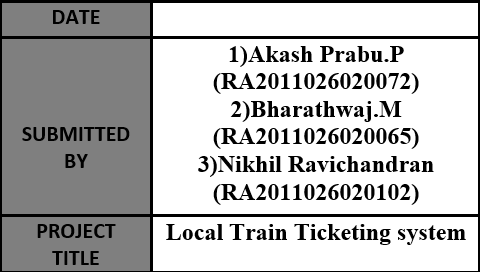
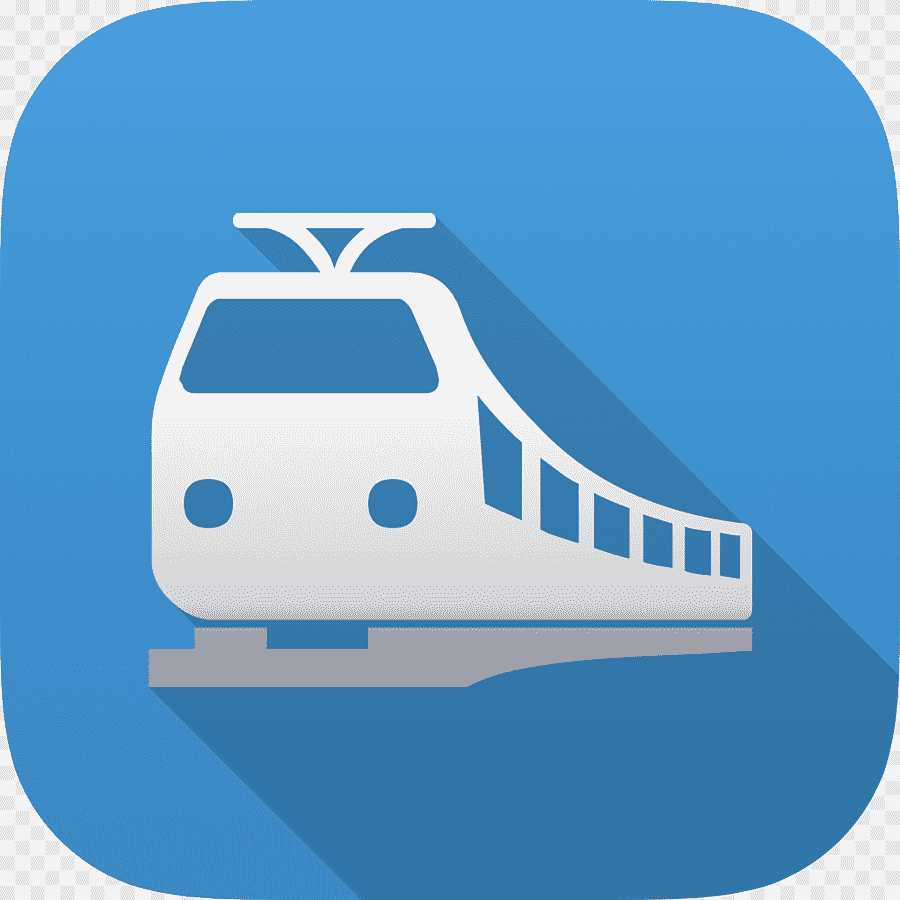
**9 ER MODELLING FROM THE PROBLEM STATEMENT**

**9.1 ER MODELLING DESCRIPTION**

**ENTITY RELATIONSHIP DIAGRAM:**

* An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.
* ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

**USES OF ER DIAGRAM:**

* Database design
* Database troubleshooting
* Business information systems
* Business process re-engineering (BPR)
* Education
* Research

**COMPONENTS OF ER DIAGRAM:**

ER Diagrams are composed of entities, relationships (Cardinality) and attributes. They also depict cardinality, which defines relationships in terms of numbers.

**1] ENTITY:**

A definable thing—such as a person, object, concept or event—that can have data stored in it.

**2] ATTRIBUTES:**

A property or characteristic of an entity.

