Student Management System (SMS)

Table of Contents

Introduction

You have been tasked by the Institute of Systems Science (ISS) to develop a '**Student Management System' (SMS).** Students of ISS will be able to access this system from ISS intranet web site.

This application is a typical management information system for academic institutions that has administrative functions are always automated. This type of information relies heavily on the data store from which they can create archived reports, profiles, tables and snapshots of data. Through integrating the concept of database management systems, the accuracy, and integrity, of simultaneous and redundant access to these data, are monitored and audited periodically.

Roles

The system will accommodate three types of employee roles, namely: Course Administrators, Lecturers and Students.

- **Students** are enabled to apply/cancel/update their course application.
- **Faculty** teach different courses based on their specialization. Faculty conduct and score exam for assigning courses.
- **Course Administrators** are responsible for leave approval/rejection of course application process. Course Administrators can also print consolidated leave reports. Course Administrators are responsible for creating, managing users and respective roles (Student and Lecturers). They are also responsible for managing the approval hierarchy.

Technical Requirements

This project implementation follows the Spring MVC configuration and setup presented in our lectures. You are free to explore WebSockets and Security additionally to make the application reactive and Agile.

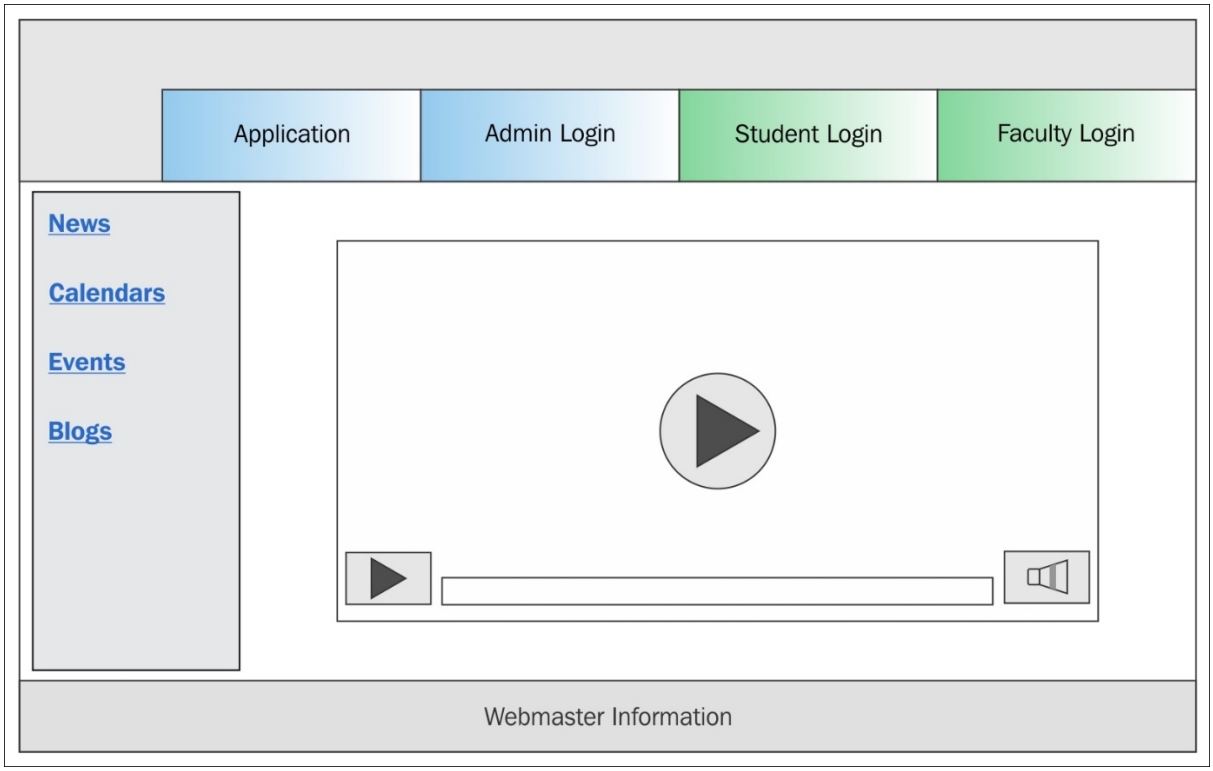
- Use any database system for Spring Boot applications
- Implement Repositories and services layers
- Use Spring Framework abstraction including Data and MVC
- Use Boot Initializer for testing where appropriate.
- Do design appropriate data scripts and also populate the database with sufficient test data.

