**<Code>**

**Version: <Version>**

**Table of Content**

*[Press F9 to edit or update table of content. Please remove this text when applying.]*

[1 Introduction 1](#_Toc511305587)

[1.1 Purposes 1](#_Toc511305588)

[1.2 Scope 1](#_Toc511305589)

[1.3 [Definitions, Acronyms, and Abbreviations] 1](#_Toc511305590)

[1.4 [References] 1](#_Toc511305591)

[1.5 Overview 1](#_Toc511305592)

[2 System Overview 2](#_Toc511305593)

[2.1 Assumptions 2](#_Toc511305594)

[2.2 Constraints 2](#_Toc511305595)

[2.3 [Technology Suggestion] 2](#_Toc511305596)

[3 Architecture Design 2](#_Toc511305597)

[3.1 System Overview 2](#_Toc511305598)

[3.2 Architecture 4](#_Toc511305599)

[3.2.0.1 Backend 4](#_Toc511305600)

[3.2.0.2 Frontend 5](#_Toc511305601)

[3.3 Software Modules 5](#_Toc511305602)

[4 [Interface Design] 5](#_Toc511305603)

[5 [Data Design] 14](#_Toc511305604)

[5.1.1 Database Management System Files 14](#_Toc511305605)

[5.1.1 Text-based Database System Files 14](#_Toc511305606)

[6 System Design 15](#_Toc511305607)

[7 [User Interface Design] 15](#_Toc511305608)

[7.1 [Interface Design Rules] 15](#_Toc511305609)

[7.2 [Common] 15](#_Toc511305610)

[7.2.1.1 [Screen Transition] 16](#_Toc511305611)

[7.2.1.2 Graphical Interface 16](#_Toc511305612)

[7.3 Application screens 16](#_Toc511305613)

[7.3.1.1 [Screen Transition] 16](#_Toc511305614)

[7.3.1.2 Graphical Interface 17](#_Toc511305615)

[7.3.1.3 [<Screen 1..n>] 17](#_Toc511305616)

[8 [Others] 17](#_Toc511305617)

**List of Tables**

[Table 1: List of software modules 5](#_Toc511207636)

[Table 3: Common Screens 15](#_Toc511207637)

[Table 6: Screens list of <Module 1..n> 16](#_Toc511207638)

**List of Figures**

[Figure 1: Overview system architecture 3](#_Toc511207665)

[Figure 2: Architecture of Angular4/SpringBoot framework 3](#_Toc511207666)

[Important notes:

- This document ONLY give the overview view about future system. So, all information in this document contains main items which are used to represent business of software.

- Write refers to other documents, if that document contains all information this sections/sub-sections.

]

1. Introduction
   1. Purposes

This document provides a technical approach for implementing Training System, using a number of different architectural diagráms to view different aspects of the system. It also presents the decisions that will govern how the system will be built from Java programming perspective.

This document’s aim is to describe the architecture at high level, in which the system will be described as a set of functional layers, with technologies suggested for each layers.

* 1. Scope

This document is prepared for the Training Systemin scope of RRC.

* 1. [Definitions, Acronyms, and Abbreviations]

[This section is optional. If this section isn’t set, remove it out of the document.

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret this document.]

* 1. [References]

[This section is optional. If this section isn’t set, remove it out of the document, please!

This subsection provides a complete list of all documents referenced elsewhere in the System Design document. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

* 1. Overview

[This subsection describes what the rest of this document contains and explains how the document is organized.]

1. System Overview
   1. Assumptions

This system is designed basing on these following assumptions:

* Windows
* Chrome/FiredFox
* Mysql
* Java
* Tomcat
  1. Constraints

This system should be complied with following items:

* Chrome and Firefox is the latest in end-user environment
* Use Bootstrap to build user interface in html
* Use Angular 4 to transfer data from browser to server.
* Every request in this system must be less than 5s.
* Verification and validation both FrontEnd and Restful API(BackEnd)
  1. [Technology Suggestion]

N/A

1. Architecture Design

[Describes all following content:

* The overview system architecture and organization (Diagrams should map to the System Architecture document.)
* The software modules including modules/block diagram
* Network/integration diagram and explanation for these diagrams.
* Identify database platform, compilers, operating systems, communication system…

In case, architecture information is described clearly at System Architecture document, this section can be referred to System Architecture document.

