**Table Design**

1.Table name: tb\_users

Use: To store the details of users of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | uid | Int(5) | Id of the user |
| 2 | email | Varchar(30) | Email id of user |
| 3 | password | Varchar(30) | Password of the user |
| 4 | utype | Varchar(1) | User type |
| 5 | status | Int(1) | Status of the user account |

Primary Key: uid

2.Table name: tb\_photographer

Use: To store the details of photographers in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | pid | Int(5) | Id of the photographer |
| 2 | name | Varchar(30) | Name of the photographer |
| 3 | phone | Varchar(13) | Phone number of photographer |
| 4 | uid | Int(5) | Foreign key |

Primary Key: pid

3.Table name: tb\_customer

Use: To store the details of customers in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | cid | Int(5) | Id of the customer |
| 2 | name | Varchar(30) | Name of the customer |
| 3 | phone | Varchar(13) | Phone number of customer |
| 4 | uid | Int(5) | Foreign key |

Primary Key: cid

4.Table name: tb\_admin

Use: To store the details of customers in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | aid | Int(5) | Id of the admin |
| 2 | email | Varchar(30) | email of the customer |
| 3 | password | Varchar(13) | Password number of customer |

Primary key : aid

5.Table name: tb\_media

Use: To store the photos and video of photographers in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | mid | Int(5) | Id of the media file |
| 2 | pid | Int(5) | Foreign key |
| 3 | photo | Varchar(150) | photos |
| 4 | video | Varchar(150) | videos |

Primary key : mid

6.Table name: tb\_location

Use: To store the location of photographers in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | lid | Int(5) | Id of the location |
| 2 | pid | Int(5) | Foreign key |
| 3 | location | Varchar(30) | Location of photographer |

Primary key : lid

6.Table name: tb\_booking

Use: To store the booking details in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| SI.No | Field | Data\_type | Description |
| 1 | bid | Int(5) | Id of the booking |
| 2 | pid | Int(5) | Foreign key |
| 3 | cid | Int(5) | Foreign key |
| 3 | date | date | Date of shooting |
| 4 | place | Varchar(30) | Shooting place |

Primary key : bid

**UML includes the following nine diagrams.**

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. Activity diagram
7. Statechart diagram
8. Deployment diagram
9. Component diagram

**USE CASE DIAGRAM**

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" in this context refers to a project or business that is under development or operation, such a mail-order goods sales and service web page. The Unified Modeling Language (UML) makes use of use case diagrams. a common notation for simulating systems and things in the actual world. Planning for overall requirements is one of the system objectives. Testing and debugging a software product, and verifying a hardware design Performing a consumer service, developing, writing an online help guide, or focused task Use cases in a product sales context, for instance, would include ordering of goods, catalogue revision, transaction processing, and client.

Diagram

Description automatically generated

#### SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence maps define the actions that the system's components take and in what order. These schematics are Businesspeople and software developers frequently employ documentation and understanding specifications for both current and future systems.

**ADMIN**

Diagram

Description automatically generated

**CUSTOMER**

Diagram

Description automatically generated

**PHOTOGRAPHER**

Diagram

Description automatically generated

**Activity Diagram**

Activities, states, and transitions between activities and states are all included in activity

diagrams.

