**CONTENT**

* 1. Introduction……........……....…………………...........................2
  2. Objective of the Project.................................................................2
  3. Project Category ………………………………………………...2
  4. Hardware and Software Requirement specifications. . ……….. 3
  5. Definition of Problem....................................................................4
  6. Functional Requirements and Technical Specifications…………5
  7. System Planning (PERT chart and GNATT chart) ……...............9
  8. System Analysis………………………………………………….10
     1. ERD …...…....……................….……………………...12
     2. DFD …………..…...............….…..................................14
     3. Use Case Diagram ..........................................................28

* 1. Database design…………………………………………………..35
  2. Number of modules and their description………………………...43
  3. Methodology Adopted, System Implementation………….………17
  4. List of report……......………..........................................................19
  5. Network Architecture……………………………………………..2
  6. Implementation of security mechanisms........................................21
  7. Future scope and further enhancement of the project…………….2
  8. Bibliography……………………………………………………….51

-------

**Introduction:**

The final task earmarked for the successful completion of M.C.A is the project work. The topic is to be selected by the student themselves or may be provided by the project guide. In consultation with my esteemed project guide, I have chosen to design and implement an internet based Photography Zone as a graphical user interface (GUI) based package. The various related activities (namely, training sales, auction, blogging, forum quering, chatting, registration, payment etc.) have been coordinated using Oracle10g as the back end tool and coding has been done in JAVA EE which is the front-end tool of my project. To represent the total system properly the help of context diagram, dataflow diagram, entity relationship diagram and use case diagram have been taken.

**Objective:**

The objective is to design and develop a digitally implemented E-Photography system with full concentration on enhancing the photographer’s comfort level by making his navigating (at home or office or on the move) a pleasant and memorable experience.

The concept has been taken from an existing photographic institution in which the entire system is implemented manually. The primary objective is to bring the task within the reach of the photographer (at the click of a mouse, so to say to kindle interest in the tech-savvy young generation, to ensure benefits of lower cost (by eliminating the need for salesmen or the property etc.) and higher volume of sales (thanks to widespread reach) and to pass part of the benefits to the customers to speed up the process of transactions through computer and also to obtain optimized error free results.

**Project Category:**

This is a fully computerized internet based photographer’s zone that falls in the category of Object Oriented Web Application with RDBMS as its back end.

**Hardware & Software Specification:**

* Hardware for client Machines

Microprocessor –i3 / 3.4 GHz

RAM – 2 GB

Hard Disk – 300 GB

Hardware for Server Machines

Microprocessor –i3 / 3.4 GHz

RAM – 4 GB

Hard Disk – 300 GB

UPS 1 KV

* Software for client Machines

Operating System – Windows 7

Browser –Internet Explorer 9+ (Java Enabled)

* Software for Server Machines

Operating System – Windows 2007 server

Front End – Internet Explorer 9+(java enabled)

Apache Tomcat 6.0 as web server

J2EE 6 or above

Database server-Oracle 10g.

* Development tools

Net beans 7.2

HTML 5

JavaScript

CSS 3

AJAX

J2SE 1.7

J2EE 7(JDBC, JSP, Servlet, EJB, JAVAMAIL API)

**Definition of the Problem**

Any computerized photography system involves proper maintenance of photographer’s record as well as photo details and the information backup of the buyers’ order details as well as transaction history. An operation of such nature on manual basis leads to various problems as follows:

1. This system is always prone to human error in keeping a large number of records.
2. Errors may creep up during preparation and issue of reports of various types, any addition & alteration in customer or product information, any deletion of user account or photo related records while using manual system.
3. Proper maintenance of detailed history of training, trainer, learner provided by the system is very difficult and time consuming in manual system.
4. Proper maintenance of detailed history of buyers and bidders and history of transactions is very difficult and time consuming in manual system.
5. Searching of photographer’s details or photo’s details is often inefficient and time consuming.
6. Appointing the appropriate Trainer for the specific level of students is often inefficient and difficult in manual system.

The new system is supposed to possess the following features:

* Very user friendly.
* Fully Web-based.
* Runs efficiently on a cheap hardware and software platform.
* Generates fast and efficient reports.
* Increases accuracy and dependability of the system.
* Increases speed of processing/operation of the system.

It is assumed that the system following would be widely appreciated by any standard photography zone or photo forum. Its simplicity and dependability are its greatest advantages.

## Various Users of the system and their functionality

## Photographer

* [Showcase Artistic Skills](file:///H:\PHOTOGRAPHY\t.html)
* [Post Photo Blog](file:///H:\PHOTOGRAPHY\t.html)
* [Access Expert Photo Forum](file:///H:\PHOTOGRAPHY\t.html)
* Chatting with others
* [Get appointment as Trainer](file:///H:\PHOTOGRAPHY\t.html)
* Get Royalty form photo rating
* [View Unrestricted Gallery](file:///H:\PHOTOGRAPHY\t.html)
* [View other Photographer's Portfolio](file:///H:\PHOTOGRAPHY\t.html)
* [Get connected with Other Photographers](file:///H:\PHOTOGRAPHY\t.html)
* [Sale his Photos in Auction Pole](file:///H:\PHOTOGRAPHY\t.html)
* [Bid Other's Photos](file:///H:\PHOTOGRAPHY\t.html)
* [Purchase Photos](file:///H:\PHOTOGRAPHY\t.html)
* Search photos

## Follower

* [Showcase Artistic Skills](file:///H:\PHOTOGRAPHY\t.html)
* [View Unrestricted Gallery](file:///H:\PHOTOGRAPHY\t.html)
* [View Photo experts' Portfolio](file:///H:\PHOTOGRAPHY\t.html)
* [Get connected with Other in Forum](file:///H:\PHOTOGRAPHY\t.html)
* View Photo Expert’s Blog
* Chatting with others
* [Bid Other's Photos](file:///H:\PHOTOGRAPHY\t.html)
* [Purchase Photos](file:///H:\PHOTOGRAPHY\t.html)
* Post comment on photo
* Send message to others
* Rating photos
* Liking photos

## Learner

* Be a learner and [Learn from famous Photo Experts](file:///H:\PHOTOGRAPHY\t.html)
* Get appropriate level of expertise to achieve his goal
* [Flourished knowledge with photo Forum](file:///H:\PHOTOGRAPHY\t.html)
* [Showcase his Artistic Skills](file:///H:\PHOTOGRAPHY\t.html)
* [Get connected with Photo Experts](file:///H:\PHOTOGRAPHY\t.html) through chatting facility
* [View photo Expert's Blog](file:///H:\PHOTOGRAPHY\t.html)
* Query for a specific information from Edu- forum
* View expert’s forum to gather information
* [View Unrestricted Gallery](file:///H:\PHOTOGRAPHY\t.html)
* [View Photo experts' Portfolio](file:///H:\PHOTOGRAPHY\t.html)
* [Purchase Photos](file:///H:\PHOTOGRAPHY\t.html)
* Bid others photo
* Post message to others
* Rating photos
* Search photos

## Administrator

* Get information about Learners
* Get information about Photographer
* Get information about Follower
* Get information about Trainer
* Get information about Photo sale
* Get information about Photo Auction
* Get information about photo rating
* Screening posted Forum
* Screening posted Blog
* Sanction royalty for photographers
* Sanction photographer’s registration
* Sanction Follower’s registration
* Sanction Learner’s registration
* Sanction Trainer’s registration
* Sanction Trainer’s payment.

