

- 1. LIBRARY MANAGEMENT**
- 2. AIRLINE**
- 3. COLLEGE (DFD , USE CASE , USE CLASS , SEQUENCE)**
- 4. ONLINE SHOPPING (DFD , USE CASE , STATE)**
- 5. BANK**
- 6. medical**
- 7. HOTEL (ONLY DFD)**
- 8. AUTH SEQUENCE DAIGRAM**
- 9. SAFE HOME**

LIBRARY MANAGEMENT SYSTEM

CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM

In Object - Oriented modelling , the main building block generally represents different objects in a system, their attributes, their different functions, and relationships among objects. These building blocks are known as Class Diagram.

Class diagrams are generally used for conceptual modeling of static view of a software application, and for modeling translating models into programming code in a detailed manner. At time of developing or construction software systems, a class diagram is widely used. They are also used for data modeling. It is used to show classes, relationships among them, interface, association, etc. Class in a class diagram simply is a blueprint of an object. It simply describes and explains different type of objects in system, and different types of relationships that exist between them.

Class Diagram for Library Management System :

Aggregation and Multiplicity are two important points that need to take into consideration while designing a Class Diagram. Let us understand in detail.

1. **Aggregation** - Aggregation simply shows a relationship where one thing can exist independently of other thing. It means to create or compose different abstractions together in defining a class. Aggregation is represented as a part of relationship in class diagram. In diagram given below, we can see that aggregation is represented by an edge with a diamond end pointing towards superclass. The “Library Management

System” is superclass that consists of various classes. These classes are User, Book, and Librarian as shown in diagram. Further, for “Account” class, “User” is a superclass. All of these, share a relationship and these relationships are known as aggregate relationships.

2. **Multiplicity** - Multiplicity means that number of elements of a class is associated with another class. These relations can be one-to-one, many-to-many, and many-to-one or one-to-many. For denoting one element we use 1, for zero elements we use 0, and for many elements we use *. We can see in diagram; many users are associated with many books denoted by * and this represents a many-to-many type of relationship. One user has only one account that is denoted by 1 and this represents a one-to-one type of relationship. Many books are associated with one librarian and this represents many-to-one or one-to-many type of relationship. All these relationships are shown in diagram.

Class Diagram for Library Management System simply describes structure of Library Management System class, attributes, methods or operations, relationship among objects.

Classes of Library Management System :

- Library Management System class - It manages all operations of Library Management System. It is central part of organization for which software is being designed.
- User Class - It manages all operations of user.
- Librarian Class - It manages all operations of Librarian.
- Book Class - It manages all operations of books. It is basic building block of system.
- Account Class - It manages all operations of account.
- Library database Class - It manages all operations of library database.
- Staff Class - It manages all operations of staff.
- Student Class - It manages all operations of student.

Attributes of Library Management System :

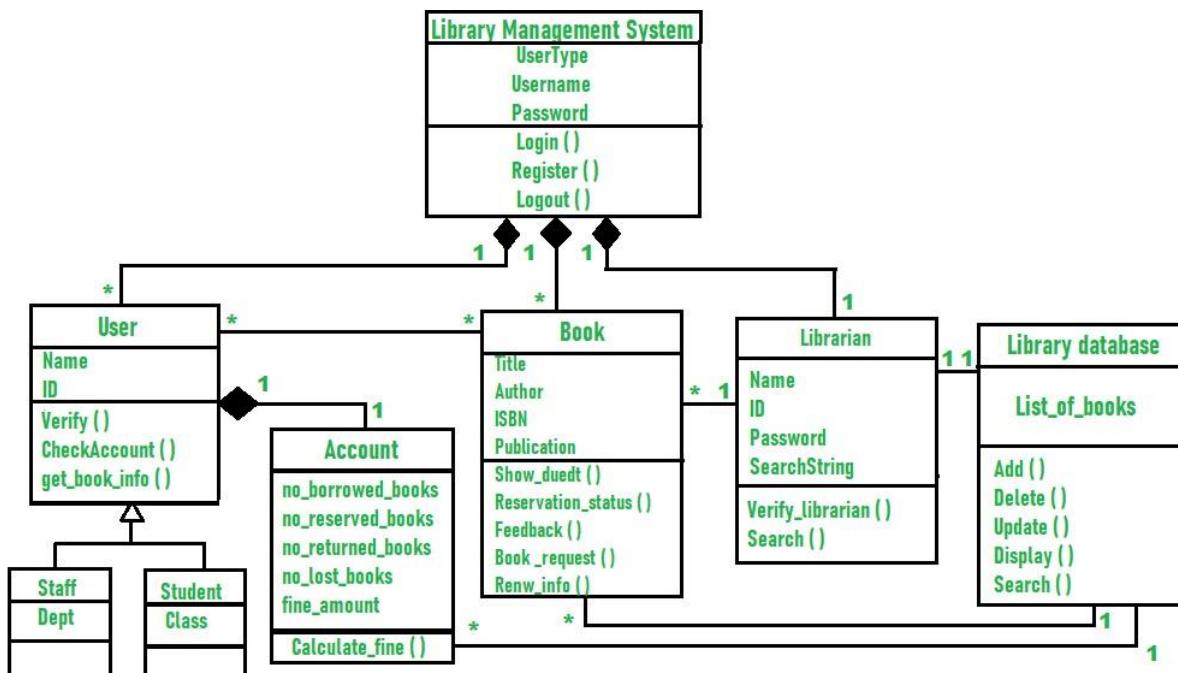
- Library Management System Attributes - UserType, Username, Password
- User Attributes - Name, Id
- Librarian Attributes - Name, Id, Password, SearchString
- Book Attributes - Title, Author, ISBN, Publication
- Account Attributes - no_borrowed_books, no_reserved_books, no_returned_books, no_lost_books fine_amount
- Library database Attributes - List_of_books
- Staff Class Attributes - Dept

- Student Class Attributes - Class

Methods of Library Management System :

- Library Management System Methods - Login(), Register(), Logout()
- User Methods - Verify(), CheckAccount(), get_book_info()
- Librarian Methods - Verify_librarian(), Search()
- Book Methods - Show_duedt(), Reservation_status(), Feedback(), Book_request(), Renew_info()
- Account Methods - Calculate_fine()
- Library database Methods - Add(), Delete(), Update(), Display(), Search() Class

Diagram of Library Management System :



CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM

DFD FOR LIBRARY MANAGEMENT SYSTEM

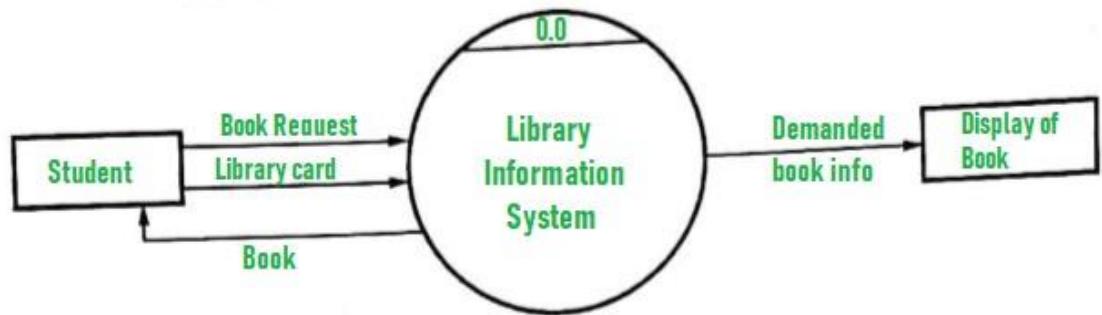
Data Flow Diagram (DFD) depicts the flow of information and the transformation applied when data moves in and out of a system. The overall system is represented and described using input, processing, and output in the DFD. The inputs can be:

- **Book request** when a student requests for a book.
- **Library card** when the student has to show or submit his/her identity as proof.

The overall processing unit will contain the following output that a system will produce or generate:

- The book will be the output as the book demanded by the students will be given to them.
- Information on the demanded book should be displayed by the library information system that can be used by the student while selecting the book which makes it easier for the student.

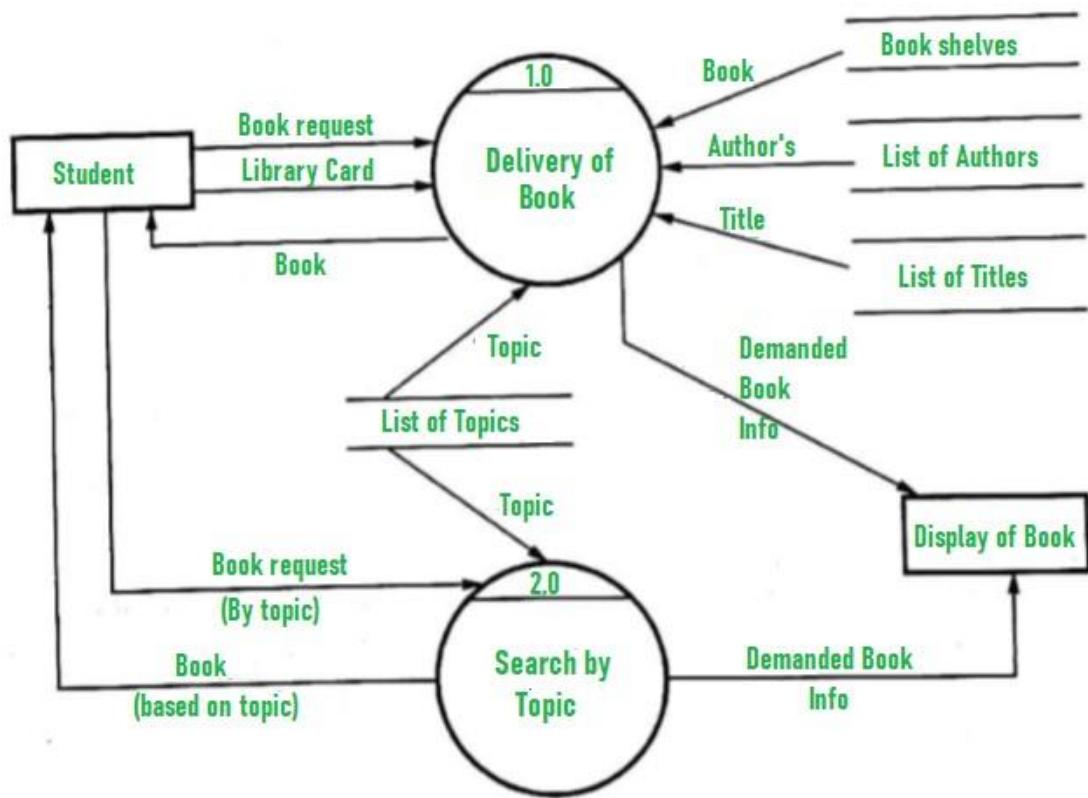
1. Level 0 DFD -



2. Level 1 DFD - At this level, the system has to show or expose with more details of processing. The processes that are important to be carried out are:

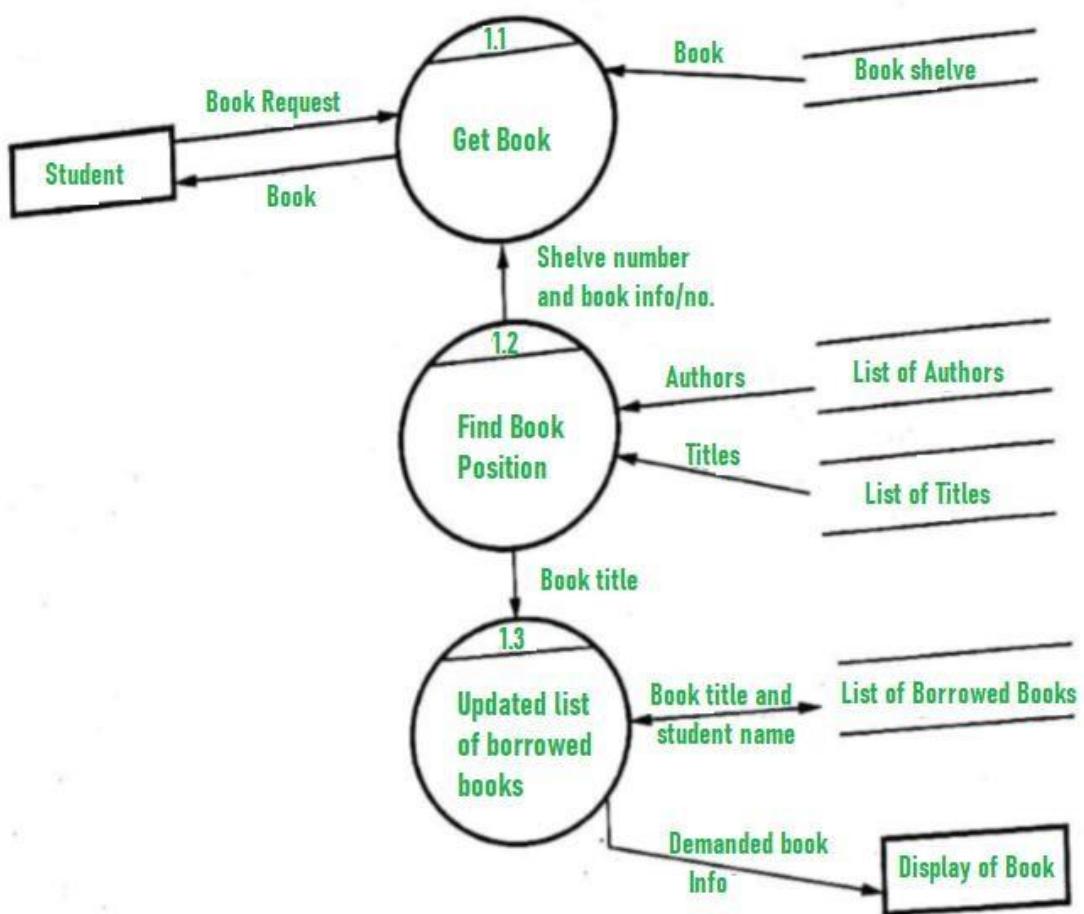
- Book delivery
- Search by topic

List of authors, List of Titles, List of Topics, the bookshelves from which books can be located are some information that is required for these processes. **Data store** is used to represent this type of information.



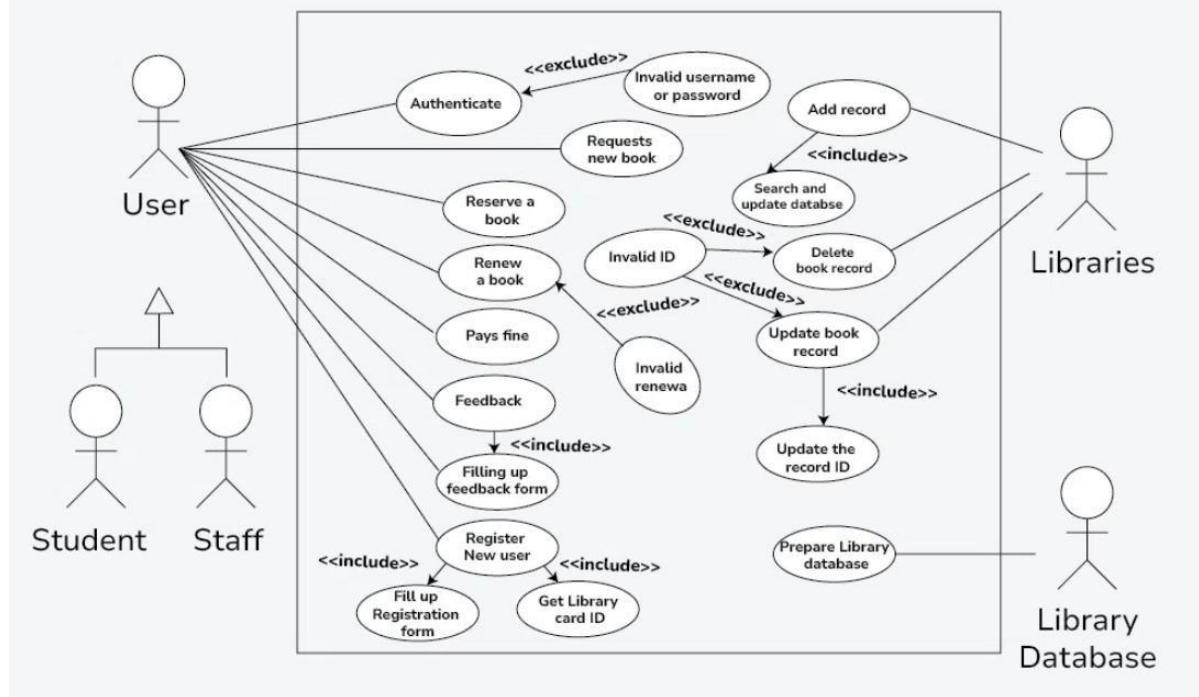
Level 1 DFD

3. Level 2 DFD -

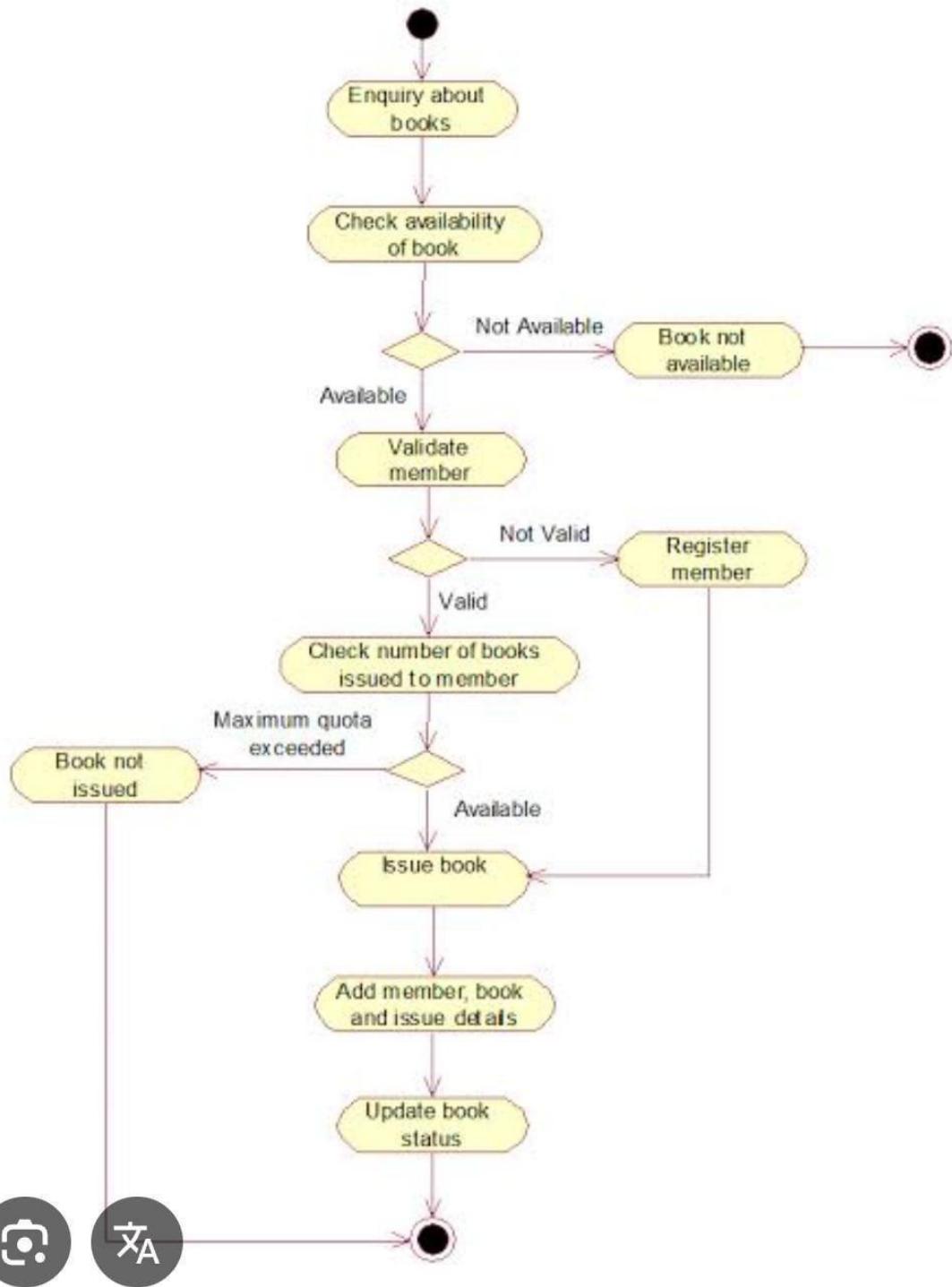


Level 2 DFD

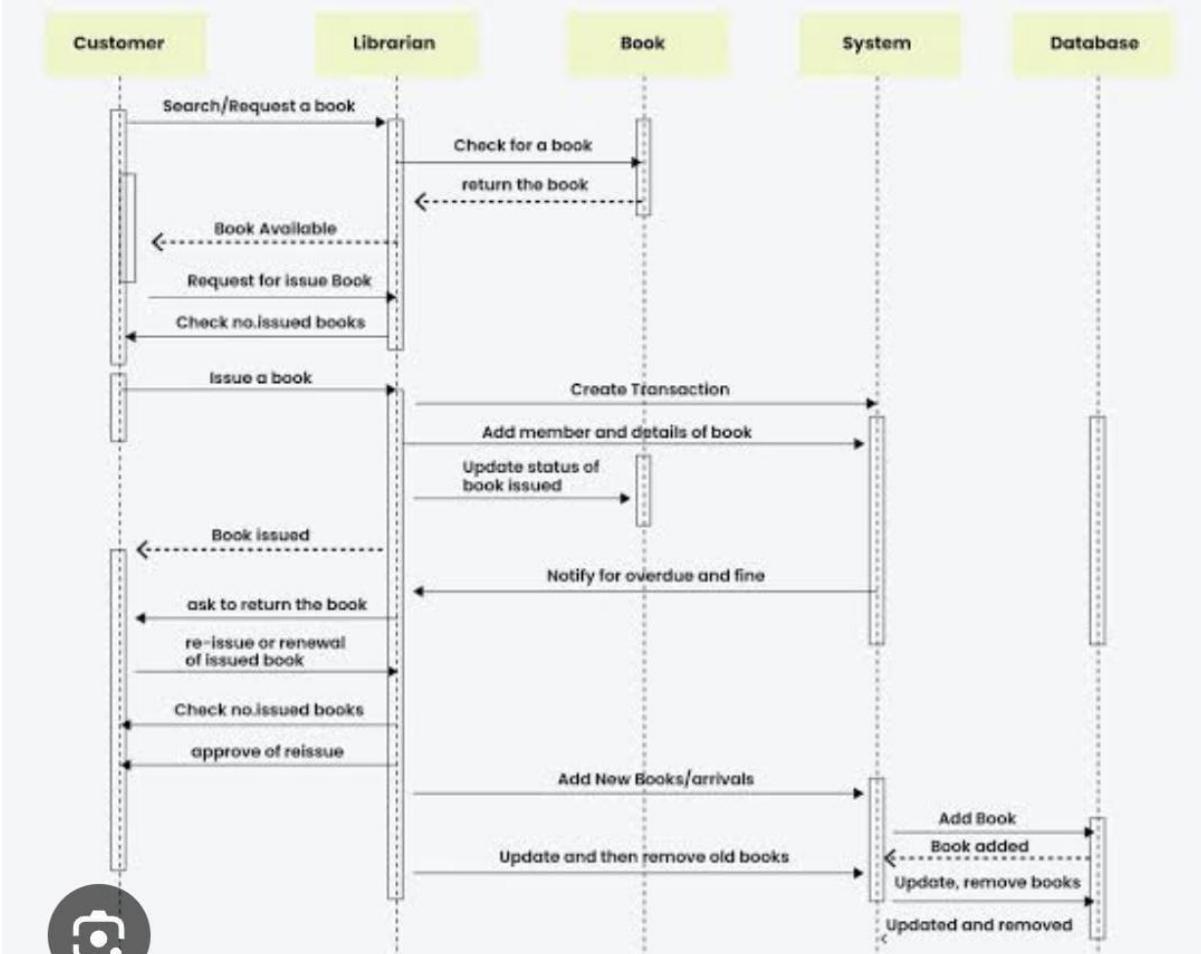
Use Case Diagram of Library Management System



