Criterion B

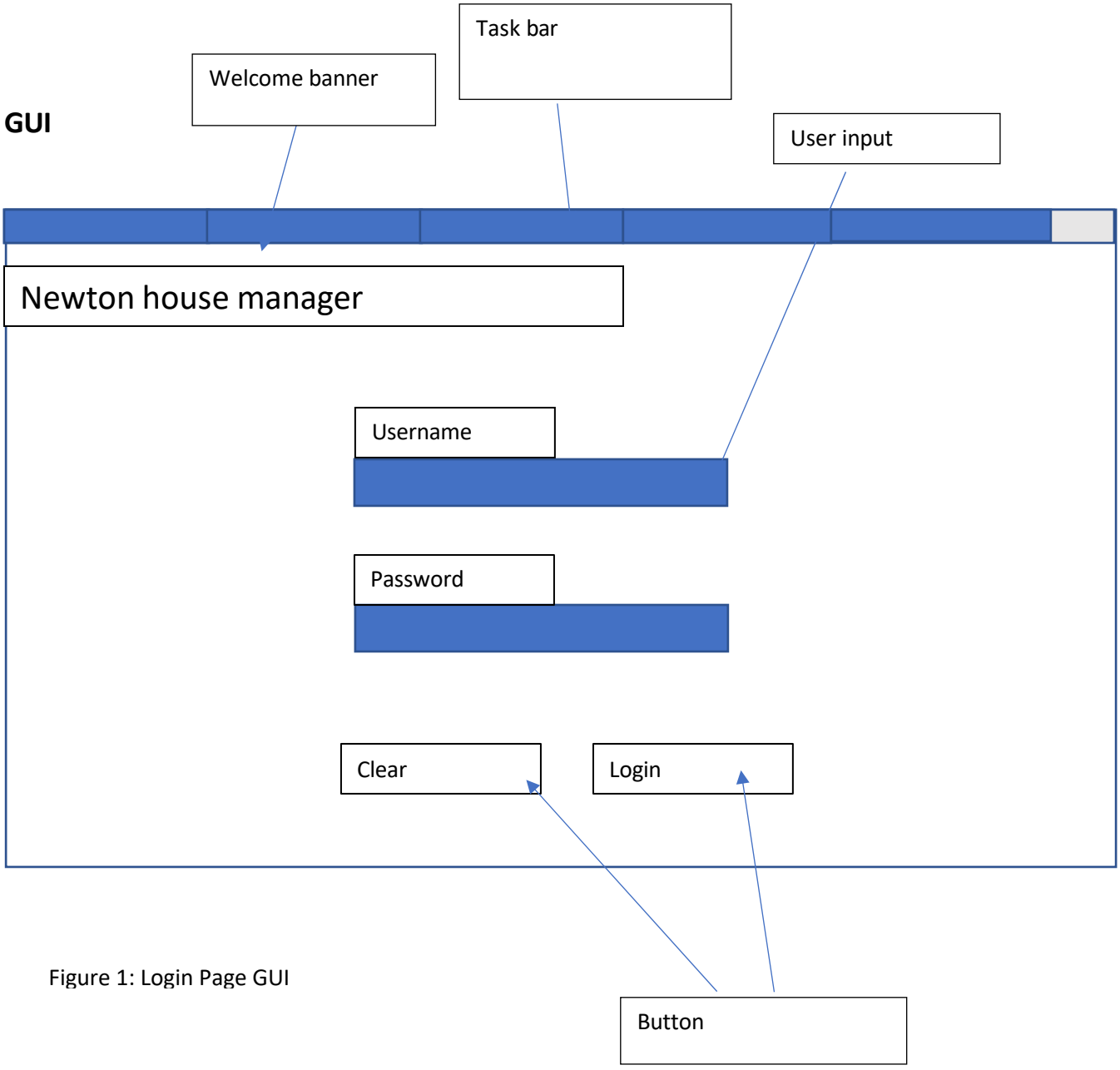

Figure 1: Login Page GUI

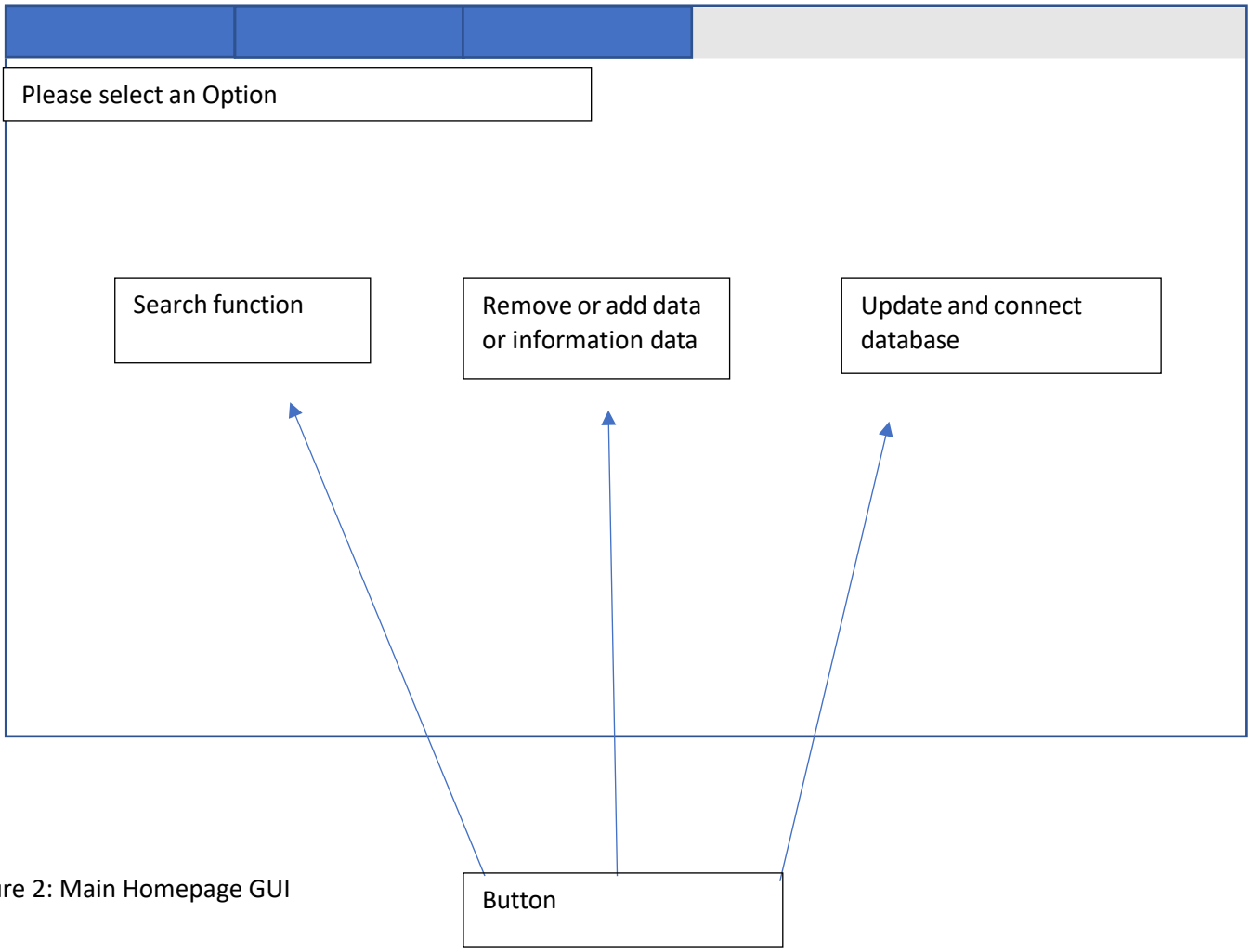

Figure 2: Main Homepage GUI

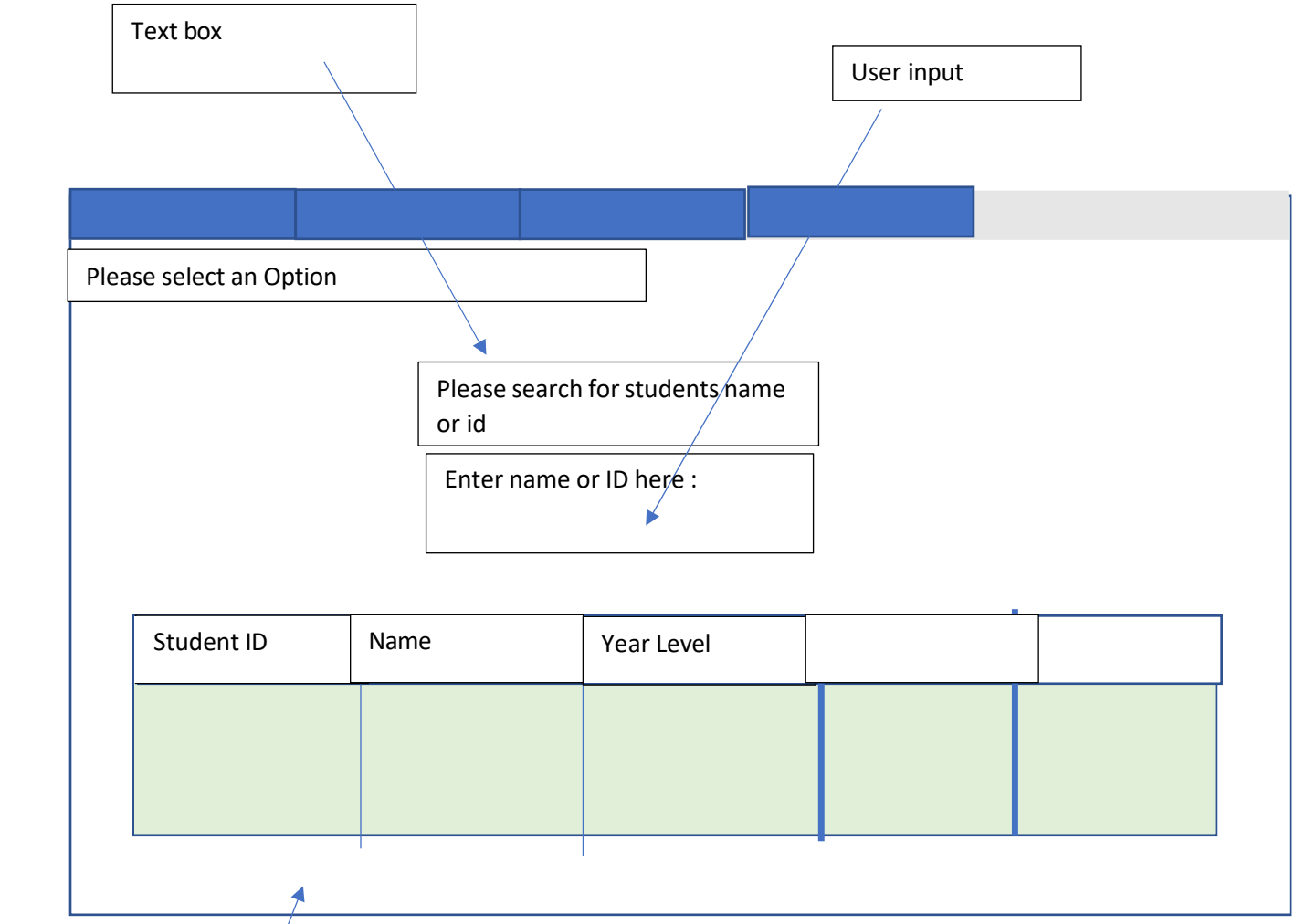

Figure 3: Database view GUI

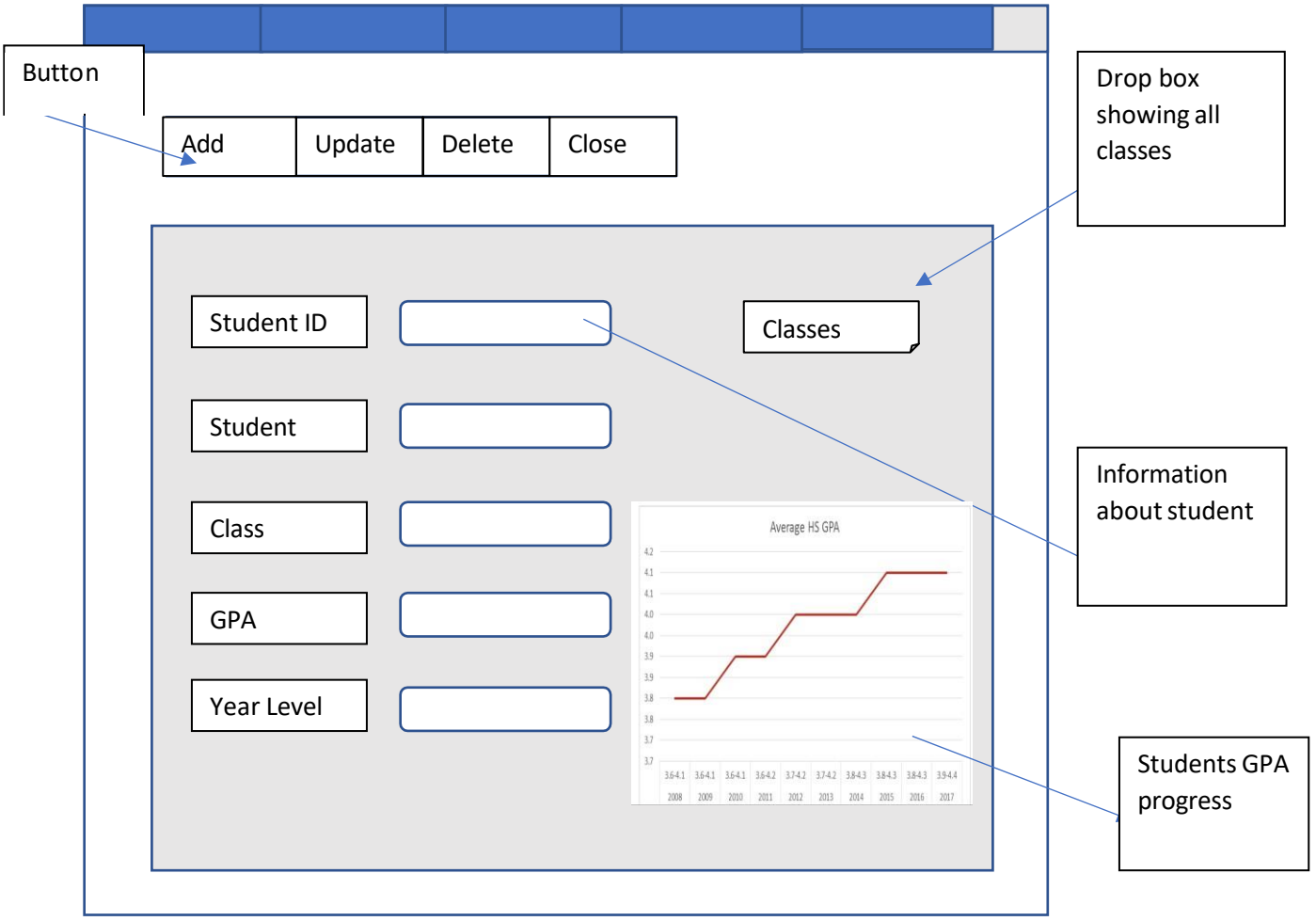

Figure 4: Student display GUI

Flow charts

System overview

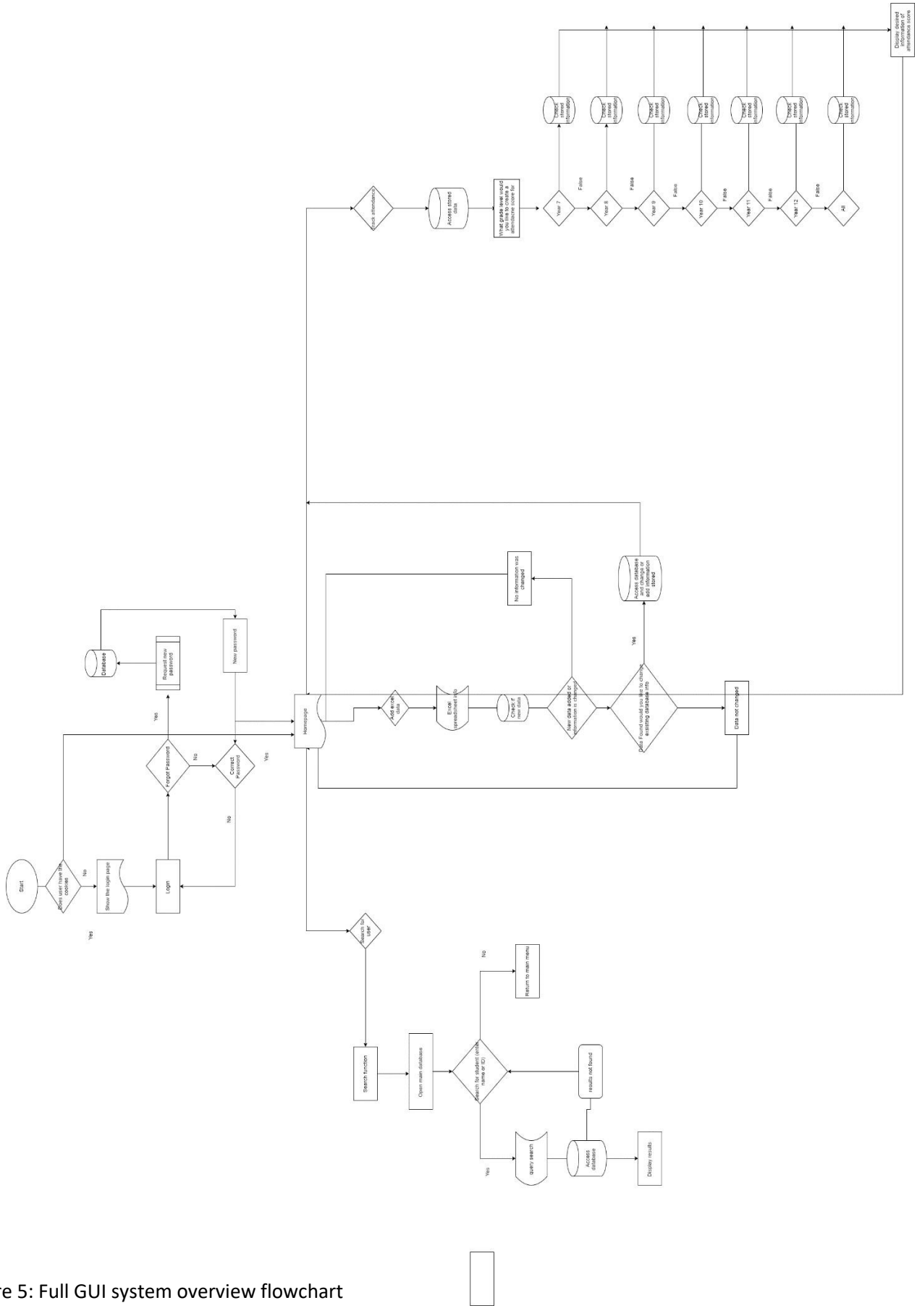

Figure 5: Full GUI system overview flowchart

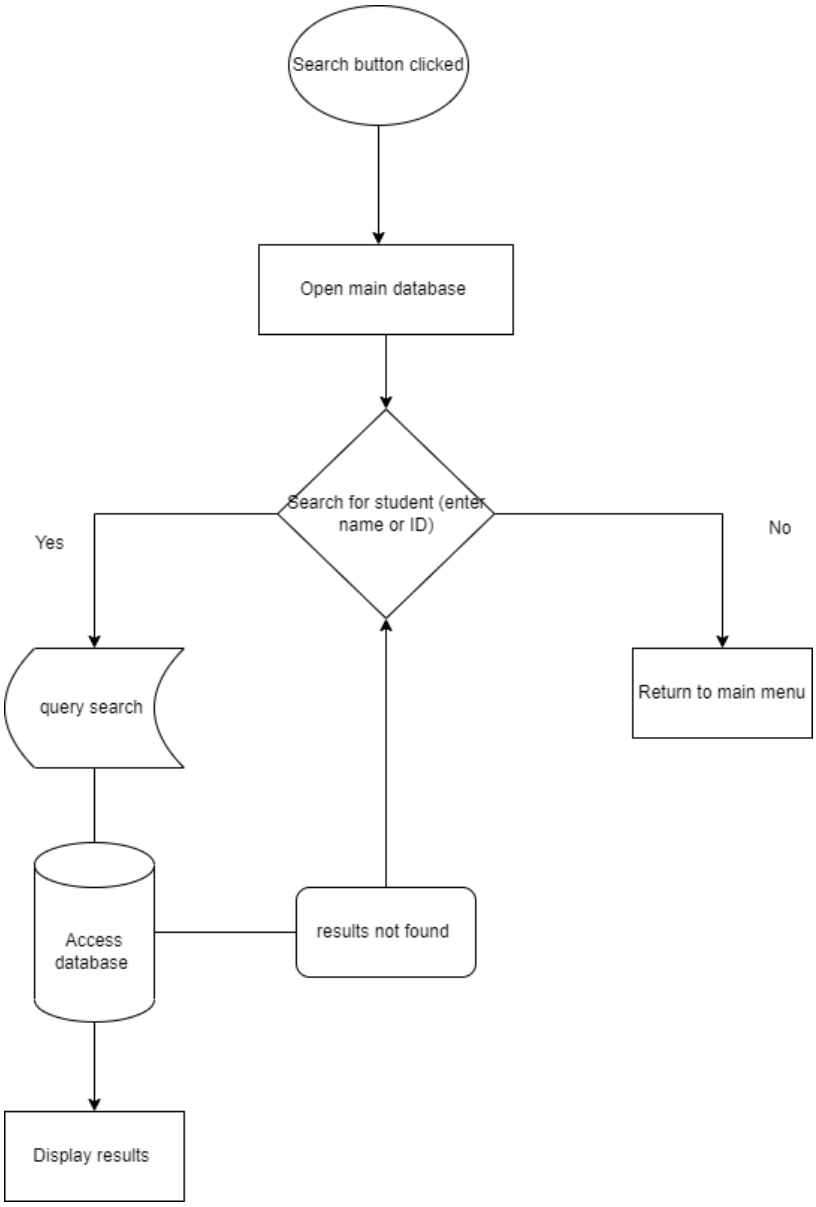

Figure 6: Search function flowchart

