# Introduction

Design is a graphical representation of a model .The process of transforming user requirements into some suitable analog which helps in software coding and implementation .

Software Design moves the concentration from problem domain to solution domains.

**Software Design Level**

* **Architectural Design** : It is the highest abstract version of the system that defines the software as a many interacting components .
* **High level Design :** It is a concept of architectural design into abstracted vie of modules and sub-systems and depicts their interaction. It recognizes the structure of system and their relation and interaction among the components .
* **Detailed Design :** It deals with the implementation part of what is seen as a system and in the previous two design . it defines logical structure of modules and their interfaces to communicate ith other modules .

**Modularization of Design**

Modularization is a technique to divide a software system into multiple discrete and independent module , which ae expected to be capable of carrying out tasks independently.

Advantages of modularization

* Easier to maintain smaller components
* Division program based on functions aspects
* Level of abstraction in the program
* Components can be aspects
* Concurrent execution
* Security aspects

**Verification of Design**

The output of software design process are :

* Design documentation
* Pseudo-codes
* Detailed logic diagram
* Process diagram
* Detailed description of functional and non-functional requirements

The implementation of a software depends upon the output of design

A good design is important for good software design ,accuracy and quality .

# Structural model

Structural modelling depicts the static features of a system .

They consist :

* Class diagram
* Object diagram
* Component diagram

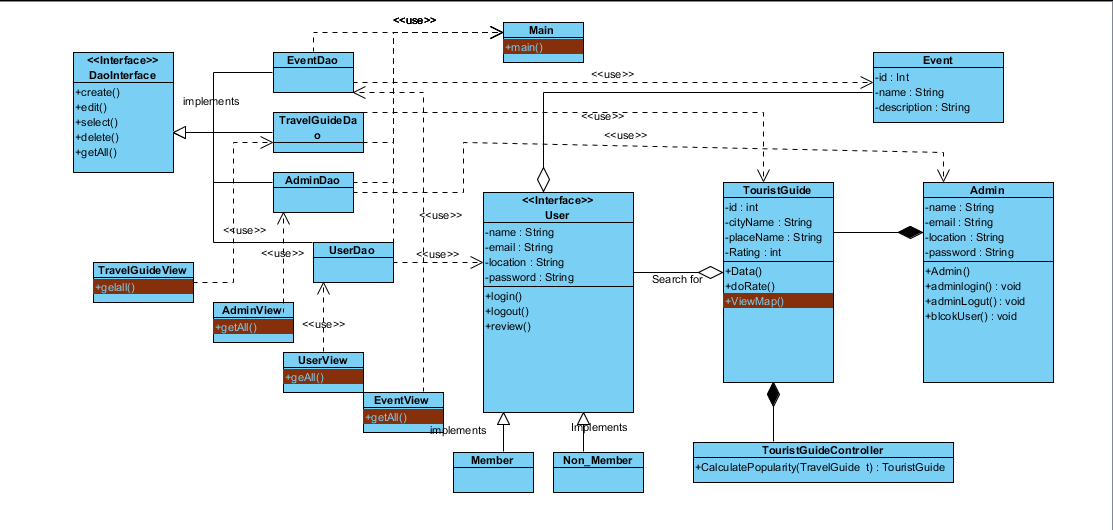
Structural model represents the framework for the system and the framework is the place where all objects and components exist. For example the class diagram , component diagram ,deployment diagram are part of structural modelling .Class diagram is widely used structural diagram . It never describes the dynamic behavior of the system .

**Class diagram**

Class diagram is a static diagram . it represents the static view of the sytem . It is not only just for describing or visualizing or documenting the different state of the system but also to construct the executable code for the system .Class diagram shows the collection of classes , interfaces assocation and constraint .It is also known as well structural diagram

The purpose of class diagram :

* Analysis and design of static vie of the system
* Elaborate responsibilities of system
* Reverse and forward engineering

****

**ER Model**

It is a conceptual model.it is the type of a relational database model based on the notion of real world entities and relation among the entities. We can depicts the real world scenario into ER model . It createsset of entites along with their attributes , a set of constraint rules and relationship among them .