**Functional Requirements and Technical Specifications**

**User Interface:**

Necessary instructions are provided using the interface as required by the application. From the project-related point of view, every screen plays an important role in overall management of activities and end users totally depend on the screen related information. The main focus is always on making the project more user-friendly. Thus, for any application to be successful it needs to have a good user interface.

A good user interface will be:

Easy to learn.

Easy to use.

Attractive.

In addition to their visual components, graphical user interfaces also make it easier to move data from one application to another. A true GUI includes standard formats for representing text and graphics. Because the formats are well-defined, different programs that run under a common GUI can share data. This makes it possible, for example, to copy a graph created by a spreadsheet program into a document created by a word processor.

**Java Platform, Standard Edition:**

Java Platform, Standard Edition (Java SE) software is the premier platform for rapidly developing and deploying secure, portable applications that run on server and desktop systems spanning most operating systems. J2SE features:

\* Portability

\* JDBC API for database access

\* Java security

**Features of Java version 7:**

Here is a brief summary of the enhancements included with the Java 7 release: Improved performance, stability and security. Enhancements in the Java Plug-in for Rich Internet Applications development and deployment.Java Programming language enhancements that enable developers with ease of writing and optimizing the Java code.

**Java Platform, Enterprise Edition 7**

Java Platform, Enterprise Edition 7 (Java EE 7) offers new features that enhance HTML5 support, increase developer productivity, and further improves how enterprise demands can be met. Java EE 7 developers will write less boilerplate code, have better support for the latest Web applications and frameworks, and gain access to enhanced scalability and richer, simpler functionality. Enterprises will benefit from new features that enable portable batch processing and improved scalability.

**EJB:**

EJB beans are specifically designed to implement the business logic of your application. As such they provide services that are often needed when implementing such logic, such as transactions, injecting of the entity manager (used for JPA, the Java Persistence API) and pooling of beans.

**The Java Database Connectivity (JDBC):**

The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases – SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access.

JDBC technology allows us to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment.

**Java Servlet:**

Java servlets are small, server-side programs that can be composed into dynamic Web applications. Servlets aren't user-invokable applications, but are hosted on Servlet containers (more about them later).

Servlet containers operate in tandem with Web servers, and invoke servlets based on requests from these servers. When deploying servlets onto Servlet containers, you can also specify canonical names to them. The Servlet container maintains the mapping between these canonical names and Servlet classes.

It can also forward the request and response objects to a JSP to create dynamic content. The service method can make use of additional utility classes and back-end layers composed of databases, Enterprise JavaBeans and more to process the request

**JSP**

JSP stands for Java Server Pages. It is a server-side technology. Java Server Pages are an extension to the Java Servlet technology that was developed by Oracle.

JSP have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements -- the actual design and display of the page -- to help make the HTML more functional (i.e. dynamic database queries).

A JSP is translated into Java Servlet before being run and it processes HTTP requests and generates responses like any Servlet. However, JSP technology provides a more convenient way to code a Servlet. Translation occurs the first time the application is run. A JSP translator is triggered by the .jsp file name extension in a URL. Jsp are fully interoperable with Servlet. You can include output from a Servlet or forward the output to a Servlet and a Servlet can include output from a JSP or forward output to a JSP.

**JAVAMAIL API:**

It is a component of JAVA EE 7 which is used to initiate an email from a java application.

**Apache Tomcat:**

Apache Tomcat is a Servlet container developed by the Apache Software Foundation (ASF). Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation. The Apache Ant software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

**Database programming:**

A database can generally be thought of as a collection of related data. In earlier database products a database was usually just a file, which contained a single table of data. In Oracle10g, a database is not necessarily tied to a file. It is more of a logical concept based on a collection of related object. A database in Oracle contains not only the raw data, it also contains the structure of the database, the security of the database, any indexes, and the other objects such as views or stored procedures related to that particular database.

A database management system (DBMS) is a collection of programs that enable user to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the process of defining, constructing and manipulating database for various applications.

There are many advantages of using databases. Some of them are:

i) Controlling redundancies and inconsistency.

ii) Restricting unauthorized access.

iii) Data independence.

iv) Enforcing integrity constraints.

In this project these approaches are used to handle data inconsistency, to protect the data from unauthorized user of the software. The databases are designed in such a way that unnecessary duplicate data can’t be stored and wastage of storage spaces is prevented. Also it is designed to ensure that only authorized users with User Ids and respective Passwords can access the data and the password of every authorized user is encrypted form user interface as well as at the back-end to introduce additional security. The “Change Password” option also can help them to protect the software from illegal user.

**Oracle10g Database:**

Oracle10g is the RDBMS product of Oracle Corporation. It is generating huge amount of interest and excitement in the market. These have become widely appreciated products in the database market. Oracle10g can run on Windows 2000 or later. Some of its features are:

Flash back query

Regular Expression

Grid Architecture

**NetBeans :**

A Java-based development environment (IDE) and platform originally developed by Sun. It includes user interface functions, source code editor, GUI editor, version control as well as support for distributed applications (CORBA, RMI, etc.) and Web applications (JSP, Servlet, etc.).

**Ajax Technology:**

AJAX is a group of interrelated [web development](http://en.wikipedia.org/wiki/Web_development) techniques used on the [client-side](http://en.wikipedia.org/wiki/Client-side) to create [asynchronous](http://en.wikipedia.org/wiki/Asynchronous_I/O) [web applications](http://en.wikipedia.org/wiki/Web_application). With Ajax, web applications can send data to, and retrieve data from,  [server](http://en.wikipedia.org/wiki/Web_server) asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the [XMLHttpRequest](http://en.wikipedia.org/wiki/XMLHttpRequest) [object](http://en.wikipedia.org/wiki/Object_(computer_science)).

