BTEC Higher National Diploma in Computing & Systems Development
Project Documentation

Institute Name: IDM Computer Studies Pvt. Ltd.

Unit Name: Software Development project

Unit Value: Unit 4

Lecturer:

Unit Outcomes:

1. **Be able to formulate a project**
2. **Be able to implement the project within agreed procedures and to specification**
3. **Be able to evaluate the project outcomes**
4. **Be able to present the project outcomes**

Grading Opportunities Available

Outcomes/ Grade Descriptors	**P4.1**	**P4.1**	**P4.3**	**P4.4**	**M1**	**M2**	**M3**	**D1**	**D2**	**D3**
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Date Set: **Due Date:**

Student Name: **Student No:**

Outcomes/Grade Descriptors Achieved (Please Tick)

Outcomes/ Grade Descriptors	**P4.1**	**P4.1**	**P4.3**	**P4.4**	**M1**	**M2**	**M3**	**D1**	**D2**	**D3**

Assessor Comments:

Signature: **Date: __/__/__**

IV Comments:

Signature: **Date: __/__/_**

STUDENT GRADE MONITORING SYSTEM (SGM)

ABSTRACT

A Student Grade Monitoring system (SGM) can be used to track the progression of students. This project is to implement a Students' Grade Monitoring System with Advanced analyzing graphs. This system is for school children from grade 6-11.

Tracking the marks of students, analyzing students' progression, and analyzing performance variation over the years in school are the main functionalities of this system.

A lot of graphs are added with the system and the user can select two parties to compare. Teachers and students are the target users of this system.

And instead of adding single student's subjects marks, the teacher can upload CSV file with student's marks. So it is a very efficient way of adding data to the database.

TABLE OF CONTENTS

TABLE OF FIGURES AND TABLES

CHAPTER 01

1 INTRODUCTION

1.1 INTRODUCTION

Students' Grade Monitoring System makes recording and calculating grades simple and efficient. Simply enter students' grades and let SGM do the rest analyzing. Teachers no longer have to spend hours over grade books, papers, reports, and calculators. SGM does all of the calculations and gives the analyzed graphs.

So teachers can monitor grades and be an active participant in every student's progression.

There is a huge issue in monitoring the progress of students in school. There are already a lot of tools/systems that keep student details and grades/marks. But the main issue in the current set up is there is no tool to analyze student progression and see how his/her performance varies over the years in school. So this system is designed due to the following requirements.

- Neediness to monitor students *individually*.
- To *compare* students' marks with their *previous* grades.
- Need to track weak students and the *turning point* of them.
- Compare the average marks of a student with a *class average*.
- View *all grades* for all subjects that he learnt.

So, this system is developed to

i. Track students' marks and
ii. Display the analyzed marks in various graphs which are fully customizable. (So that analyzing the student's progression is easy…)

1.2 RELATED WORK

There are already a lot of applications which can keep students' details and the records of grades. But the issue is there is no tool to analyze the grades and take decisions/predictions about the students' progression.

1.2.1 PowerSchool

PowerSchool empowers teachers and administrators to use real-time data insights and analysis to adjust instruction from the student to the school level. This system helps teachers to see where a student's strengths and weaknesses lie. PowerSchool helps students to understand their learning. It shows their progress, gives real-time feedback and scores and fosters engagement in learning through content and assignments. Then parents have a benefit from the transparency of their child's assignments, scores, and grades, getting real-time insight into their child's learning, growth, and weaknesses.

1.2.2 KIOSK –Student's grade monitoring system

KIOSK is a module that can easily provide the grades of the students even the faculty members are not present in the community. It only calculates the general average of the student and allows to view the previous grades also.

1.2.3 QuickSchools –GradeBook

QuickSchools Gradebook feature makes recording and calculating grades simple and efficient. Simply enter student grades and let QuickSchools do the rest. Fully customizable, you can weight individual assignments or categories to suit your class's needs, add and delete assignments, and redefine assignments and grade weighting. After new grades are entered, the overall class grade is averaged and updated.

These all applications record grades and information of students and give access to teachers and students (sometimes parents) to view previous and current grades. And also give the general average of the marks. There is no way to compare two students or to compare a student with his/her previous grades.

1.3 FEASIBILITY STUDY

An important outcome of the preliminary investigation is the determination that the system requested is feasible. A feasibility study is carried out to select the best system that meets the performance requirements.

A feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. It involves a preliminary investigation of the project and examines whether the designed system will be useful to the organization. Months or years of effort, thousand for millions of money and untold professional embarrassment can be averted if an in-conceived system is recognized early in the definition phase.

The different types of feasibility are Technical feasibility, Operational feasibility, Economical feasibility.

1.3.1 Technical feasibility

Technical Feasibility deals with the hardware as well as software requirements. Technology is not a constraint to type system development. We have to find out whether the necessary technology, the proposed equipment can hold the data, which is used in the project, should be checked to carry out this technical feasibility.

The technical feasibility issues usually raised during the feasibility stage of investigation includes these

- This software is running in Windows 2000 Operating System, which can be easily installed.
- The hardware required is a Pentium based server.
- The system can be expanded.

1.3.2 Behavioural Feasibility

This feasibility test asks if the system will work when it is developed and installed.

Operational feasibility in this project:

- The proposed system offers a greater level of user-friendliness.
- The proposed system produces the best results and gives high performance. It can be implemented easily. So this project is operationally feasible.

1.3.3 Economic feasibility

Economic Feasibility deals about the economic impact faced by the organization to implement a new system. Financial benefits must equal or exceed the costs. The cost of conducting a full system, including software and hardware cost for the class of application being considered should be evaluated. Economic Feasibility in this project:

- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

CHAPTER 02

2 SYSTEM ANALYSIS.

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to conclude. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

The preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. A preliminary study is a problem-solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be

obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 CURRENT SYSTEM DESCRIPTION

They use Gradebook as a current system and it is a new core tool for instructors to calculate and store grade information and distribute it to students online. It has been designed based on the same back-end as the Gradebook Classic tool, but with several new features and enhancements.

It provides

- Auto-calculate course grades, with the ability to override any course grade.
- Define course letter grades based on a 100% scale.
- Choose between point- or percentage-based grading.
- Add Gradebook items for manually graded or offline activities.
- Create categories to organize items and allow for weighting of grades.
- Enter, view, edit, and release to students scores, grades, and comments.
- Collect and display scores from tools such as Tests & Quizzes, Assignments, and Forums.
- Export scores and grades to Microsoft Excel (in XLS format).
- Import item scores from spreadsheet (CSV) files.
- Export a printer-friendly (PDF) version of student grades.
- Drop grades or keep the highest grades in a category.
- Specify items or categories as extra credit.

2.3 Drawbacks of the Current system

- It has additional functionalities that we do not need such as conducting assignments
- Domain knowledge required to handle the system.
- Cannot compare the ower result with other students
- More complex
- Paid licence
- Cannot get a good overview regarding the position in the class

2.4 User Requirement Specification

- Neediness to monitor students individually.
- To compare students' marks with their previous grades.
- Need to track weak students and the turning point of them.
- Compare the average marks of a student with a class average.

2.5 Proposed System

The method to solve this problem is to create a web-based Student Grade monitoring system. Here students can their progress in each year and each subject. And the Student will be able to compare their marks with among equal competitive in their class. A Teacher can upload marks CSV file in each term easily. Simple UI is used to develop the system. There are no fancy functionalities. Just students and their parents easily understand the weaken subjects and can work hard to get high marks.

- Make sure the student confidentiality
- It will ensure data accuracy.
- Minimum time needed for the various processing.
- It will provide a better Service.

2.6 OBJECTIVES AND SCOPE

2.6.1 Scope

The scope of this system is limited to students and teachers. Because of only teachers and students are allowed to access the system. And only the students in grade 6 to 11 are targeted. And also there is a limited number of choices for graphs in the dashboard.

2.6.2 Objectives

The main objective of this student grade monitoring system is to track the marks of each student and analyze those marks in a meaningful way. The way we used to analyze the marks is following.

- **a.** Track the student's progression for each subject
- **b.** Track the average of a student when he/she is different grades
- **c.** Track his/her marks for mathematics
- **d.** Compare the specify student with the marks of the best student in the class
- **e.** Track his position in the class in different grades
- **f.** Average marks for all subjects

Teachers can see the progression of each student individually and overall in parents meetings or other special occasions. Using these analyzed graphs teacher or parents can study whether there is a turning point or not. So teachers can keep touch in weak students like that.

2.7 FUNCTIONAL REQUIREMENT

Functional requirements define the fundamental actions that a system must perform.

- Authentication: To protect the privacy of information
- Displaying student information
- The flexibility of the user interface
- Faster processing
- View of previous data
- Display statistical report of a student
- Enable to compare with friends
- Displaying analyzing graphical charts
- Edit user profile
- Sessions
- Generating graphical reports

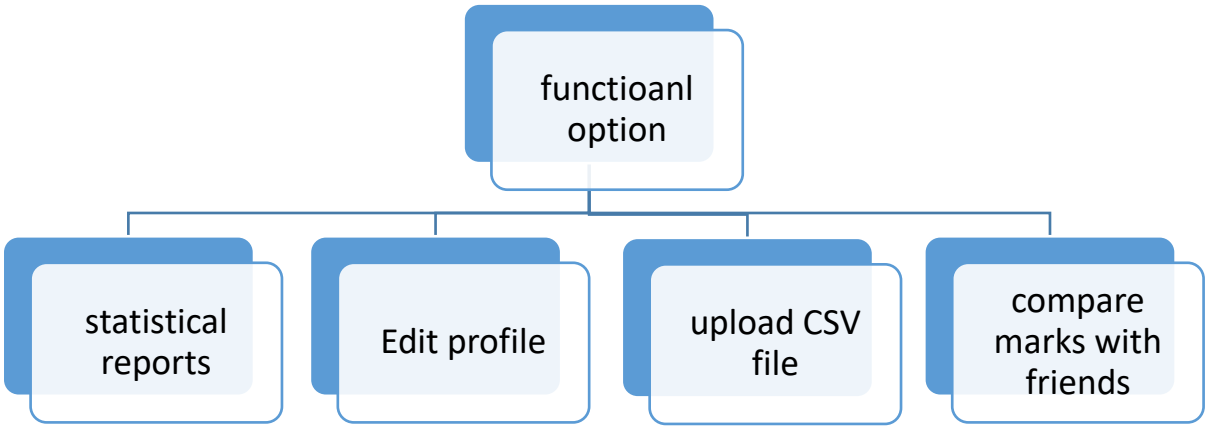


Figure 2: functional requirements

2.8 NON FUNCTIONAL REQUIREMENTS

Non-Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and as well as portability.

- Reliability is the ability of a system to perform its required functions under stated conditions for a specific period.
- Performance requirements concern the speed of operation of a system.
- Security Unauthorized access to the system and its data is not allowed.
- User Friendly is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or component.
- Modifiability Requirements about the effort required to make changes in the software
- The default username and password submitted for a student
- The system should be available for 24/7 hours.

2.9 HARDWARE REQUIREMENTS

- Processor: Pentium 4 @ 2.40 GHz or Higher
- RAM: 1GB
- Hard disk: 80 GB
- Monitor: 14” CRT Or LED
- Keyboard: Normal or Multimedia
- Mouse: Compatible mouse
- USB Flash Drive
- Printer

2.10 **SOFTWARE REQUIREMENTS**

- MY SQL database Server 5.5
- PHP 5.6
- Apache server
- Operating system: Windows XP, Windows 7, Windows 8

CHAPTER 03

3 SYSTEM DESIGN

3.1 OVERVIEW OF METHODOLOGY

The implementation of this system has done as follows.

Initially, requirements were gathered and milestones were planned. Then, the database was designed as follows. MYSQL is chosen to create the database schema and to handle the database. There are ten tables in this database system. It contains student, student users, classes, students with their class, teacher users, teachers, subjects, teachers with their subjects, marks.

CSV (comma-separated value) files were created to insert data to the database. These CSV files were uploaded to the database using PHP.

PHP would be the main programming language for the development of this system. PHP is a server-side language. PHP was chosen because it is a familiar language to us.

Then, Codeigniter (CI) was selected as the PHP framework. Codeigniter is an open-source PHP web application framework. It is a Model-View-Controller framework. Data were retrieved from the database and put them into JSON format. All graphs in this system are drawn using High Charts. HighCharts uses JSON format in adat. In HighCharts JQuery and JavaScript are used.

