**A Project Report On**

**“Online Exam Management System”**

*Submitted by*

**Mistry Mohit Arvindbhai**

*Under the Guidance of*

**Prof. Jinal Tailor**

*In fulfilment for the award of the degree*

*Of*

**Master of Computer Applications**

**(2nd semester)**

In

**Computer Science Department**

****

**S.S. Agrawal Institute of Computer Science**

Opp vidya kunj School, Nr Devina Park Soc, Gandevi Rd, Navsari

Affiliated Under

**Gujarat Technological University, Ahmedabad**

**August 2023**

**S.S. Agrawal Institute of Computer Science**

**Master of Computer Application Department**

**2022**

**CERTIFICATE**

This is to certify that the project entitled **Online Exam Management System** has been carried out by **Mistry Mohit Arvindbhai** (228060694025) under my guidance in fulfilment of the degree of Master of Computer Applications (MCA 2nd Semester) of Gujarat Technological University, Ahmedabad during the academic year 2023.

**Date:**

**Guide Name& Signature: Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Examiners’ Signature:**

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2**.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

3**.**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof Jinal Tailor**

**Head of Department**

**SSAIMT, Navsari**

**Preface**

Knowledge is power. But Power on its own cannot be a key to success. It needs to be harnessed and channelized properly according to the needs and acquirement of the circumstances. In MCA we are supposed to work on a project, which gives us a required exposure to the professional field, which we will be embracing as soon as we move from our student phase to the professional phase.

Today we have many tools at our disposal, so how to use them judiciously according to the requirements of the clients as well as to convince our client to venture out into Uncharted-Red territory and convince them to adapt to emerging technology is the responsibility of the developer.

A Master of Computer Application (MCA) is a two-year full-time course. It covers various topics associated with the area of computer science. We did a project “Online Exam Management System”. An effort has been to exhaustively deal with every part of the systems developed and at the appropriate position, so That a user can easily generate the reports.

**Acknowledgement**

**Mistry Mohit Arvindbhai** a student of MCA (Master of Computer Application) Semester – 2 at S.S Agrawal Institute of Management and Technology, Navsari have completed this project. We would like to convey our heartfelt gratitude to all those people from whom we have got considerable support and encouragement during this project.

We would like to thank our **principal Dr. Sham Sachinwala**, S.S Agrawal Institute of Computer Science for their motivation & constant support to complete this project successfully.

We would like to extend our gratitude to **Prof. Jinal Tailor (Project Guide)** for her valuable guidance and support. Her suggestions and motivation encouraged us throughout the project.

And finally, we would also like to acknowledge the faculty of the Computer Science department for their constant support during the period with them, mounding our future to survive in this highly competitive world.

**Mistry Mohit Arvindbhai**

**Learning during Project Work**

It was a wonderful learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the project. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the developer’s industry.

There are numerous things that we have learned, while working on the project in the IT industry. The project has helped us in perfecting the concepts which we have learned in the past years.

The project gave us a better understanding that there is so much more to learn from the surrounding environments of the colleagues and the power of the technological languages.

The project helped us in organizing the time effectively, and working with teammates and generating substantial output of the efforts. It has prepared us for analyzing and programming for industrial problems and to work on large projects in the future.

Even though we know so much about the technology, still there is so much more to learn from it. The most important thing that we have learned is to never stop learning.

**Abstract**

This project report on “Online Exam Management System”. During the developing of this project, we explore new ideas and functionality. In this project we have basically 3 modules. These three modules include Admin, Faculty and Student.

The admin module profile, add faculty, add students, add Subjects, add course and notification. The faculty module can add subjects and conduct exam and also see their students information and see their student results. The student module profile, students can give exam on exam time and see future exam and students can see their exam marks as well.

This document shows details and various diagrams like use case diagram, class diagram, interaction diagram, activity diagram and also add the data dictionary and screen shorts are define in this document.

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Particulars** | | **Page No**. |
| 1 | Introduction | | 08 |
| 1.1 | Existing System | 09 |
| 1.2 | Need for the New System | 10 |
| 1.3 | Objective of the New System | 10 |
| 1.4 | Problem Definition | 10 |
| 1.5 | Core Components | 11 |
| 1.6 | Project Profile | 13 |
| 1.7 | Advantages and Limitations of the Proposed System | 14 |
| 2 | Requirement Determination & Analysis | | 15 |
|  | 2.1 | Requirement Determination | 16 |
|  | 2.2 | Targeted Users | 18 |
| 3 | System Design | | 19 |
|  | 3.1 | Use Case Diagram | 20 |
|  | 3.2 | Class Diagram | 24 |
|  | 3.3 | Interaction Diagram | 25 |
|  | 3.4 | Activity Diagram | 27 |
|  | 3.5 | Data Dictionary | 32 |
| 4 | Development | | 45 |
|  | 4.1 | Coding Standards | 46 |
|  | 4.2 | Screen Shots | 61 |
| 5 | Proposed Enhancements | | 99 |
| 6 | Conclusion | | 101 |
| 7 | Bibliography | | 103 |

**1. Introduction**

**1.1 Existing System**

* Students around the world face various challenges when it comes to taking exams. They often struggle to find the right study materials and resources, which can lead to frustration and poor academic performance. To address this issue, we have developed an online exam management system.
* Our system is designed to streamline the exam process and eliminate the traditional paper-based methods of conducting exams. By automating the exam management process, our system reduces the workload on administrators and teachers, saving them valuable time and resources. Additionally, by reducing the reliance on paper-based systems, our solution helps institutions reduce their environmental impact.
* Our online exam management system offers a many features, including automatic exam scheduling, grading, and reporting. It also provides access to a vast library of study resources, such as practice exams. With our system, students can easily access all the Exam practices they need to prepare for exams, saving them time and effort.
* Overall, our online exam management system represents a major step forward in modernizing the way students and institutions approach exam management. We believe it will provide significant benefits to students, faculty, and administrators alike, making the entire process more efficient, effective, and convenient.

**1.2 Needs for the New System**

* In this systems student can give exam easily.

**1.3 Objective of the New System**

* This system is developed for Exam.
* All users have access right as per their role.
* The main objective is taking exam to student’s easily.
* Admin can add faculty, student, Subjects and course.
* Faculty can add MCQ according to their course.
* Student can see that Whats up coming exam coming. And when that time near then student can give exam etc…
* Create a database for easy to retrieval, storage and maintenance of student record.

**1.4 Problem Definition**

The Online Exam Management System currently lacks the feature of allowing instructors to upload course materials for students. This presents a number of challenges for both students and faculty, including:

* **Inability to access necessary materials:** Without the ability to upload course materials, students may miss out on important information necessary for success in their courses.
* **Inefficient communication:** Faculty may have to resort to other means of sharing materials, such as email or in-person handouts, which can be time-consuming and inefficient.
* **Limited flexibility:** Students may not be able to access required materials outside of class time, limiting their ability to study and learn at their own pace.
* **Increased workload:** Faculty members may need to spend extra time organizing and distributing materials, which can detract from their ability to focus on other aspects of their teaching.

By adding a feature to allow for easy and efficient uploading of course materials, the Online Exam Management System can help alleviate these issues and create a more streamlined and effective learning experience for all involved.

**1.5 Core Components**

**1.5.1 Scope**

* The proposed software product is the Online Exam Management System.
* The system main purpose of this system Students can give exam easily.
* Managing all records of Admin, Faculty

**1.5.2 Modules**

* There are 3 panel of our system as below: -

1. Admin
2. Faculty
3. Student

Details Functionality of the above listed modules

1. **Admin**

* Admin can Login.
* Admin can Insert Update & Delete Faculty and Student.
* Adding the Subjects based on their course
* Also, admin can update Subjects and Delete..

1. **Faculty**

* Faculty can Login using the email id and password.
* Faculty can add question according to their course
* Faculty can see overview of total question
* Faculty can see overview of total students whose give exam.
* Faculty can see all students marks according to schedule exams
* Faculty can schedule or conduct exam for future event and all students which belong to their course they can get notified on dashboard.

1. **Student**

* Student can Login using the email id and password.
* Student can view their marks.
* Student can join exam when it’s start.
* It’s shows scheduled exam.

**1.6 Project Profile**

**Project Profile:**

|  |  |
| --- | --- |
| Project Name: | **Online Exam Management System** |
| Developers Name: | Mistry Mohit Arvindbhai |
| Team strength: | 1 |
| Front end: | React JS |
| Frame Back end Work: | Laravel(DB:MYSQL) |
| Operating System: | Windows 11 |
| Project Duration: | 4 months |
| Submitted to: | S.S Agrawal Institute of Management and Technology |
| Internal guide: | Prof. Jinal Tailor |
| Software Used: | Visual studio code, Xampp or Wampp, MS Word, Edrawmax, Paint. |
| Head of Departments: | Prof. Jinal Tailor |
|  |  |

**1.7 Advantages and Limitations of the Proposed System**

* Advantages of This System as below: -
* It is very efficient way to taking exam online.
* It save stationery cost as well as it's saved time to faculty also.
* Possible for some time cutting in the Educational Process.
* Free and Easy to Understand.
* Time saving for students.
* If faculty creates all types of question then next year they do not make again and again that question they can simply conduct exam for upcoming students
* Due to its convenience and flexibility, Faculty can conduct exam anytime anywhere.
* Student can’t depend on anyone for practice and preparation for exam.
* Limitations of this systems: -
  + Network problem can affect the system and specially when students give exam.
  + Internet problem.

**2. Requirement Determination & Analysis**

**2.1 Requirement Determination**

**2.1.1 Hardware Requirement**

This phase of the software development process deals with a brief study of different hardware used in the computerized system. This is a list of hardware materials used during the making and the use of the proposed system. As the new system to be made into a computerized functional system required of a computer is must. All the hardware needed here are generally the basic configuration of an office computer.

* **Minimum Hardware Requirement:**

To run the application software of the system in the computer the minimum configuration required is as below:

|  |  |
| --- | --- |
| Processor: | Intel Core i3 |
| RAM: | **4 GB DDR4 or Higher RAM** |
| Hard Disk: | **5 GB Free Space** |
| Monitor: | **1024 DPI or LCD** |
| Keyboard: | **Multimedia or 104 Keys** |
| Mouse: | **Optical or Scroll** |
| Printer: | **Laser Printer or**  **Inkjet printer or**  **Dot Matrix** |
| Backup: | **Git hub Or**  **External Hard Disk Or**  **Pen drive** |
| Motherboard: | **Intel** |

**2.1.2 Software Requirement**

This phase of the software development process deals with a brief study of different Software used in the computerized system. This is a list of Software’s used during the making and the use of the proposed system. As the new system to be made into a computerized functional system required of a computer is must. All the Software needed here are generally the basic configuration of an office computer.

