

SOFTWARE ENGINEERING PROJECT

STUDENT RESULTS MANAGEMENT SYSTEM



AP20110010401 Divya Sri Kosaraju

GUIDED BY

Dr Randhir Kumar

Assistant Professor

Department of CSE, SRM, AP.

ABSTRACT

A student result management system is an information tool that is used to track and manage student academic performance. It has a user-friendly interface. The project stores student details and reports of every semester. The student can access the portal using unique login credentials.

This software can help educational institutions save time and resources by organizing student data and improving processes like grading and report generation, while also providing teachers, administrators, and parents with timely and accurate information about student progress. Secure student data storage, flexible grading options, personalized reporting and evaluation tools are provided in the system. Overall, a student result database management system should be designed to provide a user-friendly experience for teachers and administrators.

TABLE OF CONTENTS

S No	Description	Page No
1.	Abstract	2
2.	Table of contents	3
3.	List of figures	4
4.	List of Tables	5
5.	Problem Statement	6
6.	Ch1 Introduction	8
7.	Ch2 SRS	
8.	Ch3 Specific Requirements	
9.	Ch4 Design	

LIST OF FIGURES

1. Use case Diagram
2. DFD L0
3. DFE L1
4. DFD L2
5. Sequence Diagrams
6. Collaboration Diagrams
7. ER Diagram

LIST OF TABLES

1. Data Dictionary
2. Data Design

PROBLEM STATEMENT

The problem with managing student results in educational institutions is that it is a time-consuming process. Normal methods of managing student results involve manually entering and updating data in spreadsheets or databases, which can be time-consuming and prone to errors. These errors can lead to incorrect student records, grading, and course credits. Additionally, it can be challenging for teachers, professors, and administrators to monitor student performance and track progress.

Therefore, there is a need for a more efficient, accurate, and automated system that can manage student results in real-time, streamline the grading process, and provide teachers and administrators with instant access to student records and performance data. A comprehensive student results management system should be able to integrate with different educational systems, provide real-time analytics and dashboards, generate reports, and enable collaboration between teachers, parents, and students to enhance student learning outcomes.

Benefits of student results management software

- + Cost reduction

o

- + Data Security
- + Role Based Access Control

CHAPTER 1

INTRODUCTION

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

1.4 References

1.5 Overview

1.1 PURPOSE

Due to the large number of students and the complexity of the grading system, monitoring student results for educational institutions can be a difficult feat in the world today. Although, we can shorten and greatly enhance this process with the support of recent technologies. Our software engineering project aims to develop a user-friendly system that allows educational institutions to manage their student results efficiently. The database is single-handedly managed by an administrator. Students can view their results when released.

The result management software provides results of students of various academic years. They can access results of previous semesters easily without any hassle.

1.2 SCOPE

The software allows the creation and management of student profiles, including their personal details, academic information, and contact information. The software can automate the process of calculating, and

o

publishing student results. It can also provide analytics to help teachers and administrators analyze the results.

This software is used at educational institutions. The goal of this student results management software is for easy availability of results to parents, faculty, students which is better than report card. There are many advantages of using software rather than file based or paper report because there are chances of forgery, errors and so on.

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

- ◇ Dashboard – which is central interface
- ◆ SRS - Software Requirement Specification
- ◆ DFD - Data Flow Diagram
- + Admin - Administrator
- + Login - Logging in Existing User
- + ER Diagram - Entity Relationship Diagram
- + Pwd - Password
- + Sem - Semester
- + FID - Faculty ID
- + Srms.db - Student result management system database

1.4 REFERENCES

- ❖ www.google.com/softwareengineering

1.5 OVERVIEW

o

The goal of the software engineering project for student results management is to create a system that can handle student academic records in educational institutions in an efficient manner. Both students and faculty members will be able to utilize the system's user-friendly interface. The system's main attributes are as follows:

Student Dashboard: Students can examine their academic records, including grades, using the Student Dashboard function.

Faculty Dashboard: With the help of this function, faculty members can access and control student records like grades.

Automatic Grading System: The system will grade examinations and assignments automatically, easing the stress on professors.

Report Generation: The system will provide reports that can be used in the classroom, including information on student progress and grades.

Advantages

- + It saves time and easy access, error free.
- + It is convenient and flexible to use

Disadvantages

- Administrator should manually enter the data
- It requires large database

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 System Specifications

2.1.2.1 Hardware Requirement

2.1.2.2 Software Requirement

2.1.2.3 Communication Interface

2.2 Product Functions

2.3 Use Case Diagram

2.4 Use Case Description

2.5 User Characteristics

2.6 Constraints

2.7 Assumptions and Dependencies

2.8 Data Flow Diagram (DFD)

2.8.1 Context Level Diagram (Level 0)

o

2.8.2 Functional DFD (Level 1)

2.8.3 Detailed DFD (Level 2)

2.1 PRODUCT PERSPECTIVE

From a product perspective, a student result database management system should be designed to efficiently store, manage, and retrieve academic records for students. The system should provide an intuitive user interface for teachers and administrators to enter and access student data, as well as generate reports and statistics.

User authentication: The system should have a secure login system to ensure that only authorized personnel can access student records.

Data entry: The system should allow teachers and administrators to easily enter student data, such as test scores, attendance records, and demographic information.

Data storage and management: The system should be designed to store large amounts of data and should be able to efficiently retrieve data when needed. It should also have features to manage and organize data, such as sorting and filtering options.

o

Reporting: The system should be able to generate various types of reports. These reports should be customizable and easy to read.

Analytics: The system should provide data analytics tools to help teachers and administrators analyze student performance and identify areas for improvement.

2.1.1 SYSTEM INTERFACES

▪ USER INTERFACES

The design of the UIs should prioritize ease of use, accessibility, and clarity of information.

1. Dashboard: A dashboard is a centralized view that presents key information. It could include metrics such as student performance over time, attendance rates, and overall grades.
2. Student profile: A student profile UI could provide detailed information on a particular student, such as their personal details, academic performance, attendance history.
3. Reports: The reporting UI could allow teachers and administrators to generate reports on student performance, attendance, and other metrics.

o

4. Analytics: An analytics UI could allow teachers and administrators to perform data analysis on student performance and identify trends.
5. The protocol used shall be HTTP
6. There shall be a logical address of the system in IPv4 format

▪ **HARDWARE INTERFACES**

The hardware interfaces for a student result database management system would depend on the specific hardware components being used to run the system.

Server: The database management system may be run on a server, which could be a physical server or a virtual machine. The hardware interface for the server would include network interface cards (NICs), storage devices, and other components necessary for the server to function properly.

Workstations: Teachers and administrators may access the system using workstations, which could be desktop computers, laptops, or tablets. The hardware interfaces for workstations would include input devices such

o

as keyboards and mice, displays for viewing data, and network interface cards to connect to the server.

Printers: The system may need to print reports or other documents, in which case printers would be needed. The hardware interface for printers would include USB or network connections to the workstation or server, as well as any necessary drivers or software.

Scanners: If the system needs to input data from physical documents, such as paper grade sheets or attendance records, scanners may be needed. The hardware interface for scanners would include USB or network connections to the workstation or server, as well as any necessary drivers or software.

Wi-Fi router: Wi-Fi router is used for internetwork operations in educational institutes and simply transfer data from pc to server.

▪ SOFTWARE INTERFACES

MySQL Server

OS windows

PHP

HTML, CSS, JS

2.1.2 SYSTEM SPECIFICATIONS

o

2.1.2.1 H/W REQUIREMENT

Core i5 processor or above

2GB RAM

20GB of hard disk space in terminal machines

1TB hard disk space in Server Machine

2.1.2.2 S/W REQUIREMENT

Windows 7 or above operating system

MySQL server

Tool Kit

XAMPP, Visual studio code

2.1.3 COMMUNICATION INTERFACES

NIC (Network Interface Card) – It is a computer hardware component that allows a computer to connect to a network. NICs may be used for both wired and wireless connections.

TCP/IP protocol- Internet service provider to access and share information over the Internet

Ethernet Communications Interface- Ethernet is a frame-based computer network technology for local area networks (LANs) Ubiquitous, easy to set up and easy to use. Low cost and high data transmission rate.

o

2.2 PRODUCT FUNCTIONS

- o Provide access to registered users only.
- o Enable students to view their record.
- o Enable parents to view their record.
- o Enable faculty to update and entry of the record.
- o Approved by admin.
- o Admin access to student's record.
- o Admin Verify results and Generate reports.
- o Admin can view monthly/yearly records