Finally, a web application was designed using CSS (Bootstrap). AJAX was also a big help in developing the system. It would make the system more interactive without too many refreshes when connecting to the server. The method involved is to be able to effortlessly update a section of the Web application.

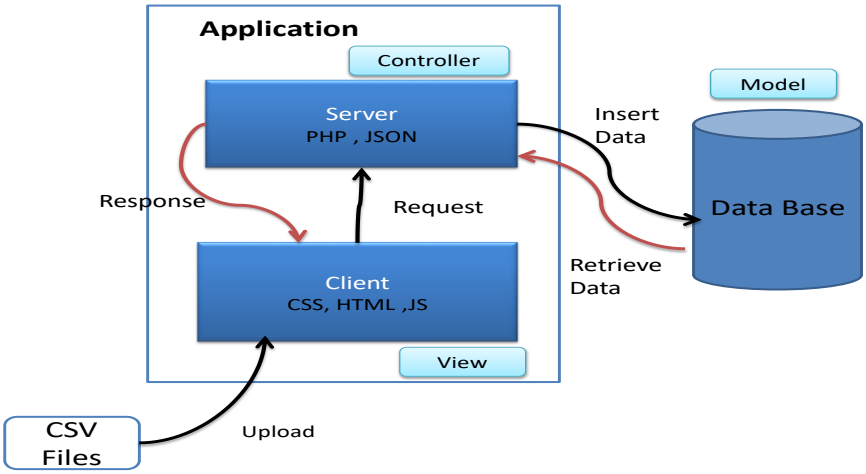


Figure 3:Abstract design

3.2 CSV FILE

CSV (Comma Separated Values) is a file format for data storage which looks like a text file. The information is organized with one record on each line and each field is separated by a comma. The file can be managed via Microsoft Excel (or similar programs) and lists the merchant's products, codes, image links, etc. The fields provided by default include path, id, name, code, price, headline and many others

3.2.1 Advantages of CSV files

- CSV is human-readable and easy to edit manually
- CSV is simple to implement and parse
- CSV is processed by almost all existing applications
- CSV provides a straightforward information schema
- CSV is faster to handle
- CSV is smaller in size

3.2.2 Format of project CSV file

- Marks for each value subject CSV file format

Student_index, subject_marks, term

- Student list CSV file format

Student_index, first_name, Last_name, Address, telephone

3.3 MVC Architecture

Since Codeigniter follows the MCV Architecture, For this project I used this common architecture pattern.

MVC is popular as it isolates the application logic from the user interface layer and supports the separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows. (tutorialspoint, n.d.)

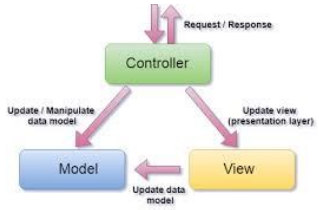


Figure 4:MCV Architecture

3.3.1 The Model

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

3.3.2 The View

It means the presentation of data in a particular format, triggered by a controller's decision to present the data. They are script-based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

3.3.3 The Controller

The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

3.4 USE CASE DIAGRAM

UML Use Case Diagrams can be used to describe the functionality of a system horizontally. That is, rather than merely representing the details of individual features of your system, UCDs can be used to show all of its available functionality. It is important to note, though, that UCDs are fundamentally different from sequence diagrams or flow charts because they do not make any attempt to represent the order or number of times that the actions and sub-actions of the system should be executed.

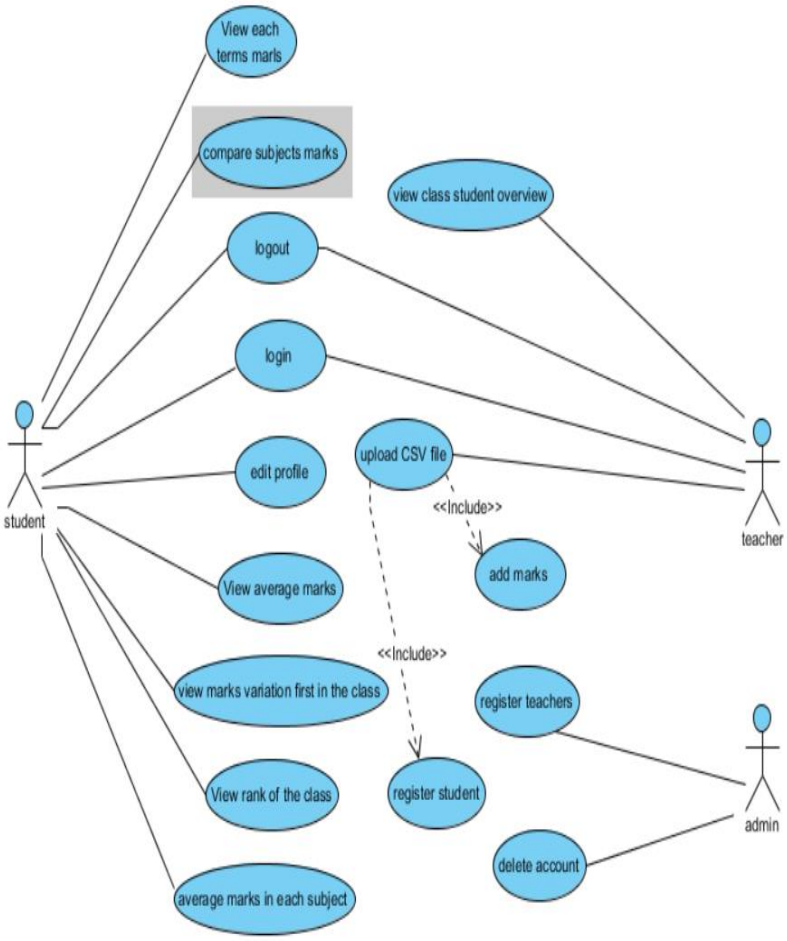


Figure 5:SGM system use case diagram

3.5 USER CASE DESCRIPTION

3.5.1 Login

1. Title:- login
2. Purpose/overview:-
    - Students able to see their grades
    - Teachers get to know their student
    - Upload marks
3. Actor:-

    Student, Teachers

4. Pre-Condition:-
    - The teacher creates a student account
    - Admin creates a student account
5. Post Condition:-
6. The normal flow of event:-
    - Student/Teacher login their username and password

3.5.2 View Rank of the class

1. Title:- view Rank of the class
2. Purpose/overview:-

    Student can measure their class rank

3. Actor:-

    Student

4. Pre-Condition:-

    The teacher creates a student account

5. Post Condition:-

    If student marks are available.

6. The normal flow of event:-
    - Student login their username and password
    - Automatically generate a graph of their rank in the class

3.5.3 View Each item subjects Marks

1. Title:- View each term subjects marks
2. Purpose/overview:-

   Students able to see their each subjects marks in each term

3. Actor:-

   Student

4. Pre-Condition:-

   The teacher creates a student account

5. Post Condition:-

   If student marks are available.

6. The normal flow of event:-

   - Student login their username and password
   - Automatically generate a graph of their each term marks

3.5.4 View Average marks of the student with class Average

1. Title:- Average marks of the student with the class average
2. Purpose/overview:-

   Students able to see their position in the class

3. Actor:-

   Students

4. Pre-Condition:-

   The teacher creates a student account

5. Post Condition:-

   If student marks are available.

6. The normal flow of event:-

   - Student login their username and password
   - Automatically generate a graph of their marks compare to the class average

3.5.5 View marks variation first in the class

1. Title:- View marks variation first in the class
2. Purpose/overview:-

   Students able to compare with the top of the class

3. Actor:-

   Student

4. Pre-Condition:-

   The teacher creates a student account

5. Post Condition:-

   If student marks are available.

6. The normal flow of event:-

   - Student login their username and password
   - Automatically generate a graph of their marks compared to the top of the class

3.5.6 View Average marks of each subject

1. Title:- view average marks of each subject
2. Purpose/overview:-

   Students able to see their position in the class

3. Actor:-

   Student

4. Pre-Condition:-

   The teacher creates a student account

5. Post Condition:-

   If student marks are available.

6. The normal flow of event:-

   - Student login their username and password
   - Automatically generate a graph of their marks each subject

3.5.7 Edit student profile

1. Title:- Edit student profile
2. Purpose/overview:-

   Students able to assign any username or password, they want instead of the default password

3. Actor:-

   Student

4. Pre-Condition:-

   The teacher creates a student account with a default password

5. Post Condition:-

   If student marks are available.

