Software Requirement Specification (SRS) for

Student Portal for IACSD

# Introduction

### 1.1 Purpose:

Portal is a specially designed website that helps to bring information together from different sources in a uniform way. Student portal for IACSD is a web application which is mainly focused on 3 types of users 1. Student 2. Faculty 3. Admin. The admin will be only one in this project and he/she can add, delete, update student as well as faculty information. Faculty and students in this portal can use different tabs to get information, to update information etc. This application will be very useful for students to get updated to notices.

### 1.2 Scope:

This system is aimed at total user-friendly as well as efficient management of varied tasks. These tasks may range from registering new students, managing students and faculty, sending assignments, displaying assignments status and remark which all are essential features necessary for making the management of IACSD effective.

### 1.3 Definitions:

SPI – Student Portal for IACSD

SRS- Software Requirement Specification

GUI- Graphical User Interface

**1.5 Overview:**

It is a system design especially for IACSD Akurdi for managing affairs related to the students. The student portal for IACSD will provide complete functionality of managing students as well as faculty. In this system, assignment submission facility and noticeboard will also be available.

This student portal can be accessed by any student which has enrolled to PG-DAC course at IACSD and it does not require any prior knowledge to use this portal as this UI is user friendly.

**EXISTING SYSTEM**

* An existing system is highly dependent on other platforms like google for submission and mock-test purpose resp.
* The admin has to add students and faculty manually so this portal will be easier to use.
* In the existing system you cannot provide feedback of the user to the admin online.

**NEED FOR NEW SYSTEM**

**2.Overall Description**:

The website Student portal for IACSD is aimed towards recording a considerable number of student records and needs online assistance for managing records of students. Website is user-friendly, quick to learn and reliable website for the above purpose. Student Information System is intended to be a stand-alone website and it is not depended on the availability of another website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

In this student portal system, there are mainly 3 users:

**1.Admin:**

- Admin can add student and faculty.

- Admin can edit details of student and faculty.

- Admin can also delete details of students as well as faculty.

**2.Faculty:**

- Faculty can add timetable and view timetable.

- Faculty can add noticeboard and view noticeboard.

- Faculty can view student details also.

- Faculty can upload assignments, also check assignments and can give remarks.

**3.Student:**

- Students can see timetable uploaded by faculty.

- Students can see noticeboard uploaded by faculty.

- Students also can see the faculty details.

- Students can upload the assignment given by faculty.

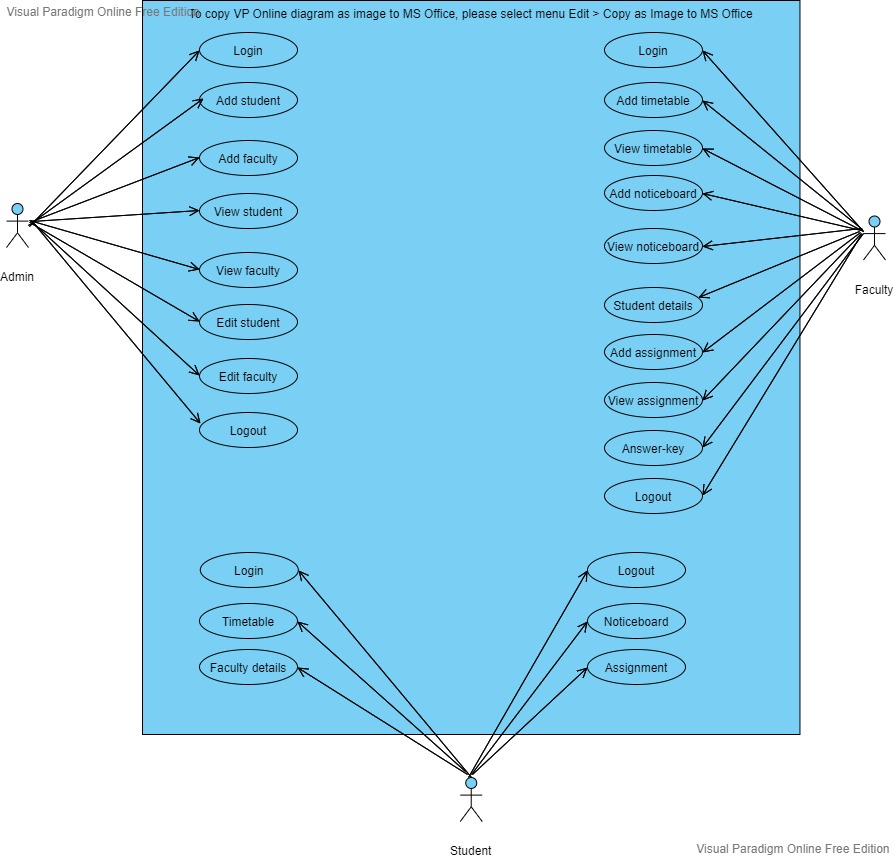
This portal can work on any platform like google, Microsoft edge etc.

Only the constraints are:

Every user must be comfortable using computer.

All operations are in English so user must have basic knowledge of English.

**Use case diagram**



**2.3 User Characteristics**:

User should be familiar with the terms like login, logout etc.

**2.4 Principle Actors**:

3 Principle Actors are student, faculty and admin.

**2.5 General Constraints**:

A good internet connection is required for SPI.

**2.6 Assumptions and Dependencies**:

Working of SPI need Internet Connection and user (student and faculty) must be associated with IACSD.

**3. Specific Requirements**:

3.1 **FUNCTIONAL SPECIFICATION**

User Specification

**Admin:**

Admin can add, update and delete faculty and student.

**Faculty:**

Faculty can upload assignments, update noticeboard, add timetable can give remark to submitted student assignment and view student details.

**Student:**

Student can view timetable, can see uploaded assignment by faculty, remark of assignment added by faculty, notice.

Student can also submit assignments on given link faculty.

**MODULE SPECIFICATION**

**Admin:**

* Add faculty:

Admin is the only person who can add faculty details.

* View faculty:

Admin can see faculty details which are uploaded earlier by admin.

* Add student:

Admin is the only person who can add student details.

* View students:

Admin can see students’ details.

* Edit student:

Admin is the only person who can edit student details.

* Edit faculty:

Admin can edit the details of faculty.

* Delete student:

Admin is the only person who can delete student from list.

* Delete faculty:

Admin can also delete faculty details.

**Faculty:**

* Add timetable:

Only faculty can create students and faculties daily course timetable. And it will be uploaded on daily basis for both students and faculty.

* View timetable:

Faculty can see the timetable in this module.

* Add Notice-board:

If there is any notice regarding guest lecture etc then only faculty can upload information about that notice prior to the time.

* View Notice-board:

If there is any notice regarding exam, guest lecture etc then only admin can upload information about that notice prior to the time.

* Upload assignment:

Faculty can upload assignments for student.

* View assignment:

Faculty can see who all students has submitted the assignments.

* View student details:

Faculty can see details of all registered Students.

**Student:**

* View timetable:

This will be one option on Student menu list in which timetable will be displayed which is uploaded by admin.

* View notice:

If there is any notice regarding guest lecture etc. then student can see notice.

* View faculty details:

Students can see faculty details in this section in case they want to contact faculty member.

* Submit assignment:

Students can submit assignments here which are uploaded by faculty.

**3.2 Non-Functional Requirements**:

Following Non-Functional Requirements will be there in the

insurance to the internet:

(i) Secure access to consumer’s confidential data.

(ii) 24X7 availability.

(iii) Better component design to get better performance at peak

time.

(iv) Flexible service-based architecture will be highly desirable for

future extension. Non-Functional Requirements define system

properties and constraints.

Various other Non-Functional Requirements are:

 Security

 Reliability

 Maintainability

 Portability

 Extensibility

 Reusability

 Compatibility

 Resource Utilization

**3.3 Performance Requirements**:

In order to maintain an acceptable speed at maximum number of uploads from students at a time. Only those can access the website who are part of student or faculty of IACSD. Also, the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

**3.4 Technical Issues**:

This system will work on client-server architecture. It will require an internet service. The system should support some commonly used browser such as chrome, Microsoft edge etc.

**HARDWARE REQUIREMENT**

Hardware requirements for insurance on internet

will be same for both parties which are as follows:

|  |  |
| --- | --- |
| **RAM** | 2 GB |
| **Hard disk** | 320 GB |
| **Processor** | Dual Core |

**Software Requirements**

**Client side:**

|  |  |
| --- | --- |
| **Web Browser** | Google Chrome or any  compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server side:**

|  |  |
| --- | --- |
| **Web Server** | TOMCAT |
| **Server side Language** | JSP |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any  compatible browser |
| **Operating System** | Windows or any equivalent OS |

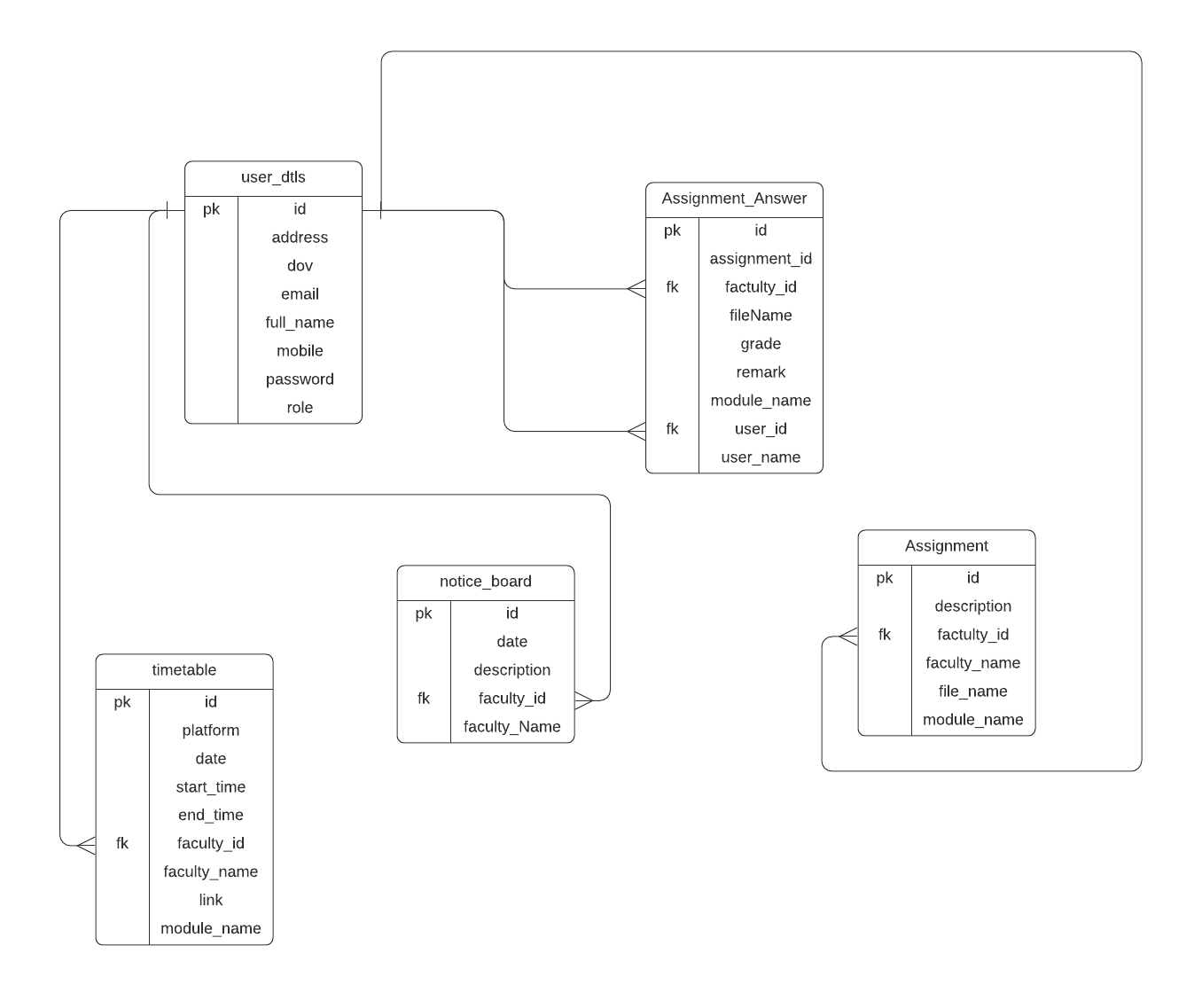
**4.System Design Specification:**

**ER DIAGRAM**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

.



**DATABASE DESIGN**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

**User details:**

|  |  |
| --- | --- |
| **Table Name** | User\_dtls |
|  |  |
| **Description** | This table is providing the information of all the users it may be admin, faculty or student. |
|  |  |
| **Primary Key** | Id |
|  |  |
| **Foreign Key** | - |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Field Name** | **Data type (Size)** | **Constraints** | **Description** |
| **No** |  |  |  |  |
|  |  |  |  |  |
| 1 | id | int (11) | Primary Key | It is store id of each user. |
|  |  |  |  |  |
| 2 | fullName | varchar (255) | Null | It is store user’s full name. |
|  |  |  |  |  |
| 3 | email | varchar (255) | Null | It is store email id of the user |
|  |  |  |  |  |
| 4 | address | Varchar (255) | Null | It stores address of the user. |
|  |  |  |  |  |
| 5 | dob | Varchar (255) | Null | It stores birthdate of the user. |
|  |  |  |  |  |
| 6 | mobileno | Varchar (255) | Null | It is store mobile no of user. |
|  |  |  |  |  |
| 7 | password | Varchar (255) | Null | It is password of user in encrypted format. |
|  |  |  |  |  |
| 8 | role | Varchar (255) | Null | It stores role of that user. |

**Assignment Table:**

|  |  |
| --- | --- |
| **Table Name** | Assignment |
|  |  |
| **Description** | This table is provide the information about assignments given by the faculty |
|  |  |
| **Primary Key** | id |
|  |  |
| **Foreign Key** | - |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | **Field Name** | **Data type (Size)** | **Constraints** | **Description** |
|  |  |  |  |  |
| 1 | id | int (11) | Primary Key | It is store Assignment Id |
|  |  |  |  |  |
| 2 | facultyid | Int (10) |  | It is store faculty id who assigned that assignment |
|  |  |  |  |  |
| 3 | Faculty\_name | varchar (255) |  | It is store Faculty name |
|  |  |  |  |  |
| 4 | Subject\_name | Varchar (255) |  | It is store subject name . |
|  |  |  |  |  |
| 55 | Description | Varchar (255) |  | It is store description about that assignment if faculty has given any. |
| 56 | Assigned\_File | Varchar (255) |  | It is to submit assignment |

**Timetable Table:**

|  |  |
| --- | --- |
| **Table Name** | Time\_table |
|  |  |
| **Description** | This table will store the information of daily timetable |
|  |  |
| **Primary Key** | Id |
|  |  |
| **Foreign Key** | Module\_id |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | **Field Name** | **Data type(Size)** | **Constraints** | **Description** |
| 1. | Id | bigint | Primary key | It stores Id of the timetable. |
| 2. | Faculty\_id | bigint |  | It stores faculty id. |
| 3. | Faculty\_name | varchar (255) |  | It stores the faculty name. |
| 4. | Subject\_name | varchar (255) |  | It stores the subject name. |
| 5. | Start\_time | varchar (255) |  | It stores start time of the lecture/lab. |
| 6. | End\_time | varchar (255) |  | It stores end time of the lecture/lab. |
| 7. | App | varchar (255) |  | It stores the app on which platform faculty is taking lecture. |
| 8. | Date | varchar (255) |  | It stores date for which that particular timetable is assigned. |
| 9. | link | varchar (255) |  | It stores link for the lecture/lab. |

**Noticeboard Table:**

|  |  |
| --- | --- |
| **Table Name** | Notice\_board |
|  |  |
| **Description** | This table will store the information about noticeboard provided by the faculty. |
|  |  |
| **Primary Key** | Id |
|  |  |
| **Foreign Key** | - |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Data type(Size)** | **Constraints** | **Description** |
|  |  |  |  |  |
| 1 | id (Primary) | Bigint (11) | Primary Key | It is store notice id |
|  |  |  |  |  |
| 2 | Faculty\_id | Bigint (10) |  | It is faculty id who has given the notice. |
|  |  |  |  |  |
| 3 | Faculty\_name | Varchar (255) |  | It is faculty name. |
| 4 | date | Varchar (255) |  | It is Date of exam |
| 5 | description | Varchar (255) |  | It gives the notice. |
|  |  |  |  |  |

**Assignment answer Table:**

|  |  |
| --- | --- |
| **Table Name** | Assignment answer |
|  |  |
| **Description** | This table will store the information about the assignment which is submitted by the students. |
|  |  |
| **Primary Key** | id |
|  |  |
| **Foreign Key** | - |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Data type (Size)** | **Constraints** | **Description** |
| 1. | Id | bigint | Primary key | It stores primary key. |
| 2. | Faculty\_id | Bigint |  | It stores faculty id to which assignment will be submitted. |
| 3. | Assignment\_id | Bigint |  | It stores assignment id |
| 4. | Subject\_name | Varchar (255) |  | It stores subject name. |
| 5. | User\_id | Bigint |  | It stores user id means here student id. |
| 6. | User\_name | Varchar (255) |  | It stores user name means student name who submitted the assignment. |
| 7. | Filename | Varchar (255) |  | It stores the filename. |
| 8. | Grade | Varchar (255) |  | It stores grade given by faculty. |
| 9. | remark | Varchar (255) |  | It stores remark. |