CP476 - Internet Computing
Term Project

Topic: Student Grade Web Server

Aidan Traboulay, Brianna Nguyen,
Mobina Tooranisama, Nausher Rao

Group 26

Dr. Lunshan Gao

April 10th, 2023

Contents

1 Introduction

1.1 Purpose

The purpose of this Software Design Document (SDD) is to offer a comprehensive overview of the design and development process for a web-based application, which aims to simplify the calculation of weighted grades – a task notorious for being time-consuming and prone to errors. This user-friendly application enables users to manage student grades with ease and efficiency.

1.2 Scope

As a critical element of the software development life cycle, this document not only ensures the system's effective and efficient design, development, and testing but also serves as a reference point for future improvements and modifications.

1.3 Overview

The SDD delineates the design and development phases for a web server that interfaces with a MySQL database server, utilizing PHP as the server-side language. The web server is configured to function on a Windows operating system, with Apache serving as the server software. The project's goal is to compute a student's final grade based on the data inputted into two distinct tables. This document is broken down into seven (7) different sections, each outlining a constituent of the web server system.

2 System Overview

The software system functions as an all-inclusive web-based platform, enabling users to smoothly engage with a MySQL database server. This engagement is made possible through the execution of a minimum of two prepared SQL statements, specifically focusing on data retrieval (SELECT statement) and data modification (UPDATE statement) operations. This architecture offers an efficient and secure means of accessing and modifying the data stored in the database.

Users have the capability to view a student's final grade in a specified format, which is obtained from data contained in two separate tables – the "Name Table" and the "Course Table." This aspect improves the system's usability by presenting vital data in an easily accessible and comprehensible way.

The software system is developed on a Windows operating system and employs Apache as the web server, with PHP serving as the server-side language. This blend of technologies guarantees a stable and dependable platform for end-users.

Based on the requirements documentation, the utilization of third-party packages like XAMPP, phpMyAdmin, and WAMP is forbidden. Thus, this system was created entirely following a bottom-up approach, building each functional aspect of it.

Significant focus has been placed on the system's security, incorporating measures to protect the data and avert SQL injection attacks. By giving precedence to data security, the system can preserve user trust and ensure the protection of sensitive information.

The software system's requirements are meticulously crafted to address the varied needs of its users while meeting crucial functional prerequisites. This considerate strategy guarantees that the system provides a worthwhile and dependable service to its intended user base.

3 System Architecture

3.1 Architectural Design

The system architecture of our web server employs a client-server model, providing a comprehensive and organized structure for efficient communication and data management. In this model, end-users interact with a front-end interface, which is developed using HTML, CSS, and JavaScript, to access the back-end, built with PHP, for fetching and displaying data from the database. This distributed computing system consists of a network of computers, with clients requesting and obtaining services or resources from servers.

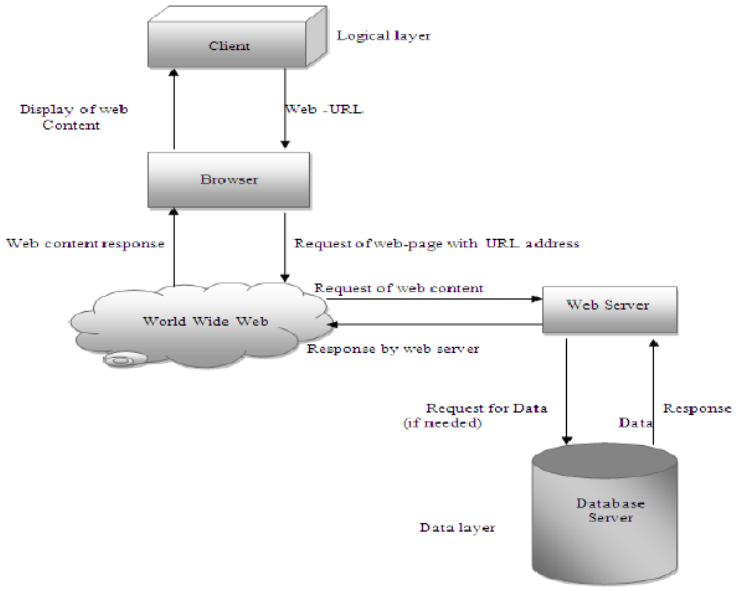


Figure 1: Architectural Design

3.2 Decomposition Description

The student grade calculation system's architecture is organized into several distinct layers of software components that interact with one another to deliver the required functionality. A detailed outline of the system architecture includes:

1. **Presentation Layer**: This layer encompasses the user interface elements that facilitate user interaction with the system. The user interface is a web application operating on a web browser, carefully designed using HTML, CSS, and JavaScript to ensure an intuitive and responsive experience for the end-users. The presentation layer ensures that users can easily access and manage their data through a user-friendly interface.