6. The normal flow of event:-
   - Student login their username and password
   - Edit their profile

3.6 ACTIVITY DIAGRAM

3.6.1 System activity Diagrams

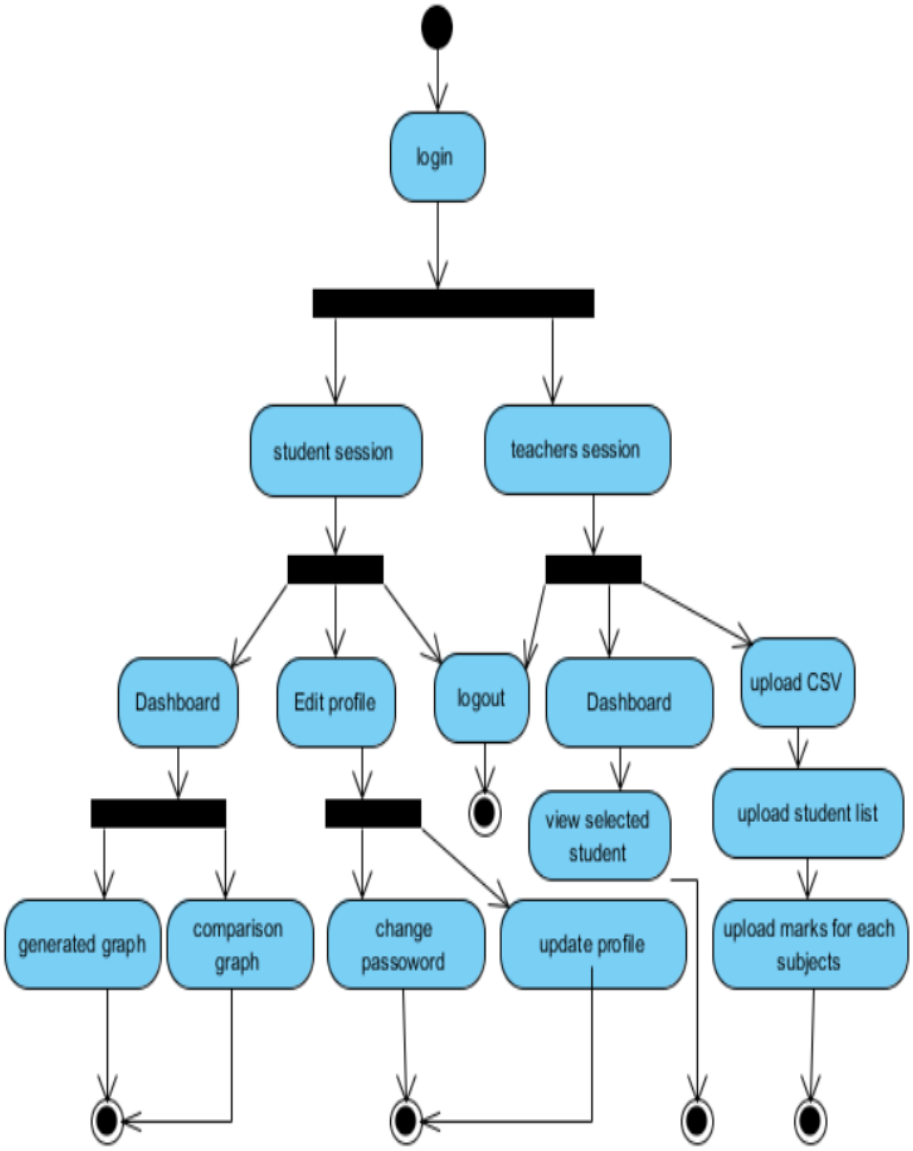


Figure 6: System activity diagram

3.5.2 **Activity Diagrams for subcomponent**

- Activity Diagram for the login use case

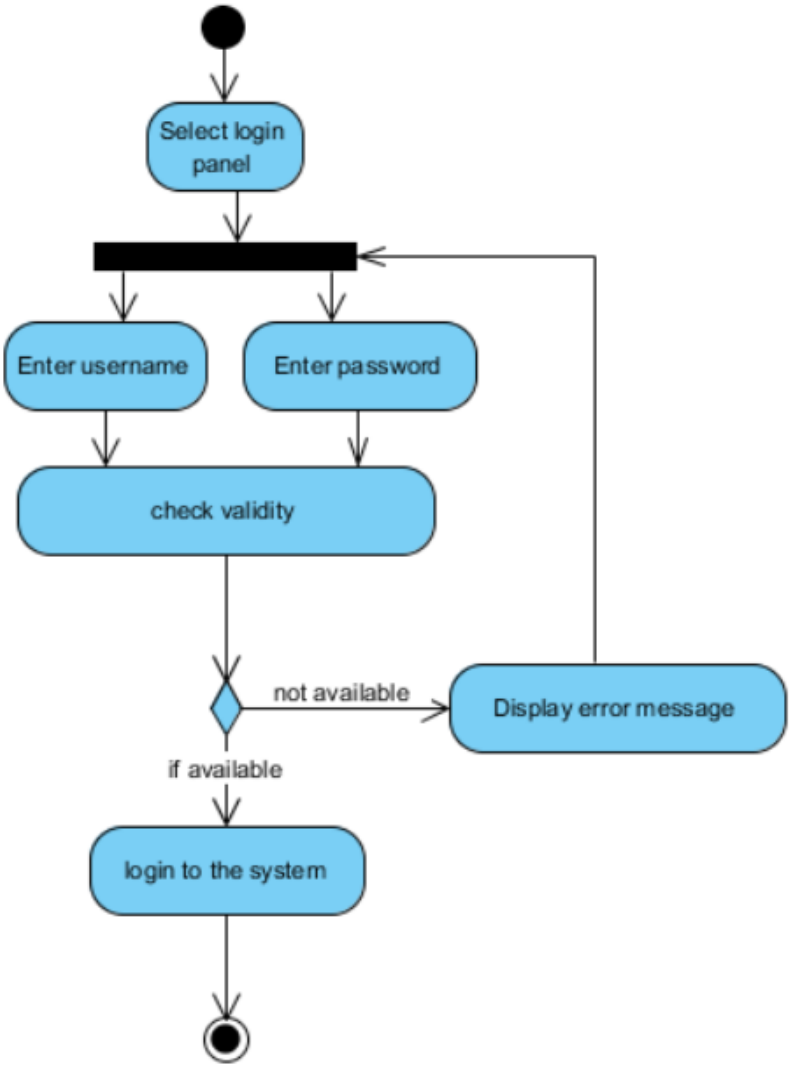


Figure 7: Activity Diagram for the login use case

- Activity Diagram for edit profile use case

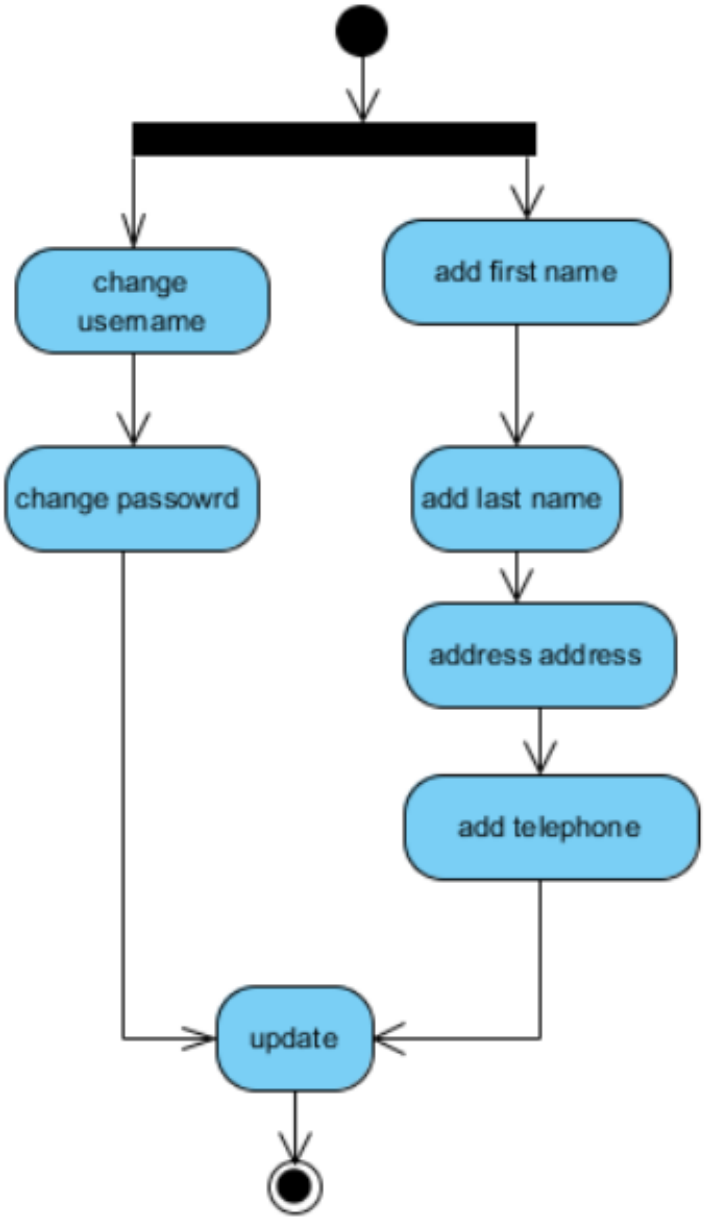


Figure 8: Activity Diagram for edit profile use case

- Activity Diagram for upload CSV file use case

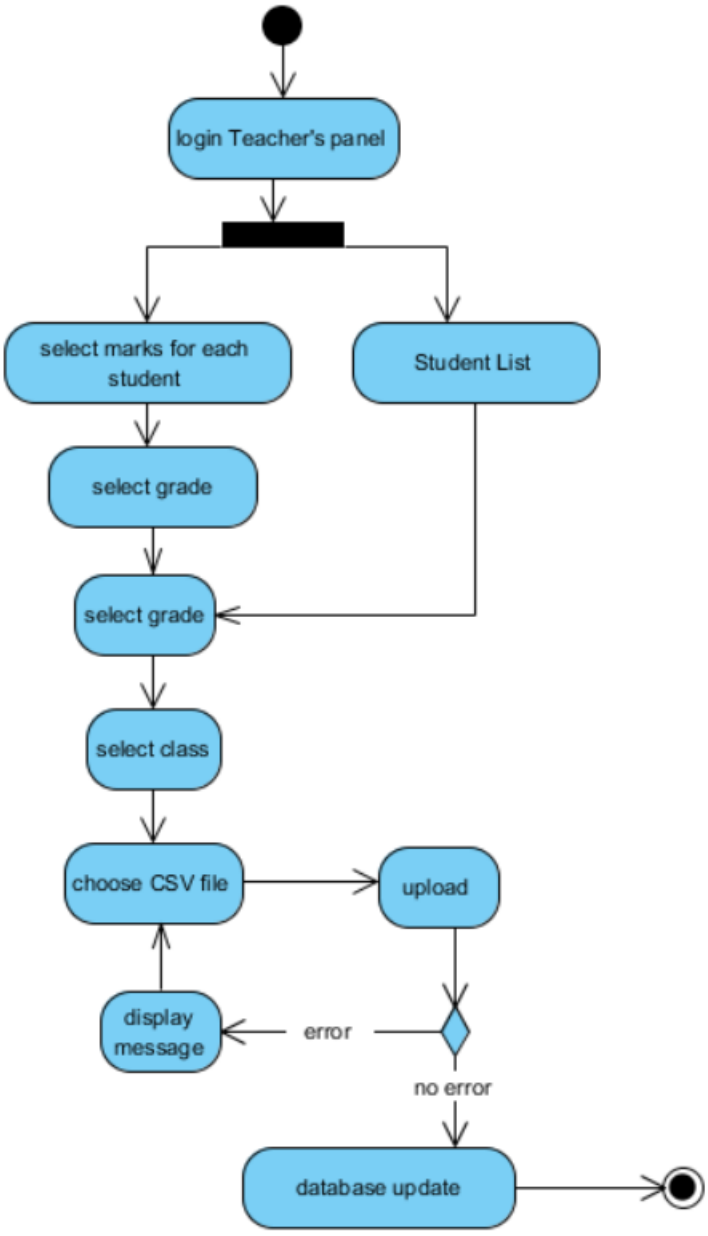


Figure 9: Activity Diagram for upload CSV file use case

- Activity Diagram for compare marks use case

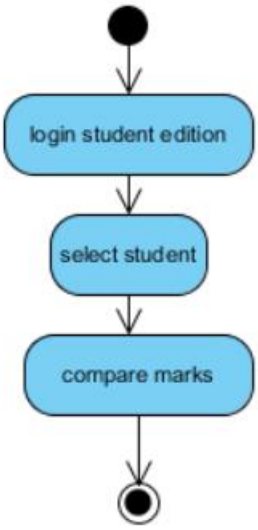


Figure 10: Activity Diagram for compare marks use case

3.7 CLASS DIAGRAM

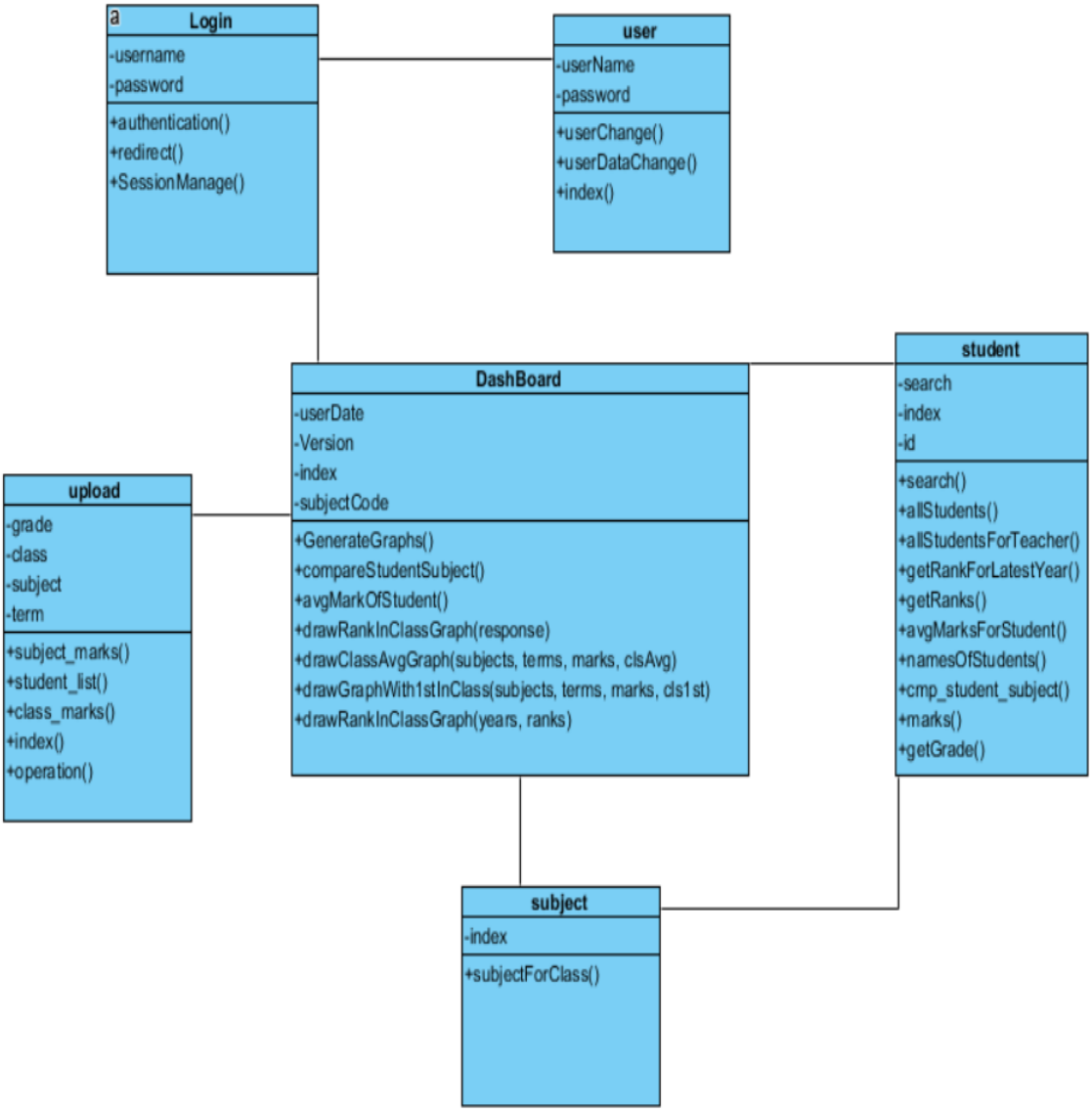


Figure 11: object Diagram of the SGM system

3.8 SEQUENCE DIAGRAM

3.8.1 Student and Teacher login sequence diagram

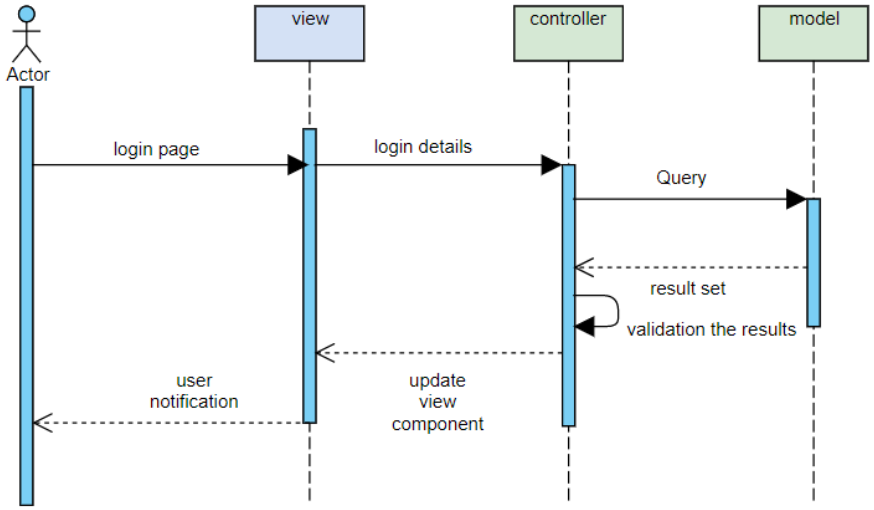


Figure 12: students and teachers login sequence diagram

3.8.2 Upload marks CSV file sequence diagram

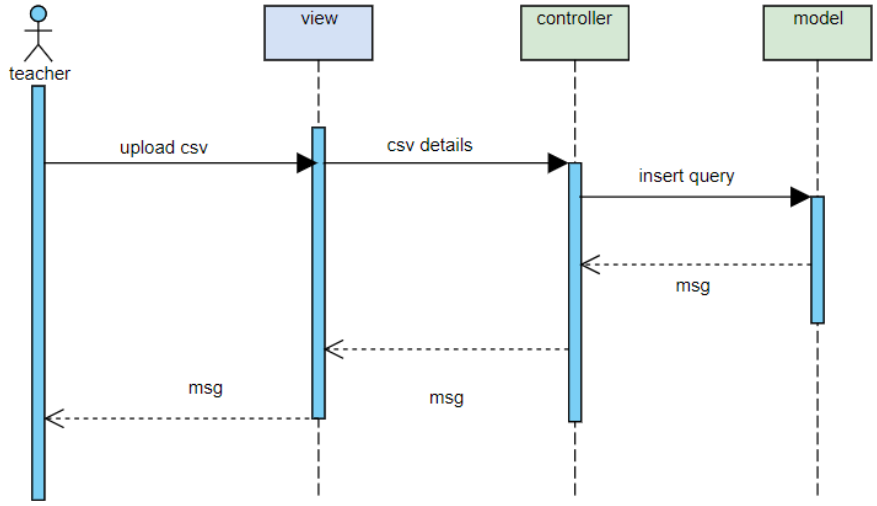


Figure 13: 3.8.2 Upload marks CSV file sequence diagram

3.8.3 View students result in the sequence diagram

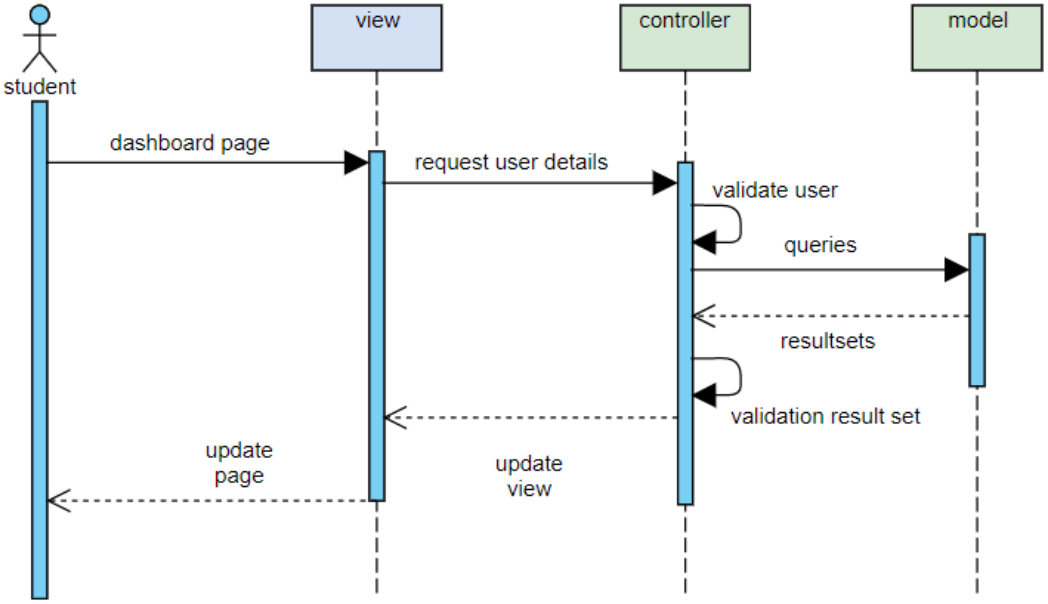


Figure 14: 3.8.3 View students result in the sequence diagram

3.9 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information effectively and efficiently. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual Database Management System (DBMS).

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

3.9.1 ER Diagram

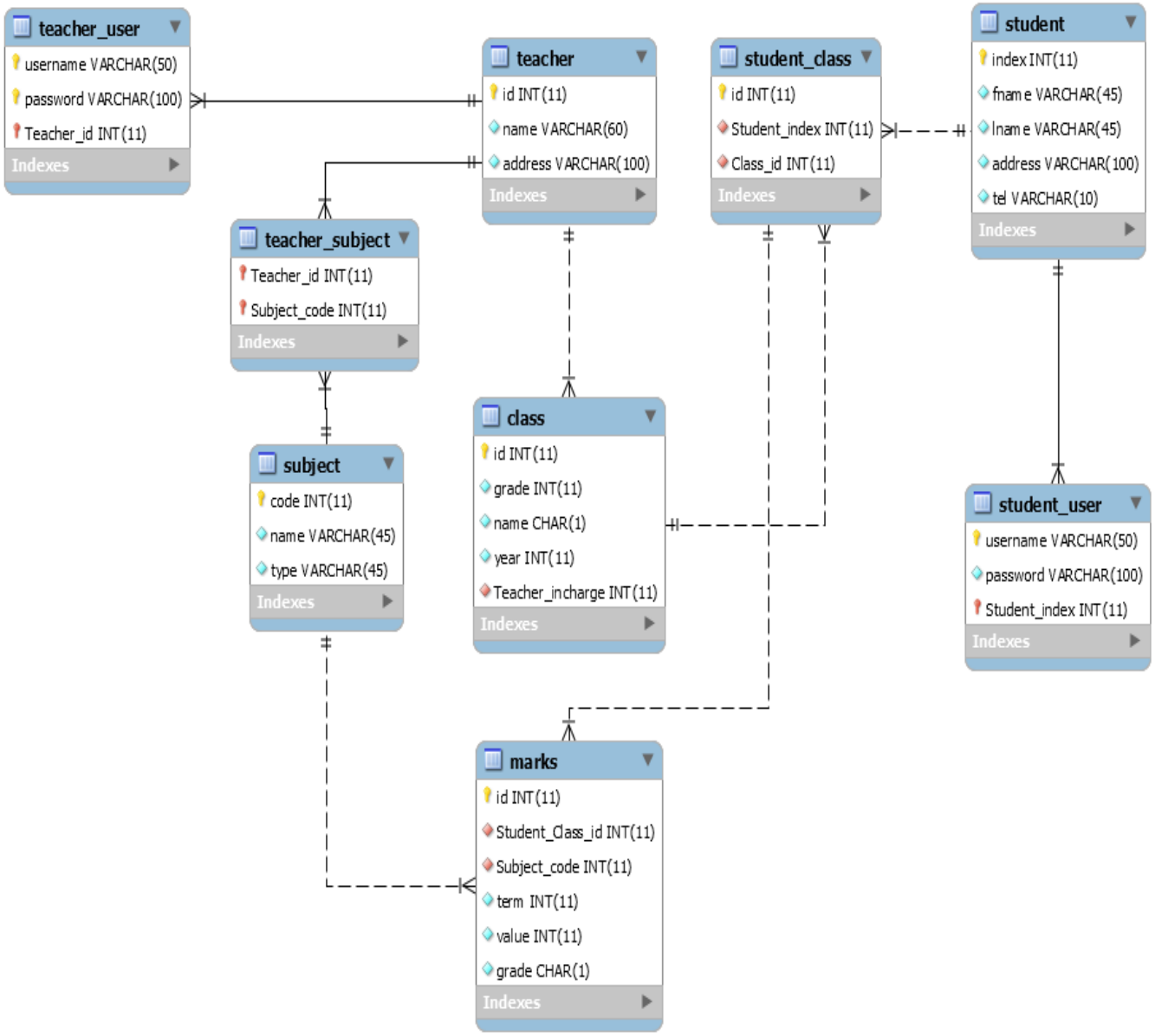


Figure 15: ER diagram of the SGM

3.9.2 Database Table Diagram

- Class Table

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	*None*
2	grade	int(11)			No	*None*
3	name	char(1)	utf8_general_ci		No	*None*
4	year	int(11)			No	*None*
5	Teacher_incharge	int(11)			No	*None*

- Marks Table

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	*None*
2	Student_Class_id	int(11)			No	*None*
3	Subject_code	int(11)			No	*None*
4	term	int(11)			No	*None*
5	value	int(11)			No	*None*
6	grade	char(1)	utf8_general_ci		No	*None*

- Students Table

#	Name	Type	Collation	Attributes	Null	Default
1	**index**	int(11)			No	*None*
2	**fname**	varchar(45)	utf8_general_ci		No	*None*
3	**lname**	varchar(45)	utf8_general_ci		No	*None*
4	**address**	varchar(100)	utf8_general_ci		No	*None*
5	**tel**	varchar(10)	utf8_general_ci		No	*None*

- Student class Tabel

#	Name	Type	Collation	Attributes	Null	Default
1	**id**	int(11)			No	*None*
2	**Student_index**	int(11)			No	*None*
3	**Class_id**	int(11)			No	*None*

- Student User Tabel

