ARCADE BUSINESS COLLEGE

Arya Kumar Road, Rajendra Nagar, Patna – 800016

SYNOPSIS

(Paper : Hons. VIII)

Session: 2021 - 2024

for

SCHOOL FEE MANAGEMENT SYSTEM

....

Group ID : PRJ2333E..............................

Submitted By :

Coll. Roll No.	Registration No.	Name	Signature
21068	202143900321	Subal Anand	
21094	202143900334	Vikarn Kumar Jha	
21112	202143900309	Satyam Kumar	
21089	202143900332	Utkarsh Kumar	

For Office use only

Approved □ **Not Approved** □

Remarks:

Signature

SCHOOL FEE MANAGEMENT SYSTEM

PRAGAYA HERITAGE INTERNATIONAL SCHOOL

(BYPASS ROAD, BARGAMA, GULABBAGH, PURNEA, BIHAR)

SCHOOL FEE MANAGEMENT SYSTEM

A project report submitted in partial

Fulfilment of the requirements

For the awards of degree

BACHELOR OF COMPUTER APPLICATION

By

Subal Anand	202143900321	21068
Vikarn Kumar Jha	202143900334	21094
Satyam Kumar	202143900309	21112
Utkarsh Kumar	202143900332	21089

PROJECT ID: PRJ2333E

Under the supervision of

Mrs. Anupam Singh

(HOD)

ARCADE BUSSINESS COLLEGE

Arya Kumar Road, Rajendra Nagar, Patna

Patliputra University, Patna

Session- 2021-2024 Date: ……………….

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher **Mrs. Anupam Singh** who gave us golden opportunity to do this wonderful project on “School Fee Management System”.

We are very grateful to our project guide **Mrs. Anupam Singh** and our respected teachers of lab who had always guided in making of this project.

We also thank Pragya Heritage International School during our System Study and Analysis stage they had been particularly helpful in providing the necessary data about the manuals school fee management system.

This Project would not have been completed without their enormous help and worthy experience whenever we were in need, they were behind us.

Secondly, we would also like to thanks our parents and friends who helped us in finalizing this project within the limited time frame.

Last but not least, we would like to convey our gratitude to the **Mrs. Anupam Singh & Md. Khurshid Aalam** in our Faculty for the good guidance during the class period of our course Coordinator and All our course faculty members.

May the good lord bless them and keep them safe. We love you all.

Thank You

DECLARATION

We hereby declare that the Project entitled- "School Fee Management System", which is being submitted as project of 6th semester in Bachelor of Computer Application to Arcade Business Collage, Patna is an authentic record of our genuine work done under the guidance of **Mrs. Anupam Singh**, Head of Department of BCA, Arcade Business Collage, Patna.

Project ID: ………………..

Date: ………………..

Supervised by:

Signature: ……………………

Anupam Singh

Senior Lecture & Head

of BCA Department,

Arcade Business College

Co-supervised by:

Signature:………………

Md. Khurshid Aalam

Lab Faculty

Arcade Business College

Submitted By:

Signature: …………………

Name: **Subal Anand**(Team Leader)

Reg.no: 202143900321

Signature: …………………

Name: **Vikarn Kumar Jha**

Reg.no: 202143900334

Signature: ………………...

Name: **Satyam Kumar**

Reg.no 202143900309

Signature: ………………..

Name: **Utkarsh Kumar**

Reg.no: 202143900332

TABLE OF CONTENT

INTRODUCTION

It started with its first school in the year 2017 and till now. We offer programmes right from Kindergarten to Class X. Our Honourable Founder Director **Late Mr. Ashok Tiwari** are great visionaries who believe that if you want to serve the nation then educate the nation and its future generation.

PRAGYA HERITAGE INTERNATIONAL SCHOOL, Bypass Road, Basgama, Gulabbagh, Purnea (Bihar) was founded in the year 2017 and ever since then we have never looked back. It is a co-educational, English medium, CBSE affiliated, day boarding school.

It will be school responsibility to expose students to the rich cultural heritage of India to the enrichment of her freedom struggle and the depth and beauty of her history, language and literature Futuristic with firm linkage with the present. And to revive the lost glory of Purnea, as seat of learning.

Owner Name: Aniket Tiwari

Working Function:-

A school fee system is a set of documented processes and guidelines that outline how school's fee collection and management procedures are carried out without the use of specialized software or a automated system Here's an overview of the working function of manual "School Fee Management System":

- **Fee Structure**: Define the fee structure, including different types of fees (e.g., tuition, computer, extracurricular activities) and the amount for each. This structure should be clearly communicated to parents and guardians.

- **Student Information**: Maintain up-to-date student records, including their names, classes, and fee-related information.

- **Fee Calculation:** Manually calculate fees for each student based on the fee structure.

- **Fee Payment**: Students or their parents/guardians can make payments through cash. Provide a clear process for making payments and issue receipts for each payment.

- **Receipt Generation**: Prepare and issue receipts for each payment made, indicating the date, amount, and purpose of the payment.

- **Fine Calculation**: Manually calculate and apply fine for overdue payments, adhering to the school's policy for fine payment penalties.

- **Record Keeping**: Keep detailed records of all free-related transactions, including receipts, payment logs, and any correspondence related to fee issues.

LIMITATIONS OF EXISTING SCHOOL FEE SYSTEM

Existing System:

At present parents maintain their month to month transaction where a user has to open register and takes the cash and maintain it in the register. During this process it takes lots of time and parents must wait till the process. Only For one payment school faces a problems with a unique challenges.

The School doesn't have any software applications which maintains data of the students. The record keeping is done manually. This leads to great difficulty when certain needs arrive of looking into student's past data.

Limitations of the Existing System:

By analysing the existing system, some of its drawbacks are listed :

- **Lots of paperwork:-** In an existing school fees system requires lots of paperwork and even a small transaction requires paperwork. Even any unnatural cause(such as fire in the school) can destroy all the data of the school fees. Loss of even a single paper laid to difficult situation because all the papers are interrelated.
- **Need more working staffs:-** The work that can be easily done by system are to be done manually, that's the disadvantage of existing system.
- **Acquire more space:-** Keeping Files, registers possess more space which can be replaced by using storage(machines). And due to lake proper arrangement we need more files to storage data
- **Searching slow:-** Since, Data of any students are not in arranged way, searching of info is slow in existing system.
- **Manual Control:-** It can cause error and human mistakes in document then loss of data will happens, it will impossible to recover it.
- **More security concern:-** This system lacks very badly in security areas and there is no safety assurance of secret information. Since everything is in human hands access of data by an unauthorized person may be possible this is one of the most harmful drawbacks.

OBJECTIVE OF SCHOOL FEE MANAGEMENT SYSTEM

The main objective is to define a school fee management system to make it user friendly and easily do all the payment very easily. The system is totally computerized and has been developed using advance language therefore it gives more facilities than present system. It provide quick access to any data. In this system user have to enter data only once and then it get linked with all files. This reduce the workload of user and it is also time saving process.

This system will easy to understand and use. More specifically this system is designed to allow an admin to manage the records of students, fee structure, payments, fine and other management in School Fee.

This system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work. The proposed system tries to eliminate or reduce these difficulties up to some extent. Through this system it is possible to print various reports and bills for the organization.

Main Objective:

- **Easy to maintain storage:-** Due to computer work it is very easy to maintain the storage.
- **No need more employees:-** In our software due to computer work we no need to more employee to performs the school fees task because a computer can do a very large work in a very short time that increases the company expenses.
- **Searching:-** In our software the searching of any information can be fast as compared to manual system .
- **Data security guaranteed:-** As we know that data is backbone of any school so every school wants that his data is must be secured. In us proposed system due to computer approach the data cannot be harm by physically.
- **Easy reports generating:-** In the computer report can be easily generated and time to generate a report become less as compared to manual system.

H/W AND S/W REQUIREMENTS

The hardware and software requirements for developing a project using visual basic and oracle:

Hardware Requirements:

Computer: A modern computer with sufficient processing power, memory, and storage capacity to comfortably run both Visual Basic and Oracle Database.

Processor: A multi-core processor such as Pentium 90 MHZ or Higher is recommended for a smooth development experience.

RAM: A minimum of 256 MB of Ram is recommended for development, but more RAM may be necessary for large projects or complex database operations.

Storage: A minimum of 100 MB or free space hard disk capacity for visual basic , Oracle database.

Software Requirements:

Visual Basic(Front End): We have to need a Microsoft visual studio that supports a visual basic 6.0 edition. Visual Studio is a comprehensive integrated development environment (IDE) that provides the necessary tools for creating VB applications.