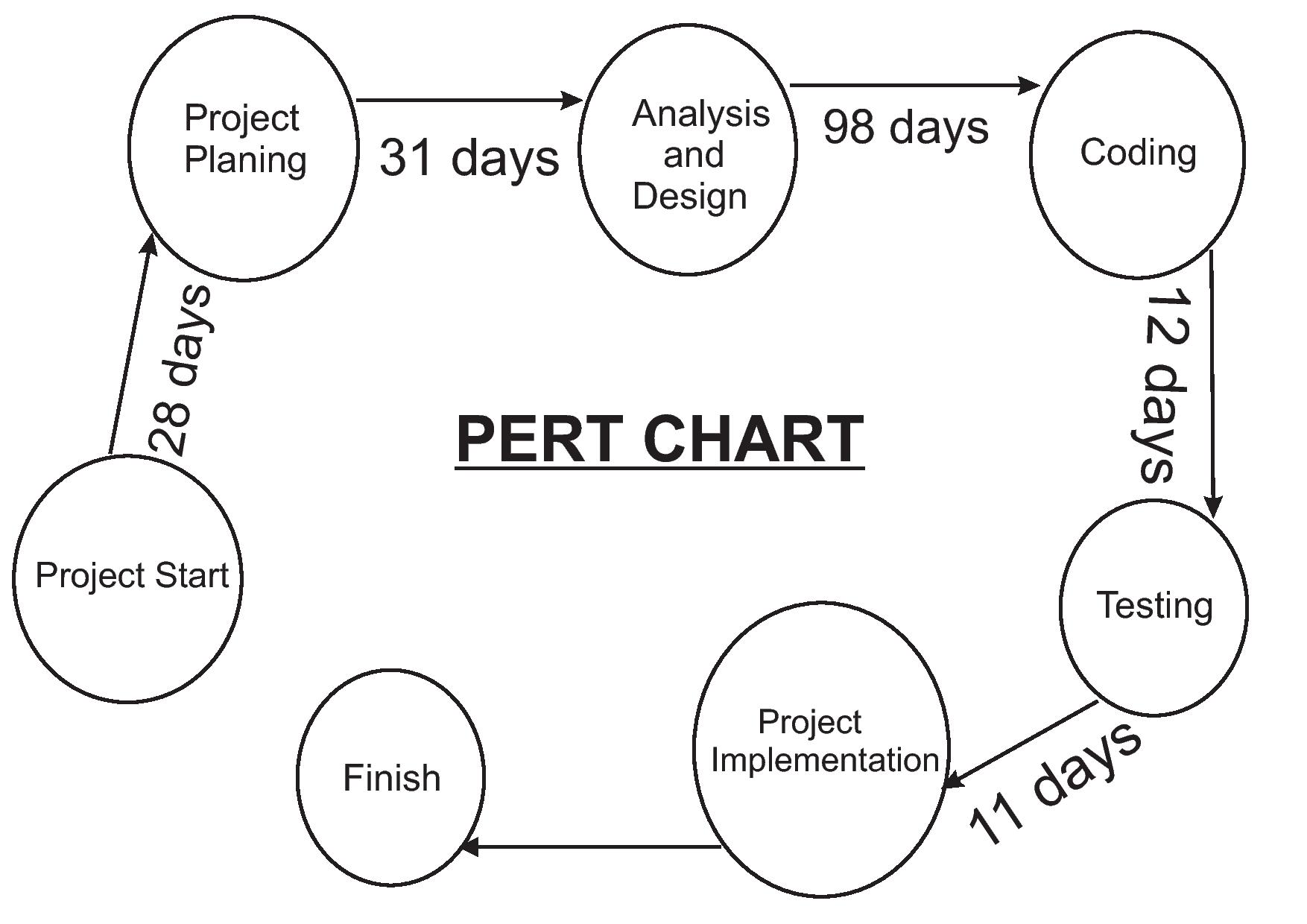
**System Planning (PERT chart)**

System planning is one of the important tasks for a system analyst to develop a system. System planning must determine how each task fits into the process. This includes stating how long a task will take, and defining its start and completion times. Moreover, a plan defines the sequence in which tasks are carried out. To make a plan, requires estimates of the effort needed to complete each of the required tasks. These estimates are then used to allocate resources to a task and in turn determine how long it will take.

Pert chart A PERT chart is a graph-based chart. It can be used to determine the activities that form the “Critical Path”, which if delayed will cause the overall project to delay. Program evaluation and review technique or PERT and critical path method or CPM is two of the most widely used techniques in project management. The objectives of project management can be described in terms of a successful project, which has been finished on time, within the budgeted cost and to technical specification, which satisfies the end users.

The total days allotted for the project to be developed is 180 days approximately. Here Saturdays, Sundays and other holidays are not taken in consideration to count the total days.

The PERT chart is described as follows:

****

**System Scheduling (GANTT chart)**

A Gantt chart is a horizontal bar chart frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that he to plan, coordinate, and track specific tasks in a project. Gantt charts may be simple versions created on graph paper or more complex automated versions created using project management applications.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the tasks that make up the project (for example, if the project is outfitting the computer with new software, the major tasks involved might be: conduct research, choose software, install software). Horizontal bars of varying lengths represent the sequences, timing, and time span for each task. The bar spans may overlap. As the project progresses, secondary bars, arrowheads, or darkened bars may be added to indicate completed tasks, or the portions of tasks that have been completed. A vertical line is used to represent the report date.

**Gantt chart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Oct** | **Nov** | **Dec** | **Jan** | **Feb** | **March** |
| **Tasks** |  |  |  |  |  |  |
| **Requirement**  **Specification** |  |  |  |  |  |  |
| **System Analysis** |  |  |  |  |  |  |
| **System Design** |  |  |  |  |  |  |
| **Coding** |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |
|  | **Oct** | **Nov** | **Dec** | **Jan** | **Feb** | **March** |

**ENTITY RELATIONSHIP DIAGRAM**

An entity relationship diagram is a detailed logical representation of data introduced by “Chean”. It is used to obtain a conceptual model of data used in an application. The ER model is expressed in terms of entities and the attributes or properties of both the entities and their relationship.

The basic notations used to create an ERD are as follows:

**ENTITY**

Data objects are represented as a labeled rectangle.

