

In 1,2 there'll be a component diagram and UML. UML is of 2 types. Structural and Behavioral.

If structural -> class diagram, behavioral -> Use case diagram

Box & Line, Formal Description, UML

Visualization using xADL = Component diagram

In box and line, boxes are the same as the components of the component diagram and just show linkage between them.

Functional vs Non-Functional Requirements

- Requirements, which are related to functional/Working aspect of software fall into this category.
- Non-Functional Requirements are expected characteristics of target software. (Security, Storage, Configuration, Performance, Cost, Interoperability, Flexibility, Disaster recovery, Accessibility)

4+1

Architectural View Model

W Model (HINDI)

① Logical view :-

- Static Modelling view
- deals with functionalities
- Sequence, Class and Communication diagrams

② Development view :-

- Component design view
- developer's view
- Component & Package diagram
- Subsystem and system development.

③ Process view :-

- Task view
- deals with runtime aspects of system
- Scalability & Performance
- Run time description of System structure
- Activity diagram

④ Physical view :-

- deployment of system
- Tools & environment of system
- System engineer's point of view
- Deployment diagram

④+1 Scenarios :-

- Use cases

4:43

Component Diagram

[Component Diagram Explained in Hindi | UML Diagram | Software Modeling and Designing Course - YouTube](#)

[UML Component Diagram for beginner with Solved Example in Hindi | SOOAD Series - YouTube](#)

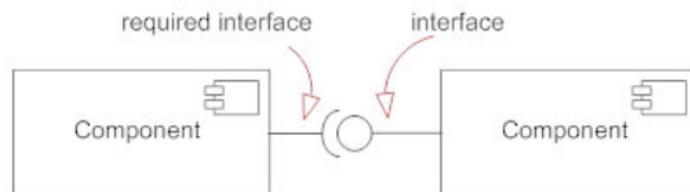
Component

A component is a logical unit block of the system, a slightly higher abstraction than classes. It is represented as a rectangle with a smaller rectangle in the upper right corner with tabs or the word written above the name of the component to help distinguish it from a class.



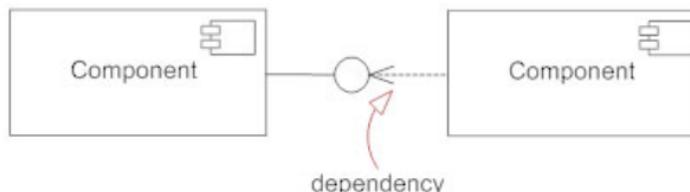
Interface

An interface (small circle or semi-circle on a stick) describes a group of operations used (required) or created (provided) by components. A full circle represents an interface created or provided by the component. A semi-circle represents a required interface, like a person's input.



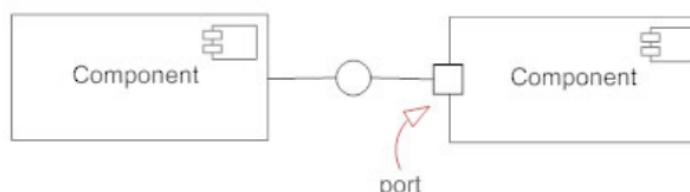
Dependencies

Draw dependencies among components using dashed arrows.