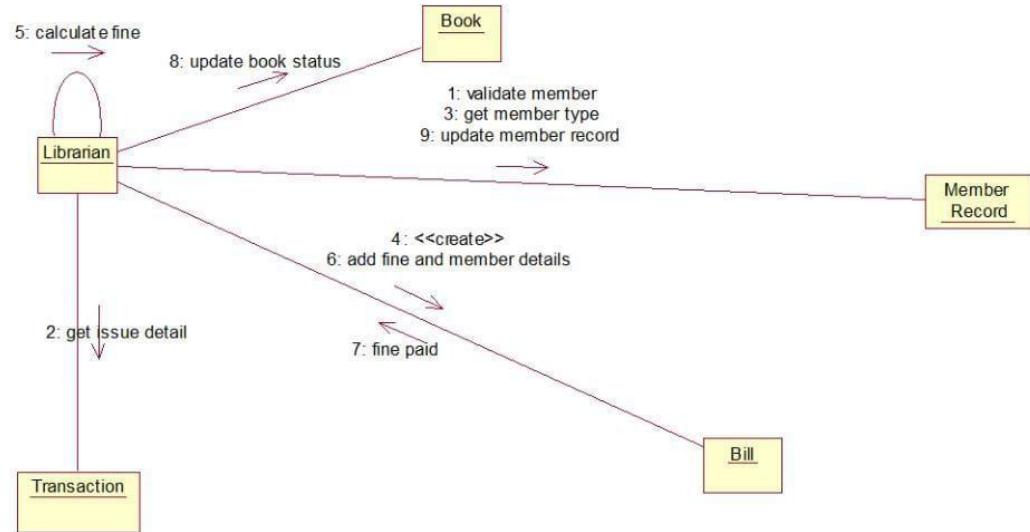
Activity diagram



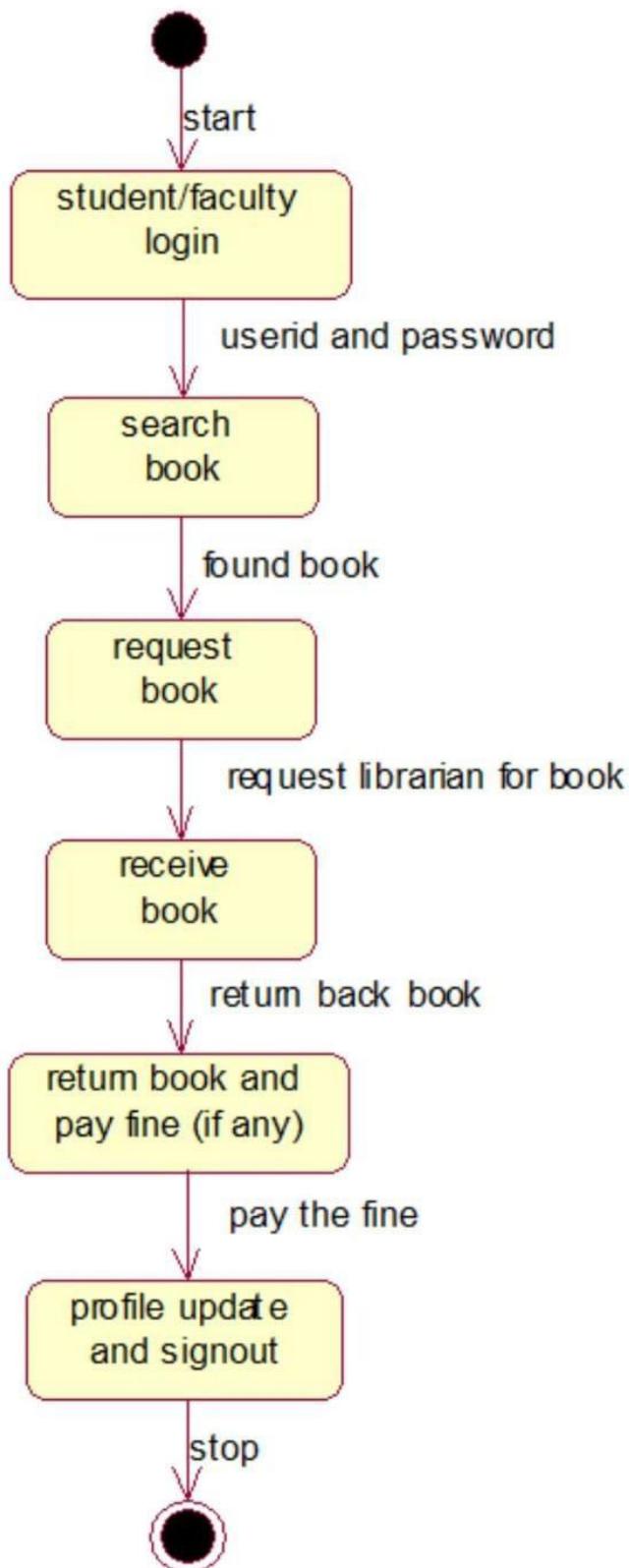
Sequence Diagram for Library Management System Design



Collaboration diagram

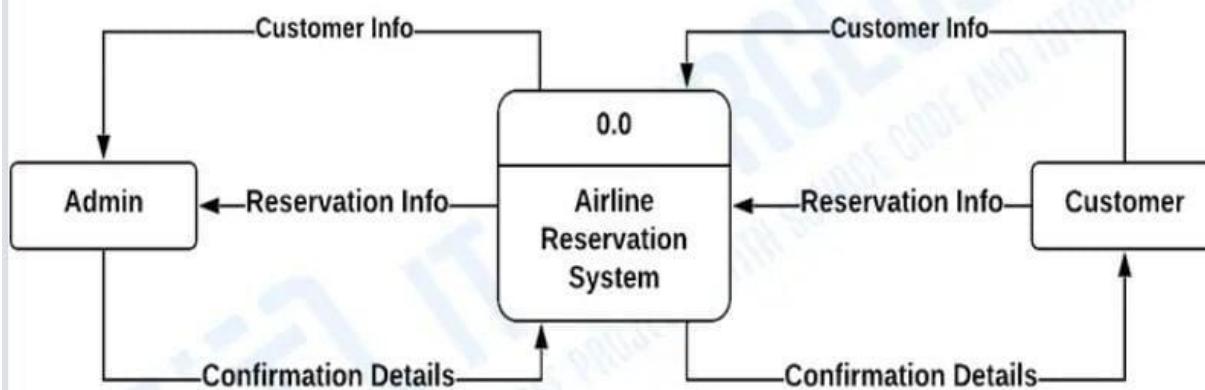


Statechart diagram



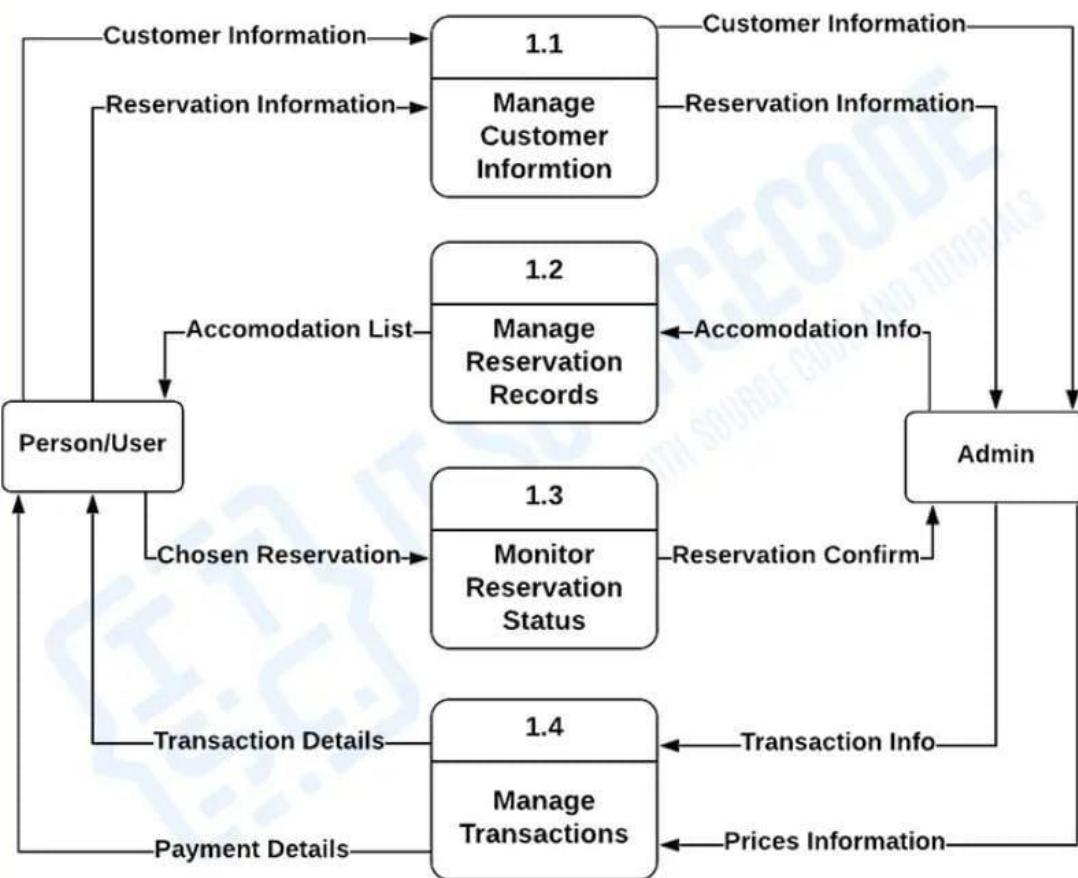
AIRLINE MANAGEMENT

AIRLINE RESERVATION SYSTEM



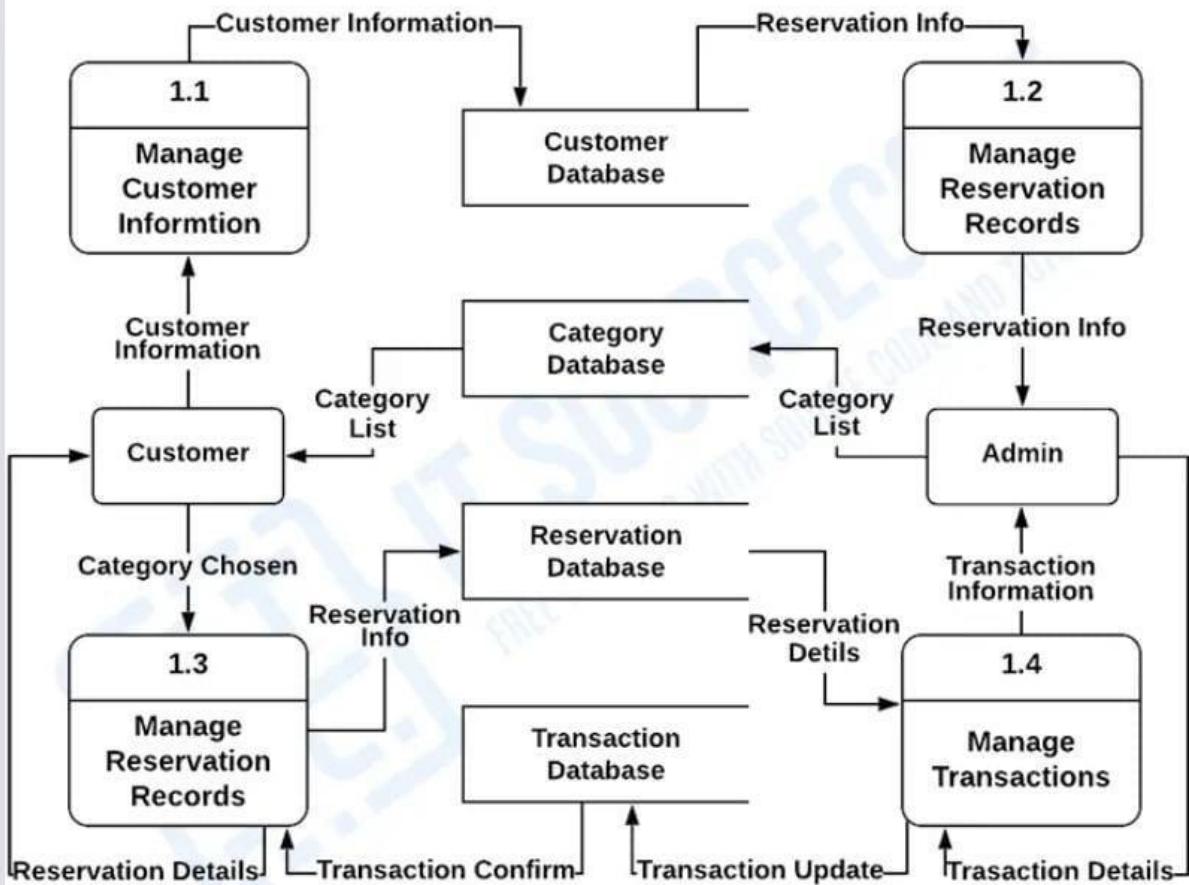
DATA FLOW DIAGRAM LEVEL 0

AIRLINE RESERVATION SYSTEM



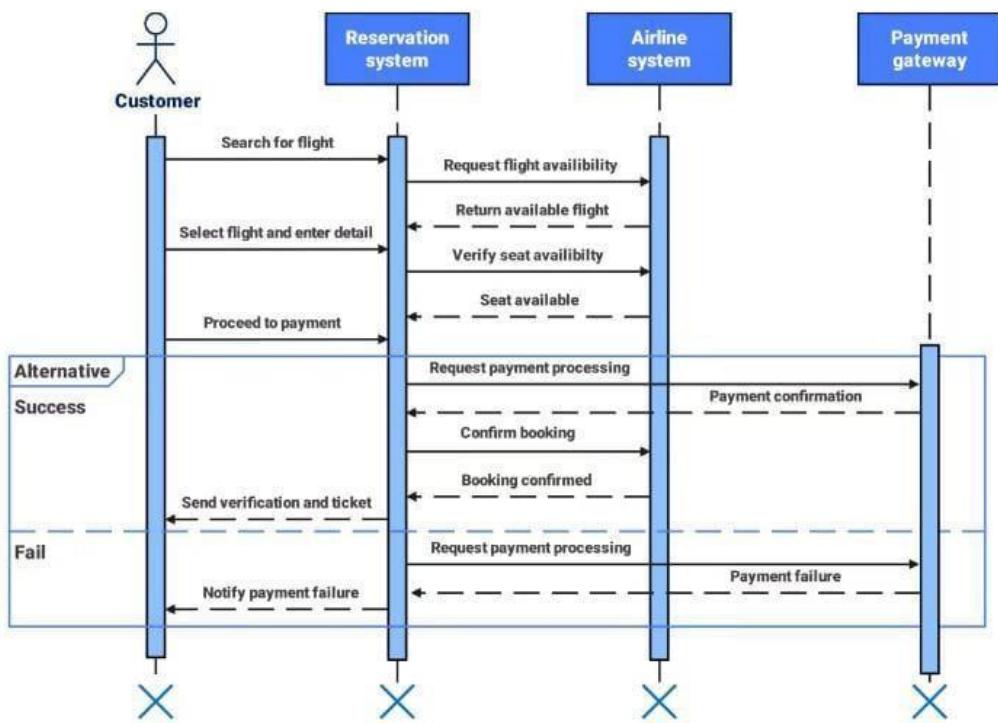
DATA FLOW DIAGRAM LEVEL 1

AIRLINE RESERVATION SYSTEM

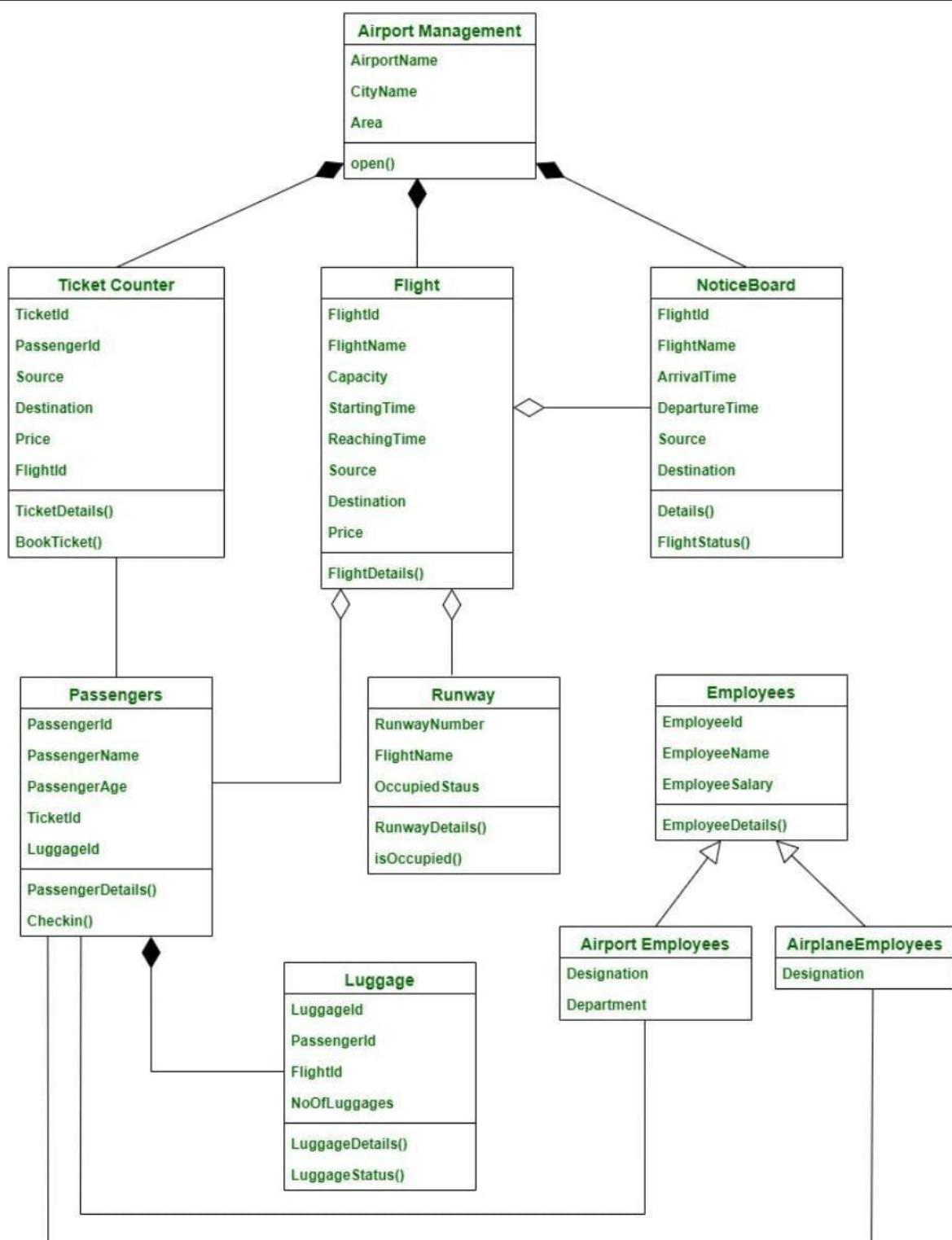


DATA FLOW DIAGRAM | FVFI ?