Design appropriate login and logout pages as appropriate since it involves sensitive information. Three layers to be added into the implementation are the controller, data access layer and the service layer.

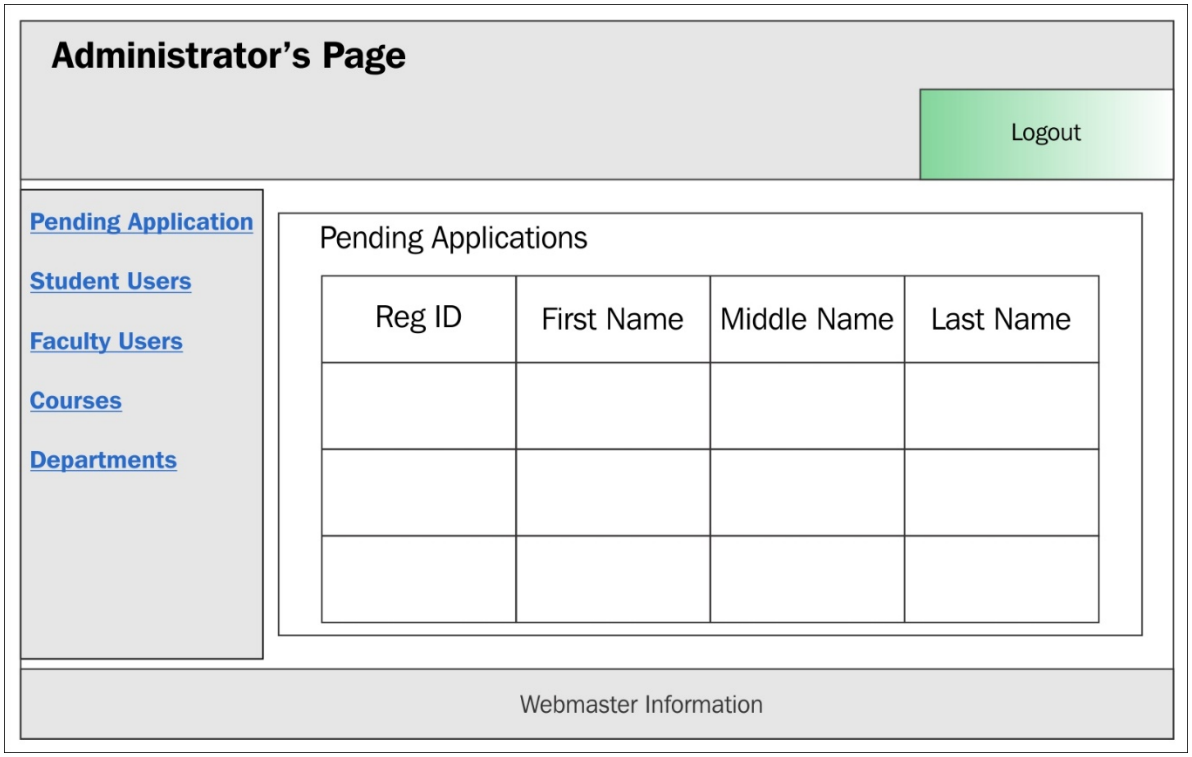
Wire Frames

The SMS application contains the following major pages, which provide some basic functionality needed by core academic software systems:

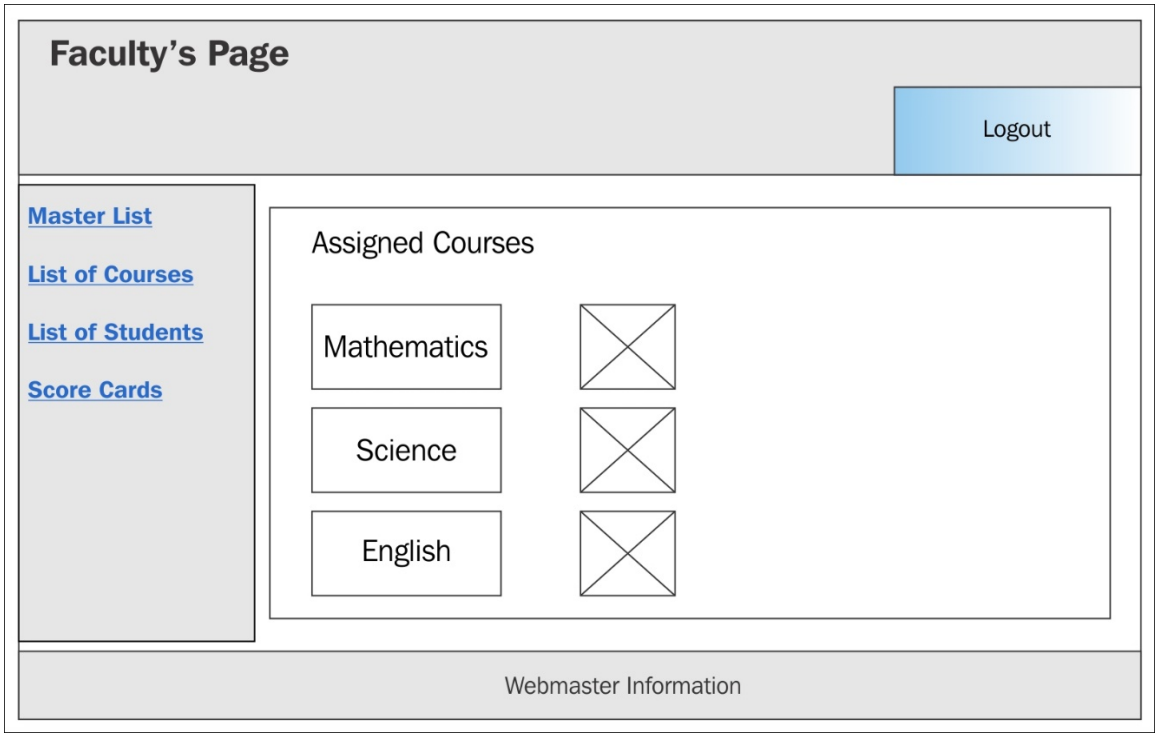
- **The main page**: This wireframe shows the façade of the institution. To attract aspirants and transferees, we plan for this page to contain some marketing concepts, like the presence of some audio-visual images, news scoops, and other campus-related images.



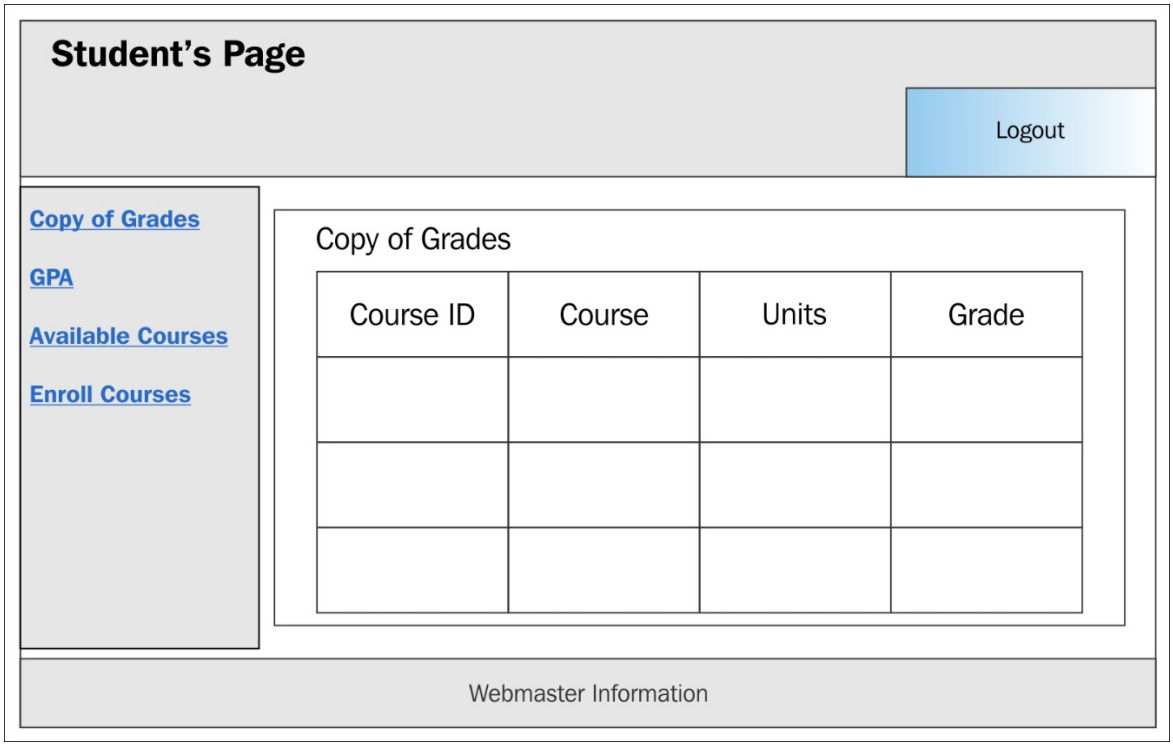
- **The admin page:** This page adds, removes, updates and retrieves all master lists of courses, departments, students, faculty members and applicants. Only administrators can access this page.



- **The faculty page**: This page is to be accessed by any valid faculty member of the institution wherein they can input grades, manage students' performance, and manages their course loads and schedules.



- **The student page**: This wireframe is dedicated to the page that manages student profiles and scholastic information. The student's individual grades, courses they are enrolled in, and the overall courses they took with the GPA, are all the concerns of this area.



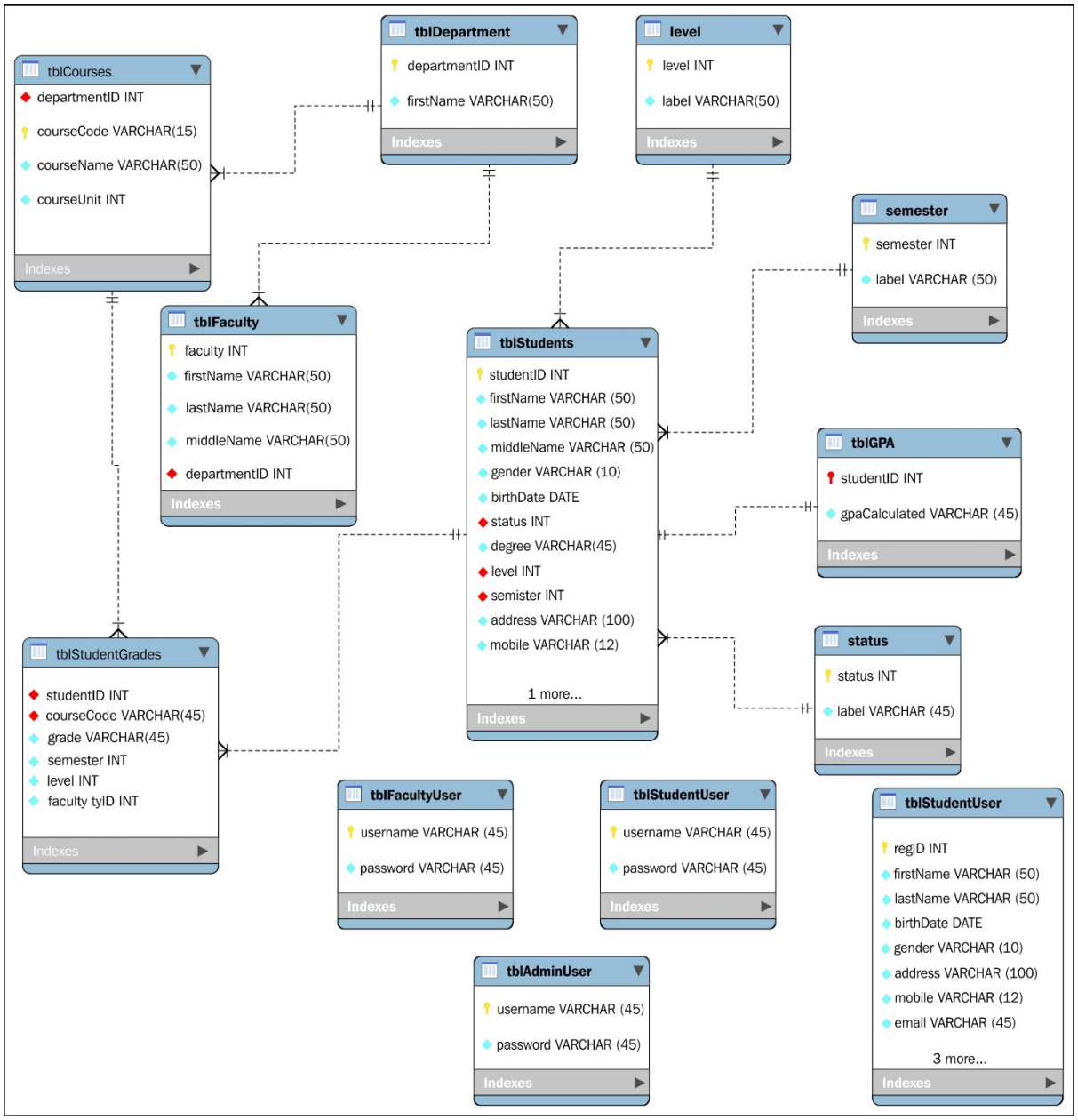
All the sub-links of the above major wireframes are also planned to be web pages showing tables, and form components, for input-output transactions.