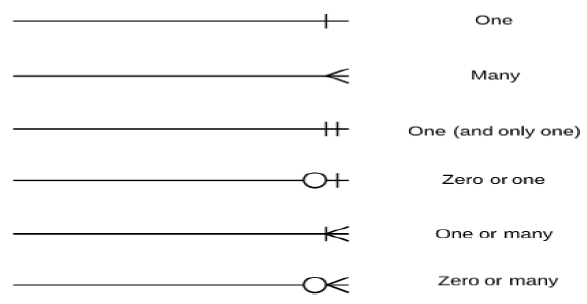
**3] KEYS:**

**PRIMARY KEY(PK):** It is unique, cannot be repeated and never null.

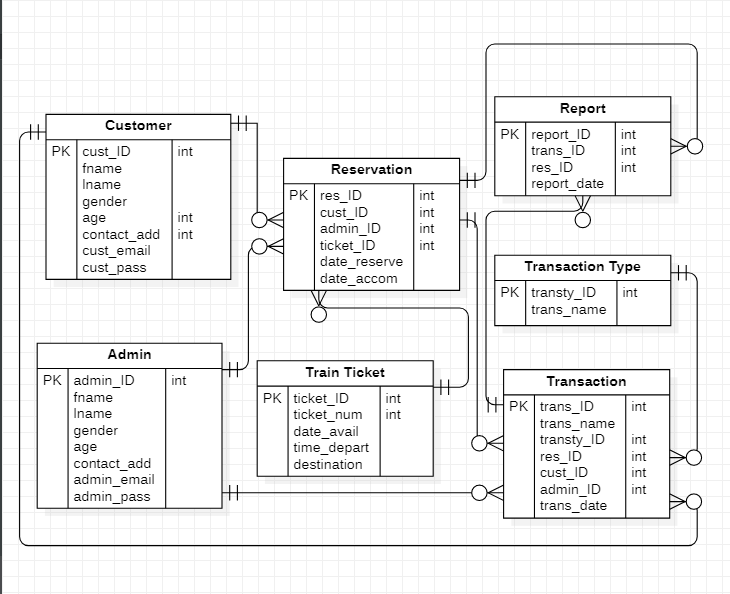
**FOREIGN KEY(FK):** It is not unique and can be repeated.

**4] CARDINALITY:**

Defines the numerical attributes of the relationship between two entities.



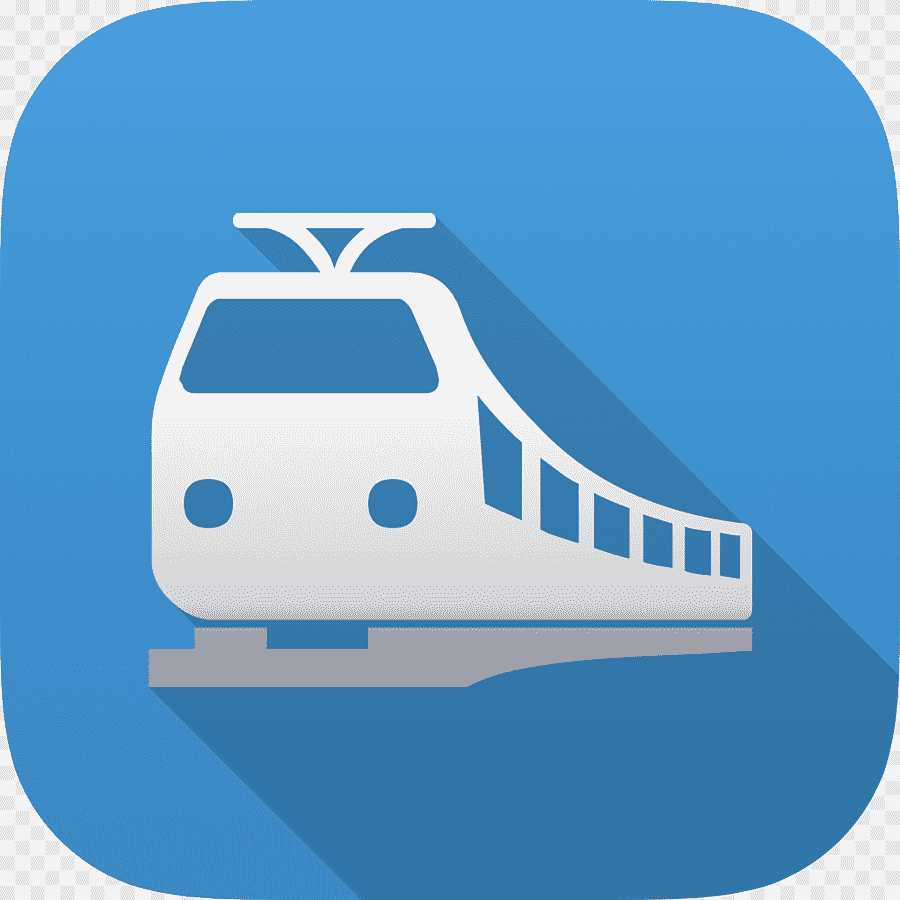
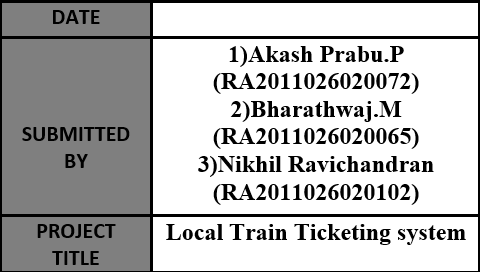
**9.2 ER DIAGRAM**

****

**In this diagram,**

**Applicant, Application, Documents, Payment, Status are the Entities.**

**11 STATECHART & COMMUNICATION MODELLING**



**11.1 STATECHART DIAGRAM DESCRIPTION**

**STATECHART DIAGRAM:**

A state diagram, sometimes known as a state machine diagram, is a type of behavioural diagram in the Unified Modelling Language (UML) that shows the transitions between various objects.