2.3 USE CASE DIAGRAM

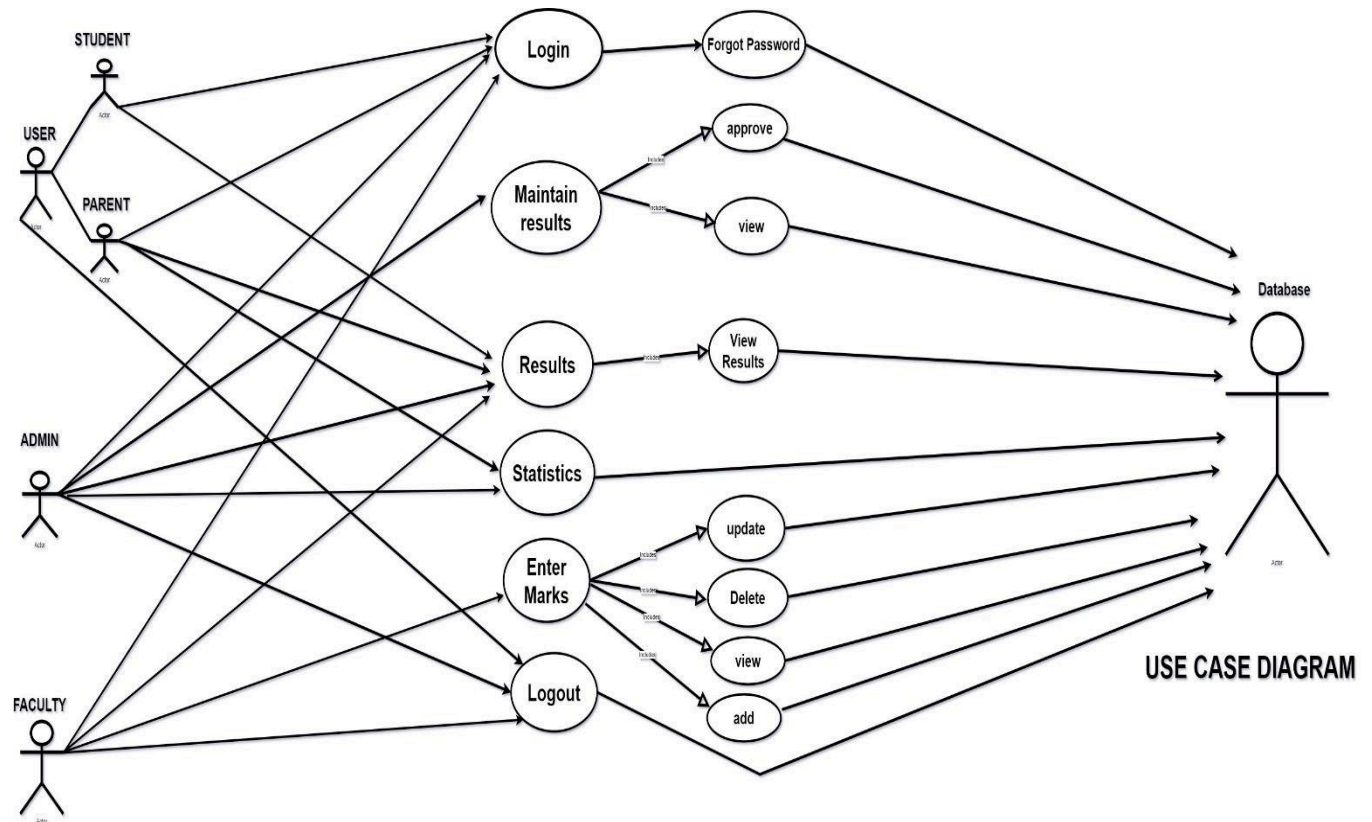


Fig.1

2.4 USE CASE DESCRIPTION

□ ACTORS

- o Admin
- o Student
- o Parent
- o Faculty
- o Database

□ USE CASES

- o

- o Login
- o Maintain Results
- o Results
- o Statistics
- o Enter Marks
- o Logout

1. Login

Brief description:

This use case describes how the user logs into the system.

Flow of events:

Basic flow:

1. The system requests the actor to enter the name and the password.
2. The actor enters the name and the password.
3. The system validates the entered name and the password and logs the actor into the system.

Alternative flow:

1. If in the basic flow the actor enters the invalid name or password then display an error message.
2. The actor can choose either to return to the beginning of the basic flow or cancel the login at which point use ends.

Preconditions:

The actor must already be a registered candidate in the result management system.

o

Post conditions:

The use case was successful, and the actor is now logged into the system. If not, the system state is unchanged.

2.Maintain Results

Brief Description:

This use case describes how the admin actor updates, manages and deletes the records of all users.

Flow of Events:

Basic Flow:

- 1.The actor manages all the records of users.
- 2.The admin approves or views the data in the database.

Alternative flow:

The admin resets the data.

Preconditions:

Login to the system through admin login view.

Postconditions:

If the use case is successful, then the admin can log out from the system.

3.Results

o

Brief Description:

Different kinds of users have different views of results.

The student uses this use case for checking their results.

Flow of Events:

Basic Flow:

Students go to the results and check all their reports available.

The statistics of the results can be accessed.

Alternative Flow:

When the results are displayed, the alternative way would be jumping to results.

Precondition:

All the results entered must be pre-approved by the admin to be able to show up in the results column.

Postcondition:

If it is successful, then students can come out of the use case, result.

4.Entry of Marks

Brief Description:

This use case is accessed by the faculty, and they can manage all the marks of students like viewing, entering, updating, deleting.

Flow of Events:

Basic Flow:

o

The faculty enters the marks of the students and waits for the approval of the admin.

Alternative Flow:

The faculty can view, edit and update records.

Precondition:

The faculty must be logged on to the system using their own faculty id to enter marks in their respective courses.

Postconditions:

After entering the marks, the approval of admin is taken, and changes are applied, the faculty can log out of the system. Records are added to the database.

5.LogOut

Brief Description:

Logging out of the system after completion provides security to the user so that every time a login action is taken password must be entered.

Flow of Events:

Basic Flow:

After the actor does all the work this can be used to log out of the system.

Alternative Flow:

o

If there is any glitch or network issue the actor stays on the page which is last visited or returns to the dashboard page until the logout process is completed and logs out of the system.

Precondition:

The actor must save the changes they made before logging out of the system because if not saved the changes might not appear after logging in again.

Postcondition:

The actor gets logged out of the system.

2.5 USER CHARACTERISTICS

□ ADMIN

Admin has full access to the system which means he can manage any activity in the system. He is the most privileged user who can access the system.

- Access student record, faculty Record.
- Add new student entry in system database.
- Confirm results and records and Generate reports.
- View Records.

□ FACULTY

Faculty has the access to entering of marks for respective student

o

Access for Updating of results

View the records

- ☐ **STUDENT-** Student can view the results
- ☐ **PARENT-** Parents can view the results and statistics

2.6 CONSTRAINTS

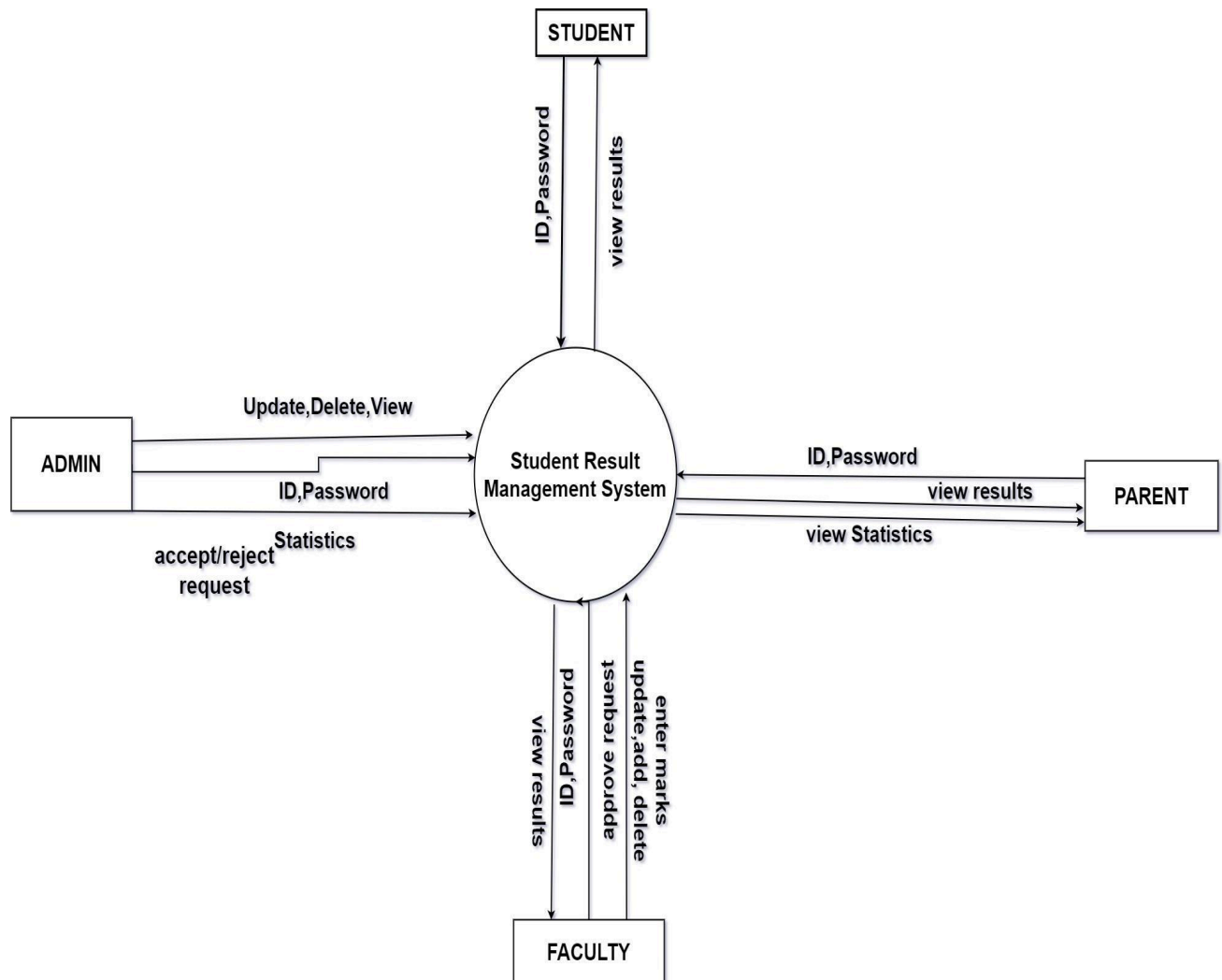
- The system is wirelessly networked with encryption.
- The system is only accessible from the institute's website only.
- Less RAM and processing power.
- Each user should have an individual ID and password.
- Only administrator can access the whole system

2.7 ASSUMPTIONS AND DEPENDENCIES

- Each user must have a valid user id and password
- Server must be running for the system to function
- Users must log in to the system to access any record.
- Only the Administrator can delete records

