# Chapter 3: DESIGN

**Introduction to Design**

The basic process of realization of concept into model, configuration, pattern, plan or specification which helps to achieve certain designated objectives is also known as the term design.

**Structural Model**

It can be defined as a set of interconnected components which compositely represents a structure in accordance to defined data.

**Final Class Diagram**

In object oriented modelling Class Diagram are the main building block or component which are used to represent different objects in a system.

**Behavioral Model**

It can be defined as a set of sequential assignment statements which compositely represents a behavior in accordance to defined data.

**Flowchart (Data Flow Diagram)**

Flowchart can defined as a visual representation of sequence of steps and decisions required to perform the processes.

**Flowchart Notations**

Start Box: Represents Start/ End Point

Process Box: Represents Process Point

Decision Box: Represents a Decisive Point

Data Input/output Box: Represents input or output Point

Arrow: Represents Relationship between representative shapes

**System Flowchart**

Start

Dashboard

Connection

Hotspot Pairing

Bluetooth Pairing

Paired Successfully?

YES

NO

Add to Chat

Add Device Details

End

Start

End

Add to Group

Removed Successful

YES

NO

Confirmation?

Remove People to Group

YES

NO

Delete Group

Delete Successful

Confirmation?

Add People to Group

Edit Group Details

Add Group Details

Edit Group

Create Group

Dashboard

**Activity Diagram**

Activity diagram is the behavioral diagram in UML representation which describes dynamic aspects of the system.

**Justification:**

It is regarded as essential tool in an analyst’s repertoire.

It is easily comprehensible for both analysts and stakeholders.

Easily understandable to end users.

**Activity Notations**

Action: Represents activity or processes.

Initial: Represents starting point.

Final: Represents ending point.

Fork node: Represents multiple concurrent flows.

Join node: Represents singular flows of multiple activity.

Decision: Represents decision activity.

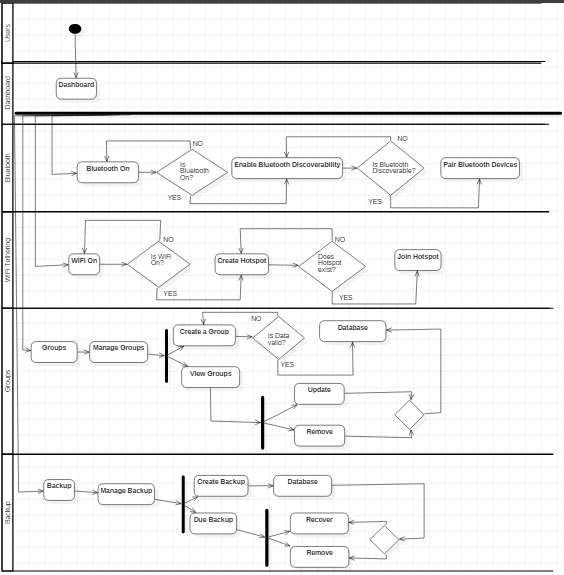
Action Flow: Represents flow of activity.

Activity Partition

Swim lane: Represents group related activities.

It can be both vertical and horizontal.

Actual Activity



This activity diagram is a simple illustration of my UMeChat Application System. Firstly User are to be upfront to dashboard where they can either make a connection through Bluetooth or WiFi tethering through which the whole files and data transferring takes place. Also User can create a group and add or exclude people from it as well as maintain backup regularly so that the files/data are stored for sustainable period.

Since it is for no respective organization and used for personal it does not require Admin and no login system.

**Sequence Diagram**

Sequence Diagram is the behavioral diagram in UML representation which defines the functionality of the system.

**Justification**

It describes how objects altogether work.

It is used by IT professionals to understand the requirement for a new system.

It is also known as event diagram.

It describes the logic of sophisticated function.

**Sequence Notations:**

Actor: It represents entity.

Lifeline: It represents time event of activities.

Object: It represents objects.

Synchronous Message: It represent flow of message at

A same instance.

Asynchronous Message: It represents flow of message at

A different instance.

Self-Message: It represents flow of message within itself.

ALT

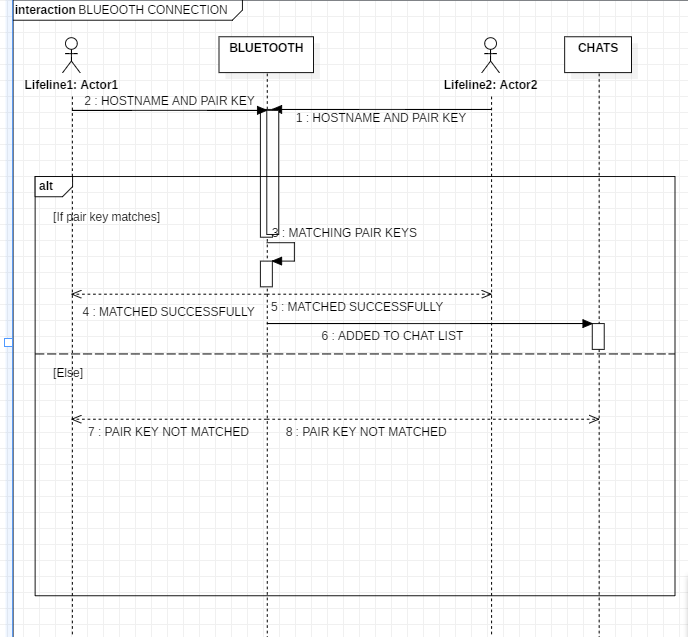
Alternative: It represents alternate activity.