Oracle Database(Back End): We have to need a Oracle 10g Database Edition for storing and managing data.

Operating System: We need to require Windows XP or other OS for project development using visual basic and oracle.

INTRODUCTION TO VB 6.0

Visual Basic 6.0, often abbreviated as VB6, was a popular programming language and integrated development environment(IDE) released by Microsoft in 1996. It was widely used for developing Windows applications and was especially popular in the late 1990s and early 2000s Here's an introduction to Visual Basic 6.0 for "School Fee Management System":

Overview of Visual Basic 6.0:

Visual Basic 6.0 is a programming language and IDE that enables developers to create Windows applications with a graphical user interface(GUI). It's part of the Visual Basic family and is known for its simplicity and ease of use, making it a great choice for beginners and small to medium-sized projects.

Key Features of Visual Basic 6.0 for this project:

- **Visual Design:** Visual Basic 6.0 offers a drag-and-drop interface for designing the user interface, making it easy to create forms, buttons, labels and other GUI elements visually.
- **Event-Driven-Programming**: VB6 is event-driven which means that you can define what happens when the user interacts with the program's elements(e.g.. Clicking a button)
- **Database integration**: VB allows you to connect to databases, which is crucial for managing student records, fee payments, and generating reports.
- **Data Binding:** We can use data binding to connect your GUI elements directly to data source(e.g. a database) simplifying the management of data.
- **Report Generation**:VB6 provides tools for creating and printing reports, which can be useful for generating fee receipts and financial reports.
- **Error Handling:** VB6 includes error handling mechanisms to handle and reports errors in our application gracefully.

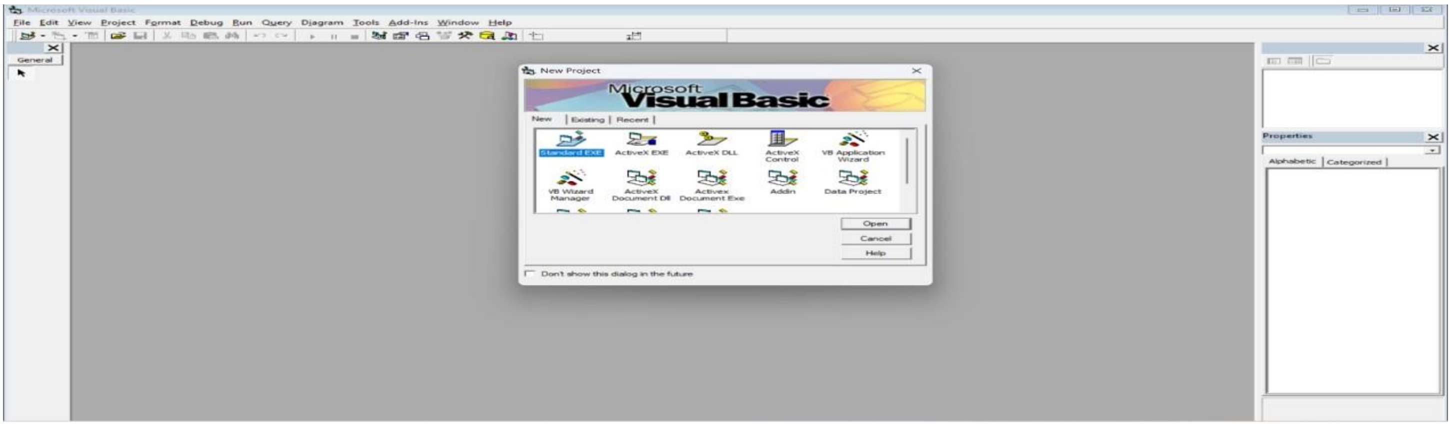
Steps for Developing the Projects:

- Define the project requirements and objectives, including the specific features and functionalities needed.
- Design the user interface for the application using the Visual Basic 6.0 IDE.
- Create a Database structure to store student and fee related data.

- Write the code to implement the desired features, including the fee-collection, receipt generation and financial reporting.
- Test the application thoroughly to ensure it functions as expected.
- Debug and refine the code as needed.
- Deploy the application to the target the computers within the school

Reasons for Choosing Visual Basic 6.0 Front End:

- **Legacy Systems**: If our project requires maintenance or integration with existing systems developed in VB6, using it as the front end can be practical.
- **Familiarity**: If our development team has expertise in VB6 and limited experience with other front-end technologies, it might be easier and costeffective to leverage their existing skills.
- **Rapid Development**: VB6 was known for its rapid application development (RAD) capabilities, allowing developers to create Windows applications quickly.
- **Desktop Applications**: VB6 is suitable for building Windows desktop applications with a graphical user interface.
- **Compatibility:** VB6 applications can run on older Windows operating systems, which might be necessary for compatibility with specific environments.
- **Simplicity:** VB6's drag-and-drop GUI design and straightforward syntax made it accessible for developers with varying levels of experience



INTODUCTION TO ORACLE 10G

Oracle 10g, often referred to as Oracle Database 10g, is a robust and advanced relational database management system(RDBMS) released by Oracle Corporation it provides a powerful platform for storing, managing and retrieving data. Oracle database commonly used in various industries, including education. Here's an introduction to using Oracle 10g for a "School Fee Management System" project.

Oracle 10g Overview:

Relational Database Management System(RDMS): Oracle 10g is a sophisticated RDBMS that uses a relational data model to organize data into tables, making it a suitable choice for managing structured data efficiently.

Key Features of Oracle 10g for this project:

- **Data Security**: Oracle 10g provides robust security features, including user authentication, authorization, and encryption, ensuring the confidentiality and integrity of sensitive fee-related data.
- **Scalability**: Oracle 10g can handle large datasets and is highly scalable, making it suitable for educational institutions of all sizes.
- **Data Integrity**: Oracle 10g enforces data integrity through constraints and referential integrity, preventing data corruption and ensuring data accuracy.
- **Concurrency Control**: Oracle 10g supports multi-user environments and provides mechanisms to manage concurrent access to data, preventing conflicts and data inconsistencies.
- **SQL and PL/SQL**: It supports the SQL query language for data retrieval and manipulation and PL/SQL for creating custom procedures and functions.
- **Reports and Analytics**: Oracle 10g supports reporting tools and data analytics to help in generating financial reports and analysing fee-related data.

Steps for Developing this project:

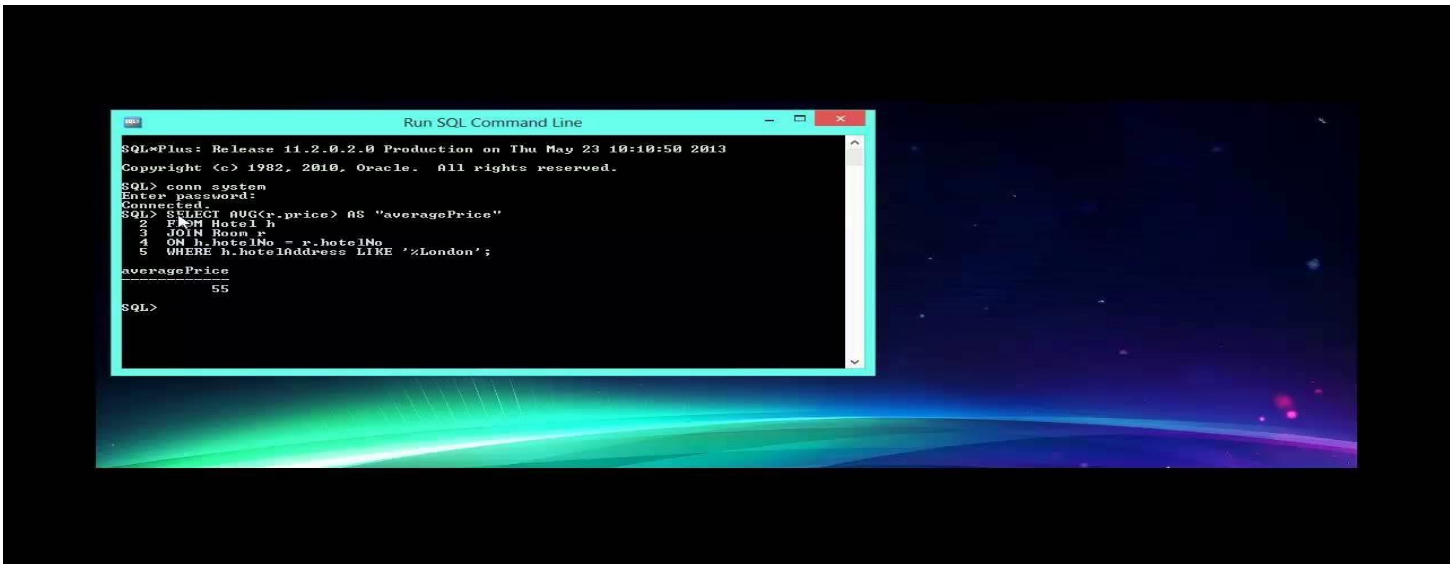
- **Requirements Analysis**: Define the project requirements objectives and the specific features needed for fee management in the school
- **Database Design**: Design the database schema to store information about students, fees transactions, and other relevant data.
- **Data Modelling**: Create entity-relationship diagram(ERD) to represent the relationships.
- **Database implementation**: Implement the database using Oracle 10g by creating tables defining constraints and writing stored procedure or triggers as needed.