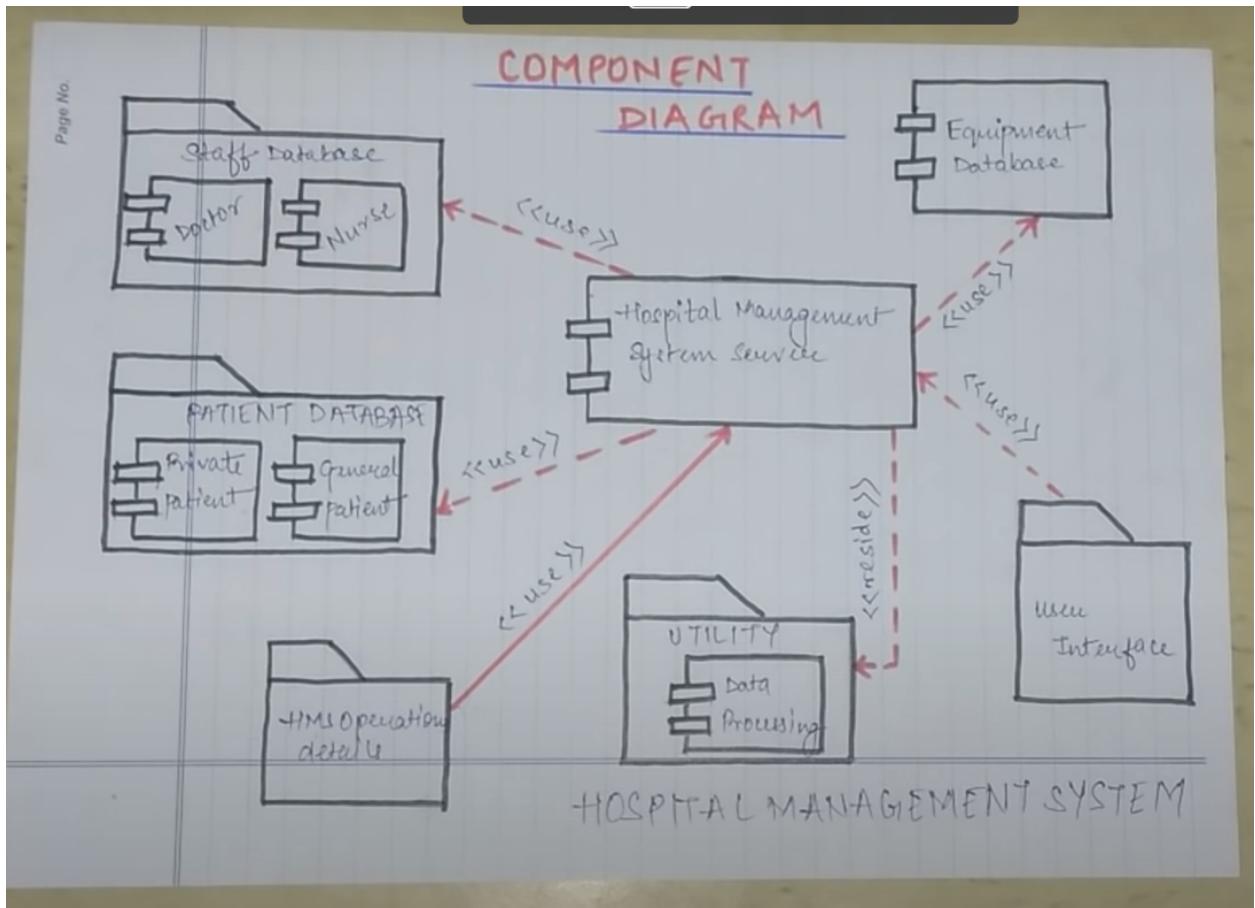


Port

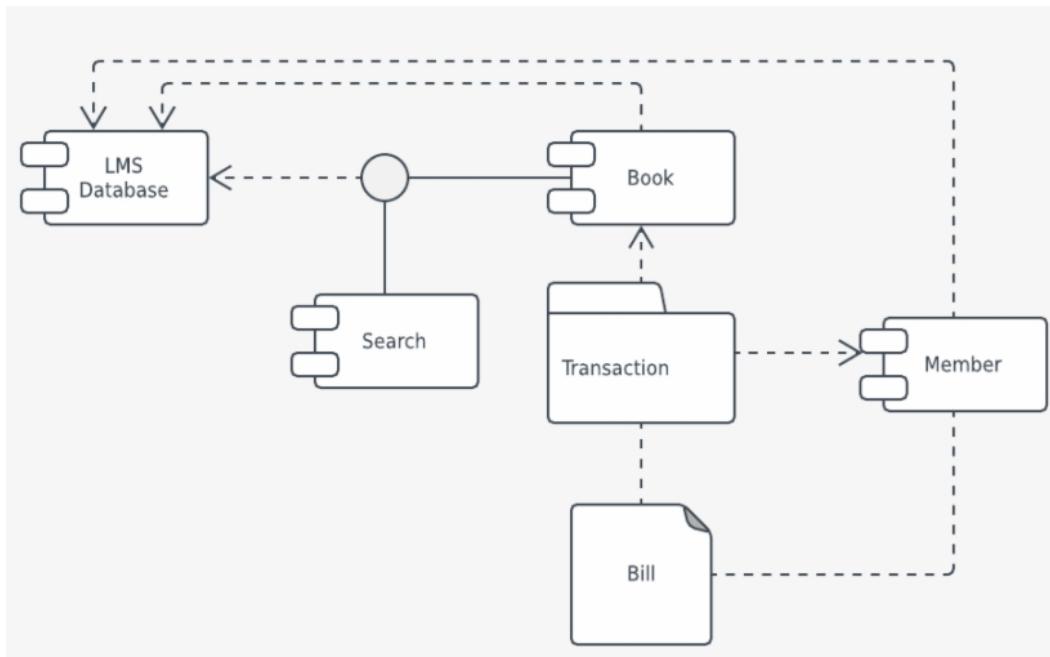
Ports are represented using a square along the edge of the system or a component. A port is often used to help expose required and provided interfaces of a component.



