Software Design Document (SDD)

Project Title: Student Record Management System

Course:CP476A – Internet Computing

Team Members: Talei Ibhanesebhor

Date:31 April 2025

1. Introduction

This project delivers a web-based Student Record Management System designed to facilitate secure management of student academic records. Built using PHP and MySQL without relying on third-party environments like XAMPP, the system supports authenticated access and enables users to search, update, delete, and calculate grades for student course records. It provides a lightweight yet functional interface that handles data dynamically through robust backend scripting and structured database interactions.

2. System Architecture

The system is based on a classic three-tier architecture:

- Client Tier: Users interact through HTML forms rendered by PHP. Data input and results are processed via this UI.
- Middle Tier: Contains PHP scripts responsible for processing logic including authentication, validation, CRUD operations, and grade calculations.
- Data Tier: A MySQL database containing two normalized tables: NameTable and CourseTable. These are connected via student_id.

3. Database Design

Tables:
- NameTable: student_id (Primary Key), full_name (VARCHAR)
- CourseTable: student_id (Foreign Key), test_score (INT), final_exam_score (INT)

The design ensures data consistency using foreign key constraints between tables. A student may have multiple course records but only one name entry.

4. Implementation Details

4.1 Login (Authentication)

File: login.php
Takes username/password via POST. Queries the DB for credentials. If valid, redirects to dashboard.php. Else, displays an error.

4.2 Search

File: search.php
Accepts student name as input. Queries NameTable using LIKE SQL keyword. Displays results in an HTML table with update/delete options.

4.3 Update

File: update.php
Fetches current student info based on ID. Provides form fields to update scores or names. Commits changes to the DB on submit.

4.4 Delete

File: delete.php
Accepts student ID via GET or POST. Deletes student record from both tables. Includes confirmation alert before deletion.

4.5 Grade Calculation

File: grades.php
Fetches all course records. Dynamically calculates the final grade using weighted averages. Displays the result in a sortable HTML table.

5. Output and Sorting

Outputs:
- Search: Table with matching names.
- Update: Confirmation of changes.
- Delete: Notification of successful deletion.
- Grades: Table with names and computed grades.

Sorting:
- Uses basic PHP array sorting or SQL ORDER BY depending on context (e.g., sort by name or grade descending).

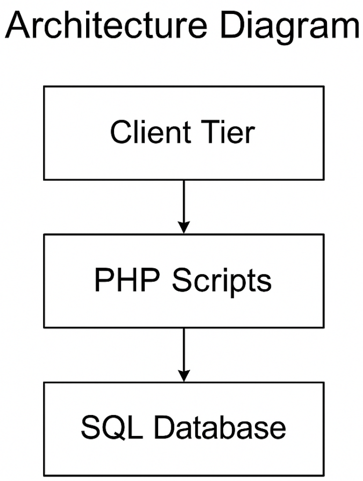
6. API Specification and Component Interaction

Component	Method	Input	Output	Function
login.php	POST	username, password	Redirect / Error	Authenticates user
search.php	GET/POST	student name	HTML table	Searches NameTable
update.php	POST	student ID, new data	Success / fail message	Updates course data
delete.php	POST	student ID	Deletion confirmation	Deletes record
grades.php	N/A	N/A	Table of grades	Calculates/ display grades

7. Diagrams

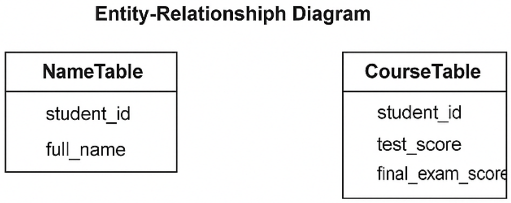
Diagrams to be included:

- Architecture Diagram



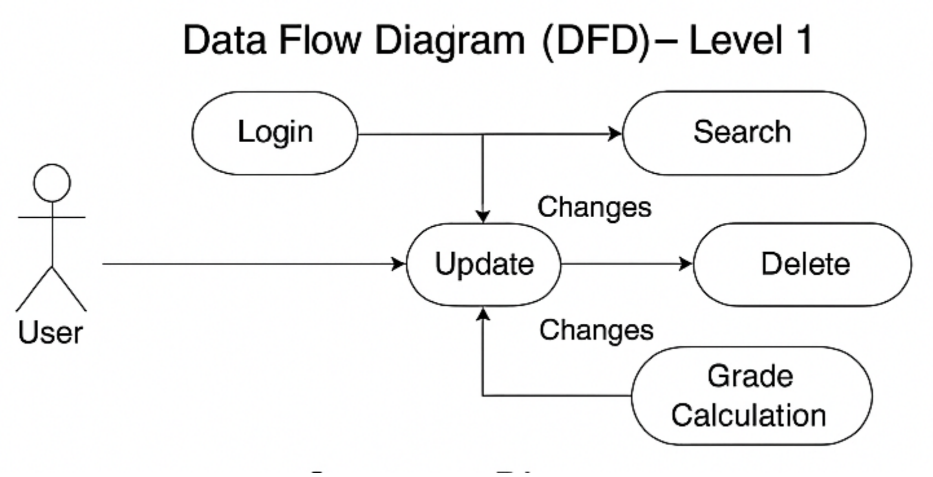
Architecture Diagram

- ER Diagram

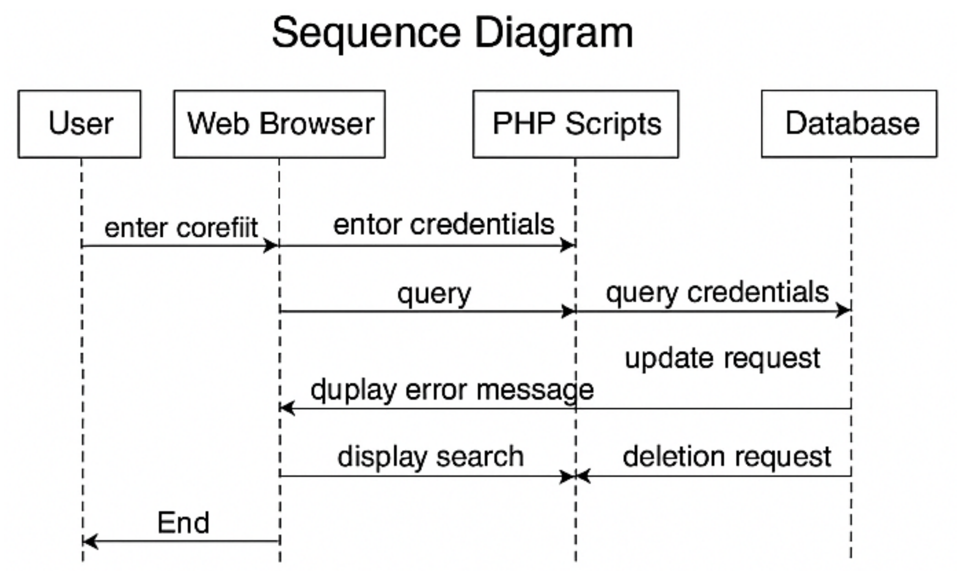


Entity-Relationshiph Diagram

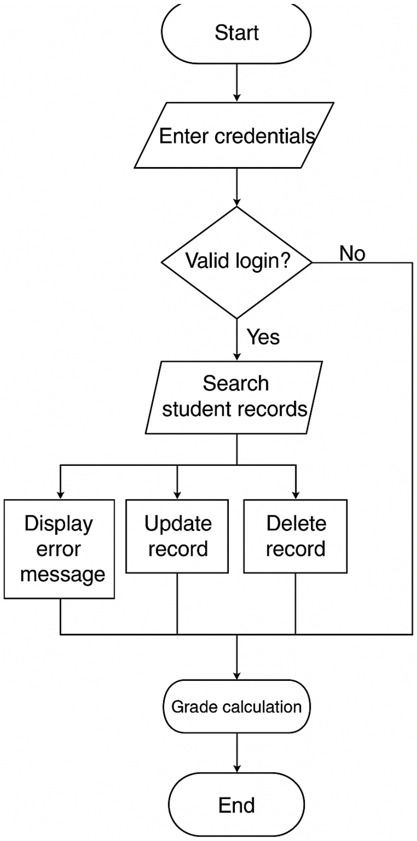
- Data Flow Diagram (DFD - Level 1)



- Sequence Diagram



- Workflow Logic Diagram



8. References

- W3Schools. (n.d.). PHP MySQL CRUD Operations. https://www.w3schools.com/php/php_mysql_intro.asp
- MySQL Documentation. (2024). MySQL 8.0 Reference Manual. https://dev.mysql.com/doc/
- GeeksforGeeks. (n.d.). PHP Form Handling. https://www.geeksforgeeks.org/php-form-handling/