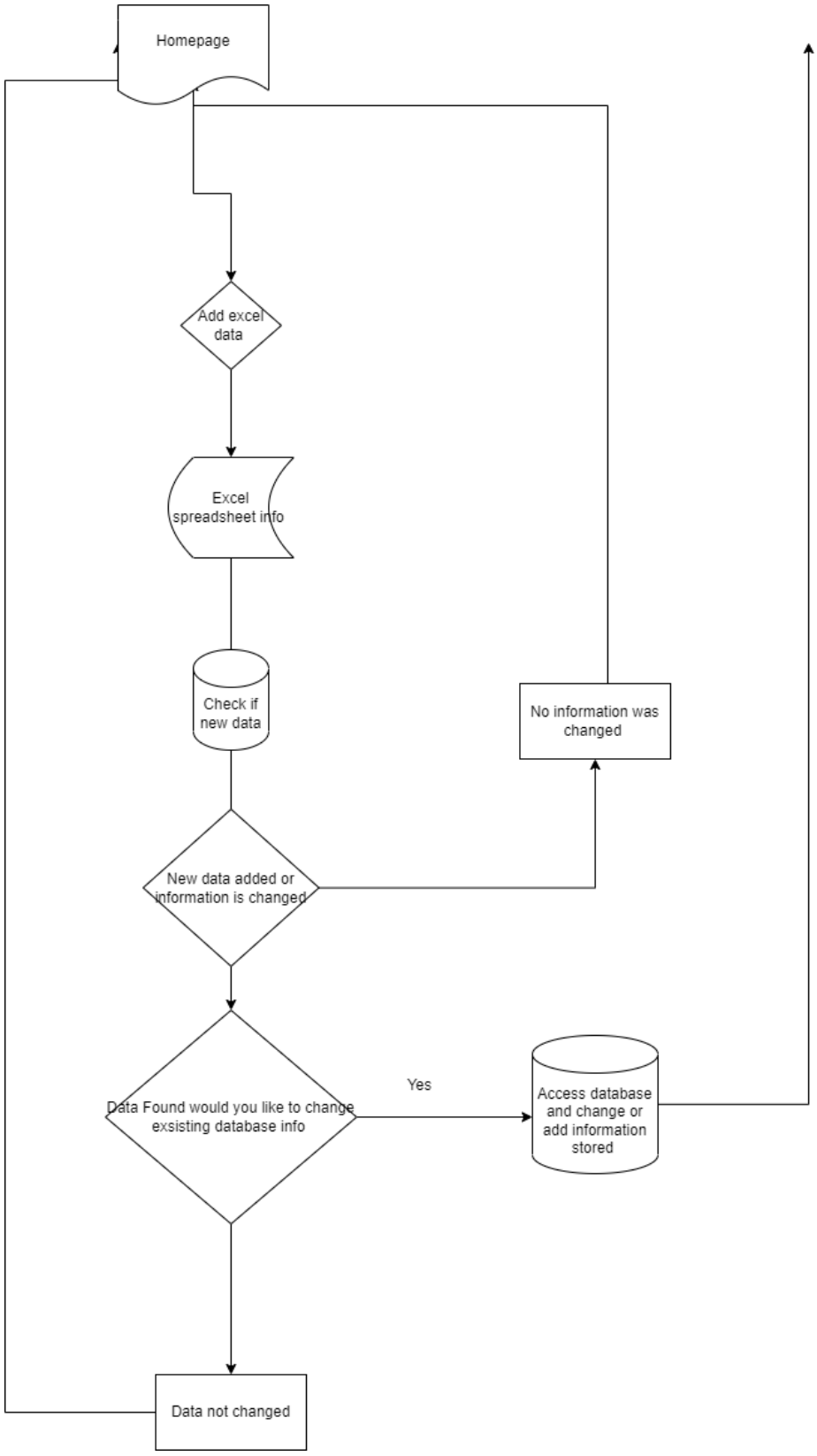

Figure 7 :Excel database import flowchart

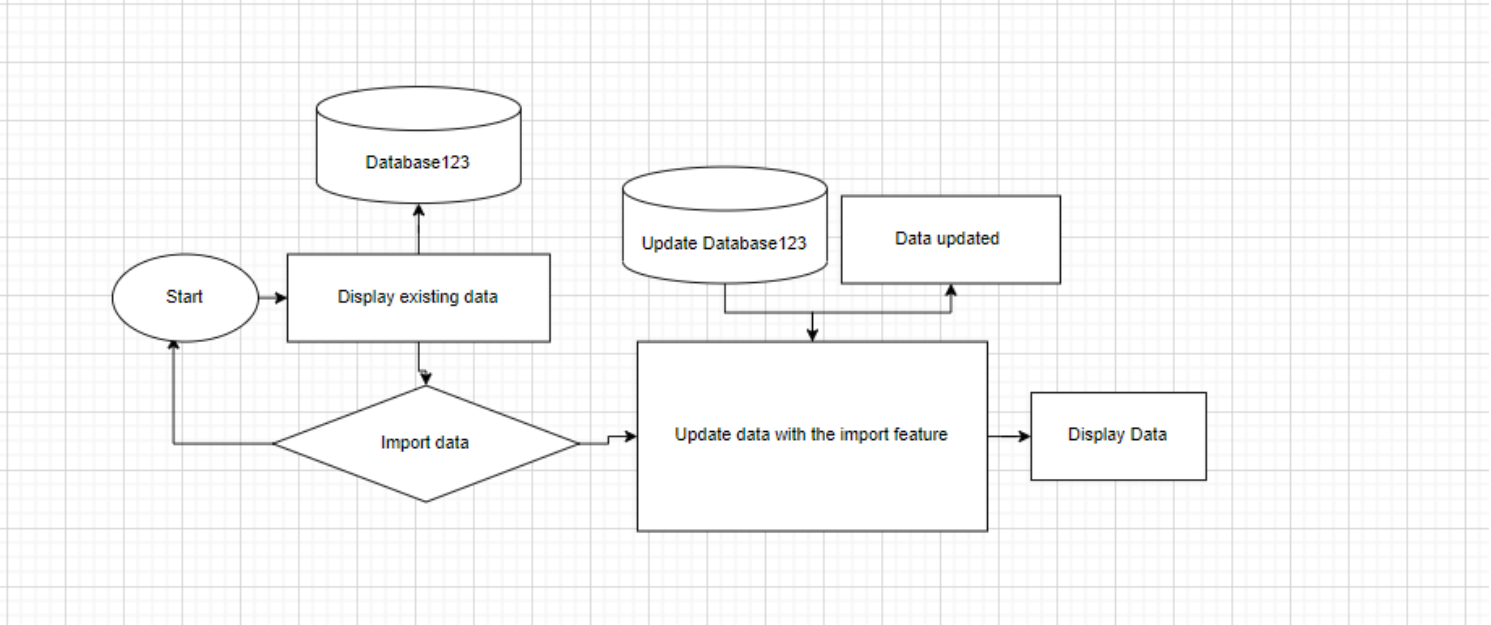

Figure 8 :Update and display all data flowchart

Table of classes

Table 1: All classes and brief description

Class	Purpose
Index.php	This file serves as the main entry point of the application. It contains the link to all other files which the user can access and also displays existing database.
Config.php	This file contains configuration settings for the application, such