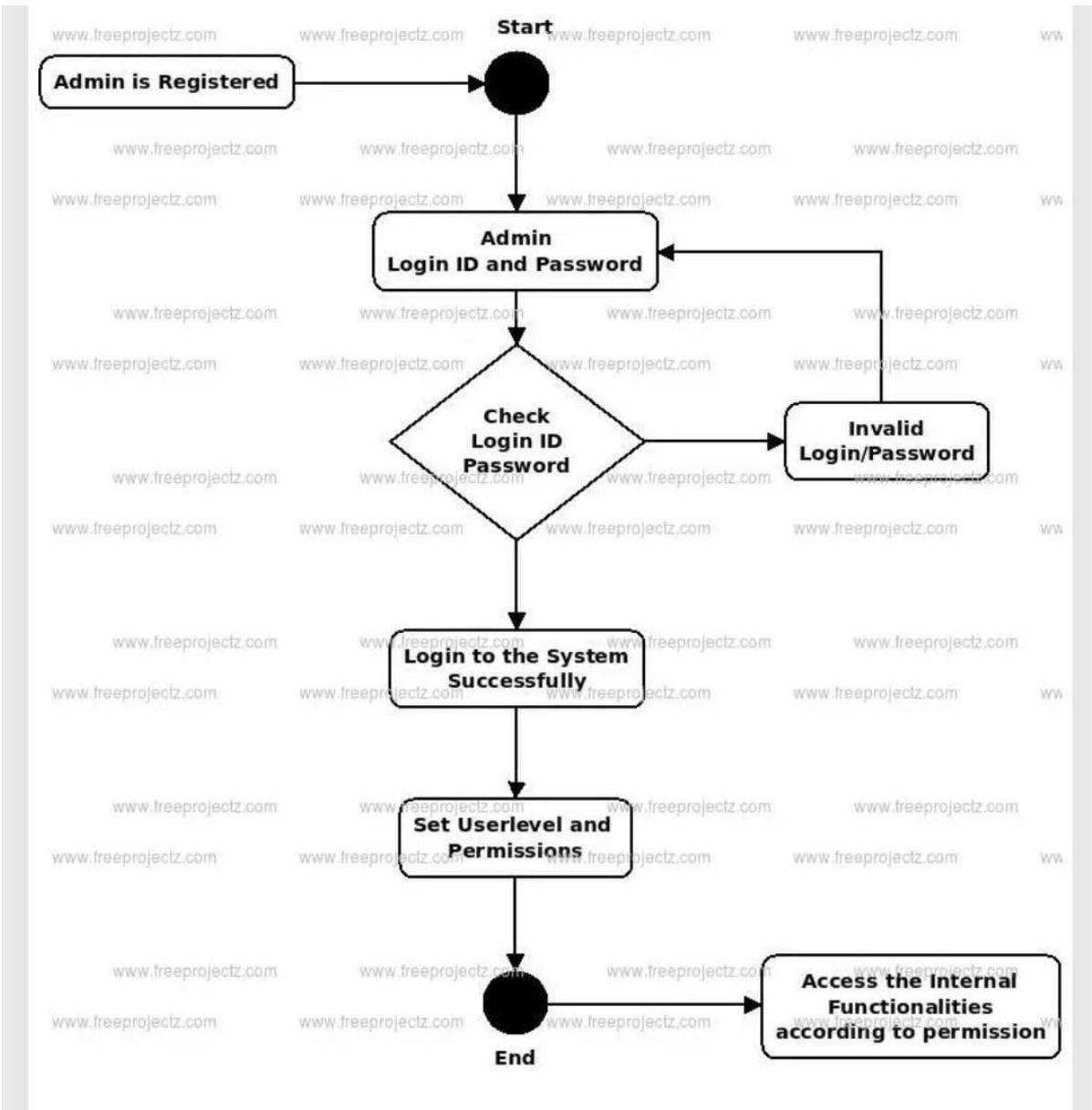
SEQUENCE

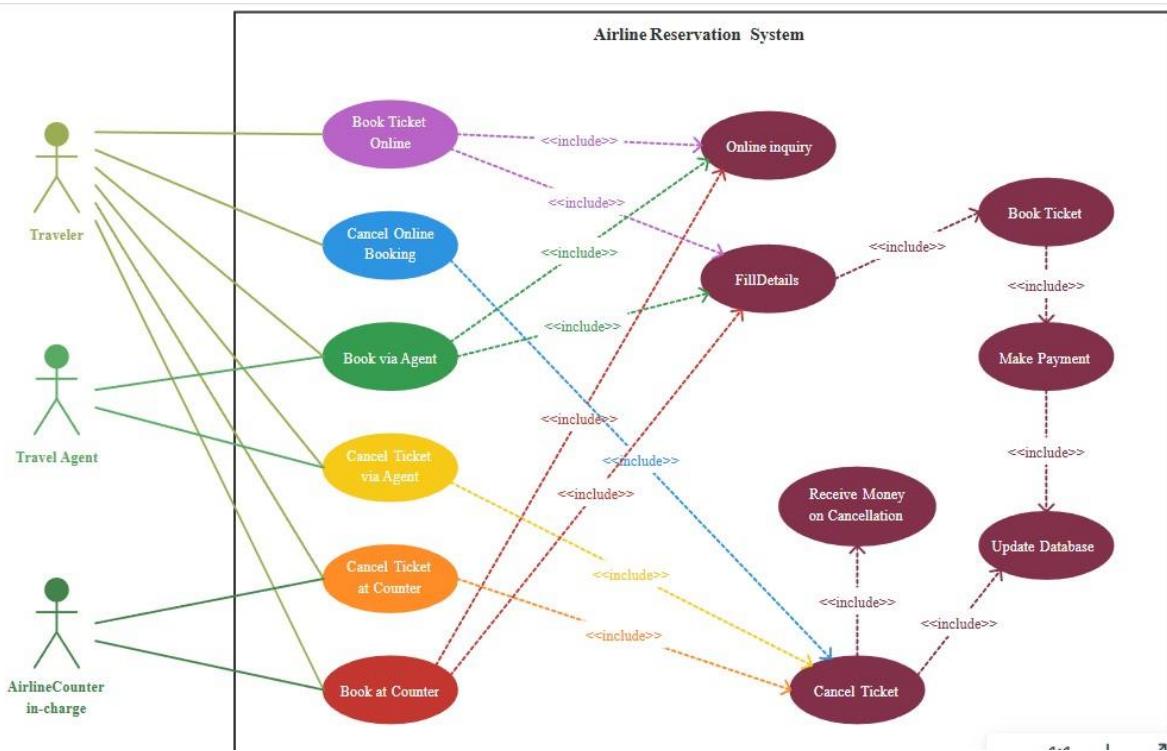


Class diagram :