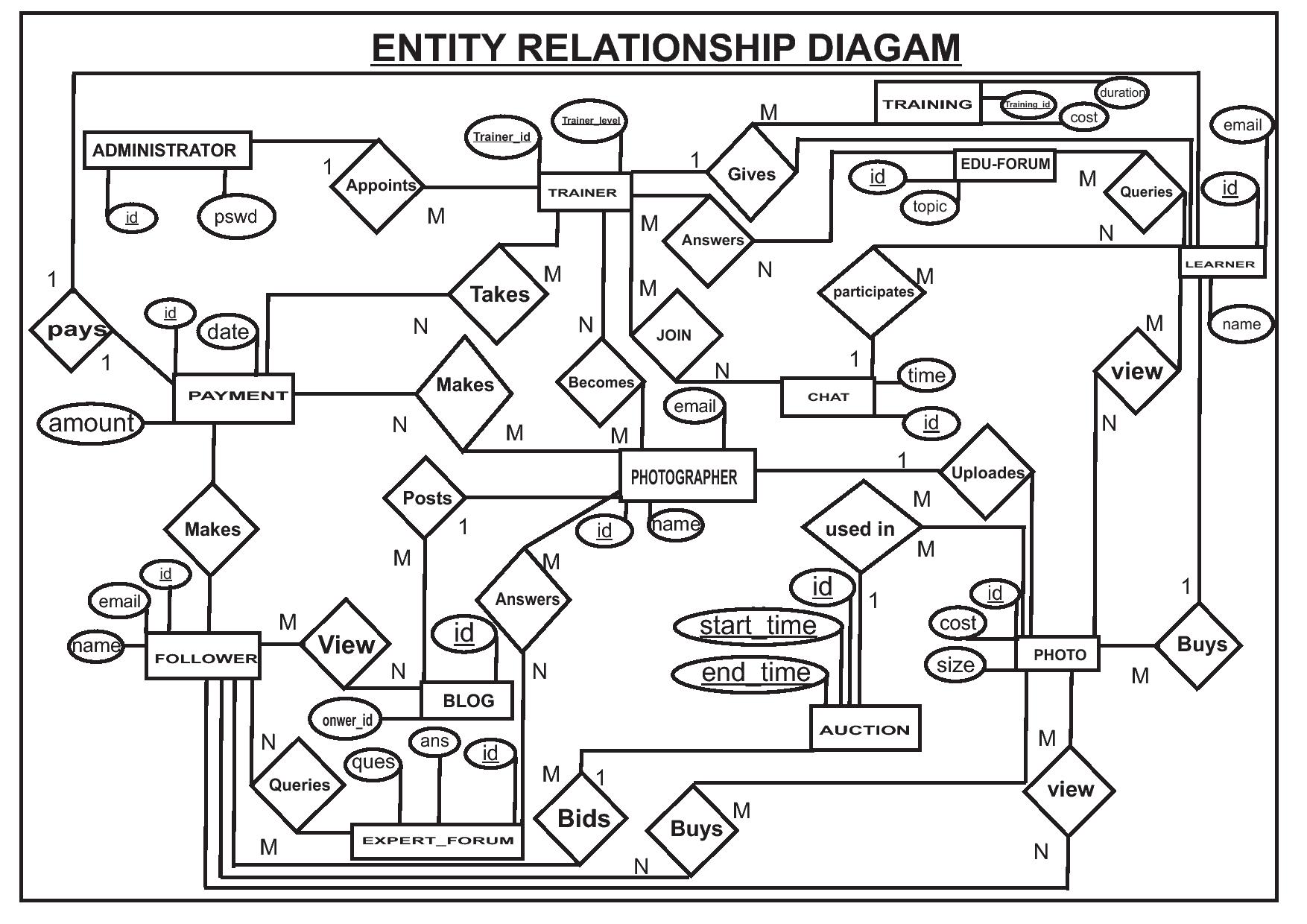
**RELATION**

Relationships are indicated with a labeled line or

diamonds connecting objects.

**ATTRIBUTE**

The attributes are shown using labeled ellipse.



**Data Flow Diagram (DFD)**

A data flow diagram is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output. It is also known as data flow graph or bubble chart. The number of bubbles or flows increases as DFD is partitioned to reveal more detail.The purpose of Data flow diagramming is to provide a semantic bridge between users and system developers, the diagram are as follows:

* Graphical, eliminating thousands of words;
* Logical representation, modeling what a system does, rather than physical model showing how it does it;
* Hierarchical showing systems at any level of details; and
* Jargon less, allowing user understanding and retrieving.

• **Dataflow diagrams have the objective of avoiding the cost of**:-

* User / developer misunderstanding of a system, resulting in a need to redo system or in not using system.
* Having to start documentation from the scratch when the physical system changes since the logical system, what gets done often remains the same when technology changes.
* System inefficiency because a system gets “computerized” before it gets “systematized”.
* Being unable to evaluate system project boundaries or degree of automation, resulting in a project in appropriate scope.

The basic notations used to create a DFD are as follows:

EXTERNAL ENTITY This is a consumer or producer of information that

resides outside the bounds of the system to be modeled.

PROCESS

A transformer of information that resides within the

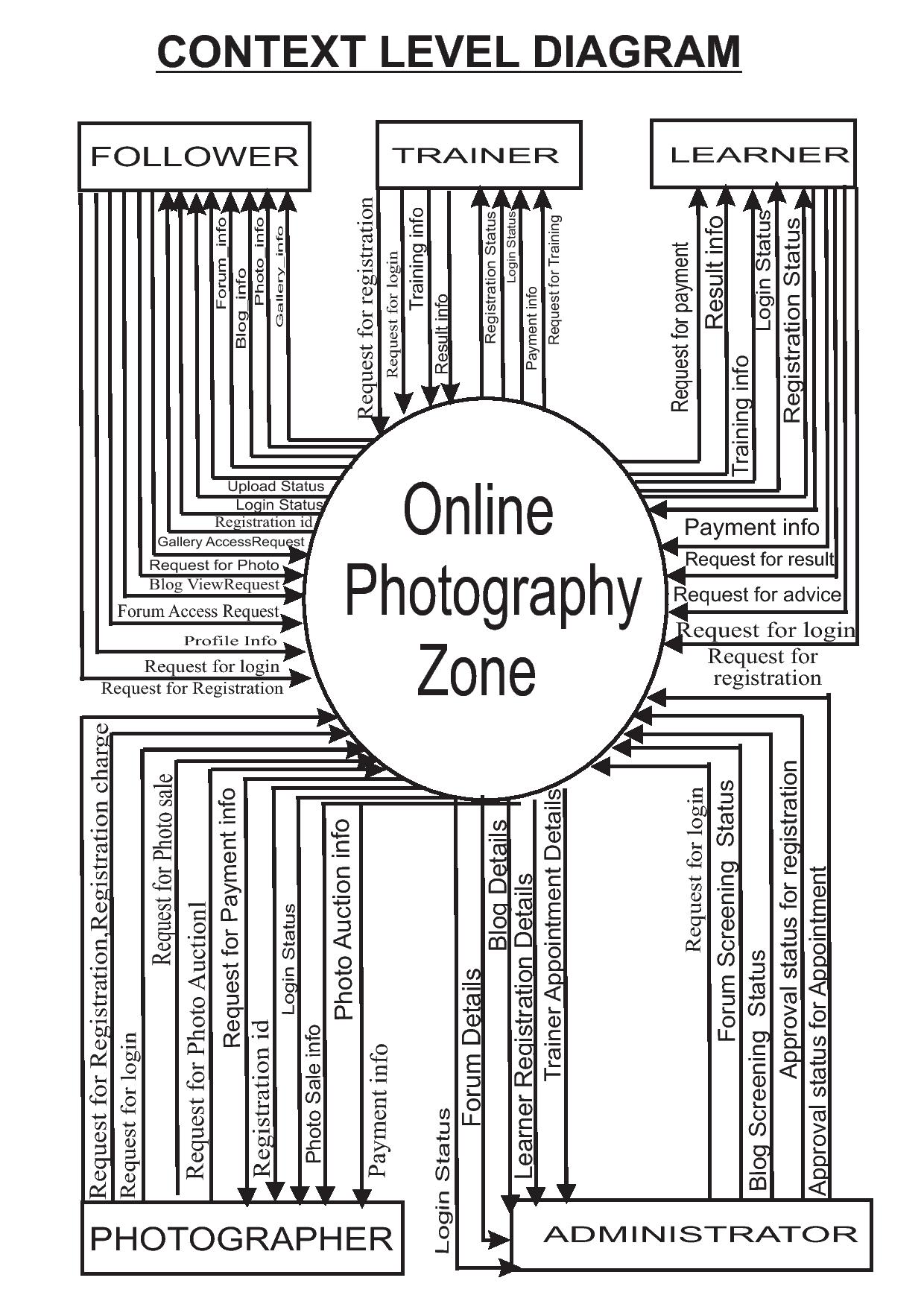
bounds of the system to be modeled.