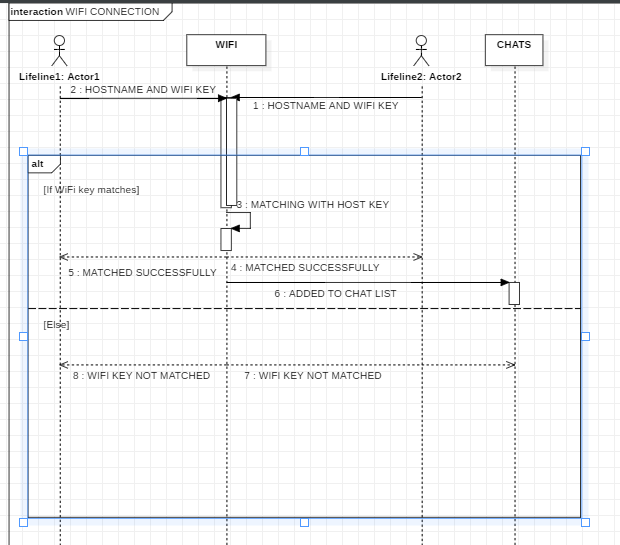
OPT

Optional: It represents optional activity.

LOOP

Loop: It represents looping activity.

Actual Sequence DiagramThis is the sequence diagram where a user can connect to the other user via Bluetooth interaction and this simply is the illustration of the sequence of Bluetooth Connection.



Whereas this is the sequential diagram for connection through WIFI which is similar to Bluetooth. In these both scenario a user can only access permission if both the defined user have paired up or match the password to host set up password through which chat system would be accessible to both of them.

**ER DIAGRAM**

Entity relationship diagram are the data structure model that represents entities and their relationship which can be one to one, one to many, many to many.

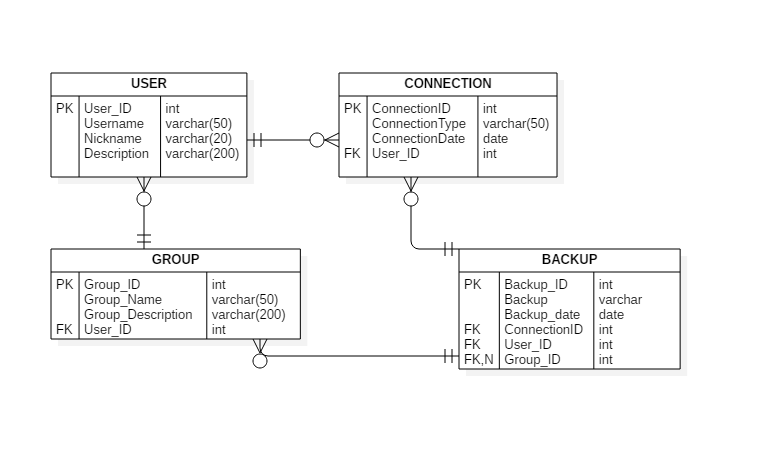


Fig: ENTITY RELATIONSHIP DIAGRAM

**DATABASE MODEL**

It is the data model which represents logical structure of database and defines how a data is to be stored.

**Data Dictionary**

It can be defined as a set of information within information I.e. it defines the contents, formats and structure of database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABLE NAME | ATTRIBUTES | DATATYPE | LENGTH | PRIMARY KEY | FOREIGN KEY | NULLABLE |
| USER | User\_ID | int |  | YES | NO | NO |
|  | Username | varchar | 50 | NO | NO | NO |
|  | Nickname | varchar | 20 | NO | NO | NO |
|  | Description | varchar | 200 | NO | NO | NO |
| GROUP | Group\_ID | int |  | YES | NO | NO |
|  | Group\_Name | varchar | 50 | NO | NO | NO |
|  | Group\_Description | varchar | 200 | NO | NO | NO |
|  | User\_ID | int |  | NO | YES | NO |
| BACKUP | Backup\_ID | int |  | YES | NO | NO |
|  | Backup | varchar |  | NO | NO | NO |
|  | Backup\_date | date |  | NO | NO | NO |
|  | ConnectionID | int |  | NO | YES | NO |
|  | User\_ID | int |  | NO | YES | NO |
|  | Group\_ID | int |  | NO | YES | YES |
| CONNECTION | ConnectionID | int |  | YES | NO | NO |
|  | ConnectionType | varchar | 50 | NO | NO | NO |
|  | ConnectionDate | date |  | NO | NO | NO |

**ARCHITECTURAL MODEL**

Architectural model can be defined as the scaling model specified to meet design of the whole system. My architectural model is followed by 3 tier architecture and they are as follows:

1. Client Layer

It is also defined as a presentation layer as because it contains User interface part of our application. It is the main designing phase.

1. Business Layer

It is also defined as a logical layer as because it contains all the business logic like data validation, calculation, etc.

It acts as an interface between Client and Data Layers.

1. Data Layer

It is also defined as a data layer as because it contains all the data manipulation methods from our database. Certain facilities like data insertion, deletion, and update are carried out in this layer.

**UI MODELLING**

User Interface is the modelling technique used by programmers for easy and effective ongoing system.

**PROTOTYPING**

It can be simply defined as sample model of the system which is made out just in case of early detection of any errors and capability of our system. I have used paper prototyping as my default prototype foundation as it is cheap and easily amendable. Also since I am working on Agile Methodology furthermore updates are possible throughout the process.

