DESIGN

1. **STRUCTURAL DESIGN**
2. **CLASS DIAGRAM**

A class diagram is a type of diagram and part of a unified modelling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods and the relationship between different classes. Class Diagram is useful in all forms of object-oriented programming (OOP). Class is a description of a group of objects all with similar roles in the system which consists of structural and behavioural features.

The importance of class diagram is:-

1. Shows the static structure of classifiers in a system.
2. Diagram provides basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too.
4. Business analysts can use class diagram to model systems from business perspectives.
5. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
6. It describes the attributes and operations of a class and also the constraints imposed in the system.

Notation used:-

A class notation consists of three parts:-

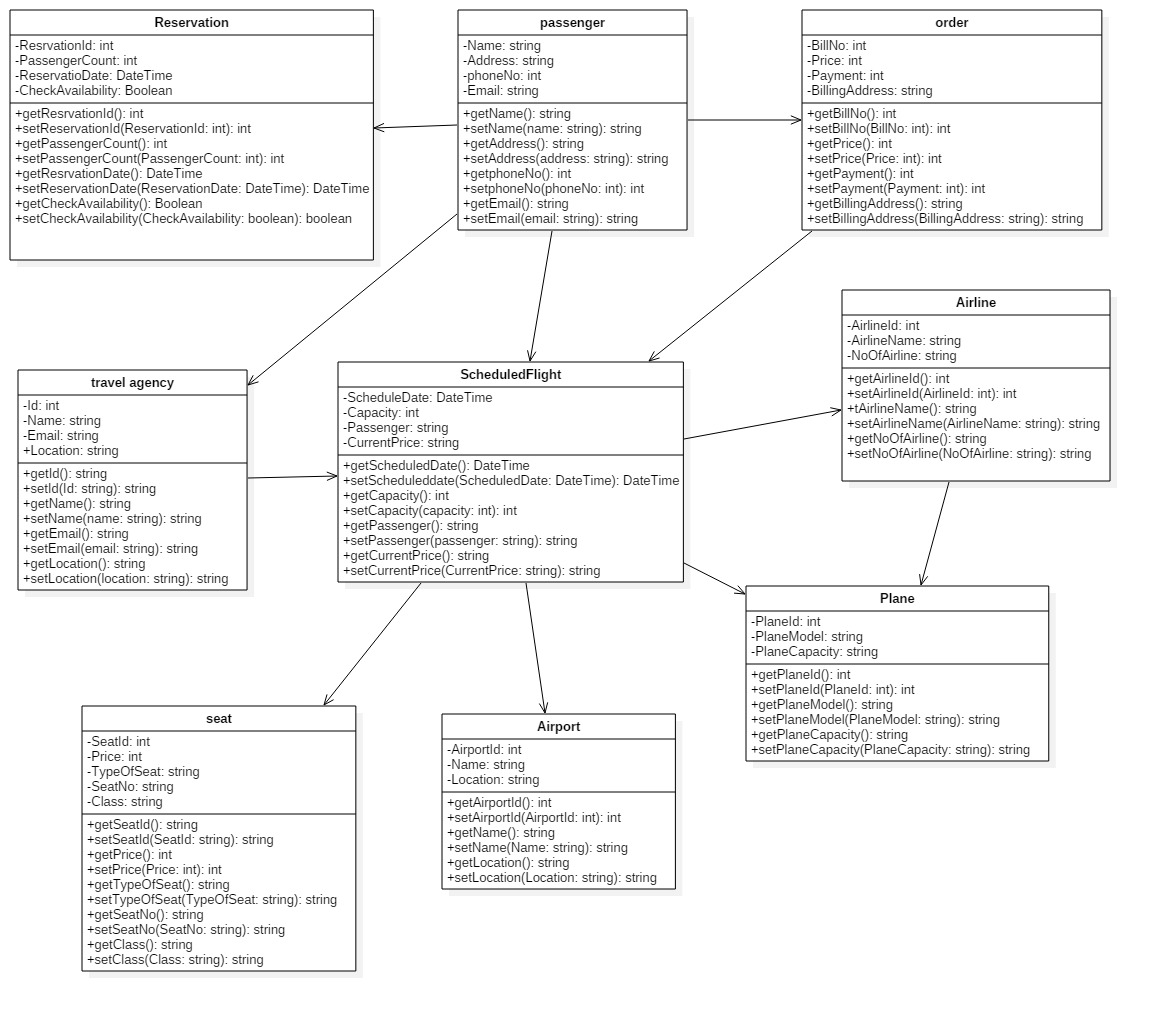
1. **Class name**: - the number of class appears in the first partition.