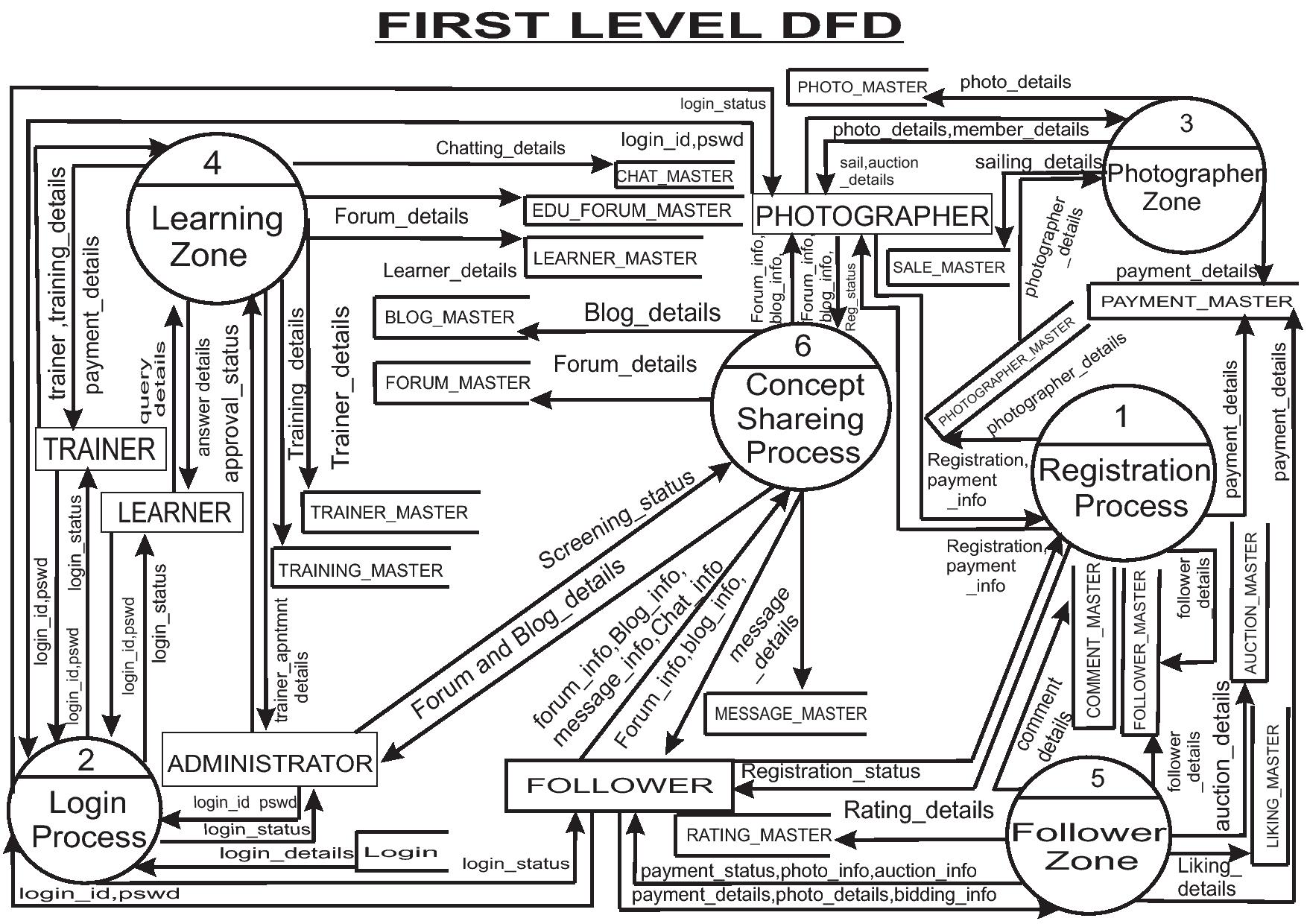
DATA FLOWA data object, the arrowhead indicates the of flow.