#	Name	Type	Collation
1	**username**	varchar(50)	utf8_general_ci
2	**password**	varchar(100)	utf8_general_ci
3	**Student_index**	int(11)	

- Subject Table

#	Name	Type	Collation	Attributes	Null	Default
1	code	int(11)			No	*None*
2	name	varchar(45)	utf8_general_ci		No	*None*
3	type	varchar(45)	utf8_general_ci		No	*None*

- Teacher Table

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	*None*
2	name	varchar(60)	utf8_general_ci		No	*None*
3	address	varchar(100)	utf8_general_ci		No	*None*

- Teacher Subject Table

#	Name	Type	Collation	Attributes	Null	Default
1	Teacher_id	int(11)			No	*None*
2	Subject_code	int(11)			No	*None*

- Teacher User Table

#	Name	Type	Collation	Attributes	Null	Default
1	username	varchar(50)	utf8_general_ci		No	*None*
2	password	varchar(100)	utf8_general_ci		No	*None*
3	Teacher_id	int(11)			No	*None*

3.10 DESIGN CONSIDERATIONS

This section describes many of the issues that are needed to be able to addressed or resolved before embarking on a complete design solution.

a) **Assumptions**

This SGM system design makes several assumptions about the software and hardware requirements as is in the SRS. All the environmental operating requirements of both the user interface and the database can be found in the SGM requirements. Both the database and the user application make the following assumptions about the operating environment. The system can be described by the operating requirements associated with this document and in the SRS. The system application in execution will have the necessary resources availed as required. This entails sufficient memory and permanent storage space and the adequate CPU for the application. The user application makes the following assumptions about its operating environment. The user machine will have Microsoft access database components installed, as they are required for the system implementation. The machine will also have the necessary database setup.

b) **Goals and guidelines**

The major goal of the Student Grade Monitoring System is to help automate the current manual process of schools and making it extremely simple and easy to use. The system is meant for the teachers and students some of whom are not technically advanced. The prompts for the user must be clear and concise since this will be the highest level of interaction between the system and the user. It is also important the series of prompts and responses be tested with the users before being deployed.24 The user should get a response in a timely fashion since users tend to lose interest if they have to wait too long for the system to respond. In this design, a minimum of data is transferred between the user and the database to retrieve the necessary information and return the requested data to the user.