* 1. System Overview

The overview system architecture

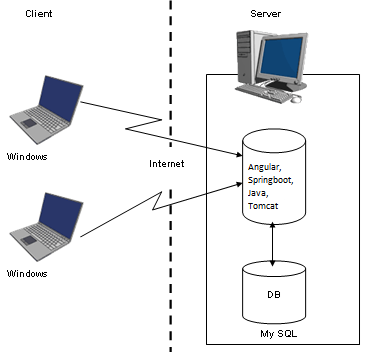


Figure 1: Overview system architecture

Explanation:

* Operating systems: Windows
* Database: MySql 5.7
* Java Development Kit(JDK): OpenJDK 1.8.0 64-Bit
* Web Application Server: Tomcat 8.5
* FrontEnd: Angular 4
* BackEnd: Spring Boot 1.5.8
* Browser: Chrome, Firefox

- Architecture of Angular 4/SpringBoot framework

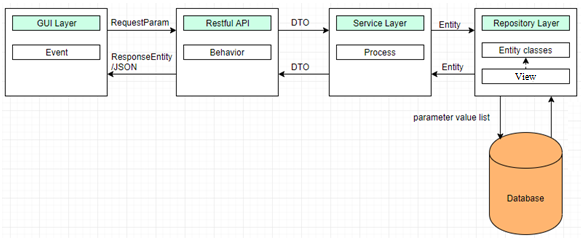


Figure 2: Architecture of Angular4/SpringBoot framework

Explanation:

GUI Layer (Front End): Tầng giao diện người dùng được xây dựng bằng Angular 4 framework có nhiệm vụ tiếp nhận thao tác và dữ liệu người dùng nhập đồng thời hiển thị các đáp ứng bằng định dạng JSON từ Restful API Layer trả về.

Restful API Layer: Có trách nhiệm quản lý truy cập và xử lý các yêu cầu với các hành vi GET/POST/PUT/DELETE sau đó chuyển xử lý cho Service Layer đồng thời trả về các đáp ứng cho Front End. Restful API Layer cũng chuyển dữ liệu hoặc các DTO (Data Transfer Object) cho Service Layer.

Service Layer: là tầng xử lý nghiệp vụ ứng dụng, service layer tiếp nhận dữ liệu hoặc các đối tượng DTO (Data Transfer Object) từ Restful API Layer để xử lý và trả về các đới tượng dữ liệu tương ứng với bảng (Entity) cho Repository Layer. Service Layer cũng nhận các dữ liệu từ Repository Layer thực hiện xử lý và trả về các DTO/Entity cho Restful API.

Repository Layer: Có trách nhiệm quản lý toàn bộ các truy cập cơ sở dữ liệu, thực hiện các thao tác thêm mới, xem sửa, xóa dữ liệu theo yêu cầu của Service Layer.

* 1. Architecture

Table 3: Common Screens

|  |  |  |
| --- | --- | --- |
| **No.** | **Screen Name** | **Description** |
| 1 | Menu | Header Menu |
| 2 | Header page | Header page |
| 3 | Footer page | Footer page |
| 4 | Pop-up | To be used for notification, warning cases |
| 5 | Tooltip | To be used to show indicators in chart |

* + - 1. Backend

pom.xml

README.md

src/main

-------------java

----------------------------------controller

--------------------------------------------------------rest: chứa các controller được xây dựng theo resfull để public api

----------------------------------service: chứa các service để xử lý business logic

----------------------------------repository: chứa các interface DAO để thao tác với dữ liệu database

----------------------------------entity: chứa các class là các table trong database được tầng repository sử dụng để

thao tác xuống database

----------------------------------dto: là các custom entity để public api ra ngoài

----------------------------------exception: chứa các class để handle exception chung

----------------------------------util

--------------------------------------------------------file: chứa Object ManagerFile.java có các method để thao tác với file

--------------------------------------------------------specification: chứa các object genneral filter, dùng để build ra lệnh

where trong sql

--------------------------------------------------------filter: chứa các object request param filter api