To run the application following minimum software must be required:

|  |  |
| --- | --- |
| Front end: | React JS |
| Back end Framework: | **Laravel (DB: MY SQL)** |
| Web Server: | **Xampp or Wampp** |
| Operating System: | **Windows 10 or Window 11** |
| Photo Editor: | **Photoshop, paint, Edrawmax** |
| Browser: | **Chrome, Firefox** |
| Documents: | **MS Word** |
| Presentation: | **MS Power point** |

**2.2 Targeted Users**

1. **Admin**

* Admin can Login.
* Admin can Insert Update & Delete Faculty and Student.
* Adding the Subjects based on their course
* Also, admin can update Subjects and Delete..

1. **Faculty**

* Faculty can Login using the email id and password.
* Faculty can add question according to their course
* Faculty can see overview of total question
* Faculty can see overview of total students whose give exam.
* Faculty can see all students marks according to schedule exams
* Faculty can schedule or conduct exam for future event and all students which belong to their course they can get notified on dashboard.

1. **Student**

* Student can Login using the email id and password.
* Student can view their marks.
* Student can join exam when it’s start.
* It’s shows scheduled exam.

**3. System Design**

**3.1 Use Case Diagram**

A Use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a “system” is something developed or operated, such as a website. The “actors” are people or entities operating under defined roles within the system.

**Basic use case diagram symbols and Notations:**

**Use case**

* Draw use cases using ovals. Labels the ovals with verbs that represent the system’s functions.

**Actors**

* Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



Actor

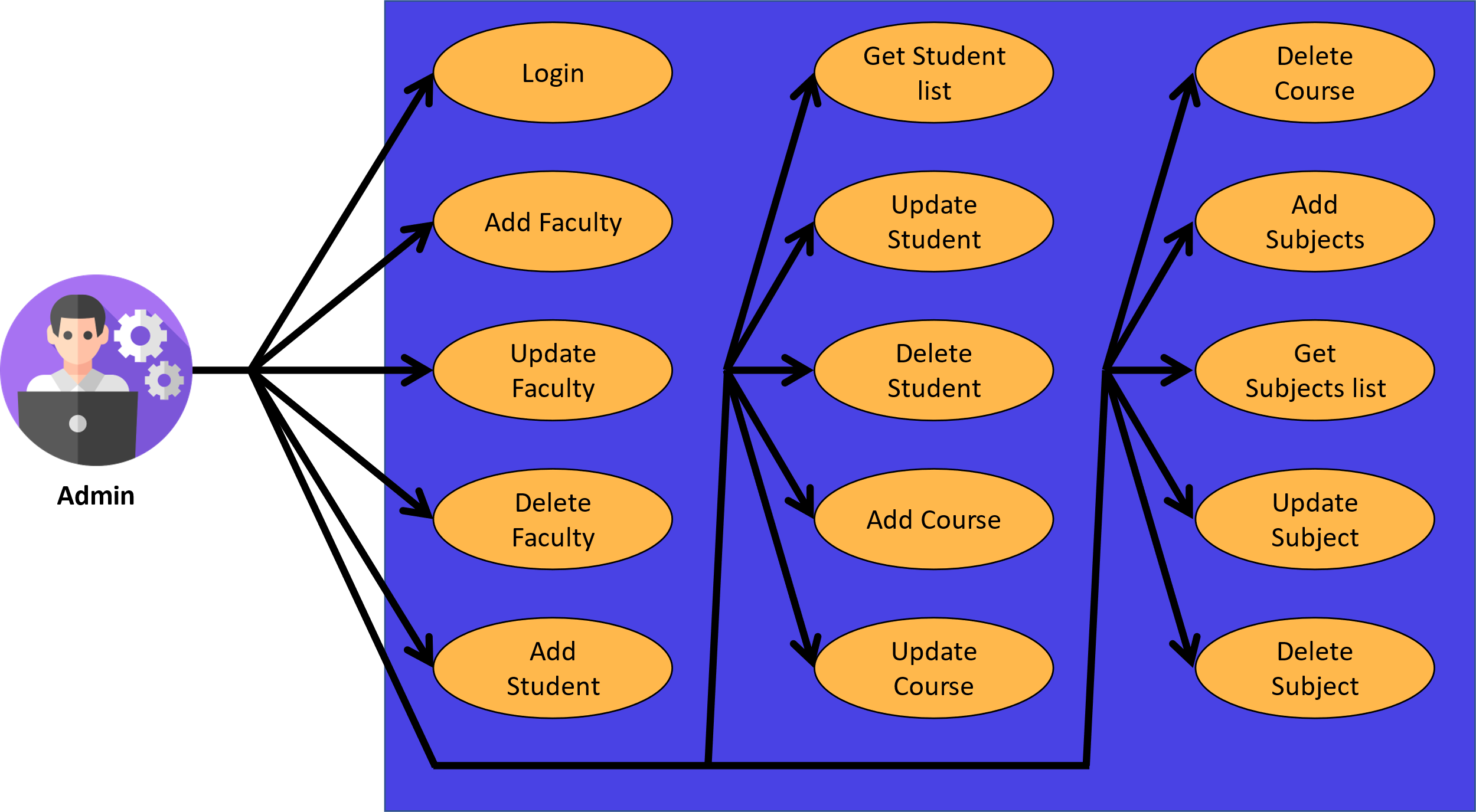
**Relationships**

* Illustrate the relationship between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either “users” or “extends”. A “users” relationship indicates that one use case is needed by another in order to perform a task. An “extends” relationship indicates alternative options under a certain use case.