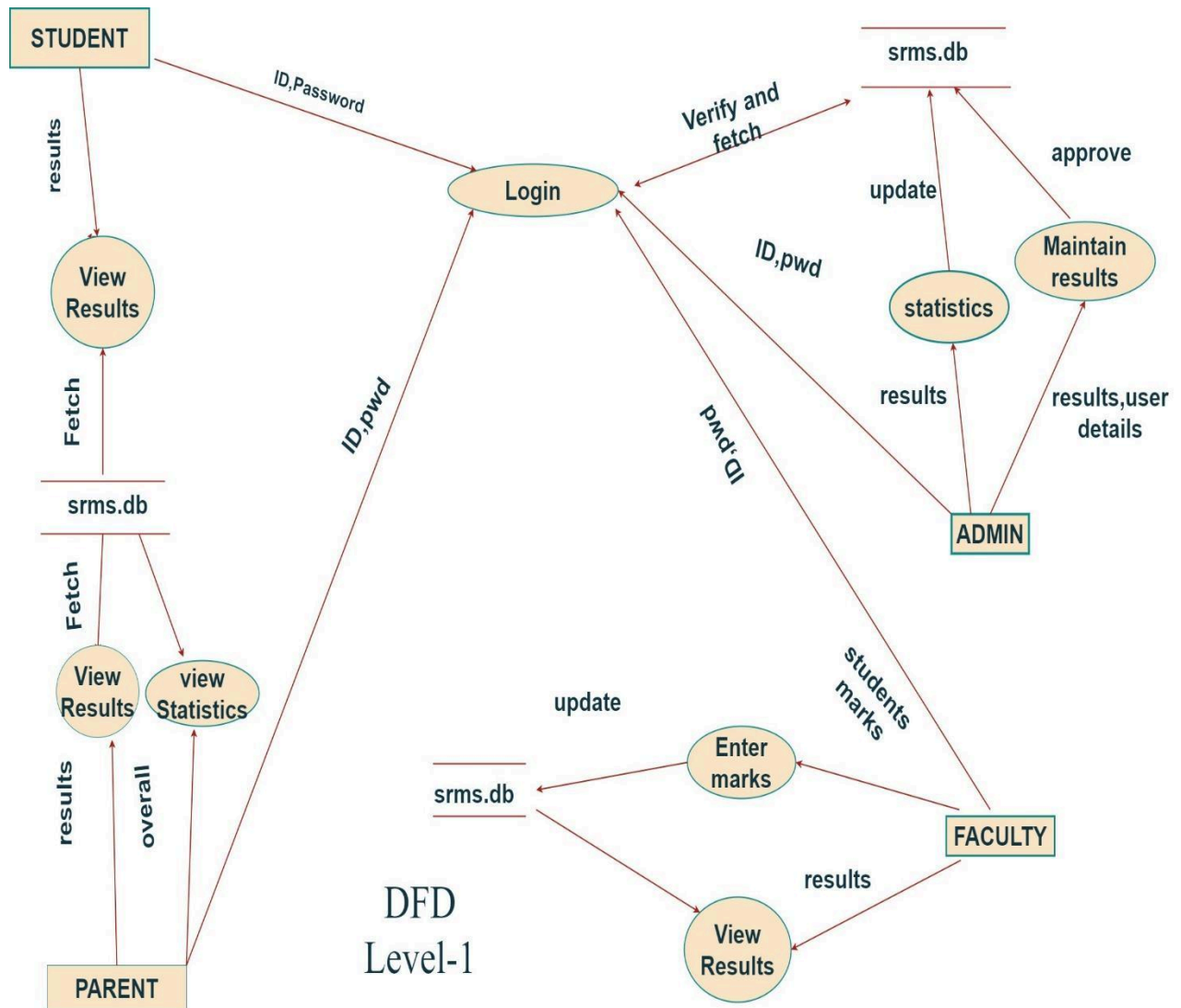
DATA FLOW DIAGRAM

0-LEVEL DFD:

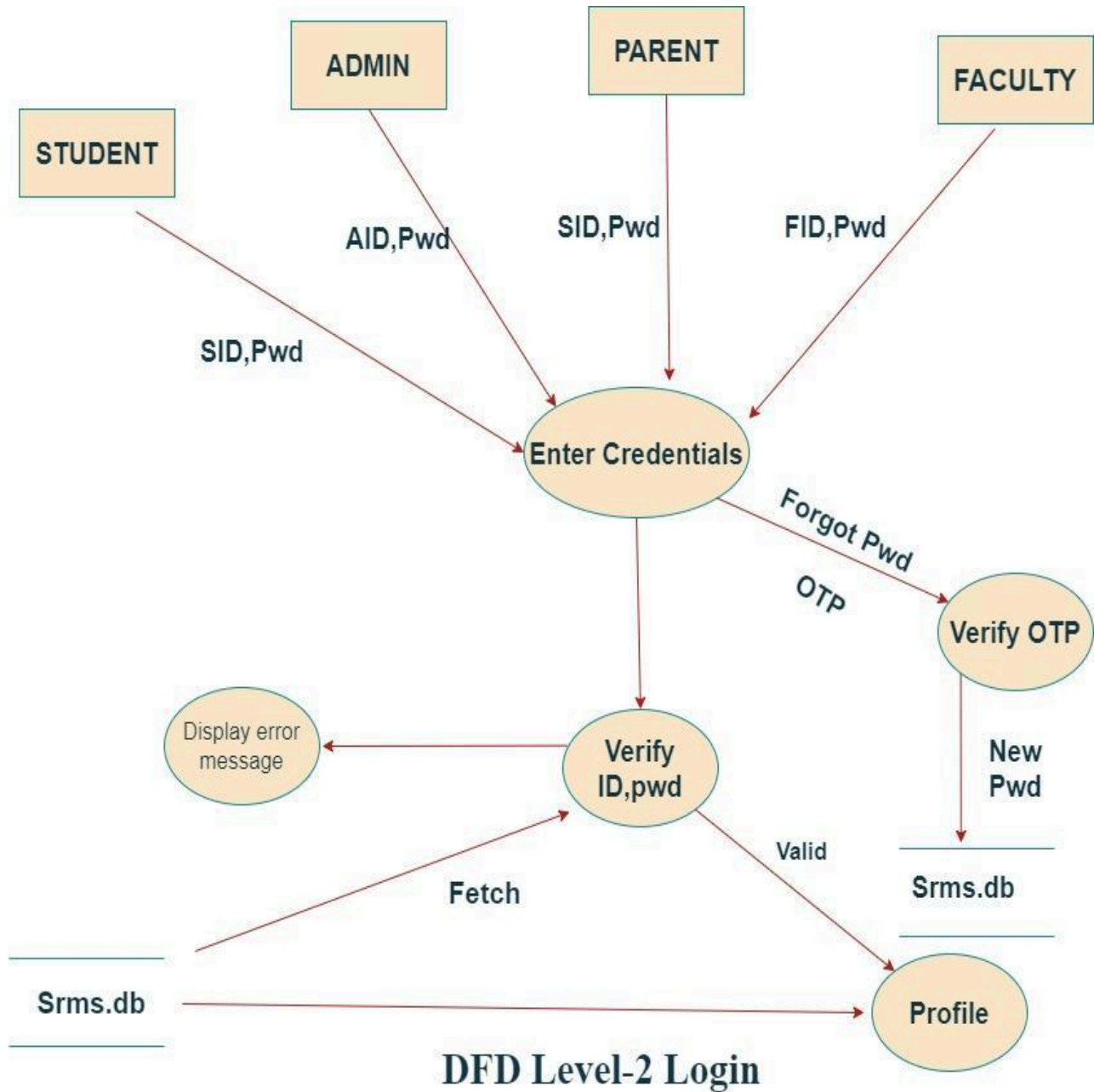


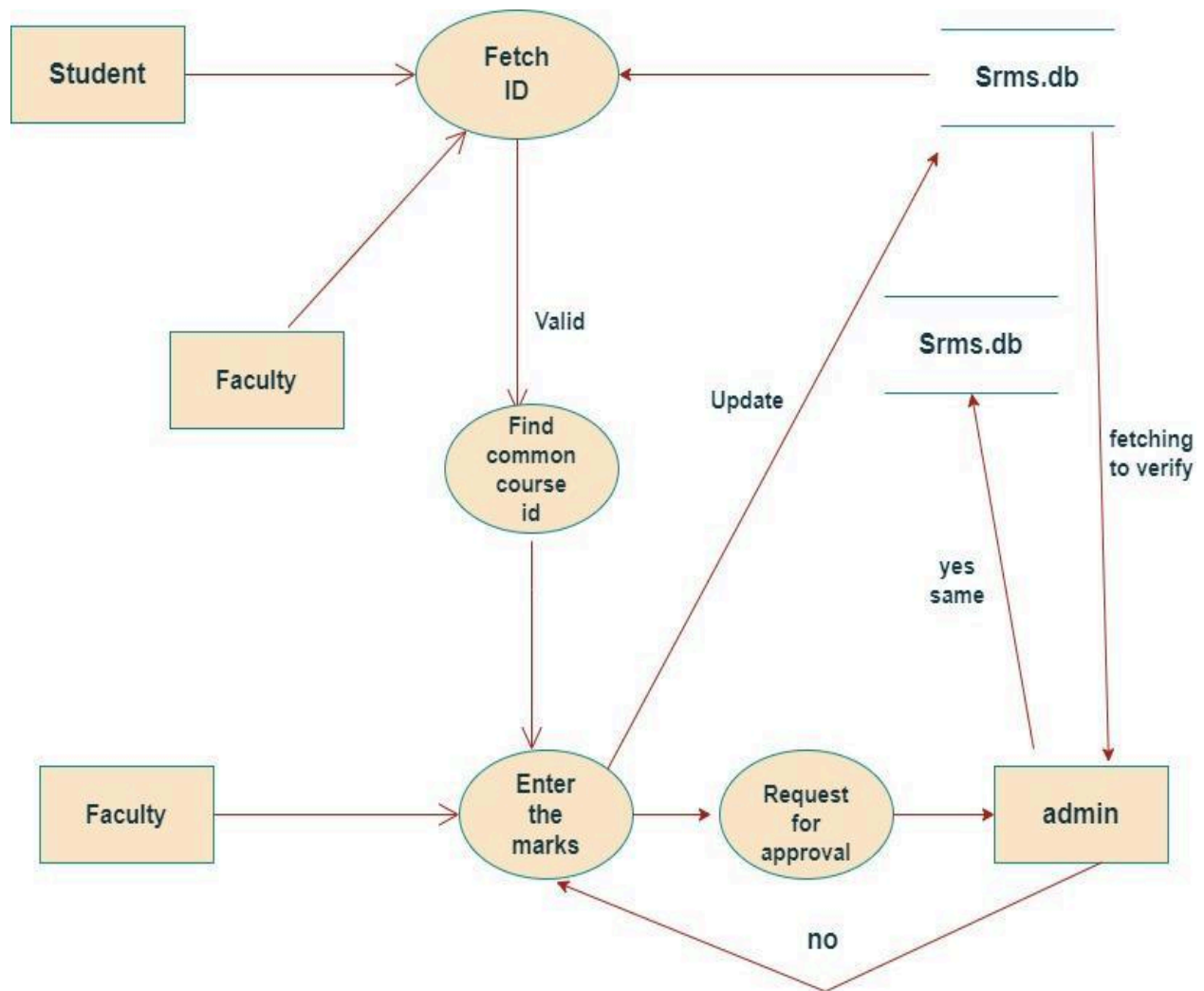
DFD Level-0

1-LEVEL DFD:

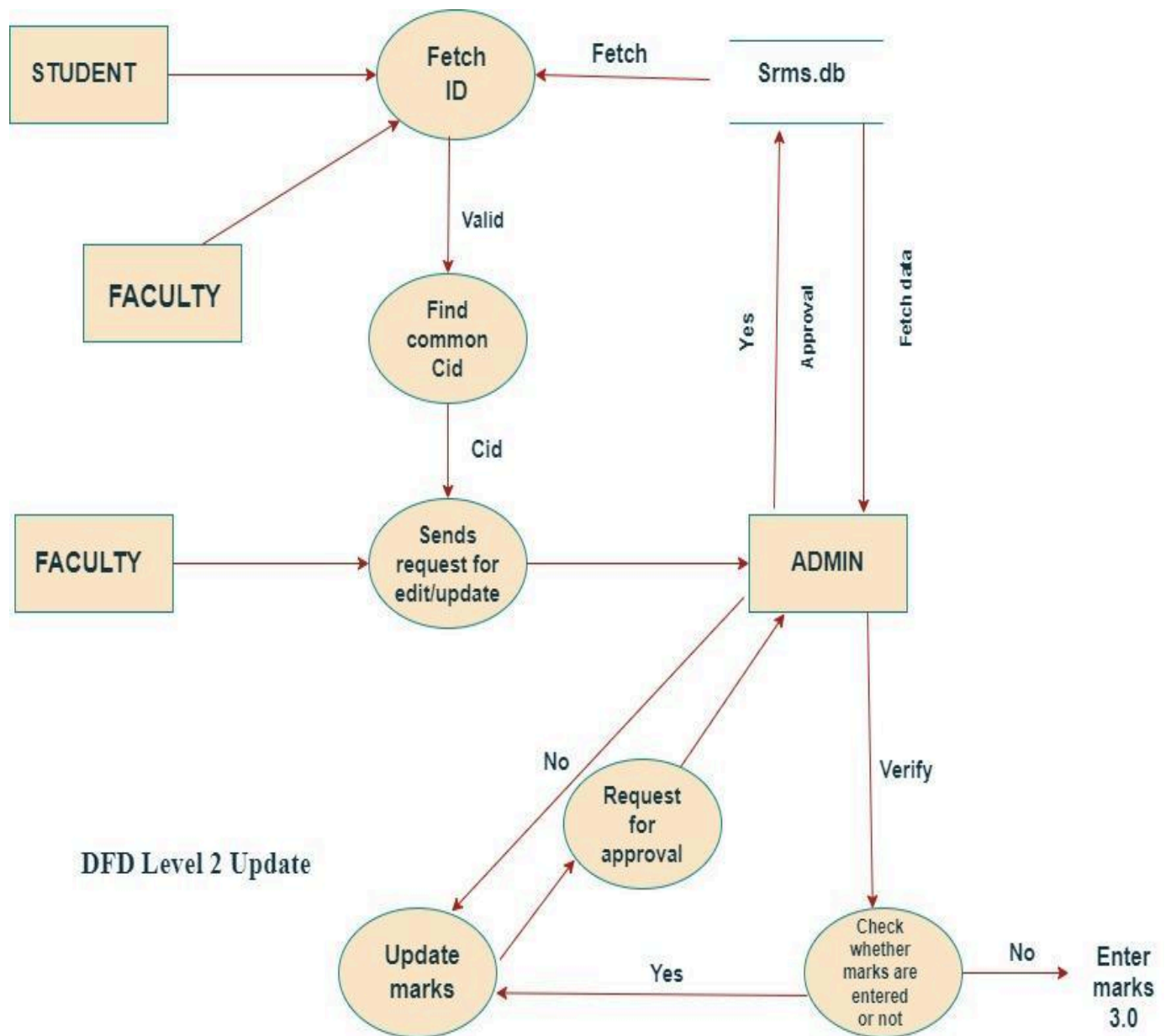


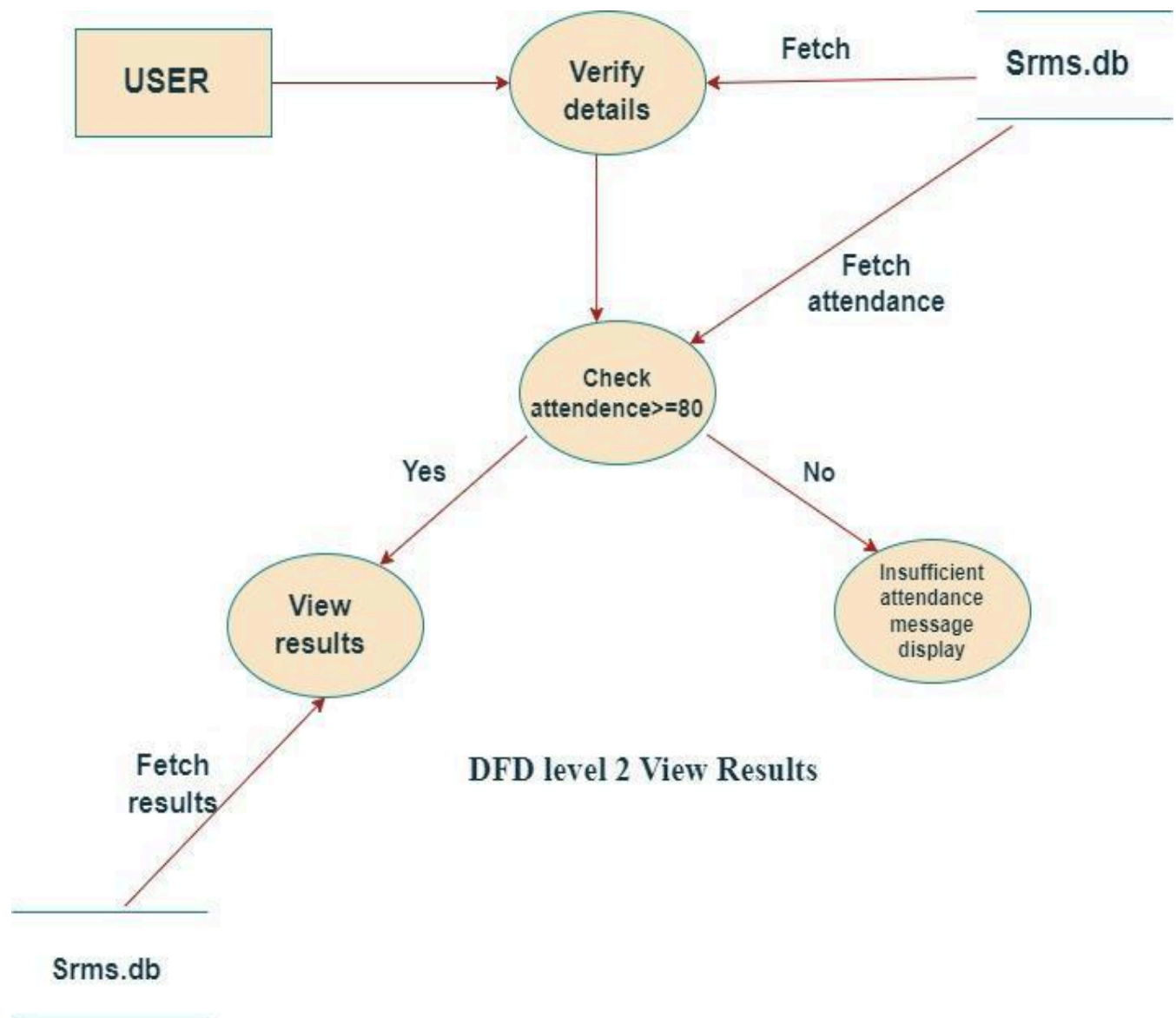
2-LEVEL DFD:



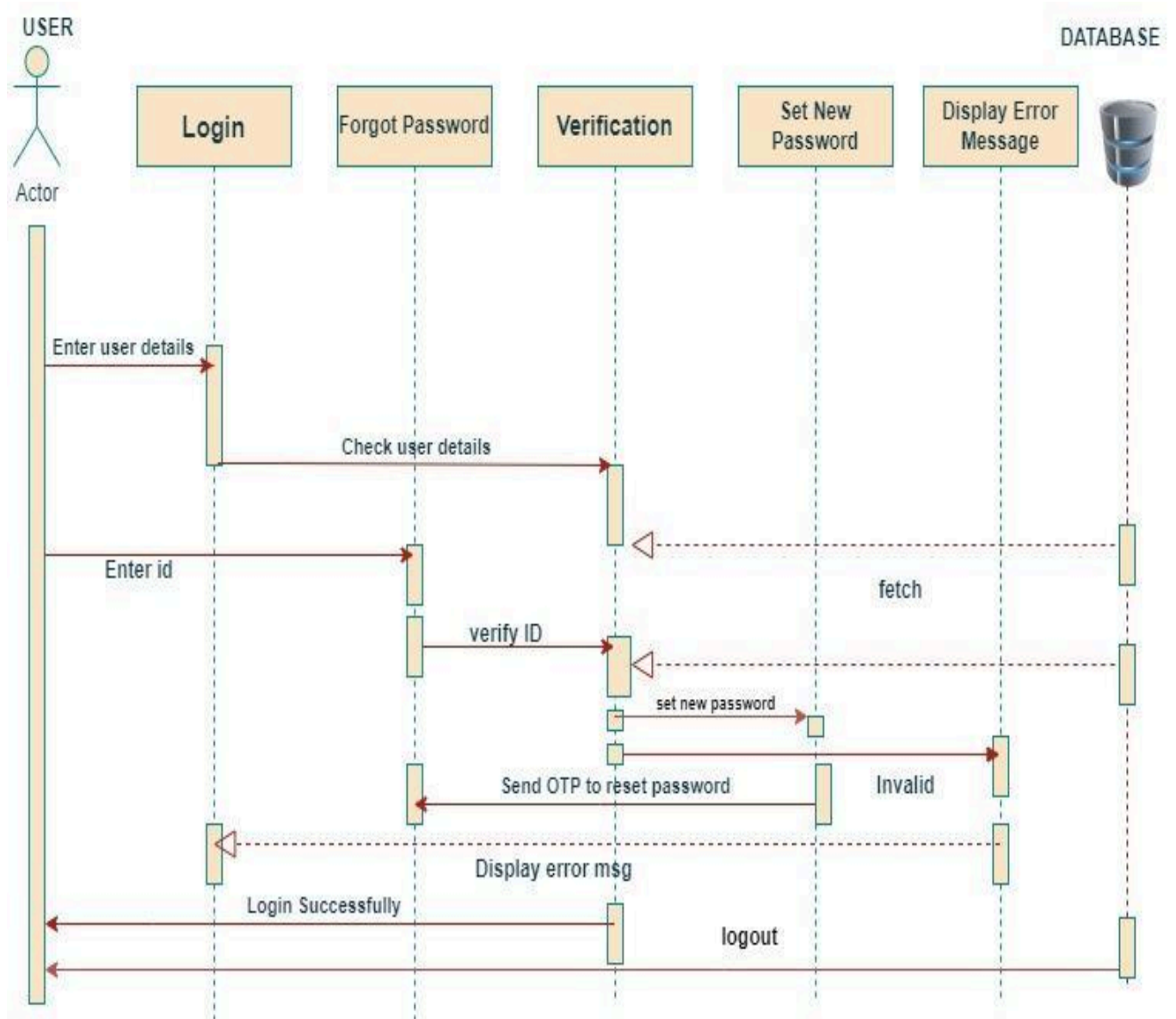


DFD level 2 Enter Marks

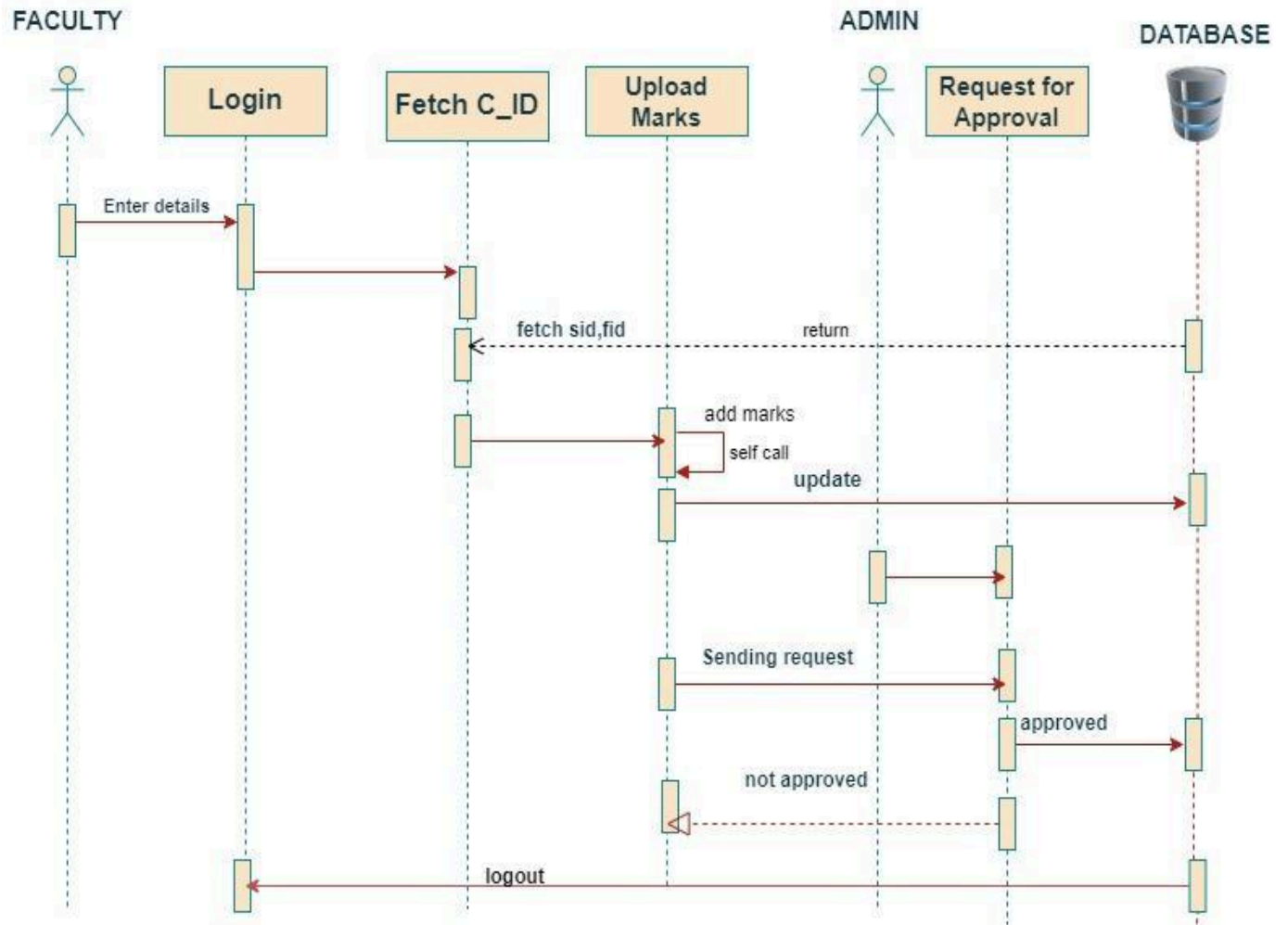




SEQUENCE DIAGRAMS

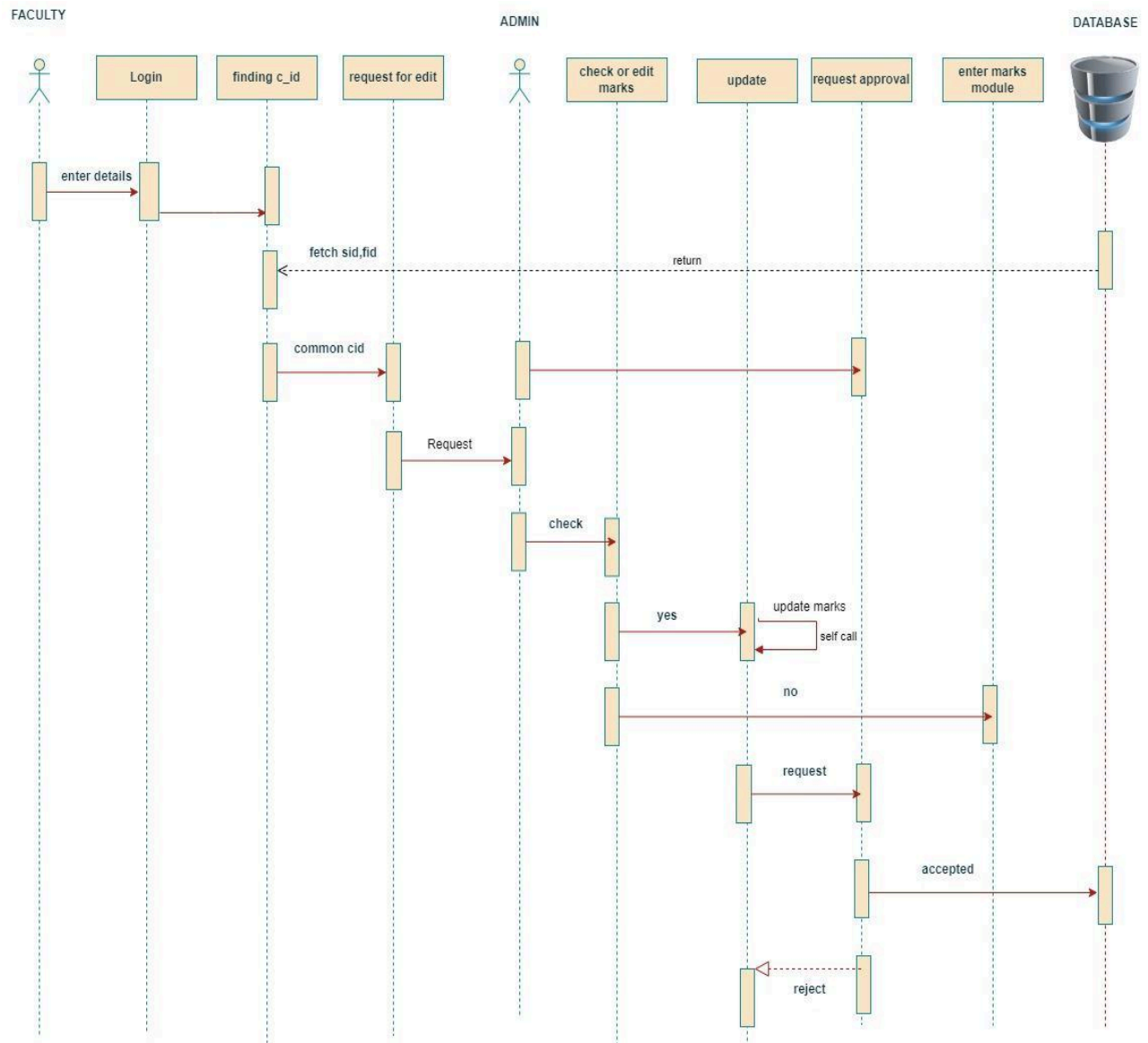


Sequence Diagram Login



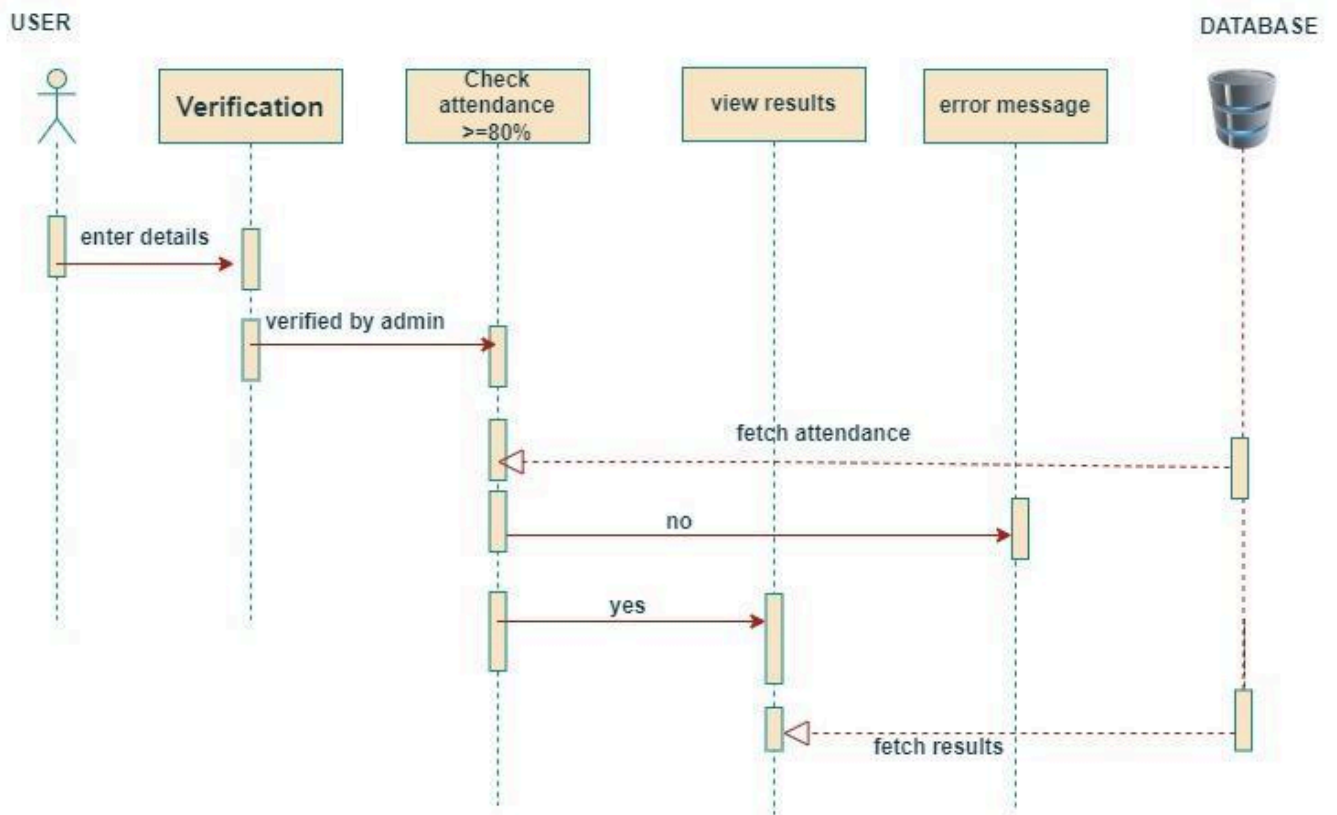
Sequence Diagram Entering Marks

0



Sequence Diagram Edit Marks

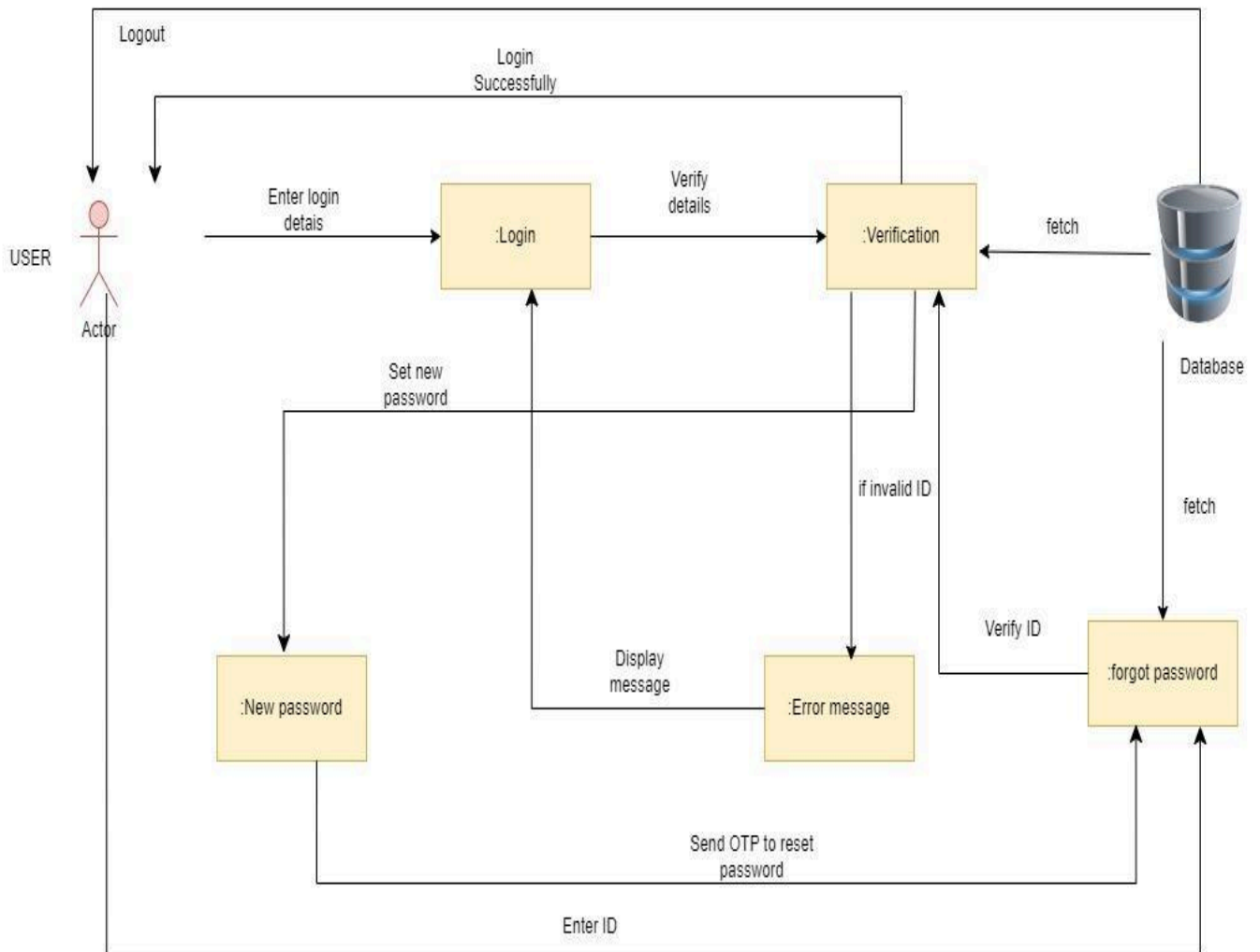
0



Sequence Diagram View Results

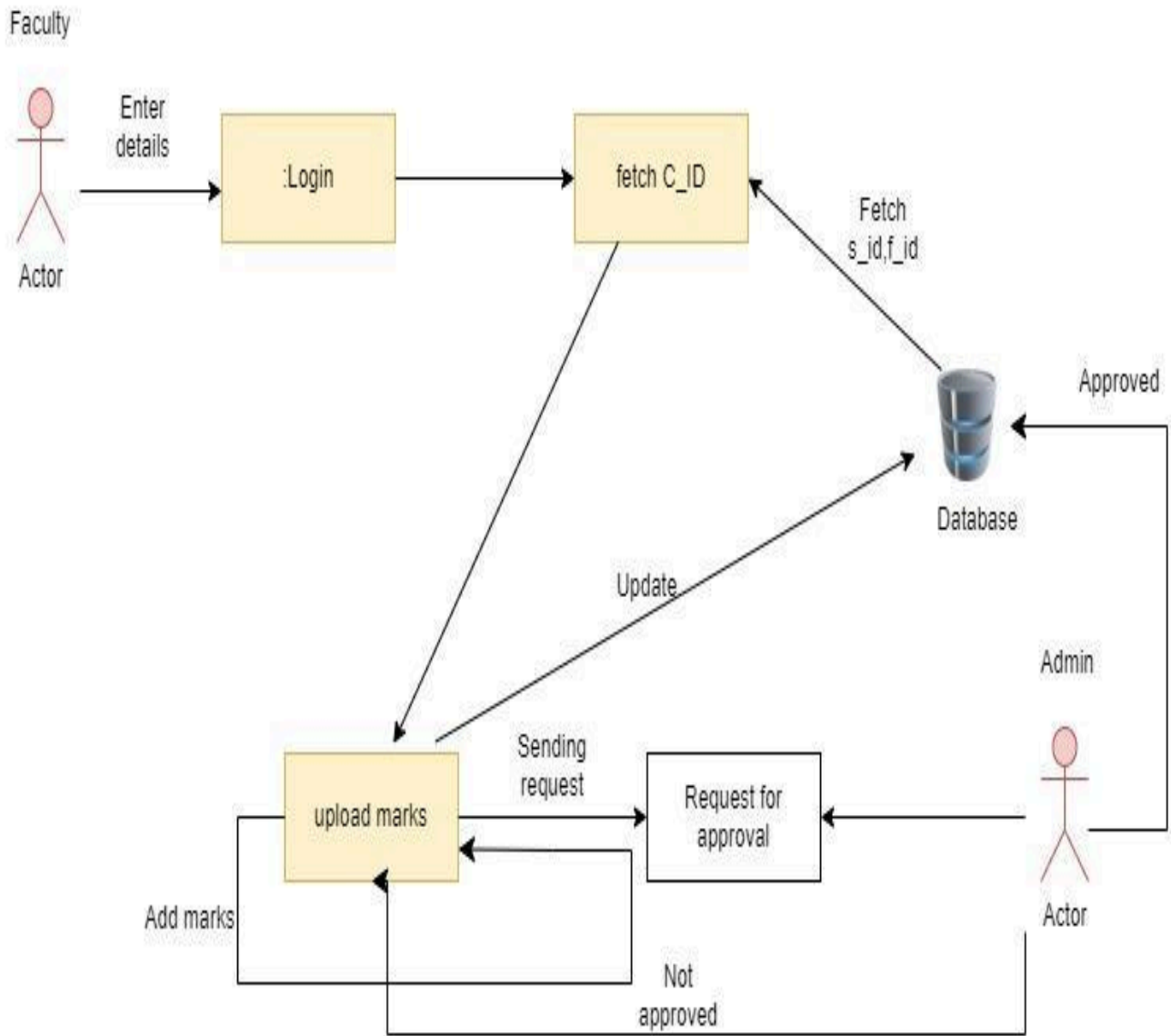
o

COLLABORATION DIAGRAM



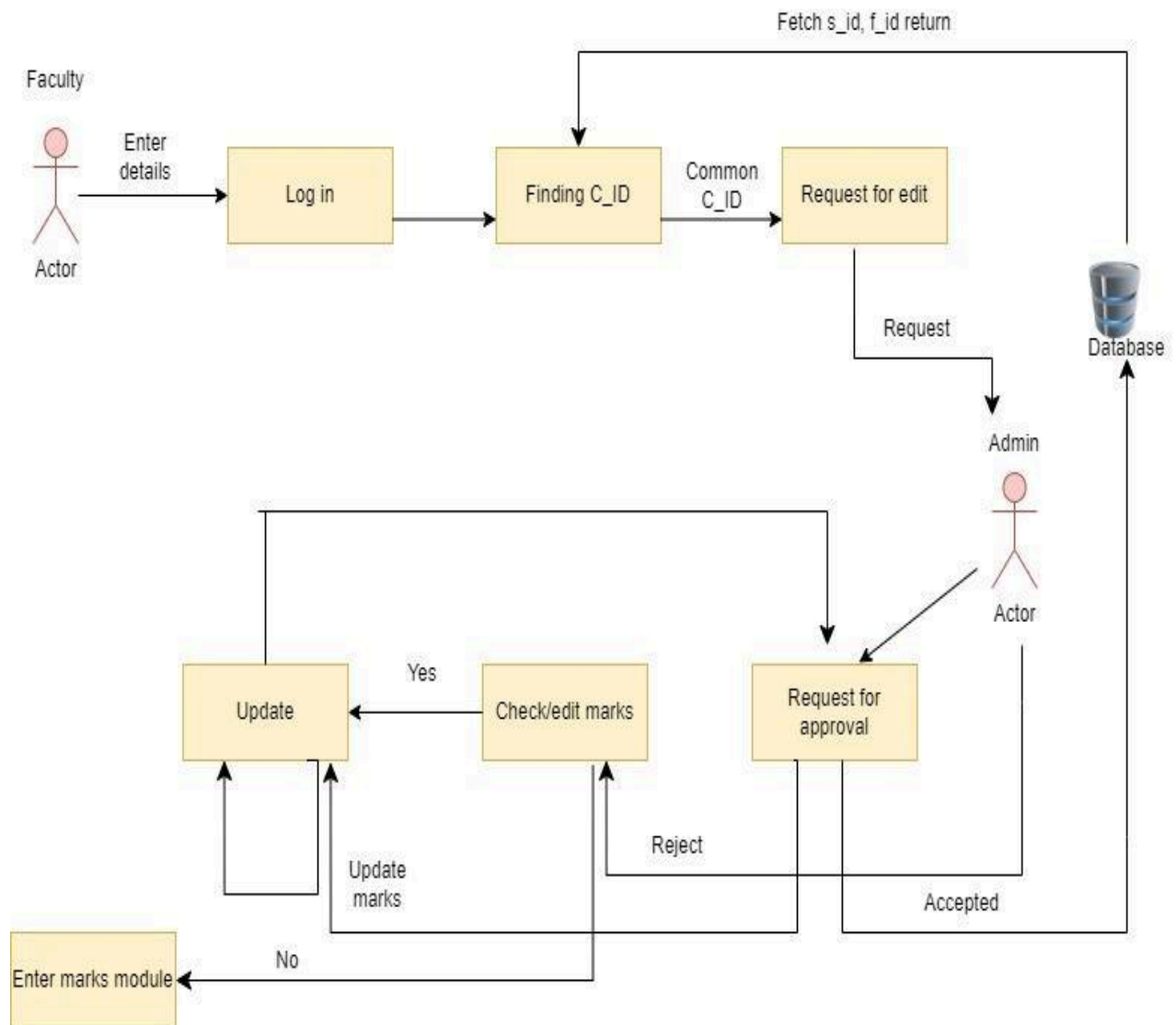