--------------------------------------------------------Paging.java & Sorting.java: Object request param paging và sorting

--------------------------------------------------------Constants.java: chứa các biến constant dùng trong application

--------------------------------------------------------ApiMessage.java: object trả về message của Api

--------------------------------------------------------DataRespone.java: object return của các method api

--------------------------------------------------------MessageUtil.java: chứa các message api public

MethodUtil.java: chứa các method dùng chung trong application

RegularExpressions.java: chứa các Regular Expression dùng để validate trong application

-------------resources

----------------------------------asset/images: chứa image

----------------------------------application.properties: chứa config database, hibernate, security, log

* + - 1. Frontend

angular-cli.json

package.json

README.md

src/

----- app/

---------------- \_core

----------------------------common:chứa các file constants

----------------------------domain:chứa model của chức năng login

----------------------------Fake-data:fake data của user(sử dụng lúc chưa có chức năng login)

----------------------------model:chưa cái model của localStorage

----------------------------service:chưa file service gọi api,file service cấu hinh.

---------------- \_share

----------------------------daterangepiker:chứa các file cấu hình cho directive daterange-piker.

----------------------------dropdown:chứa các file cấu hình cho combobox-mutiple.

----------------------------filter:chứa các file cấu hình dùng cho chức năng filter.

----------------------------sidebar-menu:chứa các file cấu hinh sidebar.

----------------------------table-directives:chứa các file cấu hình cho table-directives

----------------------------top-menu:chứa các file cấu hình cho navbar.

---------------- admin

----------------------------account:chứa component của chức năng quản lý tài khoản.

----------------------------categoryquestion:chứa component của chức năng quản lý danh mục tài khoản.

----------------------------exam:chứa component của chức năng quản lý đề thi.

----------------------------group:chứa component của chức năng quản lý nhóm.

----------------------------home-admin:chứa component trang chủ.

----------------------------question:chứ component của chức năng quản lý câu hỏi

---------------------------admin.component.css|html|spec.ts|ts:là một component.

---------------------------admin.module.ts:khai báo dependencies và các component cho các chức năng với quyền truy cập là admin và manager.

---------------------------admin.routes.ts:khai báo các link điều hướng tới các component.

---------------- user

---------------------------user.component.css|html|spec.ts|ts:là một component.

---------------------------user.module.ts:khai báo dependencies và các component cho các chức năng với quyền truy cập là member và unregistered user.

---------------------------user.routes.ts:khai báo các link điều hướng tới các component.

---------------- app.component.css|html|spec.ts|ts:là một component.

---------------- app.module.ts:khai báo dependencies và các component của ứng dụng.

----- assets/

--------------- css:chứ các file css dùng chung.

--------------- images:chứ các file ảnh.

--------------- js:chứ các file js.

----- environments/

----- environment.prod.ts|ts

----- favicon.ico

----- index.html:điểm bắt đầu của ứng dụng.

----- main.ts:chứ code khởi tạo ứng dụng.

----- polyfills.ts

----- styles.css:cài đặt css cho toàn bộ thư mục.

----- test.ts

----- tsconfig.app.json

----- tsconfig.spec.json

----- typings.d.ts:khai báo các thư viện js.

* 1. Software Modules

Hệ thống Trainning bao gồm các chức năng sau

Table 1: List of software modules

|  |  |  |
| --- | --- | --- |
| **No.** | **Module Name** | **Description** |
|  | Quản lý Tài Khoản | Quản lý tài khoản của User  Chức năng này dành cho Admin |
|  | Quản lý nhóm Tài Khoản | Quản lý Group của User  Chức năng này dành cho Admin |
|  | Quản lý Category Câu hỏi | Quản lý category của câu hỏi  Chức năng này dành cho Admin, Manager  Bao gồm: view list, thêm, sửa, xóa, update |
|  | Quản lý Câu hỏi - Đáp án | Quản lý tất cả câu hỏi  Chức năng này dành cho Admin, Manager  Bao gồm: view list, thêm, sửa, xóa, update |
|  | Quản lý Đề thi | Quản lý đề thi, Manager  Chức năng này dành cho Admin, Manager  Bao gồm: view list, thêm, sửa, xóa, update |