Hospital Management Component Diagram

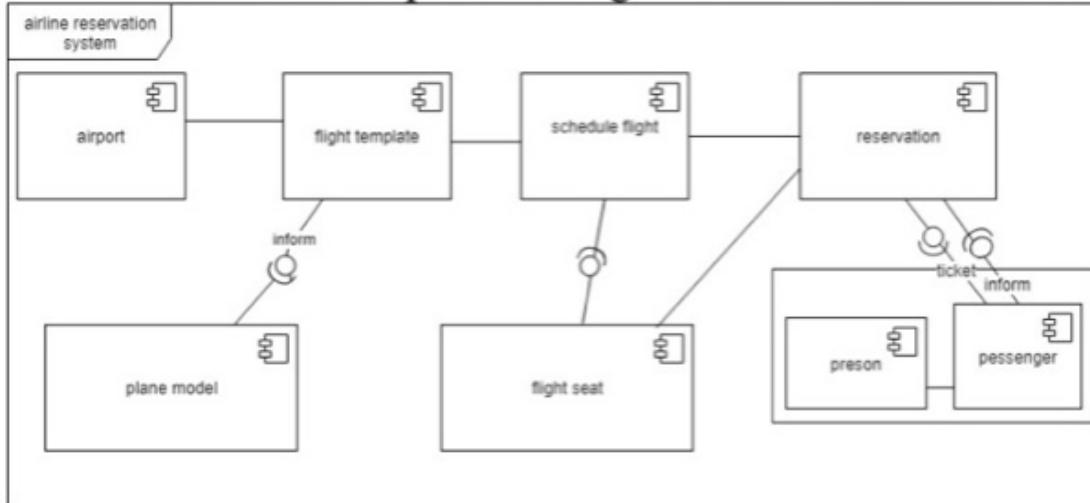


Library Management System

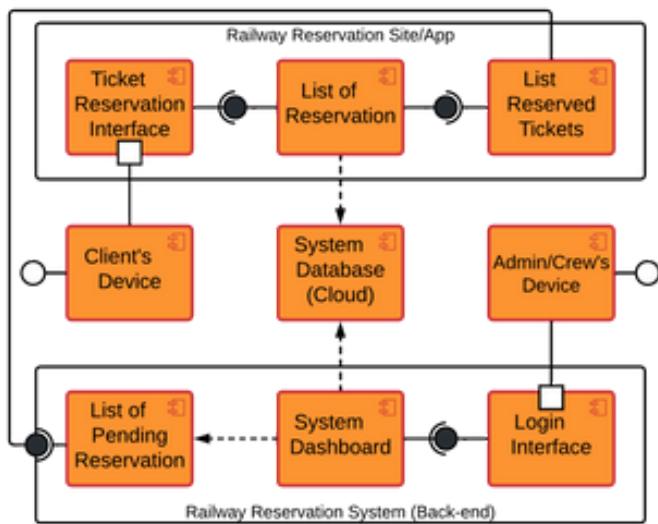


Airline Management System

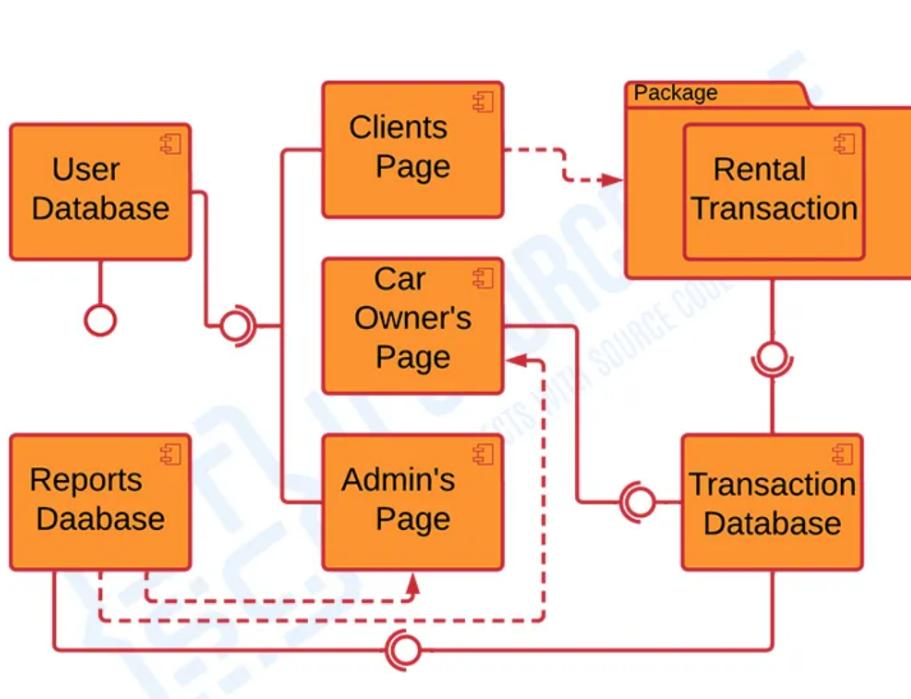
This is the component diagram of airline reservation