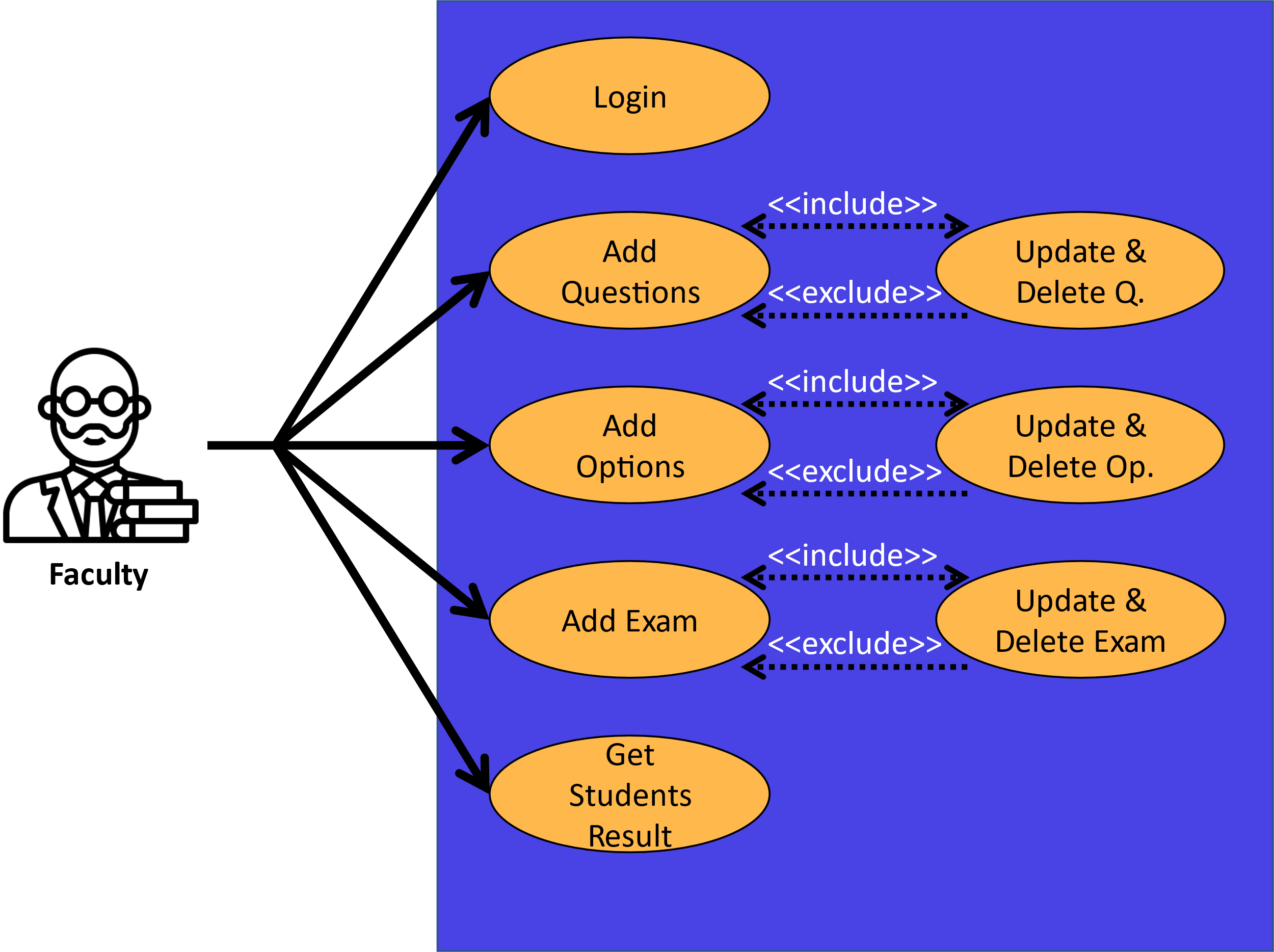


<< Include>> <<Exclude>>

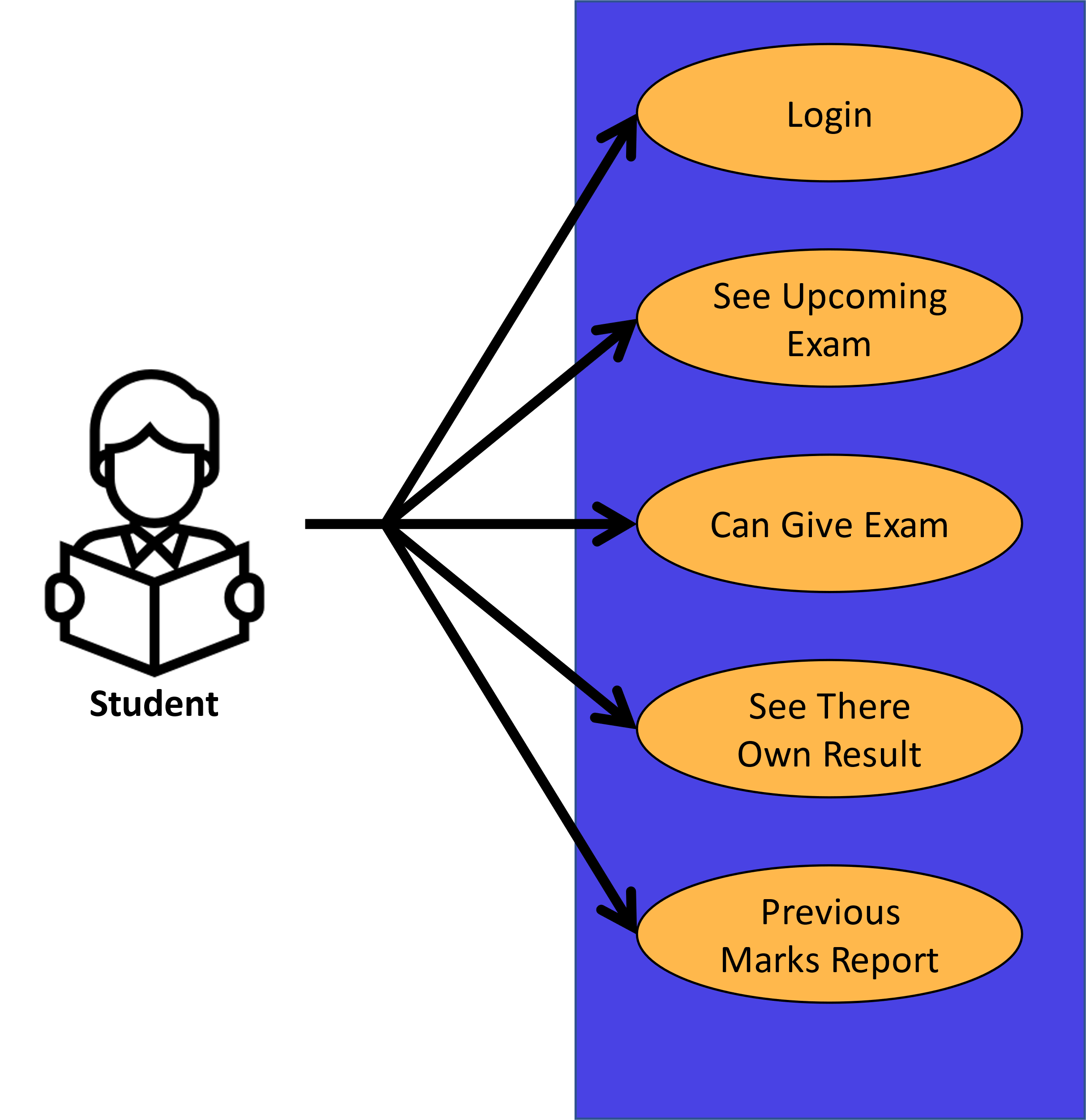
3.1.1 Admin



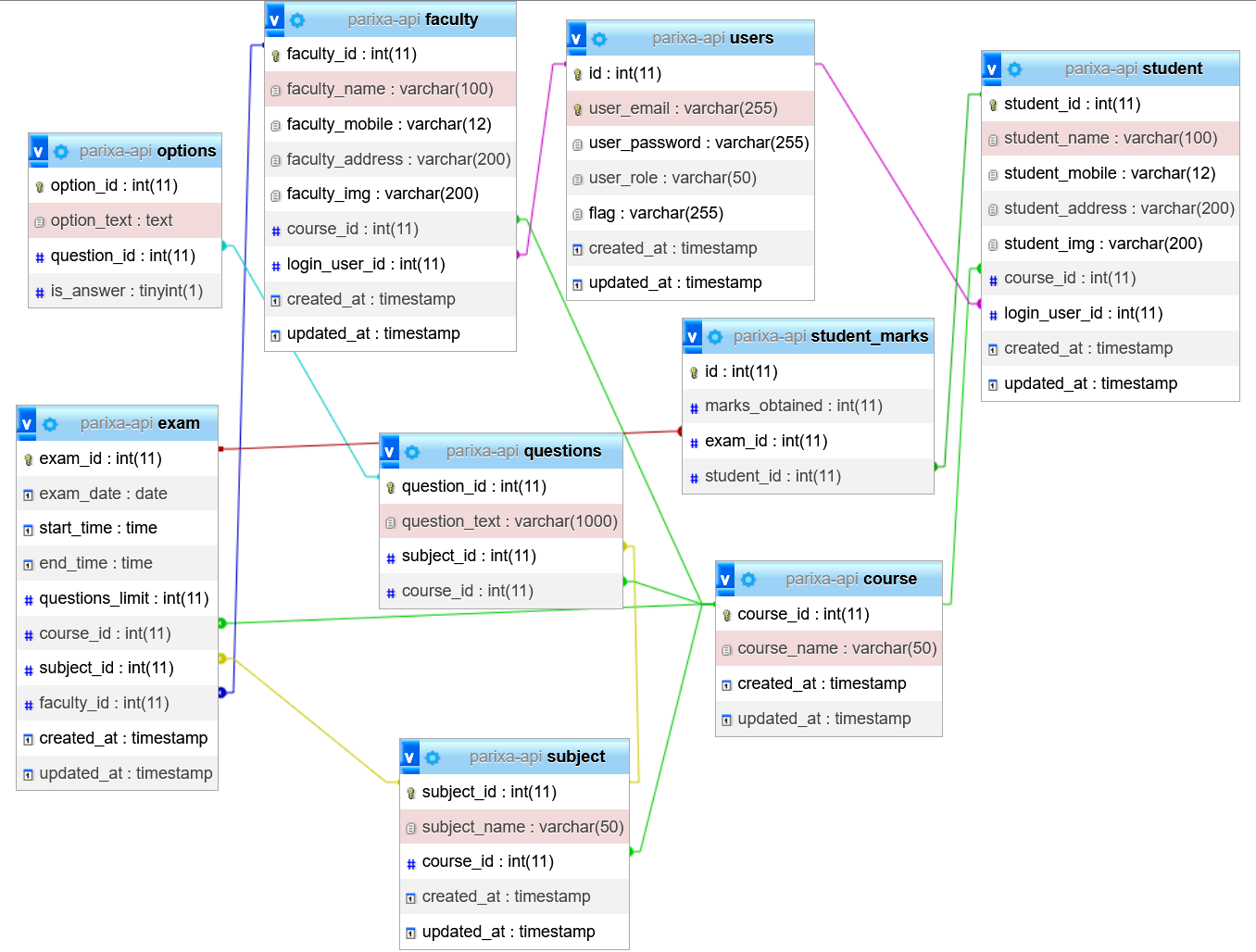
3.1.2 Faculty



3.1.3 Student



**3.2 Class Diagram**

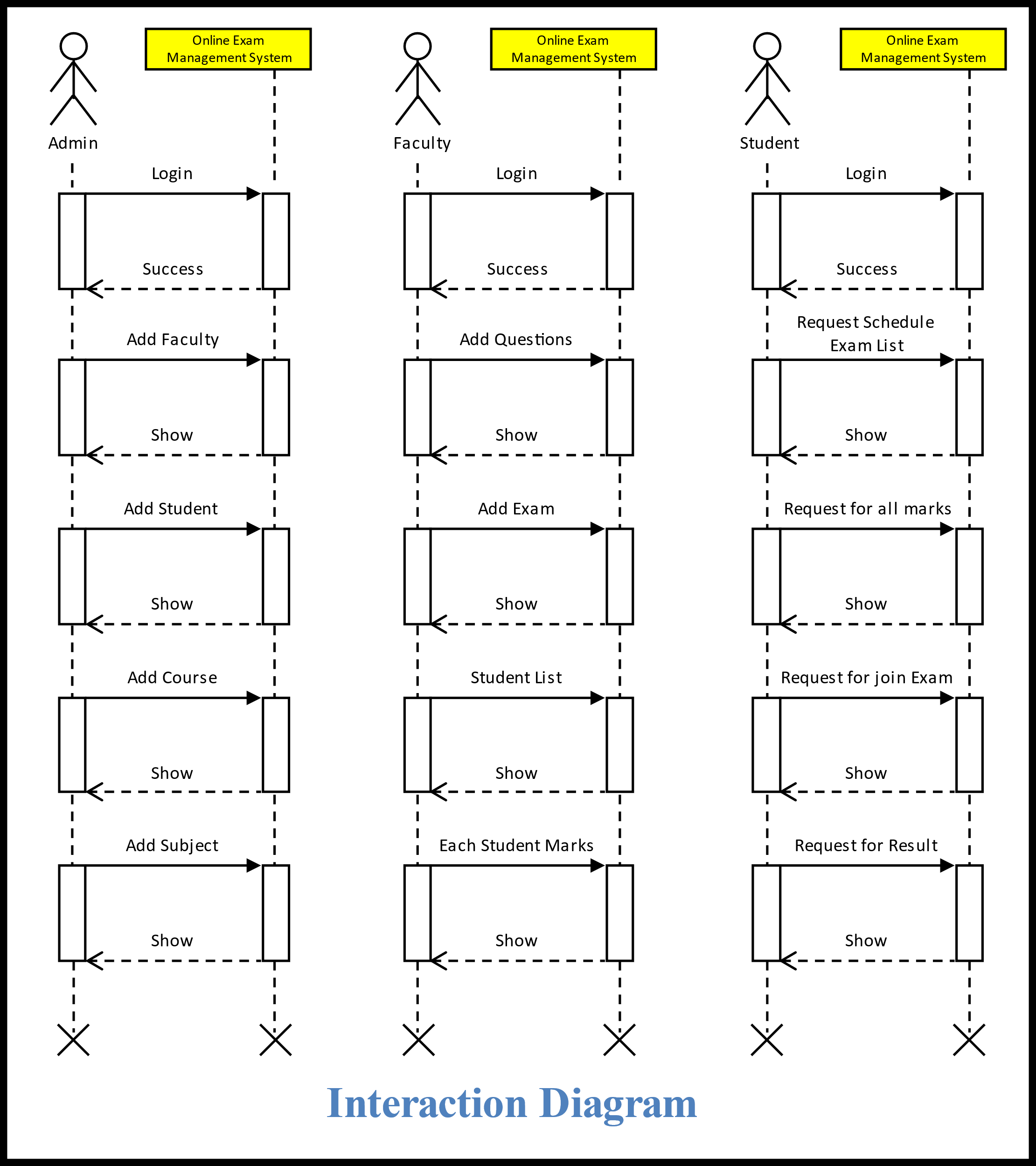


**3.3 Interactive Diagram**

Interaction diagrams are models that describe how a group of objects collaborate in some behaviour- typically a single use-case. The diagrams show a number of example objects and the messages that are passed between these objects within the use-case.