- **Application Development**: Develop the user interface and application logic using a programming language or development platform that can connect to Oracle 10g such as Java, NET, or Oracle Forms.
- **Testing**: Thoroughly test the application to ensure that it functions correctly and securely.
- **Development**: Deploy the application for the school's computer or servers.
- **Maintenance and Support**: Provide ongoing maintenance and support to address issues, update the system and add new features as required.

Reasons For choosing Oracle 10g Backend:

- **Data Integrity:** Oracle databases are known for their robust data integrity and ACID compliance, making them suitable for critical applications where data accuracy is paramount.
- **Scalability:** Oracle 10g provides scalability options, enabling your system to grow as your project's demands increase.
- **Security**: Oracle databases offer advanced security features to protect your data from unauthorized access, ensuring compliance with data protection regulations.
- **Rich Feature Set**: Oracle 10g provides a wide range of features for data management, including partitioning, indexing, and data warehousing capabilities.
- **Performance**: Oracle databases are known for their performance optimization features, which can be critical for applications requiring high throughput.
- **Reliability**: Oracle databases have a strong reputation for reliability and uptime, which is essential for mission-critical applications.

- **Support and Community**: Oracle has a large user community and extensive documentation, making it easier to find support and resources for troubleshooting and development.
- **Integration**: If our project requires integration with other Oracle-based systems or legacy databases, using Oracle 10g can simplify these connections.
- **Legacy Compatibility**: If our project involves maintaining or migrating legacy systems that use Oracle 10g, it might be more practical to continue using it.
- **Cost Considerations**: Depending on our organization's licensing and budget constraints, Oracle 10g may be a cost-effective option compared to newer database systems.



```
SQL*Plus: Release 11.2.0.2.0 Production on Thu May 23 10:10:50 2013

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn system
Enter password:
Connected.
SQL> SELECT AVG(r.price) AS "averagePrice"
  2  FROM Hotel h
  3  JOIN Room r
  4  ON h.hotelNo = r.hotelNo
  5  WHERE h.hotelAddress LIKE '%London';

averagePrice
------------
          55

SQL>
```

MODULE DESCRIPTION

What is module?

In project development, a "module" is a self-contained, organized unit of code that serves a specific function or feature within the overall project. Modules are designed to be relatively independent, making it easier to manage and develop the project as a whole

Users in School Fee Section:

1. Admin

Users Functionality:

Admin:

- Add, Update, Delete and View Student Record
- Add, Update , Delete and View Fee Structure
- New, Update , Delete and View Fee Payment Record
- New, Update , Delete and View Dues Record
- New, Update , Delete and View Fine Payment Record
- Generate report of student ,class, payment ,Dues and fine

Main Modules:

- **User management module:** This module login administrator to access the dashboard.

- **Class module:** This module give an option to the administrator to see all class student report.

- **Student module:** This module allows administrator to register the details of new students and check dues of students.

    - **Add student:** This sub module give an option to add new student details.

- **Fee payment module:** This module gives the payment details of student by checking all the table.
- **Report module:** This module generates reports on fee collections, outstanding fees, and other related data.

- **Fee receipt generation process :** This process generates fee receipts for students after they have paid their fees.

- **Student details table :** This table shows the details of students.

- **Fee structure table:** This table shows the fee structure of school.

- **Fee Dues table:** This table shows the dues of student and status of payment also.

- **Fee Fine table:** This table shows the fine of student.

How it works?

Admin can search Student details, payment details and other details using this software. The form shows the details after user searches. The search result shows the details that he want to search. User can click and do all the works that was in the form like updating fee structure, update student info, etc.

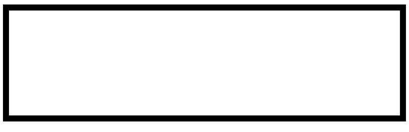
ENTITY RELATIONSHIP DIAGRAM(ERD)

An entity-relationship model(ER model) is a data model for describing the data or information aspects of a business domain or process requirements, in abstract way that lends itself to ultimately being implemented in a database such as relational database. The main components of ER model are entities (things) and the relationships that can exist among them, and databases. An entity may be defined as thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain. When we speak of an entity, we normally speak of some aspect of the real world that can be distinguished from other aspects of the real world. A relationship captures how entities are related to one another, relationships can be thought as verbs, linking two or more nouns. Examples: as owns relationship between a company and a computer, a supervises relationship between an employee and a department, a performs relationship between an artist and a song, a proved relationship between a mathematician and a theorem. Entities an relationships can both have attributes. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key. Entity-relationship diagrams don't show single entities or single instances or relations. Rather, they show entity sets and relationship sets.

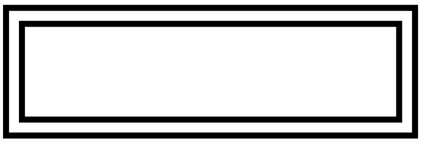
History of ER models

Peter Chen proposed ER Diagram in 1971 to create uniform convention that can be used as a conceptual modelling tool. Many models were presented and discussed, but none were suitable. The data structure diagrams offered by Charles Bachman also inspired his model

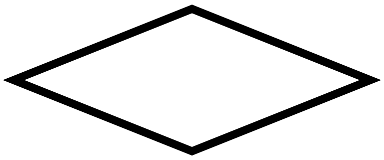
ER Diagram Notation:



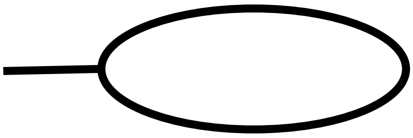
Entity



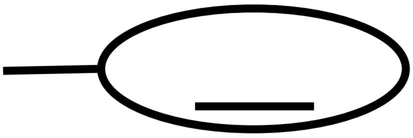
Weak Entity



Relationship

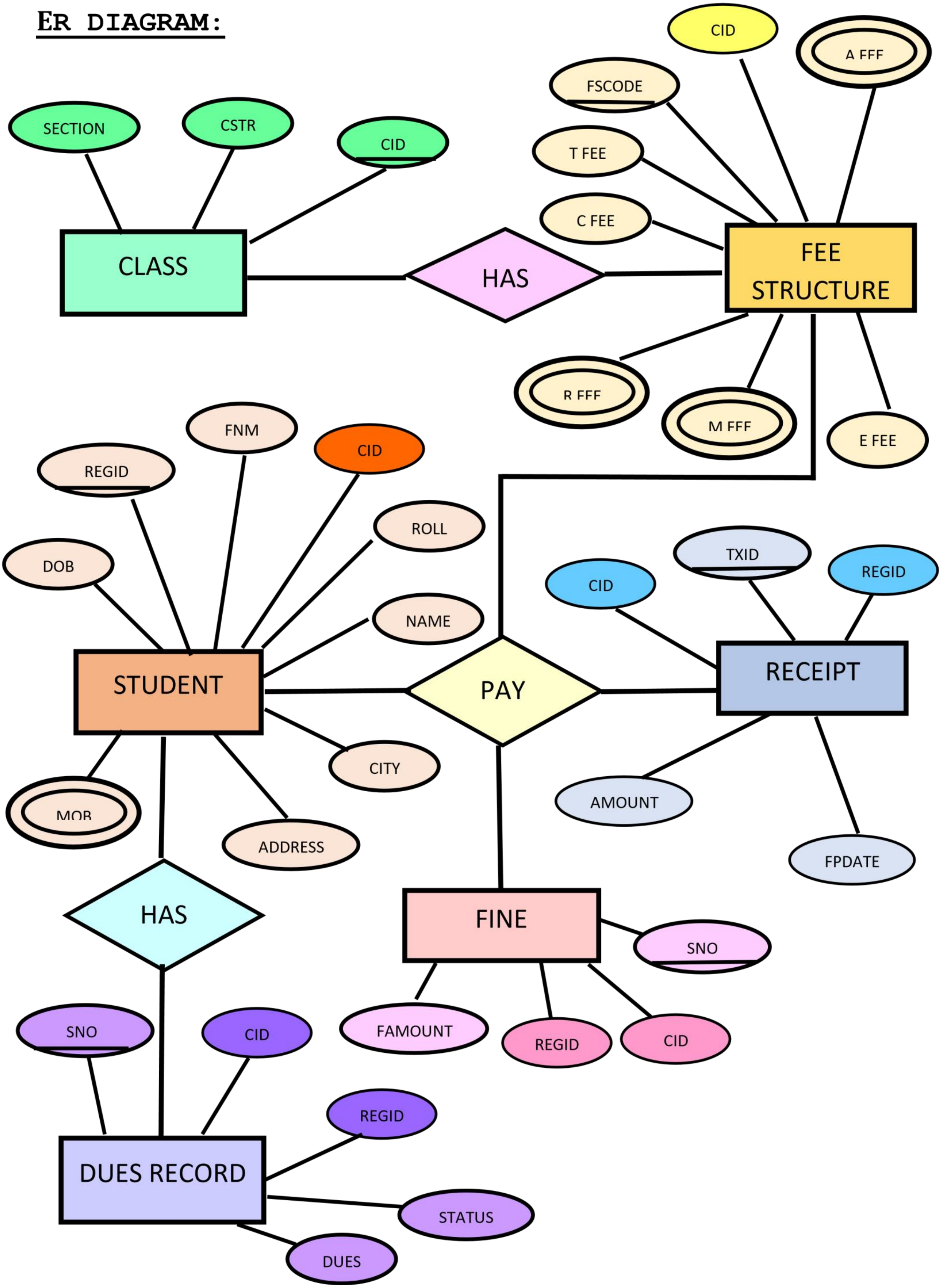


Attribute



Key Attribute

ER DIAGRAM:


CID
A FFF
FSCODE
SECTION
CSTR
CID
T FEE
C FEE
CLASS
HAS
FEE STRUCTURE
R FFF
M FFF
E FEE
FNM
REGID
CID
ROLL
DOB
TXID
CID
REGID
NAME
STUDENT
PAY
RECEIPT
MOB
CITY
AMOUNT
ADDRESS
FPDATE
HAS
FINE
SNO
FAMOUNT
REGID
CID
SNO
CID
REGID
DUES RECORD
STATUS
DUES

DATA FLOW DIAGRAM(DFD)

A Data Flow Diagram(DFD) is a graphical representation of how data flows within a system. DFD are used to model and understand the process and data flow in a system. They can be used to analyse an existing system or model a new one. Like all the best diagram and charts, a DFD can often visually say things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFD remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database oriented software or systems.

Data flow diagrams were popularized in late 1970s, arising from the book Structured Design, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the "data flow graph" computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

There are different types of symbols and rules associated with DFD. Here are some of the key elements and rules for creating DFD:

External Entities: These are entities outside of the system that interact with it. They are represented as rectangles and are labelled to indicate their names.

Processes: These are the activities or functions within the system that transform data. Processes are represented as circles or ovals and are labelled with a process number.

Data Stores: These represent where data is stored within the system. Data stores are usually represented as two parallel lines.

Data Flows: Arrows represent data flows , which indicate the movement of data between external entities, processes, and data stores. Data flows are labelled to describe the data being transferred.

Data Flow Direction: Data flows typically move from external entities to processes, from processes to data store, and from data stores to processes. The direction of the arrows indicate the flow of data.

Control Flows: In some cases, we can use dashed lines with arrows to represent control flows, which show the sequence of processes.

Rules for creating DFD:

Consistency: Ensure that the DFD is consistent with the real-world system we are modelling.

No Cross-Lines: Data flows should not cross over each other, as this can lead to confusion.

Balancing Flows: The total number of data flow arrows into a process should be equal to the total number of data flow arrows out of the process.

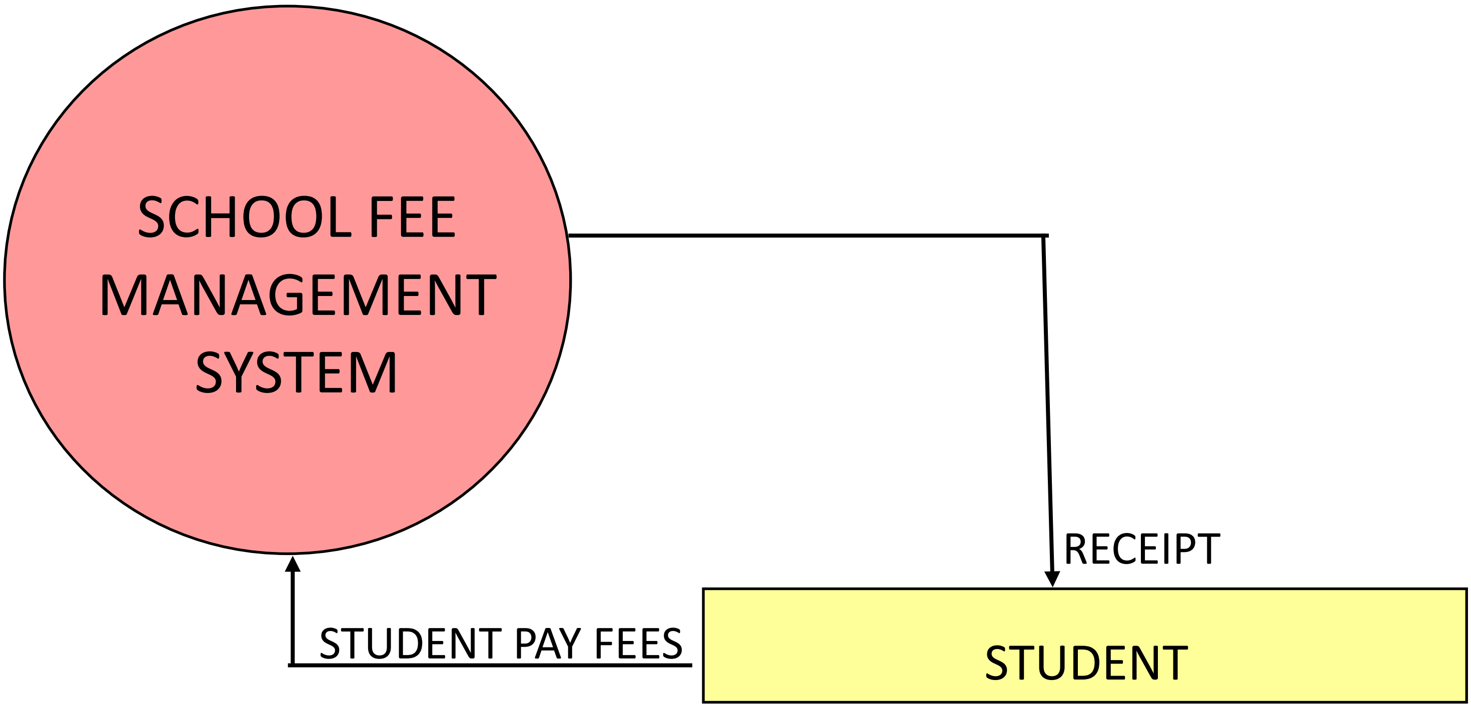
External Entities: Every process must have at least one input data flow and output data flow.

Data Store Usage: Data stores should be connected to processes by data flows, and processes should read from and write to data stores.