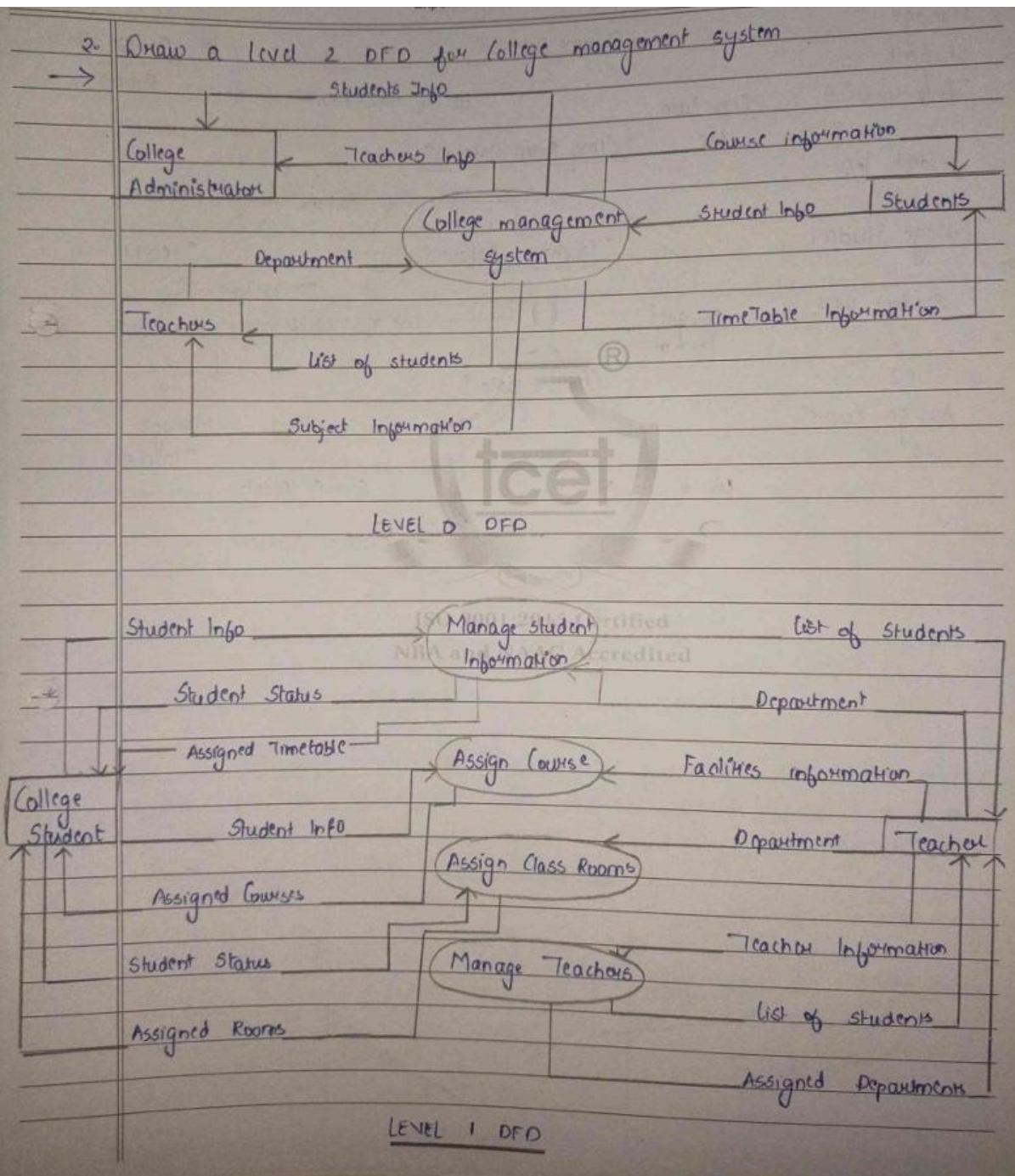
STATE DAIGRAM

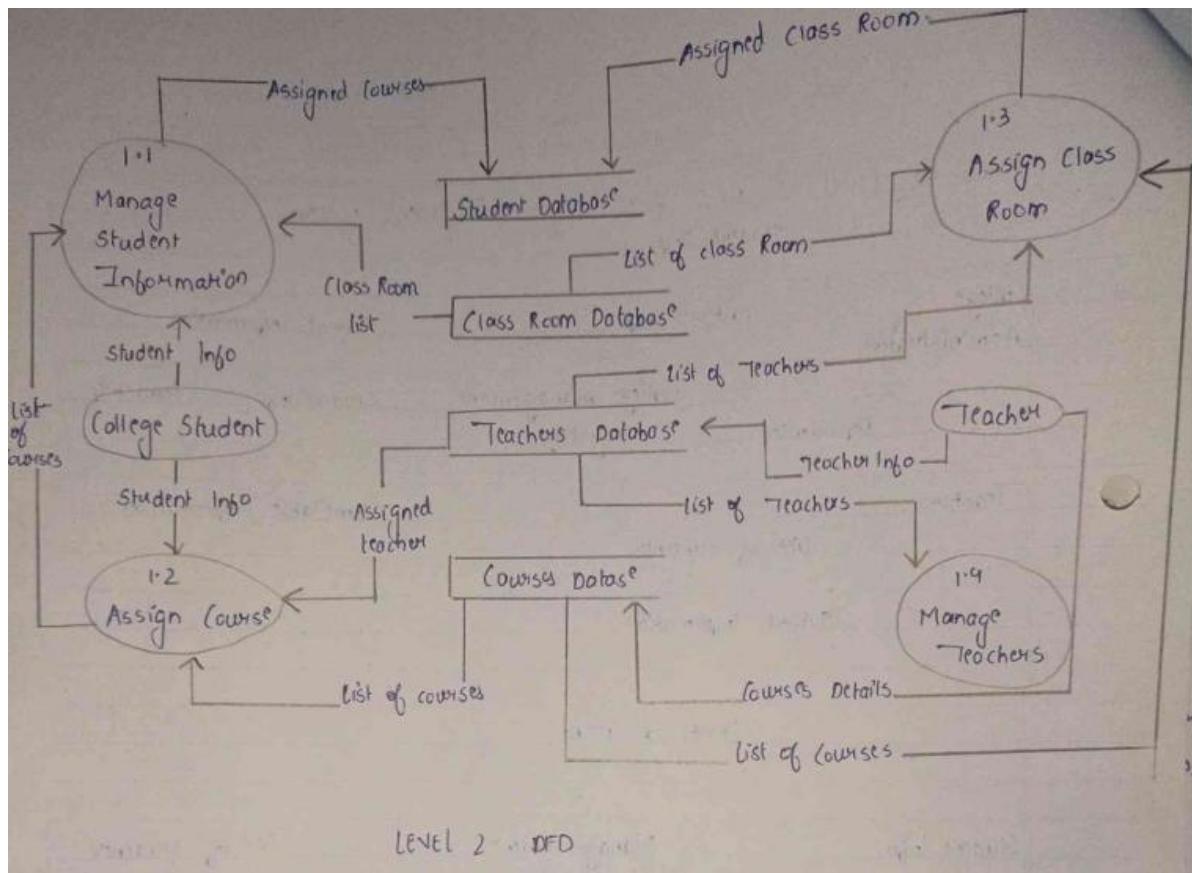




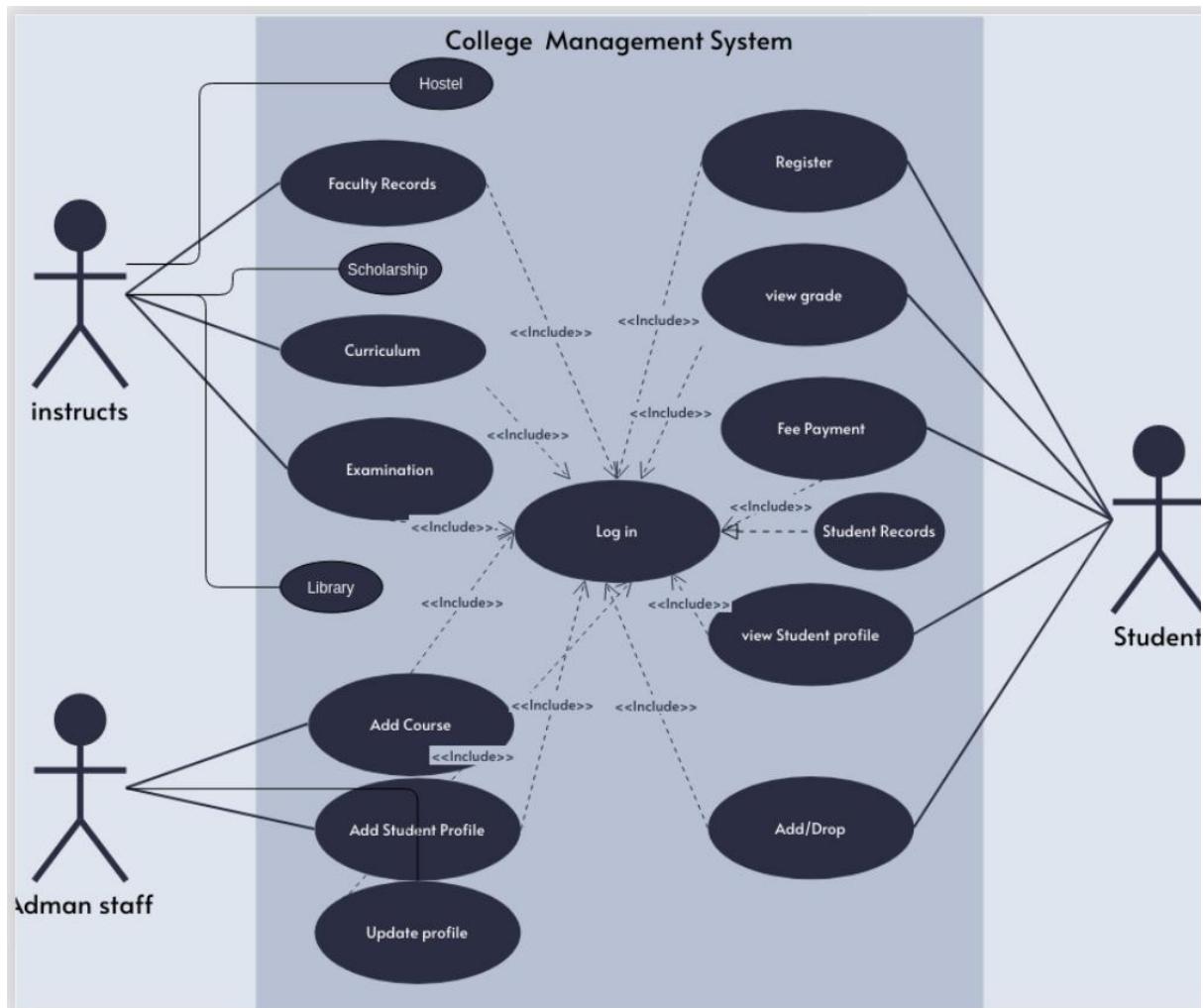
COLLEGE MANAGEMENT SYSTEM

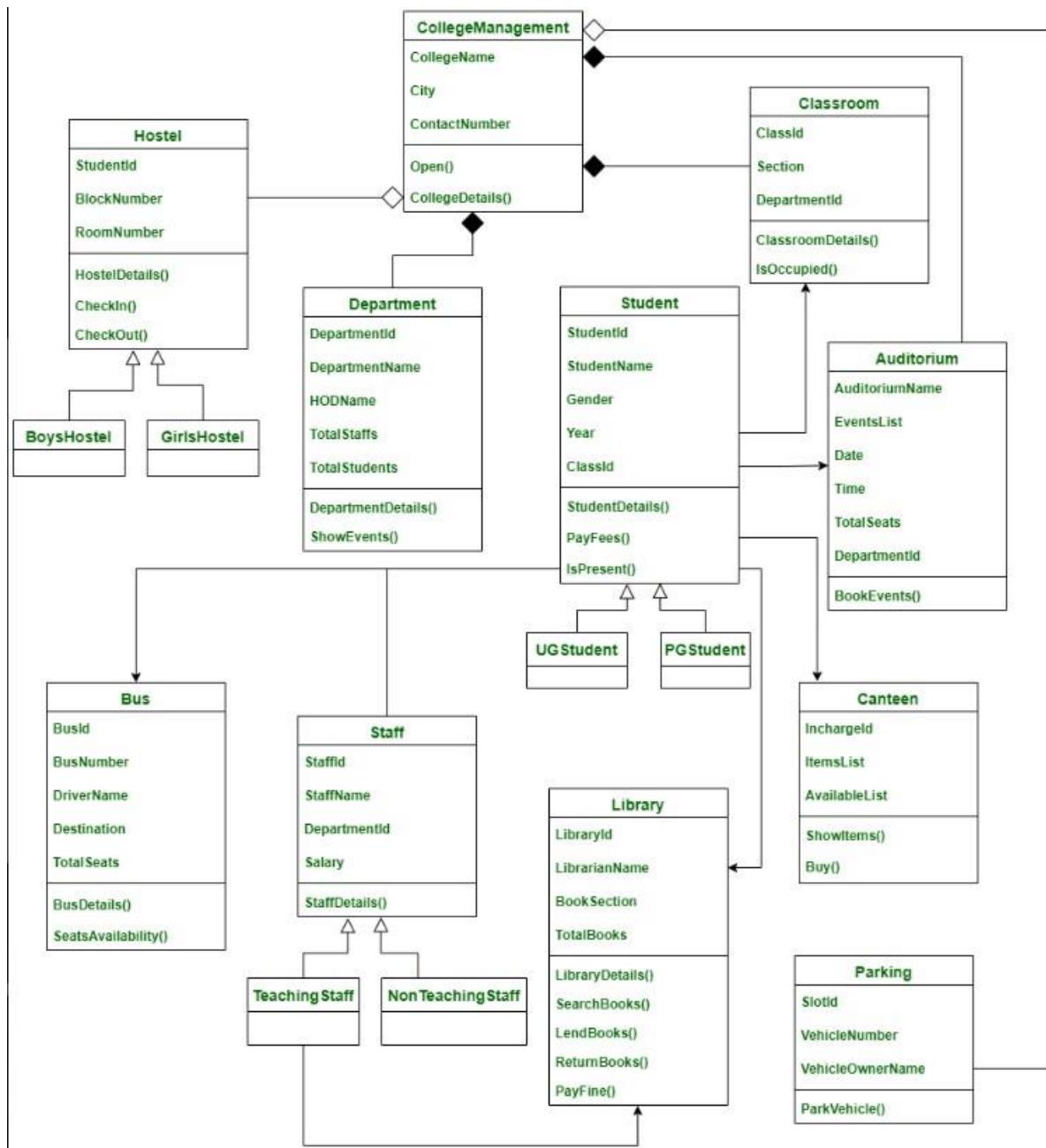
DFD FOR COLLEGE MANAGEMENT SYSTEM

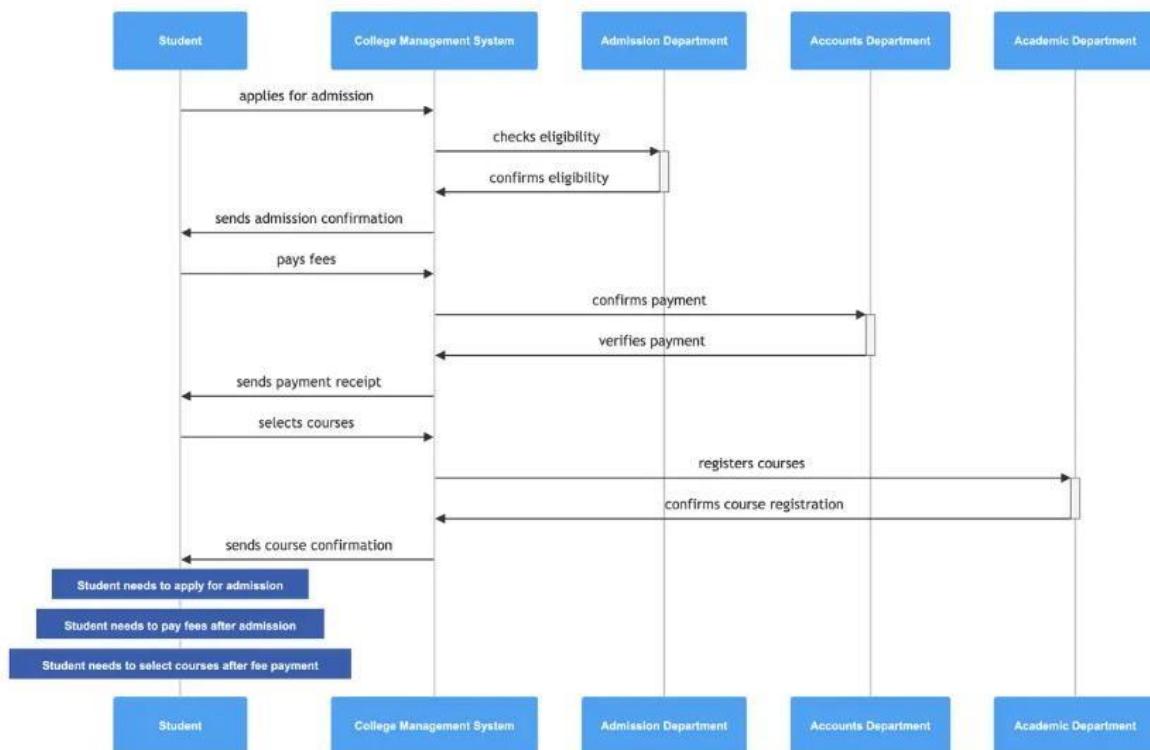




LEVEL 2 DFD

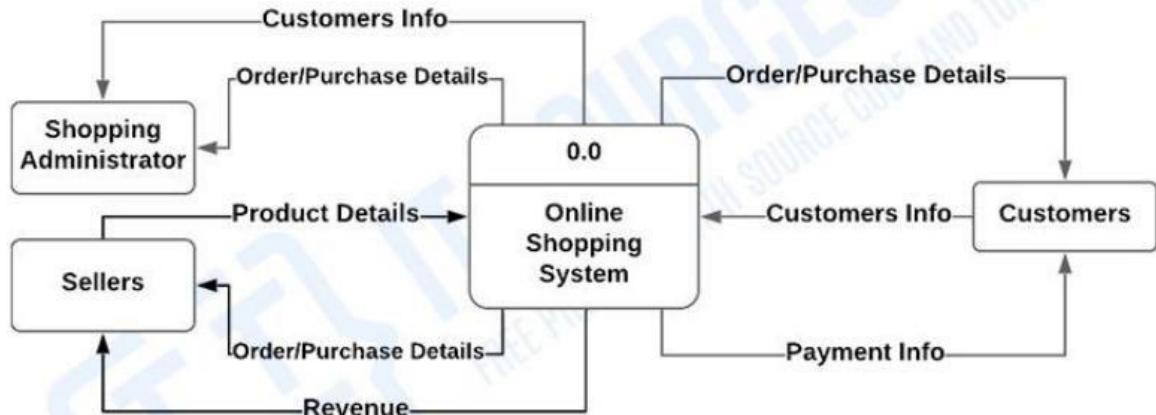




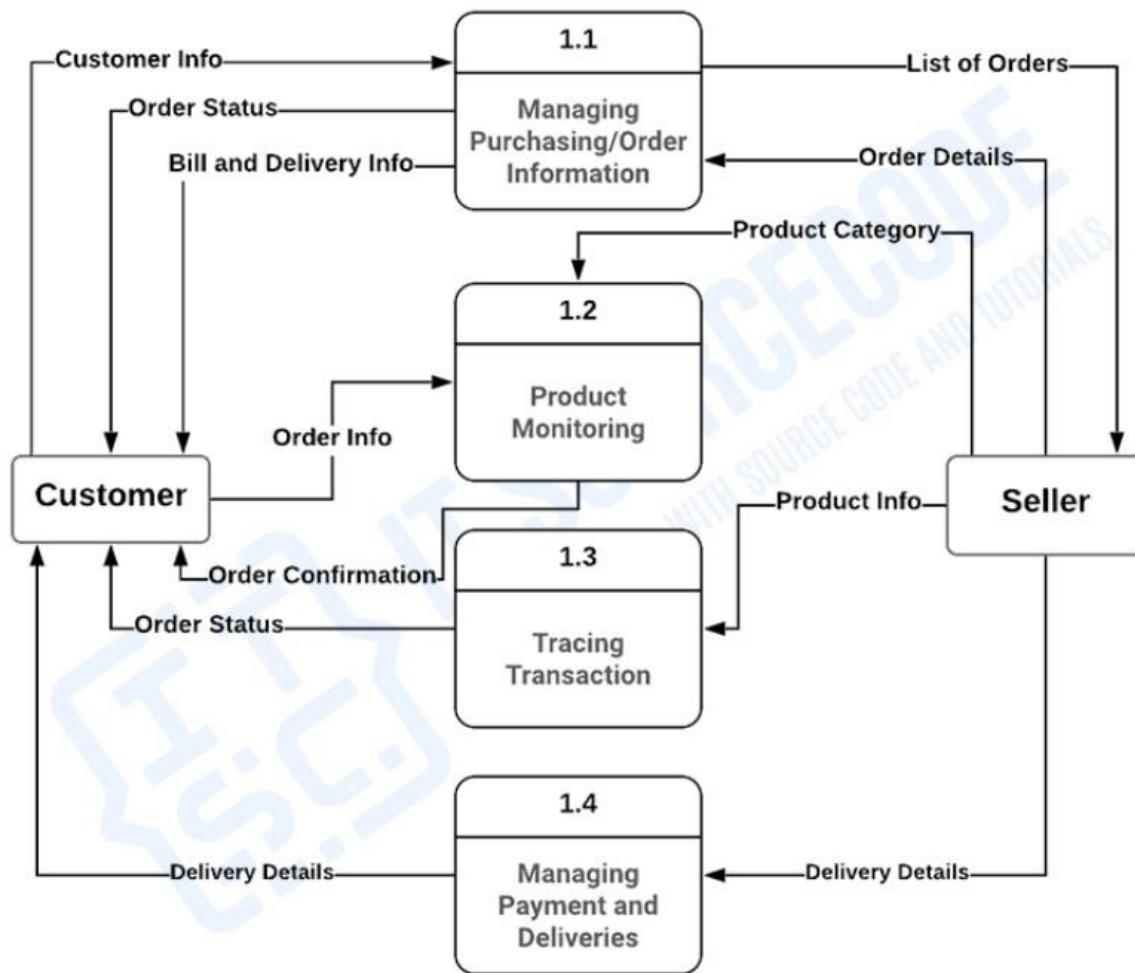


ONLINE SHOPPING SYSTEM

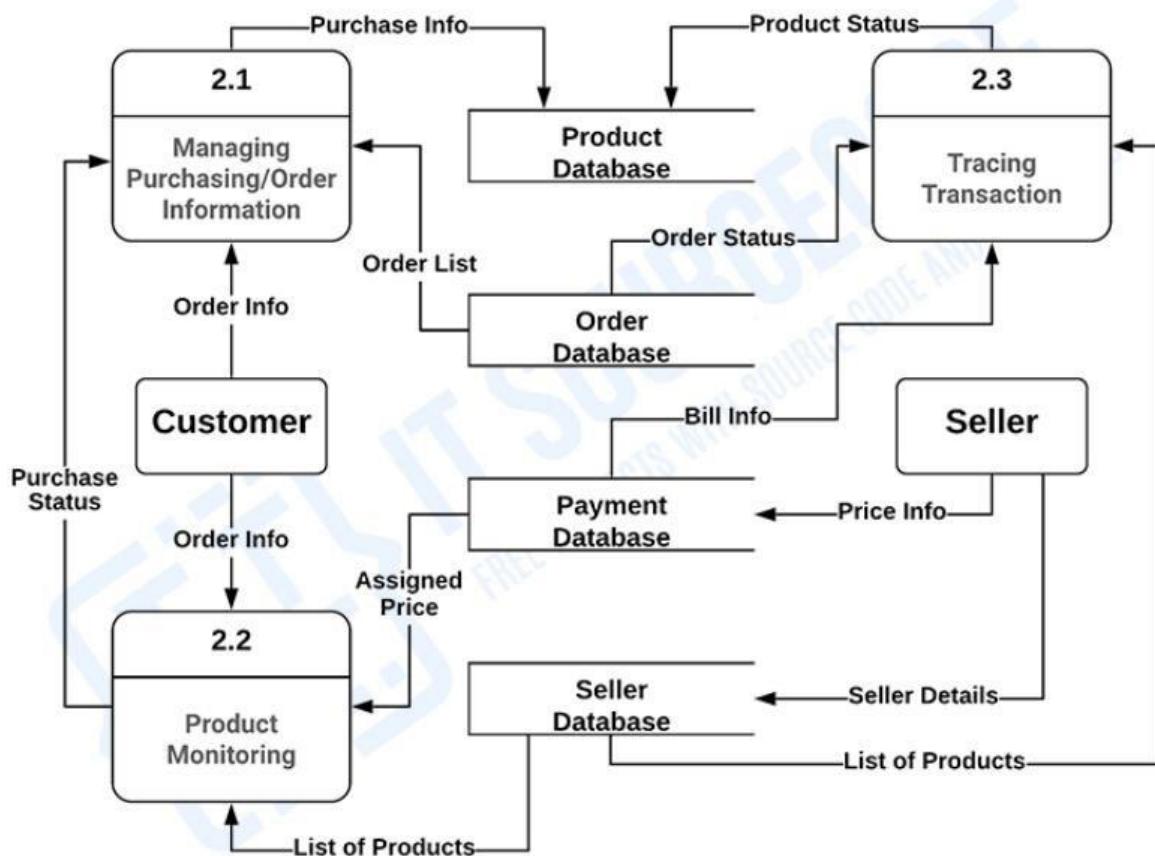
ONLINE SHOPPING SYSTEM DATA FLOW DIAGRAM (DFD)



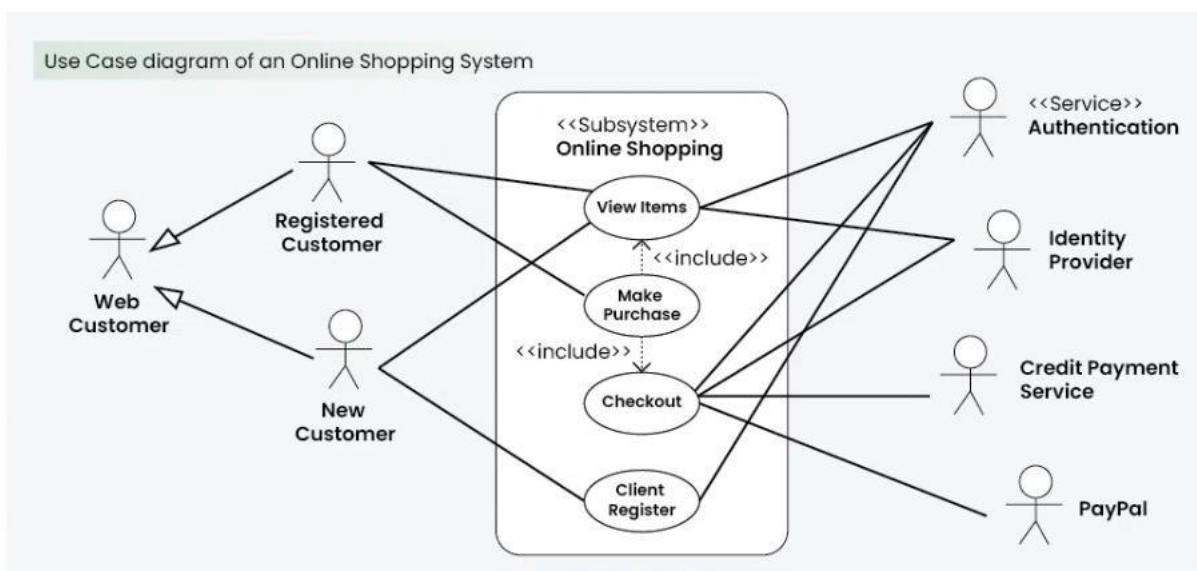
DFD for Online Shopping System Level 0

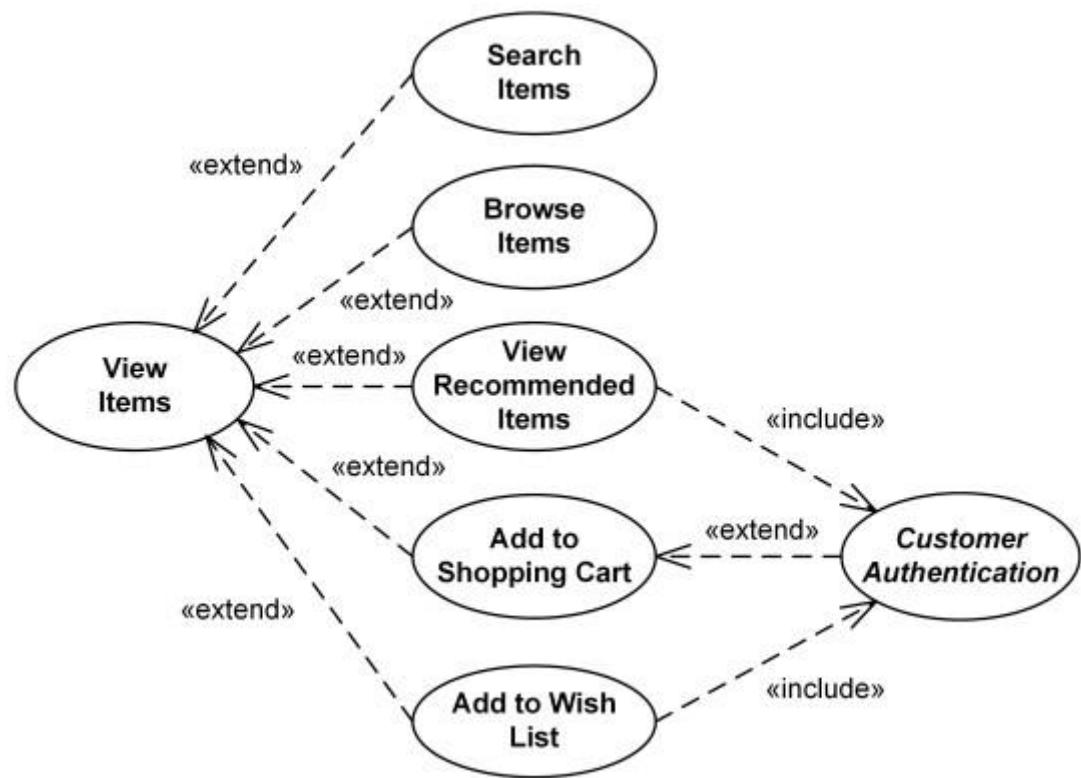


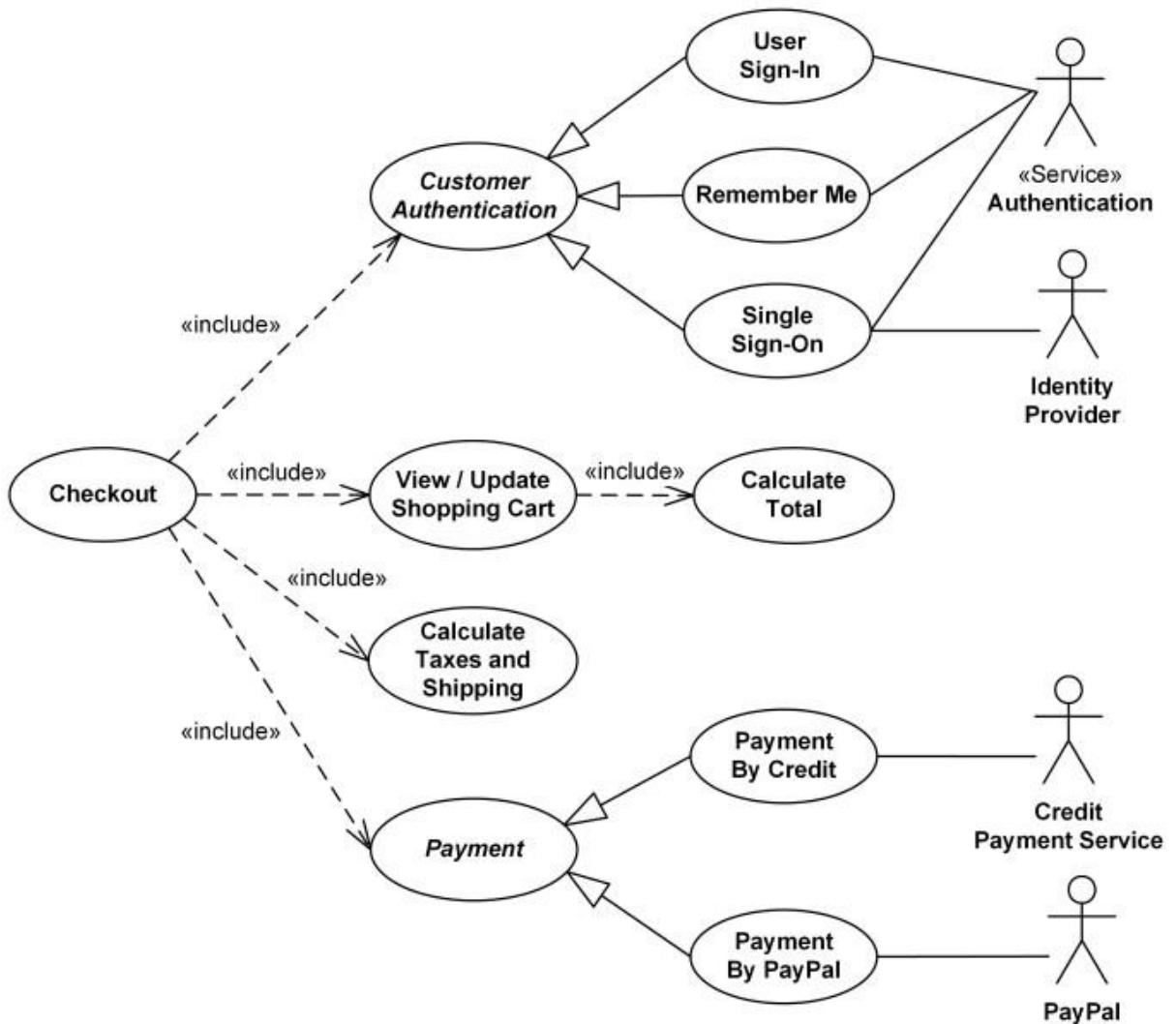
DFD for Online Shopping System Level 1

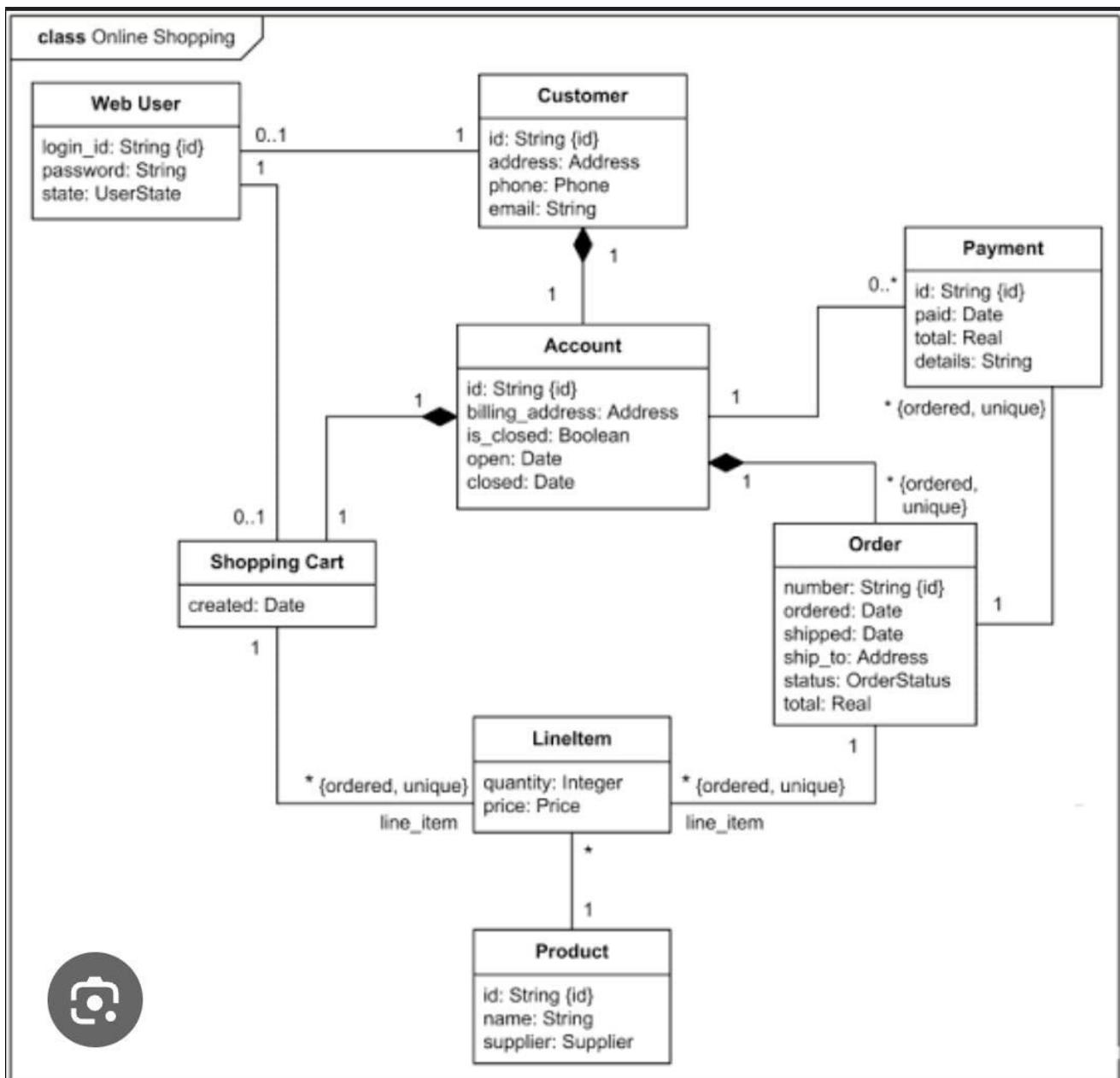


USE CASE DIAGRAM OF AN ONLINE SHOPPING SYSTEM:

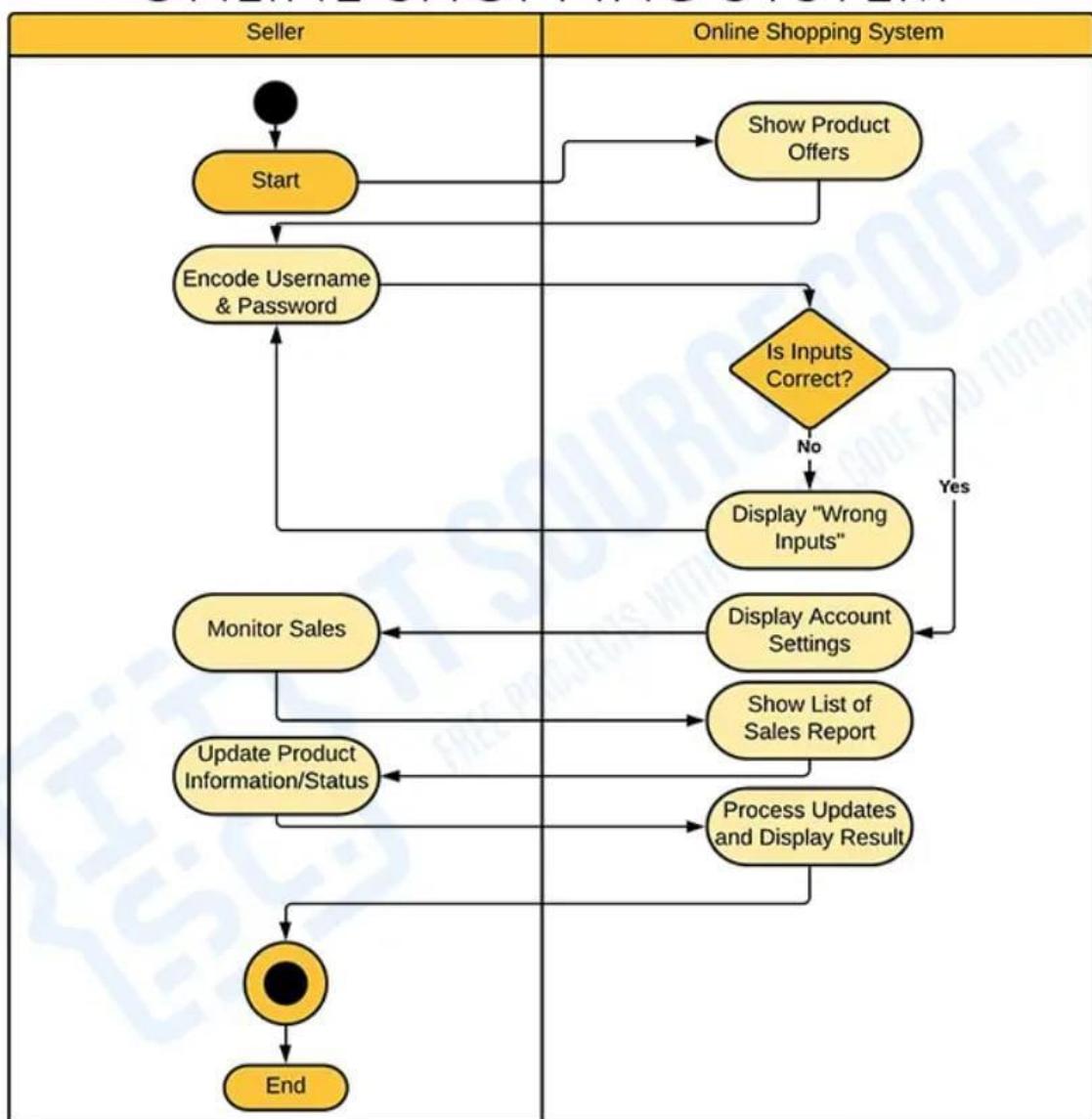








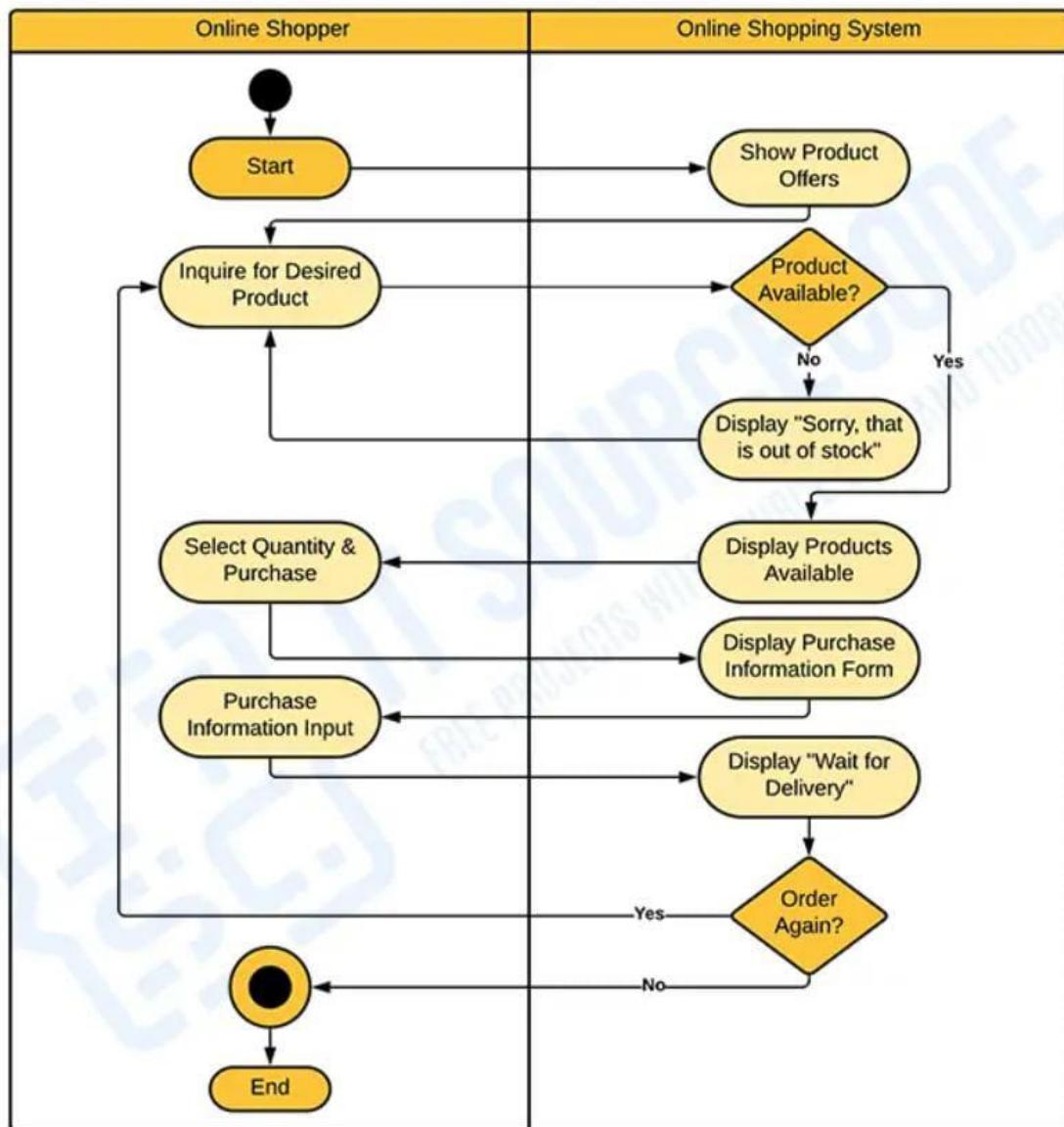
ONLINE SHOPPING SYSTEM



ACTIVITY DIAGRAM

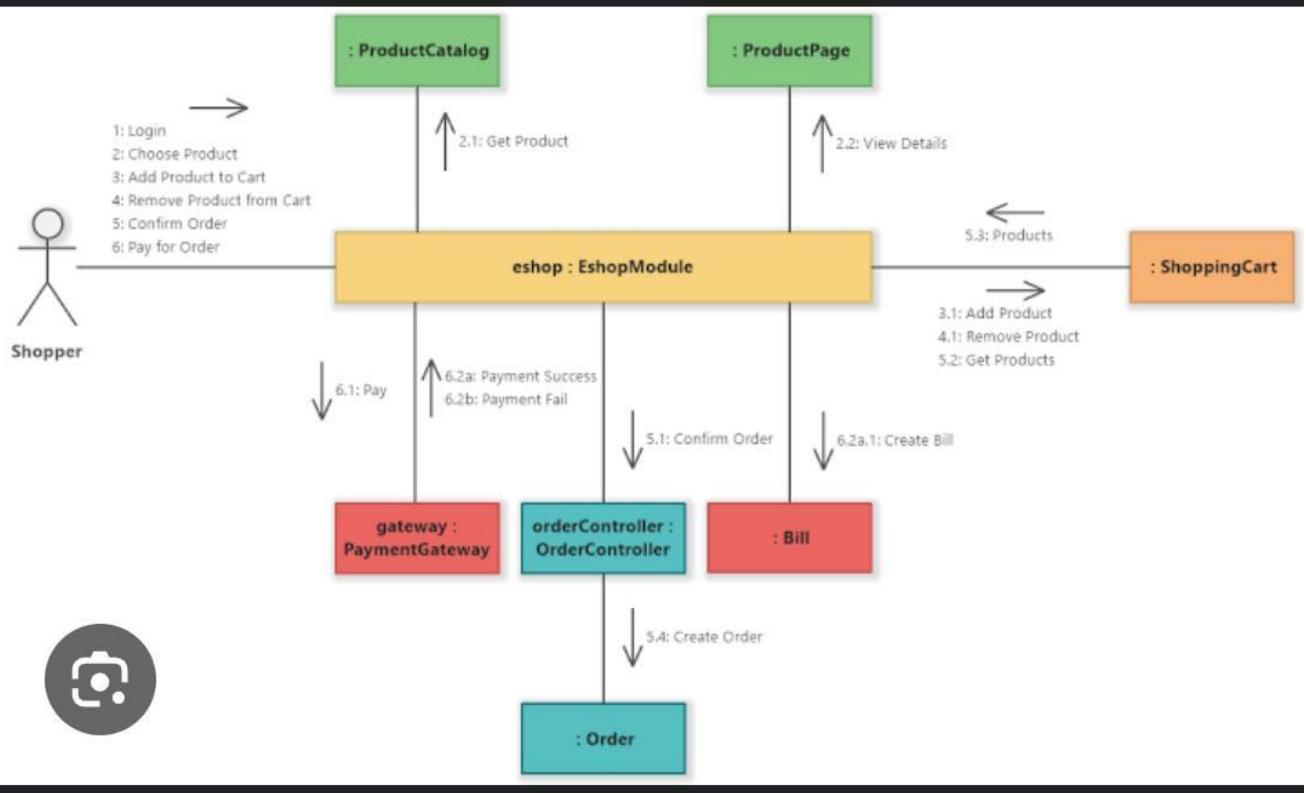
Online Shopping System Activity Diagram (Seller's Side)

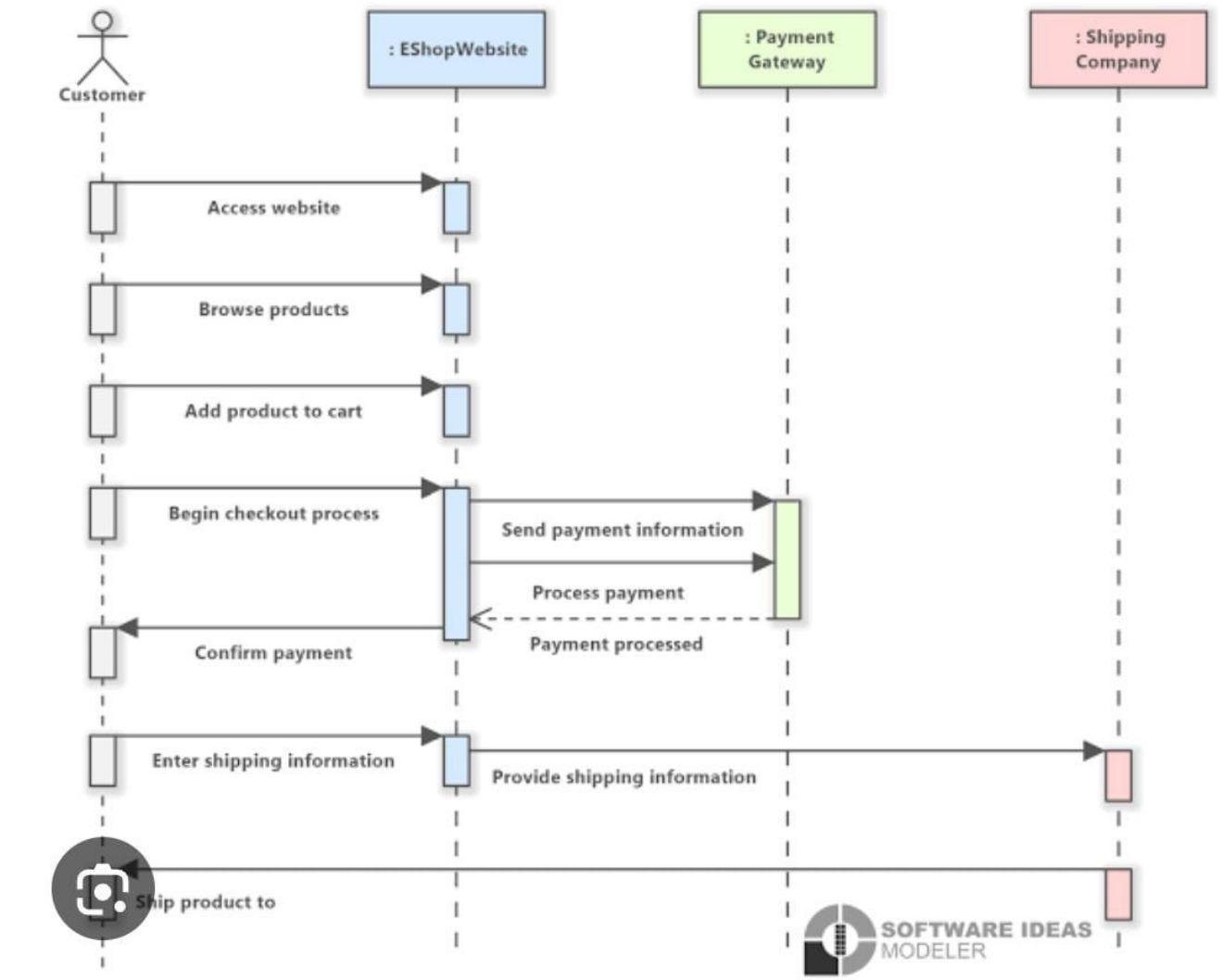
ONLINE SHOPPING SYSTEM



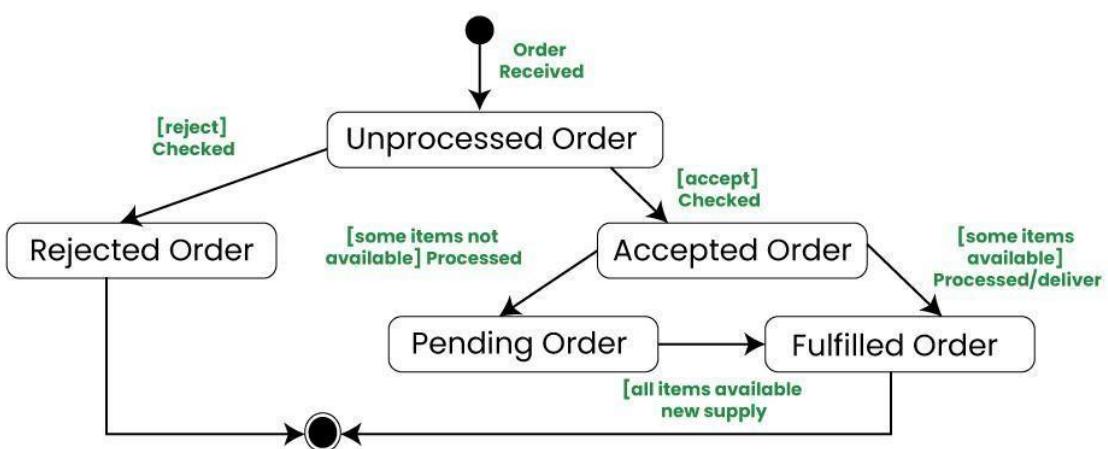
ACTIVITY DIAGRAM

Online Shopping System Activity Diagram (Shoppers' Side)

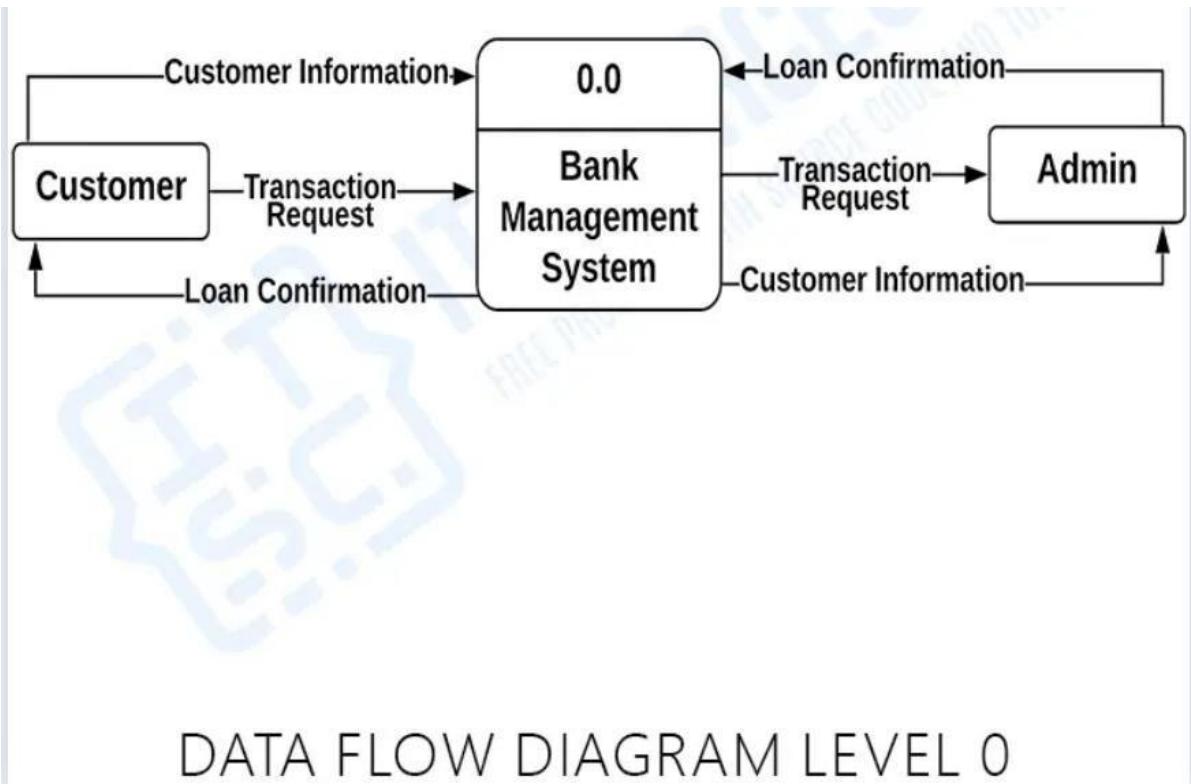




State machine diagram for an online order

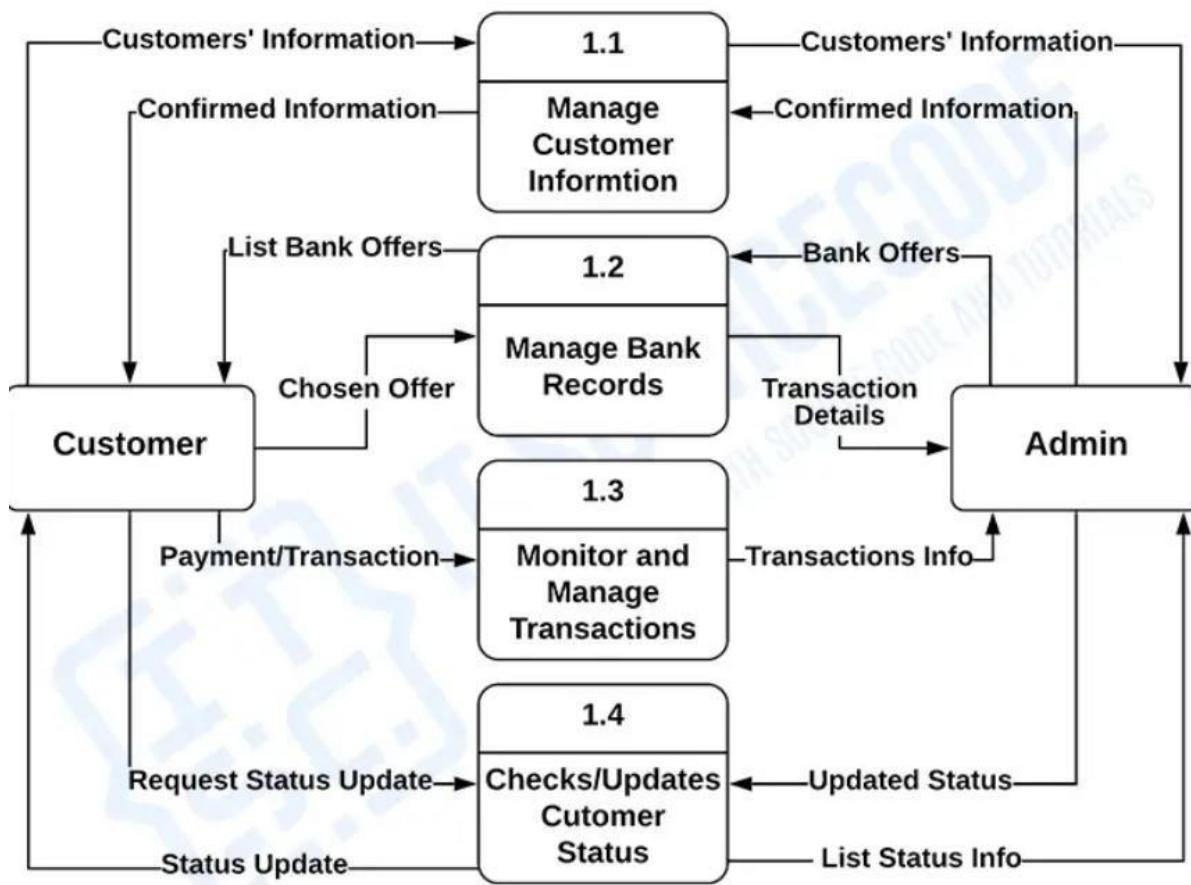


BANK MANAGEMENT SYSTEM DATA FLOW DIAGRAM (DFD)



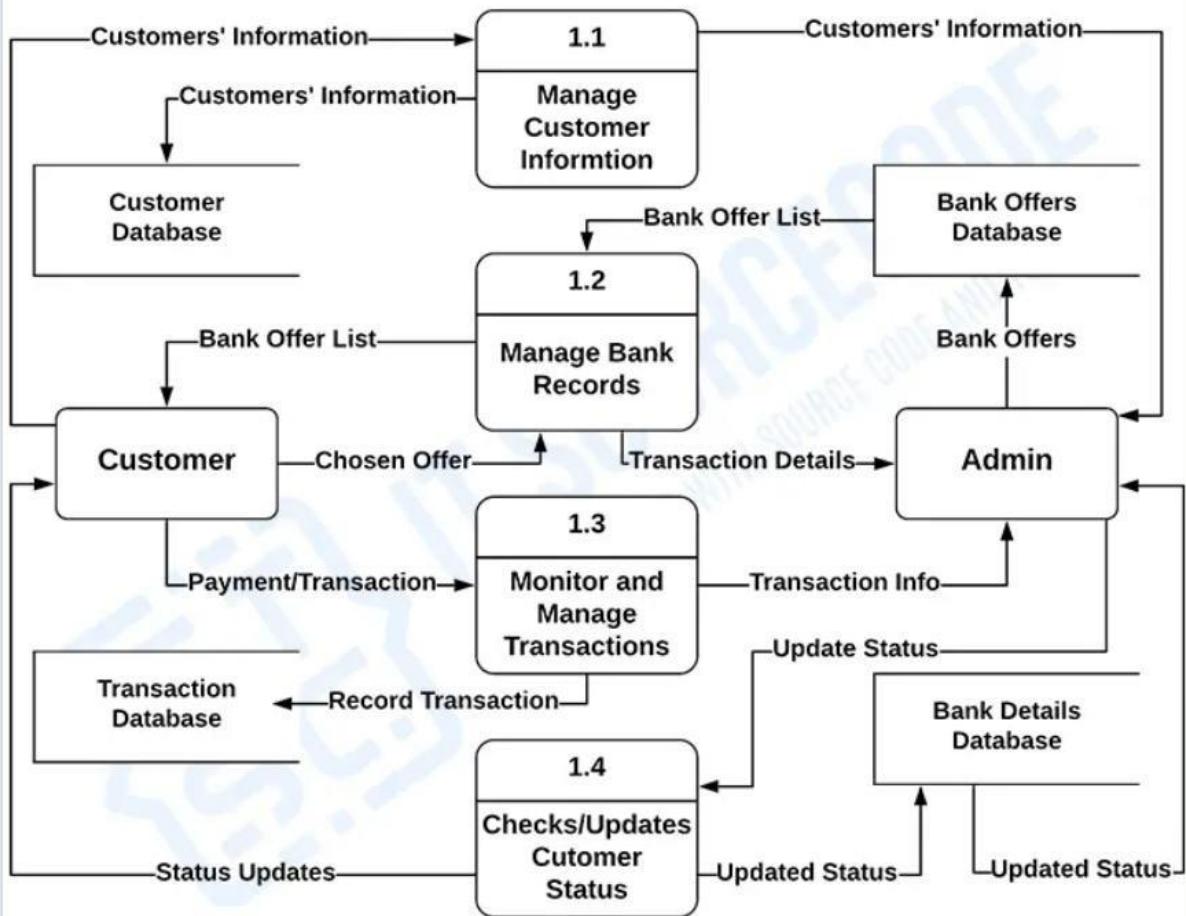
DATA FLOW DIAGRAM LEVEL 0

BANK MANAGEMENT SYSTEM



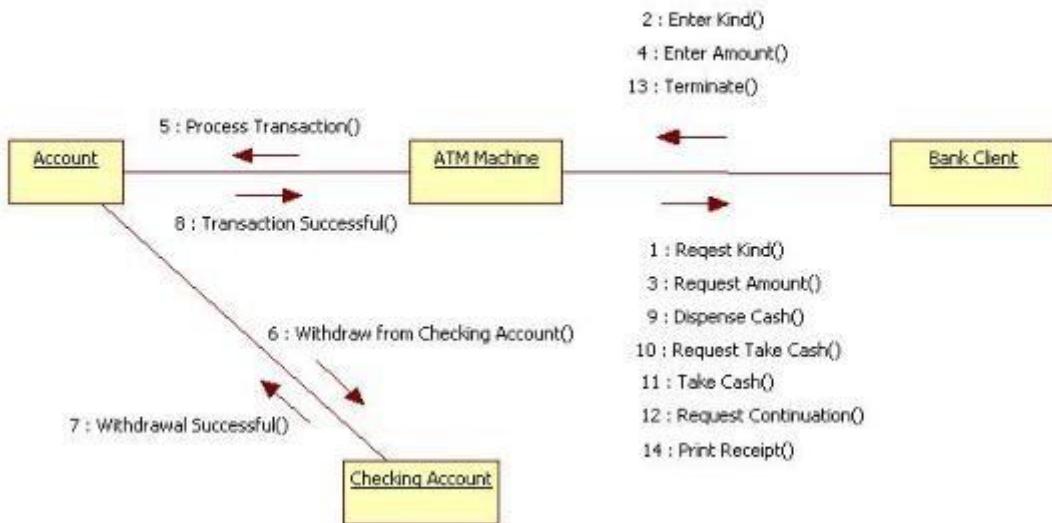
DATA FLOW DIAGRAM LEVEL 1

BANK MANAGEMENT SYSTEM



DATA FLOW DIAGRAM LEVEL 2

COLLABORATION DAIGRAM FOR ATM TRANSCATION



Participants

- **Bank Client** – the person at the ATM
- **ATM Machine** – the UI + device orchestrating the flow
- **Account** – the bank's account/service layer handling transactions
- **Checking Account** – the specific account that gets debited

Message flow (numbers = order)

1. ATM → Client: RequestKind()

ATM asks what you want to do (withdrawal, balance, etc.).