Railway Management System

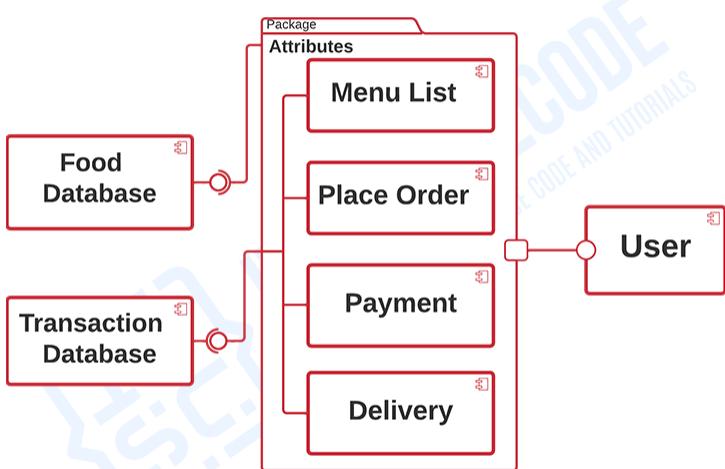


Vehicle management service



Online food delivery system

ONLINE FOOD ORDERING SYSTEM

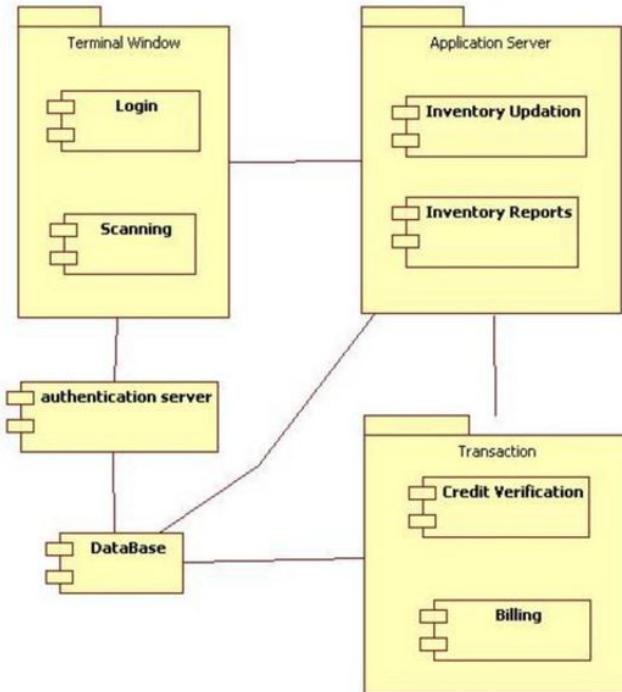


COMPONENT DIAGRAM

Online grocery system

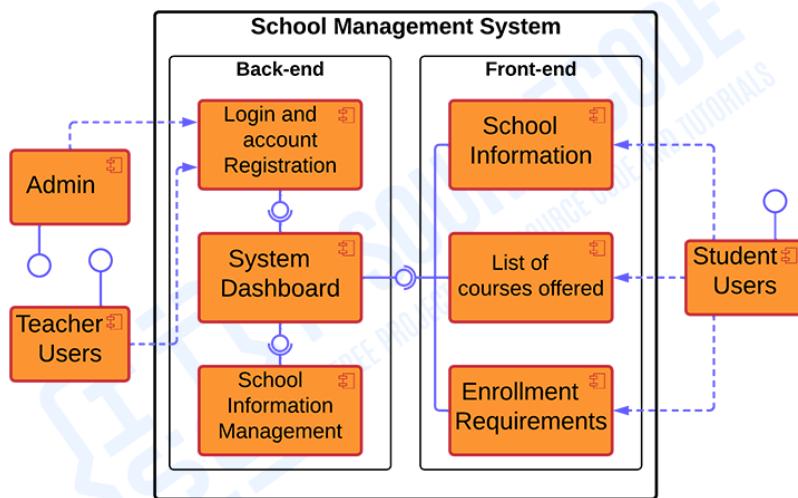
Component

Diagram



School Management System

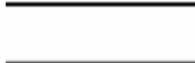
SCHOOL MANAGEMENT SYSTEM



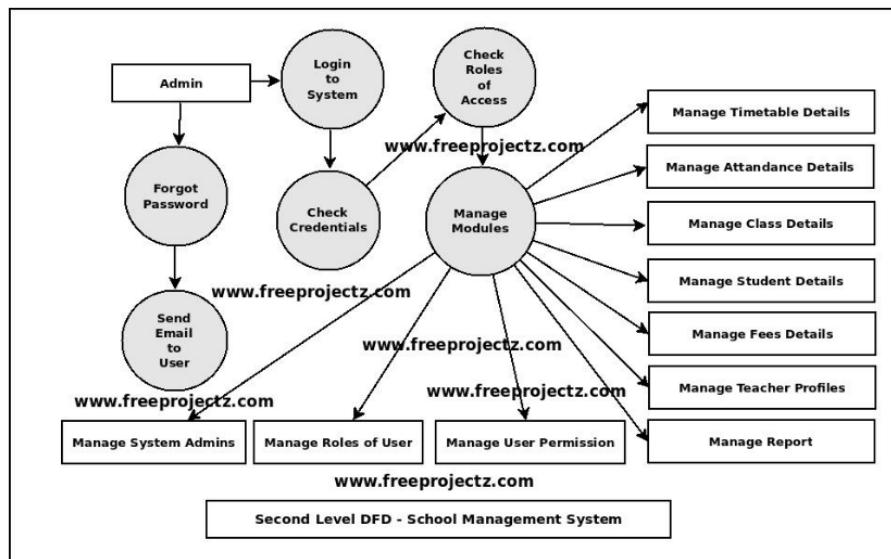
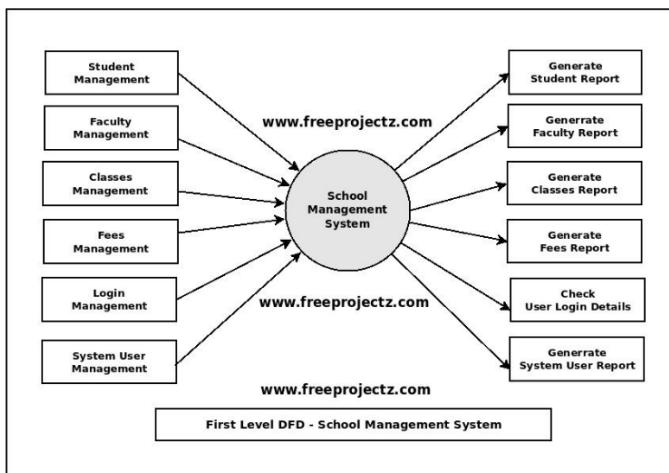
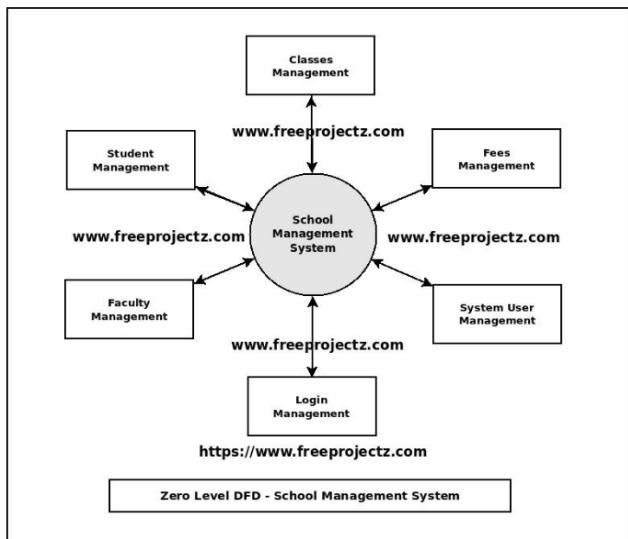
COMPONENT DIAGRAM