Fig: - Class

1. **Class attributes**: - Attributes are shown in the second partition. The attributes type is shown after the colon. Attributes map onto member variables (data members) in code.
2. **Class operation**: - operations are shown in the third partition. They service the class providers. The return type of a method is shown after the colon at the end of the method signature. The return type of method parameters are shown after the colon following the parameter name. Operations map onto class methods on code.

**Association**

**Aggregation**



**Fig.: - Class Diagram**

I have attached the class diagram in the design phase of my project because it shows the static structure of classifiers in a system. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. It helps to model the static view of an application. It helps to describe the structure of a system by showing the system’s classes, their attributes, operations and the relationship among objects.

1. **FLOWCHART**

Flowchart is the diagram of the sequence of movements or actions of people or things involved in a complex system or activity. It is a graphical representation of a computer program in relation to its sequence of functions. A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving task. The flowchart shows the steps as boxes of various kinds and their order by connecting the boxes with arrows.

The importance of flowchart is: -

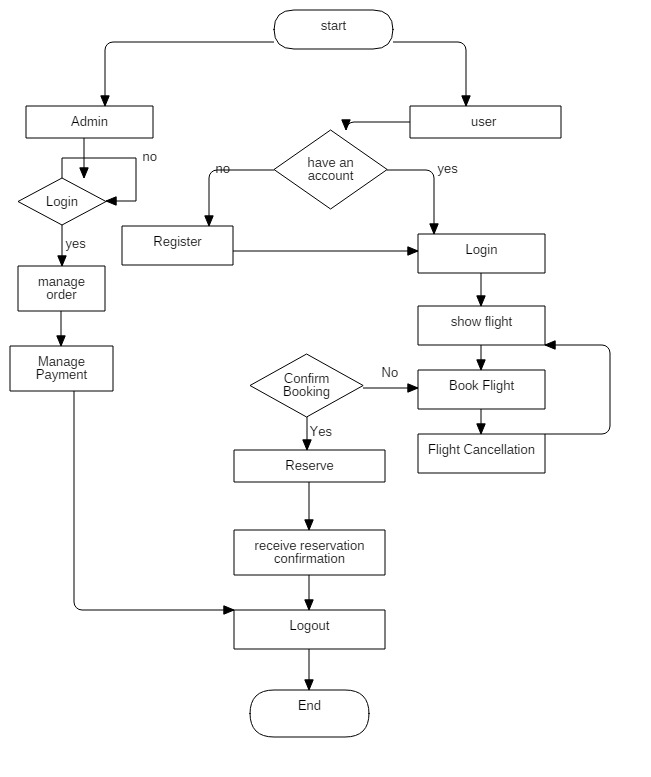
1. It is used in designing and documenting simple processes or programs.
2. It helps to visualize what is going on and thereby help understand a process.
3. It helps to find less obvious features within the process like flaws and bottlenecks.
4. It is often used in training to document an existing process or to evaluate the efficiency of that process.
5. It is depict through images how a process is performed from start to finish typically in sequential order.
6. They are better way of communicating the logic of a system to all concerned or involved.