**Data Dictionary**

Data dictionary is the information about data . It is often referenced as metadata repository . It is created with DFD model of software program and is expected to the updated whenever DFD is changed or updates .

Data dictionary contains information about the folloing

* Data flows
* Data Structure
* Data Elements
* Data Store
* Data processing

# Behavior Model

Behavior model describes the interaction between the object of the system . It represents the interaction among the structural diagram . it shows the dynamic nature of the system .

Some of the diagrams are :

* Activity diagram
* Interaction diagram
* Use case diagram

**Activity Diagram**

Activity diagram is an important diagram in UML to describe the dynamic behavior of the system.

Activity diagram is generally like a flowchart which represent the flow of one activity to another activity. The activity can be elaborated as an operation of the system.

The control flow is drawn from one operation to another. This flow can be branched , concurrent o sequential. Activity diagrams deals with the all type of flow control by using different notation such as fork, join, etc.

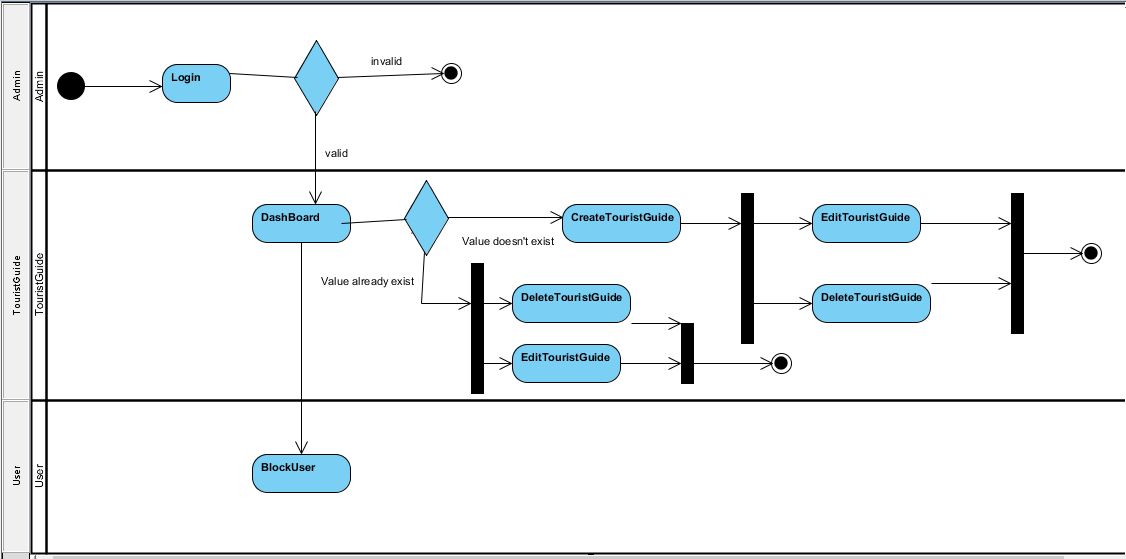
**Purpose of Activity Diagrams**

The basic purpose of activity diagram is same as other four diagrams. It depicts the dynamical behavior of the system. Other diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