DFD

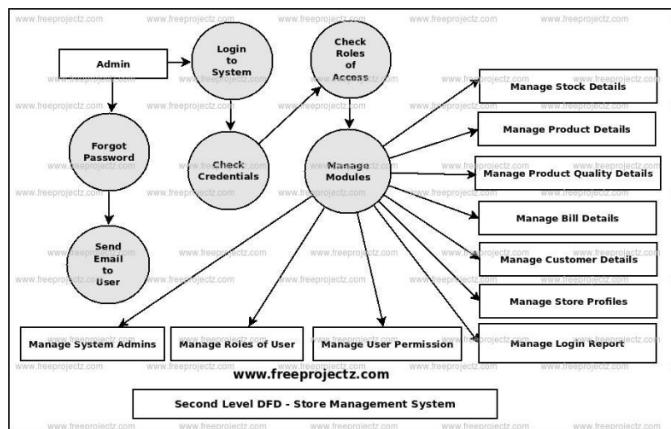
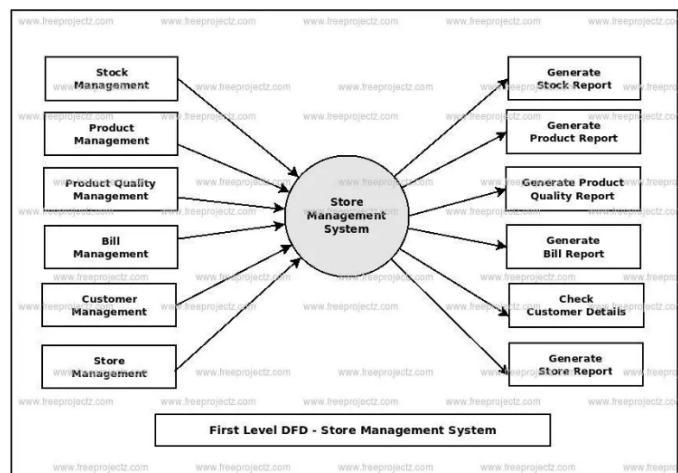
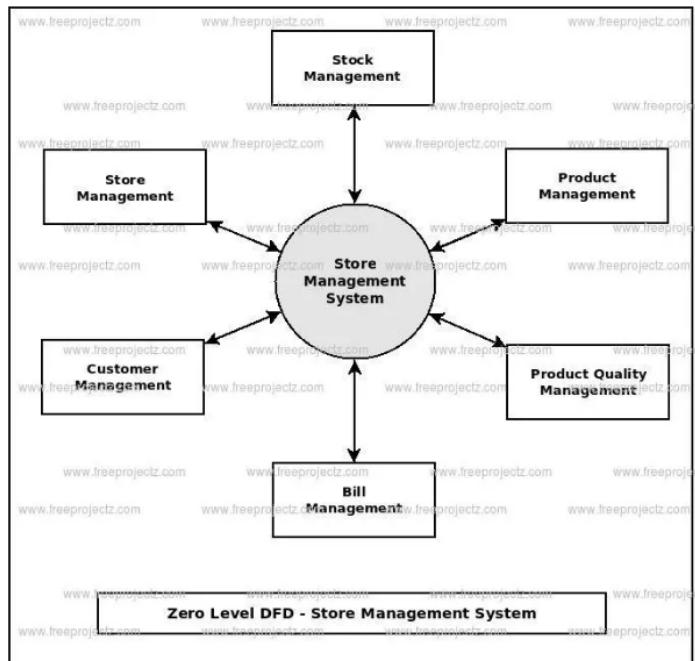
<https://www.youtube.com/watch?v=KN-inGJG540>

Symbol	Description
	Data Flow – Data flow are pipelines through the packets of information flow.
	Process : A Process or task performed by the system.
	Entity : Entity are object of the system. A source or destination data of a system.
	Data Store : A place where data to be stored.