Notation used: -

1. **The oval**: - an end or beginning

1. **The rectangle**: - a step in the flowcharting process
2. **Arrow**: - directional flow

1. **Diamond**: - decision



**Fig.: - Flowchart for online airline management system**

I have included the flowchart diagram in my project because it is used in designing and documenting simple processes and programs. It helps to find out the flaws. It is performed from start to finish typically in sequential order. It helps to understand the process and features. It is the better way of communicating also. This is one of the effective ways of analysis because with the help of flowchart, problem can be analysed in more effective way therefore reducing cost and wastage of time.

1. **BEHAVIORAL DESIGN**
2. **ACTIVITY DIAGRAM**

Activity diagram are the graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. It is another important diagram in the UML to describe the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

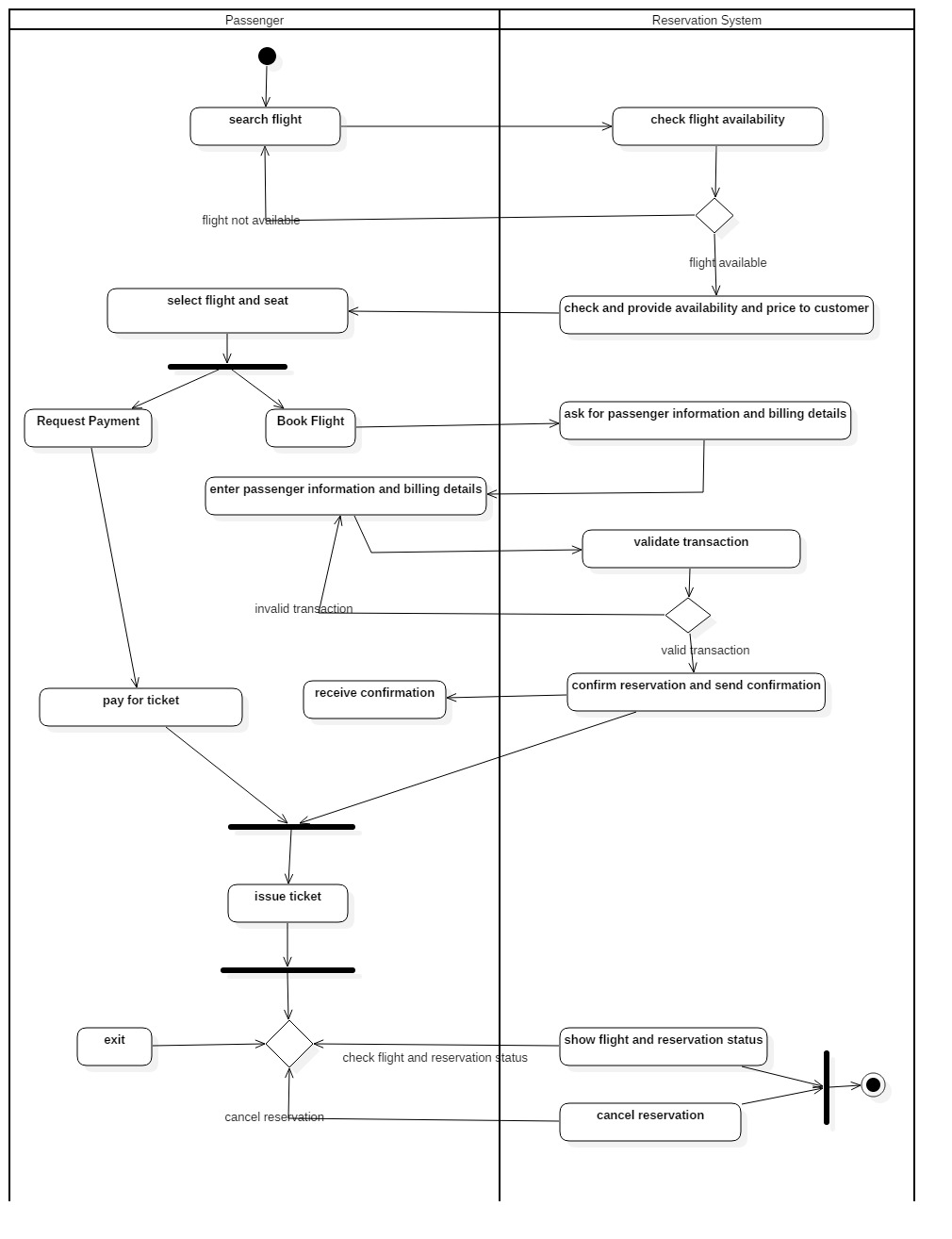
Importance of activity diagram:-

1. It is used by programmers as it is advanced version of flowchart to depict workflows.
2. It is used by developers to understand the flow of programs on a high level.
3. It also enables them to figure out constraints and conditions that cause particular events.
4. Activity diagram is used to show message flow from one activity to another.
5. It captures the dynamic behaviour of the system.

Notation used:-

1. **Activity**: - it is used to represent the set of actions.
2. **Action**: - it represents a task to be performed.
3. **Control flow**: - It shows the sequence of actions.
4. **Object flow**: - It shows the flow of an object from one activity to another activity.
5. **Initial node**: - It portrays the beginning of a set of actions or activities.
6. **Activity Final Node**: - It stops all the control flows and object flows in an activity diagram.
7. **Object Node**: - it represents an object that is connected to a set of object flows.
8. **Decision node**: - it represents a test condition to ensure that the control flow or object flow only goes down one path.
9. **Merge node**: - it bring back together different decisions path that were created using a decision node.

1. **Fork node**: - It split behaviour into a set of parallel or concurrent flows of activities for actions.