3.11 SYSTEM ENVIRONMENT

System scalability and security are the requirements for the system architecture of the Student Grade monitoring system. The system will accommodate scalability allowing flexibility within the system to expand, modify or downsize easily to meet the evolving business and technology change.

3.11.1 Development Tools

This part of the SDS specifies the tools that will be used to develop the system. They include PHP and MySQL for the database. Other tools to be used will include forms which will act as screens for input and out, tables that will be for input and data entry.

3.11.2 Design Methodology

The waterfall model will be used as the best language for this kind of system. This is because the waterfall model is suitable for visualizing, specifying, constructing and documenting the features of 25 the system. The design will take the following approach: designing the database, creating relationships, designing the user interfaces and the system processes.

3.11.3 Database Design

Database design refers to a process of modelling the information to meet the user requirements. The process will be accomplished in three stages.

3.11.4 Conceptual Design

Conceptual design refers to a process of constructing an abstract model of data to be included in a database. In creating the conceptual design for the SGM system, the following activities will be involved. Identification of the entities: the various entities included in the SGM are;

- Login
- Upload CSV files
- Student Dashboard
- Teacher Dashboard

3.11.5 Logical Design

This part of the database design will entail the selection of database model which is a collection of concepts and rules for the description of the structures of the database. A relational database model will be used for this case as it defines a database as a collection of tables containing all data and their related properties.

3.11.6 Data Normalization

This is a process of removing redundant data from the tables to improve storage efficiency, data integrity and data scalability.

3.11.7 Physical data design

The system will include the tables show below containing the attributes in them

3.12 INTERFACES DESIGN

3.12.1 login

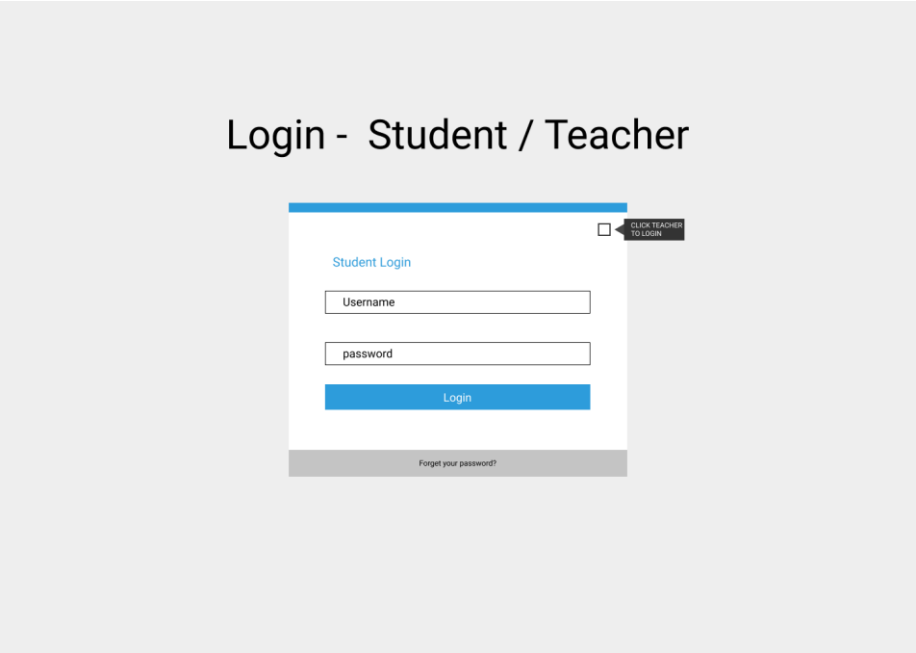


Figure 16: login wireframe

3.12.2 Student Dashboard

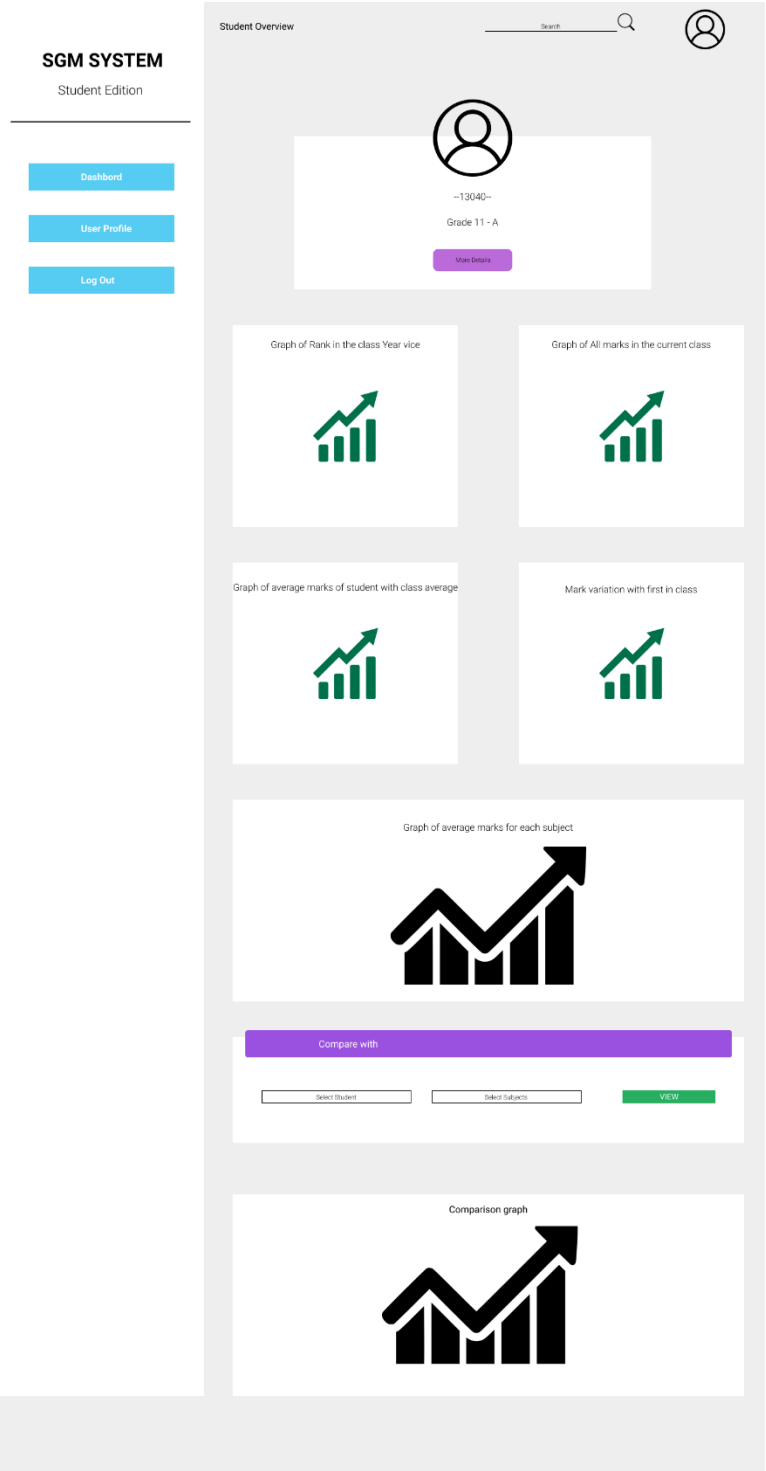


Figure 17: student dashboard wireframe

3.12.3 Teacher Dashboard

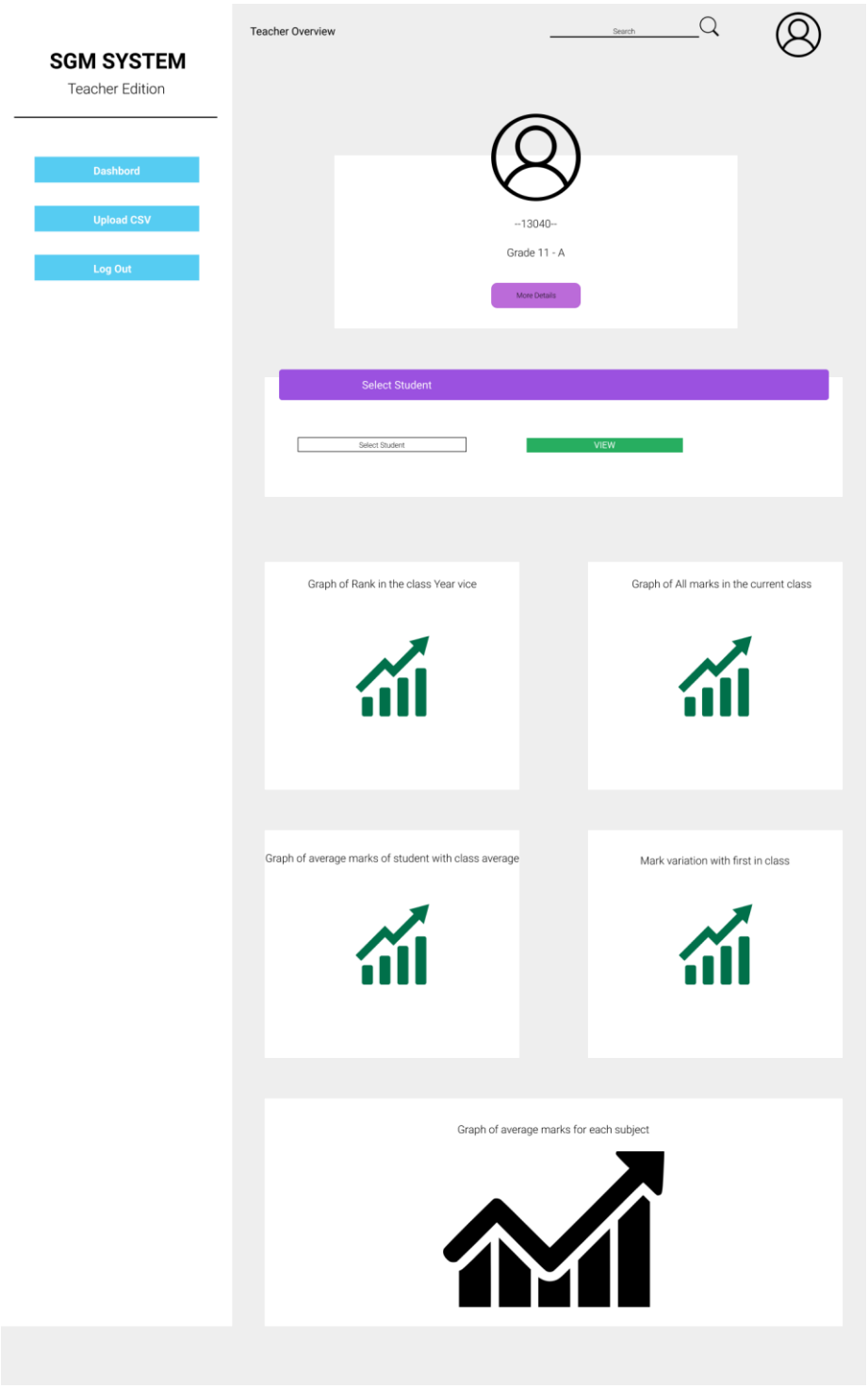
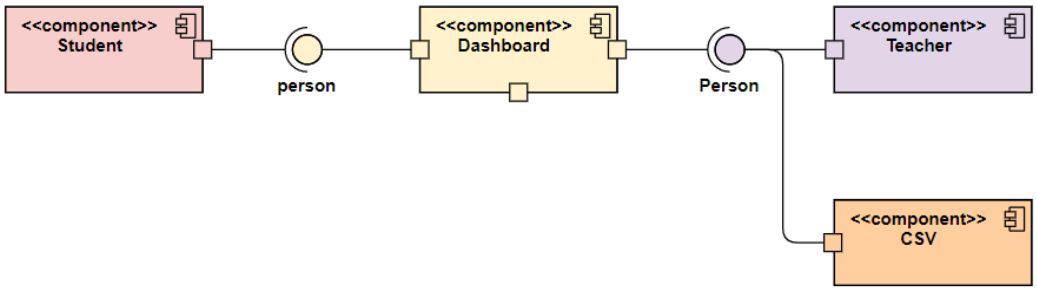


Figure 18: Teacher dashboard wireframe

3.13 COMPONENT VIEW OF PROPOSED SYSTEM

A component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with other essential elements on a need-to-know basis.



