# University Admission System

**Team Members:**

* + - 1. Kalpesh Kandi (Scrum Master)
      2. Vrushali Patil
      3. Piyush Gautam

# 1 Introduction

This document outlines a case study for sprint 2 project. The project is to develop an online university admission system as integration of all independent microservices. 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.

## 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 or higher
* Memory: 4GB of RAM (8GB or more recommended)
* Internet Explorer 6.0 or higher or Chrome 55 or above

**Software System Requirements**

* JDK 1.8
* STS 3.9
* MAVEN
* Apache Tomcat
* Postman Master
* MySQL or H2 Database
* Visual Studio

# Problem Statement

## Objective

Development of an online University Admission System (UAS) that will replace

## Abstract of the project

This project is aimed at developing an online University Admission System (UAS)for applicants. This is a web based application that can be accessed throughout the web. This system can be used to search a university program for which an applicant wishes to take admission; apply online for a selected university program without any login and fill up the form, 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.

The model followed was an agile model. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

## Functional components of the project

There are three categories of people who would access the system viz. applicants, members of admission **c**ommittee (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 online 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** 
   * Browse through the programs offered by the university
   * Register for the programs offered by the university, and will receive an auto generated application ID
   * Check the status of an application, with the application ID provided
2. **Admission Committee(MAC):**
   * Login into the system using his/her credentials.
   * 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. **Admin(Administration):**
   * Login to the system using his/her credentials
   * 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 given time period

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

# Implementation

## Summary of the functionality to be built:

The participants need to develop the Online **UAS** by integrating different micro services for backend and for Angular for frontend, developing overall application with integrating micro services with angular.

## Guidelines on the functionality to be built:

The functionality and components to be built are provided below:

1. Databases to be created:
   1. Create the following database tables:
      1. Application: This will contain the list of valid applications (Applicants fill in the application form)
      2. Programs Offered: This will contain details of programs offered by the university
      3. Programs Scheduled: This will contain details of a programs scheduled by the university, among the offered programs
      4. Participant : This will contain details of confirmed participants for a scheduled program
      5. Users : This will contain all the valid logins and passwords, there is no screen to enter values in this table, assume that this table is present with valid data, that will be used by the application
   2. The structure of the above listed tables is as follows:
      1. Application : Application\_id (auto generated serial no.),full\_name (varchar2(20)), date\_of\_birth (date), highest\_qualification (varchar2(10)), marks\_obtained (number), goals(varchar2(20), email\_ id(varchar2(20)), Scheduled\_program\_id(varchar2(5)), status(varchar2(10)),Date\_Of\_Interview(date)

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

* + 1. Programs\_Offered: ProgramName (varchar2(5)), description (varchar2(20)), applicant\_eligibility(varchar2(40)) , duration(number), degree\_certificate\_offered(varchar2(10))
    2. Programs\_Scheduled : Scheduled\_program\_id (varchar2(5)), ProgramName (varchar2(5)), Location (varchar2(10)), start\_date (date), end\_date (date), sessions\_per\_week(number)

**Notes:**

* Location can be normalized since it would contain further information like city, state, zipcode etc
* No. of seats for a program are not limited

* + 1. Participant : Roll\_no (varchar2(5)), email\_id (varchar2(20)), Application\_id (FK) , Scheduled\_program\_id(varchar2(5))
    2. Users : login\_id(varchar2(5), password(varchar2(10)), role(varchar2(5))

**Notes:** Possible values of role are ‘admin’ and ’mac’

1. OOP & UML
   1. Develop relevant Use case and Class diagrams for the **UAS** application.
2. Angular 6, Web Basics
   1. Develop the following screens/components:
      1. Home page screen: Home page for the online university portal which provides a link for the login page, apply for the programs.
      2. Login Screen: Allows the Admission committee and Admin people to login with valid credentials.
      3. View Programs: For all the applicants, this screen shows a list of programs offered by the university.
      4. Apply Online: For the applicants to apply for the programs.
      5. Add Program: This screen allows the admin to add a program to the university.
      6. Update Program: This screen allows the admin to update existing programs offered by the university
      7. Delete Program: This screen allows the admin to delete existing programs offered by the university
      8. View Reports: This screen offers admin to view various reports by clicking on an appropriate link.
3. JPA with Hibernate + Spring Boot + Microservices:
   * 1. Develop Microservice:
        + 1. Login: There will be one login service in which the role will be decided
          2. Program: This microservice will allow the admin to add new programs.
          3. Applicant: This microservice will allow applicants to apply for program online
     2. Integrate Angular frontend with microservices to complete the entire functionality
4. Documentation **(Duration: 2 hours)**
   1. Project Documentation: Document your project details (Duration: 1 hour 30 min).
   2. Project submission: Submit your project with all the artifacts including the test cases & documentation (Duration: 30 min).