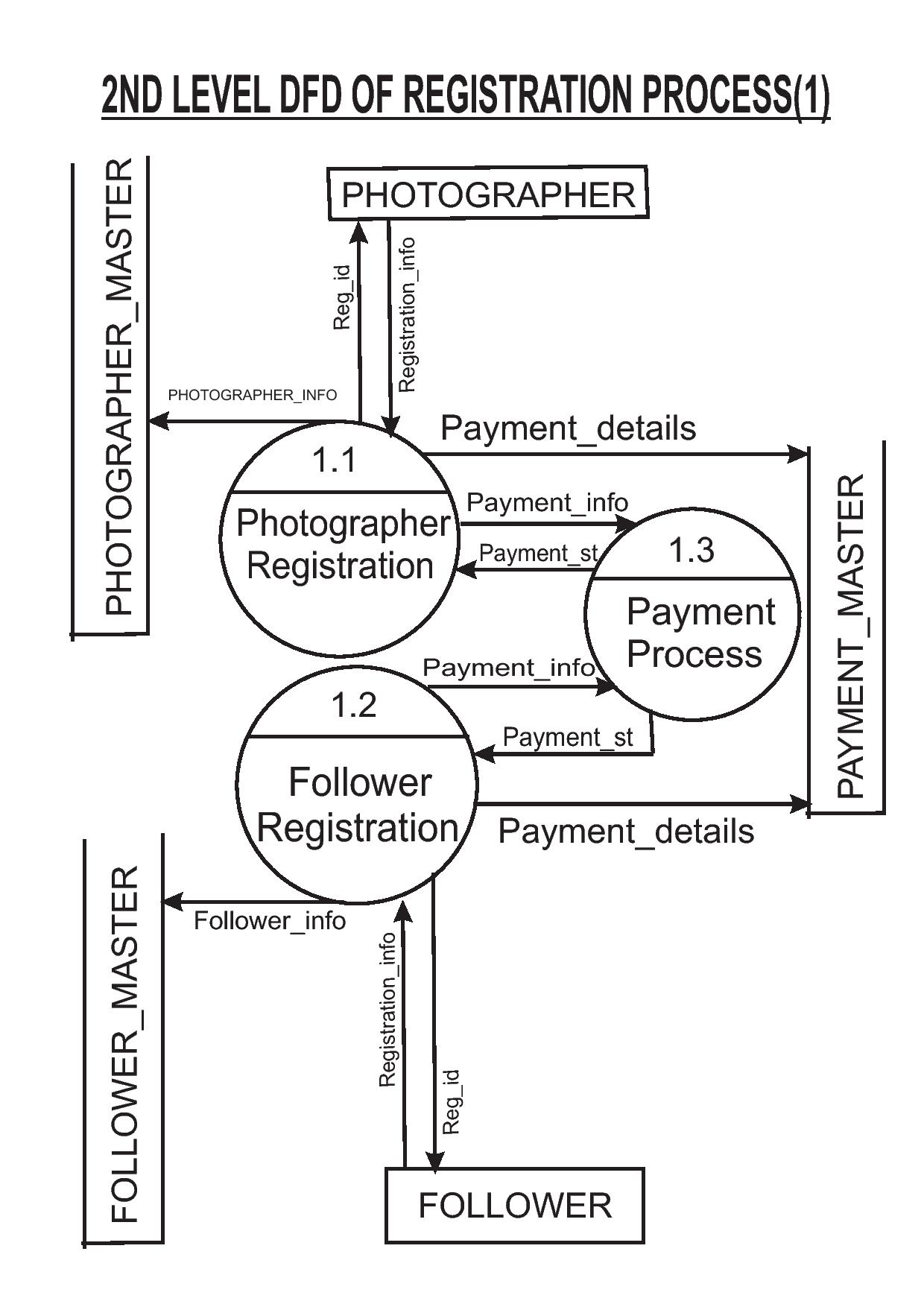
DATA STOR A repository of data that is to be stored for use by one or more processes; may be as

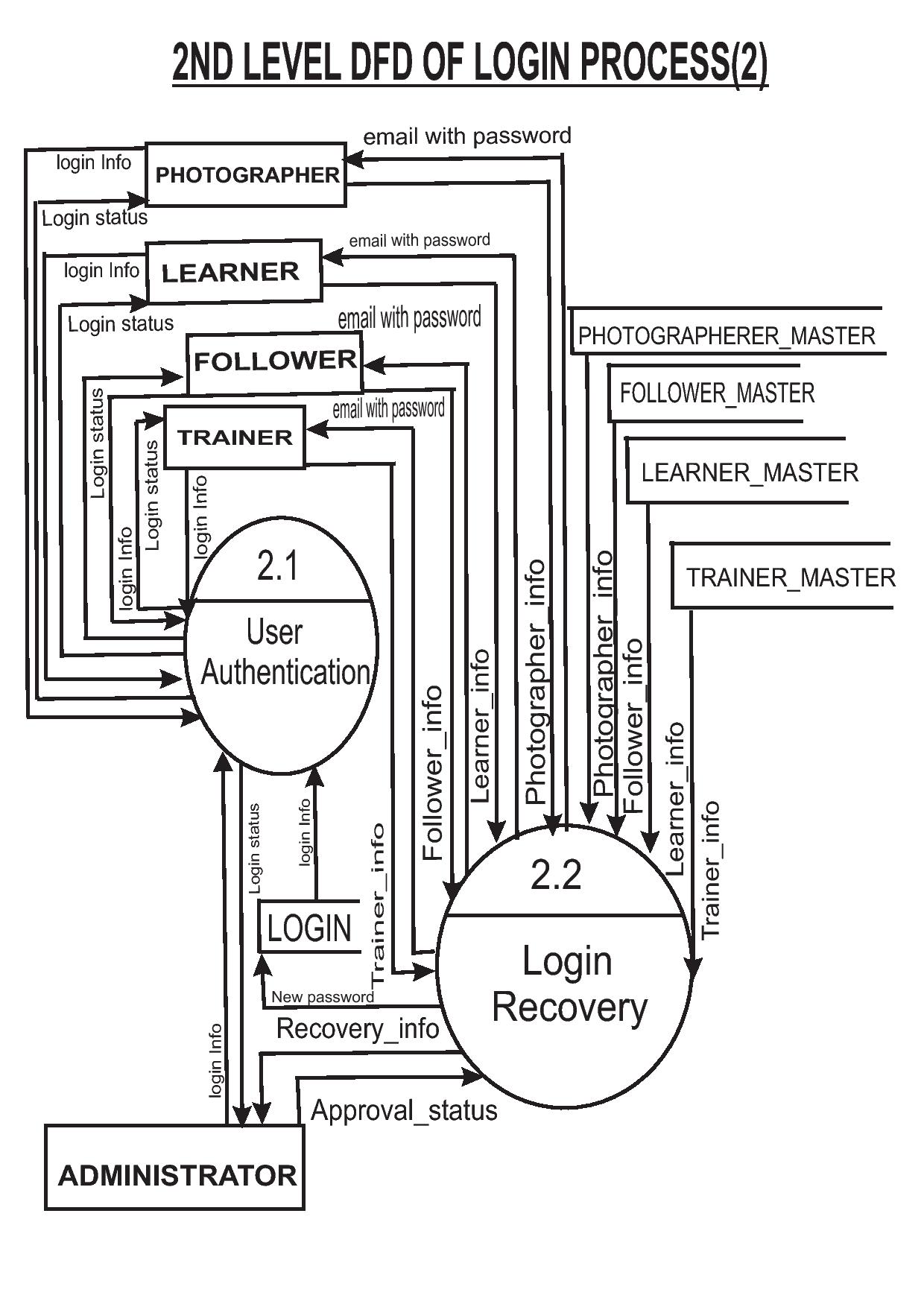
simple as a buffer or queue or as sophisticated as a relational database.