CHAPTER 04

4 SYSTEM IMPLEMENTATION

4.1 INTRODUCTION

The system has been developed using object-oriented concepts and PHP (MVC Framework called Codeigniter) backhand database using MySQL.

4.2 PURPOSE

The document contains overviews of system, description of the major tasks that are required to be done before the system is put into use.

4.3 SYSTEM DESCRIPTION

The SGM system will manage all of the student's grades relates to matters. Students can easily analysis their marks for each subject and figure out their evaluations. Teachers can easily insert all of the student's marks from CSV uploader. All the information that was uploaded, represented by the charts to students.

4.4 HARDWARE & SOFTWARE SPECIFICATIONS

4.4.1 Minimum Hardware Requirements

- Processor: Pentium 4 @ 2.40 GHz or Higher
- RAM: 1GB
- Hard disk: 80 GB
- Monitor: 14" CRT Or LED
- Keyboard: Normal or Multimedia
- Mouse: Compatible mouse
- USB Flash Drive
- Printer

4.4.2 Software Requirements

- Database: Y SQL database Server 5.5
- Web Technologies: PHP 5.6, CSS, HTML, Twitter Bootstrap 3, Codeigniter
- Server: Apache server
- Operating system: Windows XP, Windows 7, Windows 8
- Editor: PHPstorm
- Wireframe designing: Figma

4.5 TECHNICAL DESCRIPTION

I used CodeIgniter for implementation this. CodeIgniter is based on the Model-View-Controller (MVC) development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

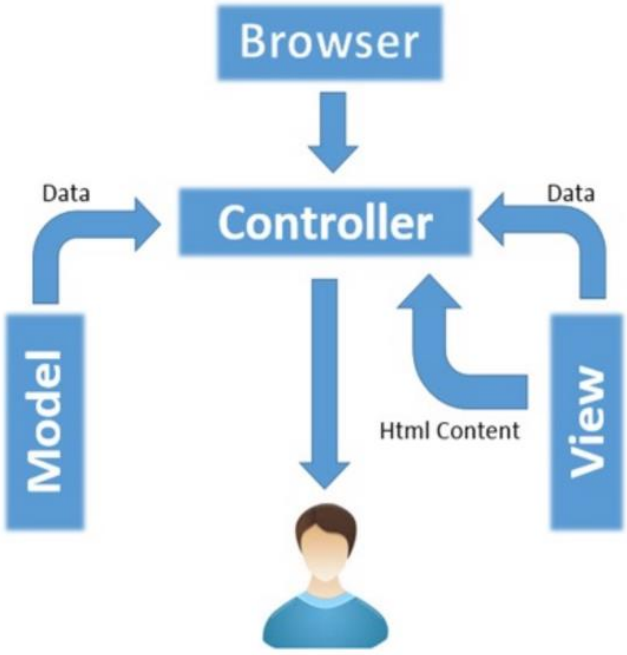


Figure 19: MVC architecture

Directory and PHP files description are given below

Models

- Class_model.php
- Marks_model.php
- Student_Class_model.php
- Student_model.php
- Subject_model.php
- User_Login_model.php
- User_Update_model.php

Views

- Dashboard.php
- Dashboard_Teacher.php
- Login.php
- Overall.php
- Password_change_Success.php
- Table_list.php
- Upload.php
- User.php

Controllers

- Dashboard.php
- Login.php
- Overall.php
- Student.php
- Subject.php

- TableList.php
- User.php
- Upload.php

4.6 IMPLEMENTATION VIEW

An architectural view called the implementation view typically captures the enumeration of all subsystems in the implementation model, the component diagrams illustrating how subsystems are organized in layers and hierarchies and illustrations of import dependencies between subsystems

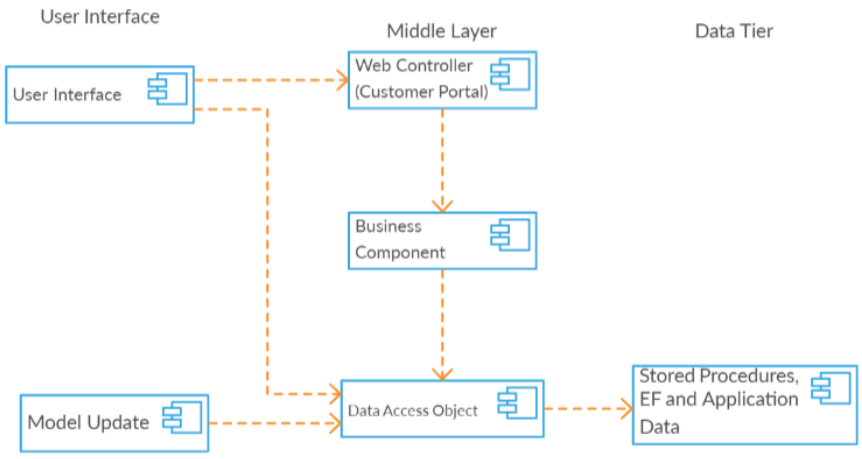
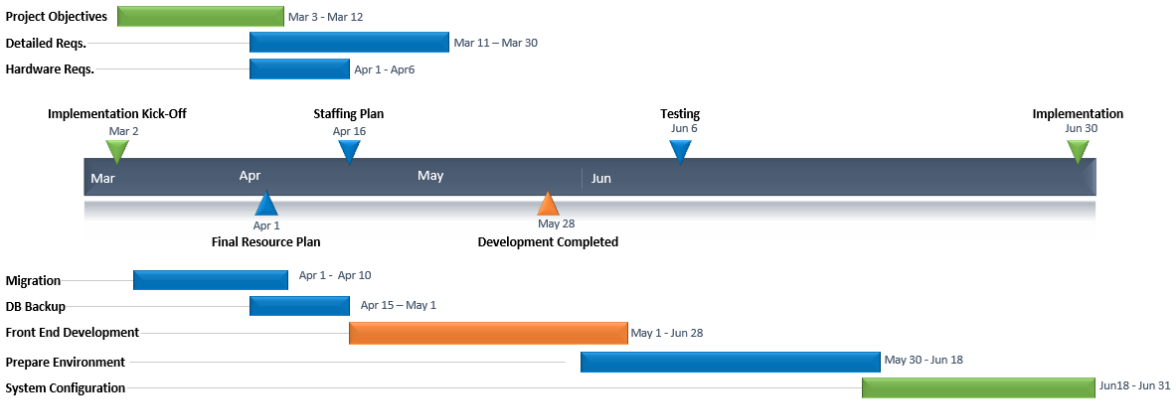


Figure 20: implementation view of the SGM

4.7 IMPLEMENTATION PLAN

This section refers to a detailed description of actions that demonstrate how to implement an activity within the project in the context of achieving project objectives, addressing requirements, and meeting expectations.



4.8 TESTING

4.8.1 Unit Testing

This is the testing process which I can do manually because in this testing program is a tested individually using dummy record to see whether that program produces satisfied output as the company and validation also.

4.8.2 Validation Testing

In these requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final Assurance

that software and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end-user)

4.8.3 System Testing

System testing is a series of the different test whose primary purpose is to fully exercise the computer base system. Where the software and other system elements are tested as a whole. To test computer software, I spiral out along streamlines that broadens the scope of testing with each turn.

The last higher-order testing step falls outside the boundary of software Engineering and into the broader context of computer system engineering. Software, once validated, must be combining with order system Elements(e.g.hardware, people, databases). System testing verifies that all the elements Mesh properly and that overall system function/performance is achieved.

1. Recovery Testing
2. Security Testing
3. Stress Testing

4.8.4 Test cases

Student case					
Case ID	Test Case	Expected Output	Actual Output	Pass/Fail	Remark
1	Login	invalid user message	Success	pass	Good
2	Login	Login with relevant details	Success	Pass	Good
3	Compare results	Generate the graph	Success	Pass	Good

4	Change the password	Change the current password	Success	Pass	Good
5	Change the user profile	Update the student information	Success	Pass	Good
6	Logout	Logout from the current session	Success	Pass	good
7	Notification	New notification	Not successful	Fail	poor

Teacher case					
Case ID	Test Case	Expected Output	Actual Output	Pass/Fail	Remark
1	Login	invalid user message	Success	pass	Good
2	Login	Login with relevant details	Success	Pass	Good
3	Select Student	Generate the statistical graph with selected student	Success	Pass	Good
4	Upload CSV	Invalid format message	Exception	Fail	Poor
5	Upload CSV	New marks added to the database	Success	Pass	Good
6	Upload CSV	New use created	Success	Pass	Good
7	Logout	Logout the current session	Success	Pass	Good

CHAPTER 5

5 CRITICAL APPRAISAL OF THE WORK

5.1 GANTT CHART

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflect the start date, duration and end date of the activity. This allows you to see at a glance

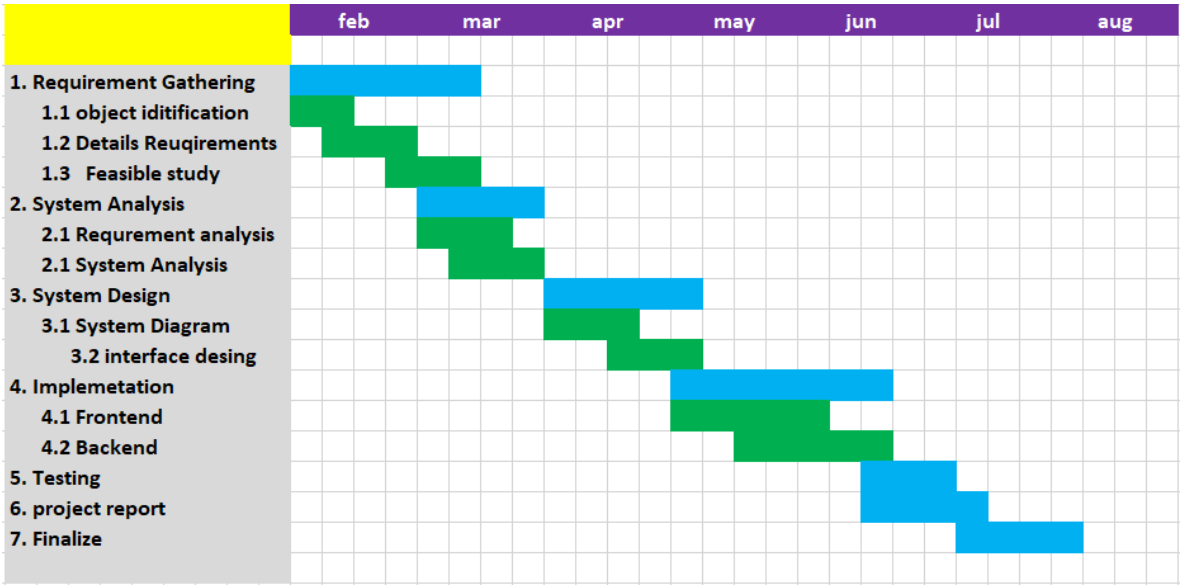


Figure 21: Gantt chart

5.2 PROBLEMS ENCOUNTERED

There are several problems that I faced during this project. The first one is, how to make things user friendly and reduce the complexity. For that, I used the more realistic graphical charts. So the students can easily evaluate their progress. I have to learn PHP framework called Codeigniter a to z. And how MVC architecture works. Also CSS framework like bootstrap, I have to learn.

5.3 WHAT WE HAVE DONE

Students and teachers can sign up to the system and after they can use the passwords to sign in to the system.

If a user login as a student, he can see:

1. His/her marks on latest term test
2. Comparison his/her marks with the best student's marks of the class
3. Comparison of his/her previous grades with current grades
4. Comparison his/her position in the class before and now
5. Average marks for each subject in each term in the dashboard.

If the student wants to compare his marks with another student, then he has to select the relevant student to see.

If the user login as a teacher, he/she can see the overall students' marks. The teacher can select a student and see his/her performance throughout the year.