2. **Join node**: - it bring back together a set of parallel or concurrent flows of activities for actions.
3. **Swimlane and Partition**: - It is a way of group activity performed by the same actor on an activity diagram or to group activities in a single thread.



**Fig.: - Activity Diagram for online airline reservation system**

I have attached the activity diagram in my project as it is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show the message flow from one activity to other.

1. **SEQUENCE DIAGRAM**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagram are sometime called event diagram or event scenarios. It is a good diagram to use to document a system’s design. The reason the sequence diagram is so useful is because: -

1. It shows the interactions logic between the objects in the system in the time order that the interactions take place.
2. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
3. Sequence diagram are more space efficient, simple to modify, allows vertical expansion rather that the horizontal which is the case for sequence diagram.
4. It is an essential component used in processes related to analysis, design and documentation.
5. In the context of UML, it represents object collaboration and is used to define event sequences between objects for a certain outcomes. Sequence diagram are more space efficient, simple to modify, allows vertical expansion rather that the horizontal which is the case for sequence diagram.

UML sequence diagram notation:-

1. **Lifeline**: - it represents each instance in an interaction.
2. **Activate**: - it is used to denote participant activation. Once a participant is activated, its lifeline occurs.
3. **Objects**: - they are model elements that represent instances of a class or of classes.
4. **Classes**: - In UML, it shows architecture and features of the designed system.
5. **Message**: - it is an element that defines a specific kind of communication between instances in an interaction.
6. **Actor**: - it specifies a role played by a user or any other system that interacts with the subjects.
7. **Note**: - it contains comments or textual information.
8. **Constraint**: - it is an extension mechanism that enables us to refine the semantics of a UML model element.
9. **Edraw Software**: - it provides us lots of shapes used in UML sequence diagram.