Interaction diagrams come in two forms, both present in the UML. The first form is the sequence diagram. In this form objects are shown as vertical lines with the messages as horizontal lines between them. This form was first popularized by Jacobson.

**As below the Interaction Diagram For This System :-**



**3.4 Activity Diagram**

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

**Basic Activity Diagram Notations and Symbols:-**

**Initial State or Start Point**

* A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column.



**Start Point/Initial State**

**Activity or Action State**

* An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.



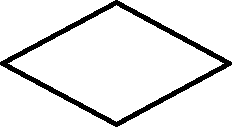
**Activity**

**Action Flow**

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

**Action Flow**

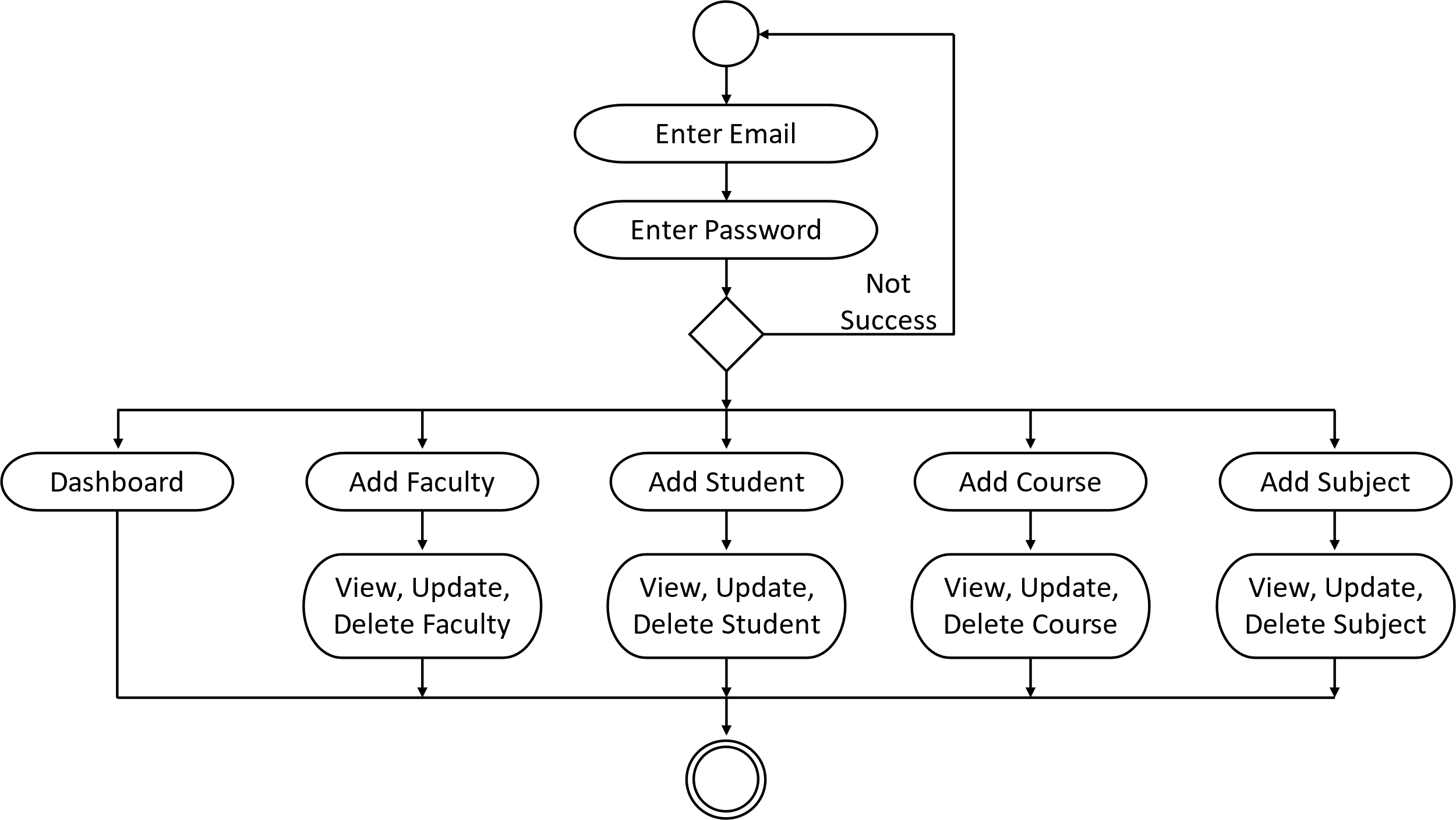
**Decisions and Branching**

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

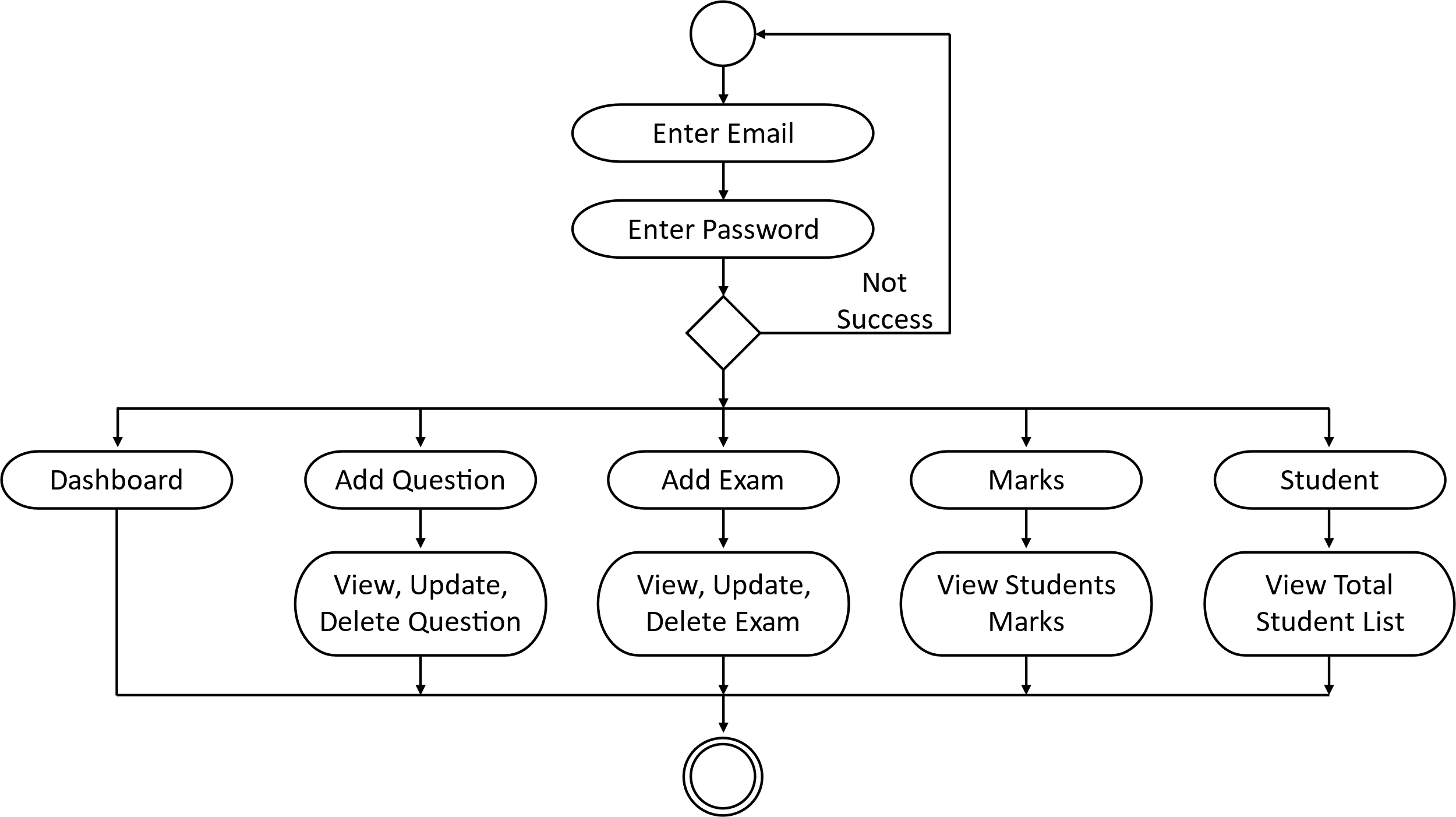
**Decision Symbol**

**As Below the Interaction Diagram for this System: -**

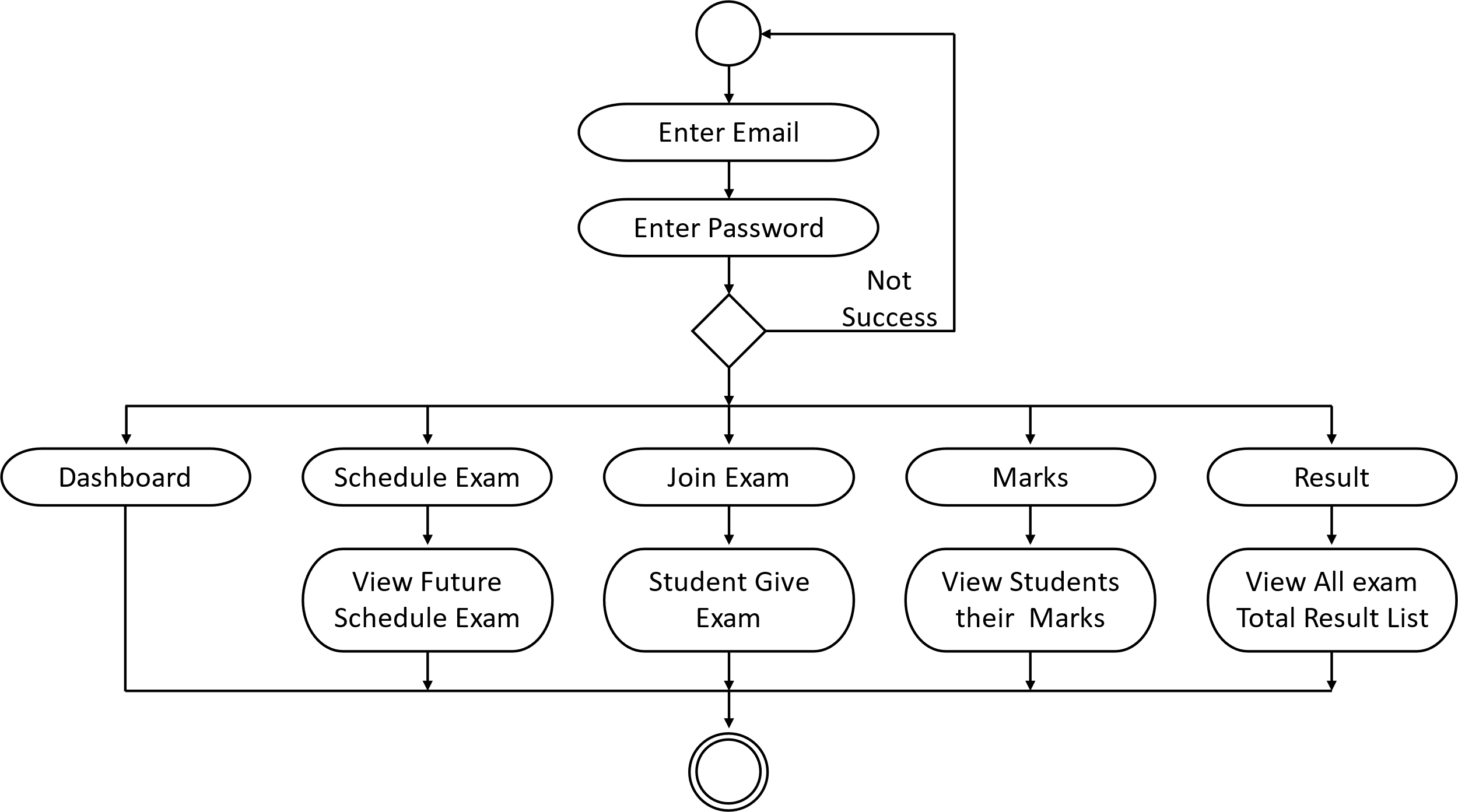
**3.4.1 Admin**



**3.4.2 Faculty**



**3.4.3 Students**



**3.5 Data Dictionary**

* **Table structure: -**

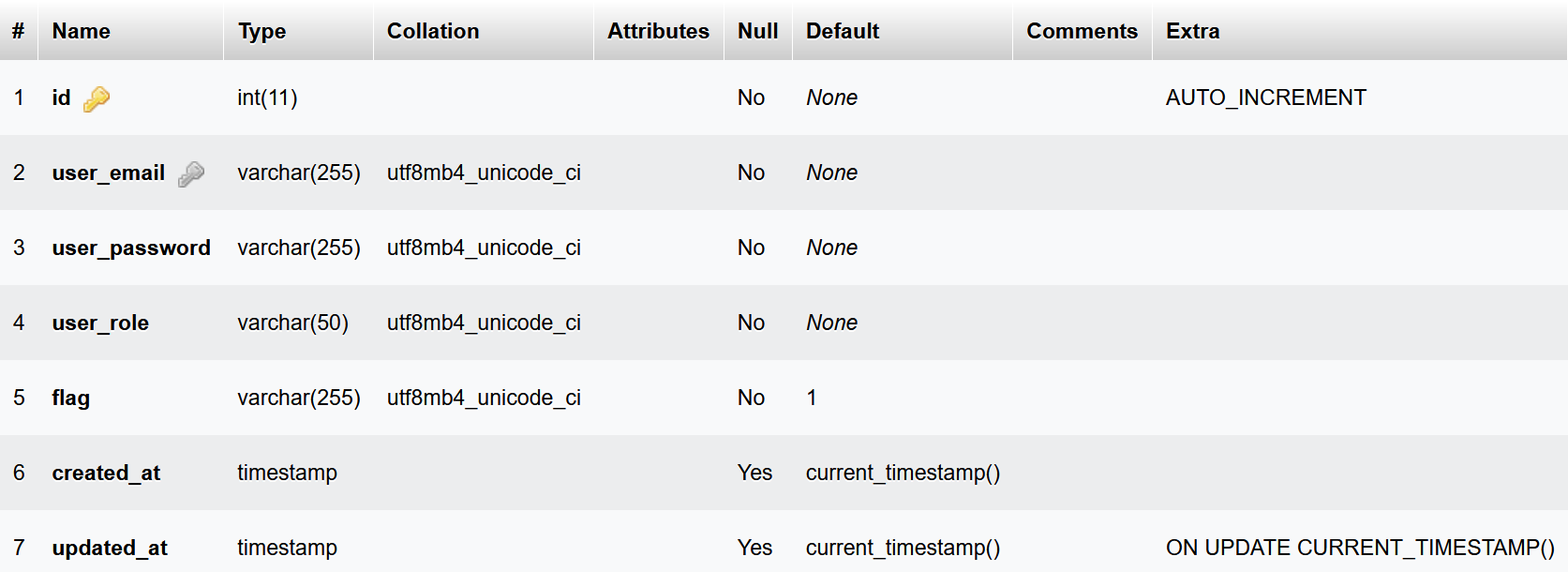