2. **Application Layer**: This layer houses the application logic responsible for processing user requests, handling business rules, and communicating with the database server for data retrieval or modification. The application layer is developed using PHP and operates on an Apache web server, ensuring efficient

and secure data processing. Additionally, this layer employs server-side validation, error handling, and user authentication, providing a robust and secure environment for the system.

3. **Data Access Layer**: This layer contains the database server and the associated software components that communicate with it for storing, retrieving, and managing data. The database server employed is MySQL, offering a reliable and scalable solution for handling large amounts of data. The software components interacting with the database server are designed using SQL queries, encapsulating the database-related operations within this layer. This separation of concerns promotes maintainability and allows for easier upgrades or changes to the database system.

4. **Network and Security**: The system architecture also emphasizes network and security aspects, ensuring that data transmission between the clients and the server is secure and reliable. Security measures such as encryption, firewalls, and access control mechanisms are employed to protect the system from unauthorized access and data breaches.

3.3 Design Rationale

The student grade calculation system is designed using a layered architecture, which separates various components of the system. This allows for improved scalability, maintainability, and security in the management and processing of student grades. By following best practices within each layer, the system is able to provide a robust and efficient environment in the context of the client-server model.

4 Data Design

4.1 Data Description

Our software system efficiently stores and organizes student grade data by employing two main data structures and various algorithms for data retrieval and manipulation. The first data structure is an associative array in PHP, which maps unique keys like ID, Name, Test 1 Grade, Test 2 Grade, Test 3 Grade, and Final Exam Grade to specific student values. This enables easy access and modification of the student's grade data when needed. The second data structure comprises arrays containing individual elements of the student's grades, such as Test 1, Test 2, and Test 3. These arrays help keep track of the different grades and calculate the student's final grade.

In terms of algorithms, our system utilizes functions like storeData() in the newCourse.php file. This function calculates the final grade by considering the attributed weights for each unique test grade using the formula: **$finalGrade = ($test1Grade * .2) + ($test2Grade * .2) + ($test3Grade * .2) + ($finalTestGrade * .4)** By taking the varying weights and grades inputted by users into account, the function calculates the overall grade, which is then inserted into the final grade data table. Together, these data structures and algorithms make it easy to access and manipulate student grade data as needed.

5 Component Design

5.1 Component Design Overview

In the component design overview of this software, there are two primary elements: the web server and the database. Utilizing Apache as the web server, it processes incoming user requests from web browsers and generates corresponding responses. The web server's responsibilities include rendering the user interface, which features a login page and a main page that presents the student name table and the student course table, both of which influence the final student grade table. Users can input data into the student and course tables via the web server's provided interface.

Additionally, the web server interface manages communication with the database server for data retrieval and storage. Utilizing MySQL as the database server, it is responsible for storing and accessing the system's data. The database contains two tables - ”Name Table” and ”Course Table” - which contribute to the calculation of a student's final grade. The interface connecting the web server and database ensures smooth communication between the two components.

In summary, the web server and database components cooperate to deliver a web-based interface, allowing users to interact with the database and view the final student grades.