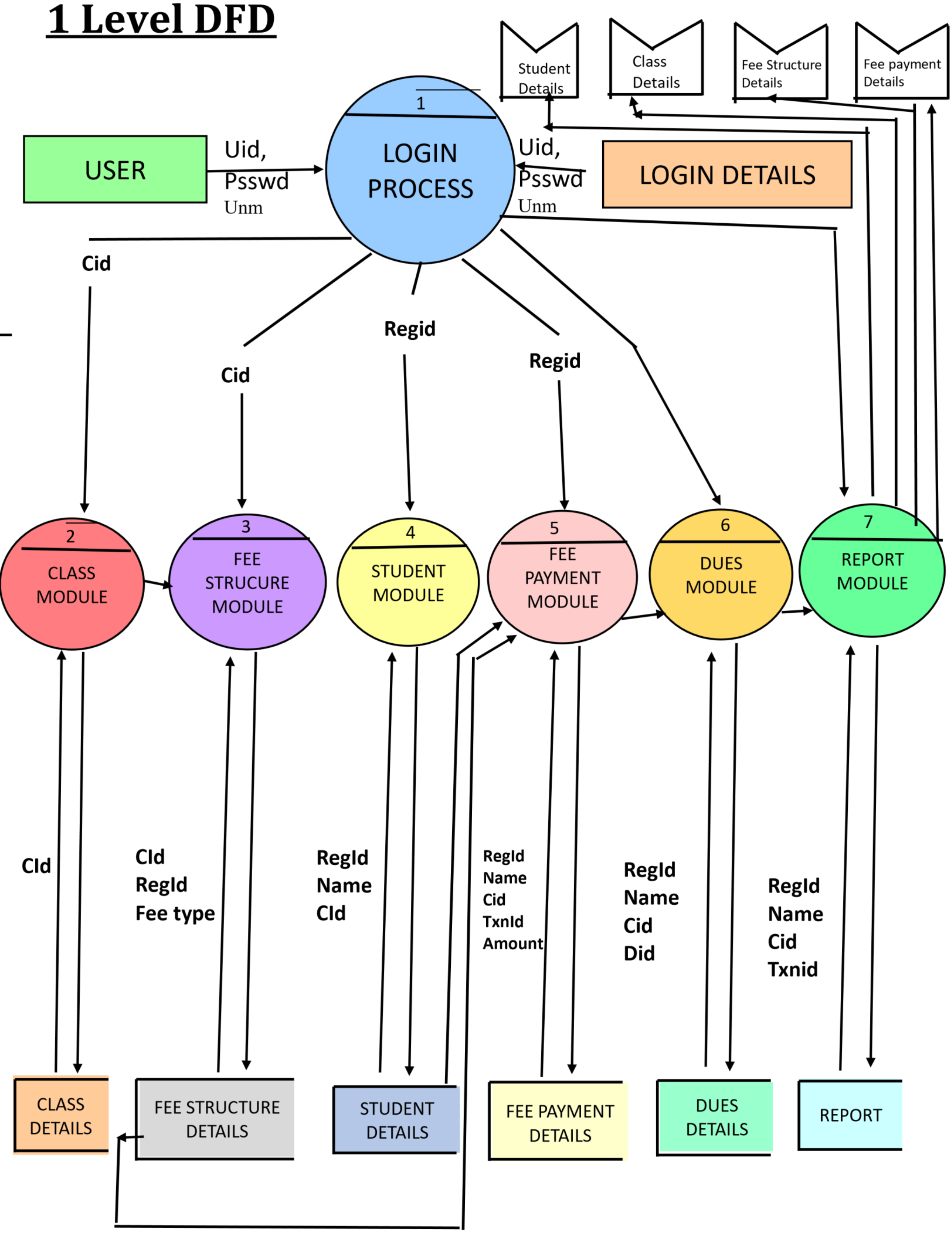
Naming Conventions: Use clear an concise labels for processes, data flows, data stores, and external entities to ensure clarity.

Hierarchy: DFD can be hierarchical with a context-level DFD providing an overview of the entire system and lower-level DFD providing more detail for specific processes.