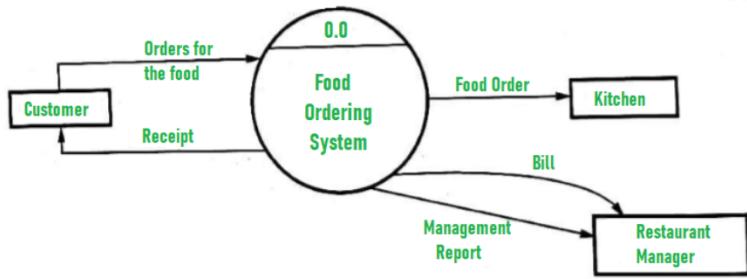
School Management System



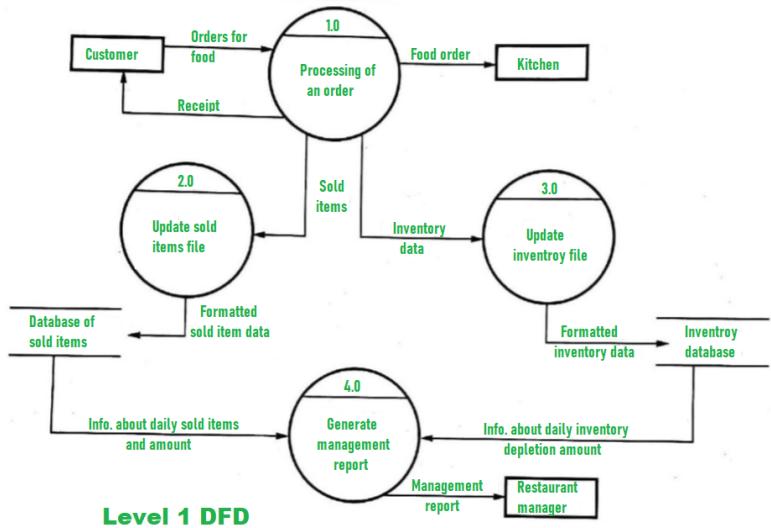
Online grocery management system



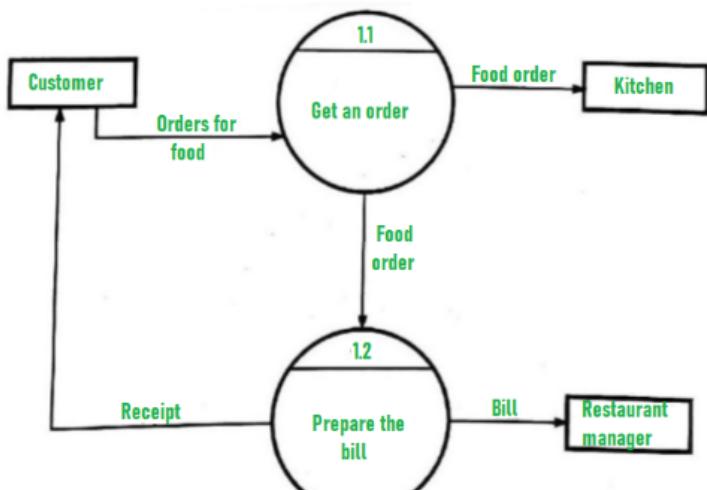
Online food delivery system



Level 0 DFD (Context Level)

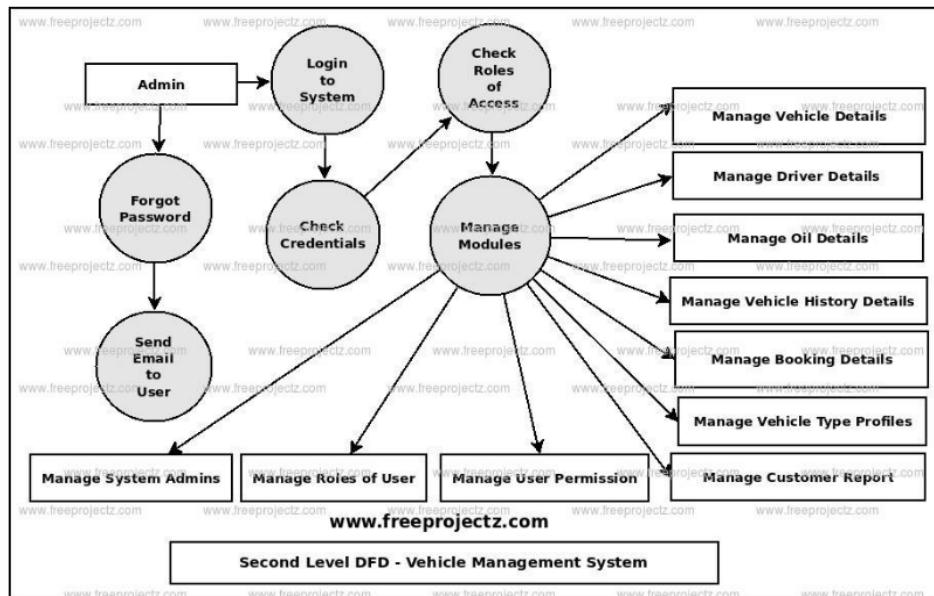
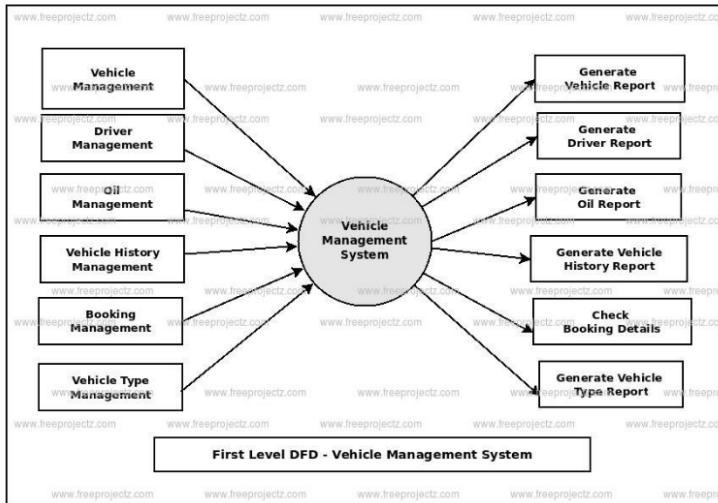
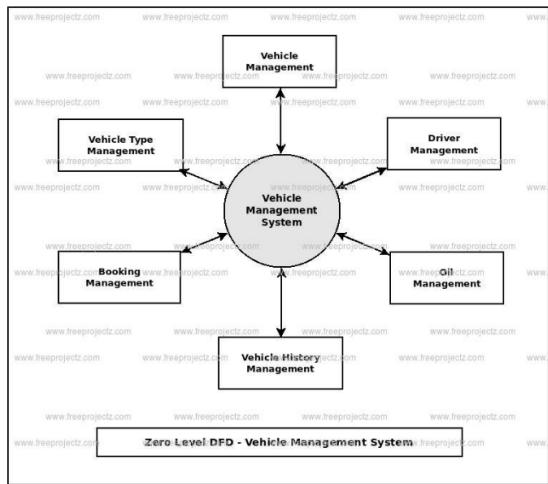