as database credentials, base URLs, and other environment-specific variables.
Display.php	The purpose of this file is not clear based on the information provided. It is a file responsible for rendering specific content or data on a webpage.
Graph.php	This file is likely responsible for generating and displaying graphs or charts based on the provided data. It utilises a graphing library or custom functions to visualize data.
Search.php	This file handles the search functionality within the application. It receives user input, queries the database, and displays the search results.

Javascript.js	This file contains JavaScript code that adds interactivity and functionality to the web pages. It handles client-side validation, AJAX requests, or other dynamic behavior and or data sorting.
Javascriptencrypt.js	The purpose of this file is not clear based on the information provided. It is a JavaScript file that handles encryption or cryptographic operations which has private student information.
Import.php	This file handles the process of importing data from external sources, such as Excel files, into the application's database. It contains code for reading the file, parsing data, and inserting it into the appropriate database tables.
Style.css	This file contains CSS code that defines the styles and visual appearance of the web pages. It controls the layout, colors, fonts, and other aspects of the user interface.
Composer.json	This file is used by Composer, a dependency management tool for PHP, to specify the project's dependencies and their versions. It defines the PHP packages that the application relies on and allows for easy installation and management of those dependencies.
PHPexcel and Vendor File	PHPExcel and Vendor files: These directories contain external dependencies and libraries used by the application. PHPExcel is a PHP library for working with Excel files, and the Vendor directory typically contains Composer-installed packages and their associated files.