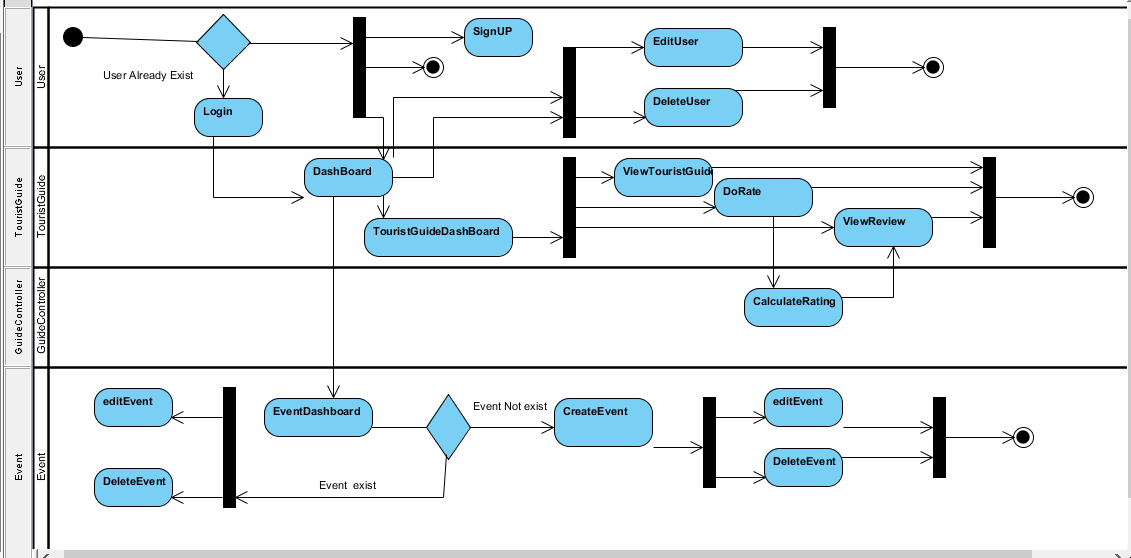
Activity is a specific operation of the system. Activity diagrams are not only used for describing the dynamic behavior of the system, but also used to construct the executable system through the use of forward and reverse engineering techniques.

The purpose of an activity diagram can be described as −

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

****

**Figure : 2 Activity**

****

**Figure 3 : activity**

**Sequence /Interaction Diagram**

By term Interaction, it is made clear the diagram is used to elaborate some types of the interactions among the varies elements or objects in the model. This interaction is a part of the dynamic behavior in system.

This interaction is represented in UML by two diagrams known as **Sequence diagram** and **Collaboration diagram**. The basic purpose of both the diagrams are similar.

Sequence diagram explains on time sequence of messages and collaboration diagram explains on the structural organization of the objects that send and receive messages.

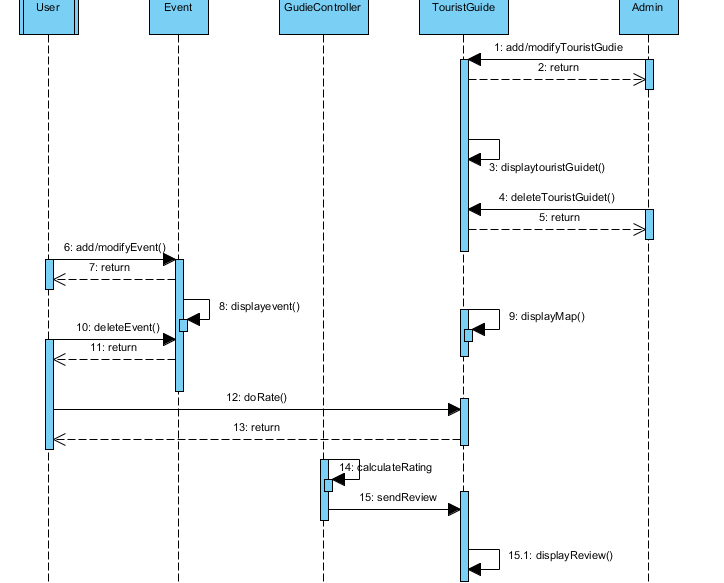
**Purpose of Interaction Diagrams**

The purpose of interaction diagrams is to visualize the interaction between objects of the system. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence diagram is used to depicts the dynamic nature but from a different point of view.

The purpose of interaction diagram is −

* To capture dynamic behavior of the system.
* To describe message flow in the system.
* To describe structural organization of the objects.
* To describe interaction among objects.

****

**Figure 4 :sequence diagram**