

**Spring Boot Restful web service  
Mini Project  
University Admission System  
(UAS)**

## Table of Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Setup Checklist for Mini Project	3
1.2	Instructions	3
<b>2</b>	<b>Problem Statement</b>	<b>4</b>
2.1	Objective	4
2.2	Abstract of the project	4
2.3	Functional components of the project	4
2.4	Schema to be used:	5
2.5	Technology used:	6
<b>3</b>	<b>Implementation in JEE LOT</b>	<b>7</b>
3.1	Summary of the functionality to be built:	7
3.1	Evaluation and assessment parameters:	8

# 1 INTRODUCTION

---

This document outlines a mini project for the JEE LOT. The project is to develop an online University Admission System (UAS). This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the JEE LOT.

## 1.1 SETUP CHECKLIST FOR MINI PROJECT

### Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7
- Memory: 32MB of RAM (64MB or more recommended)
- Google Chrome
- MySQL Community Server(6.3)
- JDK 8
- Spring Tool Suit
- JUnit 4.0
- Apache Maven
- GIT Bash
- PostMan

## 1.2 INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.
- Create a directory by your name at [\\DIN690000500\shareFolder\osce\\_training\tracker\\_updates\<name>](\\DIN690000500\shareFolder\osce_training\tracker_updates\<name>). In this directory, create a subdirectory **university\_admission\_system**. Store your Project here.
- You can refer to your course material.
- Since this project work will span over couple of weeks, you will need to take care of maintaining the code

## 2 PROBLEM STATEMENT

---

### 2.1 OBJECTIVE

Development of a University Admission System (UAS)

### 2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing a University Admission System (UAS) for applicants. This system can be used to search a university program for which an applicant wishes to take admission; apply for a selected university program without any login and fill up the details, administration staff can add/update/delete any program that university offers. Members of admission committee have the responsibility to filter the candidates for a specific program on the basis of application data and interview. There are features like report generators etc in this system.

### 2.3 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

There are three categories of people who would access the system viz. applicants, **members of admission committee (mac)** and administrators. Each one of them would have some exclusive privileges (for e.g. Applicants can apply for a program by filling up the form without any login process, members of admission committee alone will be able to view applicant details and filter the applicants for a specific program they have applied for and only the administrator has the right to keep track of the university's program details.)

1. Applicant should be able to
  - View all programs scheduled by the university
  - Apply for a scheduled program of the university, by filling up the application details that auto generates the application ID
  - View the application status, based on the application ID
2. The **member of admission committee (mac)** should be able to :
  - View applications for a specific program.

- Accept/Reject an application on the basis of the details of the applicant. If accepted, fill in the scheduled date for an interview of the applicant before confirming the applicant to take the program.
  - After the interview, update the status of the application to Confirmed/Rejected
3. The administration should be able to
- Update and manage (add or delete) information of the programs offer by the university
  - Manage (add or delete) schedules of the programs offered by the university
  - Generate various reports like:
    - View List of applicants confirmed/ accepted (waiting for interview)/rejected for a scheduled program.
    - View list of programs scheduled to commence in a give time period

Transition of Status: Applied ->Accepted/Rejected->Confirmed/Rejected

## 2.4 SCHEMA TO BE USED:

- i. **Application** : This will contain the list of valid applications

Application\_id (auto generated serial no.),full\_name (varchar2(20)),  
date\_of\_birth (date), highest\_qualification (varchar2(10)),  
marks\_obtained (number), goals(vchar2(20), email\_id(vchar2(20)),  
Scheduled\_program\_id(vchar2(5)),  
status(vchar2(10)),Date\_Of\_Interview(date)

**Note:** Possible values of status are 'applied' by default, 'accepted', 'rejected' and 'confirmed'

- ii. **Programs\_Offered**: This will contain details of programs offered by the university

ProgramName (varchar2(5)), description (varchar2(20)),  
applicant\_eligibility(vchar2(40)) , duration(number),  
degree\_certificate\_offered(vchar2(10))

- iii. **Programs\_Scheduled** : This will contain details of a programs scheduled by the university, among the offered programs

Scheduled\_program\_id (varchar2(5)), ProgramName (varchar2(5)),  
Location (varchar2(10)), start\_date (date), end\_date (date),  
sessions\_per\_week(number)

- iv. **Participant** : This will contain details of confirmed participants for a scheduled program

Roll\_no (varchar2(5)), email\_id (varchar2(20)), Application\_id (FK) ,  
Scheduled\_program\_id(vchar2(5))

- v. **Users** : This will contain all the valid logins and passwords

login\_id(vchar2(5), password(vchar2(10)), role(vchar2(5))

**Notes:** Possible values of role are 'admin' and 'mac'

## 2.5 TECHNOLOGY USED:

1. Use Swagger and Postman as API Management Tools.
2. Develop application with Springboot Restfull webservices with JPA Repositories
3. Configure Lombok, Write Junit test cases using Mockito
4. Bitbucket for Cloud Storage

## 3 IMPLEMENTATION IN JEE LOT

---

### 3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop University Administration System by building the functionality incrementally in each of the course modules of SpringBoot WebServices.

Sr.No	Tasks	Duration (in hours)
1	Project Set-up (Install Java, Jenkins and SONAR Cube in AWS VM. Use free trial or provision through help desk)	24
2	Design APIs using Apiary or Swagger (to follow JSON schema)	12
3	Create Maven multi module based Spring boot project (pls refer template service)	6
4	Develop Java models using JSON schema from apiary or swagger	6
5	Finish the functionality (Use JPA for DB connectivity)	32
6	Create build, deployment and SONAR cube integration pipelines using Jenkins	32
7	Create API end points in Kong	14
8	Configure API Authentication in Kong (using OATH2.0)	18
9	Configure API authorization using Spring security within Springboot	18
10	Create automated Junits	15
11	Create automated integration test cases using Dredd	16
12	Fix SONAR Cube issues - If any	12

### 3.1 EVALUATION AND ASSESSMENT PARAMETERS:

This mini project will be done individual.

#### Evaluation Criteria (out of 100):

Design and Documentation of API's	15
Developing Restful WebServices	15
Building Entity classes and relations	10
Implementing Services and Repositories	10
Implementing Security	20
Devops activity	20
Appropriate test cases using JUnit 4.0, SonarQube	5
Best practices	5