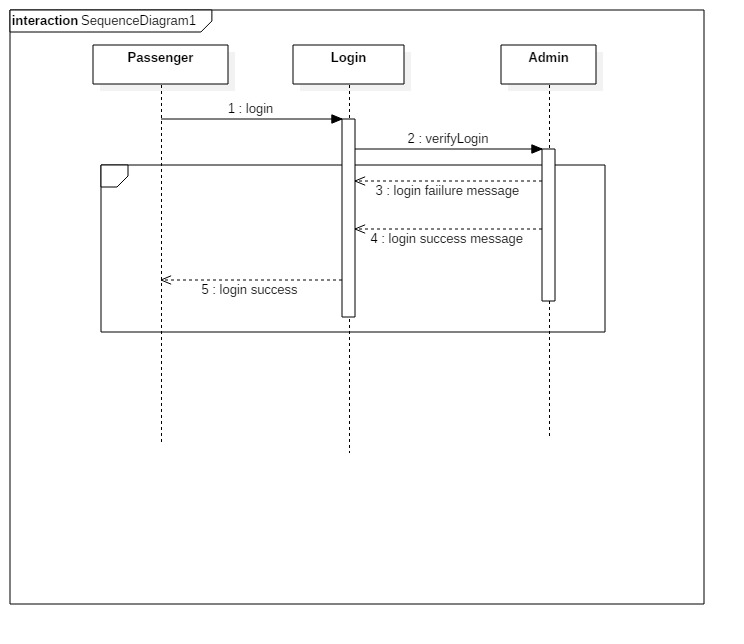
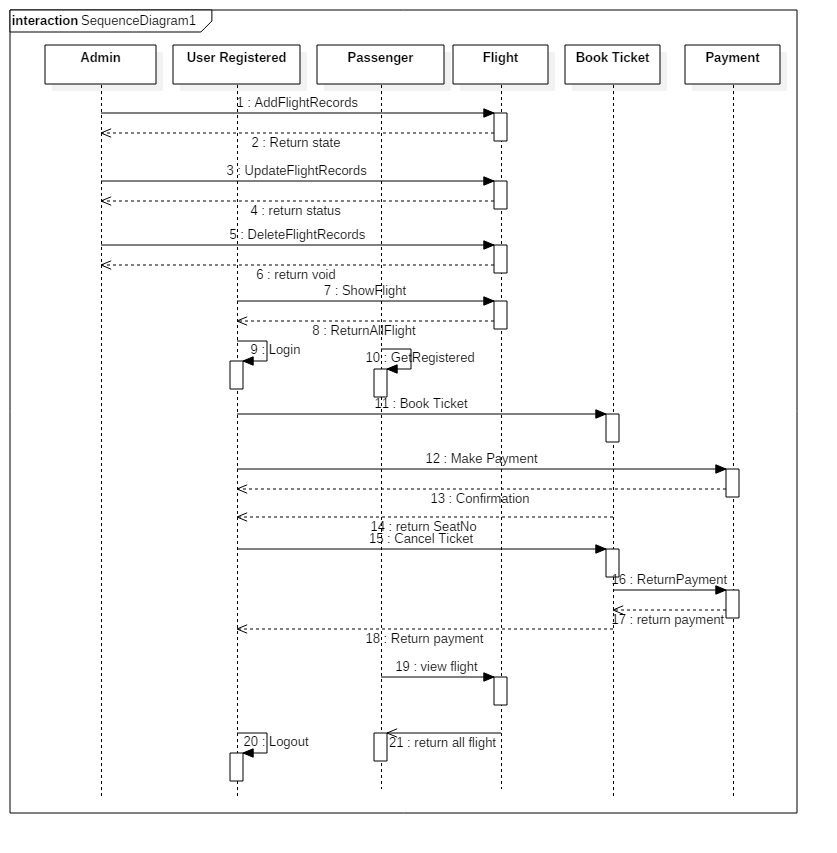


Fig.: - sequence diagram for login



**Fig.: - sequence diagram for online airline management system**

In above project, I have mentioned the sequence diagram because it is a good diagram to use to document a system’s requirement and to flush out a system’s design. It is so useful because it shows the interaction logic between the objects in the system in the time order that the interactions take place. Sequence diagram are more space efficient, simple to modify, allows vertical expansion rather that the horizontal which is the case for sequence diagram. It shows the object interactions arranged in time sequence. They are typically associated with use case realizations in the logical view of the system under development.