Level 1 DFD

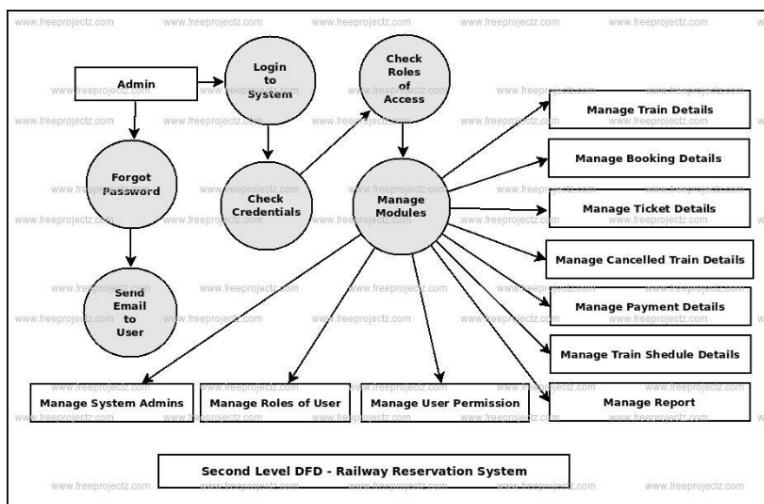
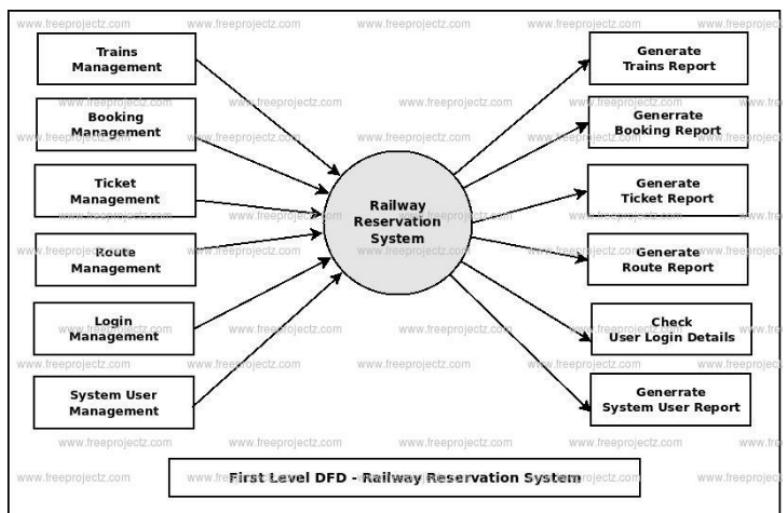
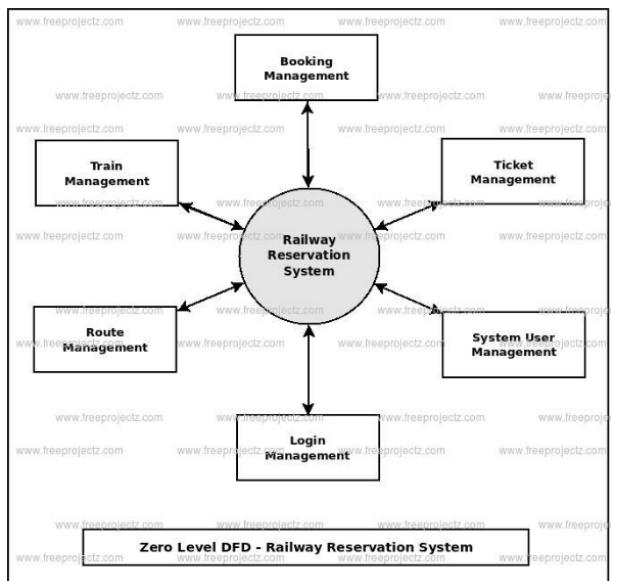