5.2 Database Entity-Relationship Diagram

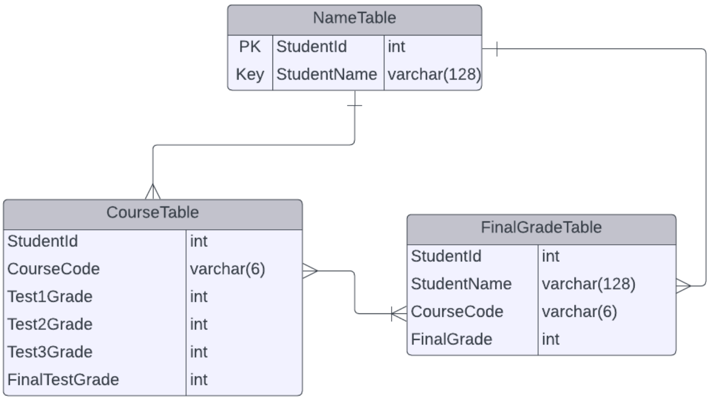


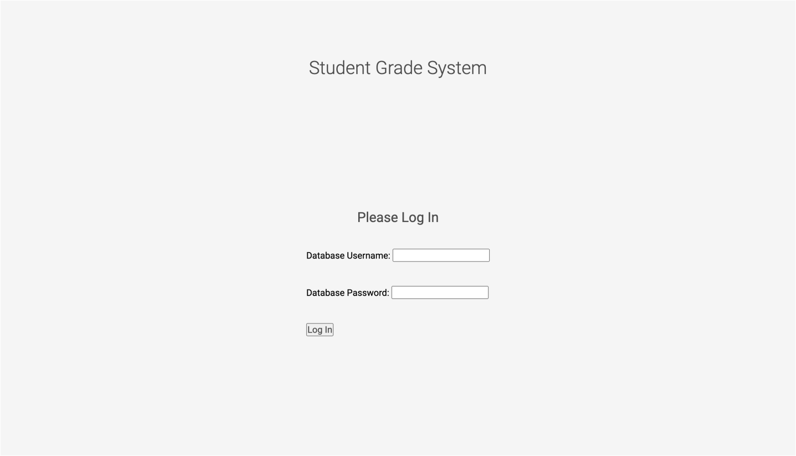
Figure 2: Database Entity-Relationship Diagram

6 User Interface Design

6.1 Overview of User Interface

Describe the functionality of the system from the user s perspective. Explain how the user will be able to use your system to complete al the expected features and the feedback information that will be displayed for the user.

6.2 Screen Images



Student Grade System

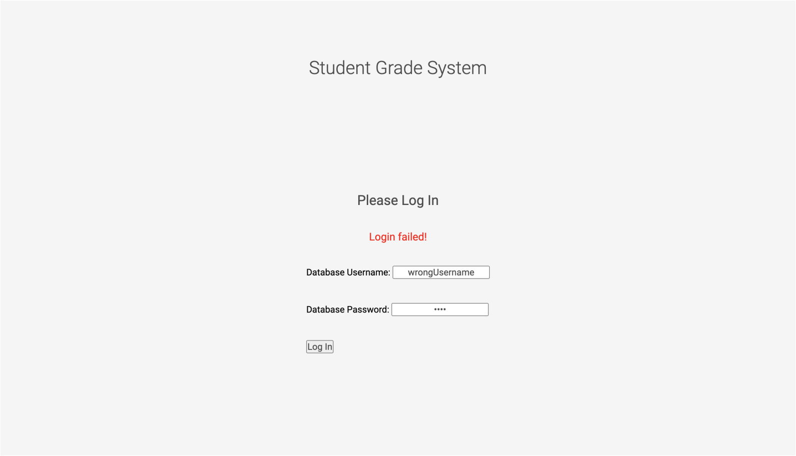
Please Log In

Database Username:

Database Password:

Log In

Figure 3: Login Page



Student Grade System

Please Log In

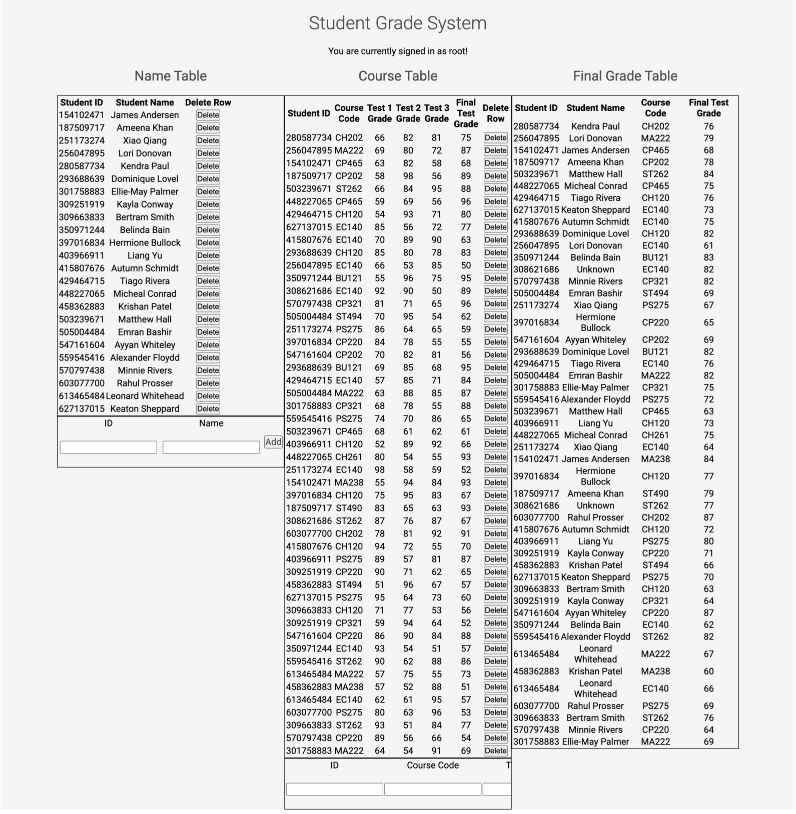
Login failed!

Database Username: wrongUsername

Database Password: ••••

Log In

Figure 4: Failed Login



Student Grade System

You are currently signed in as root!

Name Table

Student ID	Student Name	Delete Row
154102471	James Andersen	Delete
187509717	Ameena Khan	Delete
251173274	Xiao Qiang	Delete
256047895	Lori Donovan	Delete
280587734	Kendra Paul	Delete
293688639	Dominique Lovel	Delete
301758883	Ellie-May Palmer	Delete
309251919	Kayla Conway	Delete
309663833	Bertram Smith	Delete
350971244	Belinda Bain	Delete
397016834	Hermione Bullock	Delete
403966911	Liang Yu	Delete
415807676	Autumn Schmidt	Delete
429464715	Tiago Rivera	Delete
448227065	Micheal Conrad	Delete
458362883	Krishan Patel	Delete
503239671	Matthew Hall	Delete
505004484	Emran Bashir	Delete
547161604	Ayyan Whiteley	Delete
559545416	Alexander Floydd	Delete
570797438	Minnie Rivers	Delete
603077700	Rahul Prosser	Delete
613465484	Leonard Whitehead	Delete
627137015	Keaton Sheppard	Delete

Course Table

Student ID	Course Code	Test 1 Grade	Test 2 Grade	Test 3 Grade	Final Test Grade	Delete Row
280587734	CH202	66	82	81	75	Delete
256047895	MA222	69	80	72	87	Delete
154102471	CP465	63	82	58	68	Delete
187509717	CP202	58	98	56	89	Delete
503239671	ST262	66	84	95	88	Delete
448227065	CP465	59	69	56	96	Delete
429464715	CH120	54	93	71	80	Delete
627137015	EC140	85	56	72	77	Delete
415807676	EC140	70	89	90	63	Delete
293688639	CH120	85	80	78	83	Delete
256047895	EC140	66	53	85	50	Delete
350971244	BU121	55	96	75	95	Delete
308621686	EC140	92	90	50	89	Delete
570797438	CP321	81	71	65	96	Delete
505004484	ST494	70	95	54	62	Delete
251173274	PS275	86	64	65	59	Delete
397016834	CP220	84	78	55	55	Delete
547161604	CP202	70	82	81	56	Delete
293688639	BU121	69	85	68	95	Delete
429464715	EC140	57	85	71	84	Delete
505004484	MA222	63	88	85	87	Delete
301758883	CP321	68	78	55	88	Delete
559545416	PS275	74	70	86	65	Delete
503239671	CP465	68	61	62	61	Delete
403966911	CH120	52	89	92	66	Delete
448227065	CH261	80	54	55	93	Delete
251173274	EC140	98	58	59	52	Delete
154102471	MA238	55	94	84	93	Delete
397016834	CH120	75	95	83	67	Delete
187509717	ST490	83	65	63	93	Delete
308621686	ST262	87	76	87	67	Delete
603077700	CH202	78	81	92	91	Delete
415807676	CH120	94	72	55	70	Delete
403966911	PS275	89	57	81	87	Delete
309251919	CP220	90	71	62	65	Delete
458362883	ST494	51	96	67	57	Delete
627137015	PS275	95	64	73	60	Delete
309663833	CH120	71	77	53	56	Delete
309251919	CP321	59	94	64	52	Delete
547161604	CP220	86	90	84	88	Delete
350971244	EC140	93	54	51	57	Delete
559545416	ST262	90	62	88	86	Delete
613465484	MA222	57	75	55	73	Delete
458362883	MA238	57	52	88	51	Delete
613465484	EC140	62	61	95	57	Delete
603077700	PS275	80	63	96	53	Delete
309663833	ST262	93	51	84	77	Delete
570797438	CP220	89	56	66	54	Delete
301758883	MA222	64	54	91	69	Delete

