TRACK MY CLASS

Mini Project Report by

V.SOWMIYA
TU6220206111006

B.SWATHI
TU6220206111007

School of Computer Science
Submitted to the

SCHOOL OF COMPUTER SCIENCE

in partial fulfillment of the requirements for the award of the degree

BACHELOR OF COMPUTER SCIENCE

Submitted to the
FACULTY OF SCIENCES

SCHOOL OF COMPUTER SCIENCE
TAKSHASHILA UNIVERSITY
JULY – 2024

TAKSHASHILA UNIVERSITY

SCHOOL OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project entitled “TRACK MY CLASS” is a bonafide work of V.SOWMIYA, Register Number : TU6220206111006, B.SWATHI, Register Number : TU6220206111007, who carried out the project under my supervision in the partial fulfillment of the requirement for the degree of Bachelor of Computer Science during the year 2023-2024. I further certify that the project work done by him has not previously formed the basis for the award to the candidate or to any one else of any degree or any similar title.

Project Guide **School Incharge**

Mr.R.BHARADHIDASAN, M.C.A., M.Phil., (Ph.D),
Assistant Professor,
School of Computer Sciences

Submitted to Viva-Voce examination held on _______________

Internal Examiner **External Examiner**

Place: Ongur
Date:

DECLARATION

I, V.SOWMIYA , B.SWATHI, hereby declare that my project entitled “TRACK MY CLASS ” is done during 2023-2024 for the award of Bachelor of Computer Science under School of Computer Sciences and Technology of Takshashila University as a part of my curriculum.

I further declare that this project or a part of this project has not been submitted elsewhere for the award of any other degree.

Place: Ongur	V. SOWMIYA
Date:	(Reg No TU6220206111006)
	B. SWATHI
	(Reg No TU6220206111007)

CONTENT

TRACK MY CLASS

“Streamlining your Class Queries”

V.Sowmiya [TU6220206111006]

B.Swathi [TU6220206111007]

ABSTRACT

"**Track My Class**" is a Student Database Management System (SDMS) developed using **Flask**, a lightweight **Python** web framework, and **MySQL**, a robust relational database management system. This system is designed to manage and streamline various aspects of student and academic information, including student profiles, class schedules, attendance tracking, and grade management. The system allows Faculties to register, view and update their personal details, enroll in courses, and track their academic performance in real-time. Faculty members can efficiently manage course schedules, mark student attendance, input grades, and generate performance reports. The use of Flask ensures a lightweight and scalable web application, while MySQL provides a reliable backend for securely storing and retrieving student data. With its user-friendly interface, "**Track My Class**" offers easy access to important academic information, reduces administrative workload, and improves data accuracy and communication between students, faculty, and administrators. The system's realtime updates and secure data handling make it a modern, efficient solution for managing student records and enhancing the overall educational experience.

The use of Flask allows for the development of a lightweight web application that is both fast and easy to maintain, ensuring quick response times and an intuitive user interface. Meanwhile, MySQL serves as the backend database, offering reliable data storage, complex querying capabilities, and secure data management. With features like real-time updates, automated attendance tracking, and grade management, the reduces administrative workload, minimizes the chances of errors, and enhances communication.

"Track My Class" is also designed with a strong emphasis on data security, ensuring that sensitive student information is safely stored and accessible only to authorized users. Additionally, the system includes various reporting tools for administrators, enabling them to generate detailed reports on attendance, grades, and student performance. This project not only automates many tedious administrative tasks but also provides a user-friendly and efficient way for educational institutions to manage their student data, fostering a more organized, transparent, and productive academic environment. Through its use of modern technologies like Flask and MySQL, "Track My Class" serves as an innovative solution to the evolving needs of educational institutions in the digital age.

REQUIREMENTS

Software Requirements:

Operating System	Windows 11 Pro
Front End	HTML, CSS, JavaScript ,Bootstrap
Back End	Python
Program	Python
IDE	PyCharm
Framework	Flask
Database	SQLite
Server	Local Server
System type	64-bit operating system

Hardware Requirements:

System	PC/Laptop
Processor	Intel Core i3 / Intel Core i7
RAM	8.00 GB / 16.00 GB
Hard disk	SSD
Keyboard	Standard USB keyboard
Mouse	USB mouse, Track pad
Speaker	Hight -quality speaker
Graphics card	NVIDIA GeForce GTX 1660

INTRODUCTION

About the Project

Project Title: “Track My Class”, A Web-Based Student Database Management System Using Flask and MySQL.

"Track My Class" is a comprehensive Student Database Management System (SDMS) designed to manage and streamline key aspects of academic administration for educational institutions. Developed using Flask, a lightweight Python web framework, and MySQL, a powerful relational database, this system serves as an efficient tool for managing student information, class schedules, grades, and attendance. The project aims to automate and simplify administrative tasks, reducing the burden on staff while improving the accuracy and accessibility of student data.

The system enables students to create and manage personal profiles, view their academic records, enroll in courses, track attendance, and access real-time updates on grades and assignments. Faculty members can efficiently manage class schedules, mark attendance, input grades, and generate performance reports, all from a single platform. Administrators can access comprehensive data, including student records and performance analytics, through advanced reporting features.

Key functionalities include:

Student Profile Management: Students can register, update personal information, and manage academic records**.**

Course Enrollment: Students can browse and enroll in courses, while instructors can manage course schedules and student rosters.

Attendance Tracking: Faculty can mark attendance and monitor student participation.

Grade Management: Teachers can input and update grades for exams, assignments, and coursework.

Reporting and Analytics: Administrators and faculty can generate reports on student performance, attendance, and course enrollment.

Data Security and Access Control: Secure user authentication ensures that only authorized individuals can access sensitive student data.

The system's design leverages the simplicity and flexibility of Flask for the frontend and backend integration, while MySQL handles efficient and secure data storage. With real-time updates and a user-friendly interface, "Track My Class" offers a scalable, secure, and responsive solution that enhances communication and transparency between students, faculty, and administrators. By automating routine administrative tasks, this project not only simplifies data management but also contributes to better educational outcomes through more effective tracking of academic progress.

About the Takshashila University

Takshashila University was established with the vision to recreate the glory of the ancient Takshashila, India's oldest university of higher learning. Located in Ongur, Tamil Nadu, the university is part of the prestigious Sri Manakula Vinayagar and Mailam Group of Institutions, with over 25 years of experience providing quality education.Takshashila University aims to promote academic excellence through rigorous programs taught by eminent faculty and innovative teaching methods like hands-on learning and research.

Institutional Mission: The mission of "Track My Class" is to revolutionize academic administration by providing an efficient, secure, and user-friendly platform that empowers educational institutions to manage student information, attendance, grades, and course schedules with ease and accuracy. Our system aims to reduce administrative overhead, increase transparency, and enhance communication between students, faculty, and administrators, thereby fostering an environment of academic excellence. The platform's robust features ensure that educational institutions can manage student data efficiently while upholding the highest standards of data security and privacy.