**1. Table name: -** users

**Primary key: -** id

**Foreign key :** user\_email

**Description: -** To store the All users detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | id |  | int(11) | Users id store here. |
| 2. | user\_email |  | varchar (255) | Users store email. |
| 3. | user\_password |  | varchra(255) | Users store password. |
| 4. | user\_role |  | varchar (50) | Users role store data. |
| 5. | flag |  | Varchar (255) | Users Access Rights. |
| 6. | created\_at |  | timestamp | Users create timestamp. |
| 7. | updated\_at |  | timestamp | Users update timestamp. |



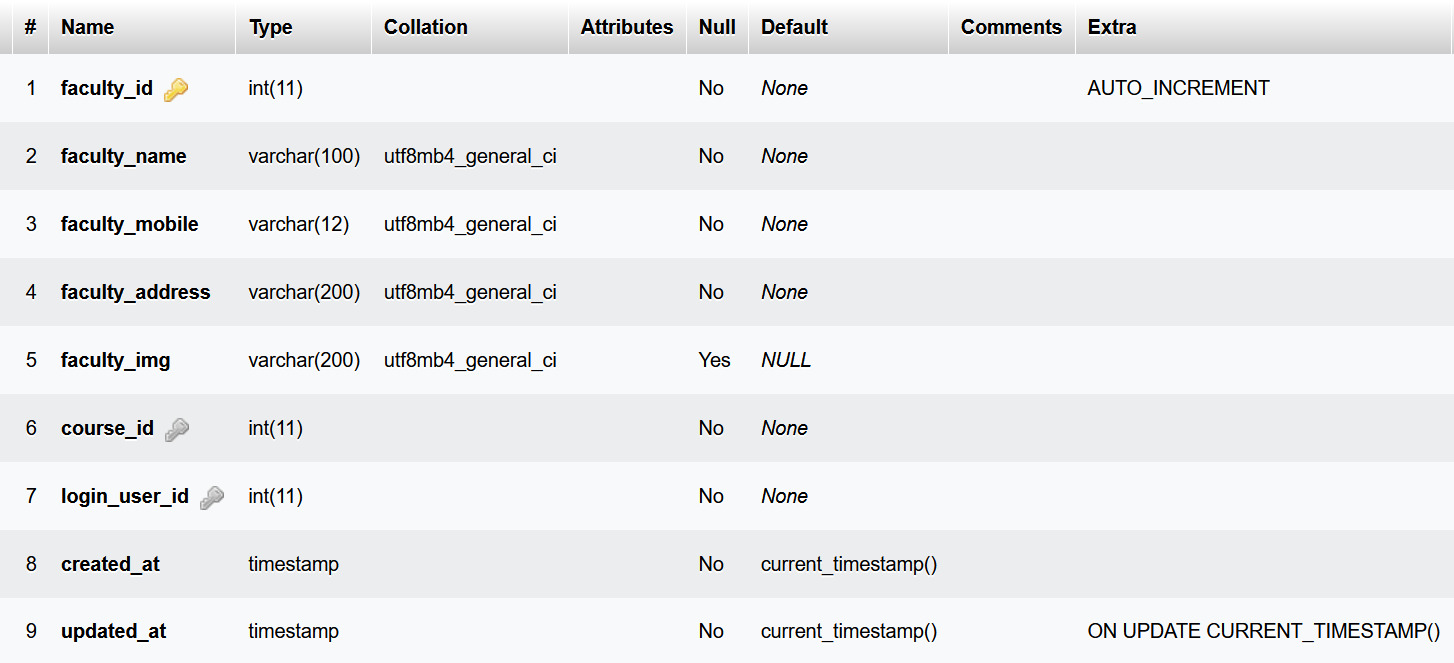
**2. Table name: -** faculty

**Primary key: -** faculty\_id

**Foreign key :** course\_id, login\_user\_id

**Description: -** To store the All Faculty detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | faculty\_id |  | int(11) | Faculty id store here. |
| 2. | faculty\_name |  | varchar (100) | Faculty store Name. |
| 3. | faculty\_mobile |  | varchra(12) | Faculty store Mobile Number. |
| 4. | faculty\_address |  | varchar (200) | Faculty Address store data. |
| 5. | course\_id |  | int(11) | Faculty which belong to course |
| 6. | login\_user\_id |  | int(11) | Faculty Login Id Store here |
| 7. | created\_at |  | timestamp | Faculty create timestamp. |
| 8. | updated\_at |  | timestamp | Faculty update timestamp. |