* **EXTERNAL ENTITIES:**
* Are named with appropriate name.
* Can be duplicated, one or more times, on the diagram to avoid line crossing.
* Determine the system boundary. They are external to the system being studied. They are often beyond the area of influence of the developer.
* Can represent another system or subsystem.
* Go on margins / edges or data flow diagram.
* **DATA FLOWS:**
* Are represented with a line with an arrowhead on the head on the end. A fork in a data flow means that the same data goes to two separate destinations. The same data coming from several locations can also be joined
* Should only represent data, not control.
* Are always named. Name is not including the word “data”.
* Are referenced by a combination of the identities of the constructs that the data flow connects.
* **DATA STORE:**
* Are generic for physical files (index cards, desk drawers, magnetic disk, magnetic tapes, shirt pocket, human etc.
* Can be duplicated, one or more times, to avoid line crossing.
* Can so to or more systems that share a data store? Adding a solid strip on the left boundary does this. This can occur in the case of some system updating data store, while the other system only access the data. For example, the data store could be a fright rate builds and maintains, but is used by the represented system.
* Are detailed in the data dictionary or with the data description diagram.
* **PROCESS:**
* So data transformation or changed. Data coming into a process must be “worked on” or transformed in some way. Thus, all process must have inputs and outputs. In some case, data inputs and outputs will only be shown at more detailed levels of the diagram. Each process is always “running” and ready to accept the data.
* Are represented by a circle.
* Are named with one carefully chosen verb and object of the verb.
* Name is not including the “process”. Each process is one function or action. If there is an “and” in the name, you likely have more than one function (process).
* Should generally move from top to bottom and left to right.