Vision: By leveraging cutting-edge technologies such as Flask and MySQL, our vision is to create a dynamic system that evolves with the needs of the education sector, offering scalable, adaptable, and innovative features that simplify administrative tasks and improve the student experience. "Track My Class" aspires to set a new standard for how institutions handle academic records, attendance tracking, grades, and course management, promoting greater transparency, accuracy, and accessibility.

Existing System

The existing systems for managing student data in educational institutions are typically either manual or involve the use of multiple disconnected software applications. These traditional methods often lead to inefficiencies, inaccuracies, and difficulties in communication between students, faculty, and administrators.

Here's a breakdown of the current practices and challenges faced by institutions:

1. Manual Record Keeping:

In many educational institutions, student data is still managed using paper records or spreadsheets. This method is prone to human errors, data loss, and inefficiencies, especially when it comes to retrieving or updating student information. Handling large volumes of data manually can be time-consuming and error-prone, which increases the administrative burden.

2. Disjointed Software Solutions:

Some institutions use a variety of software tools for different purposes, such as Excel for grades, separate systems for attendance tracking, and other tools for course scheduling. These systems rarely integrate, leading to issues like inconsistent data, data duplication, or confusion about which software holds the most up-to-date information. Faculty and administrators often need to work across multiple platforms to manage student records, which creates inefficiency and increases the chances of errors.

3. Limited Access and Security:

Many existing systems lack proper security measures, leaving sensitive student data vulnerable to unauthorized access or misuse. Furthermore, access to student data is often limited to specific departments or individuals, creating communication barriers between students, teachers, and administrators. The lack of real-time data access also limits the ability to monitor student performance or attendance consistently.

4. Lack of Real-Time Updates:

Most current systems do not offer real-time updates on academic progress or attendance. Students and faculty members may have to wait for periodic updates or reports, resulting in delays in academic decision-making and feedback. This lack of immediacy can lead to issues like students falling behind without timely interventions.

5. Difficulty in Reporting and Analytics:

Generating reports on student performance, attendance, and course enrollment is a complex and time-consuming task in many institutions. Administrators often have to manually compile data from various sources and tools, which can be errorprone and inefficient. Existing systems may not offer advanced reporting tools, making it difficult for faculty and administrators to generate customized or comprehensive performance reports.

Proposed System

Introduction:

The proposed system, "Track My Class", is a web-based student database management system designed to streamline the tracking of student information, attendance, grades, and course enrollment. Developed using Flask as the backend framework and MySQL as the database, the system will allow students, teachers, and administrators to interact with the data efficiently. Key features include student registration, profile management, course creation, attendance tracking, grade management, and student performance dashboards. The system will support rolebased access control, with admins having full control over user and course management, teachers managing their own courses, and students accessing their personal data and progress. The backend will utilize Flask-Login for secure user authentication and SQLAlchemy for database interaction. The system's frontend will be built with HTML, CSS, and JavaScript, ensuring a responsive and intuitive interface. The system will provide a centralized platform for managing academic records, improving organization and accessibility for educational institutions.

Key Features:

- **Student Registration**: New students can register with personal details such as name, roll number, email, phone number, and class details.

- **Student Profile Management**: Students can view and update their personal details and track their academic progress.

- **Course Management**: Teachers and administrators can create, update, and delete courses. Each course will have associated details like course name, instructor, and syllabus.

- **Attendance Tracking**: Administrators and teachers can track student attendance for each course, with options to mark attendance for individual classes.

- **Grade Management**: Teachers can assign grades and marks for each student in their respective courses.

- **Student Dashboard**: Students can view their overall progress, including grades, attendance, and upcoming assignments.

- **Search & Reporting**: Admins and teachers can search for students based on various criteria, like name, roll number, or course. Reports on student performance can be generated.

- **Login and Authentication**: The system will support different user roles (Admin, Teacher, Student), each with specific permissions. Admins will have full access to the system, while teachers can manage only their courses and students, and students can manage their profiles.

System Workflow

1. **User Authentication**:

    - Users (Admin, Teacher, Student) will log in using their credentials. Flask-Login will manage user sessions and provide secure authentication.

2. **Student Registration**:

    - Admins will add new students to the system via a registration form.

      The student's information will be stored in the `students` table.

3. **Attendance Management**:

    - Teachers will mark attendance for each class session. The attendance data will be recorded in the `attendance` table and can be tracked by students and teachers.

User Roles and Permissions

- **Admin**: ○ Full access to all features. ○ Can manage students, courses, and users (teachers).

  ○ Can generate reports.
- **Teacher**: ○ Can manage their own courses (create/update/delete courses). ○ Can input grades and mark attendance for their students.
- **Student**: ○ Can view their profile, grades, and attendance.

  ○ Can update personal information.

  ○ Can view available courses and their own performance in each course.

Security Considerations

- **Password Encryption**: Passwords will be securely stored using hashing (e.g., using Flask-Bcrypt).
- **Role-based Access Control**: The system will implement role-based access control to ensure only authorized users can access certain functionalities.
- **SQL Injection Prevention**: All database queries will be parameterized to avoid SQL injection attacks.
- **Session Management**: User sessions will be securely managed to prevent unauthorized access.

Conclusion

The "Track My Class" system is a comprehensive solution for managing student information, attendance, and grades in a classroom or educational setting. Using Flask for the backend and MySQL for the database ensures that the system is scalable, secure, and efficient. This system provides a user-friendly interface for students, teachers, and administrators to interact with and manage academic data, helping to streamline educational administration and improve overall student tracking and performance monitoring.

MODULE

1.Navigation Bar (Header)

- Logo
- Home
- Student Management
- Course Management
- Enrollment
- Attendance
- Grade Management
- Fee Management
- Reports

2.Sidebar

- Quick Access Links Search
- Box.
- Recent Activities

3.Dashboard (Main Content Area)

- Overview/Statistics
- Graphical Charts
- Recent Activity
- Quick Actions

4. Student Management Section

□ Add New Student Form
□ Student List/View
□ Search & Filter
□ Edit/Delete Options

5.Attendance Section

□ Mark Attendance
□ View Attendance
□ Track Absences

6. User Profile & Settings

□ Update Profile
□ Change Password
□ Manage Roles & Permissions
□ System Preferences

7. Footer Section

□ Contact Information
□ Privacy Policy & Terms of Use
□ Help & Support
□ Copyright Information

8.Login/Authentication Page (if applicable)

□ Login Form
□ Forgot Password

SYSTEM DESIGN

Introduction System design is a process through which requirements are translated into a representation of software. Initially the representation depicts a holistic view of software. Subsequent refinement leads to a design representation that is very close to source code. Design is a place where quality fostered in software development. Design provides us with representation of software that can be assessed for quality; this is the only way that can accurately translate the customer requirements into finished software product or system. System design serves as the foundation for all software engineering and software maintenance steps that follow.

We look the design process from three distinct perspectives:

- Conceptual Design
- Logical Design
- Physical Design

The higher view is the conceptual view, followed by the logical view and finally the physical view. In designing an application, we generally begin and end each phase in a sequentially order, although they may overlap one another along the way.

Conceptual Design:

Conceptual Design is the process of acquiring and evaluating, documenting and then validating what the user envisions to be the business relation. It identifies the user and business requirements of the application and leads to a business solution as seen by the user.

At any point in the design process, the current state of the design should be directly traceable to a business problem and requirements. To achieve this conceptual design is driven by developing usage scenarios. These scenarios are a direct representation of the user's view of the solution to a specific business problem. A conceptual view places the emphasize on solving a business problem and deriving a solution that corresponds to the needs and requirements of the users. It is based on deriving the behavior of the solution with a primary emphasizes on the user. Beginning with a emphasis on the activities of the business rather than aspects of software development, underscores the fact that systems exists to serve the business. A strong focus on the user in the beginning of the project will help in maintaining a proper perspective throughput the development lifecycle . The conceptual design results in the first description of what the system does to solve the business problem articulated in the vision/scope document.

Logical Design

Logical Design derives business objects and their related services directly from these usage scenarios. The logical view of the solution provides a basis for evaluating different physical options. It also formalizes the solution for the project team.

The idea of the application is that the system first emerges in logical design.Its boundaries and business objects and it contains the system definition. Logical design specifies the interfaces between the system and external entities,such as users and other systems. Within a system there may be a number of sub-systems, and these boundaries are also specified.

Logical System Design consists of the following steps:

- Input/Output Specifications
- File Specifications
- Processing Specifications

Logical design should be technologically independent as possible, inorder to separate system behavior issues from system implementation issues. Implementation constraints should only be considered only after the project team verifies that the essential behaviour has been incorporated onto a logical design. This approach does not establish a technical direction until the system is well understood and documented.

Physical Design

The purpose of Physical Design is to translate the logical design into a solution that can be implemented effectively, according to performance, administration and development process requirements. This physical view should correctly implement the desired system behavior while meeting the constraints imposed by the technology.

In Physical Design, the perspective shifts from an abstraction of system behavior to an implementation of the behavior. Whereas the logical design is largely technology independent, physical design is necessarily tied to chosen set of technologies, these being the hardware and software on which the application will run.

The aim of physical design is to specify how to build portioned applications from software components. The interaction of these components through defined interfaces results in the desired behavior of the system as a whole. The rules for communicating between components are defined by interaction standards: what a component does and how it communicates are major considerations in physical design.

Physical design consists of the following steps:

1. Design the physical media
    - Specify input/output media.
    - Design the database and specify backup procedures.
    - Design physical information flow through the system.

2. Plan the system implementation
    - Prepare a conversion schedule target date.
    - Determine training procedure, courses and timetable

3. Device a test and implementation plan.
4. Specify any new Hardware/Software usage.
5. Update benefits, costs, conversion date and system constraints.

UML Diagrams

Introduction

Design is the first step in the development phase for an engineered product or system. Design is the place where quality is fostered in software development. Design is the only way that we can accurately translate a user's requirements into a finished software product or system. Software design serves as the foundation for all software engineers and software maintenance steps that follow. Without design we risk building an unstable design -one that will fail when small changes are made, one that may be difficult to test, and one whose quantity cannot be accessed until late in the software engineering process.

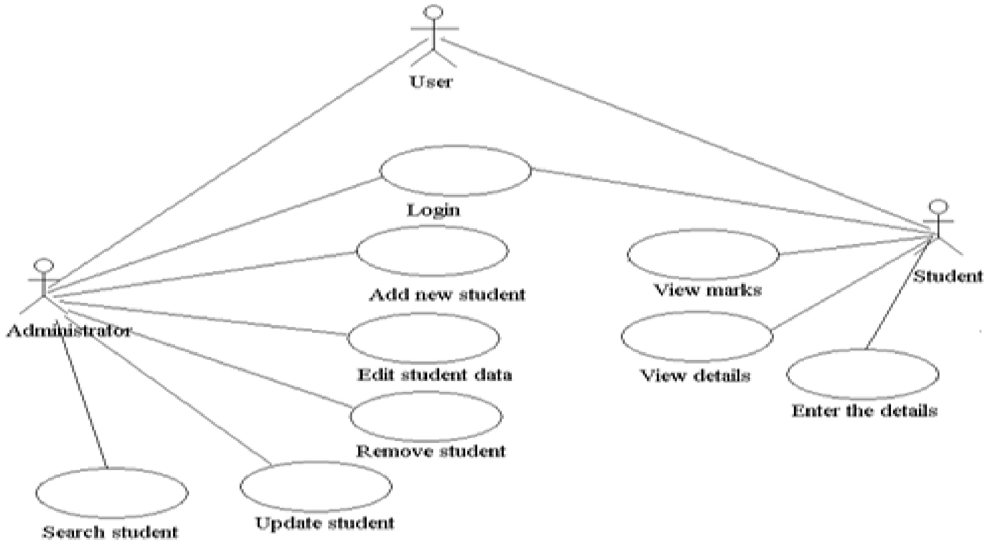
Taking software requirements specification document of analysis phase as input to the design phase we have drawn Unified Modeling Language (UML) diagrams. UML depends on the visual modeling of the system. Visual modeling is the process of taking the information from the model and displaying it graphically using some sort of standards set of graphical elements.

UML Diagrams are drawn using the Pace Star UML Diagrammed Software. We seem to able to understand complexity better when it is displayed to us visually as opposed to written textually. By producing visual models of a system, we can show how system works on several levels. We can model and the interactions between the users and the system.

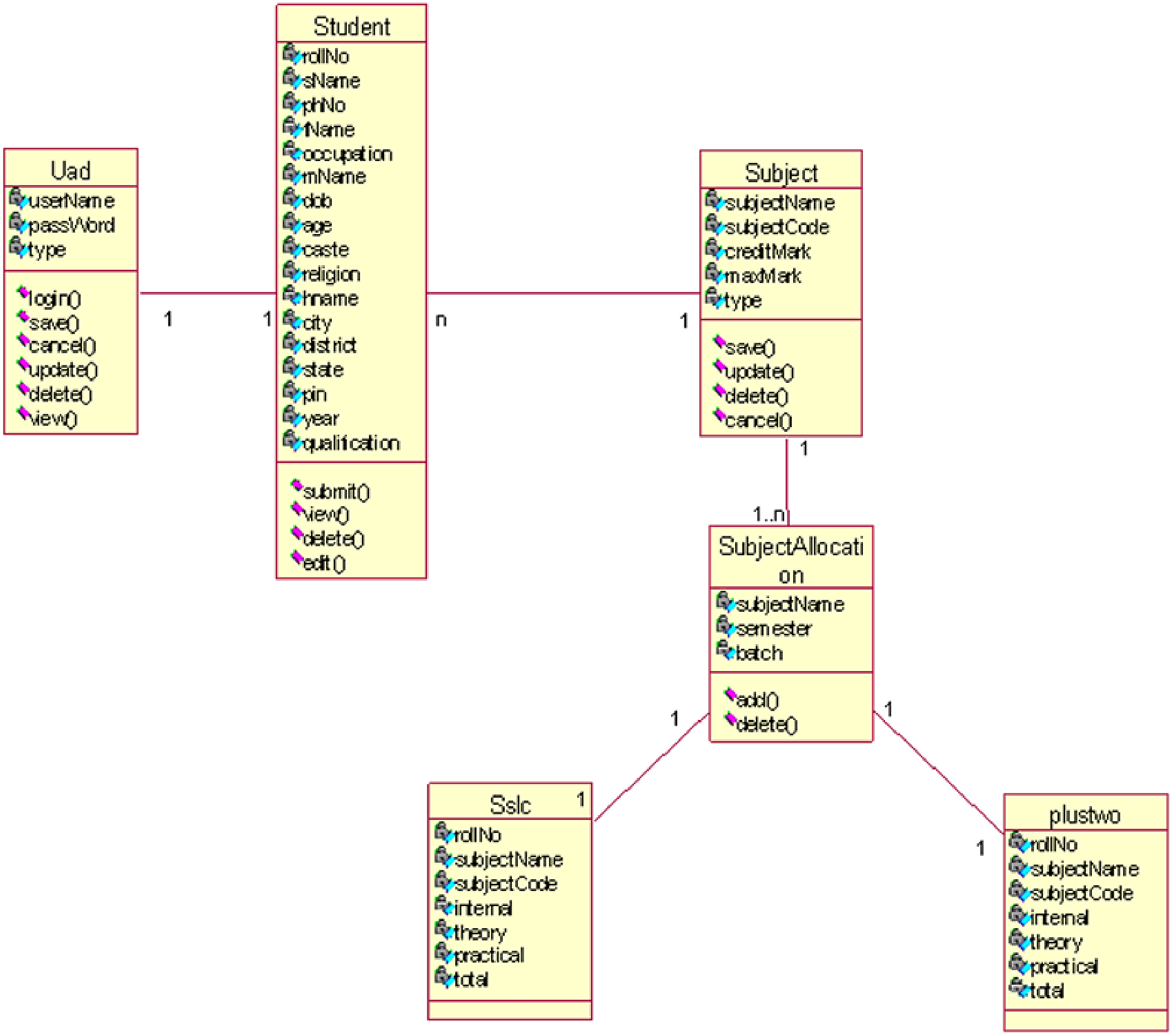
Types of UML Diagrams:

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modeling tools include

Use Case Diagram displays the relationship among actors and use cases.

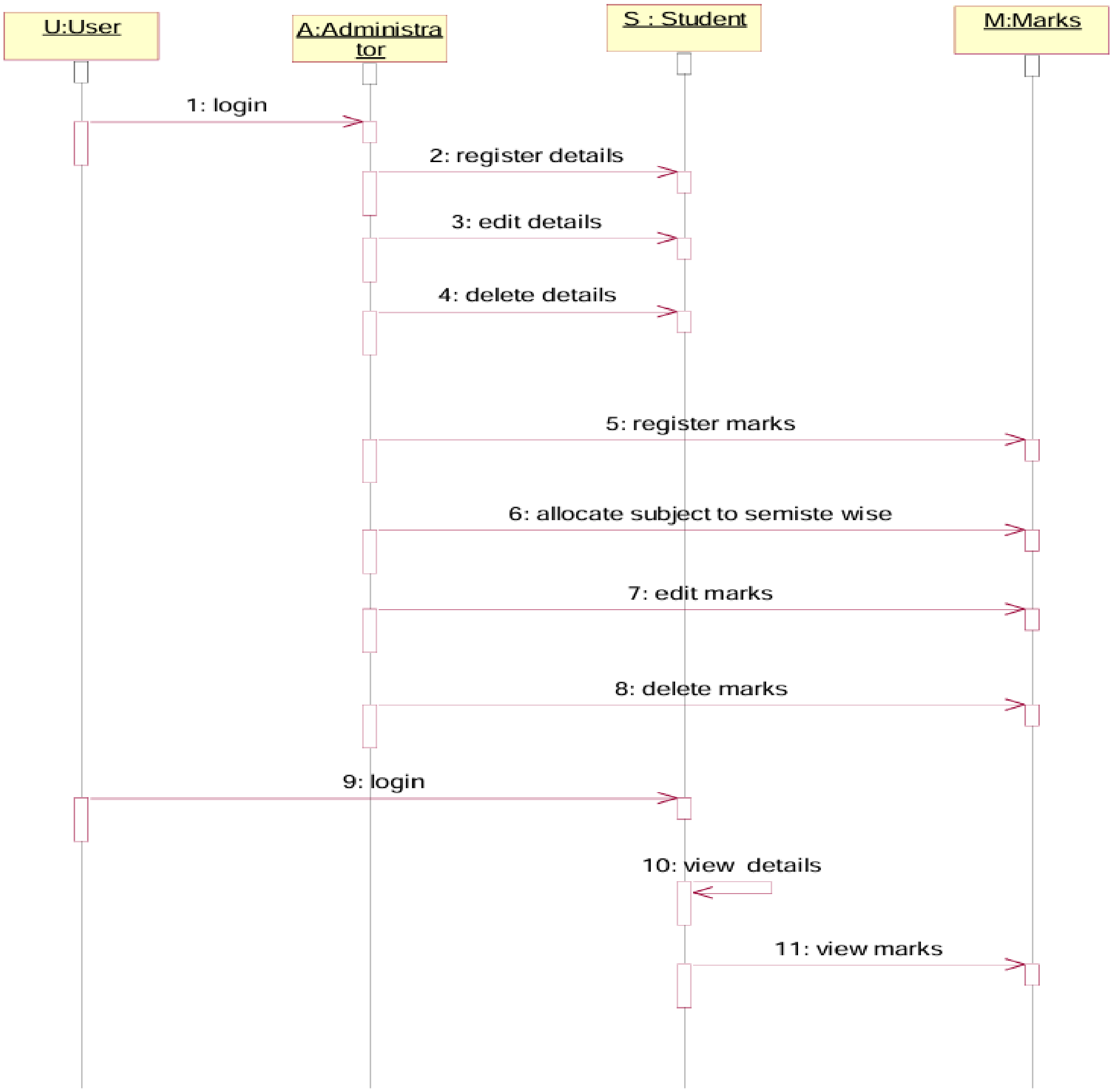


Class Diagram models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.

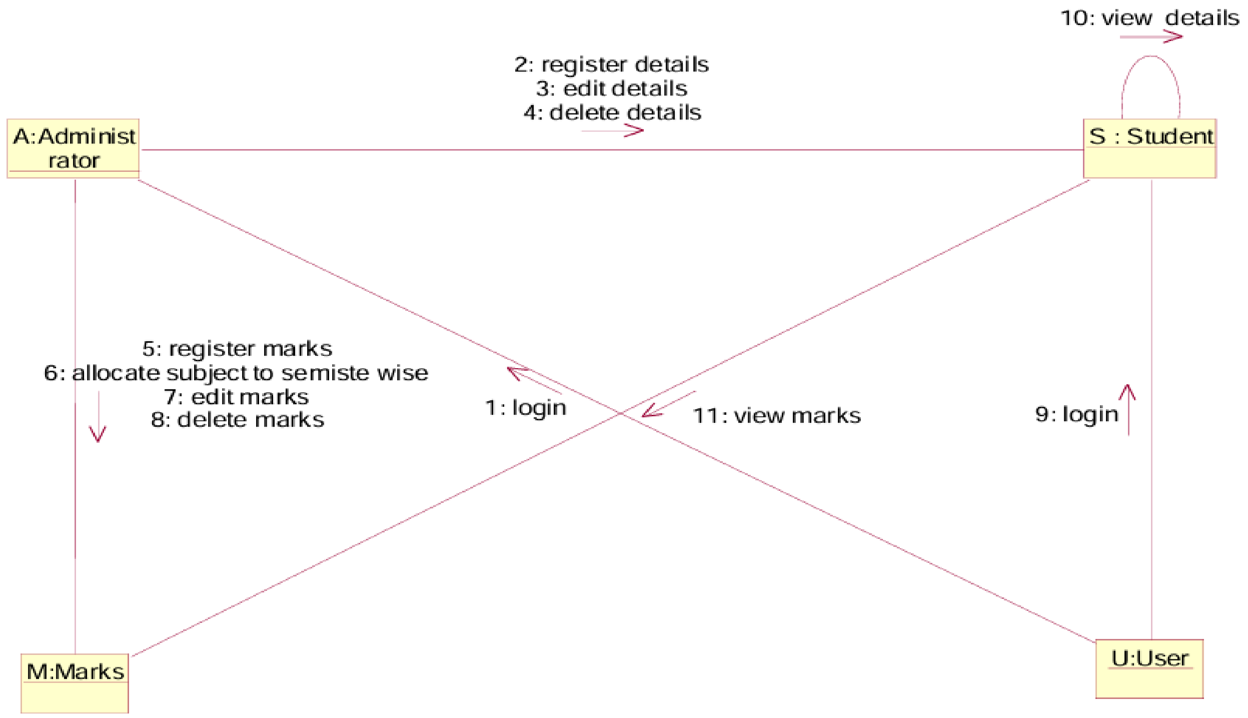


Interaction Diagrams:

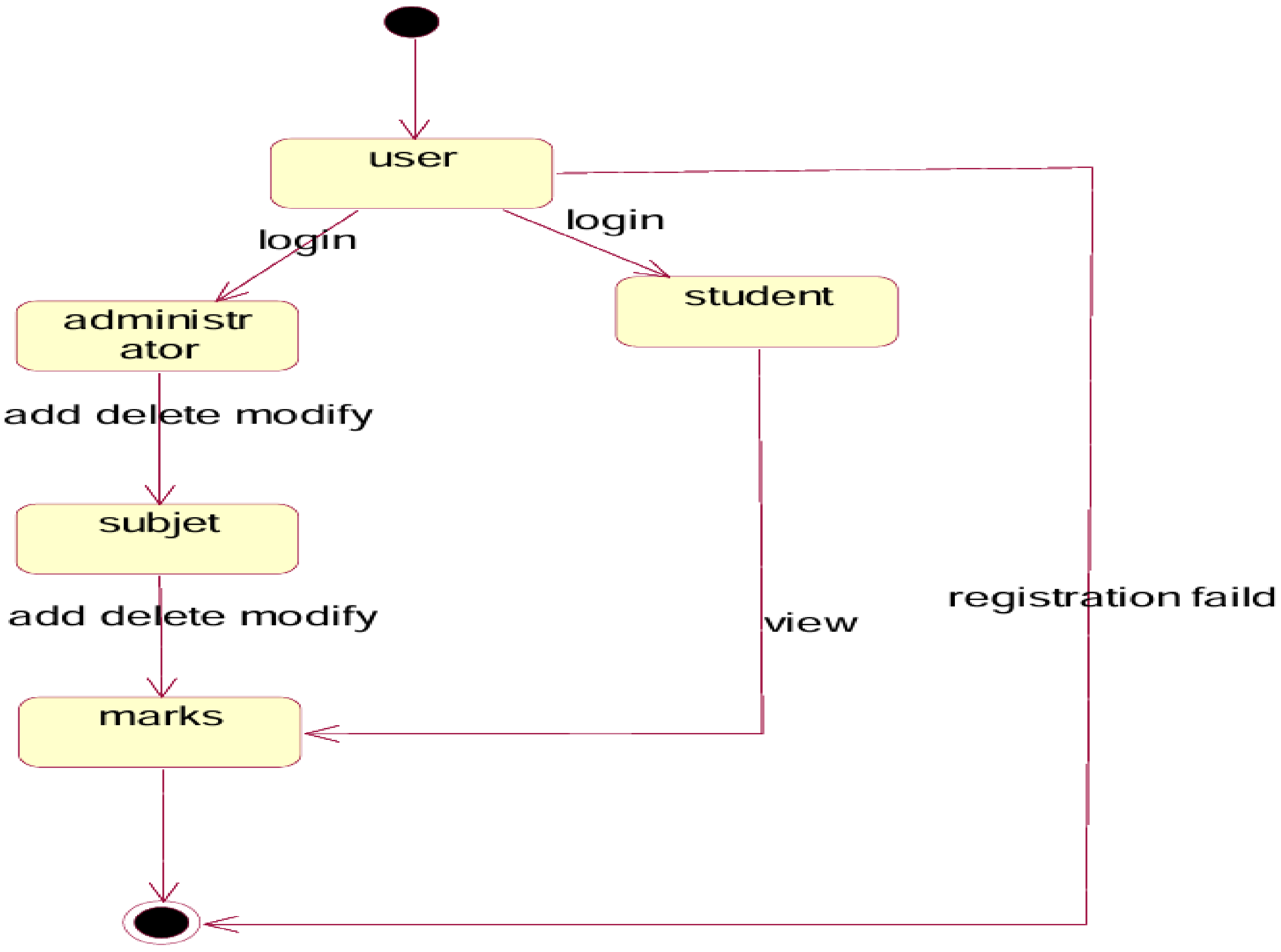
Sequence Diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).



Collaboration Diagram displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.



State Diagram displays the sequence of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.



Activity Diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

Physical Diagrams:

Component Diagram displays the high level packaged structure of the code itself. Dependencies among components are shown; include source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time.

Deployment Diagram displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code units.

DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy quick and flexible for user. In database design several objectives are considered.

Control Redundancy:

Redundant occupies space and therefore, is wasteful. If versions of the data are in different phases of updating the system often gives conflicting information. A unique aspect of database design is storing only once, which controls redundancy and improves system performance.

E-R DIAGRAMS:

Entity-Relationship Model:

The Entity-Relationship data model is based on a perception of a real world, which is consists of set of basic object called entities and relationships among these objects.
An entity is an object that exists and is distinguishable from other objects/entity is an object as a concept meaningful to the organization. An entity set is a set of entities of the same type. A primary key is an attribute which when take, allows us to identify uniquely an entity in the entity set.

DATA FLOW DIAGRAM

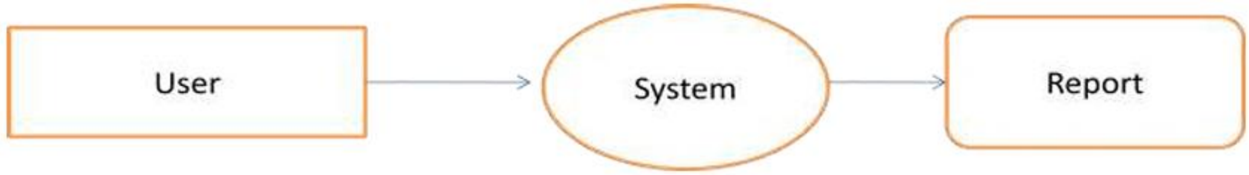
A **data-flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

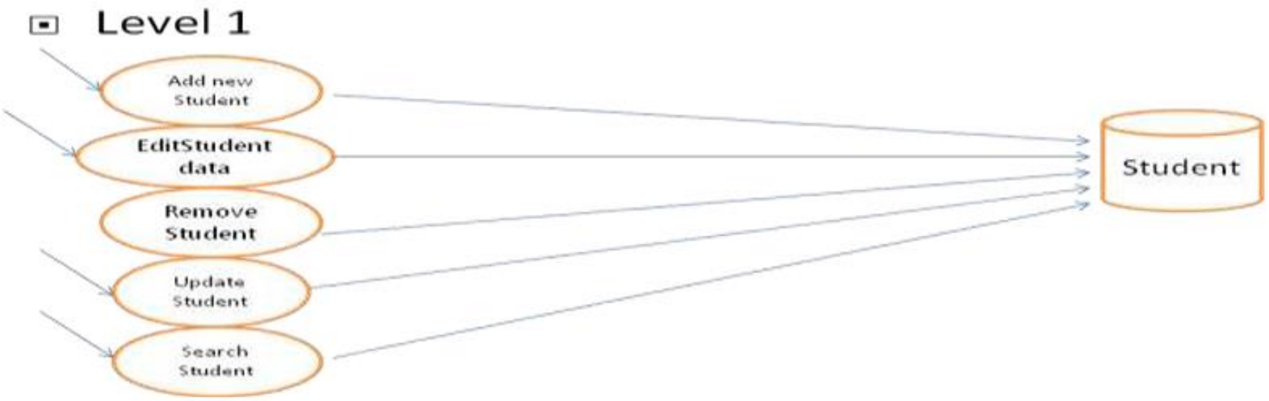
A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

Context Diagram

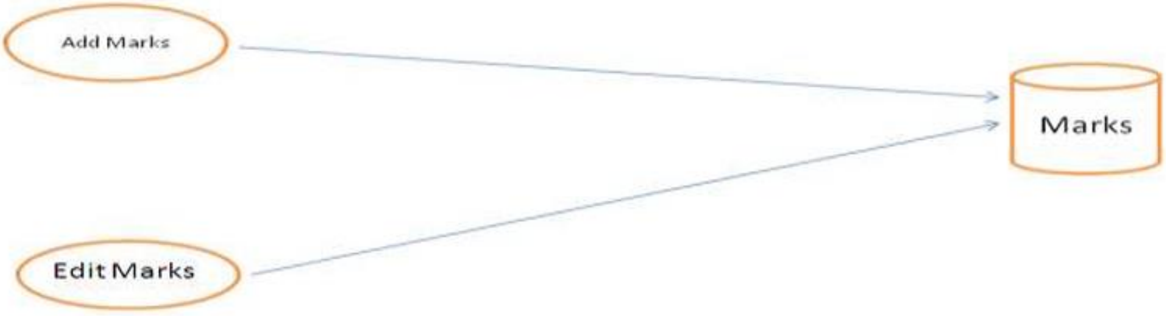
USER

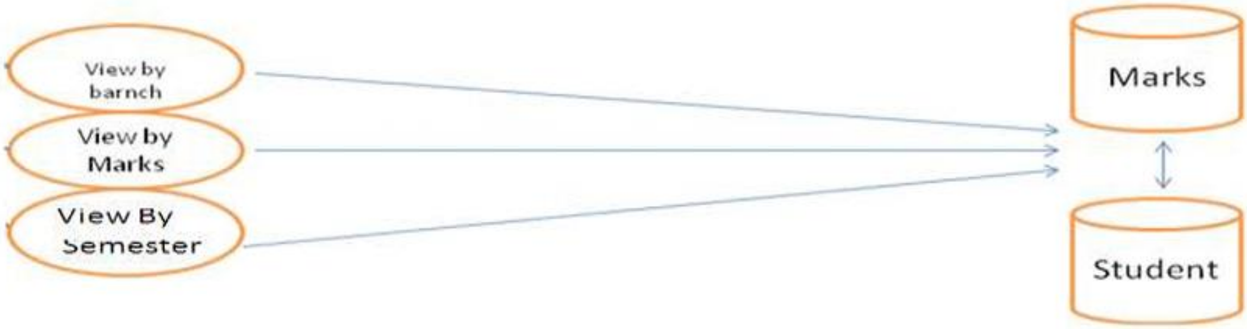


STUDENT



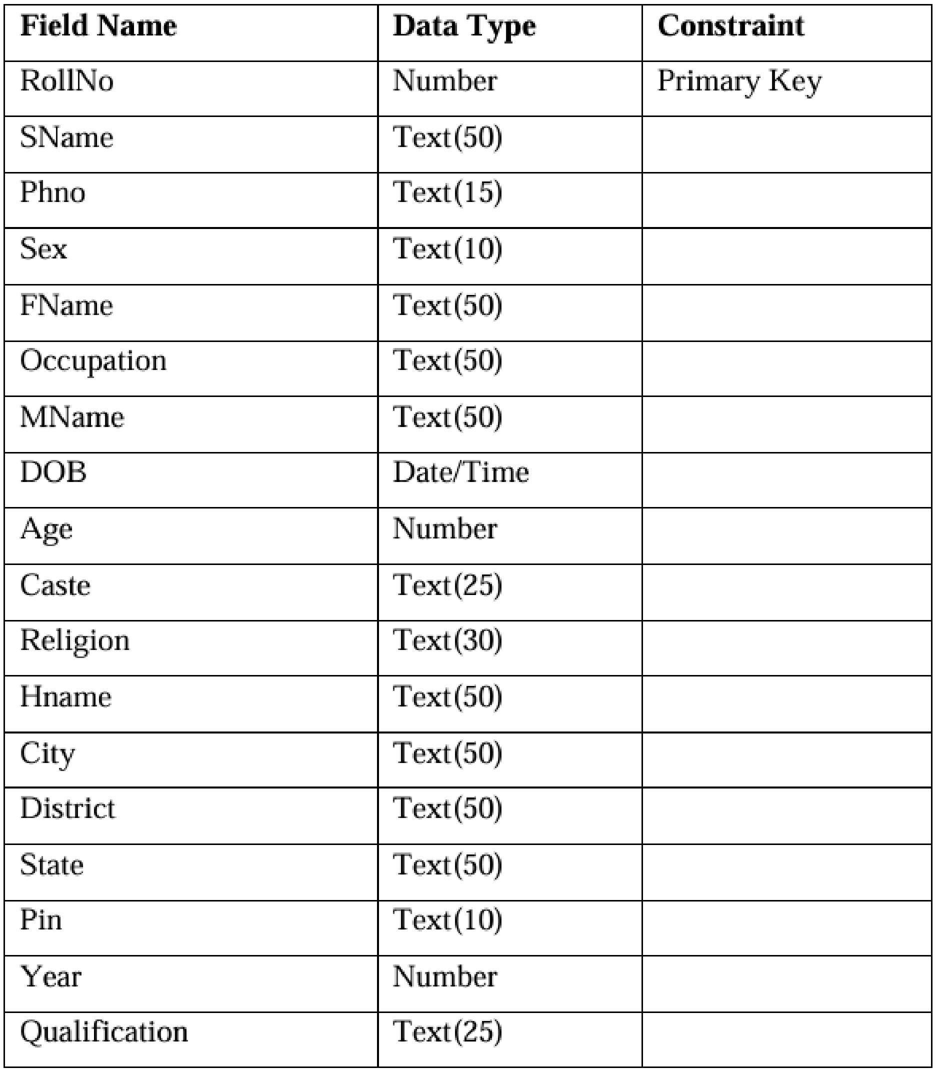
MARKS


Add Marks
Marks
Edit Marks


View by barnch
View by Marks
View By Semester
Marks
Student

TABLES STRUCTURES

Student Table



Field Name	Data Type	Constraint
RollNo	Number	Primary Key
SName	Text(50)	
Phno	Text(15)	
Sex	Text(10)	
FName	Text(50)	
Occupation	Text(50)	
MName	Text(50)	
DOB	Date/Time	
Age	Number	
Caste	Text(25)	
Religion	Text(30)	
Hname	Text(50)	
City	Text(50)	
District	Text(50)	
State	Text(50)	
Pin	Text(10)	
Year	Number	
Qualification	Text(25)	

UAD Table

Field Name	**Data Type**	**Constraint**
Username	Text(25)	Primary Key
Password	Text(15)	
Type	Text(15)	

Subjects Table

Field Name	**Data Type**	**Constraint**
Subjectcode	Text(10)	Primary Key
Subjectname	Text(50)	
Creditmark	Number	
MaxMark	Number	
Type	Text(25)	

SubjectAllocation Table

Field Name	**Data Type**	**Constraint**
Subjectname	Text(50)	
Semester	Number	Primary key
Batch	Text(15)	

SSLC1 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SSLC2 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SSLC3 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SSLC4 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SSLC5 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SSLC6 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign Key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

PLUSTWO1 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

PLUSTWO2 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

PLUSTWO3 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

PLUSTWO4 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

PLUSTWO5 Table

Field Name	Data Type	Constraint
RollNo	Number	Foreign key
SubjectName	Text(50)	
Subjectcode	Text(15)	Foreign Key
Internal	Number	
Theory	Number	
Practical	Number	
Total	Number	

SYSTEM IMPLEMENTATION

Introduction

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project. They are

- End user training
- End user Education
- Training on the application software
- System Design
- Parallel Run and To New System
- Post implementation Review

End user Training:

The successful implementation of the new system will purely upon the involvement of the officers working in that department. The officers will be imparted the necessary training on the new technology

End User Education:

The education of the end user start after the implementation and testing is over. When the system is found to be more difficult to understand and complex, more effort is put to educate the end used to make them aware of the system, giving them lectures about the new system and providing them necessary documents and materials about how the system can do this.

Training of application software:

After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

Post Implementation View:

The department is planning a method to know the states of t he past implementation process. For that regular meeting will be arranged by the concerned officers about the implementation problem and success.

SYSTEM TESTING

Introduction

Is the menu bar displayed in the appropriate contested some system related features included either in menus or tools? Do pull –Down menu operation and Tool-bars work properly? Are all menu function and pull down sub function properly listed ?; Is it possible to invoke each menu function using a logical assumptions that if all parts of the system are correct, the goal will be successfully achieved .? In adequate testing or non-testing will leads to errors that may appear few months later.

This create two problem

1. Time delay between the cause and appearance of the problem.
2. The effect of the system errors on files and records within the system.

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

Testing Methods

There are two major type of testing they are

1) White Box Testing.
2) Black Box Testing.

White Box Testing

White box sometimes called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests were made on the system

a) All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed.

b) All logical decisions were checked for the truth and falsity of the values.

Black box Testing

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like..

1) Interface errors

2) Performance in data structure

3) Performance errors

4) Initializing and termination errors

Unit testing

Unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use.

A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure.

Ideally, each test case is independent from the others: substitutes like method stubs, objects, fakes and test harnesses can be used to assist testing a module in isolation.

Integration Testing:

This testing is sometimes called Integration and Testing. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group.

It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan

Validation Testing:

Validation Testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonably expected by a customer. After validation test has been conducted, one of the following two possible conditions exists. The functions or performance characteristics confirm to specification and are accepted.

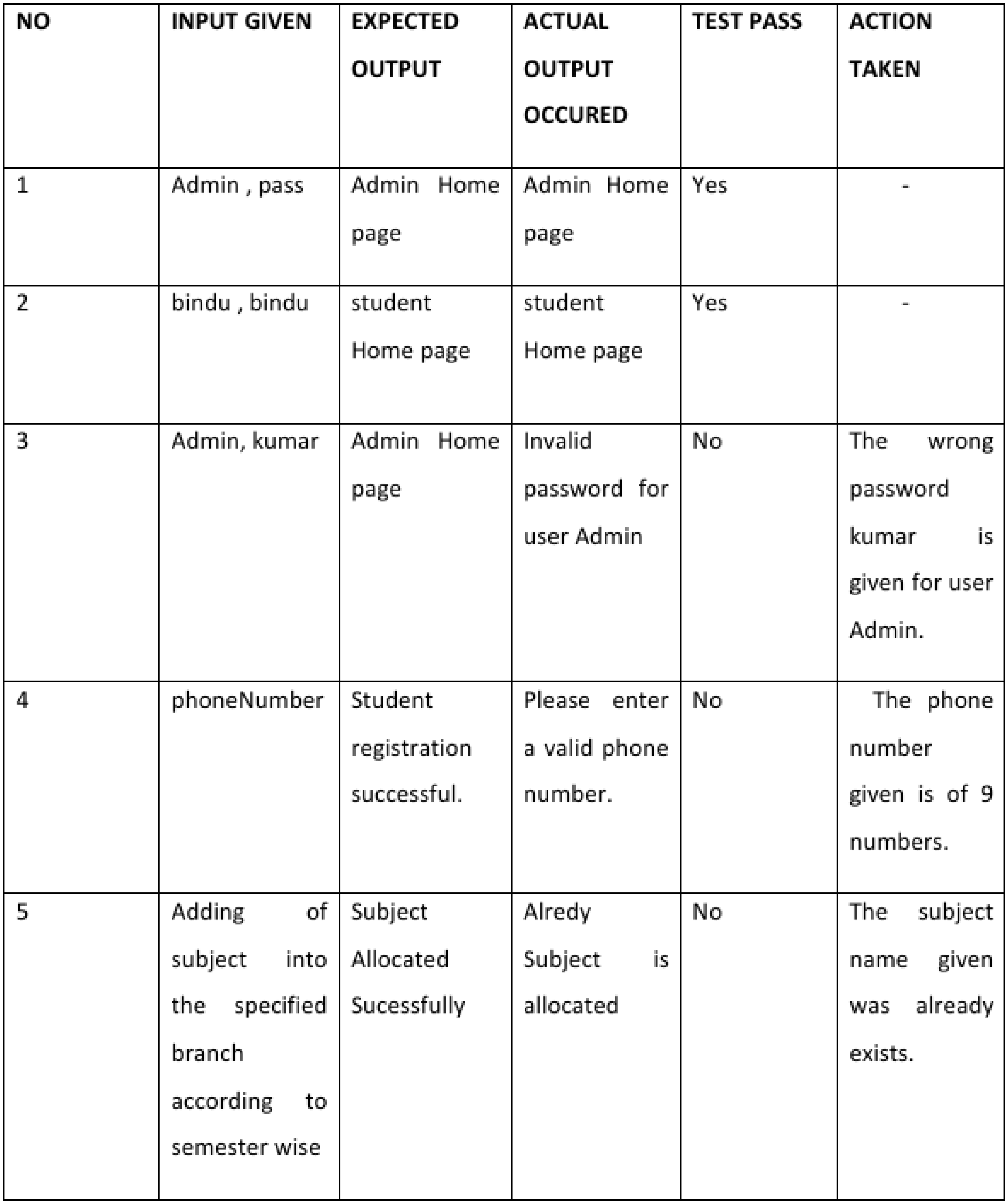
- In the administrator and marks modules, all the fields must be filled.
- In the student registration, mobile number should contain exactly 10 numbers.

User Acceptance Testing:

User acceptance of a system is a key factor of any system. The system under consideration is tested for the acceptance by constantly keeping in touch with the prospective system users at the same time of developing and marketing changes whenever required. This is done in regard to the following points:

- Input Screen Design
- Output Screen Design

Test Cases



NO	INPUT GIVEN	EXPECTED OUTPUT	ACTUAL OUTPUT OCCURED	TEST PASS	ACTION TAKEN
1	Admin , pass	Admin Home page	Admin Home page	Yes	-
2	bindu , bindu	student Home page	student Home page	Yes	-
3	Admin, kumar	Admin Home page	Invalid password for user Admin	No	The wrong password kumar is given for user Admin.
4	phoneNumber	Student registration successful.	Please enter a valid phone number.	No	The phone number given is of 9 numbers.
5	Adding of subject into the specified branch according to semester wise	Subject Allocated Sucessfully	Alredy Subject is allocated	No	The subject name given was already exists.

CONCLUSION

Our project is only a humble venture to satisfy the needs in an Institution. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

Flask

1. Introduction

Flask is a microframework for web development in Python. It was created by Armin Ronacher in 2010 as part of the Pallets Project. Flask is designed to be simple and flexible, allowing developers to scale applications from small prototypes to large-scale projects.

2. Key Features

- Lightweight and Modular: Flask is built with simplicity in mind, providing core functionality without unnecessary overhead.

- Built-in Development Server: Includes a debugger and a server for testing and development purposes.

- RESTful Request Handling: Supports the development of APIs and web applications using the REST architecture.

- Jinja2 Templating Engine: Offers a powerful and secure templating engine for dynamic HTML generation.

3. Core Components

Flask Class: The primary entry point to create an application instance.
Routing: Allows defining URL rules for application endpoints.
Request and Response: Manages incoming requests and outgoing responses seamlessly.
Blueprints: Facilitates modular application design by splitting applications into smaller, reusable components.

4. Advantages

- Simple and easy to learn for beginners.
- Flexible structure that adapts to various use cases.
- Rich ecosystem of extensions for adding functionality.
- Active community and excellent documentation.

5. Limitations

- Not suitable for very large applications without careful structuring.
- Lacks built-in features like database support (compared to Django).
- Requires more third-party tools to achieve full-stack capabilities.

6. Use Cases

- Developing small to medium-sized web applications.
- Creating RESTful APIs.
- Rapid prototyping of web ideas.
- Building single-page or microservices architectures.

8. Conclusion

Flask is an excellent choice for developers seeking a balance of simplicity and power. It’s widely used in industries ranging from startups to enterprises for creating scalable, maintainable, and high-performing web applications

Introduction to SQL

Database A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence).

SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

Relational Database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

Foreign Key

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

Advantages of RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared
- Standards can be enforced
- Security restrictions ca be applied
- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

Disadvantages of DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

Features of SQL Server(RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability.

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are: The row level lock manager.