Collaboration Diagram login

o



Collaboration Diagram Entering Marks

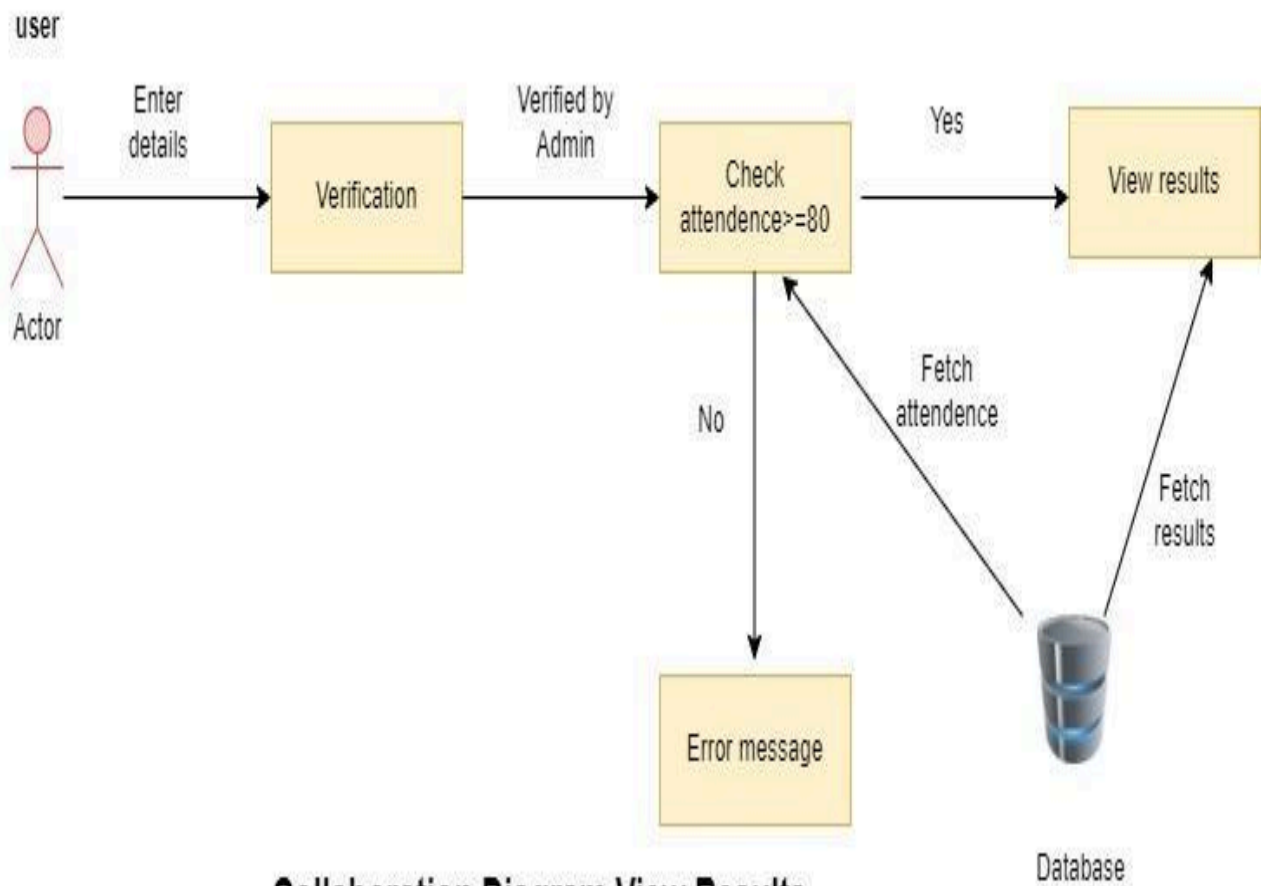
o



Collaboration Diagram Edit marks

Edit marks

o



Collaboration Diagram View Results

CHAPTER 3

SPECIFIC REQUIREMENTS

3.1 Performance requirements

3.2 Safety requirements

3.3 Security constraints

3.4 Software system attributes

3.4.1 Usability

3.4.2 Availability

3.4.3 Correctness

3.4.4 Maintainability

3.4.5 Accessibility

3.5 Functional Requirement

3.1 PERFORMANCE REQUIREMENTS

- Response time- The system will give responses within 1 second after checking information.
- Capacity-The system must support 1000 people at a time
- User interface- User interface screen will response within 5 seconds
- User-Friendliness: The ERP should be easy to use and navigate, with an intuitive user interface that minimizes the need for training or support
- Reliability: The system should be always reliable and available.