5.4 FUTURE DEVELOPMENT

- The security methods of the system have to be improved further.
- Have to give access to the system
- Add more graphs to the system
- Add admin panel

5.5 CONCLUSION

Students' Grade Monitoring system is used to view the analysis of the marks of a particular student or overall in class. By seeing the graphs, the teacher or student can be able to track the turning point of the student because there can be some patterns in students' performance. After identifying the turning point, the teacher can get actions to improve the performance of the student.

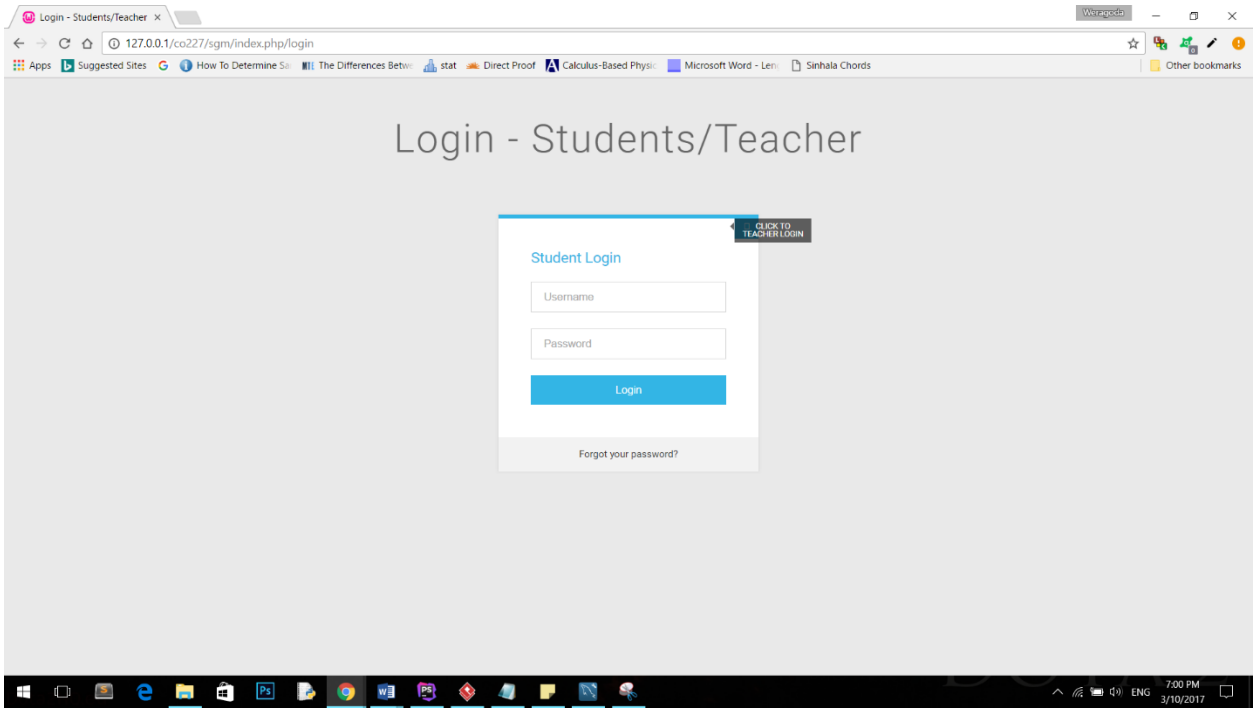
These graphs represent the variation of marks for each subject. So the teacher can identify the students who are good with art subjects or a particular area. Then adults can help a student to create his/her future career path.