Level 0

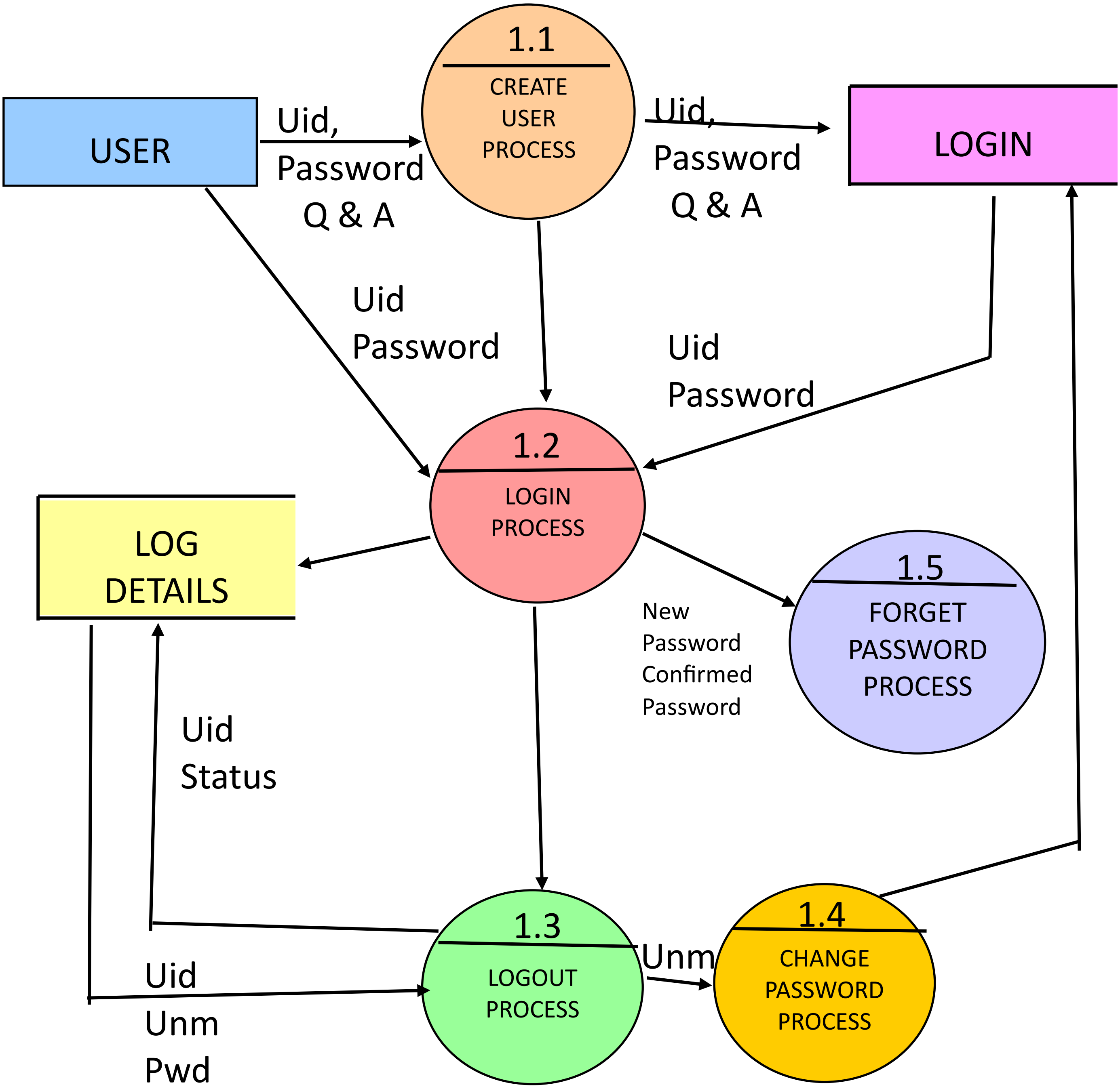

SCHOOL FEE MANAGEMENT SYSTEM
RECEIPT
STUDENT PAY FEES
STUDENT

1 Level DFD

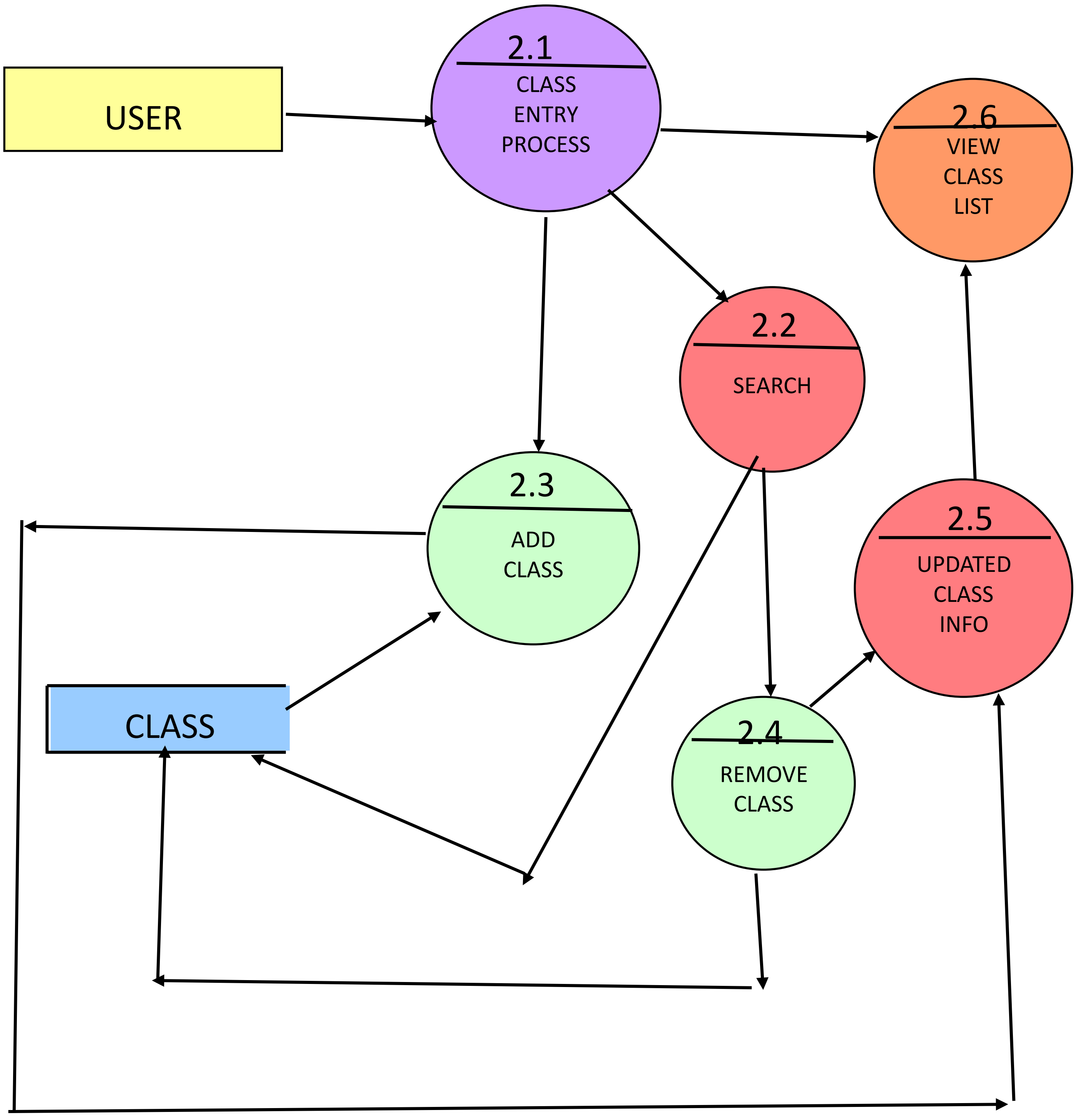

Student Details
Class Details
Fee Structure Details
Fee payment Details
1
LOGIN PROCESS
USER
Uid,
Psswd
Unm
Uid,
Psswd
Unm
LOGIN DETAILS
Cid
Cid
Regid
Regid
2
CLASS MODULE
3
FEE STRUCURE MODULE
4
STUDENT MODULE
5
FEE PAYMENT MODULE
6
DUES MODULE
7
REPORT MODULE
CId
CId
RegId
Fee type
RegId
Name
CId
RegId
Name
Cid
TxnId
Amount
RegId
Name
Cid
Did
RegId
Name
Cid
Txnid
CLASS DETAILS
FEE STRUCTURE DETAILS
STUDENT DETAILS
FEE PAYMENT DETAILS
DUES DETAILS
REPORT

LEVEL 2:

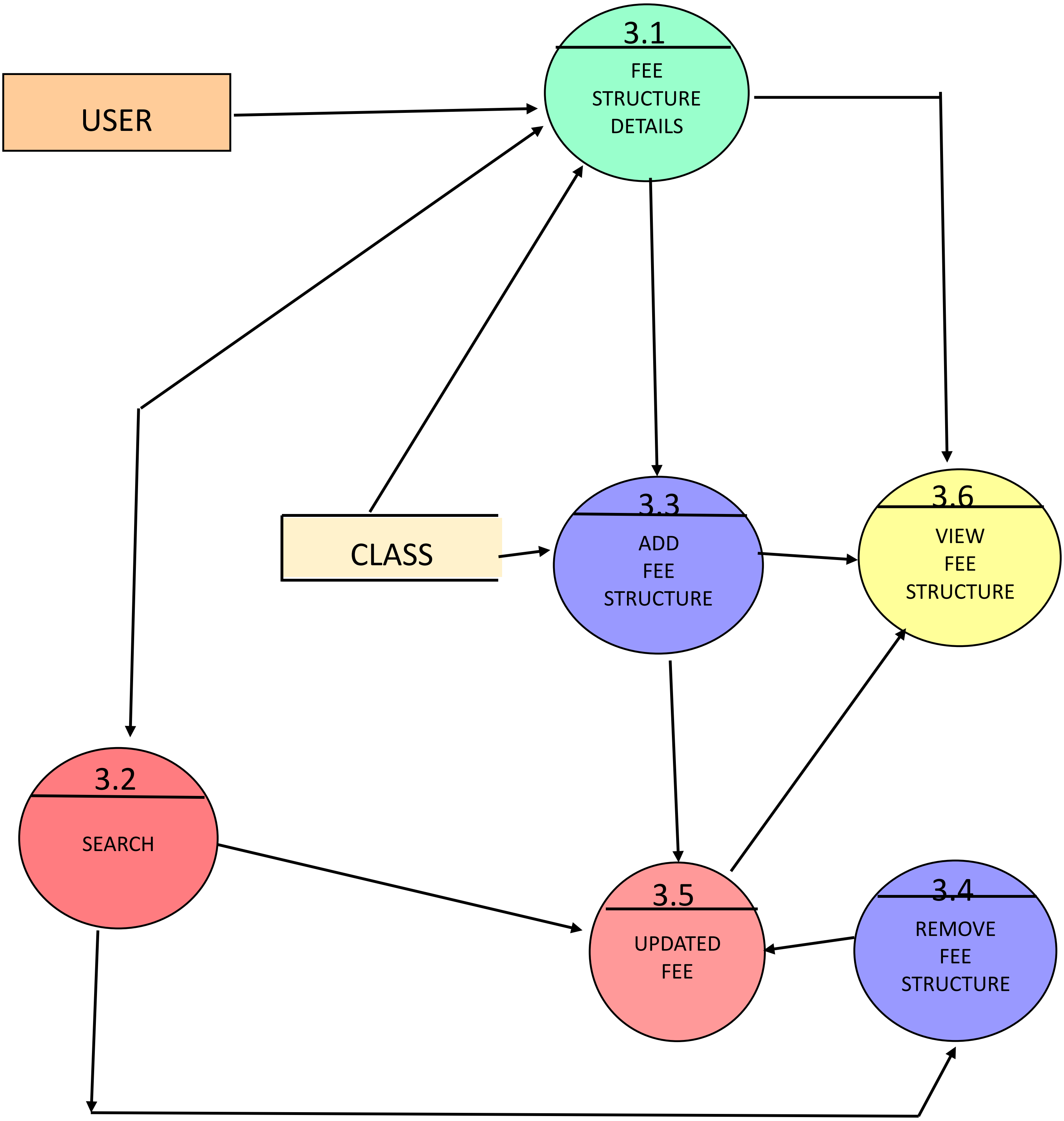
1 Login



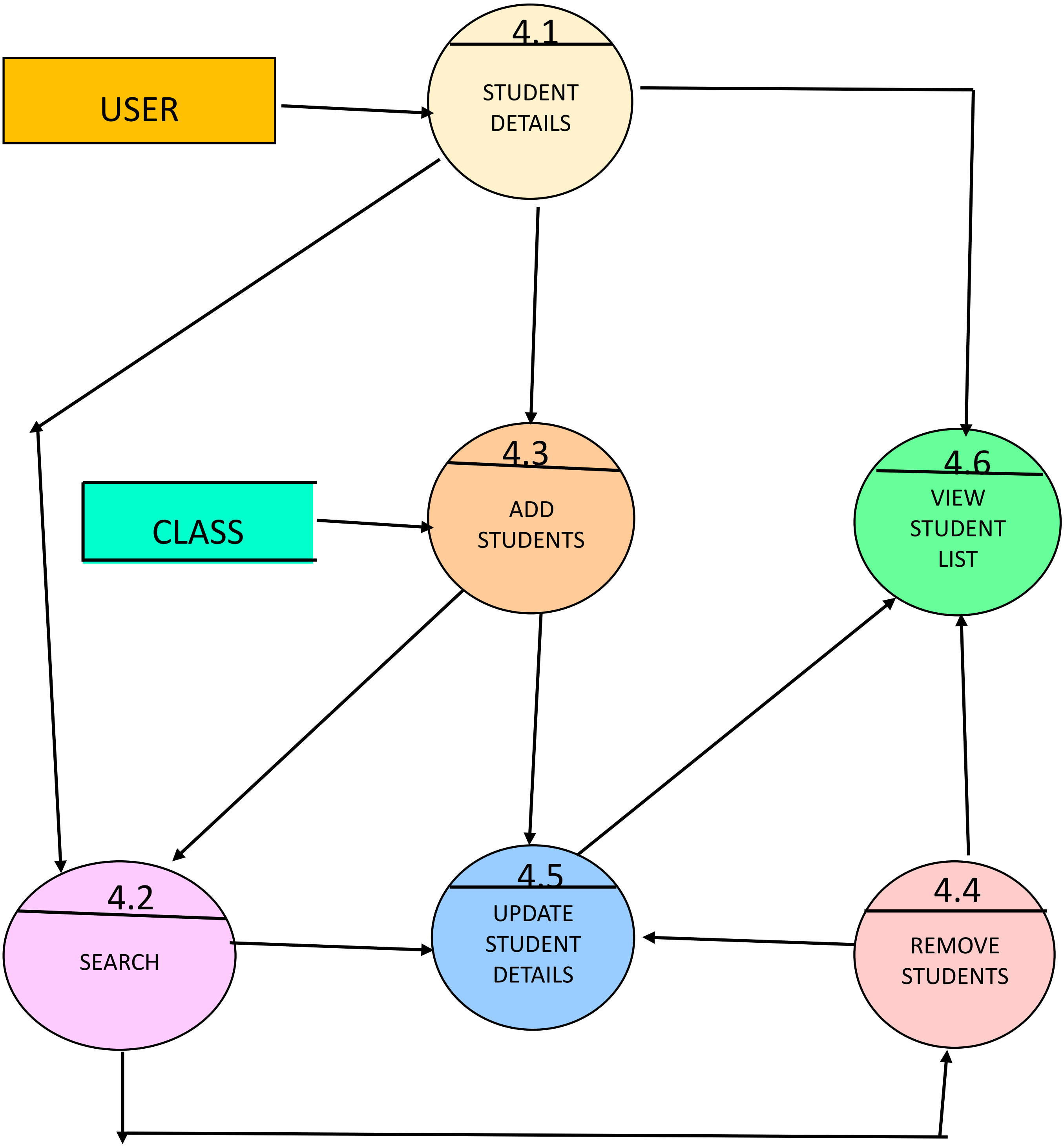
2 Class



3 Fee Structure


USER
3.1
FEE STRUCTURE DETAILS
CLASS
3.3
ADD FEE STRUCTURE
3.6
VIEW FEE STRUCTURE
3.2
SEARCH
3.5
UPDATED FEE
3.4
REMOVE FEE STRUCTURE

4 Student



5 Fee Payment

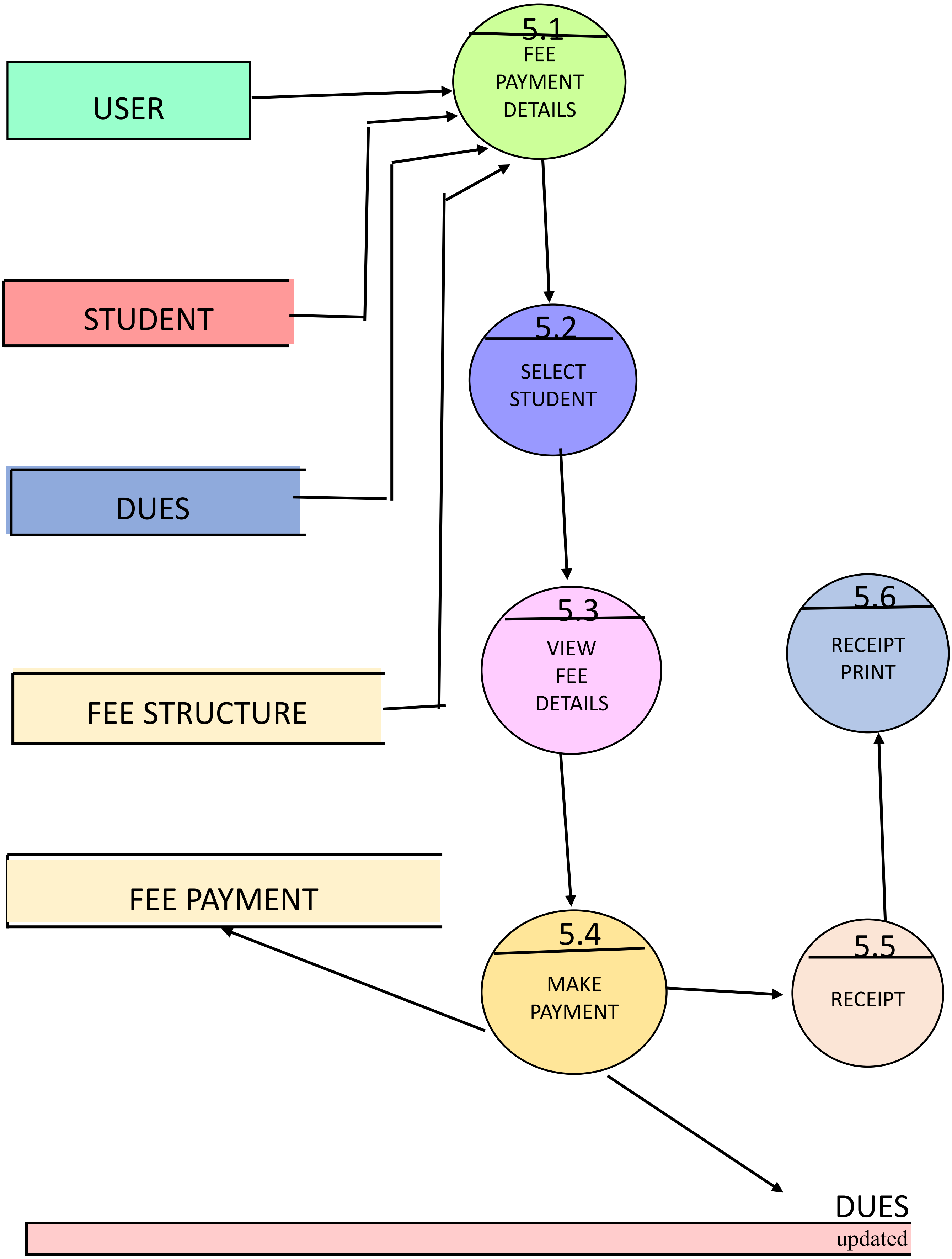


Table Description

LOGIN

S.No.	Field Name	Data Type	Size	Constraint	Description
1	UID	VARCHAR	20	PRIMARY KEY	Used to login in the system
2	PWD	VARCHAR	20	NOT NULL	Password
3	UNM	VARCHAR	20	NOT NULL	User name

LOGIN DETAILS

S.No.	Field Name	Data Type	Size	Constraint	Description
1	SNO	VARCHAR	5	PRIMARY KEY	Counts a serial number for user login
2	UID	VARCHAR	20	FOREIGN KEY	Used to store the Name of user login id
3	STATUS	VARCHAR	10	NOT NULL	Status after login to the system

Class Table

S.No.	Field Name	Data Type	Size	Constraint	Description
1	CID	VARCHAR	10	PRIMARY KEY	Class Id
2	SECTION	VARCHAR	10	NOT NULL	Section
3	CSTR	NUMBER	2	NOT NULL	Class strength

FEE STRUCTURE TABLE

S.No.	Field Name	Data Type	Size	Constraint	Description
1	FSCODE	NUMBER	10	PRIMARY KEY	Fees structure code
2	CID	VARCHAR	10	FOREIGN KEY	Class id
3	ADM_FEE	NUMBER	6,2	NULL	Admission fee
4	REG_FEE	NUMBER	6,2	NULL	Registration fee
5	MISC_FEE	NUMBER	6,2	NULL	Miscellaneous fee
6	EXAM_FEE	NUMBER	6,2	NULL	Examination fee
7	TUT_FEE	NUMBER	6,2	NOT NULL	Tuition fee
8	COMP_FEE	NUMBER	6,2	NOT NULL	Computer fee
9	FINE	NUMBER	6,2	NOT NULL	Fine