3.2 SAFETY REQUIREMENTS

Data Security: The system should be designed with strong data encryption and secure authentication mechanisms to protect sensitive information, such as student names, addresses, and grades. All data should be stored in secure databases and access to this data should be restricted to authorized personnel.

Backup and Recovery: The system should be designed with a backup and recovery mechanism to ensure that all data is protected in the event of a system failure or data loss. Regular backups should be taken and

o

stored securely in a remote location to ensure that the data can be restored in the event of a disaster.

Access Controls: The system should be designed with appropriate access controls to ensure that only authorized personnel can access the system and the data. This should include features such as password protection, multi-factor authentication, and role-based access control.

System Maintenance: The system should be maintained regularly to ensure that it is up-to-date and free of vulnerabilities. This should include applying security patches and updates, as well as regularly testing the system for vulnerabilities.

Disaster Recovery: The system should be designed with a disaster recovery plan in place to ensure that the system can be quickly restored in the event of a disaster, such as a natural disaster or cyber-attack.

Audit Trail: The system should be designed with an audit trail that logs all activities related to the system and the data. This will help to track any unauthorized access or changes to the data.

3.3 SECURITY REQUIREMENTS

o

1. Want to take responsibility for failures due to hardware malfunctioning.
2. The Warranty period for maintaining the software would be one year.
3. Additional payments will be analyzed and charged for further maintenance.
4. If any error occurs due to a user's improper use. The Warranty will not be allocated to it.
5. No money back returns for the software.

3.4 SOFTWARE SYSTEM ATTRIBUTES

3.4.1 Usability: Software can be used again and again without distortion.

3.4.2 Availability: The system shall be available all the time.

3.4.3 Correctness: Bug free software which fulfills the correct need/requirements of the client.

3.4.4 Maintainability: The ability to maintain, modify information and update fix problems of the system.

3.4.5 Accessibility: Administrator and many other users can access the system, but the access level is controlled for each user according to their work scope

3.5 FUNCTIONAL REQUIREMENTS

1. LOGIN

PARENT ADMIN STUDENT FACULTY: Can login using unique Id and Password after this system shall show his/her profile.

2. VIEW

PARENT: Can view the Results and Statistics.

ADMIN: can view user Data and Manage the Website.

STUDENT: can view the Results.

FACULTY: can view the Marks entered and info of student.

3.UPDATE

ADMIN: can update the user details and marks after request from faculty.

4.ENTRY:

FACULTY: can enter the marks and results of students.

CHAPTER 4

DESIGN

4.1 Data Dictionary

4.2 ER Diagram

4.3 Data Design

4.1 DATA DICTIONARY

1.Student_ID = [0-9]*

2.Admin_ID= [0-9] *

3.Faculty_ID= [0-9] *

4.Contact = [0-9]{10}

5.Name=[a-zA-Z] *

6.Age= [0-9] *

7.Gender=[a-zA-Z] *

8.Marks= [0-9] *

9.Grade= [0-9.0-9] *

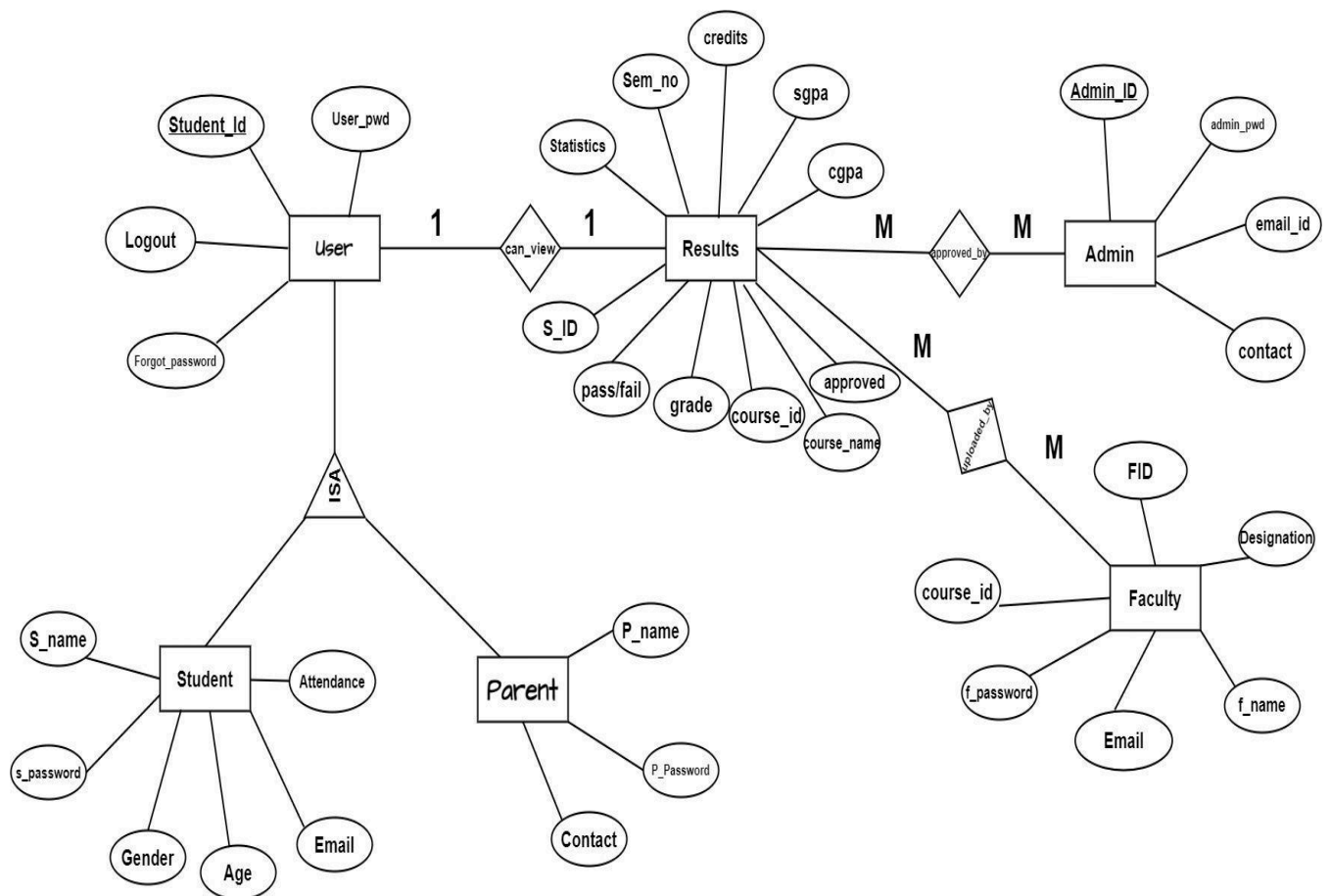
10.Password= [0-9] *

o

4.2 ER DIAGRAM

Entities:

1. Student
2. Parent
3. Results
4. Admin
5. Faculty



4.3 DATA DESIGN

1.	S_ID	PK	INT
2.	F_ID	PK	INT
3.	A_ID	PK	INT
4.	PWD		INT
5.	EMAIL		VARCHAR
6.	CONTACT		INT
7.	AGE		INT
8.	ATTENDANCE		INT
9.	NAME		VARCHAR
10.	SEM_NO		INT
11.	SGPA	CGPA	INT
12.	C_NAME		VARCHAR

The screenshot shows a code editor with a file explorer on the left. The file explorer lists a project structure with folders like 'app', 'Models', 'Support', and 'tests'. The 'tests' folder is expanded, showing 'LoginTest.php'. The main editor displays the content of 'LoginTest.php', which includes a PHP script for testing login credentials. The script defines a 'validateCredentials' function and a 'LoginTest' class that extends 'PHPUnit\Framework\TestCase'. The class contains two test methods: 'testValidCredentials()' and 'testValidCredentials1()'. The 'testValidCredentials()' method sets '\$name' to '1' and '\$pass' to '12366', calls 'validateCredentials', and asserts the result is true. The 'testValidCredentials1()' method sets '\$name' to '2' and '\$pass' to '25446', calls 'validateCredentials', and asserts the result is true. The bottom panel shows the 'TERMINAL' output, indicating the test was successful: '2 tests, 2 assertions'. The terminal also shows the PHP version (8.2.4) and the configuration path (C:\xampp\htdocs\SE\phpunit.xml).

```

1 <?php
2 function validateCredentials($name, $pass) {
3     if (($name === '1' && $pass === '12366') || ($name === '2' && $pass === '25446')){
4         return true; }else {
5             return false;}}
6 }
7 <?php
8 use PHPUnit\Framework\TestCase; //require_once 'student_login.php';
9 class LoginTest extends TestCase {
10     public function testValidCredentials() {
11         $name = '1';
12         $pass = '12366';
13         $result = validateCredentials($name, $pass);
14         $this->assertTrue($result);
15     }
16     public function testValidCredentials1() {
17         $name = '2';
18         $pass = '25446';
19         $result = validateCredentials($name, $pass);

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Divya@DIVYA21-PC MINGW64 /c:/xampp/htdocs/SE
 PHPUnit 10.1.3 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.4
 Configuration: C:\xampp\htdocs\SE\phpunit.xml

Time: 00:00.018, Memory: 6.00 MB

OK (2 tests, 2 assertions)