CHAPTER 06

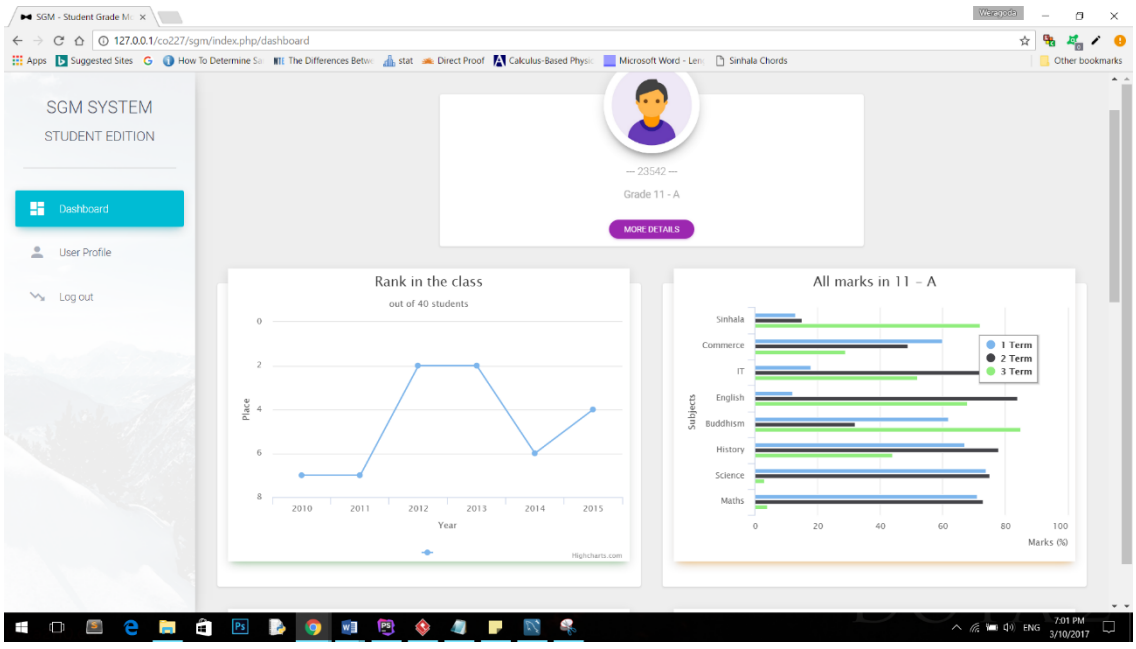
6 APPENDICES

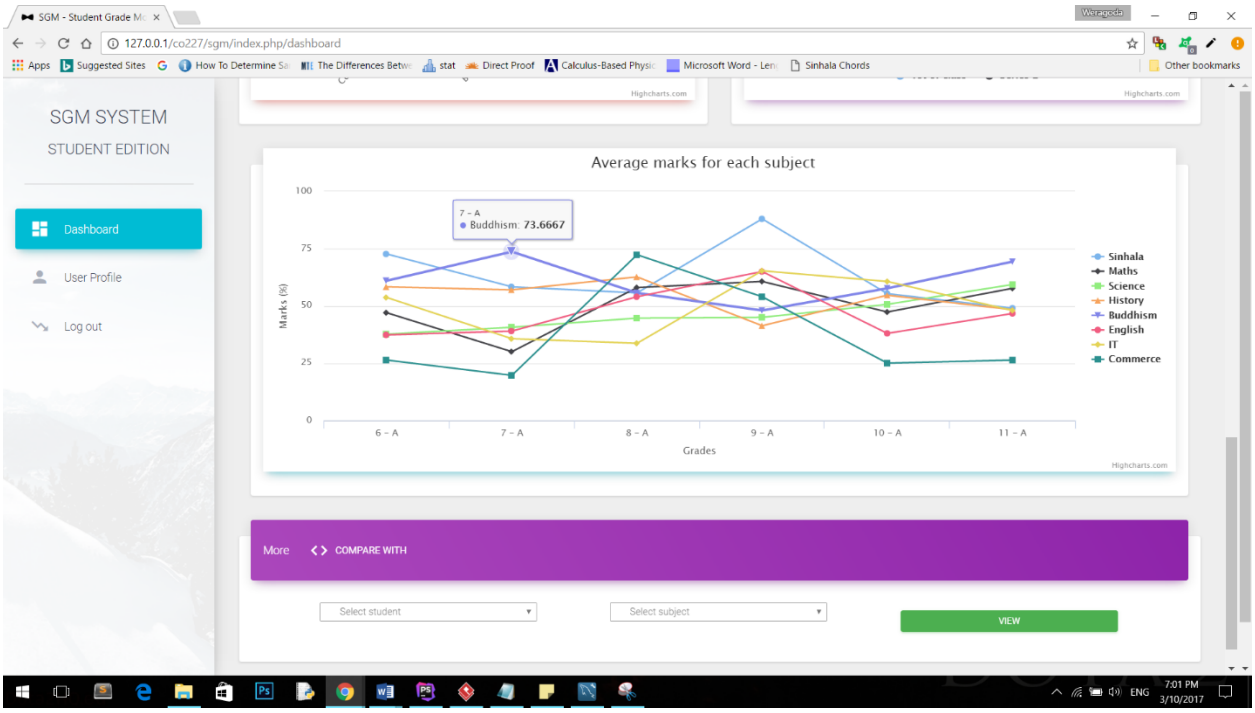
6.1 INTERFACES DESIGN

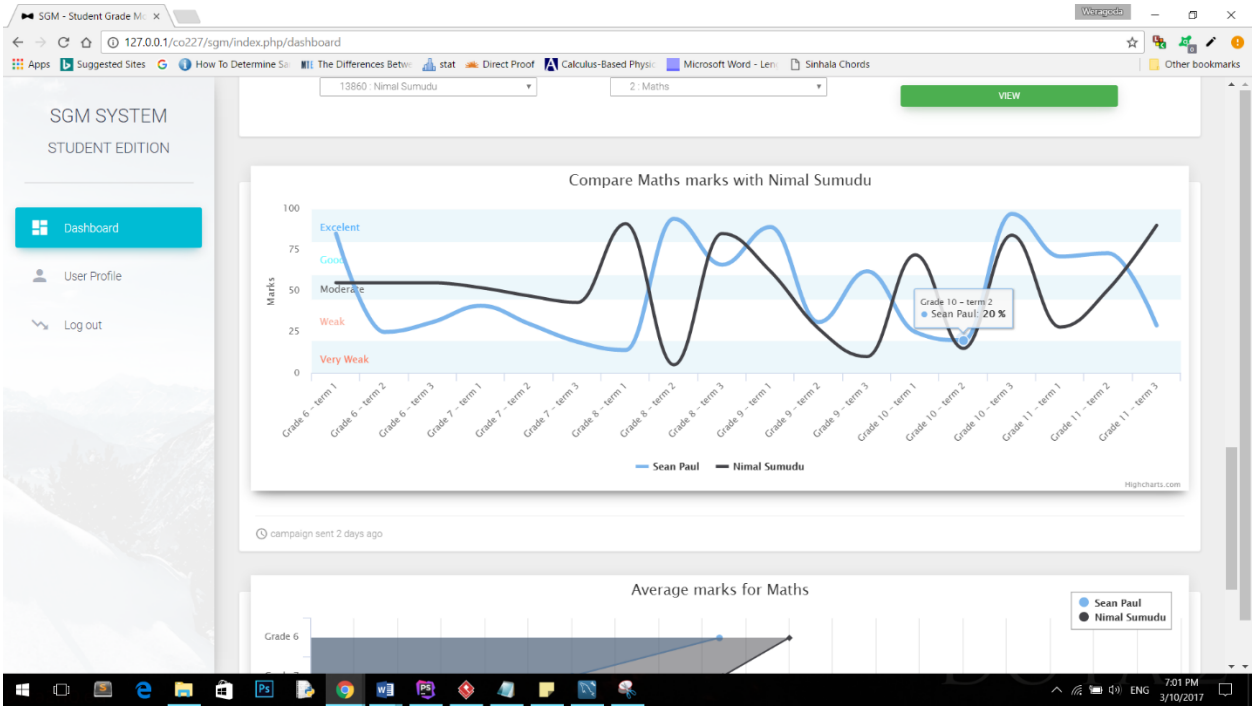
6.1.1 login



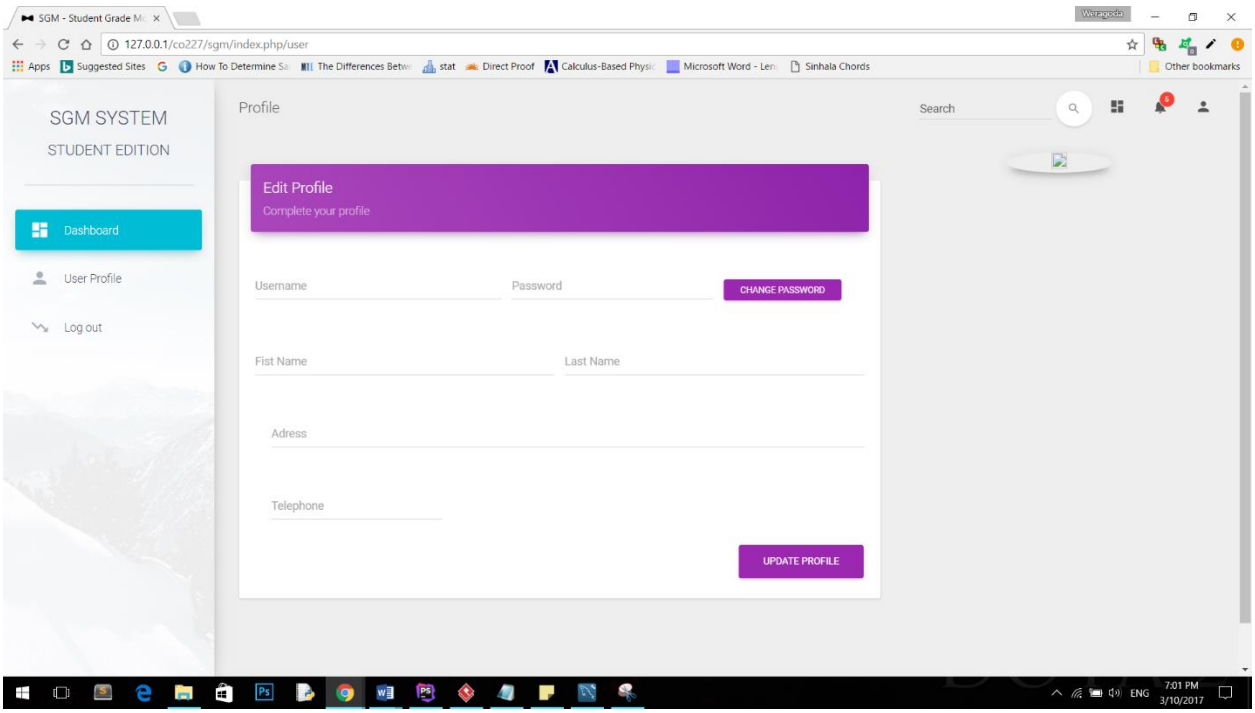
6.1.2 Student Dashboard



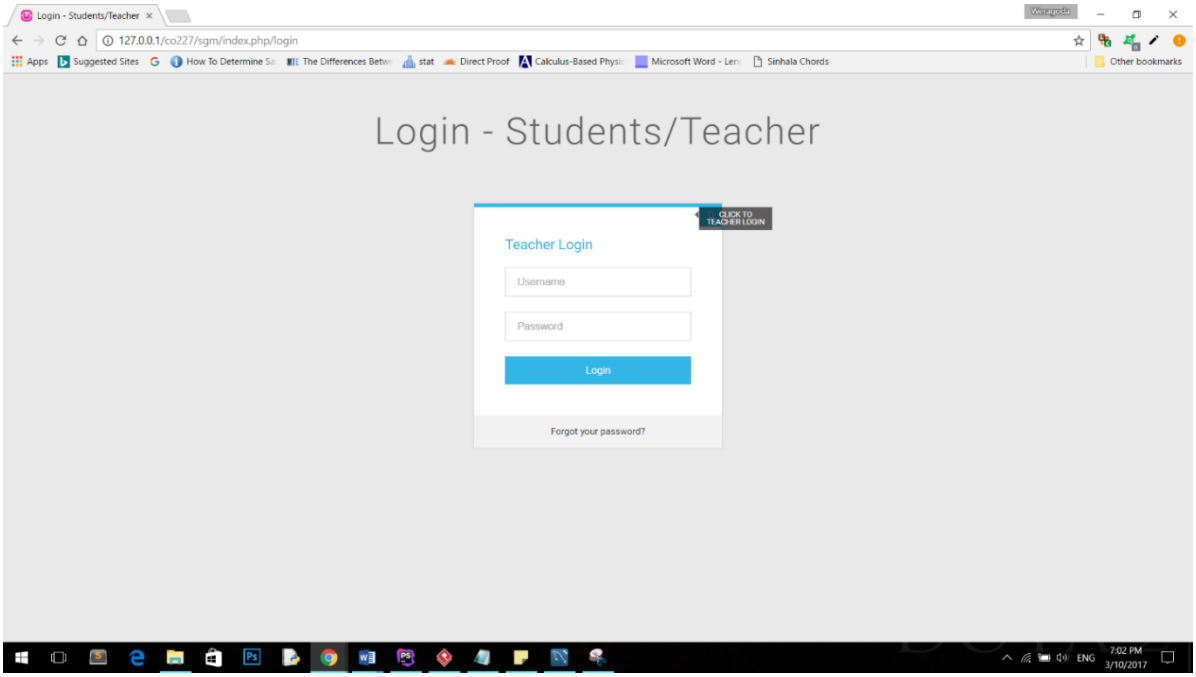

SGM - Student Grade M
127.0.0.1/co227/sgm/index.php/dashboard
SGM SYSTEM
STUDENT EDITION
Dashboard
User Profile
Log out
Average marks for each subject
7 - A
Buddhism: 73.6667
Marks (%)
Grades
6 - A
7 - A
8 - A
9 - A
10 - A
11 - A
Sinhala
Maths
Science
History
Buddhism
English
IT
Commerce
Highcharts.com
More
COMPARE WITH
Select student
Select subject
VIEW
ENG
7:01 PM
3/10/2017


SGM - Student Grade M
127.0.0.1/co227/sgm/index.php/dashboard
13860 : Nimal Sumudu
2 : Maths
VIEW
SGM SYSTEM
STUDENT EDITION
Dashboard
User Profile
Log out
Compare Maths marks with Nimal Sumudu
Marks
Excelent
Good
Moderate
Weak
Very Weak
Grade 10 - term 2
Sean Paul: 20 %
Sean Paul
Nimal Sumudu
Highcharts.com
campaign sent 2 days ago
Average marks for Maths
Grade 6
Sean Paul
Nimal Sumudu
ENG
7:01 PM
3/10/2017

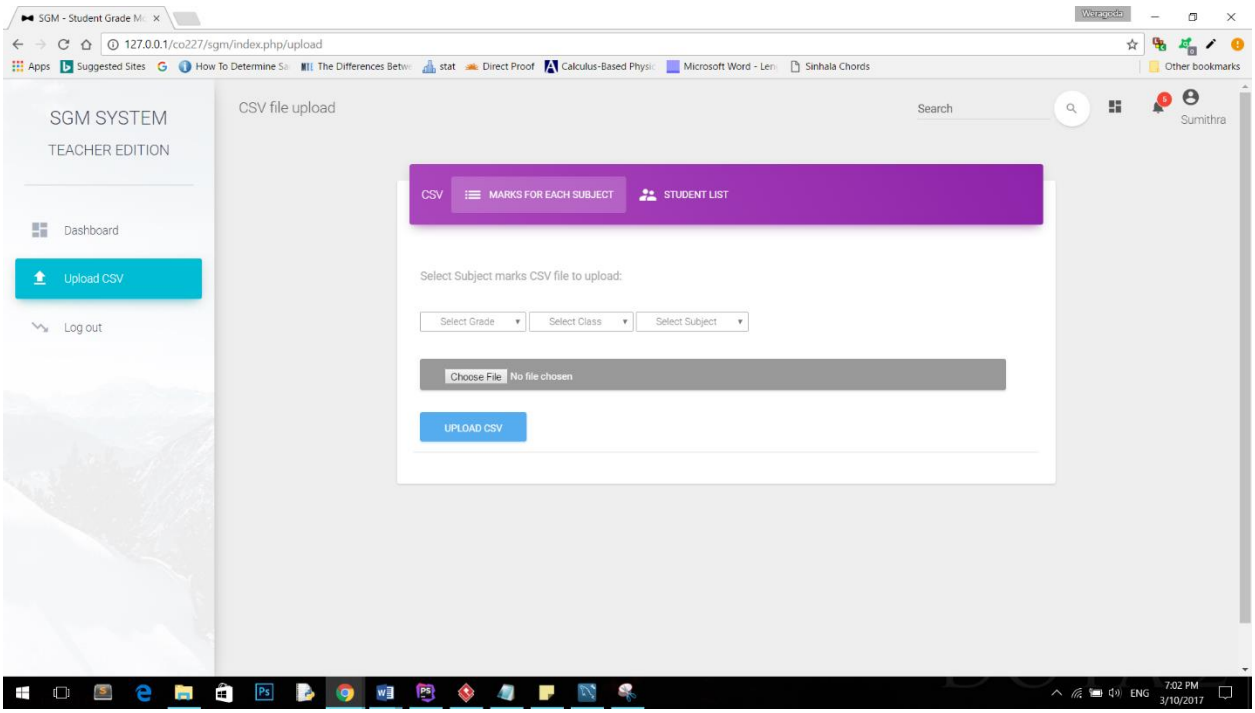
6.1.3 Edit profile Interface

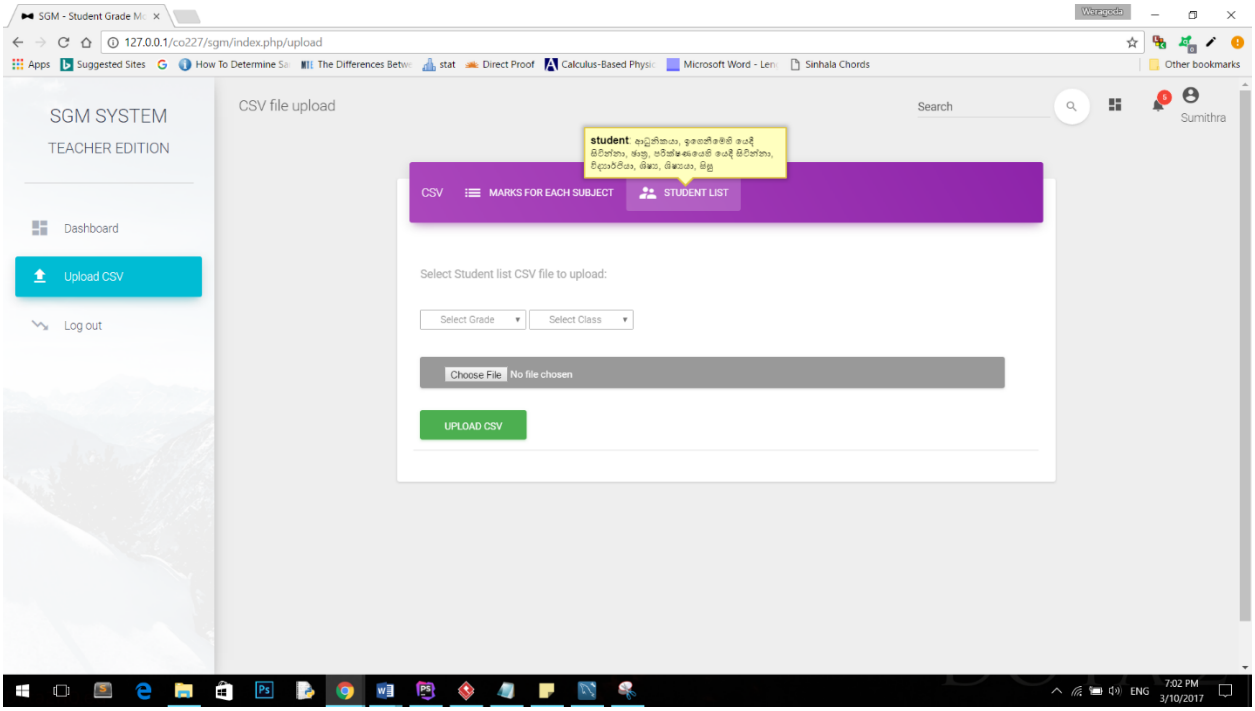


6.1.4 Teacher's Login



6.1.5 CSV upload Interface





CHAPTER 07

7 REFERENCES

Data, R., n.d. [Online]

Available at: https://www.w3schools.com/

opensource, n.d. [Online]

Available at: https://www.highcharts.com/

PowerSchool, n.d. [Online]

Available at: http://www.powerschool.com/

QuickSchools, I., n.d. [Online]

Available at: https://www.quickschools.com/

Technology, B. C. I. o., n.d. [Online]

Available at: https://codeigniter.com/userguide3/tutorial/index.html

tutorialspoint, n.d. [Online]

Available at: https://www.tutorialspoint.com/index.htm

tutorialspoint, n.d. *basic_mvc_architecture.htm.* [Online]

Available at: https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm