

Ex:2 Stakeholders and Process Models

Stake holders:

1. Executive leadership
2. IT leaders
3. Inventory Managers
4. Sales Managers
5. Supply chain Managers

User Description:

1. Executive Leadership: Required to approve the resources needed to build the system and obtain required data.
2. IT Leaders: Consist of the Project Team and Developers needed to code the system and test it.
3. Inventory Managers: Consist of the users that use the system to log and modify inventory.
4. Sales Managers: Consist of the users that obtain data from the system to carry out sales.

Supply Chain Partners: Required for keeping track of items sold and demand for a certain product or service.

User Story:

Executive Leadership: Executive Leadership consists of the team used to plan and approve the allocation of needed resources and govern the stages of the project. They also receive the benefit of having a system to keep track of the company's inventory, as the

data recorded by the system is helpful for effective decision making and for setting measurable objectives.

IT Leaders : IT Leaders direct the project team and developers that plan, build and test the system.

Inventory managers: Inventory managers are the primary users of the system. They use it to record and modify the inventory. This helps them report data that helps determine the resources the organization needs to invest in.

Sales managers: Sales managers benefit from an inventory system as it keeps track of the products that are sold and the products not in popular demand. This helps important functions that help them carry out sales, such as adjusting sales strategies to advertise an unpopular product or to create new strategies to appeal to an untapped demographic.

Supply Chain partners: Supply chain partners require data recorded in an inventory system in order to keep track of items sold and a demand for a certain product or service. Since supply chains sell the products that arrive from the inventory, requests for a popular product that runs out of stock easily, choosing to not sell an unpopular product and other functions affect the inventory and thus, affect the inventory system. A system compatible with the existing system of the supplier is necessary.

Identifying the appropriate process model:

A process is defined as:

A defined order of tasks or activities spread through time and space, with a beginning, an end, and clearly defined inputs and outputs.

Thus a process model is used to understand the process and describe it. The process model that will be used to build the inventory management system will be the waterfall model.

This model is divided into the following phases:

1. Feasibility study: The deadline to build the system is set. The feasibility study focuses on the resources needed to build the system and the problem that it solves. The project is finalized in this phase.
2. Requirements analysis: This is the analysis of the requirements of the project, such as the information needed to be gathered, such as information on how an inventory is managed manually, the features of the programming language that we need to use to build the system and the specifications of the system required by the customer. The requirements of the system are well documented in order to set objectives and evaluate every part of the process of building it.
3. Design: The requirements specified in the documentation are converted into a structure that can be implemented by a programming language.
4. Coding and Unit Testing: The code for the system is written and any errors are identified by testing and

rectified. The code is written in various modules and each module is tested.

5. Integration and system testing: The modules are integrated into one system and the full working system is tested to correct any remaining errors. After deployment, the customer engages in acceptance testing, which allows them to provide feedback on the system.
6. Maintenance: The maintenance of the system is a continuous process. This includes adding new features to the product, fixing bugs and modifying the product to meet the changing needs of the customer. This keeps the produced system flexible and prevents it from becoming obsolete.

Problem Statement:

To coordinate the executive and IT units of an enterprise to design and maintenance of an inventory system useful to inventory managers of the enterprise. The system must be compatible with the system of the supply chain partners and must be useful in providing data to the executive, IT and sales units of the organization, in order to promote effective production of goods and services.

Comparative study with the agile model:

The agile model prioritizes customer satisfaction above all else. As opposed to a conventional process model, an agile model's development life cycle is divided into smaller phases called iterations. Each iteration consists of Planning, Requirements analysis, Designing, Building and Testing. Each iteration of the model results in a working software build, and the final iteration consists of all the

features required by the customer. Unlike the waterfall model, the agile model produces a working product at every stage, and relies on repeated consumer feedback. In a conventional model each stage of the process finalizes an aspect of the product, while in the agile model the product goes through updating at every stage.

Modules:

The program is divided into the following:

- Login: This module is involved in checking the password and logging the user in.
- Update: This module keeps track of the inventory, by using functions that update the stock.
- Margin: This module helps in calculating the profit and loss margins for a product in the inventory.
- Display: It has a function that displays the inventory in a table format.
- Check: This module includes a function that searches the table for the particular product.

Ex 3: IDENTIFYING THE REQUIREMENTS FROM PROBLEM STATEMENTS

Requirements Elicitation

It is the process of researching and getting to know about the requirements of a system from the users, customers, and mainly concerned to incorporate the business requirements of the stakeholders. The practice is also referred to as "requirement gathering".

Customer requirements:

1. User login page: The user enters the password and the system verifies and logs the user in.
2. Displaying the functions of the system: Displaying all the operations that can be performed on the inventory table .
3. User interface: Using buttons to display the options and make interaction with the program easier.
4. Changes made to the inventory are reflected in the database - displays the changes made immediately in table form.
5. Displaying the inventory table.
6. Calculating profit and loss margins from the inventory database.
7. Predicting future sales using existing data.
8. Notifications - Displaying messages when a product goes out of stock and needs to be reordered.

Developer specifications:

1. User login: Can verify entered password by reading and checking correct password with file.
2. Displaying the functions of the system: Functions of the system can be displayed.
3. User Interface: Can be made with buttons using software like the tkinter library in Python.
4. Changes made to the inventory are reflected in the original database: Done using files
5. The inventory table can be displayed
6. Profit and loss margins can be calculated from inventory data

This is done to ensure that:

1. Identify what the users or owners want to see .
2. The system must also make sure it is not compromised on the profit factor and quality of the project (stakeholder requirements), thus satisfying the user end as well as the stakeholder side.

Requirement elicitation techniques:

a. Interviews: Objective of conducting an interview is to understand the customer's expectations from the software. Representatives from related groups of the project are selected based on their expertise and credibility.

b. Brainstorming Sessions: It is a group technique which is intended to generate lots of new ideas hence providing a platform to share views and handle group bias and group conflicts. Every idea is documented so that everyone can see it.

c. Facilitated Application Specification Technique (FAST): Its objective is to bridge the expectation gap – difference between what

the developers think they are supposed to build and what customers think they are going to get.

d. Quality Function Deployment (QFD): In this technique customer satisfaction is of prime concern, hence it emphasizes on the requirements which are valuable to the customer.

e. Use Case Approach: This technique combines text and pictures graphically to provide a better understanding of the requirements. The use cases describe the 'what', of a system and not 'how'. The UML Use case diagram best represents the most effective way among the other UML diagrams to represent our problem statement.

Technical Feasibility

The school management system is web based and thus can be accessed through any browsers. The solution is practical since the objectives of the system development are achievable and realistic. The technology to be used is available, this includes use of programming language PHP and HTML and MySQL database to develop web based applications.

Economic Feasibility – Costs and benefits

a) Costs

Development Costs

Development cost is very minimal since the tools and technologies used are available online. It's a group project so there are no personnel costs. Development time is well planned and will not affect other operations and activities of the individuals. Once the system has been developed, the companies purchasing the system will be provided with a manual for training purposes, also physical training will be provided for those individuals that need it. There is no need to purchase new hardware since the existing computers can still be used.

b) Benefits

i. Performance Benefits

Increased speed of report production.Faster creation, access, modification and retrieval of data.Decreased redundancy or duplication of data.Timely access of students' information and school update.Improved interaction between parents and teachers.

ii.Cost-Avoidance Benefits

Avoid costs of purchasing data backups and storage since everything will be online and automated. Avoid costs of parents having to travel long distances to check their kids' performance.Staff Reductions will reduce the cost of salaries since and integrated system will need just a few employees, no need for employees in every department.It is evident that the system is economically feasible, the benefits outweigh the system costs within the defined period of time acceptable to the user/client. The use of system places the school at a competitive advantage.

FUNCTIONAL REQUIREMENTS

Student:

1. The Student shall be able to login to System.
2. The Student shall be able to Upload Assignments to System.
3. The Student shall be able to Download Assignments, and lectures from the System.
4. The Student Shall be able to Solve Exams through Webpage within 1 Hours of the time it was posted in the system.
5. The Student shall be able to View and print the Grade Report.
6. The Student shall be able to View the Available Courses in the

Student Plan.

Teacher:

1. The Teacher shall be able to login to System.
2. The Teacher shall be able to Grade Student's Exams, uploaded Assignment
3. The Teacher shall be able to Post Assignment with their time due date.
4. The Teacher shall be able to Post Course's lectures.

Admin:

1. The Admin Shall be able to Login to the system.
2. The Admin Shall be able to Create User login for Student/ Teacher.
3. The Admin Shall be able to add courses.
4. The Admin Shall be able to delete courses.
5. The Admin Shall be able to edit courses.
6. The Admin Shall be able to assign Teachers to courses.
7. The Admin Shall be able to remove courses form teachers

8. The Admin Shall be able to change Teachers assigned to courses.
9. The Admin Shall be able to assign Students to courses
10. The Admin Shall be able to remove courses form Student.
11. The Admin Shall be able to Change Courses assigned to students.

Non-Functional Requirements

.Availability : The system should be available 24 hours per 6 days.

Security :The system should encrypt users password by using AES-256 encryption system.

Usability : The users should learn how to use the system at most in 2 hours.

Maintainability :The system should be closed 24 hours per week for maintenance and elaboration.

Reliability :The system should never crash more than 10 minutes per month.

System Requirements:

The System needs the ability to do the following tasks as its requirements:

1. Manage sale and purchase of Goods: products purchased and sold by the Enterprise.
2. Govern the Stock: account the products that have not been sold yet
3. Return of products: the products that have been returned due to errors
4. Profit/loss: Calculate overall Profit/Loss incurred by the Firm.

Software/ Hardware Requirements:

For implementing this software efficiently ,the following software and hardware requirements are essential.

1. 7th generation or above CPU

2. Monitor
3. Windows 7 or higher OS
4. The system must be connected to a server
5. Python programming language for programming the software.
6. GUI library tkinter
7. C++ and VS code

Ex 4: PROJECT PLAN

School Management Software is a complete and customized solution for a school handling regular management work. This software includes the management of students, Employee, Accounts, Transportation, library, Fees, Exam, Time Table, Inventory and its respective reports with administrative module to provide better security.

Project Members

1. SukanyaSahoo – RA1811026020043
2. Ankit Yadav – RA1811026020055
3. Apoorv Shekher – RA1811026020039

Cost Estimation and Budget

Requirement	Cost
Visual Studio Enterprise 2019	INR 54,000
Team Workforce	INR 50,000
Documentation	INR 1500
Anaconda for Python	Free of Cost

Project Schedule

TASK	START DATE	END DATE
Business case development	23-12-2019	29-12-2019
Identify Stakeholders, Process Models and Required Modules	30-12-2019	05-01-2020
Identify requirements	01-01-2020	06-01-2020
Setting cost estimates and budget	07-01-2020	11-01-2020
Uml Diagrams	12-02-2020	22-02-2020
Coding	23-02-2020	16-03-2020
Final Revision	17-03-2020	20-03-2020
Project Delivery	21-03-2020	-

System Requirements and Resources

1. Processor : Intel core i5 (& above)
2. RAM : Minimum 4 GB
3. System space requirements : Minimum 5 GB
4. Windows 8 or higher

Modules

1. Login system
2. Student Database Management

3. Teacher Database Management
4. Fee submission
5. Bus Transport Management
6. Feedback

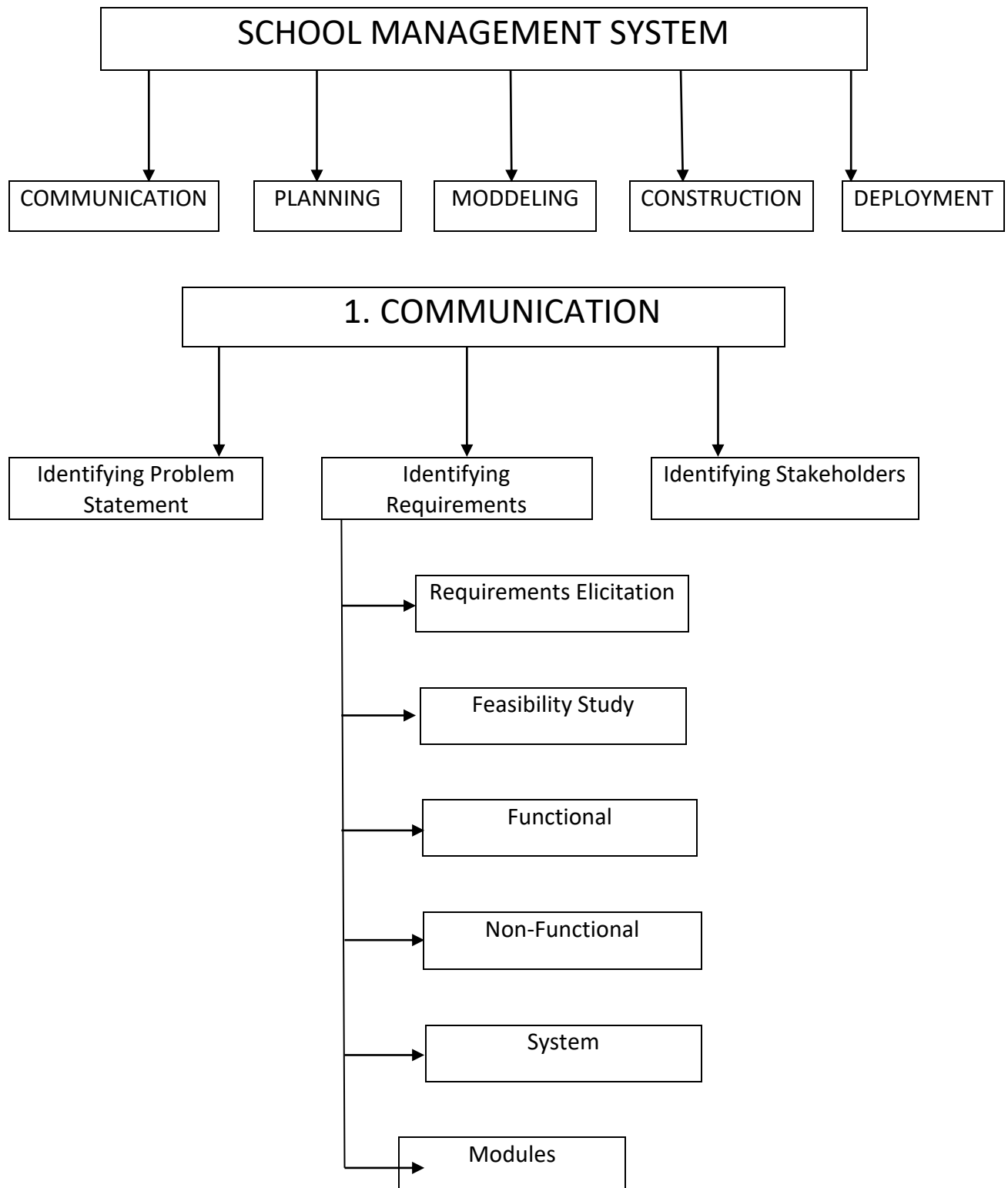
Job Roles and Responsibility

Member	Role & Responsibilities
SukanyaSahoo RA1811026020043	<ul style="list-style-type: none">• Team leader: Has the responsibility of coordinating and guiding the team for efficient and successful accomplishment of the project• Developer: Responsible for coding the main framework module and design.• Team Member: Has responsibility of contributing to documentation, fixing timeline of the project.
Ankit Yadav RA1811026020055	<ul style="list-style-type: none">• Team Member: Responsible for contributing to the documentation, reviewing, providing expertise and technical details.• Developer: Responsible for coding the management services which include login and update modules.
Apoorv Shekher RA1811026020039	<ul style="list-style-type: none">• Team Member: Responsible for contributing to the documentation, reviewing and providing technical expertise.• Developer: Responsible for coding, testing the software and debugging.

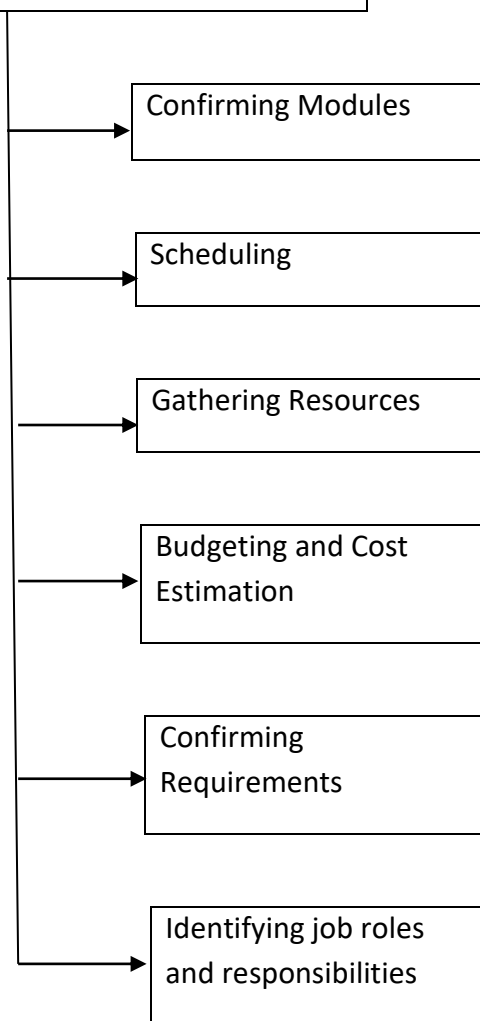
Ex 5: Project effort based on resources

WORK BREAKDOWN STRUCTURE

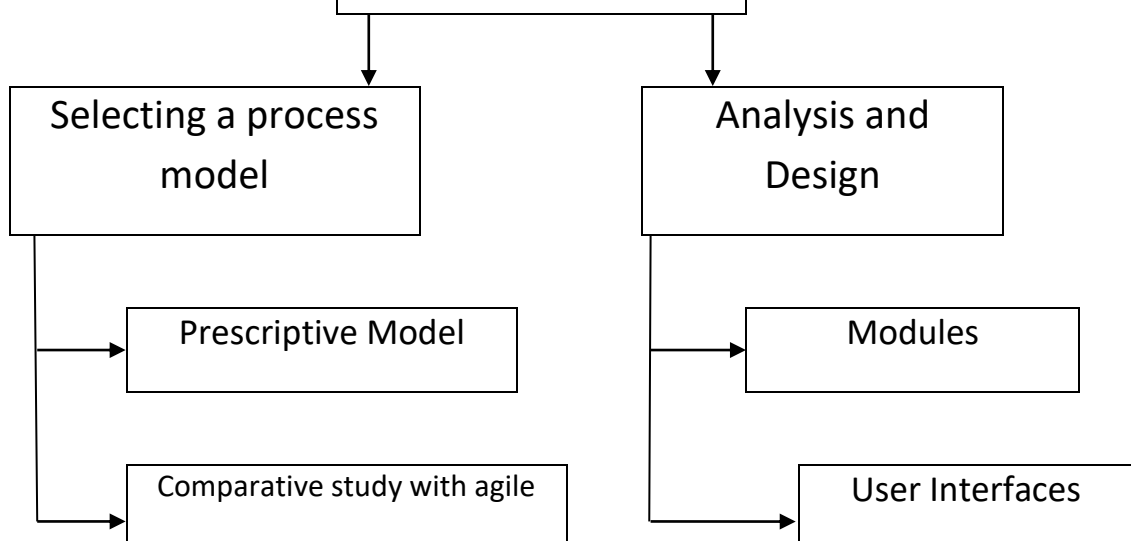
The work breakdown structure of the project is given below:

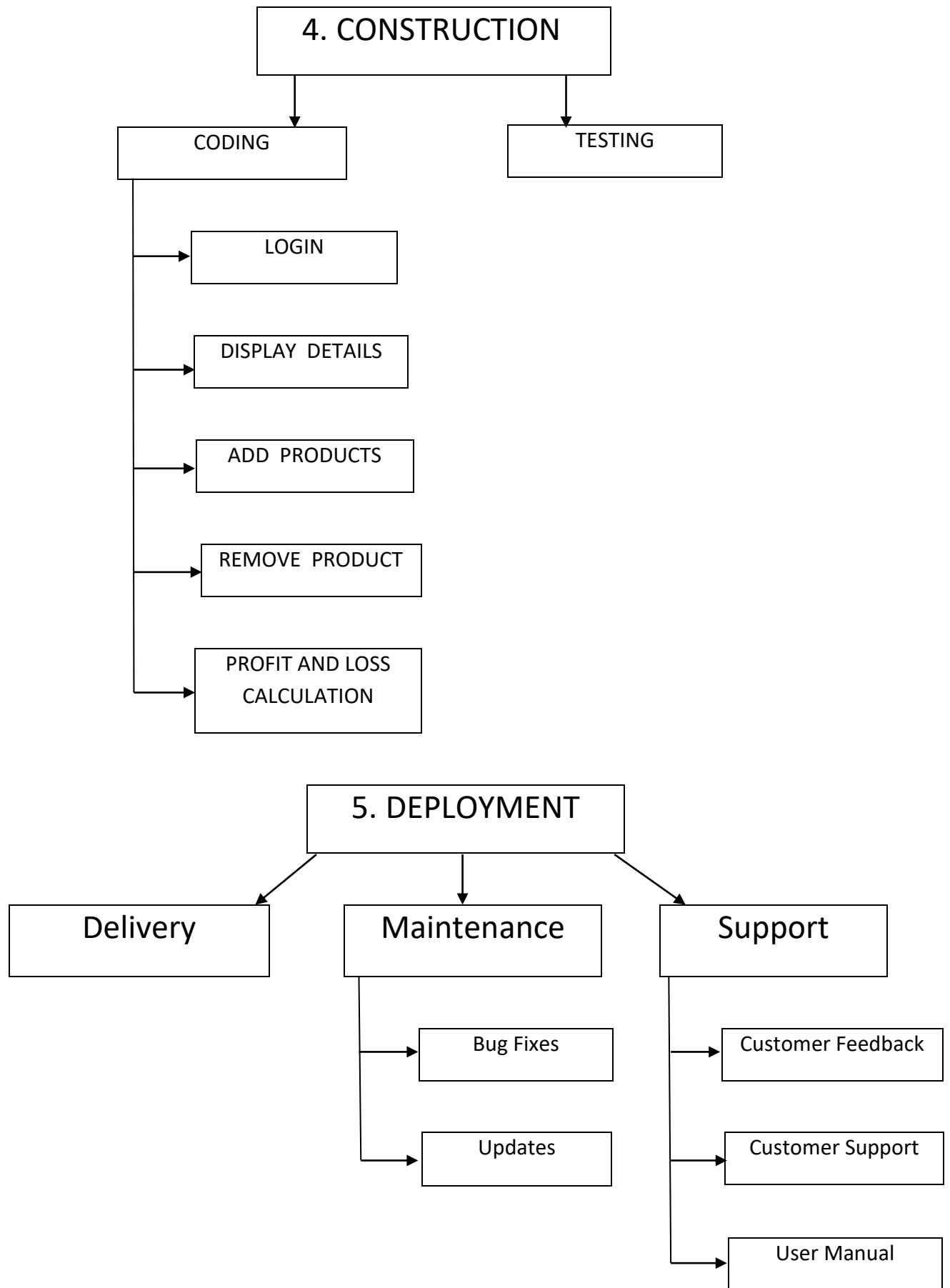


2. PLANNING



3.MODELING





RISK ANALYSIS

RISK	IMPACT	MITIGATION
TECHNICAL: Fault in hardware or unavailability of hardware at required time	1. Delay in Project Delivery 2. Loss of data and loss of time 3. Dissatisfaction from customers	Resolving the technical difficulties at an early stage to prevent loss of data and time.
ORGANIZATIONAL: Project dependencies, logistics, Resources, budgets, etc.	1. Improper or troubled execution of the project. 2. Dissatisfaction from the project team can be expected as well	The organisation has to ensure the timely availability of all kinds of support to the project team in order to ensure the successful completion of the project.
PROJECT MANAGEMENT: Communication, Planning, Modelling, Controlling, Delivery	1. Affects the complete development and delivery of the project	The project team needs to be meticulous in achieving the timely milestones of the project development to avoid drastic impact.
EXTERNAL: Customer, Contract, Markets	1. If the product is not well received in the market or by the by the customer or if the customers are not satisfied with the product it would lead to waste of efforts and resources.	Customer and market requirements need to be understood clearly before the development of the project. Periodic consultation with the customer in clarifying the needs is essential.

SWOT analysis:

Strengths: Cost effective, simple, easy to use	Weaknesses: offline, slow, cannot handle complex operation
Opportunities: for small business	Threat: Incompatible with user's system, vendor's database

Ex 6: Estimation of project metrics

Function point analysis| COCOMO model:

Measurement parameter	Count		Weighing factor	Product
1.Number of External Inputs(EI)	3	*	3	9
2. Number of External Outputs(EO)	5	*	4	20
3.Number of External Inquiries (EQ)	7	*	3	21
4.Number of Internal Files(ILF)	1	*	7	7
5.Number of External Interfaces(EIF)	1	*	5	5
Total count				62

The Function Point(FP) is calculated as:

FP= Total count *[0.65+0.01+GSC count]

= 62*[0.65+(0.01*28)]

=57.66

External Inputs (EI):

- Password
- Product Details to update
- Name of product to search

External Outputs (EO):

- Inventory Table displayed
- Password checking and login
- Profit and loss margin
- Message when product is updated
- Retrieval of searched product details

External Inquiries (EQ):

- Enter password
- Search product (Asks for name)
- Display inventory table (displays the table)
- Back to previous page
- Add product (asks for details)
- Remove product (asks for name)
- Calculate profit and loss margin

Internal Files(ILF):

- File that stores password

External Interfaces(EIF):

- Excel file containing the inventory details

The Lines Of Code(LOC) is calculated as

$$\begin{aligned}\text{LOC} &= 57.66 * \text{The factor for the programming language used} \\ &= 57.66 * 67 \\ &= 3,863.22 \text{ lines}\end{aligned}$$

Basic COCOMO Model:

Kilo Lines Of Code is calculated as

$$\text{KLOC} = 3.863$$

Cost estimation is given by

$$E = a * (\text{KLOC})^b$$

For an Organic project, $a=2.4$ and $b=1.05$

$$\text{So } E = 10.4$$

Intermediate COCOMO Model:

List of Cost Drivers and their corresponding values:

The below mentioned list of cost drivers and their values hold good for our project. The extensive list of values for all other cases can be obtained from external sources.

	COST DRIVERS:	CORRESPONDING VALUE FOR OUR PROJECT
	PRODUCT ATTRIBUTES:	
1.	Required software reliability	1.00-Normal
2.	Size of application database	1.08-High
3.	Complexity of product	0.85-Low
	HARDWARE ATTRIBUTES	
4.	Runtime performance constraints	1.00-Normal
5.	Memory constraints	1.00-Normal
6.	Volatility of the virtual machine environment	1.00-Normal
7.	Required turnabout time	1.00-Normal

	PERSONNEL ATTRIBUTE	
8.	Analyst capability	1.00-Normal
9.	Application experience	0.91-High
10.	Software engineer capability	0.86-High
11.	Virtual Machine Experience	1.00-Normal
12.	Programming language experience	1.00-Normal
	PROJECT ATTRIBUTES	
13.	Application of software engineering methods	0.82-Very High
14.	Use of software tools	0.91-High
15.	Required development schedule	1.04-High

Our project is Organic, and hence we are going to determine the effort, development time, productivity based on the Intermediate COCOMO Model.

1. Effort, $E = a_i (\text{KLOC})^{b_i} \text{EAF}$, where a_i and b_i are constants whose values depend on the type of software project; and EAF is effort adjustment factor which is obtained by multiplying the values of the cost drivers.
2. Development time, $D = c_i (E)^{d_i}$
3. Productivity $= \text{KLOC}/E$
4. Staff Size $= E/D$

Calculation:

1. $E = a_i (\text{KLOC})^{b_i} \text{EAF}$.

$$a_i = 3.2$$

$$b_i = 1.05$$

$$\text{EAF} = 0.557$$

$$\text{KLOC} = 3.863$$

$$E = 3.2(3.863)^{1.05} * 0.557$$

$$= (4.133) * 3.2 * 0.557$$

$$= 7.3667 \text{pm}$$

Effort E= 7.3667 person months

2. $D = c_i (E)^{d_i}$

$$c_i = 2.5$$

$$d_i = 0.38$$

$$D = 2.5(7.3667)^{0.38}$$

$$= 2.5(2.135)$$

$$= 5.339 \text{ M}$$

Development Time=5.339 months

3. $P = \text{KLOC}/E$

$$\text{KLOC} = 3.863$$

$$E = 7.3667$$

$$P = 3.863/7.3667$$

$$=0.52$$

Productivity=0.52

4. $S=E/D$

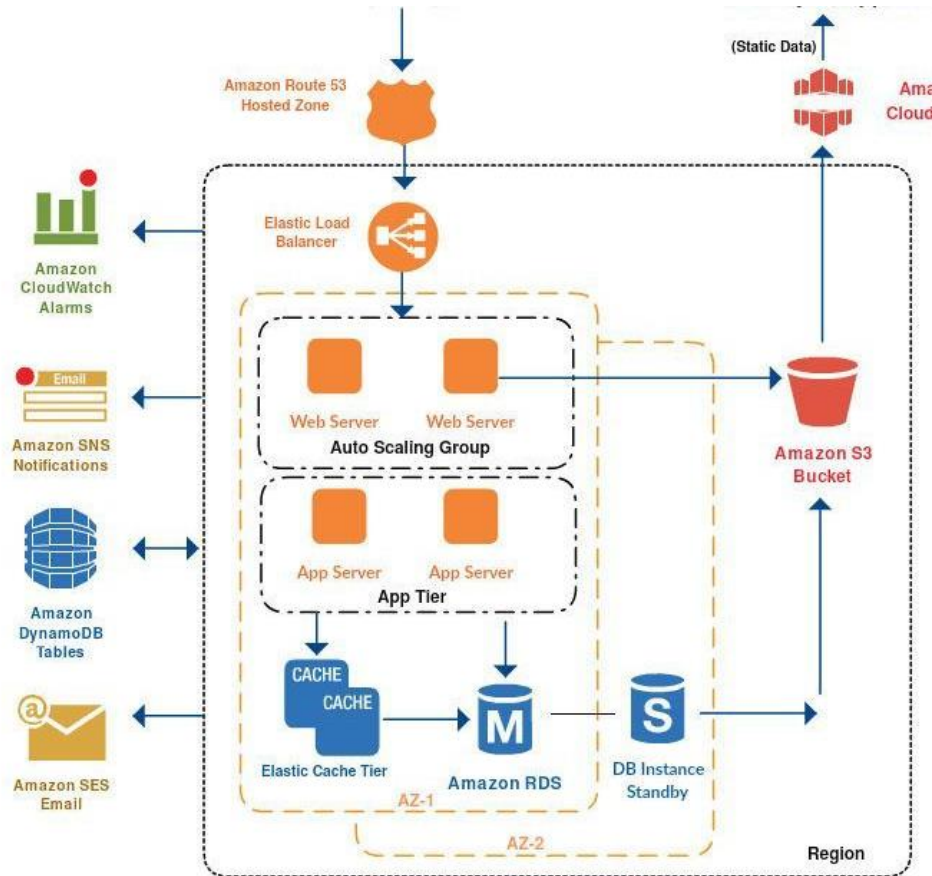
$$= 7.3667/5.339$$

$$=1.37$$

Staff Size=1.37

Ex 7: Design

System Architecture:



Ex 8: Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Identifying actors and use cases:

Actors:

- Inventory database
- Admin

Use cases:

- Login
- Display Inventory
- Add product
- Remove product
- Check product detail
- Profit-Loss %
- New Product's Data
- Product name

Association between actors and use cases:

- Login is used by Admin to access the Inventory database
- Display inventory is an option that uses data in the Inventory database to display the inventory table to the Admin
- Add product is an option that adds data to the Inventory database that is entered by the Admin
- Remove product removes data from the Inventory database on input from the Admin.
- Check product detail pulls the data necessary from the Inventory database and displays it to the Admin.
- Profit-Loss % calculates the profit and loss margin from data in the Inventory database and displays it to Admin.
- New Product's Data is a case included in Add product that takes input from the Admin for Add product.

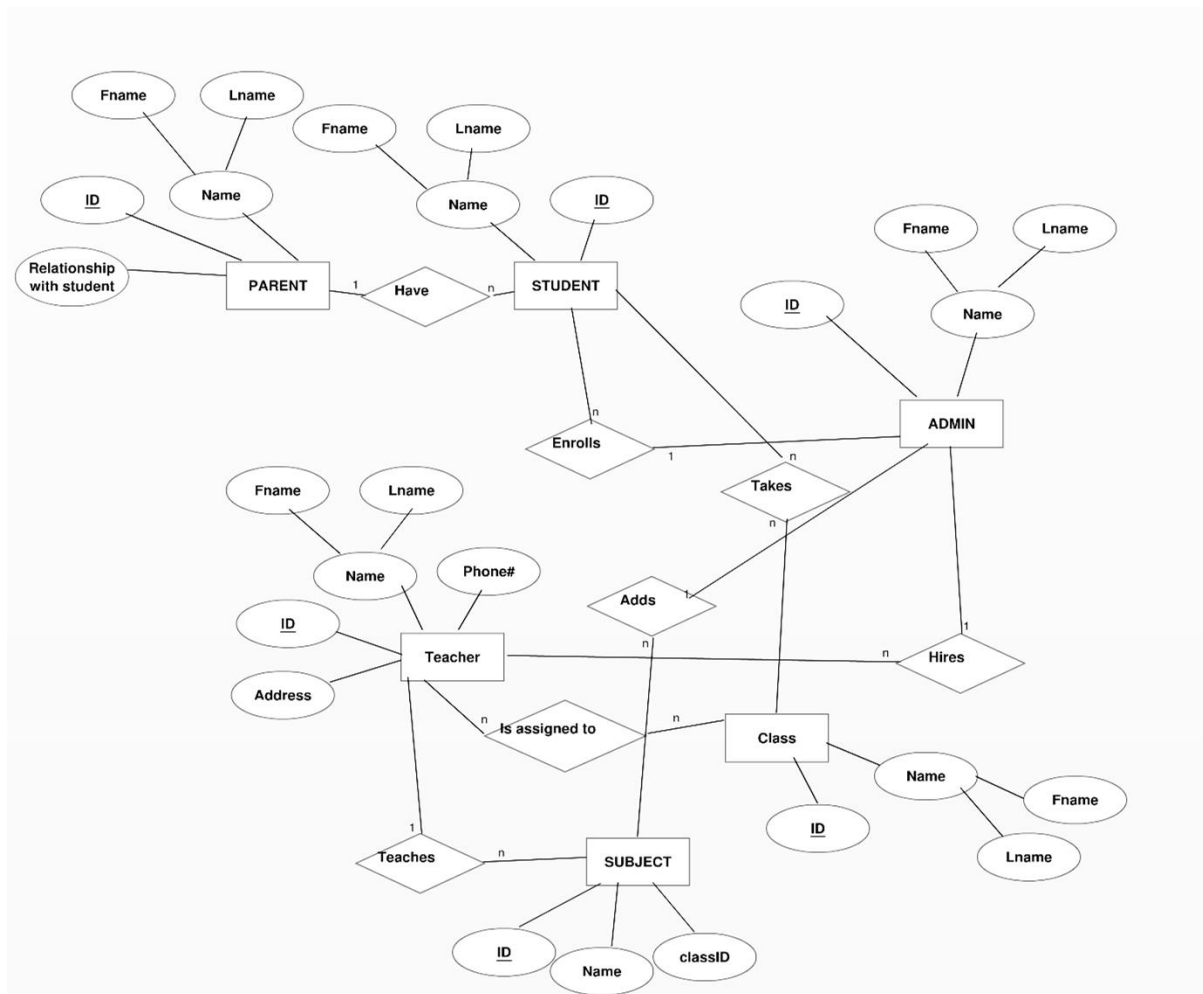
- Product name is a case included in Remove product and check product detail to receive input from the Admin.

Use Case relationships:

Active relationships are referred to as behavioural relationships and are used primarily in use case diagrams. There are four basic types of behavioural relationships: communicates, includes, extends, and generalizes.

- **COMMUNICATES.** The behavioural relationship communicates is used to connect an actor to a use case. So in our case, the actor is Admin. He communicates with the various use cases of the system like Login, Display inventory, Add product, remove product, check product and profit/loss %.
- **EXTENDS.** The extends relationship describes the situation in which one use case possesses the behaviour that allows the new use case to handle a variation or exception from the basic use case. In our case, apart from the usual feature of checking the status of the inventory, in case the admin wishes to add/remove products owing to the launch or discontinuation of a product, it can be done.
- **INCLUDES.** The includes relationship (also called uses relationship) describes the situation in which a use case contains behaviour that is common to more than one use case. For the inventory management system Product Name is a common behaviour for Display inventory, Add product, remove product, check product.
- **GENERALIZES.** The generalizes relationship implies that one thing is more typical than the other thing. This relationship may exist between two actors or two use cases. There is no generalizes relationship here.

Ex 9: E-R Modeling from the Problem Statements



Description:

- The ER Diagram represent the model of School Management System Entity.
- The entity-relationship diagram of School Management System shows all the visual instrument of database tables and the relations b/w Students, Teachers, Admins etc.
- It used structure data and to define the relationships b/w structured data groups of School Management System functionalities.

- The main entities of the School Management System are Parents, Teachers, Students, Admin, and Subject.

School Management System entities and their attributes:

- Parents: It can only access attributes i.e. Relationship with students, ID, Name,,Fname, Lname.
- Student: It can only access Fname, Lname, Name, ID.
- Admin: It can only access Fname, Lname, Name, ID.
- Teacher: It can only access Fname, Lname, Name, ID, Phone, Address.
- Class: It can only access Fname, Lname, Name, ID.
- Subject: It can only accessclassID, Name, ID.

School Management System entities and their relationships:

- Parents: It can only Student.
- Student: It can only access to take class, add subject and enroll to admin.
- Admin: It can only access to add subject and hire teacher.
- Teacher: It can only access to assign h.w to class and teach subject.
- Class: It can't access anything.
- Subject: It can't access anything.

Ex 10: Domain classes

In this diagram the domain classes are login and student database.

1.Login:

According to the type of entity of the method to access suitable database is provided.

It has two subclasses:

- User: The user has a unique ID and a password. It can only view the details and cannot modify them.
- Admin: The admin has its own admin password and can modify the database.

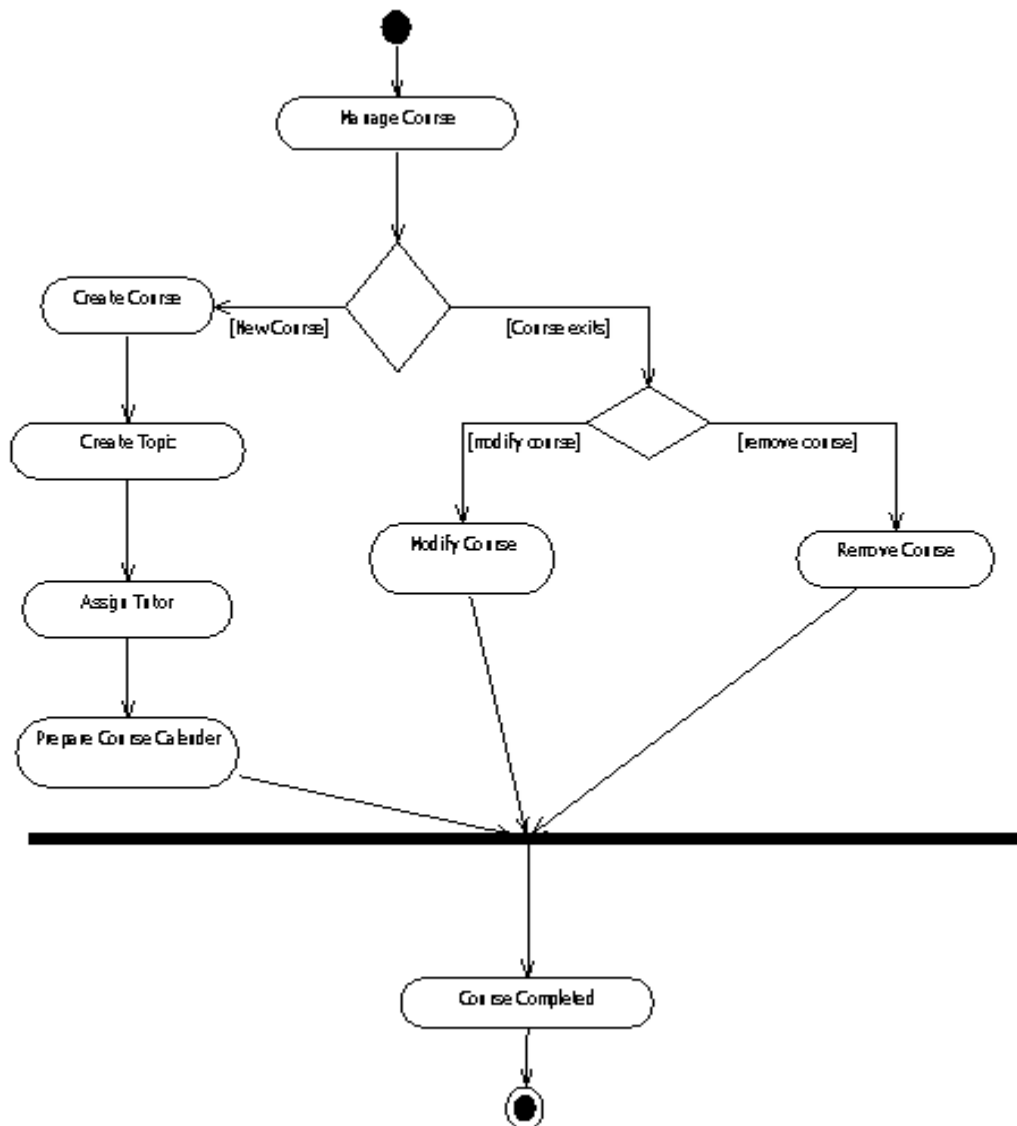
2.School Database:

It stores the details of the students, faculty, and their information.

These data can only be changed by the admin and the users can only read them. These data are updated time-to-time by the admin according to requirements and changes. The details modified by the admin can be retrieved to be displayed to both admin and user.

Ex 11: Modeling UML Class Diagrams and Sequence diagrams

State Chart Diagram:



State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems.

Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

States and Actions:

1. Start- The software begins.
2. Login- Password entered. If the entered credential is true, it proceeds to the next state. Else, it goes back to login.
3. There are 4 simultaneous states: Check, Add, Remove, Profit/Loss. Any one of it can be selected at a given point of time.
4. Check- It goes to a new page to perform next state.
 - 4.1 Enter product name- product name is entered. Now it goes to a new page to perform action of next state.
 - 4.2 Display product detail- the product detail for the searched product is displayed.
5. Add- goes to new page to perform action.

5.1Enter product Details- product details entered. Goes to next page for next state.

5.2Display added product- details of added product displayed.

6. Remove- goes to next page for next state.

7.

7.1Enter product name- product name is entered. Now it goes to a new page to perform action of next state.

7.2Display “Product Removed”- message displayed.

8. Profit/ Loss Margin- goes to next page for next state.

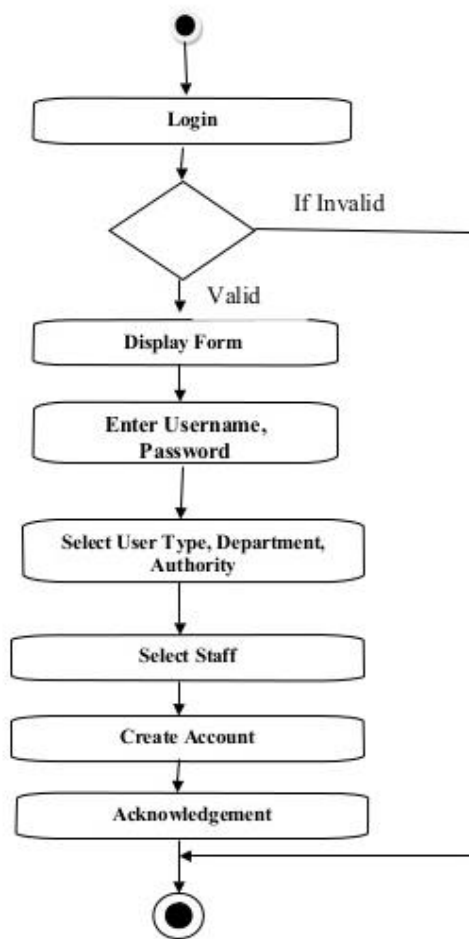
8.1Display margin- output displayed.

8.Stop.

Note:

- When the control goes to next page for the next state or to perform an action, it refers to a transition.
- The above mentioned 8 points indicate various states of the system.

Activity Diagram:



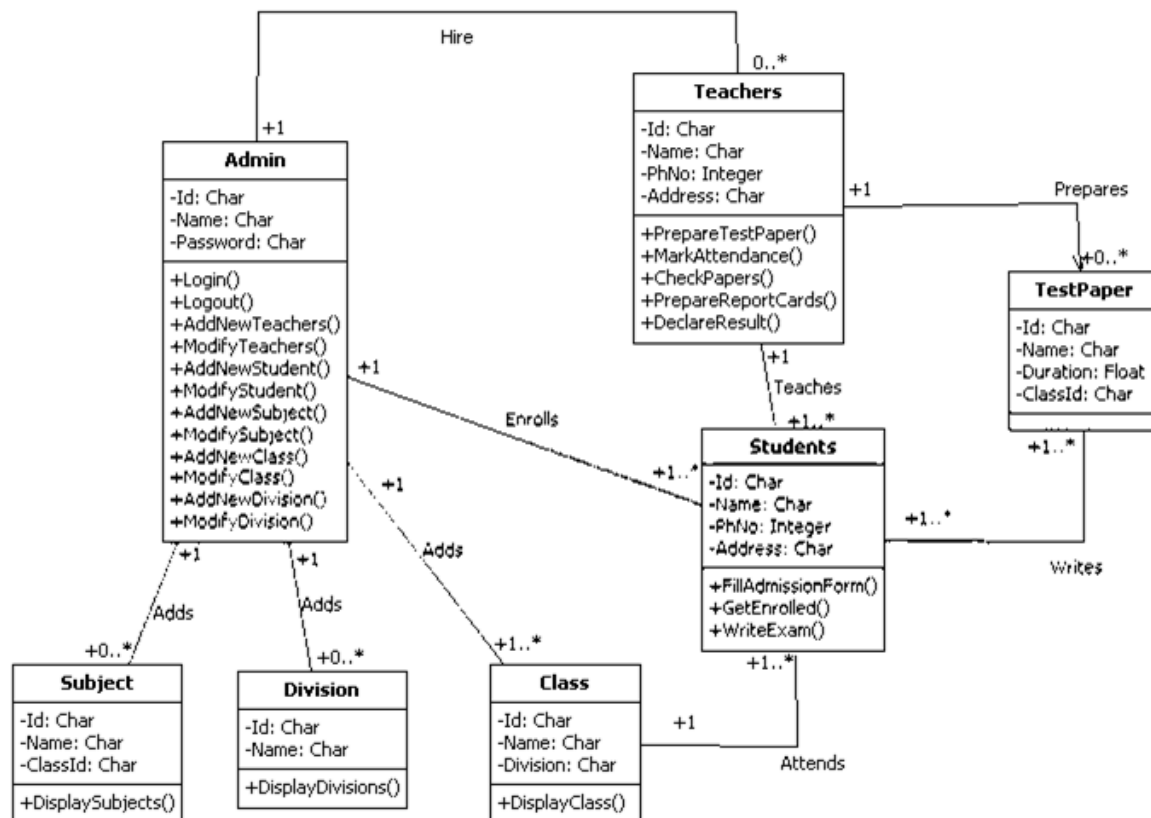
An **activity diagram** visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. **Activity diagrams** are often used in business process modeling. They can also describe the steps in a use case diagram.

Our project is an School Management System for managing the activities that take place in school from hiring a teacher to teaching a students and managing other financial and social activities that take place in school and depict the flow of activities ,data and control to various modules and actors.

There is a common login activity each for – admin, Teacher and Student, if the login is successful then the control is transferred to one of the five modules according to instruction given.

Ex 12: Modeling UML Class Diagrams and Sequence diagrams

Class Diagram:

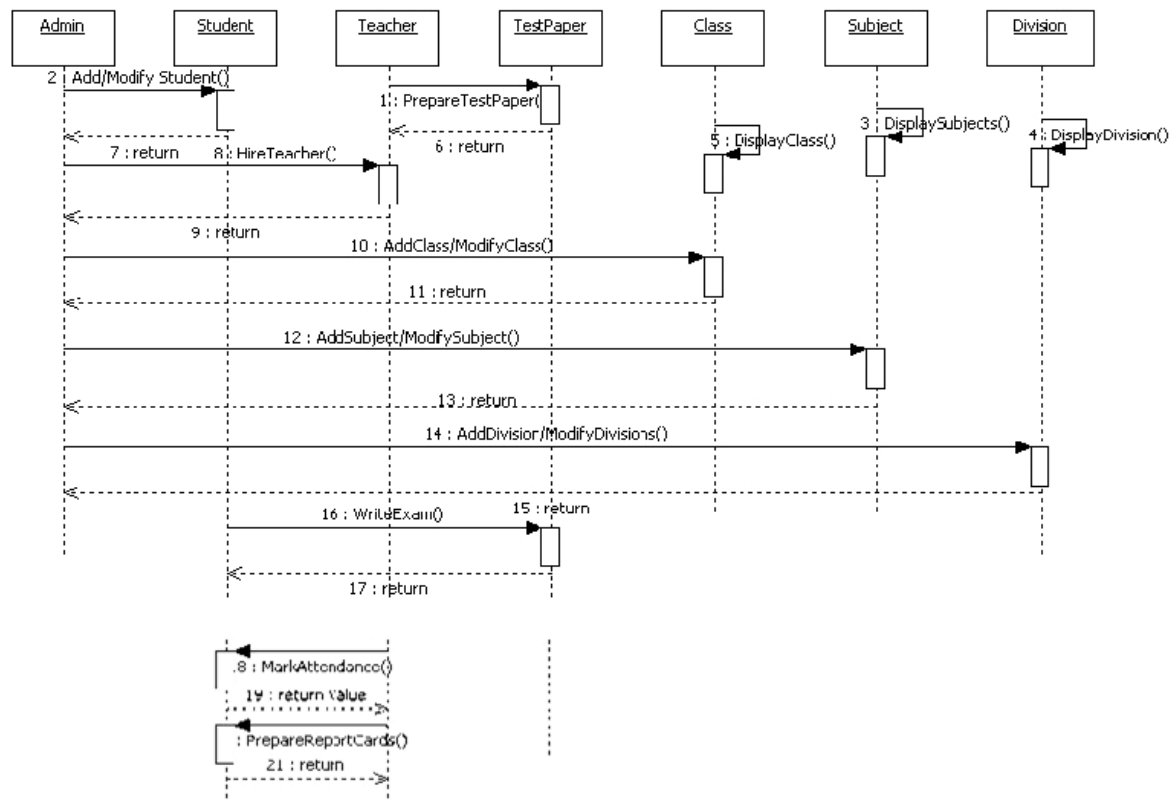


Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Our class diagram has one main class and six sub classes. The classes are:

Class Name	Data members	Member functions
School Database	-Students -Faculty members -subjects - classes -scores -curriculum -fees	+Display()
Login	-password	+verify_password()
Inventory table		+display_table()
Add Student	+student name +class +marks +fees	-add() +display_student_details()
Remove Student	+product name	-remove() +display_removed()
Details	+student name +faculty name	-fetch_details() +display_details()

Sequence diagram:



The sequence diagram consists of three dataflows: the admin/user's, the interface's and the inventory database's. The user has to login first using the password. The credentials are then verified. If the credentials are wrong the user is asked to enter again.

1.Display details: The interface gets data from the database, and displays it in a table format.

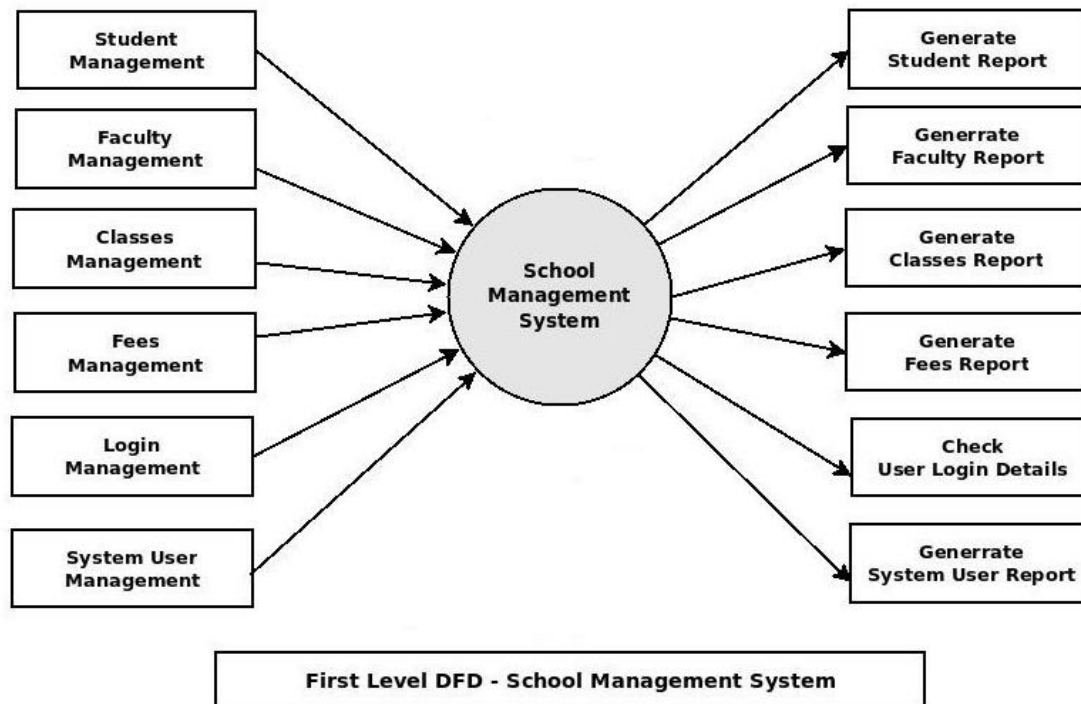
2.Add Student: The interface gets the product details from the admin and adds the entry to the database.

3.Remove Student: The interface gets the product details from the admin and removes the entry to the inventory database.

4.Check student details: The interface gets the student name from the admin/user, gets its details and displays it to the user.

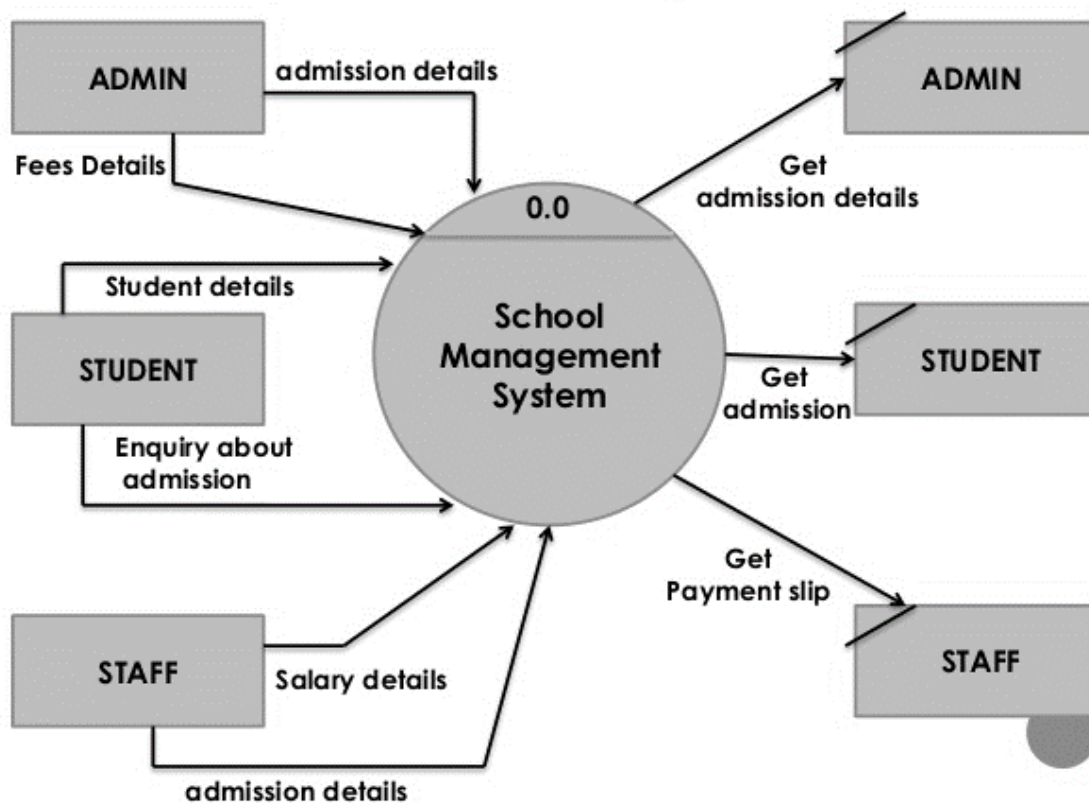
Ex 13: Modelling data flow diagrams

Data Flow Diagram:



- This is the First Level DFD of School Management System, where we have elaborated the high level process of School.
- It is a basic overview of the whole School Management System or process being analyzed or modeled.
- It is designed to be an at-a-glance view of Fees, Attendance and Subjects showing the system as a single high-level process, with its relationship to external entities of Students, Class and Teacher.
- It should be easily understood by a wide audience, including Students, Teachers and Fees. In first level DFD of School Management System, we have described the high level flow of the School system.

Context diagram:



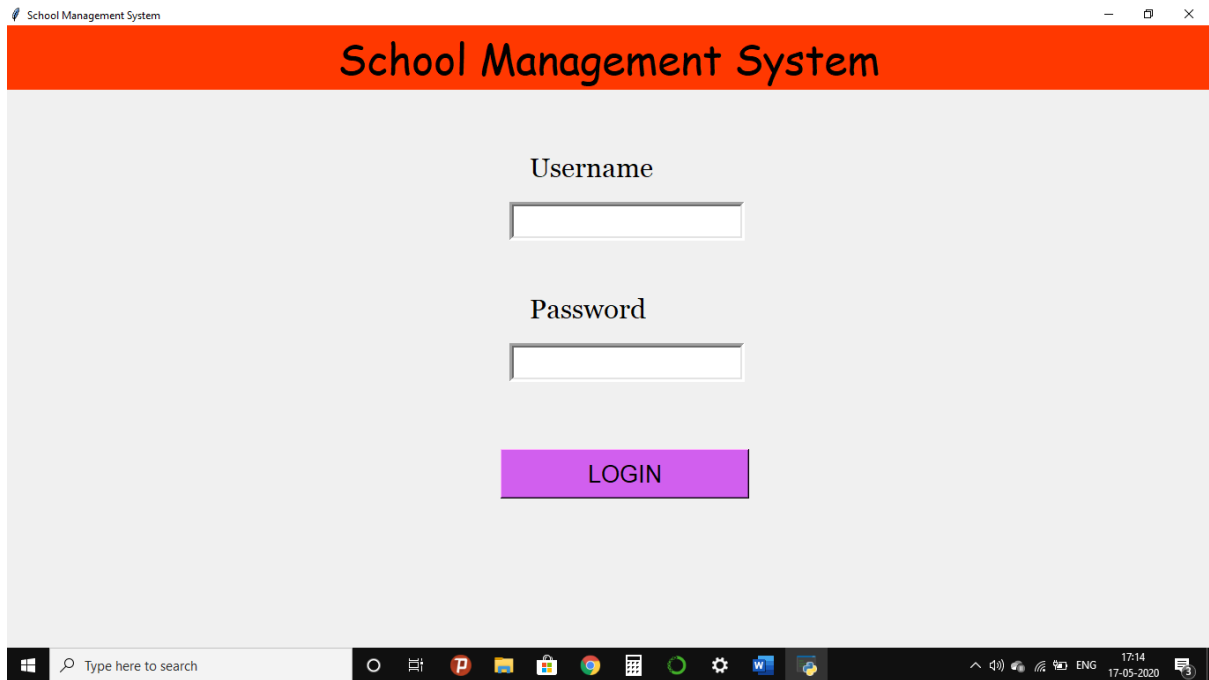
The context of the system involves entities accessing the Inventory database:

- Admin: The admin can view admission and fee details by requesting them and the database supplies the details
- Student: The student can modify and view student and admission details and the database makes the changes in the details and displays the details for viewing.
- Staff: The staff can ask about their salary and admission details and management return payment slip

EXP 14 : IMPLEMENTATION

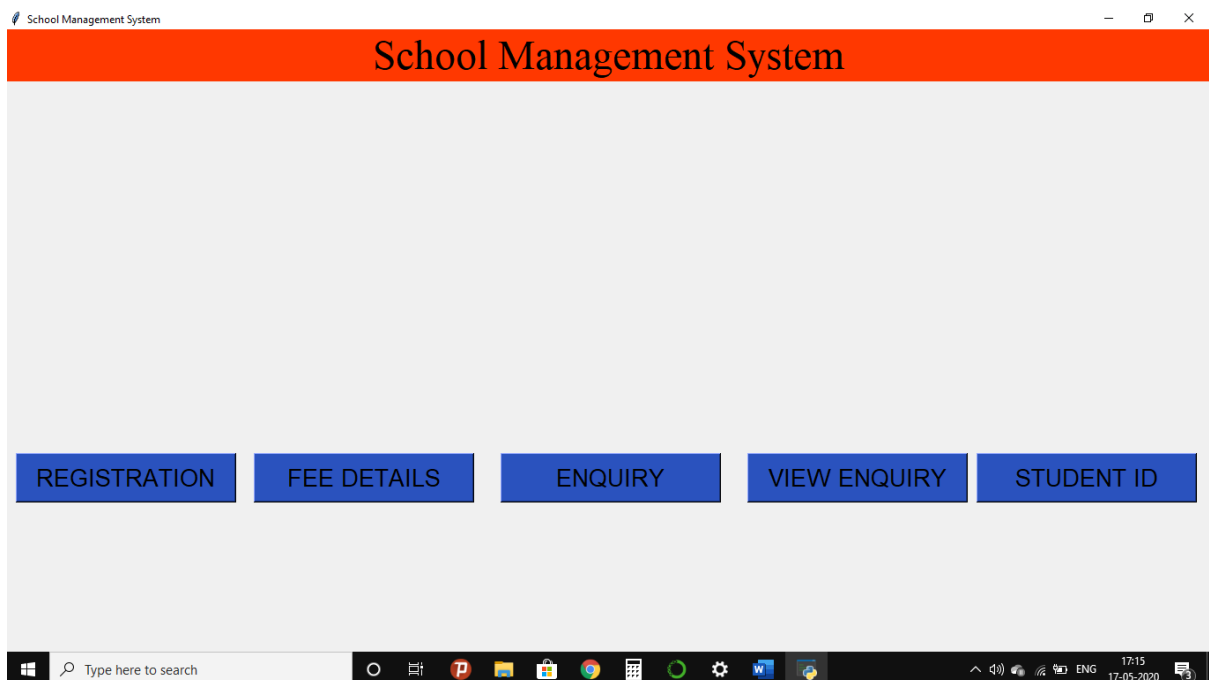
Screenshots:

1. Login Page



The screenshot shows a web browser window titled "School Management System". The page has a red header bar with the text "School Management System" in black. Below the header, the page is light gray. In the center, there are two input fields: "Username" and "Password". Below these fields is a purple button labeled "LOGIN". The Windows taskbar is visible at the bottom, showing the search bar and various application icons. The system clock in the bottom right corner indicates the time is 17:14 on 17-05-2020.

2. Main Menu



The screenshot shows a web browser window titled "School Management System". The page has a red header bar with the text "School Management System" in black. Below the header, the page is light gray. At the bottom of the page, there are five blue buttons arranged horizontally: "REGISTRATION", "FEE DETAILS", "ENQUIRY", "VIEW ENQUIRY", and "STUDENT ID". The Windows taskbar is visible at the bottom, showing the search bar and various application icons. The system clock in the bottom right corner indicates the time is 17:15 on 17-05-2020.

3. Registration Form

School Management System

BACK

REGISTRATION FORM

Restore Down

NAME:

SURNAME:

EMAIL:

GENDER: ☐ MALE ☐ FEMALE

BRANCH:

COURSE: ☐ JAVA ☐ C ☐ C++ ☐ PYTHON

TOTAL FEE:

Type here to search

17:15 17-05-2020

4. Fee Details

engiry

BACK

RESET

ENTER STUDENT ID

REMAINING FEE PAID FEE TOTEL FEE

Type here to search

17:16 17-05-2020

5. Enquiry

The screenshot shows a web browser window with the title "ENQUIRY". The page has a pink header bar with the word "ENQUIRY" in white. Below the header, there is a form with three input fields: "NAME:", "PHONE NO.:", and "PURPOSE:". The "PURPOSE:" field is a dropdown menu. A pink "submit" button is located below the form. In the top left corner of the browser window, there is a small icon and the text "ENQUIRY". In the top right corner, there are window control buttons (minimize, maximize, close). The Windows taskbar is visible at the bottom of the screen, showing the search bar and various application icons. The system clock in the bottom right corner shows the time as 17:17 and the date as 17-05-2020.

ENQUIRY

NAME:

PHONE NO.:

PURPOSE:

submit

6. Visitor Screen

The screenshot shows a web browser window with the title "VISITOR". The page has a purple header bar with the word "VISITOR" in white. Below the header, there is a table with four columns: "NAME", "EMAIL", "ENQUIRY", and "DATE". The table is currently empty. In the top left corner of the browser window, there is a small icon and the text "VISITORS". In the top right corner, there are window control buttons (minimize, maximize, close). The Windows taskbar is visible at the bottom of the screen, showing the search bar and various application icons. The system clock in the bottom right corner shows the time as 17:19 and the date as 17-05-2020.

VISITOR

NAME	EMAIL	ENQUIRY	DATE
------	-------	---------	------

7. Student Database

VISITORS

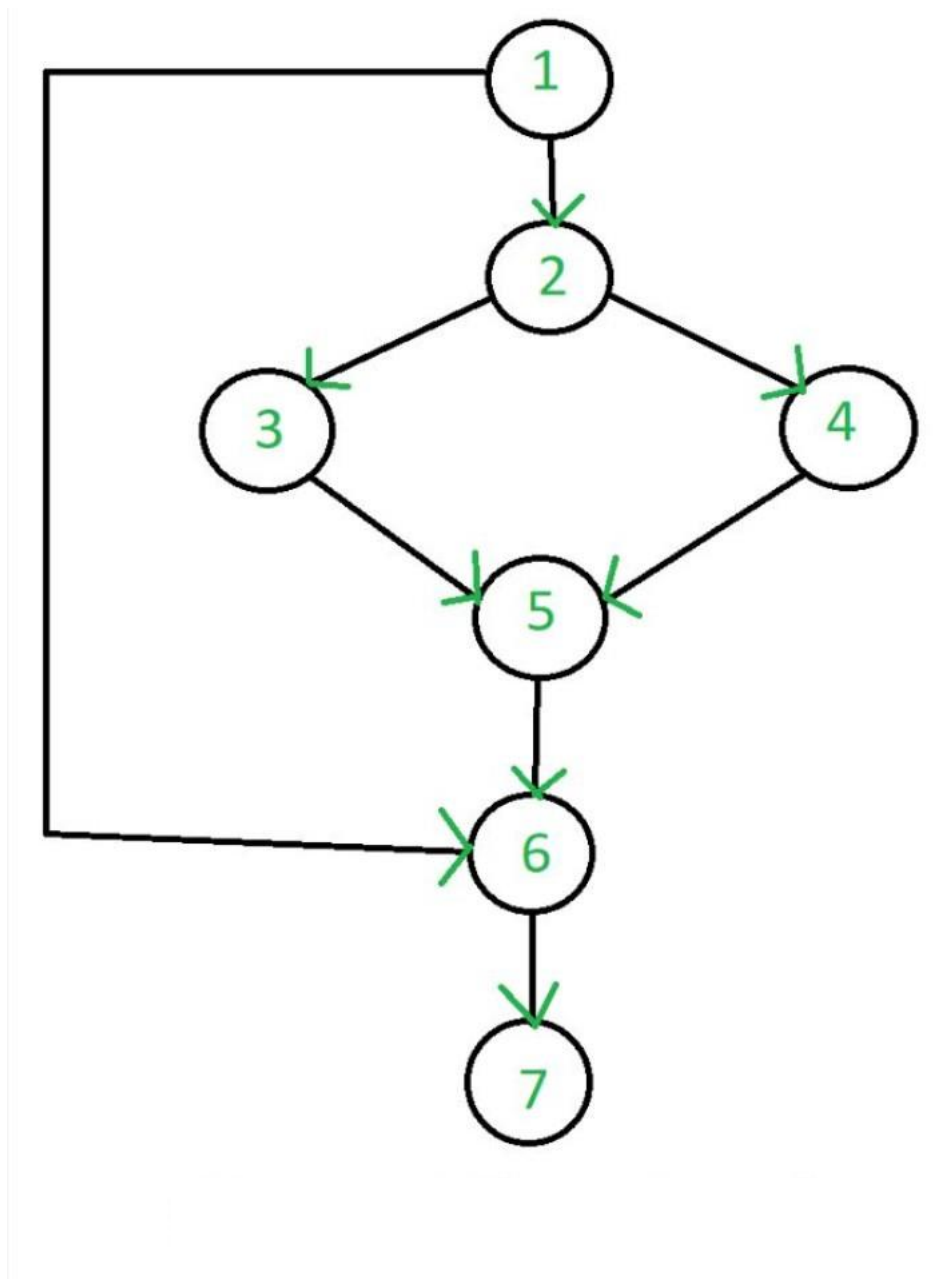
ENQUIRY DETAILS

ID	NAME	FEE	LAST NAME
----	------	-----	-----------

Type here to search

17:21
17-05-2020

EXP 15: CONTROL FLOW GRAPGH



MCCABE'S CYCLOMATIC COMPLEXITY:

E = number of
edges in the flow
graph. N =
number of nodes
in the flow graph.