2. Client → ATM: EnterKind()

User chooses Withdraw from Checking.

3. ATM → Client: RequestAmount()

ATM prompts for the amount.

4. Client → ATM: EnterAmount()

User enters, say, \$100.

5. ATM → Account: ProcessTransaction()

ATM hands the request to the bank backend to process the withdrawal

(includes rules/limits).

6. Account → Checking Account: Withdraw from CheckingAccount() Backend tells the specific checking account to debit the funds.

7. Checking Account → Account: WithdrawSuccessful()

Debit completes (or is reserved); success returned to the account service.

8. Account → ATM: TransactionSuccessful()

Backend confirms the withdrawal to the ATM.

9. ATM → Client: DispenseCash() The machine dispenses the cash.

10. ATM → Client: RequestTakeCash() Prompt to take the cash.

11. Client → ATM: TakeCash() User removes the cash.

12. ATM → Client: RequestContinuation()

“Do you want another transaction?”

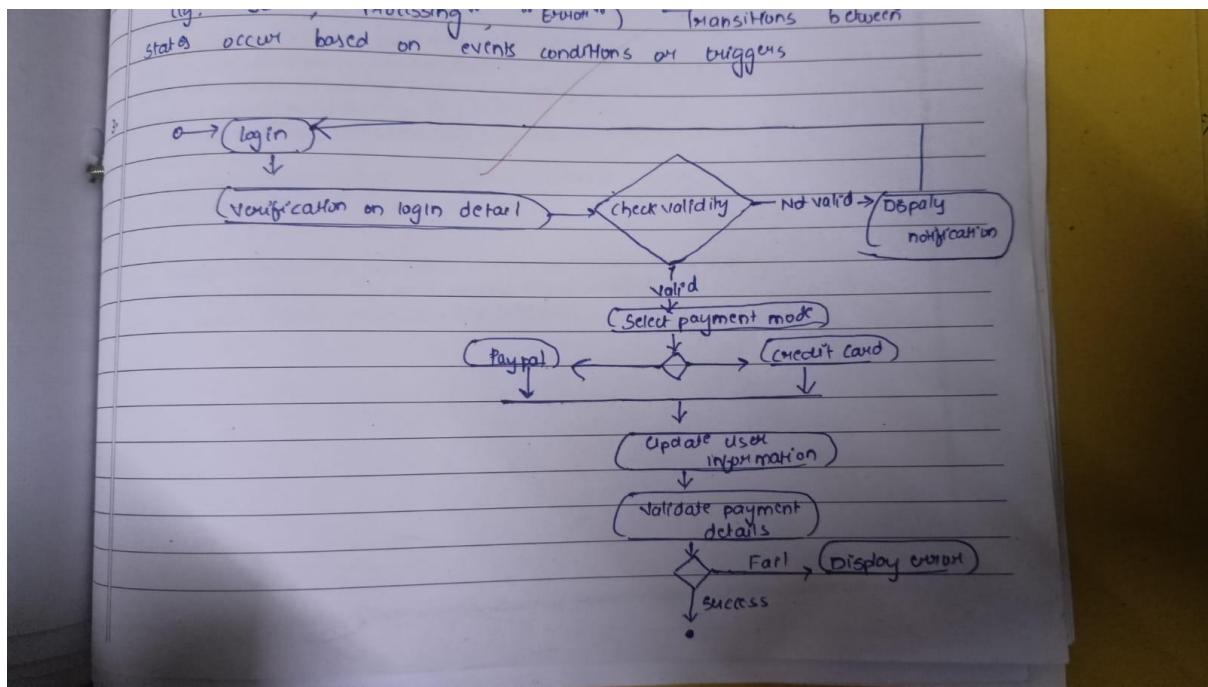
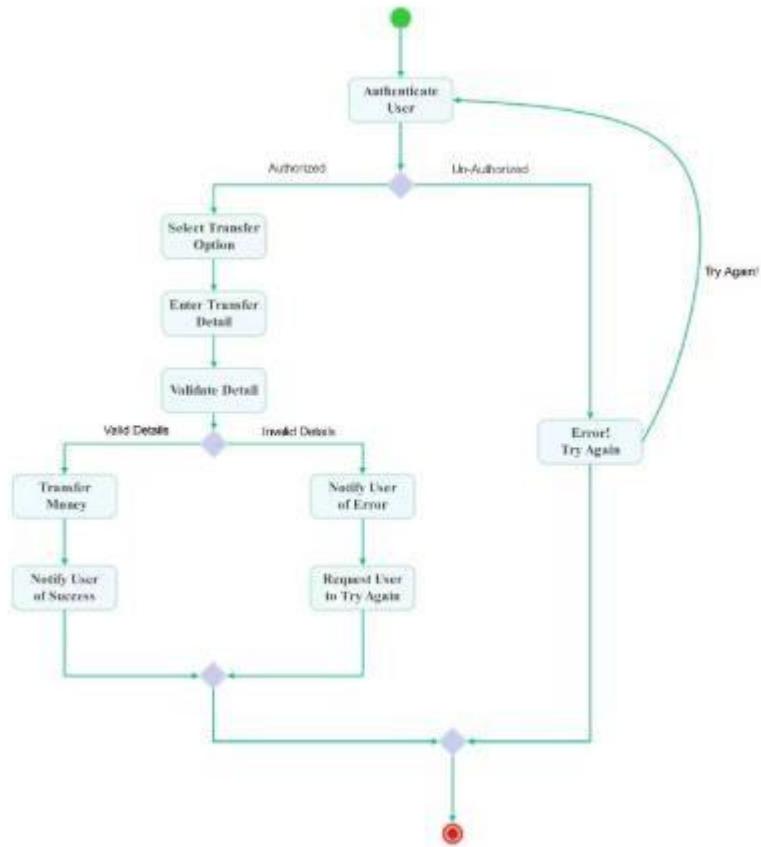
13. Client → ATM: Terminate()

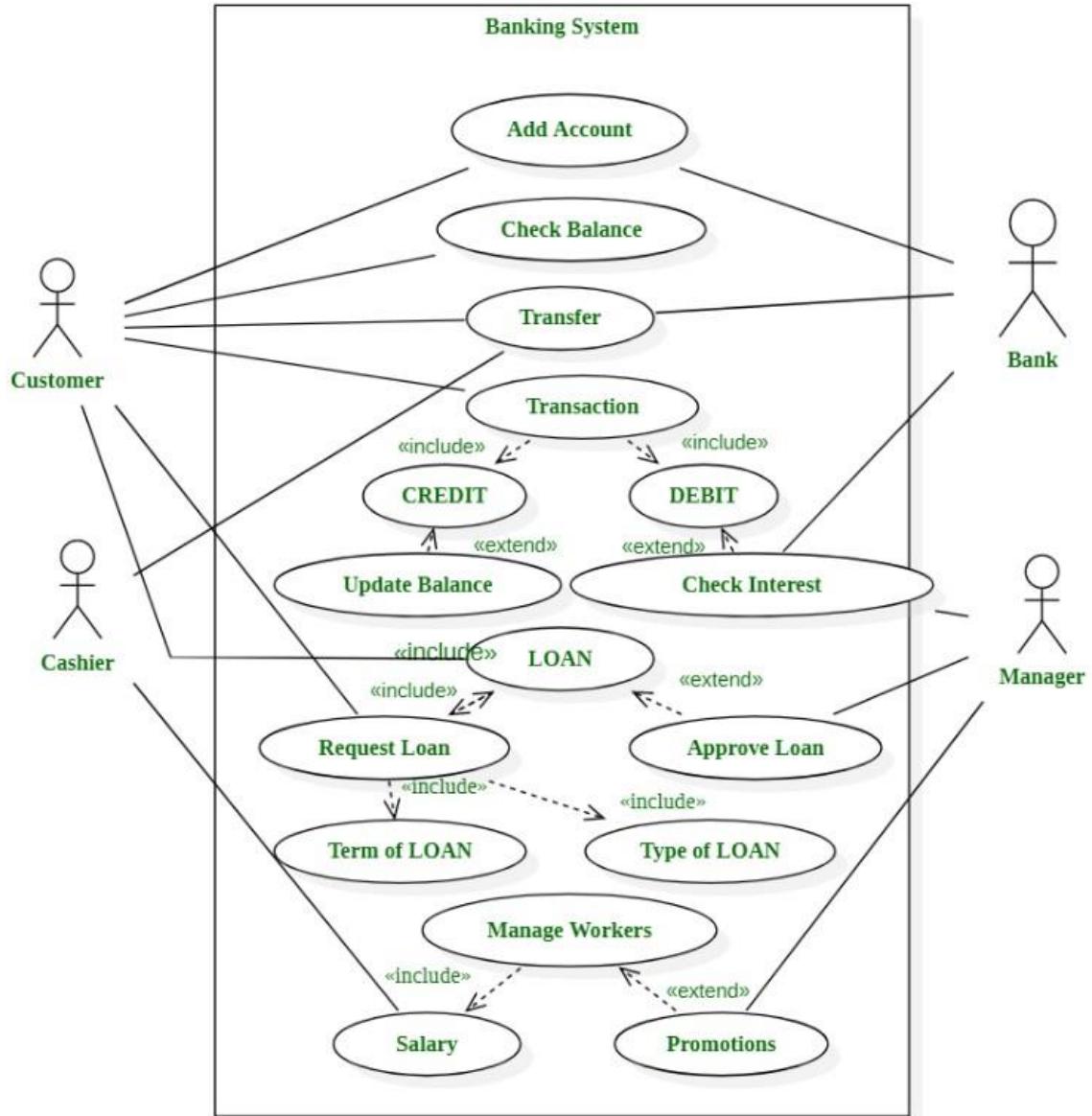
User selects No / end session.

14. ATM → Client: PrintReceipt()

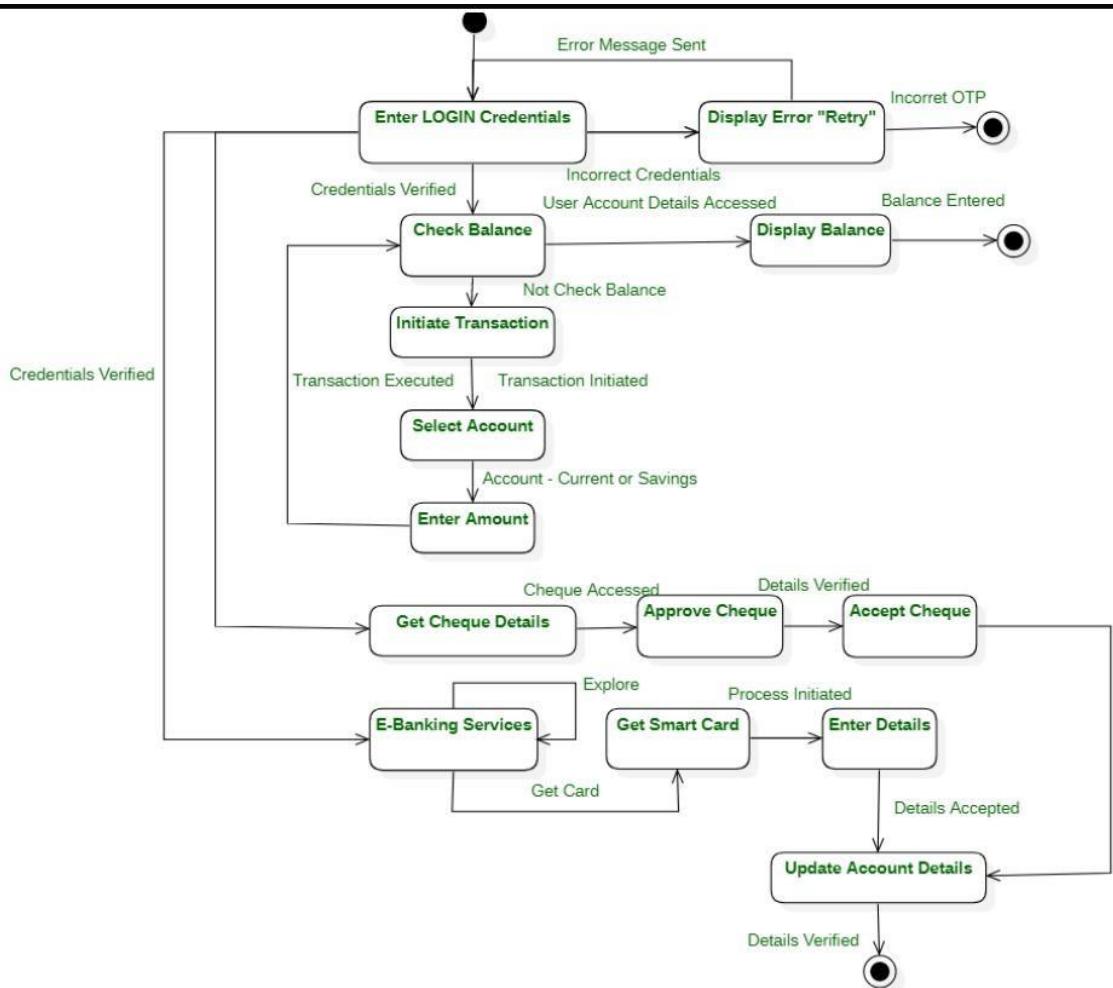
Receipt is printed and the session ends.

ACTIVITY DAIGRAM FOR ONLINE PAYMENT SYSTEM

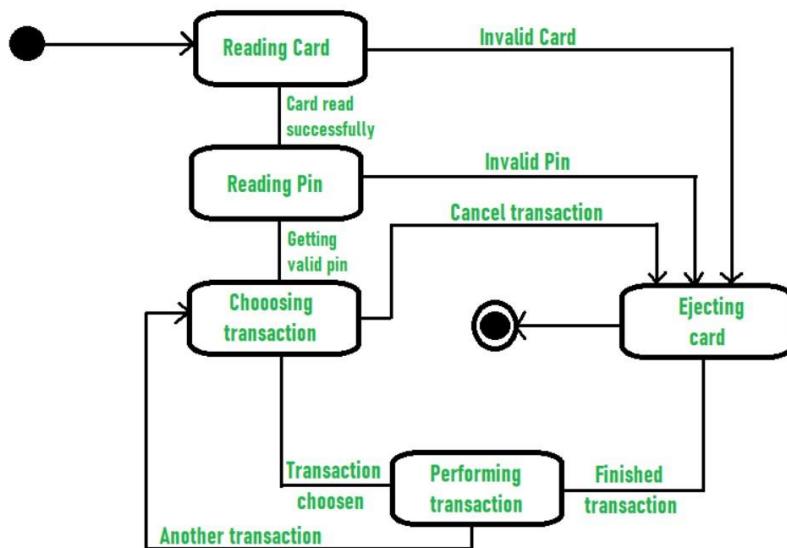




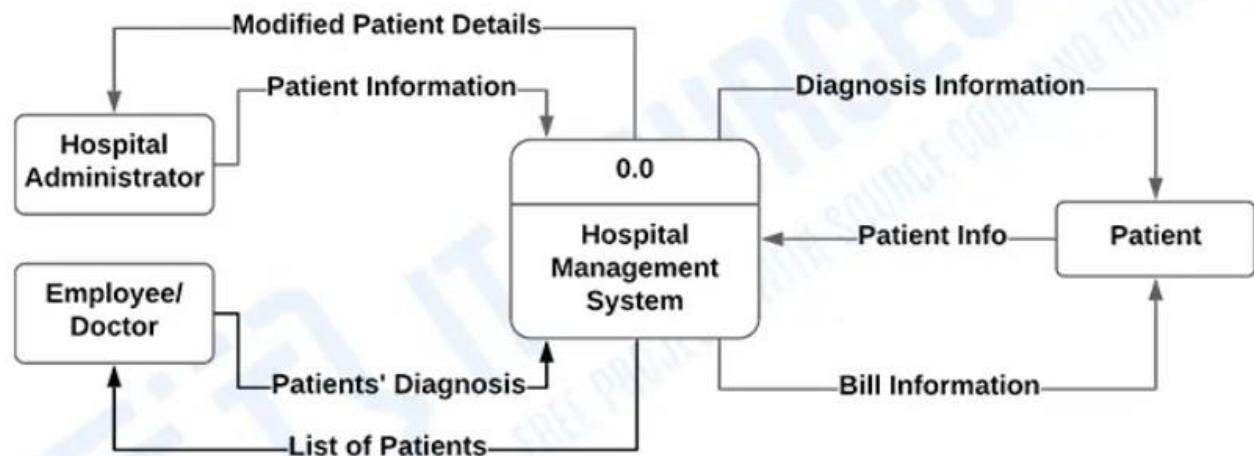
State Diagram for Online Banking System

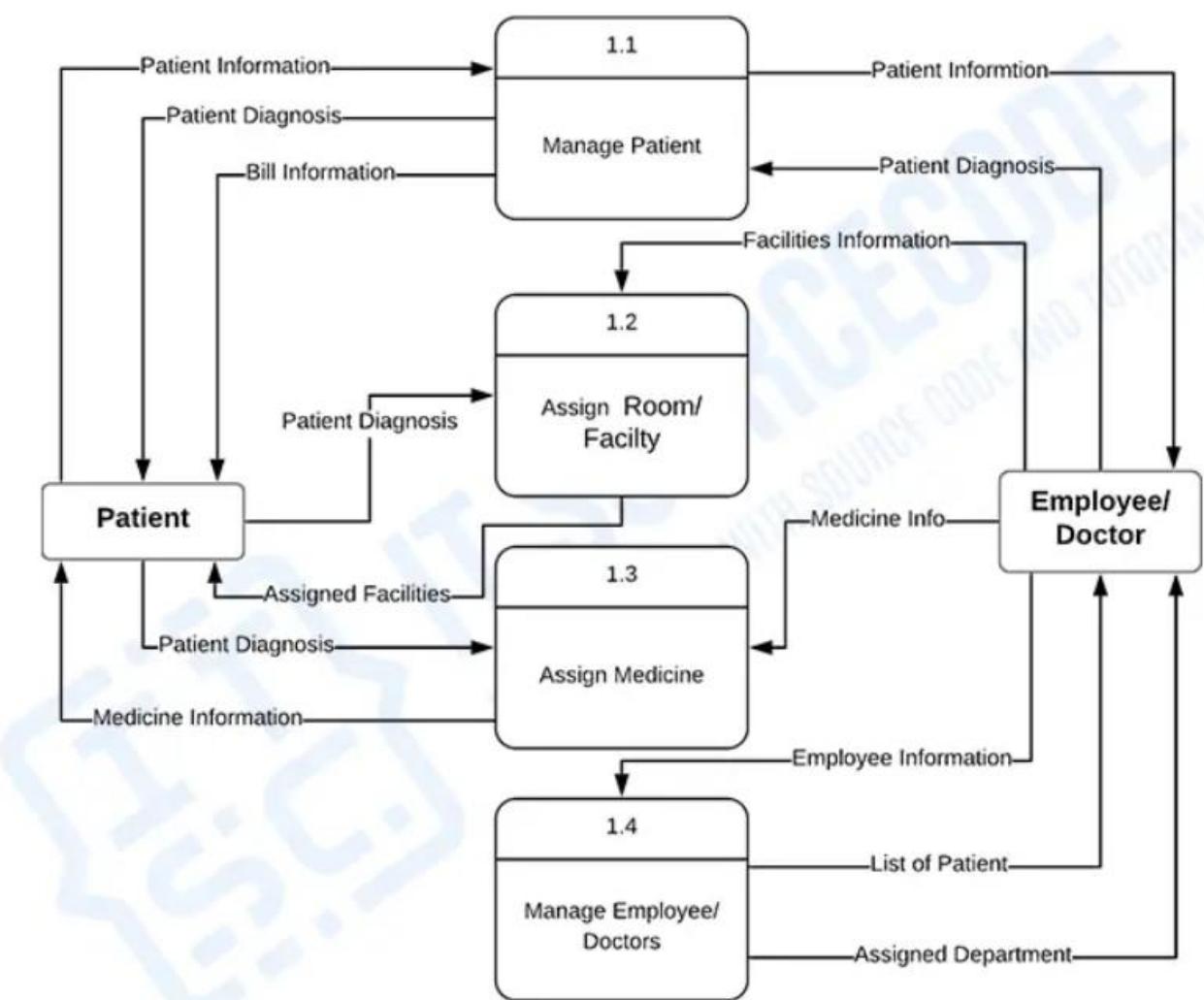


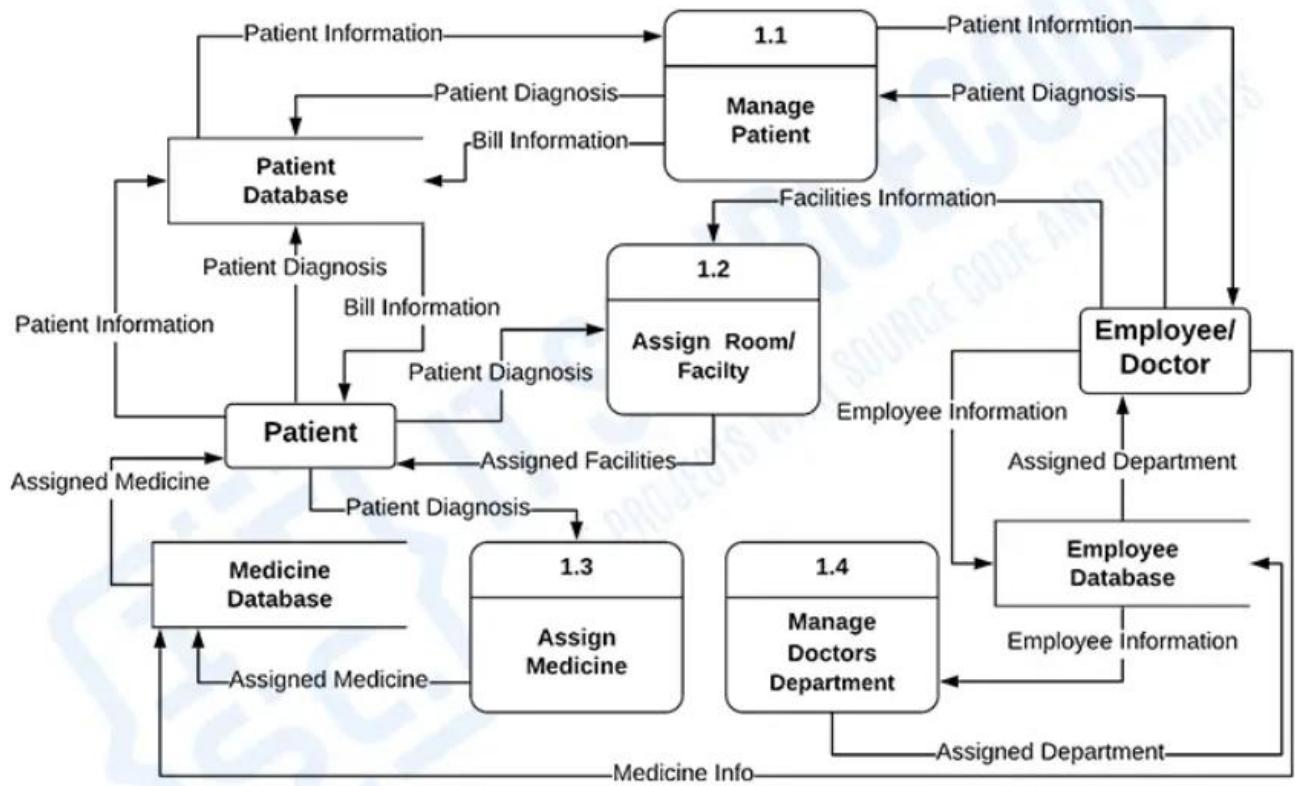
State Transition Diagram for an ATM System