Table 2: All methods and a brief description

Method	Description
Connect()	Establishes a connection to the database.
Querey ()	Executes a SQL query on the database.
Fetch ()	Retrieves a row or multiple rows from a database result set.
Insert ()	Inserts data into a database table.
Update()	Inserts data into a database table.
Delete()	Deletes records from a database table.
Fopen()	Opens a file for reading or writing.
Fwrite ()	Opens a file for reading or writing.
Fclose()	Opens a file for reading or writing.
Echo()	Outputs data to the browser.
Includer()	Includes a separate HTML or PHP file in the current file.
Header ()	Sets HTTP headers for redirection or other purposes.
Isset()	Checks if a form input field is set or

	has a value.
$_POST[]	Retrieves data sent via HTTP POST method.
$_GET[]	Retrieves data sent via HTTP GET method.
Jsonencode()	Converts a PHP array or object to a JSON string.
json_decode()	Converts a JSON string to a PHP array or object.
Array_Push()	Converts a JSON string to a PHP array or object.
Array_merge()	Merges multiple arrays into a single array.
generateGraphData($data)	This function generates the data required for creating the graph. It takes an array of data as input and returns an array of graph data.
generateGraph($graphData)	This function generates the actual graph using the provided graph data. eIt takes the graph data as input and renders the graph on the web page.
getGraphData()	This method retrieves the data from the database and prepares it for graph generation. It queries the database and fetches the necessary data.
getStudentData()	This method retrieves the student data from the database. It queries the database and fetches the required information about the students.

UML

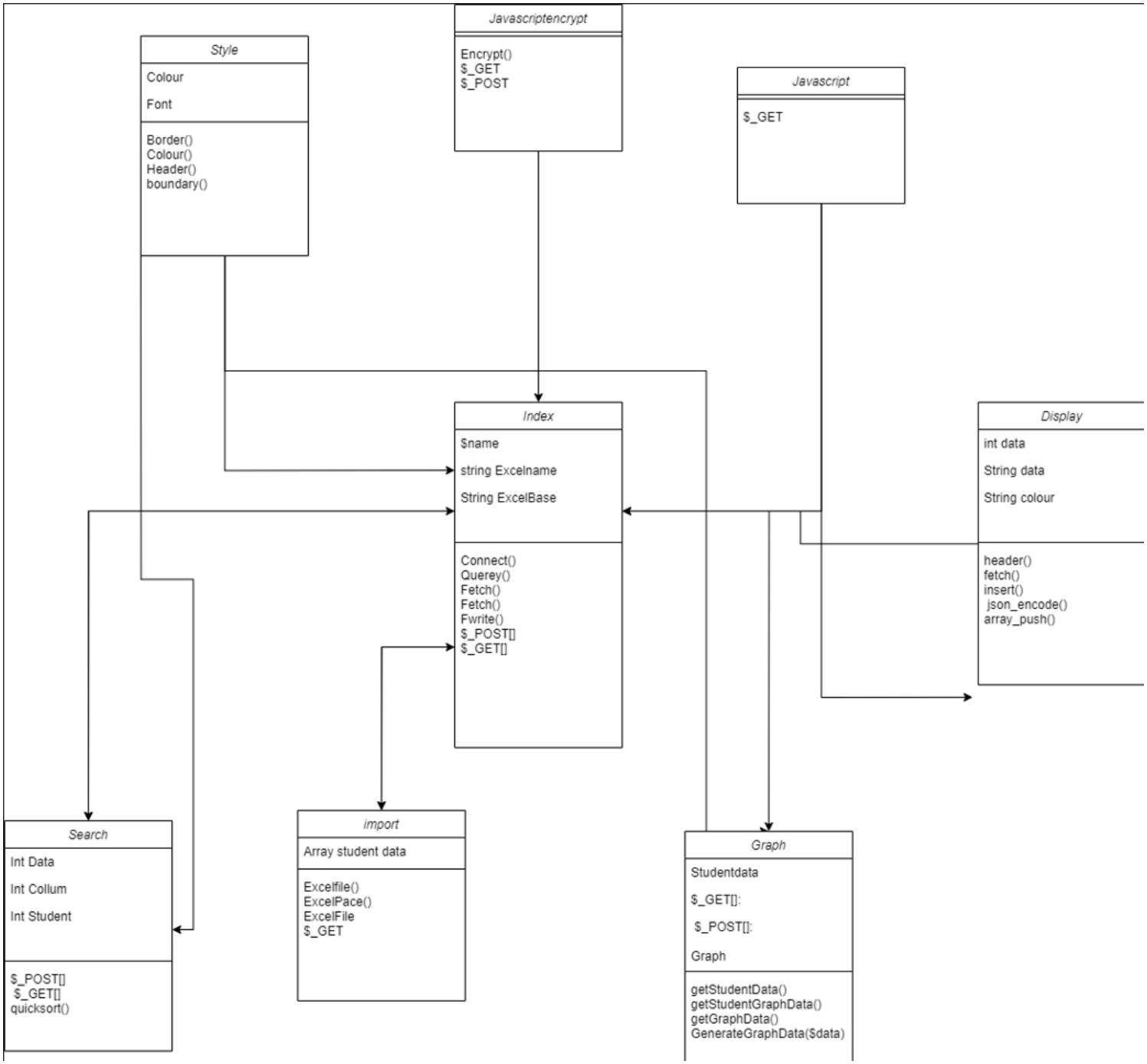

Figure 9: UML and interaction diagram

Justification of data structures

Table 3 – Data structures

Data structure	Data to be held	Justification
MySQL	All data(Students,grades,academic progression, attendance, detentions)	MySQL is an open-source database system that is reliable, fast, and easy to use. It is highly scalable and can handle large amounts of data. It is also platform-independent and can be used on many different operating systems. MySQL is secure and offers a wide range of features such as triggers, stored procedures, and views to help developers create powerful applications.

