

Online Application Title: Student's Online Portal.

Team: Aarti Madre

Sonali Nikam

SRS Document:

Purpose of Project:

This document contains the complete software requirements for Students Online Portal and describe the design decisions, architectural design needed to implement the system. It provides the visibility in design and provide

information needed for software support. Its new, reliable and fast school management which helps for managing daily routines.

Student's Online Portal (SOP) for electronics student portal is intended to provide complete solutions for students as well as management using a single path named Internet. It will allow the faculty to share the resources that are required to students and the students can receive it from the same platform. An administrator can play an important role by making a schedule for the given management and maintain the decorum of the management.

Scope of project:

The system allows the admin to add or remove the daily routine of the management, update feedback system, update faculty details (add or remove a faculty details or add new faculty), update data storage spaces. Whereas the faculty can update assignments, notes, manage attendance, update solutions of queries similarly the student can update his/her assignment, upload queries, etc.

Definition:

SOP-Student's Online Portal.

SRS-Software Requirement Specification.

GUI-Graphical User Interface.

Stack Holder-The person who will participate in the system (Admin, Student, Faculty).

Overview:

This system provides an easy solution for students as well as faculty to communicate with each other in such a pandemic condition. Student can easily get his requirements (like notes, assignments, daily schedules, placement records, etc) on the given portal. Even it will be easier for the faculty to share the student requirements on the same platform in minutes.

This proposed system can be used by students studying in school, colleges, etc and system doesn't require any experienced user in computer field but the user must have basic knowledge of using or handling the computer.

Overall Description:

The Student's Online Portal (SOP) application allows/enables the faculty to share the required resources among the student's in less time whereas the student can gather his study material on the same portal and continue studying.

An admin can update the daily routines of the management and notify the users registered on the portal by sending mails of making and important notification on the portal. Admin has right to add or remove any student or faculty registered to the portal to manage the decorum.

The Student's portal will use the internet as the sole method for selling goods to its consumers.

Functional Requirement:

This section provides requirement overview of the system.

Registration: The faculty as well as student needs to be registered to the portal.

Login: The student and the faculty need to be logged in onto the portal for accessing the resources or updating any resources.

Logout: After accessing the requirement or updating of information the user has to logout from the system.

Non-Functional Requirement:

- Secure access to user's confidential data.

- 24X7 availability.
- Better designing to get best performance.
- Various non-functional requirements are:
 - Security.
 - Reusability.
 - Maintainability.
 - Reliability.
 - Portability.
 - Extensibility.

List of features:

- Admission management.
- Account management.
- Examination management.
- Staff management.
- Placement reports.
- Management Reports.

ER listing:

- Students:
 - PRN
 - First Name
 - Last Name
 - DOB
 - Address
 - Course
 - Course Fees

E-mail

Contact No

- Faculty:

ID

First Name

Last Name

DOB

Subject

Address

Contact No

- Course:

ID

Course Name

- Subject:

Sub Id

Sub Name

Course Id

Faculty Id

- Admin Student

PRN

First Name

Last Name

- Admin Faculty

Faculty Id

First Name

Last Name

- Assignment
 - Assignment No
 - Assignment link
 - Assignment Name
 - Date of submission
 - Subject Id
- Notes
 - Notes id
 - Notes Link
 - Course Id
- Classroom
 - Class Id
 - Classroom Id
 - Classroom Password
 - Course Id
- Placement
 - PRN
 - Student Name
 - Batch
 - Company Name
 - Course Name
- Announcement
 - Announcement Id
 - Announcement Name
- Contact Number
 - Contact1
 - Contact2
 - Contact3
- Schedules
 - dailyId

start_datetime

End_dateTime

Links(zoom link, Youtube link)

Course Id

Subject Name

Application Architecture:

Application = Logic + data

Logic =(UI Logic + Business Logic + DataAccess Logic)

Data =(structured data , Non Structured data)

Online Application:

- Web based Application: deployed on web and accessed by users from anywhere.

[Student's Portal-----Web portal-- used remotely by end users (student, faculty).]

Logic:

- UI Logic:
 - Web Pages + HTML controls + Web Components (Angular)
 - Navigation: (UI Routing) HTML Links, Routing mechanism
 - Data Binding: DOM + JSP tags (JSTL) + `{{}}` ng Model,
- Event Binding: action handler.

HTTP Request: GET:-----Doget

POST:-----Dopost

PUT:

DELETE:

- Client Side UI-----HTML, CSS, JavaScript, bootstrap

UI (Client side UI Framework)

Angular, React, Vue,.....

- Web Logic: (Server side processing)

Server UI----- JSP, servlet, (classical java web technology)

spring MVC (to take advantage of MVC design Pattern using
readymade framework)

Model, View, Controller

Router

(SOA layer)

Spring Boot API

CRUD REST API

ORM Technique: Hibernate (ORM)

JPA

JDBC (database Connectivity)

State management

Client-side state management

cookies, query string, form collection, hidden variables local storage,
session storage, Web SQL, Server side state management session,
Cache, database

- Business Logic: Java console application will be used to test your business Logic

Advanced Java: will contain

- 1.business query processing
- 2.business operation management
- 3.Business data manipulation

From Student's Portal point of view

- **Modules:**

- Registration: Student Registration, Faculty Registration.
- Scheduling: Exam scheduling, course scheduling (zoom meetings)
- Inventory: Course catalog, subject catalog, student catalog, faculty catalog (Insert, Retrieve, Update, Delete)
- Inspection: Notes uploading, Assignment verification, test verification
- Security: Authentication, Authorization (get email Id, get PRN, create password, change password)

- **Data:**

- Structured Data
 - i. RDBMS
 - ii. fields
 - iii. tables
 - iv. constraints
 - v. Not null, auto increment, PK, FK, Unique, check

Add some dummy records in your newly created database.

Write reusable SQL queries against those database tables to check business queries

Test those SQL queries on existing dummy database you built.

- **List of tables:**

- Student:

- Fields: PRN, Student Roll No, FirstName, LastName, email, contact number, course, address, schedules (primary Key: PRN)

- Faculty:

- Fields: Faculty ID, FirstName, LastName, email, contact number, course, address (primary Key: ID)

- Courses:

- Fields: Course_ID, Name, Course code, Subject_Id

- Subjects:

- Fields: Subject_ID, Name, Course code (Foreign Key)

- Admin:

- Fields: Name, Email, Password

- Schedules:

- Fields: s_id, start_time, End_Time, Links(zoom link, Youtube link), date, day, subject_Id, course_Id, Student_Roll_No

- Subject:

- Fields:(Sub Id,Sub Name,Course Id,Faculty Id)

- Admin Student:

- Fields(PRN,First Name,Last Name)

- Admin Faculty:

- Fields:(Faculty Id,First Name,Last Name)

- Assignment:

- Fields:(Assignment No,Assignment link,Assignment Name,Date of submission,Subject Id)

- Notes:

- Fields:(Notes id,Notes Link,Course Id)

- Classroom:
Fields:(Class Id,Classroom Id,Classroom Password,Course Id)
- Placement:
Fields:(PRN,Student Name,Batch,Company Name,Course Name)
- Announcement:
Fields:(Announcement Id,Announcement Name)
- Contact Number:
Fields:(Contact1,Contact2,Contact3)

Entity Relationships:

Courses(1)------(M)dailyschedules

Courses(1)------(M)notes

Courses(1)------(M)classrooms

Courses(1)------(M)subjects

Courses(1)------(M)students

Subjects(1)------(M)faculties

Subjects(1)------(M)assignment

Subjects(1)------(M)test

- Create .sql file
 - ddl.sql--->file will contain DDL commands for Table creation

- dml.sql--->file will contain insert commands for filling dummy data to tables which we have created
- bqquery.sql-->file will contain SQL queries mapped for business queries