STUDENT TABLE

S.No.	Field Name	Data Type	Size	Constraint	Description
1	REGID	NUMBER	15	PRIMARY KEY	Registration Number
2	CID	VARCHAR	10	FOREIGN KEY	Class id
3	SECTION	VARCHAR	10	NOT NULL	Section
4	ROLL	VARCHAR	5	NOT NULL	Roll number
5	NAME	VARCHAR	20	NOT NULL	Student Name
6	FNM	VARCHAR	20	NOT NULL	Father Name
7	MNM	VARCHAR	20	NOT NULL	Mother Name
8	DOB	DATE	-	NOT NULL	Date of Birth

9	MOBNO	VARCHAR	12	NULL	Mobile number
10	EMAILID	VARCHAR	25	NULL	Email id
11	GENDER	VARCHAR	15	NOT NULL	Gender
12	CITY	VARCHAR	20	NOT NULL	City
13	SESSION	VARCHAR	10	NOT NULL	Year
14	AADHAR	VARCHAR	12	NULL	Aadhar Number
15	DOA	DATE	-	NOT NULL	Date of Admission
16	ADDRESS	VARCHAR	30	NOT NULL	Address

DUES TABLE

S.No.	Field Name	Data Type	Size	Constraint	Description
1	SNO	VARCHAR	10	PRIMARY KEY	Serial number
2	CID	VARCHAR	10	FOREIGN KEY	Class id
3	SECTION	VARCHAR	10	NOT NULL	Section
4	REGID	VARCHAR	15	FOREIGN KEY	Registration id
5	SNAME	VARCHAR	20	NOT NULL	Student Name
6	DUES	NUMBER	10,2	NULL	Dues
7	STATUS	VARCHAR	10	NOT NULL	Status

FEE PAYMENT TABLE

S.No.	Field Name	Data Type	Size	Constraint	Description
1	TXID	VARCHAR	10	PRIMARY KEY	Transcition Id
2	CID	VARCHAR	10	FOREIGN KEY	Class id
3	SECTION	VARCHAR	10	NOT NULL	Section
4	REGID	VARCHAR	15	FOREIGN KEY	Registration id
5	SNAME	VARCHAR	20	NOT NULL	Student Name
6	AMOUNT	NUMBER	10,2	NOT NULL	Payable Amount
7	FPDATE	DATE	-	NOT NULL	Fee payment date

FINE TABLE

S.No.	Field Name	Data Type	Size	Constraint	Description
1	SNO	VARCHAR	10	PRIMARY KEY	Serial No
2	CID	VARCHAR	10	FOREIGN KEY	Class id
3	SECTION	VARCHAR	10	NOT NULL	Section
4	REGID	VARCHAR	15	FOREIGN KEY	Registration id
5	SNAME	VARCHAR	20	NOT NULL	Student Name
6	FAMOUNT	NUMBER	10	NULL	Fine Amount

PROJECT PLANNING

PHASES	**MEMBERS**	**TOTAL DAYS**
➢ ANALYSIS PRELIMINARY INVESTIGATION FEASIBILITY STUDY COSTBENEFIT ANALYSIS SYNOPSIS	SUBAL ANAND VIKARN KUMAR JHA SATYAM KUMAR UTKARSH KUMAR	TIME TAKEN = 73 DAYS
➢ DESIGN	SUBAL ANAND SATYAM KUMAR VIKARN KUMAR JHA	TIME TAKEN = 27 DAYS
➢ CODING	SUBAL ANAND SATYAM KUMAR VIKARN KUMAR JHA	TIME TAKEN =47 DAYS
➢ TESTING	SUBAL ANAND VIKARN KUMAR JHA	TIME TAKEN = 24 DAYS
➢ DOCUMENTATION	SUBAL ANAND VIKARN KUMAR JHA SATYAM KUMAR UTKARSH KUMAR	TIME TAKEN = 11 DAYS

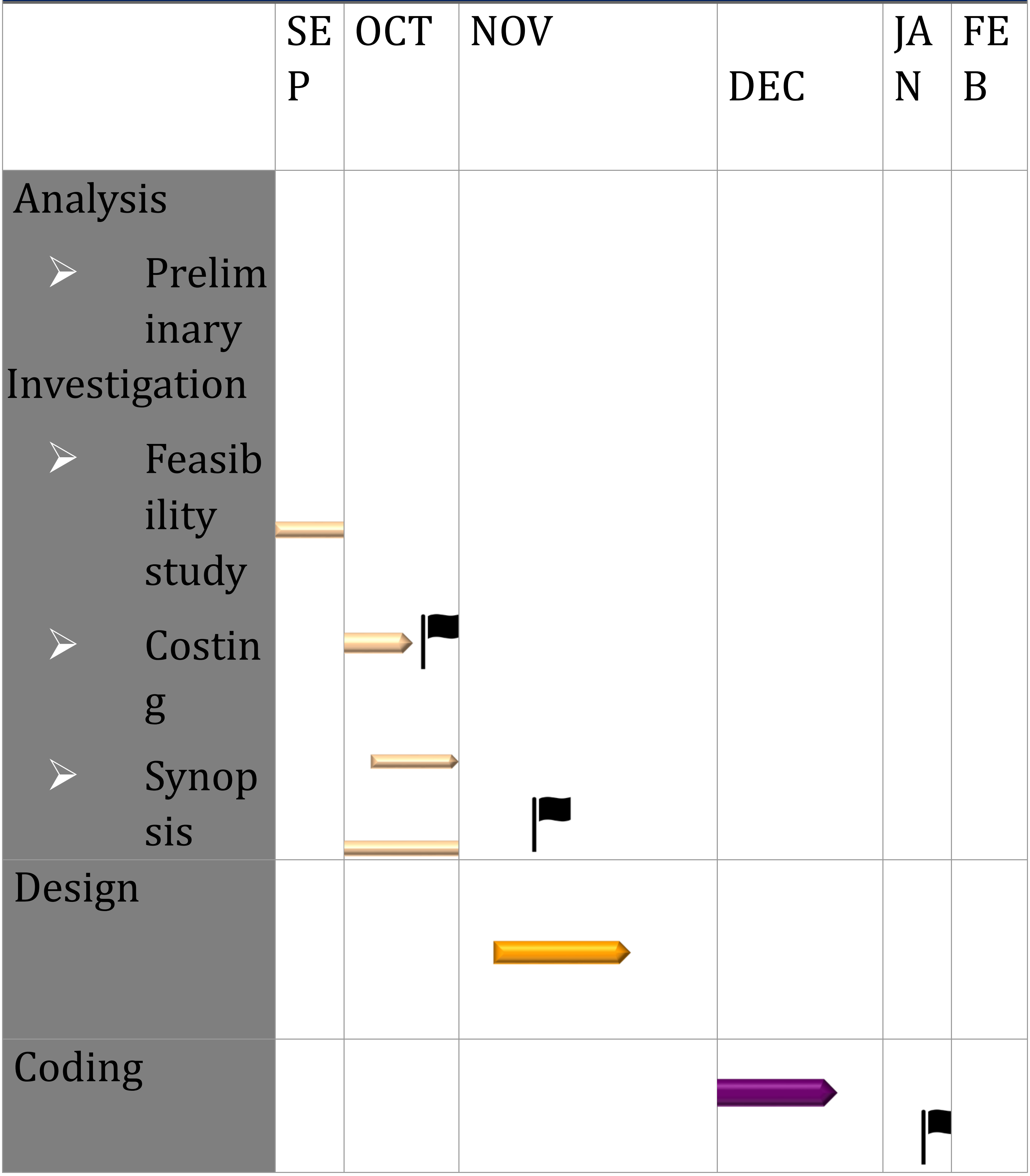
□ **Start date:-** 01-09-2023

□ **Last date:-** 29-02-2024

□ **Total days:-** 182 Days

➢➢ **Total weeks:Months:-** 6 Months**-** 26 Weeks

GANTT CHART(Time Scheduling Chart)



Testing				🟥		⚑
Documentation			🟥			⚑

FUTURE SCOPE

Modification can easily done according to requirements as when necessary.

This application can be easily implemented under various situation. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

Software Scope:

- **Extensibility:** This software is extendable in ways that its original developers may not accept. The following principle enhances extensibility like hide data structure, avoid traversing multiple links or methods, avoid case statements on object type and distinguish public and private operation.
- **Reusability:** Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort over several design. Reducing this amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both type of reusability. Sharing of newly written code within a project and reuse of previously written code on new projects.
- **Understandability:** A method is understandable if someone other than creator of the method can understand the code(as

well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.

- **Backup and Restore:** Develop a mechanism for data backup and restoration to prevent data loss in case of system failures.