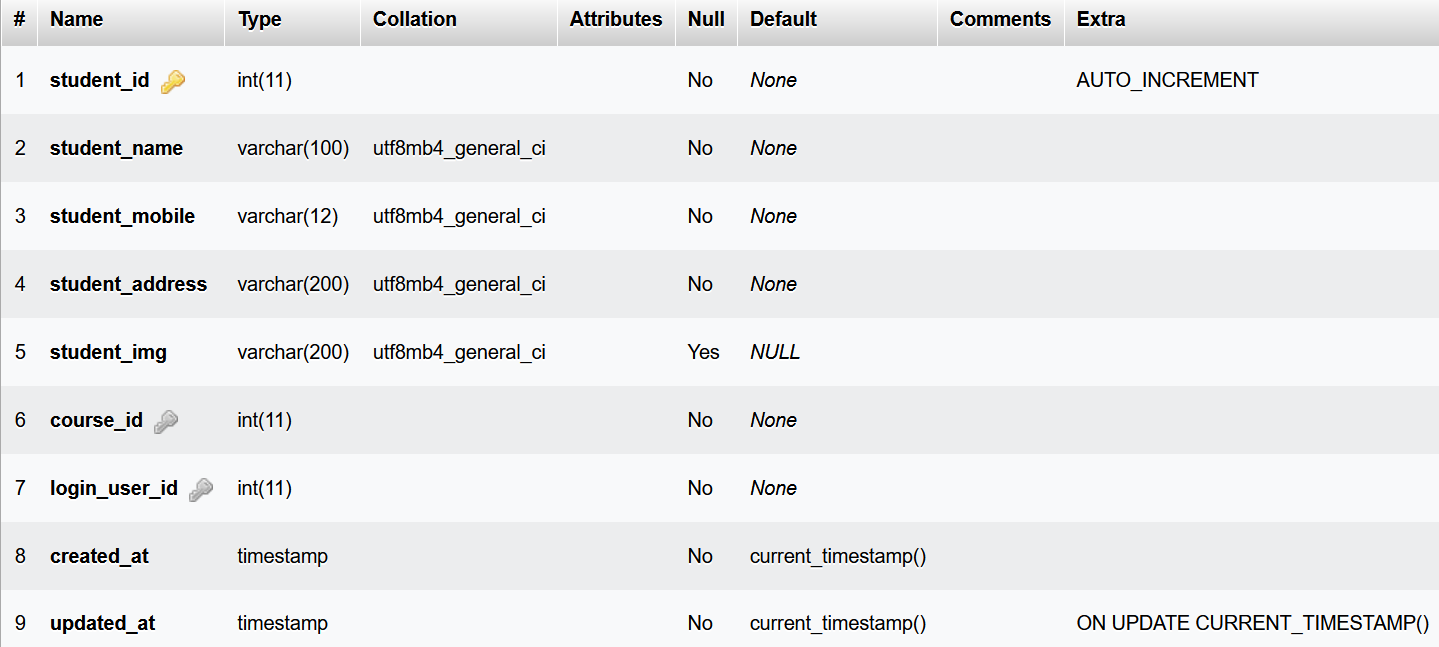
**3. Table name: -** student

**Primary key: -** student\_id

**Foreign key :** course\_id, login\_user\_id

**Description: -** To store the All Student detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | student\_id |  | int(11) | Student id store here. |
| 2. | student\_name |  | varchar (100) | Student store Name. |
| 3. | student\_mobile |  | varchra(12) | Student store Mobile Number. |
| 4. | student\_address |  | varchar (200) | Student Address store data. |
| 5. | course\_id |  | int(11) | Student which belong to course |
| 6. | login\_user\_id |  | int(11) | Student Login Id Store here |
| 7. | created\_at |  | timestamp | Student create timestamp. |
| 8. | updated\_at |  | timestamp | Student update timestamp. |

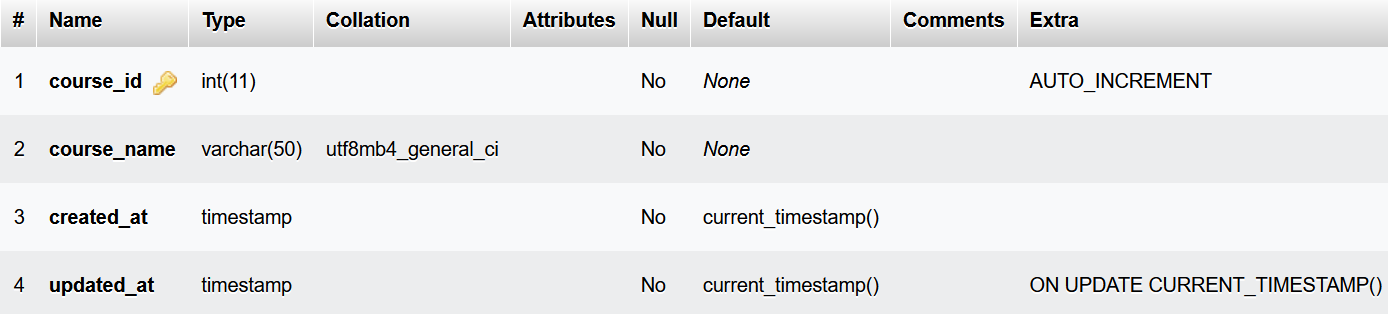


**4. Table name: -** course

**Primary key: -** course\_id

**Description: -** To store the All course detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | course\_id |  | int(11) | Course id store here. |
| 2. | course\_name |  | varchar (50) | Course store Name. |
| 3. | created\_at |  | timestamp | Course create timestamp. |
| 4. | updated\_at |  | timestamp | Course update timestamp. |



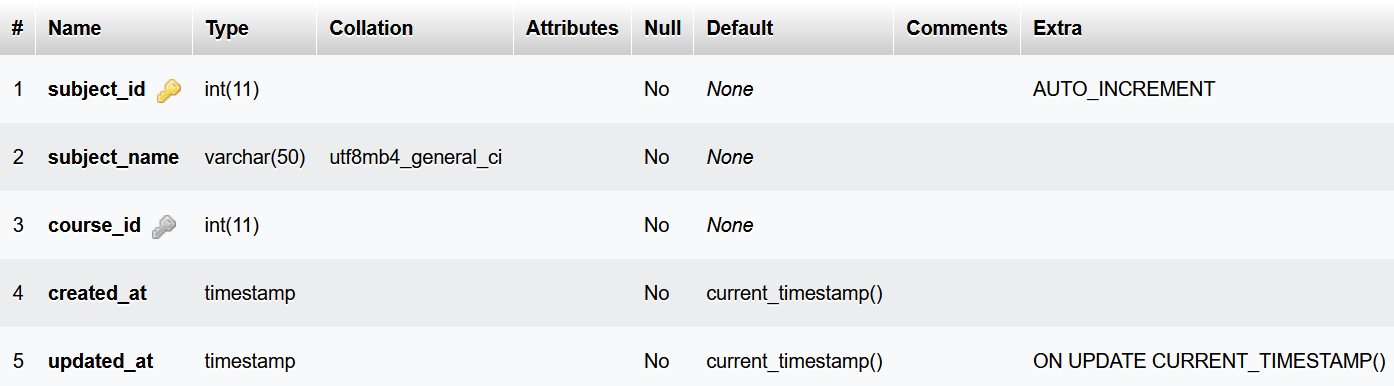
**5. Table name: -** subject

**Primary key: -** subject\_id

**Foreign key:-** course\_id

**Description: -** To store the All subject detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | subject\_id |  | int(11) | Subject id store here. |
| 2. | subject\_name |  | varchar (50) | Subject store Name. |
| 3. | course\_id |  | int(11) | Store course Id |
| 4. | created\_at |  | timestamp | Subject create timestamp. |
| 5. | updated\_at |  | timestamp | Subject update timestamp. |



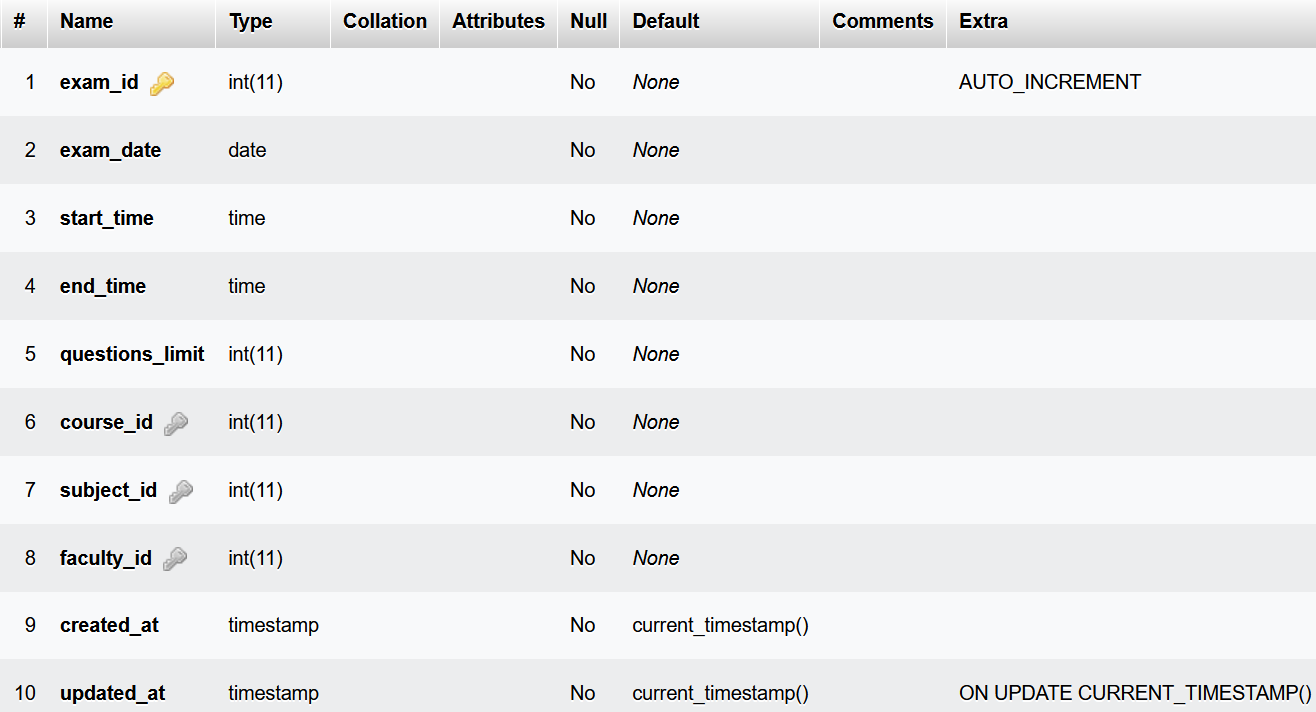
**6. Table name: -** exam

**Primary key: -** exam\_id

**Foreign key:-** course\_id, subject\_id, faculty\_id

**Description: -** To store the All Schedule Exam detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | exam\_id |  | int(11) | Store Exam Id |
| 2. | exam\_date |  | date | Store Exam Date |
| 3. | start\_time |  | time | Store start exam time |
| 4. | end\_time |  | time | Store end exam time |
| 5. | questions\_limit |  | int(11) | Store number of questions |
| 6. | course\_id |  | int(11) | Store course id |
| 7. | subject\_id |  | int(11) | Store subject id |
| 8. | faculty\_id |  | int(11) | Store faculty id |
| 9. | created\_at |  | timestamp | Exam create timestamp |
| 10. | updated\_at |  | timestamp | Exam update timestamp |