Level 2 DFD

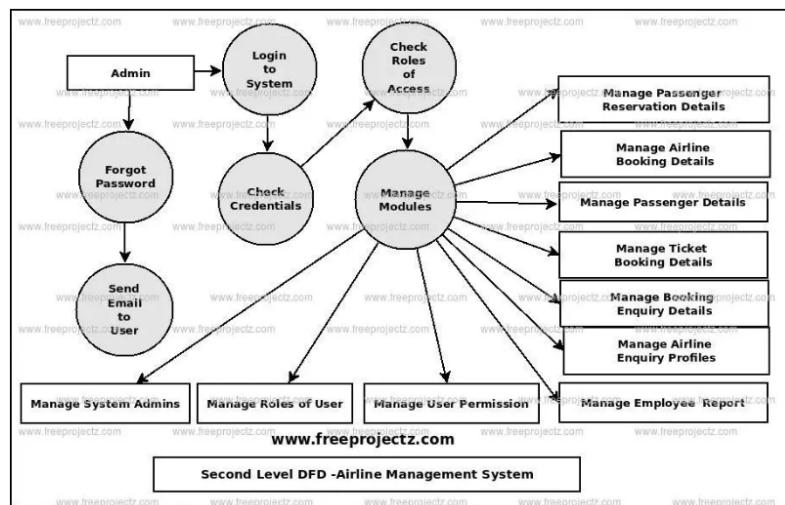
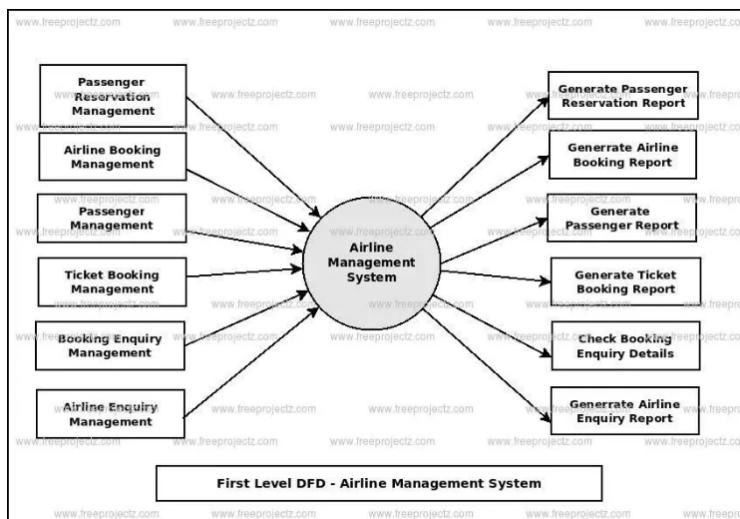
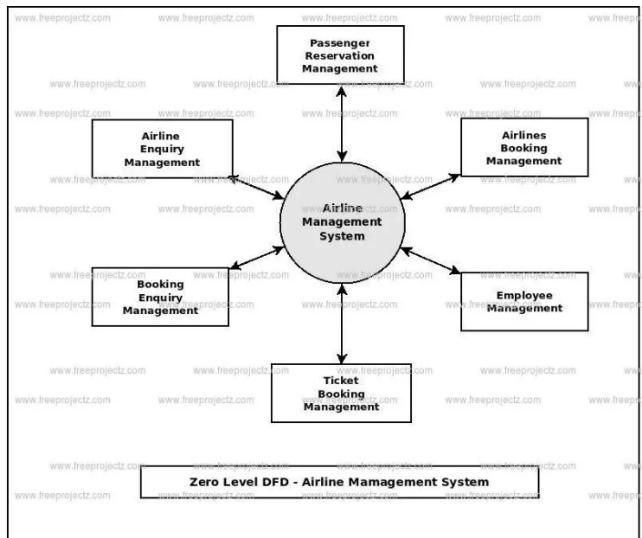
Vehicle Management System



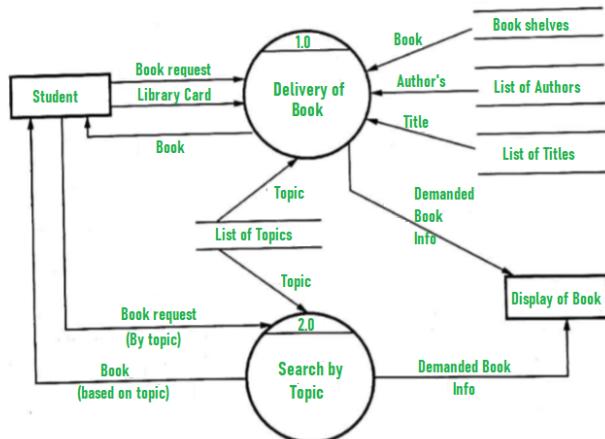
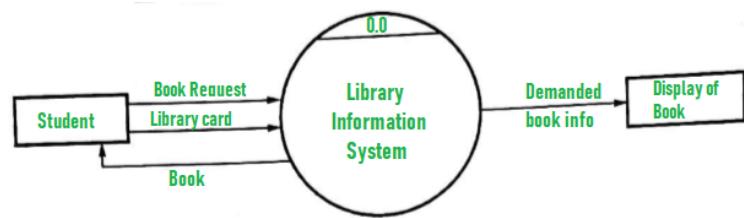
Railway management system



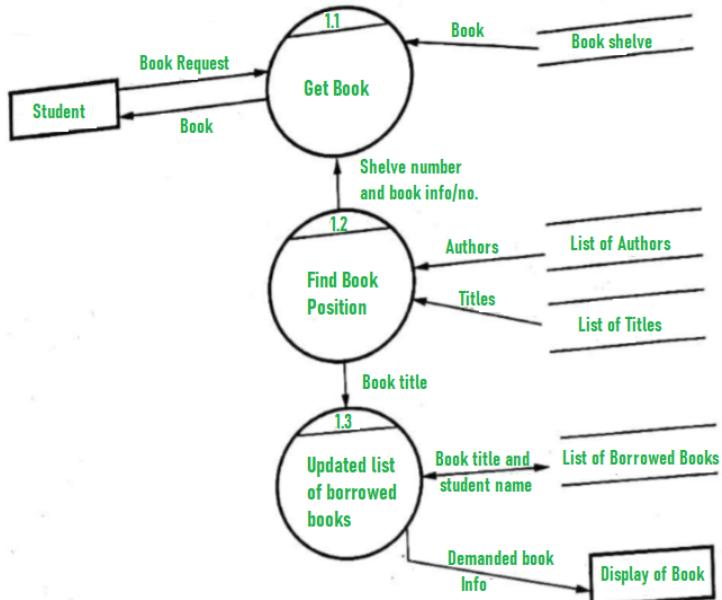
Airline management system



Library management system



Level 1 DFD



Level 2 DFD

Hospital Management System