Final Grade Table

Student ID	Student Name	Course Code	Final Test Grade
280587734	Kendra Paul	CH202	76
256047895	Lori Donovan	MA222	79
154102471	James Andersen	CP465	68
187509717	Ameena Khan	CP202	78
503239671	Matthew Hall	ST262	84
448227065	Micheal Conrad	CP465	75
429464715	Tiago Rivera	CH120	76
627137015	Keaton Sheppard	EC140	73
415807676	Autumn Schmidt	EC140	75
293688639	Dominique Lovel	CH120	82
256047895	Lori Donovan	EC140	61
350971244	Belinda Bain	BU121	83
308621686	Unknown	EC140	82
570797438	Minnie Rivers	CP321	82
505004484	Emran Bashir	ST494	69
251173274	Xiao Qiang	PS275	67
397016834	Hermione Bullock	CP220	65
547161604	Ayyan Whiteley	CP202	69
293688639	Dominique Lovel	BU121	82
429464715	Tiago Rivera	EC140	76
505004484	Emran Bashir	MA222	82
301758883	Ellie-May Palmer	CP321	75
559545416	Alexander Floydd	PS275	72
503239671	Matthew Hall	CP465	63
403966911	Liang Yu	CH120	73
448227065	Micheal Conrad	CH261	75
251173274	Xiao Qiang	EC140	64
154102471	James Andersen	MA238	84
397016834	Hermione Bullock	CH120	77
187509717	Ameena Khan	ST490	79
308621686	Unknown	ST262	77
603077700	Rahul Prosser	CH202	87
415807676	Autumn Schmidt	CH120	72
403966911	Liang Yu	PS275	80
309251919	Kayla Conway	CP220	71
458362883	Krishan Patel	ST494	66
627137015	Keaton Sheppard	PS275	70
309663833	Bertram Smith	CH120	63
309251919	Kayla Conway	CP321	64
547161604	Ayyan Whiteley	CP220	87
350971244	Belinda Bain	EC140	62
559545416	Alexander Floydd	ST262	82
613465484	Leonard Whitehead	MA222	67
458362883	Krishan Patel	MA238	60
613465484	Leonard Whitehead	EC140	66
603077700	Rahul Prosser	PS275	69
309663833	Bertram Smith	ST262	76
570797438	Minnie Rivers	CP220	64
301758883	Ellie-May Palmer	MA222	69

Figure 5: User Interface with Data

6.3 User Interface Actions Overview

The system follows the rules of CRUD actions, CRUD is the Create, Read, Update and Delete actions of a database. In the user interface, we created multiple buttons to perform these actions. These actions are seen in **Figure 5** - the main page and user interface of the program.

7 References

Gao, L. (2023). CP476 Internet Computing-week-6-2-php-summary [PowerPoint slides]. Internet Computing. Retrieved April 7, 2023, from MyLearningSpace website: https://mylearningspace.wlu.ca/d2l/le/content/468503/viewContent/3285839/View

InterviewBit. (2021). What is Client-Server Architecture? Retrieved April 10, 2023, from https://www.interviewbit.com/blog/client-server-architecture/

Ranjan, A., Kumar, R., Dhar, J. (2012). A Comparative Study between Dynamic Web Scripting Languages [Figure 1: Client-Server Architecture]. Scientific Figure on ResearchGate. Retrieved April 10, 2023 from https://www.researchgate.net/figure/Client-Server-Architecture_fig1_256404467