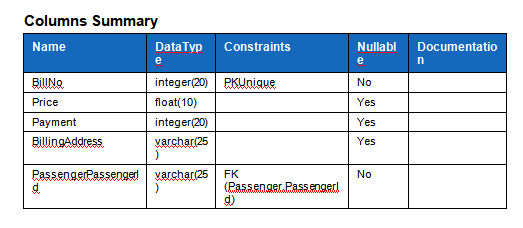
DATABASE

1. Data Dictionary

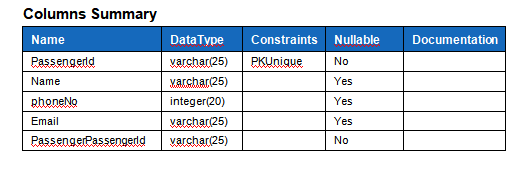
A data dictionary is the set of information that describes the contents, format and structures of a database and the relationship between its elements, used to control access to and manipulation of the database.

Given below, I have included the screenshot of the summary column from my project: -

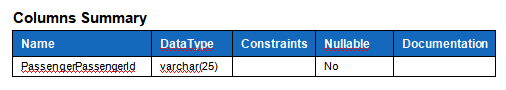
Order



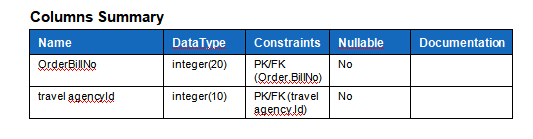
Passenger



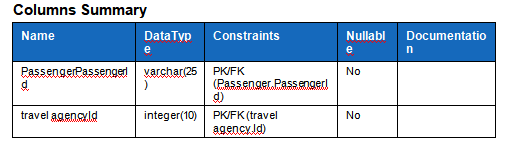
Entity



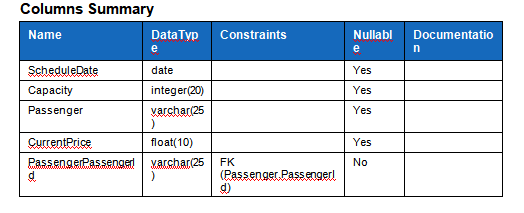
Order\_travel agency



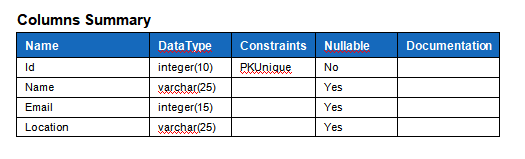
Passenger\_travel agency



Scheduledflight

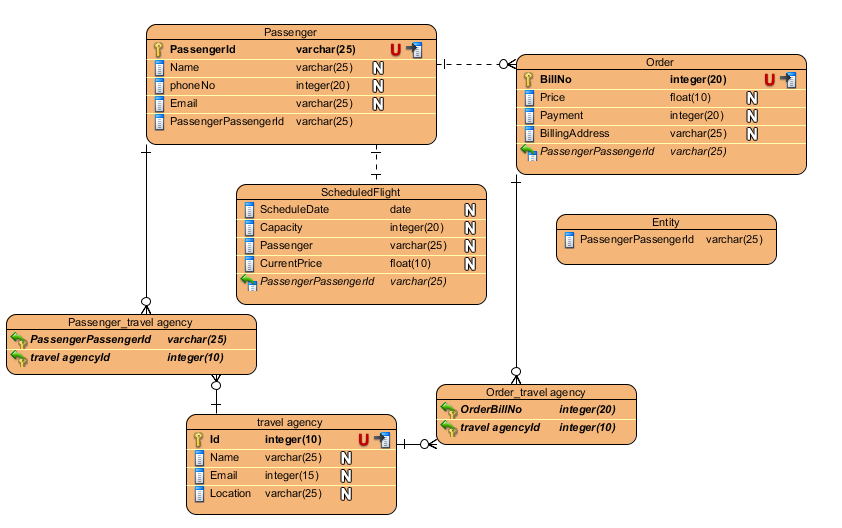


Travel agency



1. **ER-Diagram**

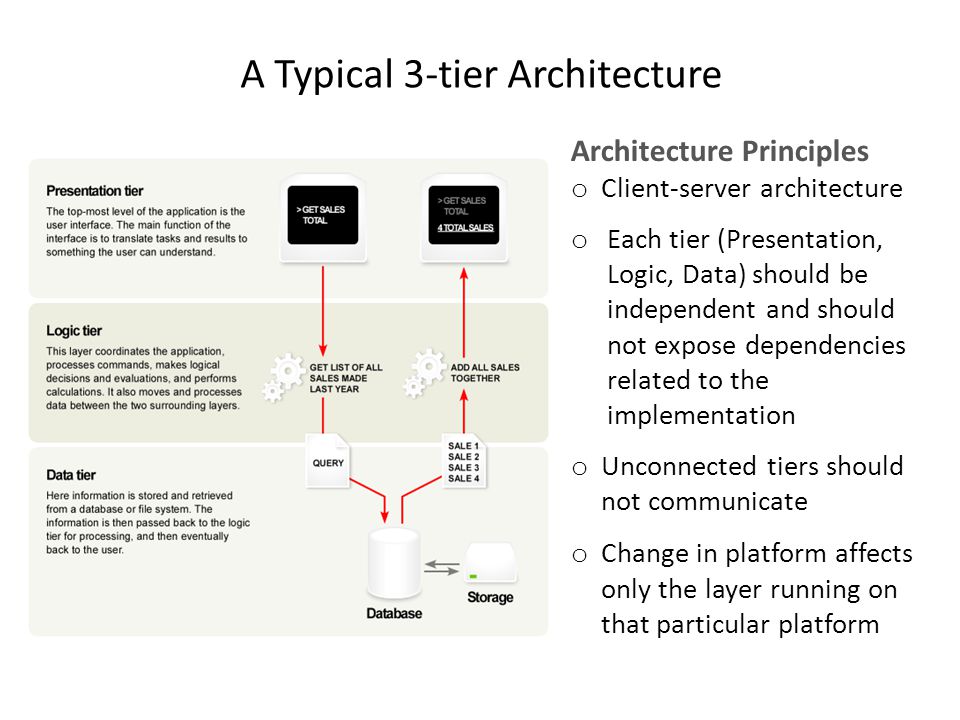
An entity relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationship that can exist between entities. An entity can be any object, place, person or class. In ER Diagram, an entity is represented by using rectangles. I have mentioned the ER diagram for my project given below:-



**Fig: - ER-Diagram**

**ARCHITECTURE**

System architecture is the type of architecture which defines the representation of whole layer. It is the conceptual model that defines the structure, behaviour and view of system.



I have chosen system architecture because it helps to enable the implementation, understanding, maintenance, repair and further development of a system. The architect describe a system in architecture in order to enables its usage, flexible, building and further development. It is organized in a ways that it supports reasoning about the structures and behaviours of the system. It is composed of three tiers which provide many benefits for production and development environments by modularizing the user interface, business logic and storage layers. A three tier structure is a client server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

**UI**

**PROTOTYPING**

DIGITAL PROTOTYPING

A digital prototyping gives conceptual design, engineering, manufacturing and sales and marketing departments the ability to virtually explore a complete product before it’s built. Digital Prototyping is used to design, iterate, optimize, validate and visualize their products digitally throughout the product development process. It goes beyond simply creating product in 3D. It gives product development team a way to assess the operation of moving parts. It helps to save money, time, compatibility, efficiency and more accuracy. Since, digital prototyping is quicker because it follows that we can get more tests in before this stage of product is abandoned.