**COMPONENTS OF STATECHART DIAGRAM:**

We can include many different shapes in a state diagram, particularly if we choose to combine it with another diagram. This list summarizes the most common shapes we may encounter.

**1] START STATE:**

It is denoted by black filled circle and represents the beginning of the state.

**2] STATE:**

We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant of time.

**3] TRANSITION:**

We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.

**4] FORK:**

We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.

**5] JOIN:**

We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrow towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.

**6] DECISION:**

We use a diamond symbol to apply a condition wherever necessary.

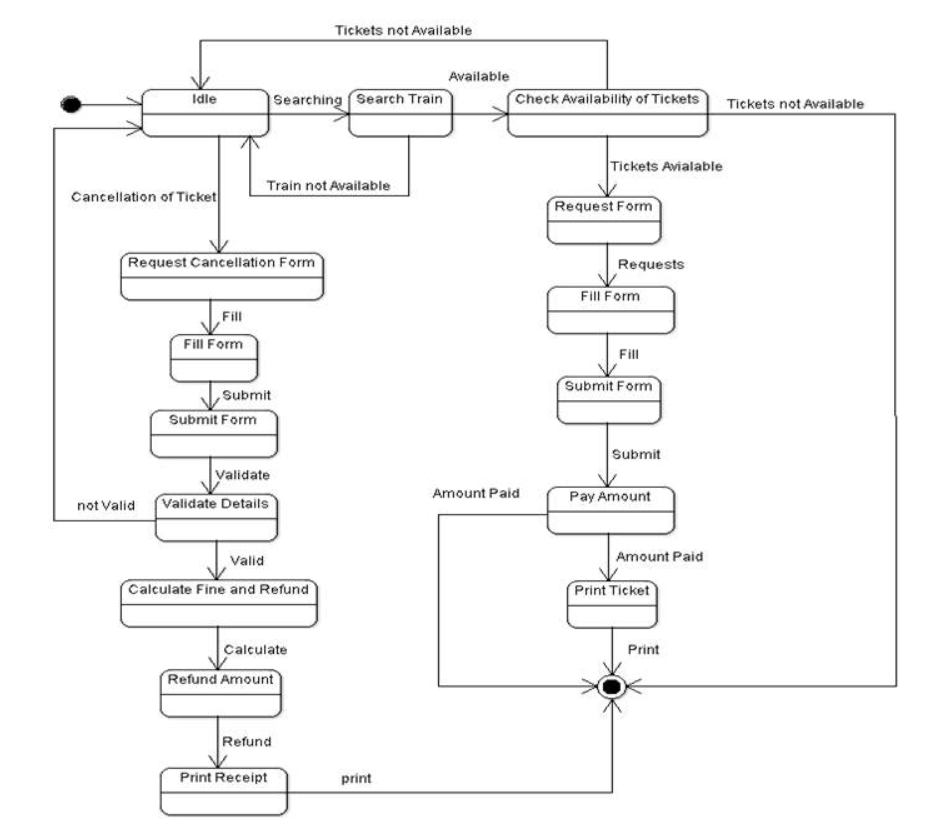
**7] COMPOSITE STATE:**

We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.

**8] FINAL STATE:**

We use a filled circle within a circle notation to represent the final state in a state machine diagram.

**11.2 STARTCHART DIAGRAM**



**11.3 COMMUNICATION DIAGRAM DESCRIPTION**

**COMMUNICATION DIAGRAM:**

Communication diagrams, formerly known as collaboration diagrams, are almost identical to sequence diagrams in UML, but they focus more on the relationships of objects—how they associate and connect through messages in a sequence rather than interactions.

**COMPONENTS OF COMMUNICATION DIAGRAM:**

**1] OBJECTS:**

Objects can be classed as either a supplier or a client. Suppliers call the function that supplies the message. Clients send the message to the supplier, who receives it. It is represented by rounded rectangle.

**2] ACTORS:**

Stick figure represents the actor. It is the instances that invokes the interaction. Each actor has a specific name and a role.

**3] LINKS:**

A straight line connecting two objects indicates a relationship between them. The two objects are able to send messages to each other.

**4] MESSAGES:**

Typically, messages will have a number and description next to them. The number determines the order in which messages should be read.

**NOTE:**

Communication diagrams shows much more than just a sequence of events. Using a communication diagram to model your system allows you to breakdown a series of complex interactions in said system.