P = number of nodes that have exit points

EXMAPLE

If =10 then

If B>C then

A=B

Else

Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

Cyclomatic complexity= $E - N + 2 * P$

Where,

A=C

End if

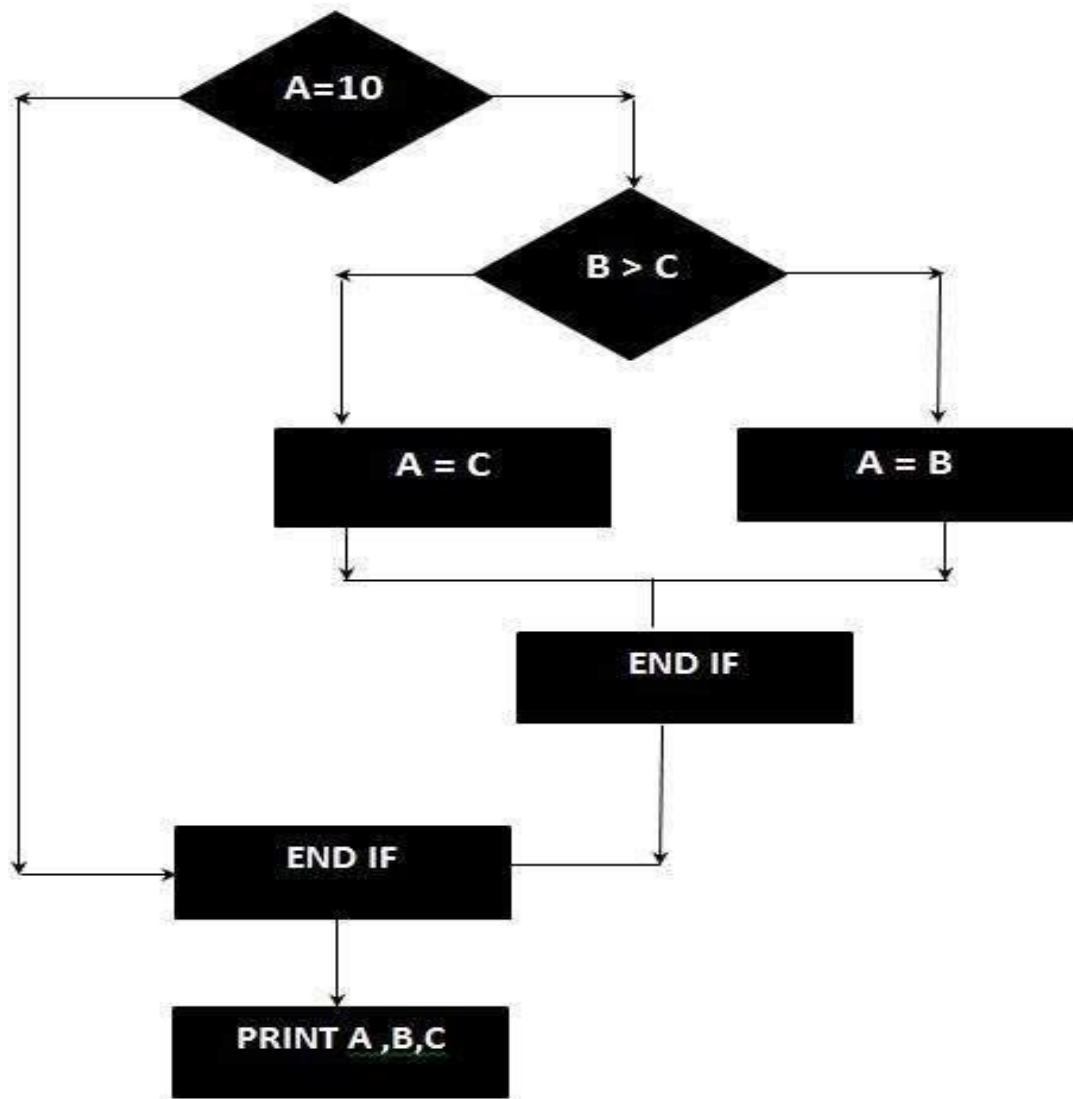
End if

Print A

Print B

Print C

Flow Graph



The Cyclomatic complexity is calculated using the above control flow diagram that shows seven nodes(shapes) and eight edges (lines), hence the cyclomatic complexity is $8 - 7 + 2 = 3$

OPTIMUM VALUE OF CYCLOMATIC COMPLEXITY

An application consisting of zero decision points (IF, FOR, etc.) has an intricacy score of 1 because it contains a single path in the source code. If the program contains an IF statement consisting of one condition, the code would contain a total of two paths: TRUE or FALSE. The cyclomatic complexity algorithm is used to derive a measurable value based on the number of edges and nodes within the graph as well as the total count of connected components or exit nodes. It is represented as shown below:

$$\text{Complexity (M)} = \text{Edges (E)} - \text{Nodes (N)} + \text{Exit Nodes (P)}$$

This software measurement aids in limiting routine complexity during the development process and makes it easier to split modules into smaller, easier-to-manage components.

Programs with a level below 10 are considered to be within the cyclomatic complexity acceptable range. This measurement can be used to identify areas of improvement at the source-code level. Risk is assessed based on the provided value:

- 01 to 10 – MinimalRisk
- 11 to 20 – ModerateRisk
- 21 to 50 – HighRisk
- Over 50 – Very HighRisk

Ex 16: Designing Test Suites

Software Testing:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

Testing Frameworks:

A testing framework or more specifically a testing automation framework is an execution environment for automated tests. It is the overall system in which the tests will be automated. It is defined as the set of assumptions, concepts, and practices that constitute a work platform or support for automated testing.

The Testing framework is responsible for:

- Defining the format in which to express expectations.
- Creating a mechanism to hook into or drive the application under test.

- Executing the tests
- Reporting results

Properties of a testing framework:

- It is application independent.
- It is easy to expand, maintain and perpetuate.

Master Test Plan

The master test plan is a document that describes in detail how the testing is being planned and how it will be managed across different test levels. It gives a bird's eye view of the key decisions taken, the strategies to be implemented and the testing efforts involved in the project.

Details provided by Master Test Plan:

In simple terms, the master test plan for software testing provides the following details:

- List of tests to be performed
- Testing levels to be covered
- Relationship among different test levels and related coding activity
- Test implementation strategy
- Explain testing effort which is a component of the project
- Master test plan should align with test policy and test strategy. It should list any exceptions or deviations and their possible impact.

Manual Testing:

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user whereby they use most of the application's features to ensure correct behavior. To guarantee completeness of testing, the tester often follows a written test plan that leads them through a set of important test cases.

Stages:

Unit Testing

This initial stage in testing normally carried out by the developer who wrote the code and sometimes by a peer using the white box testing technique.

Integration Testing

This stage is carried out in two modes, as a complete package or as an increment to the earlier package. Most of the time black box testing technique is used. However, sometimes a combination of Black and White box testing is also used in this stage.

System Testing

In this stage the software is tested from all possible dimensions for all intended purposes and platforms. In this stage Black box testing technique is normally used.

User Acceptance Testing

This testing stage carried out in order to get customer sign-off of finished product. A 'pass' in this stage also ensures that the customer has accepted the software and is ready for their use.

Release or Deployment Testing

Onsite team will go to customer site to install the system in customer configured environment and will check for the following points:

1. Whether SetUp.exe is running or not.
2. There are easy screens during installation

EXP 17:DEPLOYMENT REPORT

Configuration Document:

Step 1 Configuring python:

- 1) Download python 3 from python.org.
- 2) Install python using the installer.
- 3) Install any additional packages using pip in command prompt.

Step 2 Getting the system:

- 1) Go to GitHub , go to the project page .
- 2) Click clone or download.
- 3) Install requirements from requirements.txt
- 4) Download it as a zip and extract it.

Step 3 Running the system:

1)Run main.py from the extracted files using python.
From here you should be able to perform all the functions mentioned.

Version 0.1:

- Introduced the login screen.

Version 0.2:

- Made modifications to the login menu.
- Added an Encryption system for added security.
- Ui/Ux changes.

Version 0.3:

- Added admin menu.
- Bug fixes.

Version 0.4:

- Bug fixes.

Version 0.5:

- Added more functionality for pre existing systems.
- Added more menus.
- Bug fixes.

Version 0.6:

- Added complaint or suggestion systems to all menus.

Version 0.7(Major update):

- Added attendance and backup system.

Version 1.0(Major update):

- Migrated to python due to various reasons such as better compatibility with sqlite3.

Version 1.1(Minor update):

- Bug fixes

Version 1.2:

- Ui/Ux changes.
- Added User menu.
- Finished rate handler.

Version 1.3:

- Added more options to pre existing menus.

Version 2.0(Major update):

- Completed all the menus with additional options.
- Bug fixes and patches.
- Completion of the software with full-fledged functionalities.

Exp 18: Conclusion

- The School Management system is developed to help schools, colleges, educational institutes and home based tuition centers to maintain the record of their students and faculty members.
- The development of this software involved the creation of various modules, taking into consideration the day to day activities of the institution who utilize this software.
- This system is completely offline which is ease for village areas.
- Privileges have been made for multiple users without compromising the security of the system.
- The software takes care of all the requirements of the user, and is capable of providing easy and effective storage of information regarding the products sold
- The system is designed to be friendly, easy-to-use, that people can start using right away without much trouble or confusion.

Thus the School Management System has been built and its output is suits to the needs of the Educational Institutes.

EXP 19: REFERENCES

- www.stackoverflow.com
- www.tutorialspoint.com
- www.geeksforgeeks.com
- <https://www.uml-diagrams.org/software-licensing-domain-diagram-example.html>
- <https://www.uml-diagrams.org/software-licensing-domain-diagram-example.html>
- <https://www.softwaretestinghelp.com/software-test-metrics-and-measurements/>
- https://en.wikipedia.org/wiki/Test_suite
- Python for Everybody by Charles Severence
- <https://www.geeksforgeeks.org/python-reading-excel-file-using-openpyxl-module/>
- <https://www.smartdraw.com/entity-relationship-diagram/>