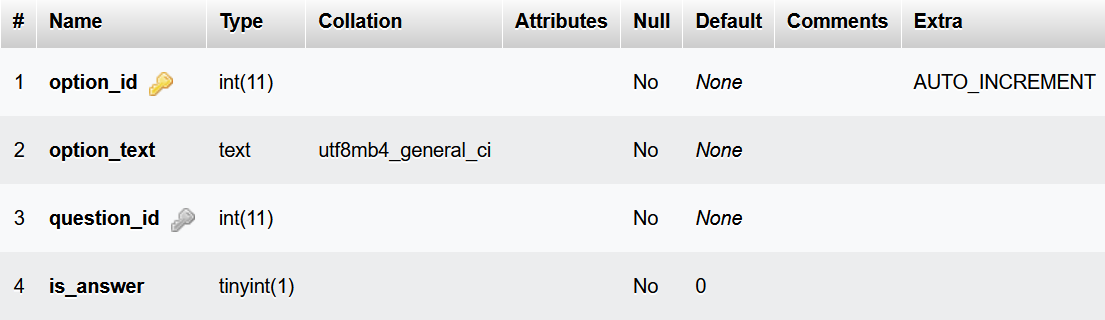
**7. Table name: -** options

**Primary key: -** option\_id

**Foreign key:-** question\_id

**Description: -** To store the All Options of questions detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | option\_id |  | int(11) | Store option Id |
| 2. | option\_text |  | text | Store option Date |
| 3. | question\_id |  | int(11) | Store question id |
| 4. | is\_answer |  | Tinyint(1) | Store answer |



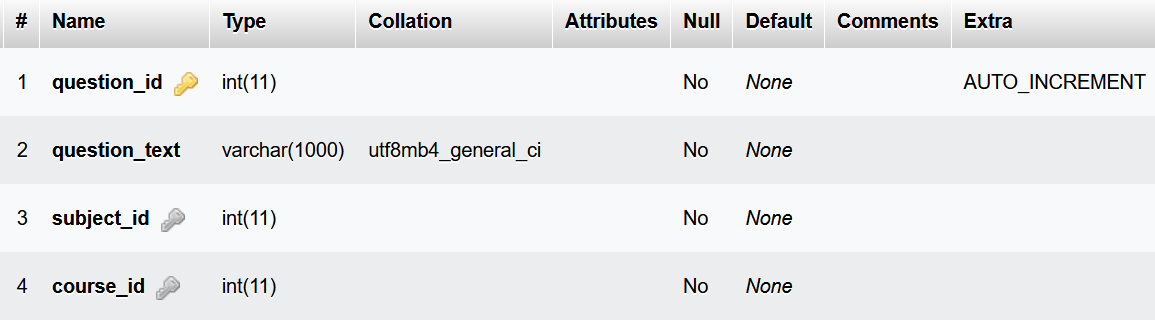
**8. Table name: -** questions

**Primary key: -** question\_id

**Foreign key:-** subject\_id, course\_id

**Description: -** To store the All questions detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | question\_id |  | int(11) | Store question Id |
| 2. | question\_text |  | varchar(1000) | Store question text |
| 3. | subject\_id |  | int(11) | Store subject id |
| 4. | course\_id |  | int(11) | Store course id |



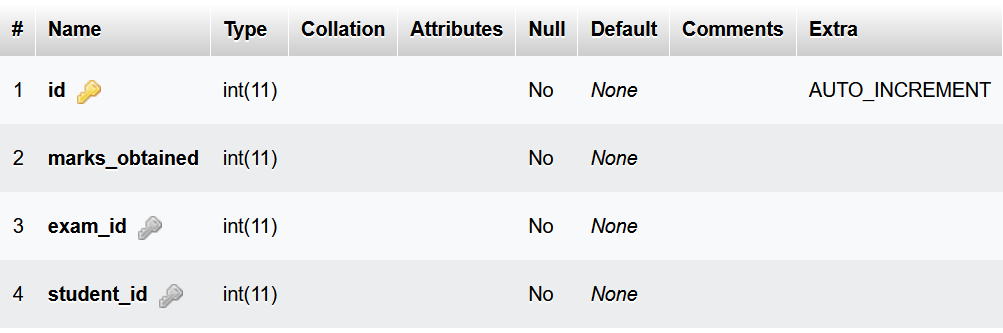
**9. Table name: -** student\_marks

**Primary key: -** id

**Foreign key:-** exam\_id, student\_id

**Description: -** To store the marks of student detail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no. | FEILDS |  | DATA\_TYPE | DESCRIPTION |
| 1. | id |  | int(11) | Store mark Id |
| 2. | marks\_obtained |  | int(11) | Store marks |
| 3. | exam\_id |  | int(11) | Store exam id |
| 4. | student\_id |  | int(11) | Store student id |



**4. Development**

**4.1 Coding Standards**

**Coding Standards**

Different modules specified in the design document are coded in the Coding phase according to the module specification. The main goal of the coding phase is to code from the design document prepared after the design phase through a high-level language and then to unit test this code.

Good software development organizations want their programmers to maintain to some well-defined and standard style of coding called coding standards. They usually make their own coding standards and guidelines depending on what suits their organization best and based on the types of software they develop. It is very important for the programmers to maintain the coding standards otherwise the code will be rejected during code review.

**Purpose of Having Coding Standards:**

* A coding standard gives a uniform appearance to the codes written by different engineers.
* It improves readability, and maintainability of the code and it reduces complexity also.
* It helps in code reuse and helps to detect error easily.
* It promotes sound programming practices and increases efficiency of the programmers.

Some of the coding standards are given below:

1. **Limited use of globals:**  
   **These rules tell about which types of data that can be declared global and the data that can’t be.**

1. **Standard headers for different modules:**  
   **For better understanding and maintenance of the code, the header of different modules should follow some standard format and information. The header format must contain below things that is being used in various companies:**
   * Name of the module
   * Date of module creation
   * Author of the module
   * Modification history
   * Synopsis of the module about what the module does
   * Different functions supported in the module along with their input output parameters
   * Global variables accessed or modified by the module
2. **Naming conventions for local variables, global variables, constants and functions:**  
   **Some of the naming conventions are given below:**
   * Meaningful and understandable variables name helps anyone to understand the reason of using it.
   * Local variables should be named using camel case lettering starting with small letter (e.g. **localData**) whereas Global variables names should start with a capital letter (e.g. **GlobalData**). Constant names should be formed using capital letters only (e.g. **CONSDATA**).
   * It is better to avoid the use of digits in variable names.
   * The names of the function should be written in camel case starting with small letters.
   * The name of the function must describe the reason of using the function clearly and briefly.

1. **Indentation:**  
   **Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use White spaces properly. Some of the spacing conventions are given below:**
   * There must be a space after giving a comma between two function arguments.
   * Each nested block should be properly indented and spaced.
   * Proper Indentation should be there at the beginning and at the end of each block in the program.
   * All braces should start from a new line and the code following the end of braces also start from a new line.
2. **Error return values and exception handling conventions:**  
   **All functions that encountering an error condition should either return a 0 or 1 for simplifying the debugging.**

**Advantages of Coding Guidelines:**

* Coding guidelines increase the efficiency of the software and reduces the development time.
* Coding guidelines help in detecting errors in the early phases, so it helps to reduce the extra cost incurred by the software project.
* If coding guidelines are maintained properly, then the software code increases readability and understandability thus it reduces the complexity of the code.

It reduces the hidden cost for developing the software.

Location : D:\MCAs\Sem 2\Project\Backend\routes\api.php (Backend)

<?php

*use* App\Http\Controllers\Admin;

*use* App\Http\Controllers\Auth\_User;

*use* App\Http\Controllers\Faculty;

*use* App\Http\Controllers\Student;

*use* Illuminate\Http\Request;

*use* Illuminate\Support\Facades\Route;

*/\**

*|--------------------------------------------------------------------------*

*| API Routes*

*|--------------------------------------------------------------------------*

*|*

*| Here is where you can register API routes for your application. These*

*| routes are loaded by the RouteServiceProvider within a group which*

*| is assigned the "api" middleware group. Enjoy building your API!*

*|*

*\*/*

Route*::middleware*('auth:sanctum')*->get*('/user', *function* (Request *$request*) {

*return* *$request->user*();

});

Route*::get*('/', *function* () {

*return* *response*()*->json*([

        'msg' *=>* 'hello world'

    ]);

});

*// used api middleware for session*

Route*::middleware*('api-session')*->group*(

*function* () {

        Route*::prefix*('auth')*->group*(*function* () {

            Route*::post*('/login', [Auth\_User*::class*, "login"]);

            Route*::post*('/signup', [Auth\_User*::class*, "signup"]);

            Route*::post*('/logout', [Auth\_User*::class*, "logout"]);

            Route*::post*('/isUserLogin', [Auth\_User*::class*, "isUserLogin"]);

            Route*::post*('/session', [Auth\_User*::class*, "fetchSessionData"]);

        });

        Route*::prefix*('admin')*->middleware*(['admin-auth'])*->group*(*function* () {

            Route*::post*('/', [Admin*::class*, "dashboard"]);

            Route*::prefix*('student')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Admin*::class*, "fetchStudentData"]);

                Route*::post*('/add', [Admin*::class*, "addStudentData"]);

                Route*::patch*("/update/{id}", [Admin*::class*, "updateStudentData"]);

                Route*::delete*("/delete/{id}", [Admin*::class*, "deleteStudentData"]);

            });

            Route*::prefix*('faculty')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Admin*::class*, "fetchFacultyData"]);

                Route*::post*('/add', [Admin*::class*, "addFacultyData"]);

                Route*::patch*("/update/{id}", [Admin*::class*, "updateFacultyData"]);

                Route*::delete*("/delete/{id}", [Admin*::class*, "deleteFacultyData"]);

            });

            Route*::prefix*('course')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Admin*::class*, "fetchCourseData"]);

                Route*::post*('/add', [Admin*::class*, "addCourseData"]);

                Route*::patch*("/update/{id}", [Admin*::class*, "updateCourseData"]);

                Route*::delete*("/delete/{id}", [Admin*::class*, "deleteCourseData"]);

            });

            Route*::prefix*('subject')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Admin*::class*, "fetchSubjectData"]);

                Route*::post*('/add', [Admin*::class*, "addSubjectData"]);

                Route*::patch*("/update/{id}", [Admin*::class*, "updateSubjectData"]);

                Route*::delete*("/delete/{id}", [Admin*::class*, "deleteSubjectData"]);

            });

        });

        Route*::prefix*('faculty')*->middleware*(['faculty-auth'])*->group*(*function* () {

            Route*::post*('/', [Faculty*::class*, "dashboard"]);

            Route*::prefix*('question')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchQuestionData"]);

                Route*::post*('/add', [Faculty*::class*, "addQuestionData"]);

                Route*::patch*("/update/{id}", [Faculty*::class*, "updateQuestionData"]);

                Route*::delete*("/delete/{id}", [Faculty*::class*, "deleteQuestionData"]);

            });

            Route*::prefix*('course')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchCourseData"]);

            });

            Route*::prefix*('exam')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchScheduleExamData"]);

                Route*::post*('/add', [Faculty*::class*, "addScheduleExamData"]);

                Route*::patch*('/update/{id}', [Faculty*::class*, "updateScheduleExamData"]);

                Route*::delete*('/delete/{id}', [Faculty*::class*, "deleteScheduleExamData"]);

            });

            Route*::prefix*('doneexam')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchDoneExamData"]);

            });

            Route*::prefix*('exammarks')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchExamMarksData"]);

            });

            Route*::prefix*('student')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Faculty*::class*, "fetchStudentData"]);

            });

        });

        Route*::prefix*('student')*->middleware*(['student-auth'])*->group*(*function* () {

            Route*::post*('/', [Student*::class*, "dashboard"]);

            Route*::prefix*('exam')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Student*::class*, "fetchStudentScheduleExamData"]);

            });

            Route*::prefix*('todayexam')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Student*::class*, "fetchStudentTodayExamData"]);

            });

            Route*::prefix*('doneexam')*->group*(*function* () {

                Route*::post*('/fetch/{id?}', [Student*::class*, "fetchStudentDoneExamData"]);

            });

            Route*::prefix*('examquestion')*->group*(*function* () {

                Route*::post*('/fetch/{id}', [Student*::class*, "fetchExamQuestionData"]);

                Route*::post*('/add/', [Student*::class*, "addExamMarksData"]);

            });

        });

    }

);

Frontend Routes

D:\MCAs\Sem 2\Project\Frontend\src\Views\Admin\SidebarMenuOptions.js (For Admin Route)

*import* { *lazy* } *from* "react";

*import* { *FontAwesomeIcon* } *from* "@fortawesome/react-fontawesome";

*import* { *Route* } *from* "react-router-dom";

*//*

*import* *Dashboard* *from* "./Screens/Dashboard";

*import* *Students* *from* "./Screens/Students";

*import* *Faculty* *from* "./Screens/Faculty";

*import* *Course* *from* "./Screens/Course";

*import* *Subject* *from* "./Screens/Subject";

*const AllRoutes = [*

*{*

*label:* "Dashboard"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "table"*]} color={*"white"*} />,*

*to:* "/admin"*,*

*route:* "/"*,*

*componet: <*Dashboard */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Course"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "book-reader"*]} color={*"white"*} />*

*),*

*to:* "/admin/course"*,*

*route:* "/course"*,*

*componet: <*Course */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Subject"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "book"*]} color={*"white"*} />,*

*to:* "/admin/Subject"*,*

*route:* "/Subject"*,*

*componet: <*Subject */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Student"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "users"*]} color={*"white"*} />,*

*to:* "/admin/students"*,*

*route:* "/students"*,*

*componet: <*Students */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Faculty"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "chalkboard-teacher"*]} color={*"white"*} />*

*),*

*to:* "/admin/facultys"*,*

*route:* "/facultys"*,*

*componet: <*Faculty */>,*

*isSideMenuComponent: true,*

*},*

*]*;

*const SidebarMenuContent = AllRoutes.filter((*ls*) => {*

*if (ls.isSideMenuComponent) {*

*return* ls*;*

*}*

*})*;

*const getRoutes = () =>*

*AllRoutes.map((*route*,* i*) => {*

*return <*Route *exact path={route.route} element={route.componet} key={*i*} />;*

*})*;

*export* { *SidebarMenuContent*, *getRoutes* };

D:\MCAs\Sem 2\Project\Frontend\src\Views\Faculty\SidebarMenuOptions.js

*import* { *lazy* } *from* "react";

*import* { *FontAwesomeIcon* } *from* "@fortawesome/react-fontawesome";

*import* { *Route* } *from* "react-router-dom";

*//*

*import* *Dashboard* *from* "./Screens/Dashboard";

*import* *Students* *from* "./Screens/Students";

*import* *Faculty* *from* "./Screens/Faculty";

*import* *QuestionsScreen* *from* "./Screens/Questions";

*import* *DoneExam* *from* "./Screens/DoneExam";

*import* *ScheduleExamScreen* *from* "./Screens/ScheduleExam";

*// import Approval from "./Screens/Approval";*

*// import Profile from "./Screens/Profile";*

*const AllRoutes = [*

*{*

*label:* "Dashboard"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "table"*]} color={*"white"*} />,*

*to:* "/faculty"*,*

*route:* "/"*,*

*componet: <*Dashboard */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "questions"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "book-reader"*]} color={*"white"*} />*

*),*

*to:* "/faculty/questions"*,*

*route:* "/questions"*,*

*componet: <*QuestionsScreen */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Schedule Exam"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "book"*]} color={*"white"*} />,*

*to:* "/faculty/exam"*,*

*route:* "/exam"*,*

*componet: <*ScheduleExamScreen */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Student"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "users"*]} color={*"white"*} />,*

*to:* "/faculty/students"*,*

*route:* "/students"*,*

*componet: <*Students */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Done Exam"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "calendar-check"*]} color={*"white"*} />*

*),*

*to:* "/faculty/doneexam"*,*

*route:* "/doneexam"*,*

*componet: <*DoneExam */>,*

*isSideMenuComponent: true,*

*},*

*]*;

*const SidebarMenuContent = AllRoutes.filter((*ls*) => {*

*if (ls.isSideMenuComponent) {*

*return* ls*;*

*}*

*})*;

*const getRoutes = () =>*

*AllRoutes.map((*route*,* i*) => {*

*return <*Route *exact path={route.route} element={route.componet} key={*i*} />;*

*})*;

*export* { *SidebarMenuContent*, *getRoutes* };

D:\MCAs\Sem 2\Project\Frontend\src\Views\Student\SidebarMenuOptions.js

*import* { *lazy* } *from* "react";

*import* { *FontAwesomeIcon* } *from* "@fortawesome/react-fontawesome";

*import* { *Route* } *from* "react-router-dom";

*//*

*import* *Dashboard* *from* "./Screens/Dashboard";

*import* *Exam* *from* "./Screens/Exam";

*import* *ScheduleExamScreen* *from* "./Screens/ScheduleExam";

*import* *DoneExamScreen* *from* "./Screens/DoneExam";

*const AllRoutes = [*

*{*

*label:* "Dashboard"*,*

*icon: () => <*FontAwesomeIcon *icon={[*"fa"*,* "table"*]} color={*"white"*} />,*

*to:* "/student"*,*

*route:* "/"*,*

*componet: <*Dashboard */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Today's Exam"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "book-reader"*]} color={*"white"*} />*

*),*

*to:* "/student/exam"*,*

*route:* "/exam"*,*

*componet: <*Exam */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Shedule Exam"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "clipboard-list"*]} color={*"white"*} />*

*),*

*to:* "/student/sheduleexam"*,*

*route:* "/sheduleexam"*,*

*componet: <*ScheduleExamScreen */>,*

*isSideMenuComponent: true,*

*},*

*{*

*label:* "Done Exam"*,*

*icon: () => (*

*<*FontAwesomeIcon *icon={[*"fa"*,* "calendar-check"*]} color={*"white"*} />*

*),*

*to:* "/student/doneexam"*,*

*route:* "/doneexam"*,*

*componet: <*DoneExamScreen */>,*

*isSideMenuComponent: true,*

*},*

*]*;

*const SidebarMenuContent = AllRoutes.filter((*ls*) => {*

*if (ls.isSideMenuComponent) {*

*return* ls*;*

*}*

*})*;

*const getRoutes = () =>*

*AllRoutes.map((*route*,* i*) => {*

*return <*Route *exact path={route.route} element={route.componet} key={*i*} />;*

*})*;

*export* { *SidebarMenuContent*, *getRoutes* };

**4.2 Screen Shorts**

**5. Proposed Enhancements**

**5. Proposed Enhancements**

0

No project can be perfect. With change in technology and security measures, updates are required every now and then. All the features in the system are implanted to fill the gap between the person who wants to invest on a daily basis and the person who has the knowledge about it.

**6. Conclusion**

**6. Conclusion**

* It is an externally wonderful movement while concluding this report. This my first experience to perform such professional work. Objectives of this project were to satisfy user’s requirement, successful implementation of the system, design a user friendly and easy to operate system.
* This software provides appropriate information to user.
* This project is developed for material management.

Students download the materials at any time.