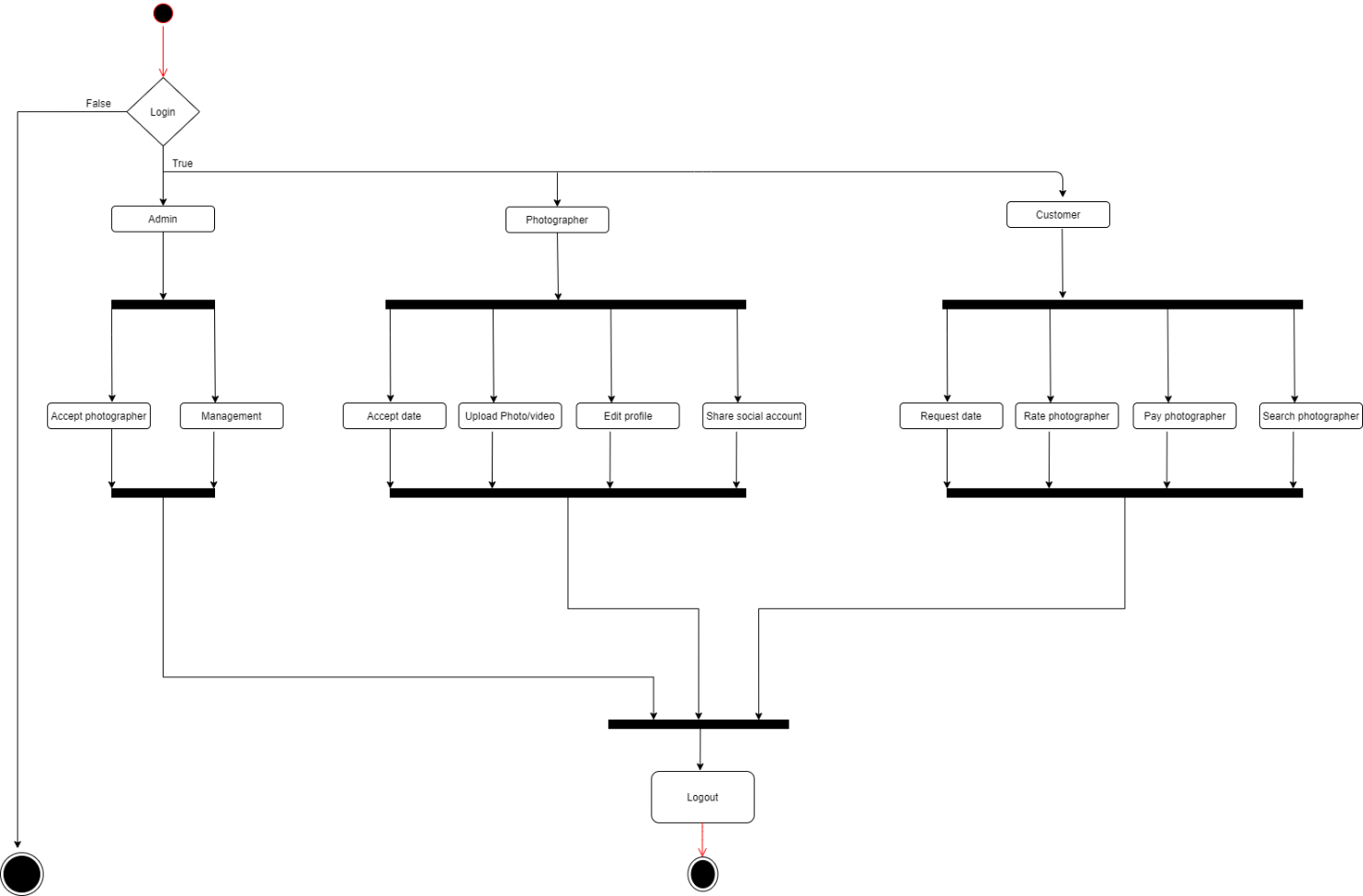
**Activities Diagrams explain**

- how a service is provided by coordinating activities

- the activities required to complete some operation.

- the connections between the events in a single use case.

- how a series of use cases work together to form a workflow for an organisation



**Statechart Diagram**

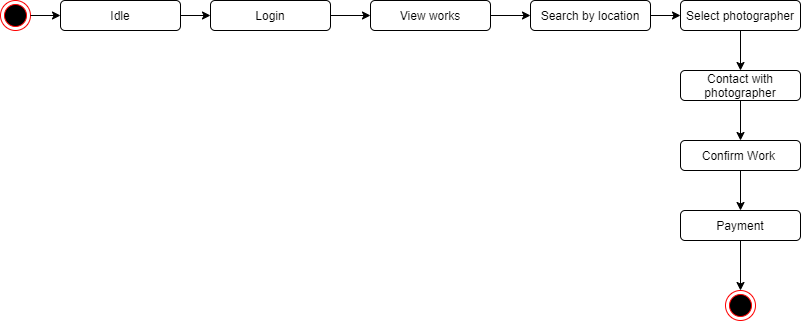
The behaviour of classes in reaction to outside stimuli is depicted in a state diagram. A state

diagram specifically shows how one object behaves in response to a sequence of system events.

It is also sometimes referred to as a state machine diagram or a Harel state chart. This UML

diagram represents the dynamic flow of control for a specific object within a system as it

changes states.



**Deployment Diagram**

Deployment diagrams show the topology of a system's physical components, where the software components are installed. Deployment diagrams are used to describe a system's static deployment view. The key elements of deployment diagrams are nodes and connections between them.

Diagram

Description automatically generated

**Component Diagram**

Component diagrams come in a variety of behaviours and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams can also be used to construct systems that can be run.

Diagram

Description automatically generated

**Class Diagram**

Class diagrams is a Static diagrams . It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly transferred to object-oriented languages, class diagrams are frequently employed in the modelling of object-oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram.

Diagram

Description automatically generated

**Object Diagram**

Class diagrams are a requirement for object diagrams because they are the source of class diagrams. An object diagram illustrates a specific instance of a class diagram. The basic concepts of class diagrams and object diagrams are the same. Object diagrams are also used to describe a system's static view, which is a snapshot of the system taken at a particular point in time. You can see a group of things and their relationships by using object diagrams.

Diagram

Description automatically generated