Test Plan

Table 4: Test plan

Test type	Description of test	Test	Expected outcome
1	Website and Database Connection	Verify if the client can access the website and successfully connect to the MySQL database.	The client should be able to access the website and establish a connection to the database without any errors.
2	Data Display in HTML	Check if the MySQL table data is displayed correctly on the HTML file, and if changes in the database are reflected in real-time on the webpage.	The client should see theaccurate representation of the data from the MySQL table on the HTML page, and any updates or additions to the database should be immediately visible on the webpage.
3	Excel File Import	Validate the functionality of importing an Excel file and adding its data to the MySQL database.	After importing the Excel file, the client should see the data successfully added to the MySQL database without any duplication or data integrity issues.

4	Double Entry Prevention	Ensure that the system prevents the client from adding duplicate data when adding new student records.	The client should receive a notification or error message when attempting to add a student record that already exists in the database, indicating that duplicate entries are not allowed.
5	Data Security and Encryption	Evaluate the security measures implemented to protect the client's data, ensuring that it is encrypted and secure when the web files are launched	The client's data should be encrypted and protected from unauthorized access or manipulation, ensuring its confidentiality and integrity.
6	Student Search Functionality	Verify if the client can successfully search for a student using either their ID or name, even when multiple students have the same name.	The client should be able to search for a specific student using their ID or name, and the system should accurately retrieve the relevant student information, distinguishing between students with the same name.
7	Student Search Functionality	Test the ability of the client to create, add, delete, and save notes for specific students, ensuring that the saved information persists and can be accessed later.	The client should be able to create, add, delete, and save notes for individual students, and the saved notes should be accessible and displayed when revisiting the student's profile.
8	Whole House Statistics	Verify if the client can access and view comprehensive statistics and data for the entire house, based on the client's search criteria.	The client should be able to search for specific criteria and view relevant statistics and data in a concise and meaningful manner, providing a comp
9	Detailed Student Information	Ensure that the client can search for a specific student and view all their grades and stored data in a	The client should be able to search for a specific student, and upon selection, view all their grades and relevant

		comprehensive manner.	stored data in a detailed and organized format.
10	Accurate Data Display	Confirm that the data displayed on the webpages accurately reflects the data stored in the MySQL database or Excel spreadsheet.	The data displayed on the webpages should mirror the data stored in the backend accurately, ensuring consistency and reliability in the information presented to the client.

Table 5: Record of Task

Task Number	Planned actions	Planned outcome	Time estimate	Target completion date	Criterion
1	Contact with client	Propose the solution to the client	15 minutes	31/10/2022	A
2	Meeting	Discuss the expectations for the produce and the requirements for satisfaction of product	30 minutes	17/11/2022	A
3	Discussion with supervisor	To check if products plan fits into requirements	20 minutes	17/11/2022	A
4	Creating a design for client	Demonstrate an idea for what the product will look like	40 minutes	20/11/2022	B
5	Second contact with supervisor	Check over function of design and demonstrating idea would work	25 minutes	30/11/2022	A

6	Basic GUI	Create GUI design to demonstrate appearance of code	10 minutes	8/12/2022	B
7	Flow chart	Create flow chart to show data flow and user flow	35 minutes	13/1/2023	B
8	UML diagram	Shows interactions between classes and databases	25 minutes	18/1/2023	B
9	Test plan	Show test cases and description of what will be tested and expected outcome	25 minutes	19/1/2023	BB
10	Update UML diagram	Update the Uml and system overview	15 minutes	28/1/2023	B
11	Second meeting with client	Learn more about what the client wants	25 minutes	4/2/2023	B
12	Update GUI	Update the GUI and get clients opinion	15 minutes	8/2/2023	B
13	Annotate GUI	Annotate based of clients feedback	20 minutes	15/2/2023	B
14	Code Import file	Code the import excel file	50 minutes	19/2/2023	C
15	Code JavaScript file	Code basic javascript for website	40 minutes	21/2/2023	C
16	Code display file	Code the display file so it displays tables and data	30 minutes	9/3/2023	C
17	Code index file	Code the main location page	20 minutes	16/3/2023	C
18	Code search file	Code the search algorithm and student sort	30 minutes	19/3/2023	C
19	Code style file	Code the basic	40 minute	24/3/2023	C

		appearance file			
20	Code JavaScript encrypt file	Code the file that encrypts student data	50 minutes	28/3/2023	C
21	Client feedback on product	Talk to client and get feedback of first version of the product	45 minutes	6/4/2023	C
22	Code graph file	Code the file which breaks down student data	35 minutes	10/4/2023	C
23	Update gui	Update gui based of data(Appendix B)	40 minutes	13/4/2023	C
24	Bug fixing	Fix bugs	25 minutes	16/4/2023	C
25	Install compiler json for excel	Install needed dependencies	10 minutes	20/4/2023	C
26	Install excel sheet reader	Install needed githup file	40 minutes	24/4/2023	C
27	Quick sort algorithm explained	Explained the ingenious or complex technique	30 minutes	26/4/2023	C
28	Graph data explained	Explained the ingenious or complex technique	25 minutes	28/4/2023	C
29	Encryption data explained	Explained the ingenious or complex technique	25 minutes	1/5/2023	C
30	Create dummy data	Create excel file dummy data	30 minutes	3/5/2023	C
31	Test final product and get feedback from client and peer	Get client feedback and test the final product	35 minutes	5/5/2023	C/E
32	Record crit D video	Record Crit D	15 minutes	9/5/2023	D
33	Write criterion E Documentation	Write and finish Code	35 minutes	11/5/2023	E
34	Submit	Zip and Submit product	20 minutes	12/5/2023	All