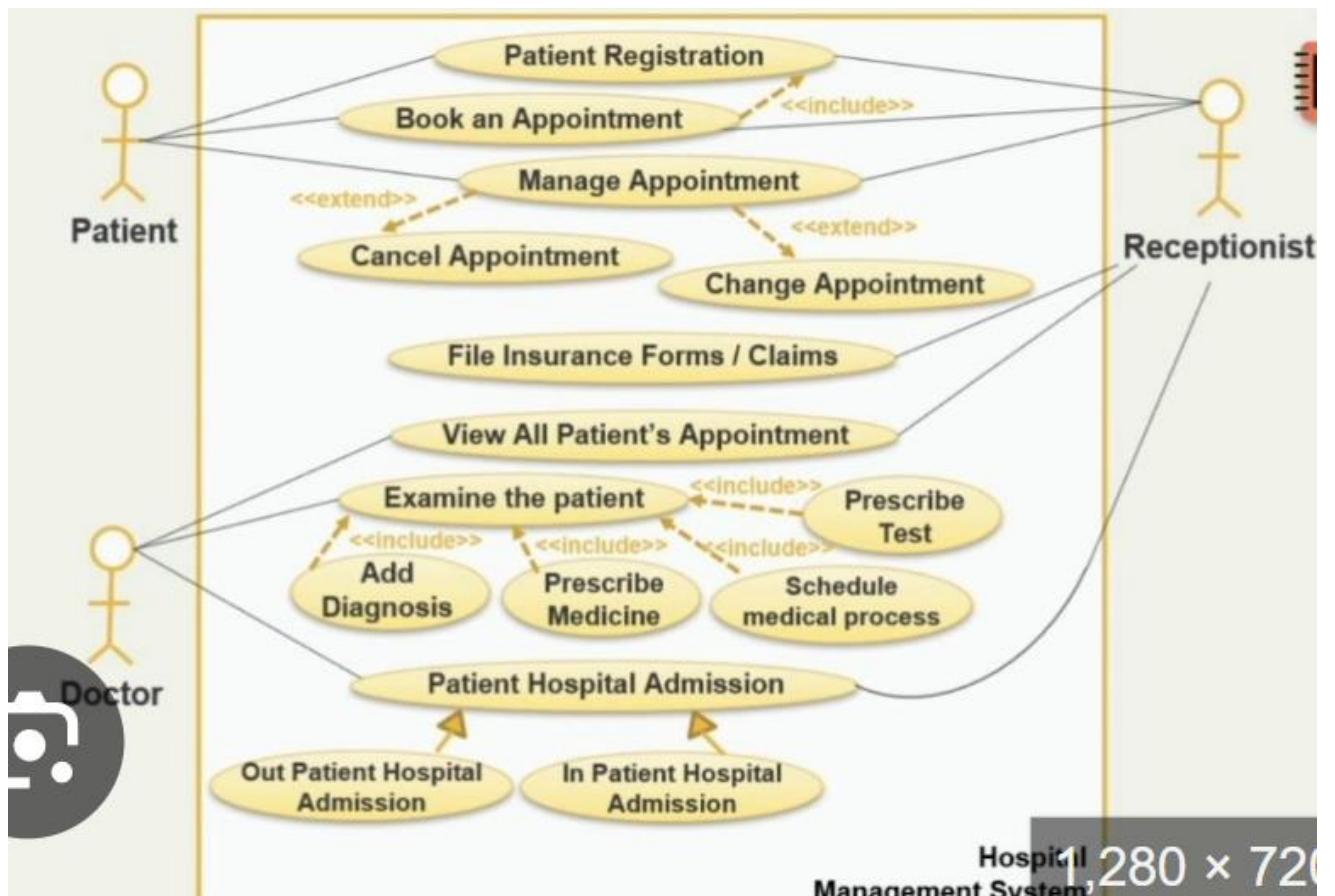


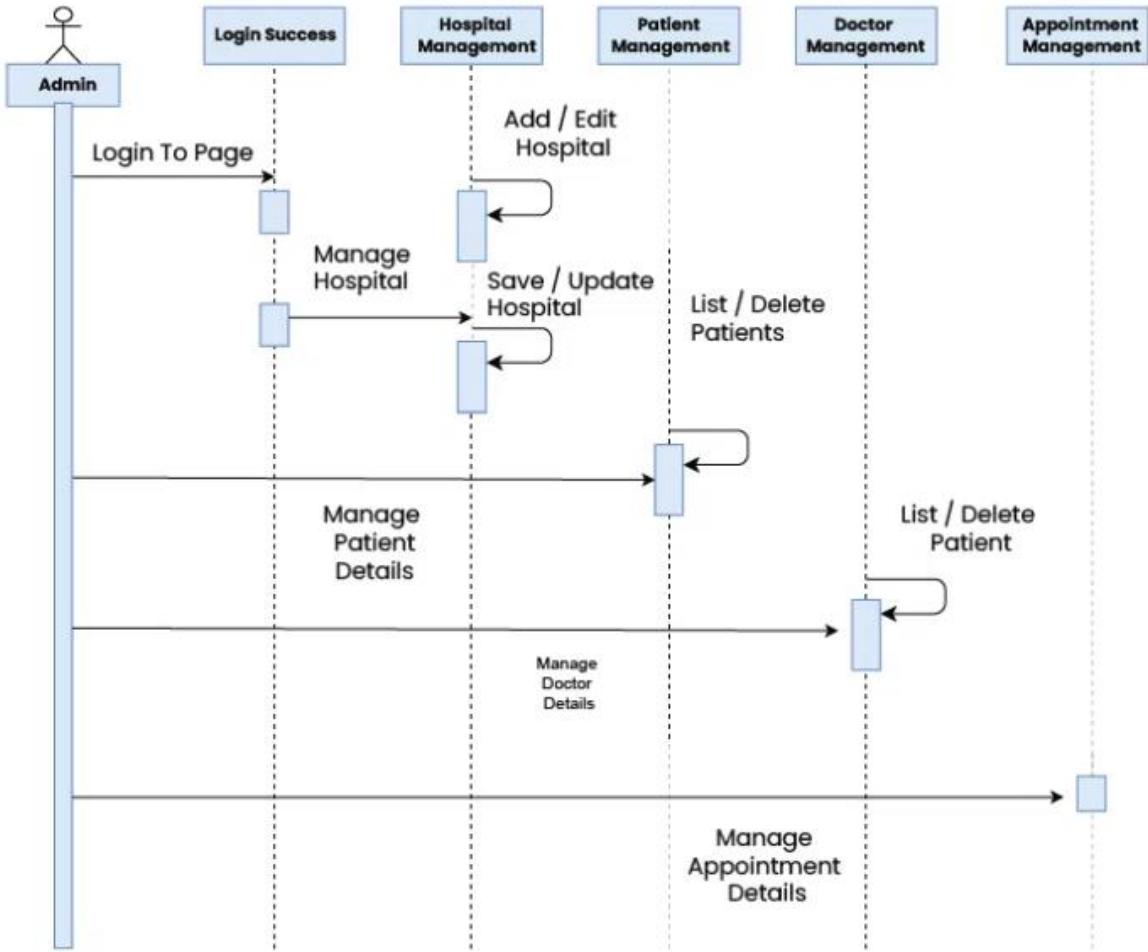
Medical management





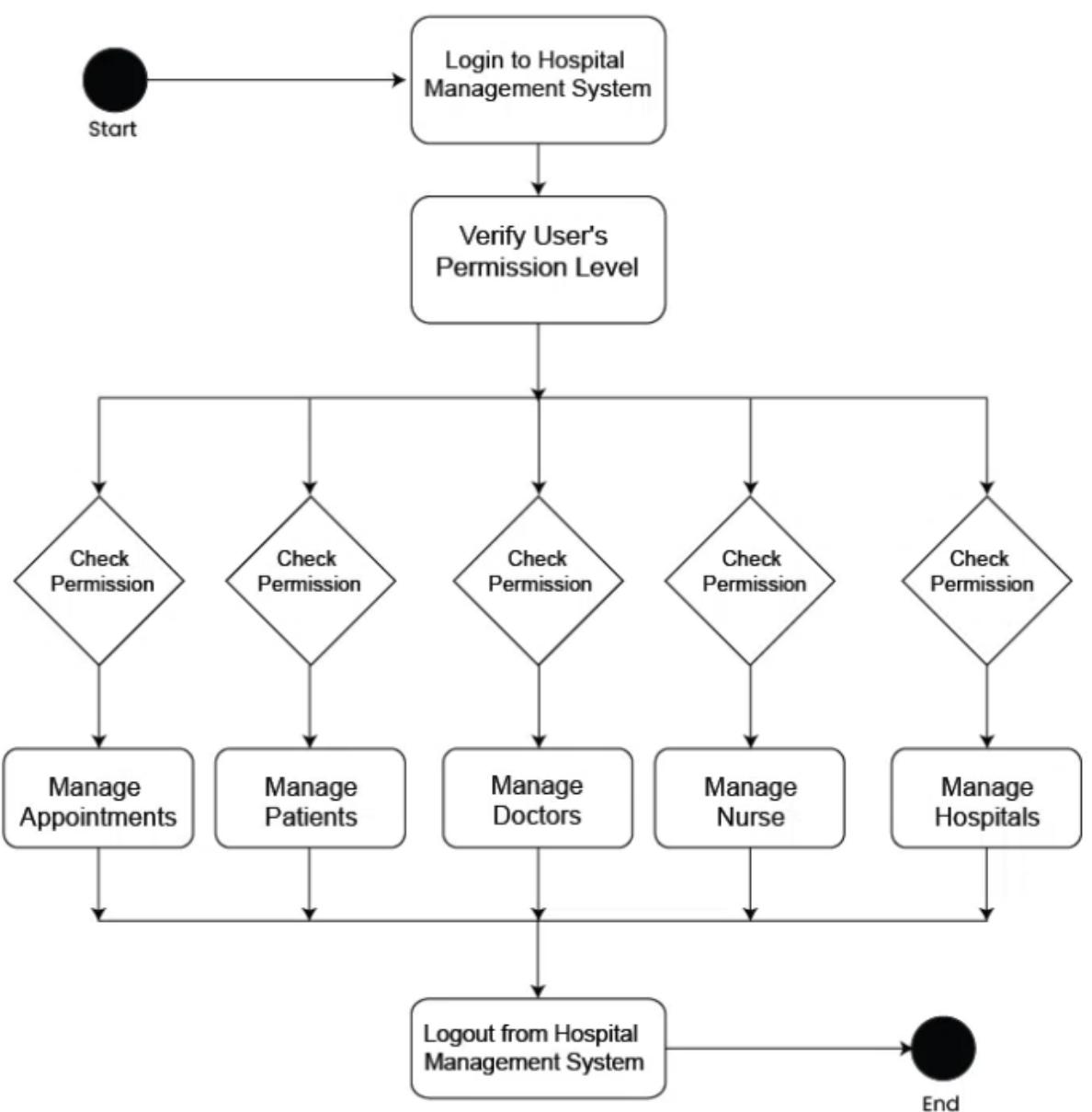






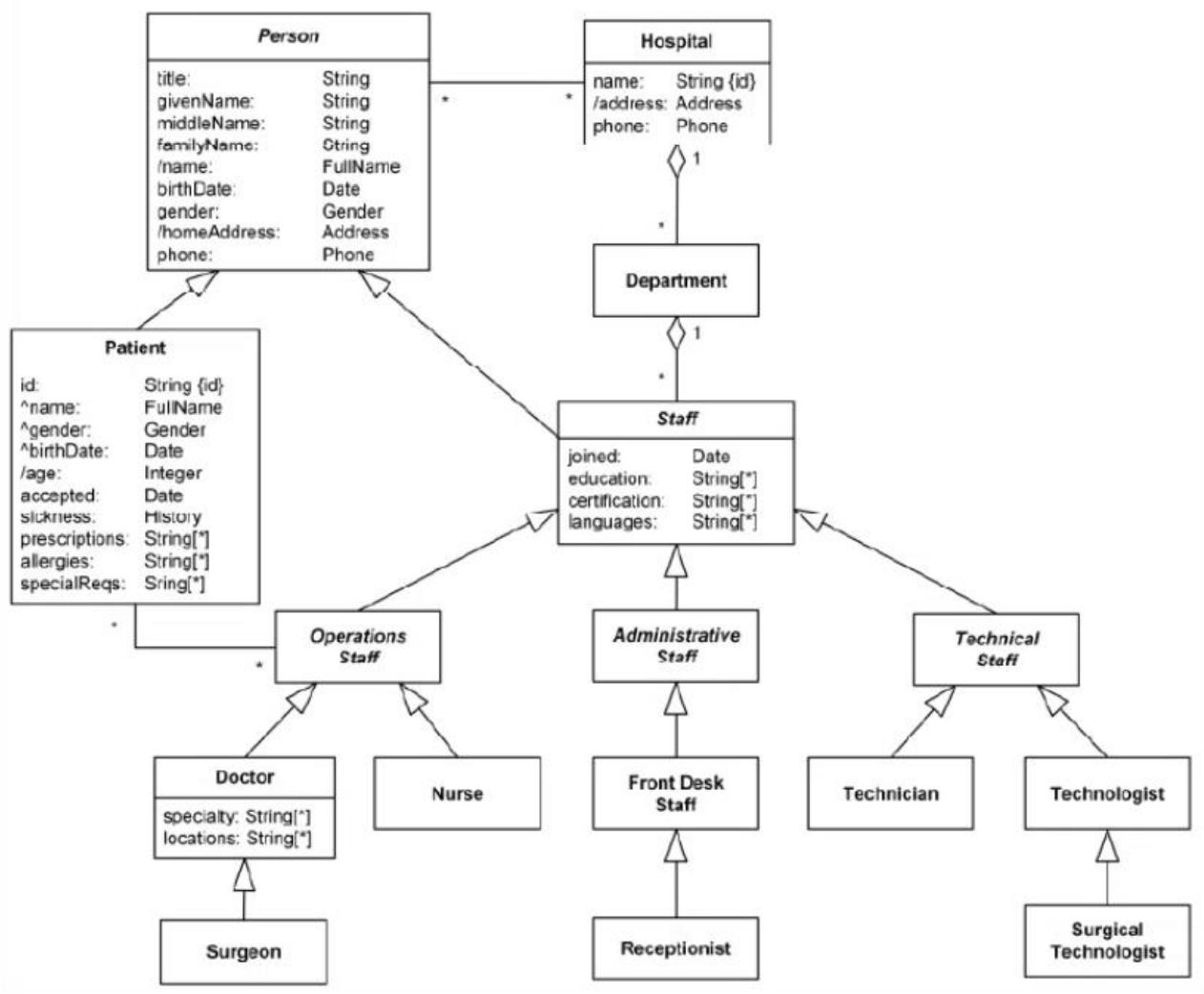
Sequence Diagram of Hospital Management System





Activity Diagram of Hospital Management System

class Organization



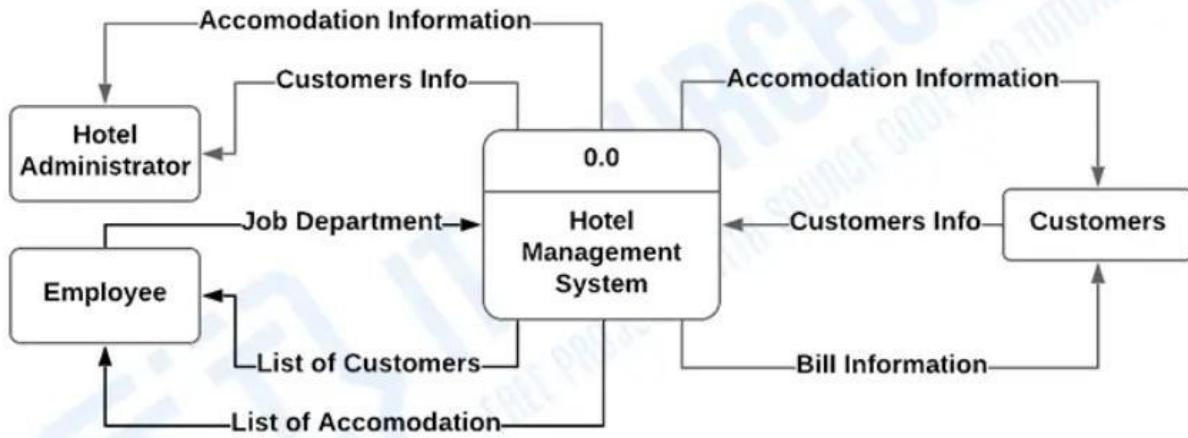
Class Diagram

26

Hotel Management System

The first level in discussing the system's data flow diagram is the **context diagram**. This level shows the system's core function in a one-line process.

Example:



Context Diagram for Hotel Management System

The example context diagram gives the general scenario when managing a hotel. Its concept is to reveal the general process, external entities (users), and data flow.

In the hotel management process, we have the following external entities:

- Hotel Administrator
- Employee
- Customer

These entities will be the main source of data inputs and recipients of data outputs.

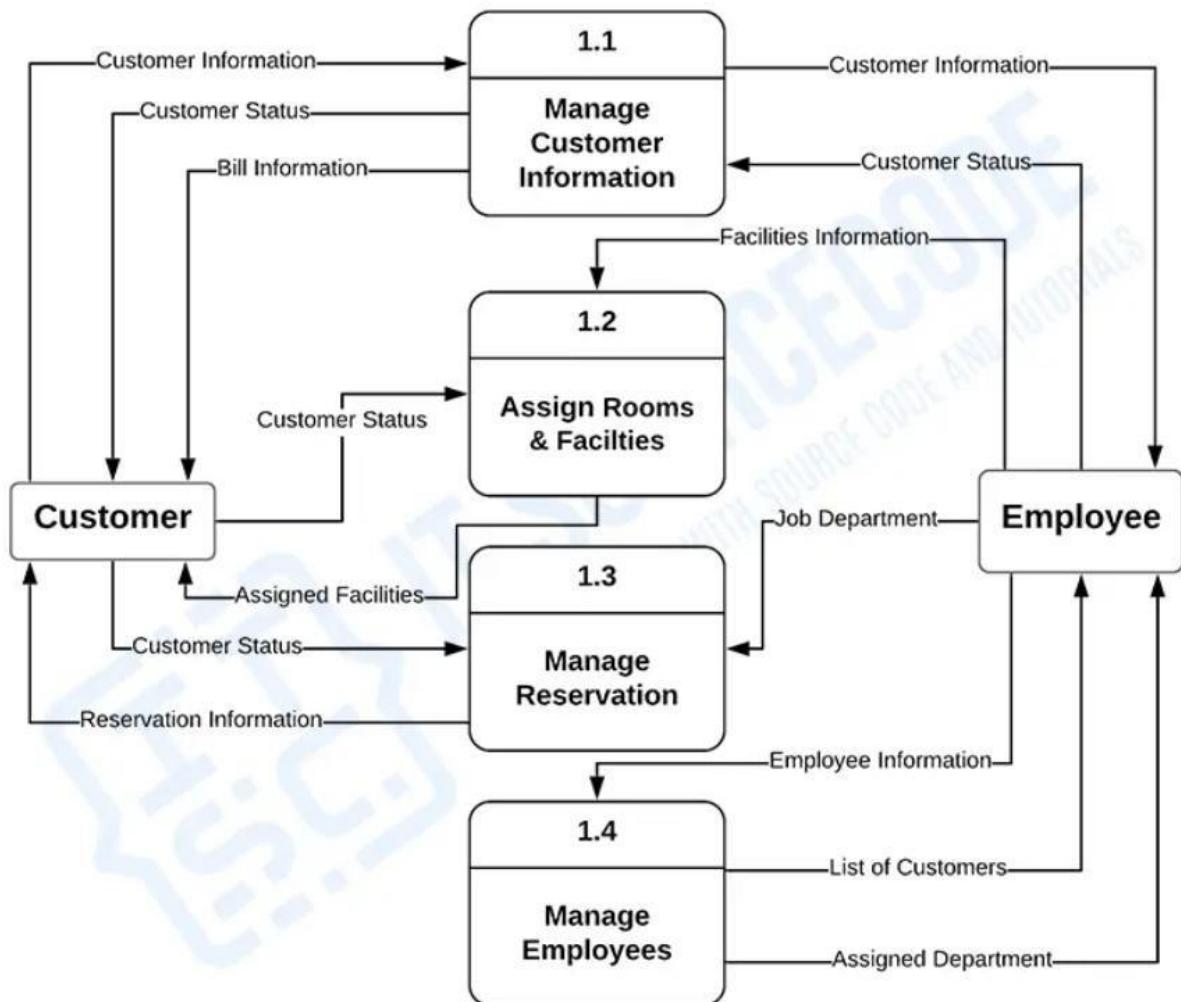
Remember: This context diagram originated from the common idea of hotel management.

Its context can be changed or modified to produce a new and unique project.

Level 1 DFD for Hotel Management System

To expand the idea of the context diagram, **DFD level 1** is the answer.

The DFD level 1 is the expanded view of the former level which highlights its subprocesses.



Level 1 DFD for Hotel Management System

This example diagram shows 4 of the sub-processes from the context diagram:

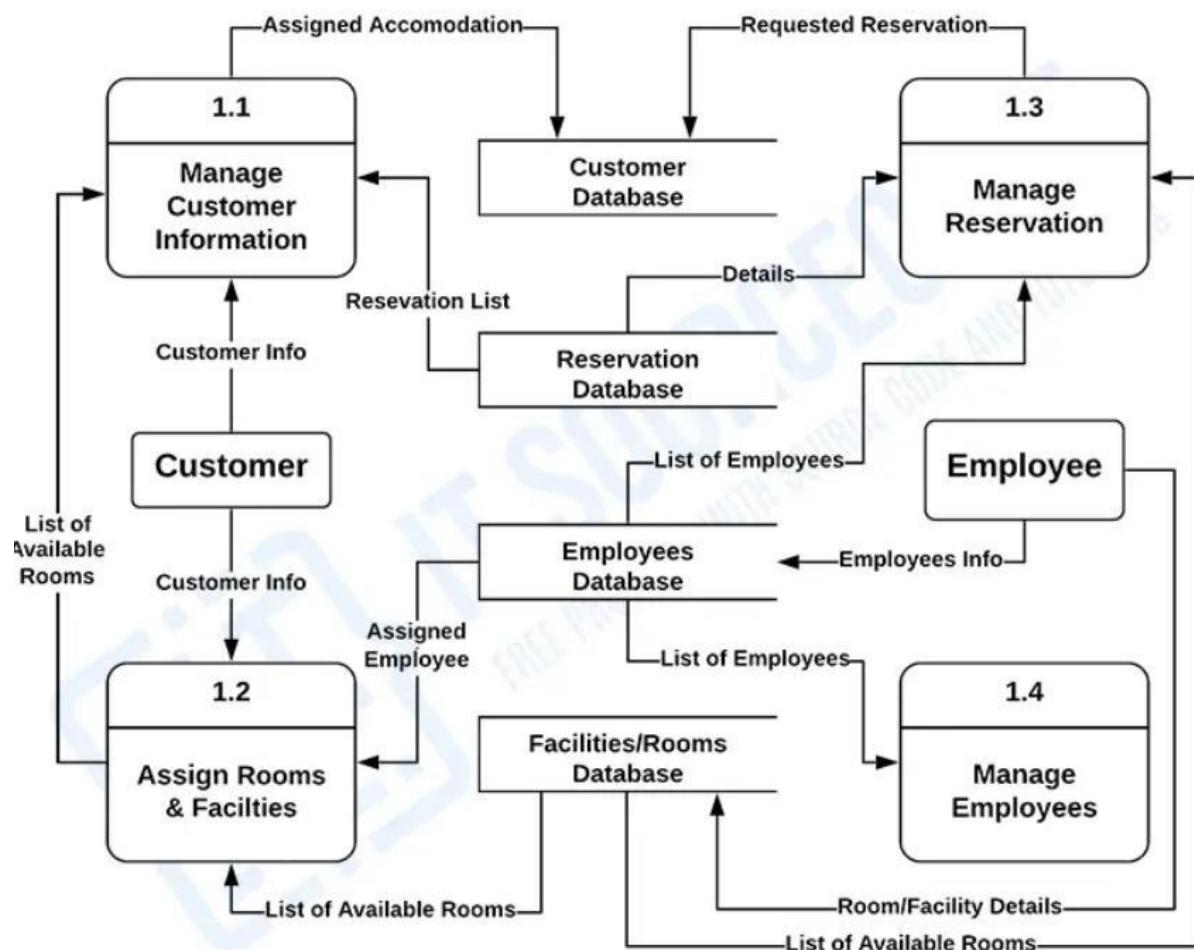
- **Customer Information Management** – is an information system where the customer's data is validated and used as a basis for other processes just like a reservation.
- **Assigning Rooms and Facilities** – this process may vary according to the request of certain customers. Its assignments are based on the orders of clients and guests (customers).
- **Reservation Management** – are available in the system and are offered to customers online. This business process is one of the core functions of the project.

- **Employees Management** – is the process used by the admin to designate their employees in different departments. This process gathers the employee information and monitors their workloads and performances.

These sub-processes show various data processing.

Level 2 DFD for Hotel Management System

While the former levels focus on processes, **DFD level 2** gives more attention to the other aspects of the project. Example:



Level 2 DFD for Hotel Management System

This level shows more emphasis on the system's databases (data stores). The databases are as follows:

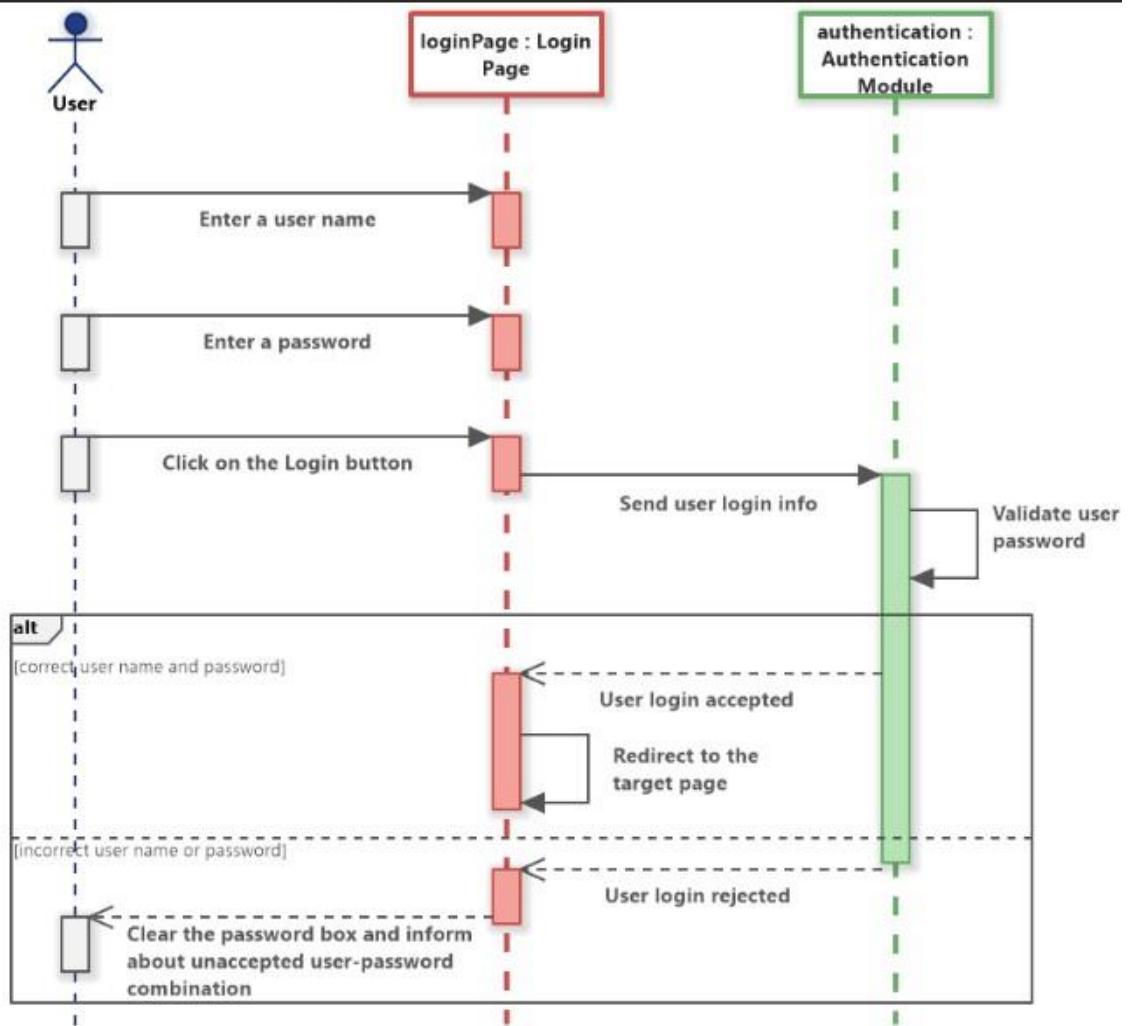
- Customer
- Reservation
- Employees

- Facilities or Rooms

The processed inputs are stored in these databases and are used to produce outputs.

SEQUENCE DAIGRAM FOR AUTHENTICATION SYSTEM

- **Lifelines:** Dashed vertical lines showing the life of each object during the process.
- **Activation Bars:** Thin rectangles showing when an object is active/processing a request.
- **Solid Arrows:** Messages or method calls between objects.
- **Dashed Arrows:** Responses or return messages.
- **alt Frame:** Used to depict branching logic (success vs failure paths).



Actors and Objects: 1) User – The person trying to log in (actor).

2) Login Page – The UI where the user enters credentials.

3) Authentication Module – Backend logic that verifies credentials.

Flow of Events:

User → Login Page

Enter a username

Enter a password

Click on the Login button

Login Page → Authentication Module

Sends the login credentials to the backend for validation.

Authentication Module → Authentication Module (self)

Validates the entered username and password against stored data.

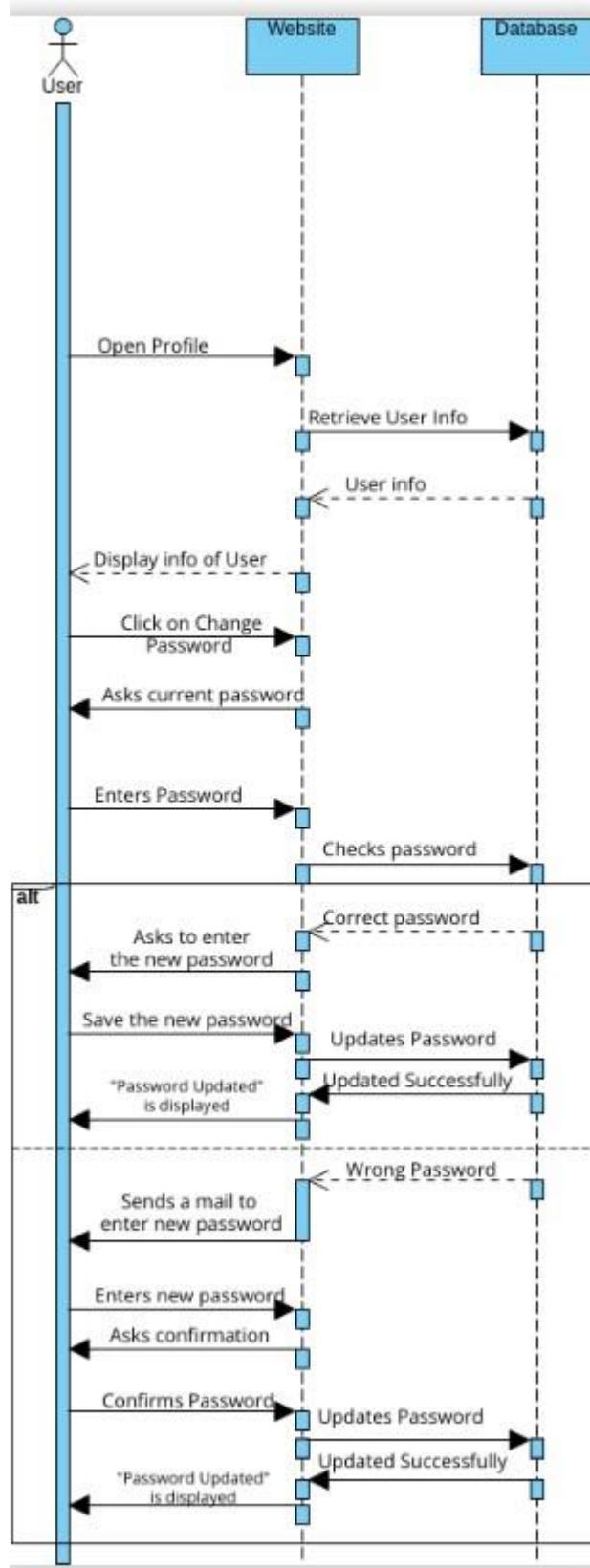
Alternative Outcomes (alt block): The alt frame represents alternative flows:

1)[correct username and password]

- Authentication Module sends a message: User login accepted.
- Login Page redirects the user to the target page.

2)[incorrect username or password]

- Authentication Module sends a message: User login rejected.
- Login Page clears the password field and displays an error message about the invalid combination.



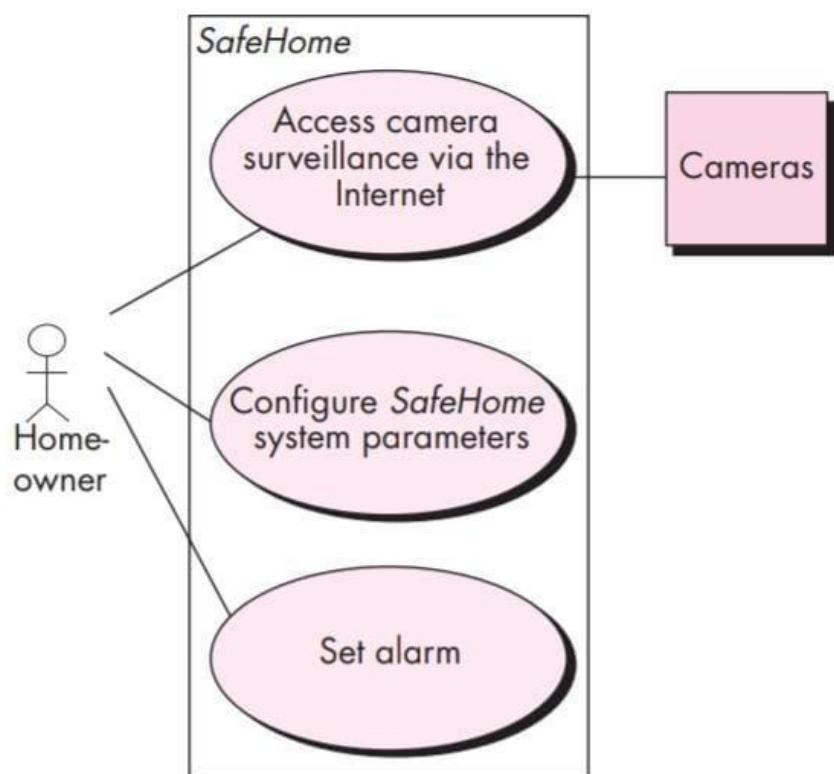


FIGURE 6.5

Activity diagram for Access camera surveillance via the Internet—display camera views function.

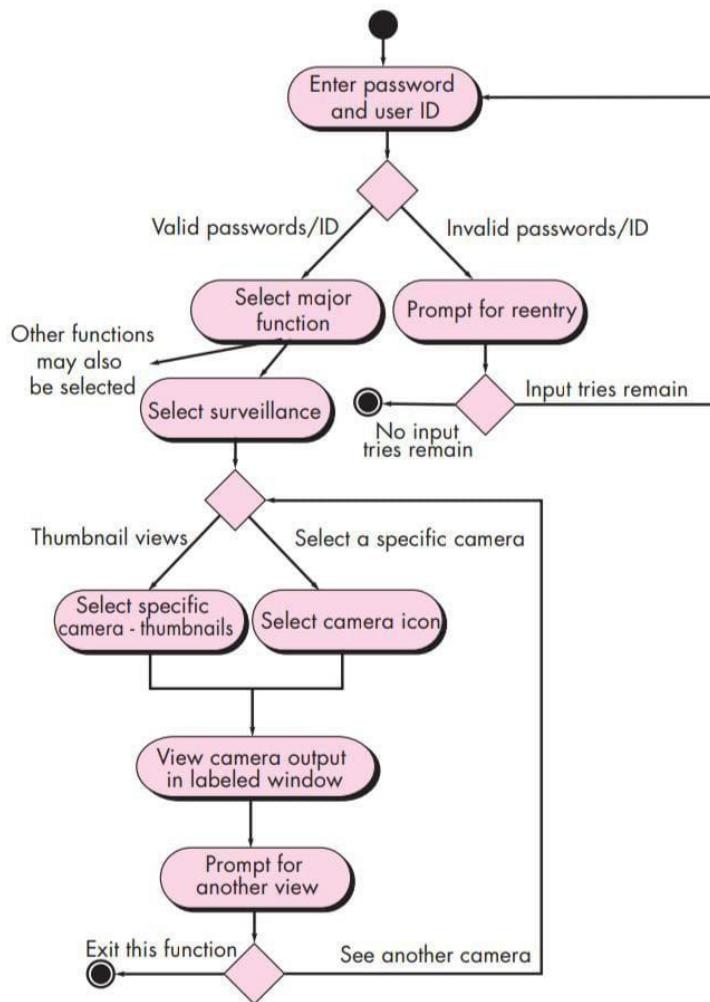


FIGURE 6.6 Swimlane diagram for Access camera surveillance via the Internet—display camera views function

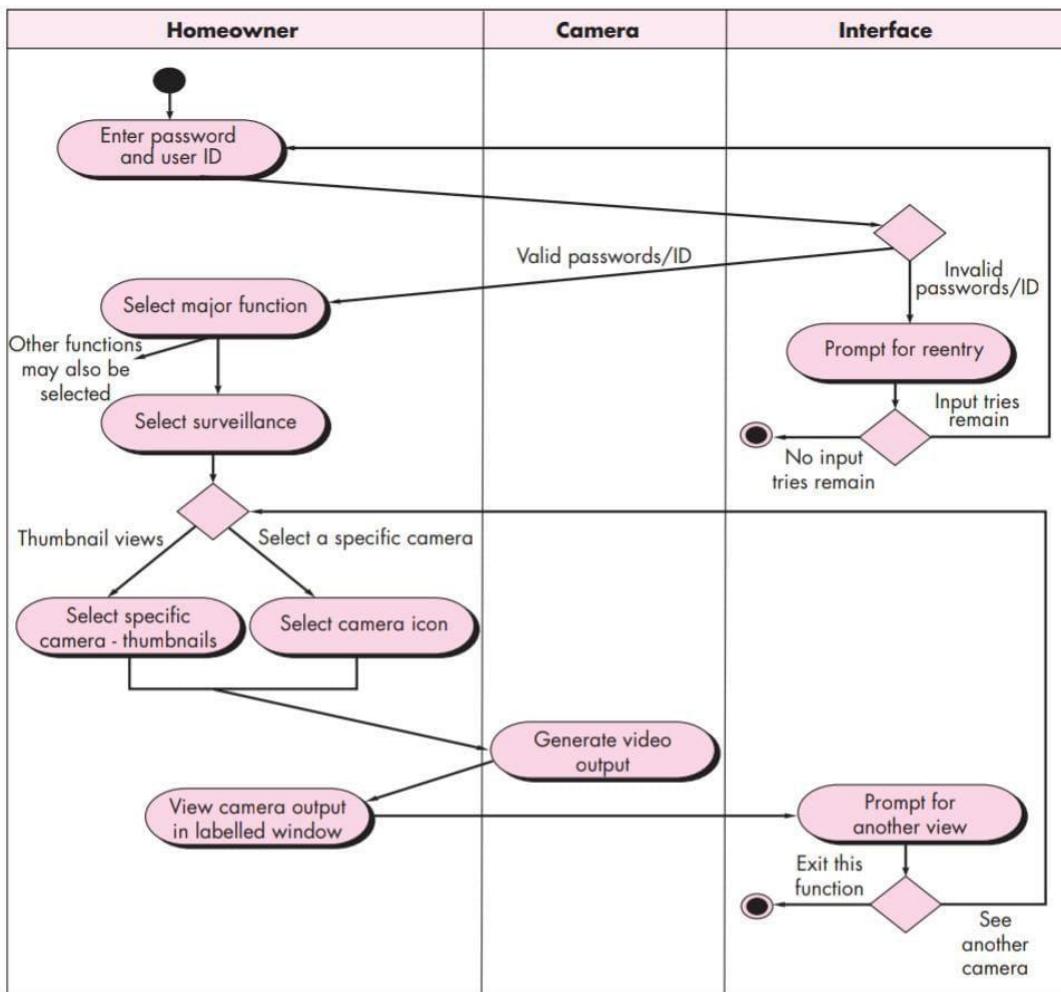


FIGURE 7.1

Context-level
DFD for the
SafeHome
security
function

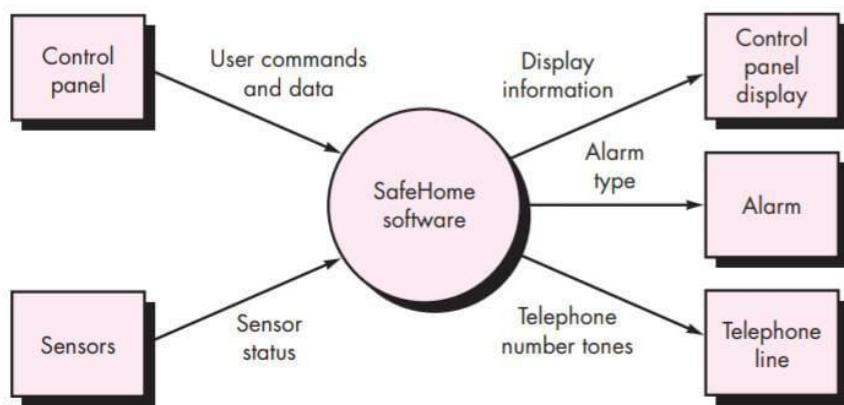


FIGURE 7.2

Level 1 DFD for
SafeHome
security
function

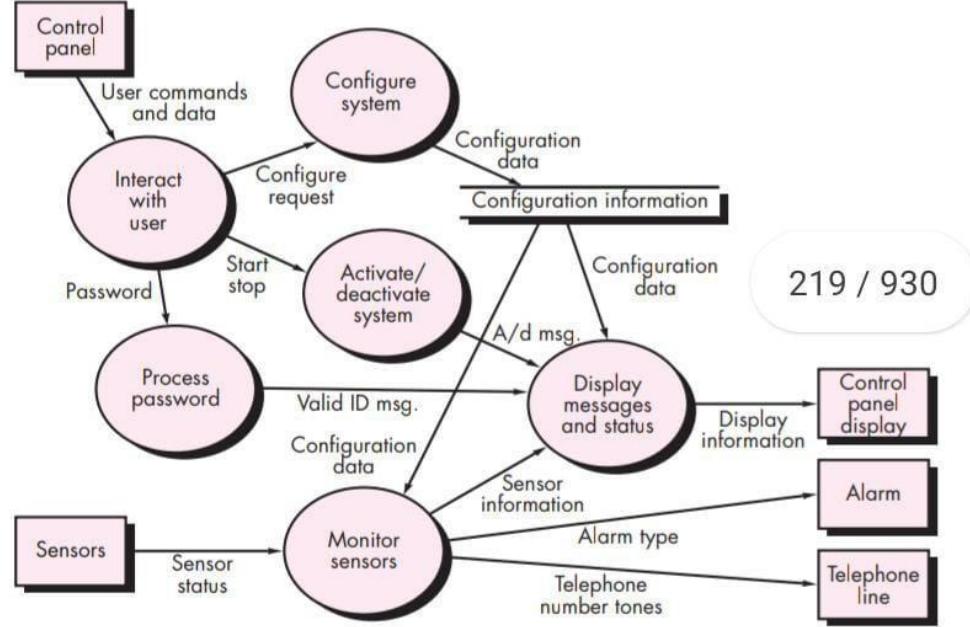


FIGURE 7.3

Level 2 DFD
that refines
the monitor
sensors process

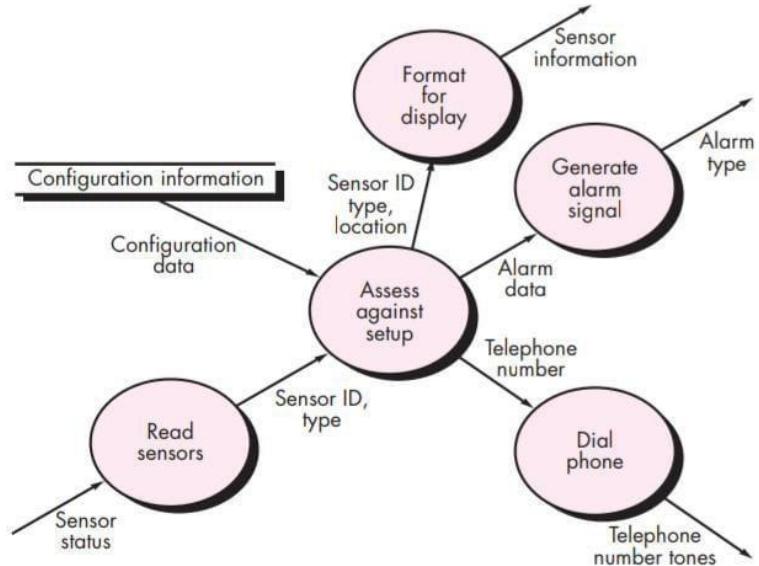


FIGURE 7.4 State diagram for SafeHome security function

