayList<Questions>();
Exam exam21 = new Exam();
exam21.setQuestionHeading("Logical Reasoning Questions");
```

→ Add Question:

User can add the questions

```html
<div class="wrapper">
  <div class="search_body">
  <h1><span style="color:#3369E8">Wel</span><span style="color:#D50F25">Come</span><span styl
      <div><input type="text" placeholder="Search Quiz.." id="before_search" autofocus>
      <input type="button" class="button" value="Search" id="srchbtn">
        <input type="button" class="button" value="Add Questions" id="addbtn">
```

## → Preparing Questions:

```
public Exam getExam() {
    List<Questions> q = new ArrayList<Questions>();
    Exam exam = new Exam();
    exam.setQuestionHeading("Information Techonogy Quiz");

    Questions questions = new Questions();
    questions.setQuestion("Which of the following languages is more suited to a structured pr
    questions.setOption1("PL/1");
    questions.setOption2("FORTRAN");
    questions.setOption3("BASIC");
    questions.setOption4("PASCAL");
    questions.setCorrectIndex("0");
    q.add(questions);
```

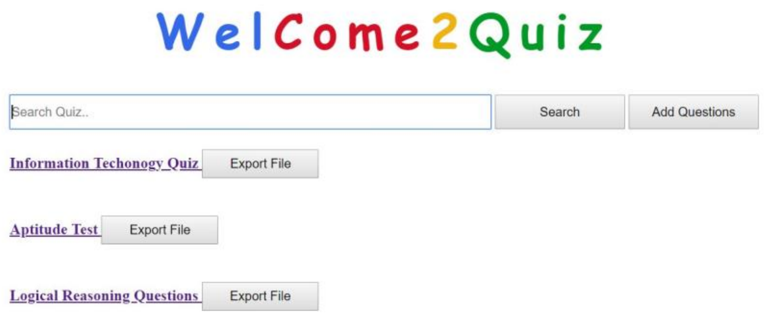
## → Save, search & send:

```
@GetMapping("/saveExam")
public String saveExam() {
    utillservices.saveExam();
    return "";
}

@GetMapping("/sendExamquestions")
public Exam sendExamquestions() {
    Integer examId = (Integer) httpSession.getAttribute("examID");
    return exambusiiness.findById(examId);
}

@PostMapping("/searchExam")
public  List<Exam> searchExam(@RequestBody String searchExam) {
    return exambusiiness.searchExam(searchExam);
}
```

Please find the welcome screen below

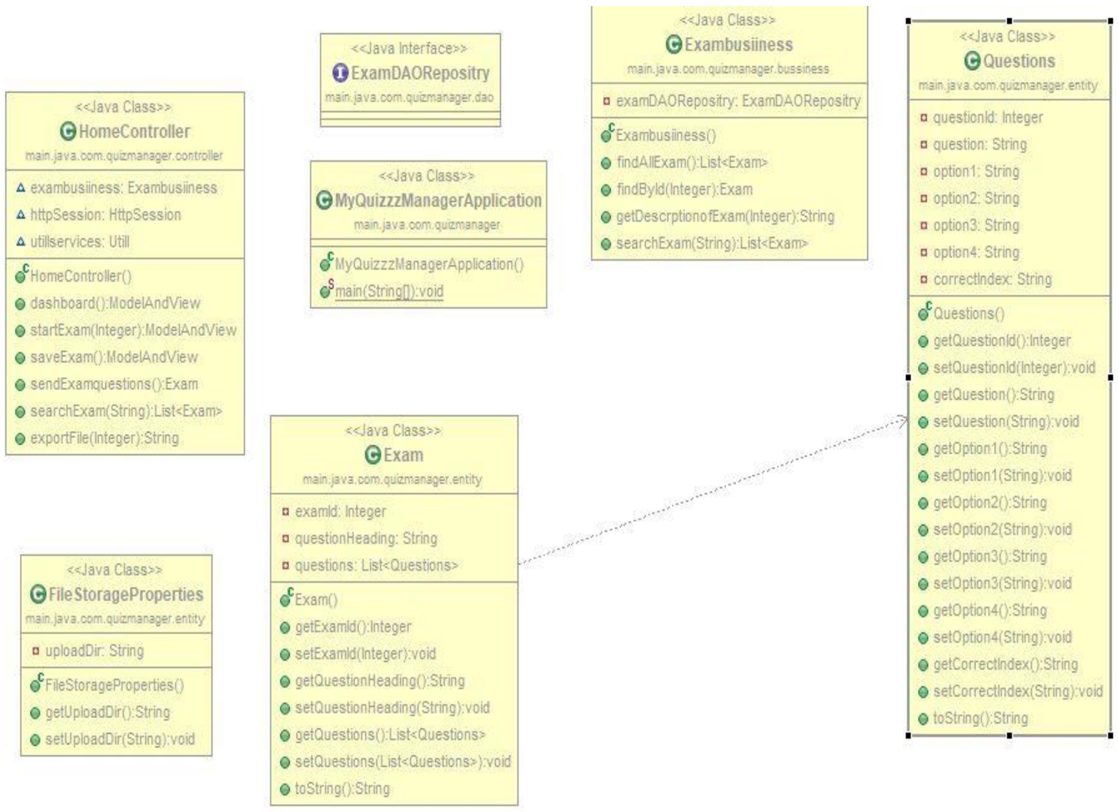


## → Hardware Requirements:

| # | HARDWARE | REQUIREMENT |
|---|---|---|
| 1. | Operating System | Compatible with Windows, Mac OS X, Linux |
| 2. | RAM | Minimum required 124MB |
| 3. | Disk Space | Minimum required 124MB |
| 4. | Processor | 64-bit, four-core, 2.5 GHz minimum per core |

## UML DIAGRAM:

### Class diagram



## BIBILIOGRAPHY:

https://thomas-broussard.fr/work/java/courses/project/fundamental.xhtml