Validation Requirements

Choose selected forms and demonstrate the three different forms of Spring Validation in your application. 1. Client Side Validation 2. Validate form using Validator 3. Validate model using Data validation.

The SMS Database

SMS uses the following ERD design, done using MySQL WorkBench. The database server is MySQL Server 8.x. You are free to change the design as appropriate. You are also free to come up with lesser tables that the detailed one as shown here.



Aesthetics

Building an information system using any programming language and frameworks is very tough, since all the details of the requirement must be considered starting with login transactions, content of the pages, functionalities highlighted by the system, security and integrity of the data saved, and retrieved from its data store. Teams are free to make any assumptions convenient for them. Please state the assumptions made upfront while presenting.

Optional Features (for additional marks)

Reporting

Faculty must be able to produce various reporting views such as

- Students List based on Scores for a particular course
- Aggregated CGPA report
- Students List based on Scores for a selected list of courses

Course Administrators must be able to produce various reporting views such as

- Class enrolment for a particular course
- Export the above report to comma-delimited (CSV) file format.

Movement Register

This is a menu available for all users and upon clicking the hyperlink, the system displays details of all students and faculty on leave during the current month. Users can also navigate to the previous and next month using a dropdown choice list.

Pagination

If many results are returned (say more than 30 records), the system should present them over several pages. Each page will contain navigation facilities so that other pages of search results can be shown (similar to search engine results). The number of results per page should ideally be selectable by the user (e.g. 10, 20, 25).

Email interaction

When the faculty wants to send an announcement digitally, he/she should send a notification e-mail to students of a particular course. The email message should contain a direct link to the login page where the user can view the comments.

Security

Secure your application using Spring Container Security

Evaluation Criteria

Everybody contributes and no excuses. The team will be evaluated on the technical quality of the application deliverable. Beyond working solution some expectations would be:

- Implementation best practices; for example, object encapsulation, layering of architecture etc.

- Credits would be allocated for proper exception handling implementations, server validation logic, test cases and utility classes
- Sharing of ideas among team members is always a welcome.

Deliverables

This assignment is part of the continuous assessment for this course. You will be evaluated for 15 marks on the whole. You will work in your team. No individual work will be accepted. The followings are the deliverables.

- A 20-minute presentation, explaining the design and code of your team work.
- A simple 5 slide presentation with the following
    - One Slide explaining how the Team carried work distribution
    - One slide explaining Architecture/Layer Description
    - One slide explaining Class Diagram or ER Diagram
    - One slide explaining Technologies Used
    - One slide summarizing Lessons Learnt
- Your Java project workspace containing all the source code, static and templates files along with any other files (Like data scripts) required to run the application must be uploaded to LumiNus. If the team is using additional plug-in tool or library, it is to be bundled with deliverables.
- You will complete a Peer evaluation form during the exam study period.

Plagiarism Notice: All students share the responsibility for upholding the academic standards and reputation of the University. Academic honesty is a prerequisite condition in the pursuit and acquisition of knowledge. Academic dishonesty is any misrepresentation with the intent to deceive or failure to acknowledge the source or falsification of information or inaccuracy of statements or cheating at examinations/tests or inappropriate use of resources. Please avoid sharing code, you will be penalized if found guilty.

Schedule

FINAL SUBMISSION: 16th December 2019 Monday. Detail schedule will be announced nearer to the presentations.