1. [Interface Design]

These are all about APIs between frontend and backend in Dashboard application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **URI** | **Method** | **Params** | **Response** |
| List Menu | /api/v1/menus | GET |  | [  {  "menuId": 1,  "iconCss": "fa-home",  "menuName": "Home",  "menuLink": "/tranning-system/home/index",  "subMenuCollection":[]  },  {  "menuId": 2,  "iconCss": "fa-user",  "menuName": "Account management",  "menuLink": "/tranning-system/account/view",  "subMenuCollection":[]  },  {  "menuId": 3,  "iconCss": "fa-users",  "menuName": "Group management",  "menuLink": "/tranning-system/group/view",  "subMenuCollection":[]  },  {  "menuId": 4,  "iconCss": "fa-bars",  "menuName": "Category question",  "menuLink": "/tranning-system/categoryquestion/view",  "subMenuCollection":[]  },  {  "menuId": 5,  "iconCss": "fa-file",  "menuName": "Exam management",  "menuLink": "/tranning-system/exam/view",  "subMenuCollection":[]  },  {  "menuId": 6,  "iconCss": "fa-question",  "menuName": "Question Answer",  "menuLink": "/tranning-system/question-answer/view",  "subMenuCollection":[]  }  ] |
| List Category Question | /api/v1/questions/categorys | GET | **// paging**  pageNumber: 1,  **// sort**  sortType: “asc”,  sortField: “createDate”,  **// search**  keywordSearch: “Ja”,  **// filter**  creatorId: 1,  createDateFrom: “2018/01/01”,  createDateTo: “2018/04/30”, | {  "pageCurrent": 1,  "totalPage": 1,  "numberRecordPerPage": 1,  "totalElement": null,  "data":[  {  "categoryId": 1,  "categoryName": "Java",  "categoryCode": "JAVA",  "createDate": "2018-01 02T09:50:53.000+0000",  "creatorId":{  "accountId": 1,  "fullname": "Lê Hữu Đoàn"  }  }  ]  } |
| Detail category question | /api/v1/questions/categorys/{id} | GET | id=1 | {  "categoryId": 1,  "categoryName": "Java",  "categoryCode": "JAVA",  "createDate": "2018-01 02T09:50:53.000+0000",  "creatorId":{  "accountId": 1,  "fullname": "Lê Hữu Đoàn"  }  } |
| Create Category Question | /api/v1/questions/categorys | POST | {  "categoryName": "Java",  "creatorId": 5  } | **// success**  {  "categoryId": 35,  "categoryName": "Java",  "categoryCode": "JAVA",  "createDate": "2018-09 04T09:50:53.000+0000",  "creatorId":{  "accountId": 1,  "fullname": "Lê Hữu Đoàn"  }  }  **// fail**  "status": "CONFLICT",  "message": "Name Category exists" |
| Update Category Question | /api/v1/questions/categorys | PUT | {  "categoryId": 1,  "categoryName":  "Java",  "creatorId": 1  } | **// success**  {  "categoryId": 1,  "categoryName": "Java",  "categoryCode": "JAVA",  "createDate": "2018-09 04T09:50:53.000+0000",  "creatorId":{  "accountId": 1,  "fullname": "Lê Hữu Đoàn"  }  }  **// fail**  "status": "CONFLICT",  "message": "Name Category exists" |
| Delete Category Question | /api/v1/questions/categorys | DELETE | id: "1,2,3" | **// success**  "status": "OK",  "message": "Delete successfully"  **// fail**  "status": "UNPROCESSABLE\_ENTITY",  "message": "List id not exist" |
| upLoadImage | /api/v1/files/image | POST | Image: ATTACH FILE IMAGE | **// success**  \src\main\resources\asset\images\*name\_image*.png  **// fail**  "status": "UNPROCESSABLE\_ENTITY",  "message": "Image is must less 3MB" |
| Delete file | /api/v1/files | DELETE | path: "\src\main\resources\asset\images\*name\_image*.png" | **// success**  "status": "OK",  "message": "Delete successfully"  **// fail**  "status": "NOT\_FOUND",  "message": "File not exist" |
| List Question | /api/v1/questions | GET | **//paging**  pageNumber:1,  **//sort**  sortField: “createDate”,  sortType: “desc”,  **// search**  keywordSearch: “la”,  **// filter**  categoryId: 1,  levelId: 2,  typeId: 2,  creatorId: 1,  createDateFrom: “2018/03/01”,  createDateTo: “2018/03/31” | {  "pageCurrent": 1,  "totalPage": 1,  "numberRecordPerPage": 1,  "totalElement": null,  "data":[  {  "questionId": 3,  "content": "Trình biên dịch JIT là gì?",  "fileImage": null,  "createDate": "2018-03-  13T17:00:00.000+0000",  "levelName": "Medium",  "categoryName": "Java",  "typeName": "Multiple-Choice",  "fullname": "Lê Hữu Đoàn",  }  ]  } |
| List Level Question | /api/v1/questions/levelQuestions | GET |  | [  {  "levelId": 1,  "levelName": "Easy"  },  {  "levelId": 2,  "levelName": "Medium"  },  {  "levelId": 3,  "levelName": "Difficult"  }  ] |
| List Type Question | /api/v1/questions/typeQuestions | GET |  | [  {  "typeId": 1,  "typeName": "Essay"  },  {  "typeId": 2,  "typeName": "Multiple-Choice"  }  ] |
| Create Question | /api/v1/questions | POST | {  "levelId": 1,  "typeId": 1,  "categoryId": 5,  "creatorId": 1,  "content": "content của câu hỏi"  } | **// success**  {  "questionId": 16,  "content": "content của câu hỏi",  "fileImage": null,  "createDate": "2018-09 04T09:50:53.000+0000",  "levelId":{  "levelId": 1  },  "categoryId":{  "categoryId": 5  },  "typeId":{  "typeId": 1  },  "creatorId":{  "accountId": 1  },  "answerCollection": null,  "contentSearch": "content của câu hỏi content cua cau hoi"  } |
| Detail Group | api/v1/groups/{id} | GET | Id: 6 | {  "groupId": 6,  "createDate": "2018-01-07T09:50:53.000+0000",  "groupName": "Hội yêu Java",  "creatorId":{  "fullname": "Nguyễn Văn Đông",  },  "numberOfMembers": 2  } |
| List account of group | /api/v1/accounts/group/v2 | GET | Id: 6,  **// paging**  pageNumber: 1,  **// sort**  sortField: ”joinDate”,  sortType: ”desc”,  **// search**  keywordSearch: ”ho”  **// filter**  departmentName: 1,  positionName: 5,  joinDateFrom: ”2018/01/01”,  joinDateTo: ”2018/02/24” | {  "pageCurrent": 1,  "totalPage": 1,  "numberRecordPerPage": 1,  "totalElement": 2,  "data":[  {  "accountId": 16,  "createDate": "2018-01-17T09:50:53.000+0000",  "username": "pxhoang",  "fullname": "Phùng Xuân Hoàng",  "email": "pxhoang@cmc.com.vn",  "departmentName": "RRC",  "departmentId": "1",  "positionName": "PO",  "positionId": "5",  "joinDate": "2018-02-10T09:50:53.000+0000",  "groupId": 6  }  ]  } |
| List exam | /api/v1/exams | GET | **// paging**  pageNumber: 1,  **// sort**  sortField: “createDate”,  sortType: “desc”,  **// search**  keywordSearch: “toan”,  **// filter**  categoryName: 1,  durationValueFrom: 20,  durationValueTo: 40,  numberOfQuestionFrom: 50,  numberOfQuestionTo: 60,  fullname: 1,  statusName: 1,  createDateFrom: “2018/01/01”,  createDateTo: “2018/03/23” | {  "pageCurrent": 1,  "totalPage": 1,  "numberRecordPerPage": 1,  "totalElement": null,  "data":[  {  "examId": 22,  "title": "Java toàn tập",  "categoryName": "Java",  "durationValue": 20,  "numberOfQuestion": 50,  "isAssign": false,  "createDate": "2018-03-03T09:50:53.000+0000",  "fullname": "Lê Hữu Đoàn",  "categoryId": "1",  "creatorId": "1",  "statusId": "1",  "statusName": "Public"  }  ]  } |

1. [Data Design]

Here’s all usecase in Trainning system

|  |  |
| --- | --- |
| Quản lý Tài Khoản | Hiển thị danh sách tài khoản |
| View profile User |
| Quản lý nhóm Tài Khoản | View List nhóm tài khoản |
| Tạo nhóm tài khoản |
| Delete nhóm tài khoản |
| Update nhóm tài khoản |
| View details nhóm tài khoản |
| Quản lý Category Câu hỏi | Tạo category câu hỏi |
| Update category câu hỏi |
| Delete category câu hỏi |
| Quản lý Câu hỏi - Đáp án | View list Câu hỏi |
| Tạo câu hỏi |
| Delete Question |
| Quản lý Đề thi | Hiển thị danh sách đề thi |
| Tạo Đề thi |
| Delete Đề thi |

* + 1. Database Management System Files

This is the entity relationship diagram.

Please refer to <http://115.146.126.146:81/svn/RRC-Training-System/02.Project%20Document/G.API/Diagram>

* + 1. Text-based Database System Files

N/A

1. System Design

Here’s all usecase in Dashboard system

|  |  |
| --- | --- |
| Quản lý Tài Khoản | Hiển thị danh sách tài khoản |
| View profile User |
| Quản lý nhóm Tài Khoản | View List nhóm tài khoản |
| Tạo nhóm tài khoản |
| Delete nhóm tài khoản |
| Update nhóm tài khoản |
| View details nhóm tài khoản |
| Quản lý Category Câu hỏi | Tạo category câu hỏi |
| Update category câu hỏi |
| Delete category câu hỏi |
| Quản lý Câu hỏi - Đáp án | View list Câu hỏi |
| Tạo câu hỏi |
| Delete Question |
| Quản lý Đề thi | Hiển thị danh sách đề thi |
| Tạo Đề thi |
| Delete Đề thi |

System Design includes mockup, sequence diagrams which is located in SVN at the address <http://115.146.126.146:81/svn/RRC-Training-System/02.Project%20Document/C.Design>

1. [User Interface Design]
   1. [Interface Design Rules]

User Interface Design is built from Bootstrap framework. So all conventions and standards follows to what Bootstrap have defined.

* 1. [Common]

Table 3: Common Screens

|  |  |  |
| --- | --- | --- |
| **No.** | **Screen Name** | **Description** |
| 1 | Menu | Header Menu |
| 2 | Header page | Header page |
| 3 | Footer page | Footer page |
| 4 | Pop-up | To be used for notification, warning cases |
| 5 | Tooltip | To be used to show indicators in chart |

* + - 1. [Screen Transition]

The screen transition is discribled in detail design.

Please refer to http://115.146.126.146:81/svn/RRC-Training-System/02.Project%20Document/C.Design

* + - 1. Graphical Interface

N/A

* 1. Application screens

Here is screens list of dashboard application

Table 6: Screens list of <Module 1..n>

|  |  |
| --- | --- |
| **No.** | **Screen Name** |
|  | Hiển thị danh sách tài khoản |
|  | View profile User |
|  | View List nhóm tài khoản |
|  | Tạo nhóm tài khoản |
|  | Delete nhóm tài khoản |
|  | Update nhóm tài khoản |
|  | View details nhóm tài khoản |
|  | Tạo category câu hỏi |
|  | Update category câu hỏi |
|  | Delete category câu hỏi |
|  | View list Câu hỏi |
|  | Tạo câu hỏi |
|  | Delete Question |
|  | Hiển thị danh sách đề thi |
|  | Tạo Đề thi |
|  | Delete Đề thi |

* + - 1. [Screen Transition]

The screen transition is discribled in detail design.

Please refer to http://115.146.126.146:81/svn/RRC-Training-System/02.Project%20Document/C.Design

* + - 1. Graphical Interface
      2. [<Screen 1..n>]

N/A

1. [Others]

[This section is optional. If this section isn’t set, remove it out of the document.]

Purpose:

Provide any other developing areas that are not covered in all sections above]