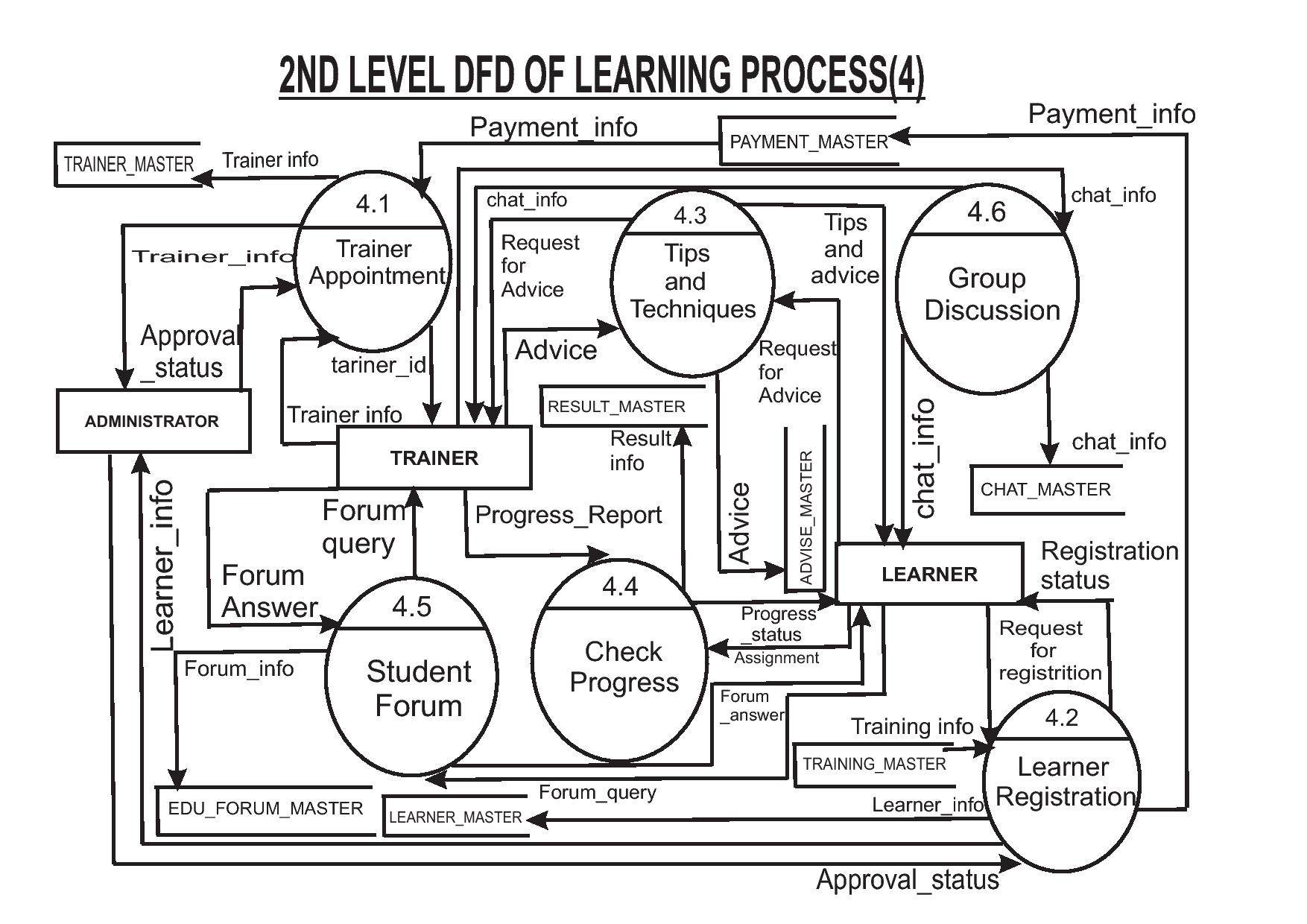


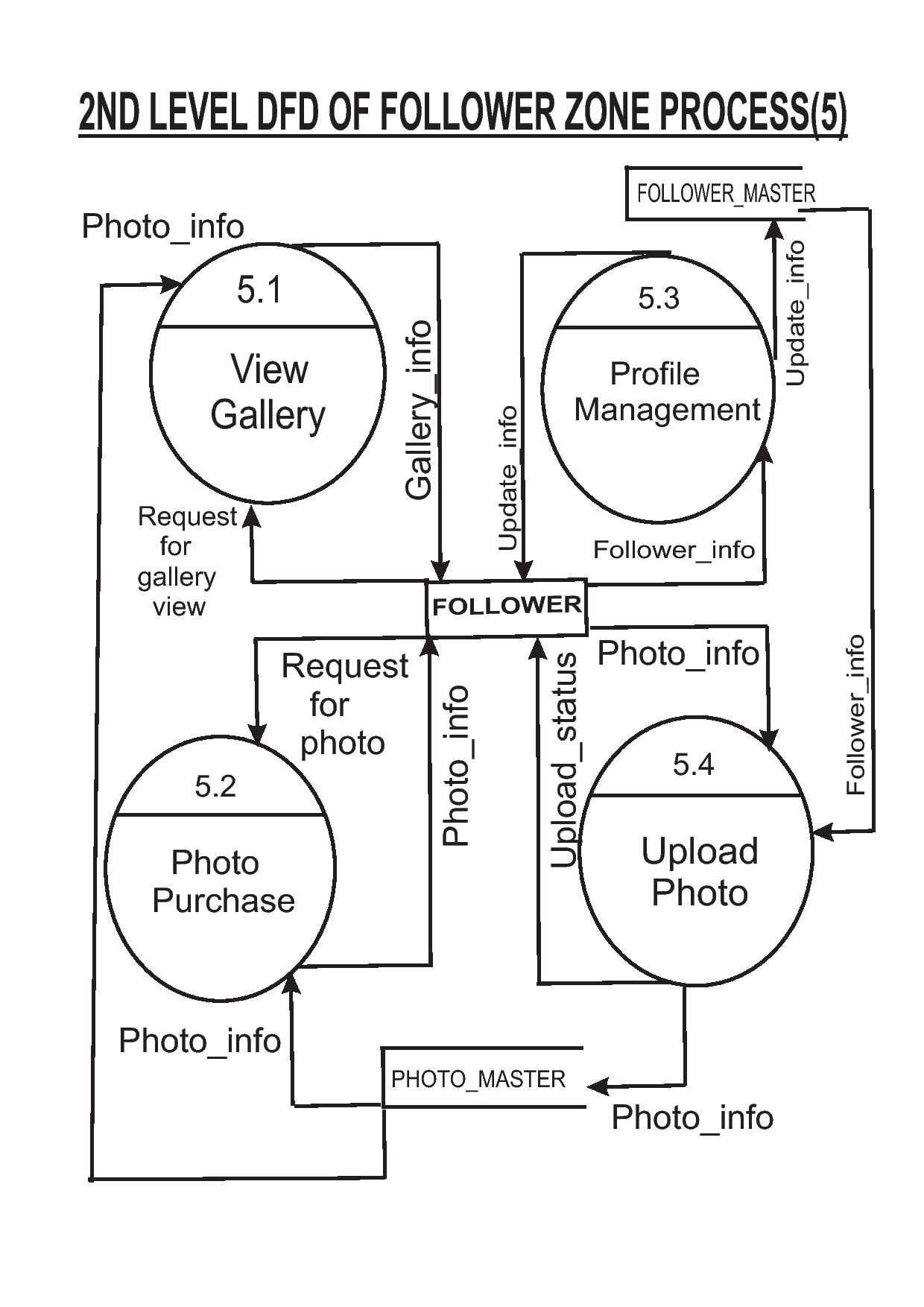


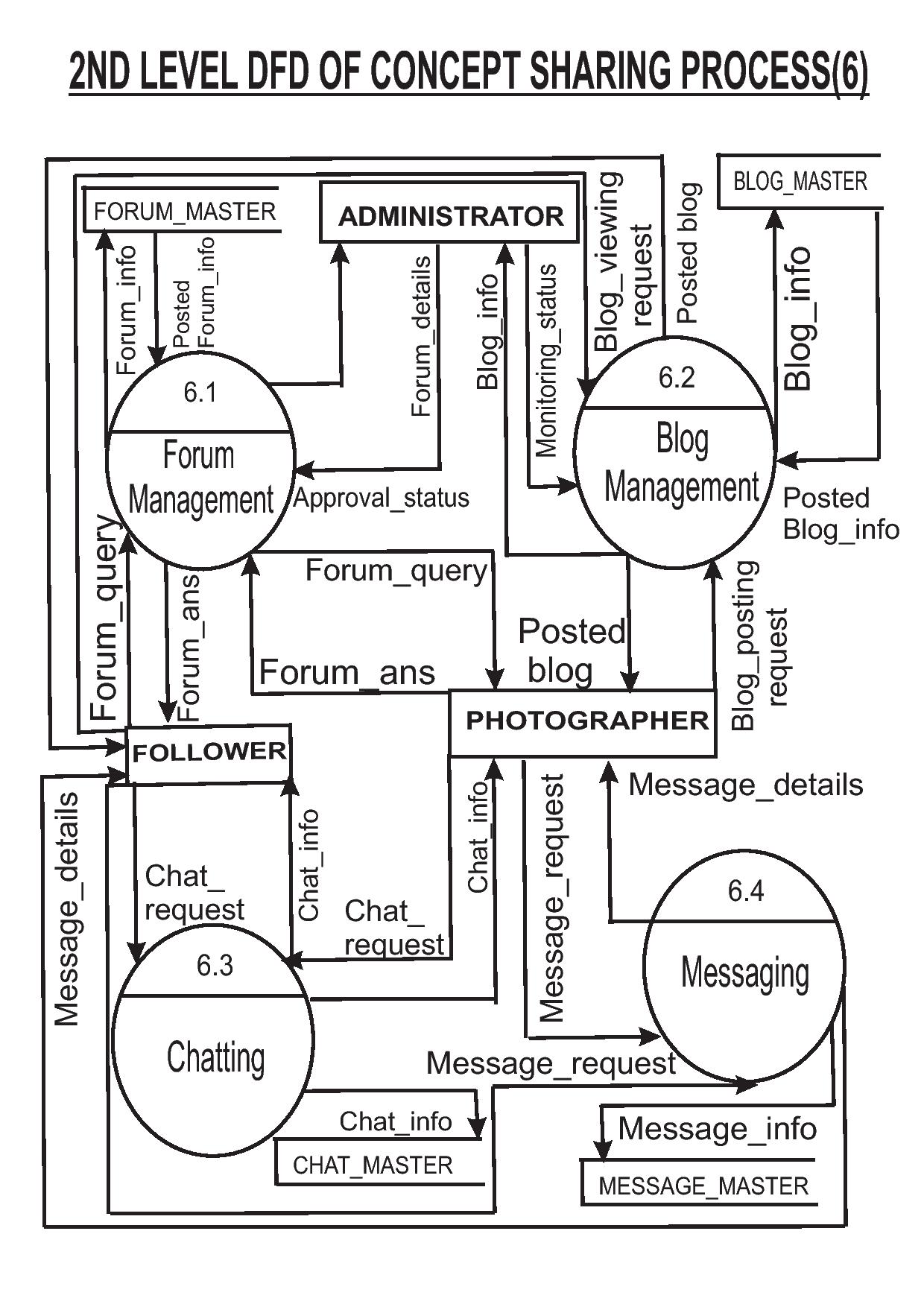


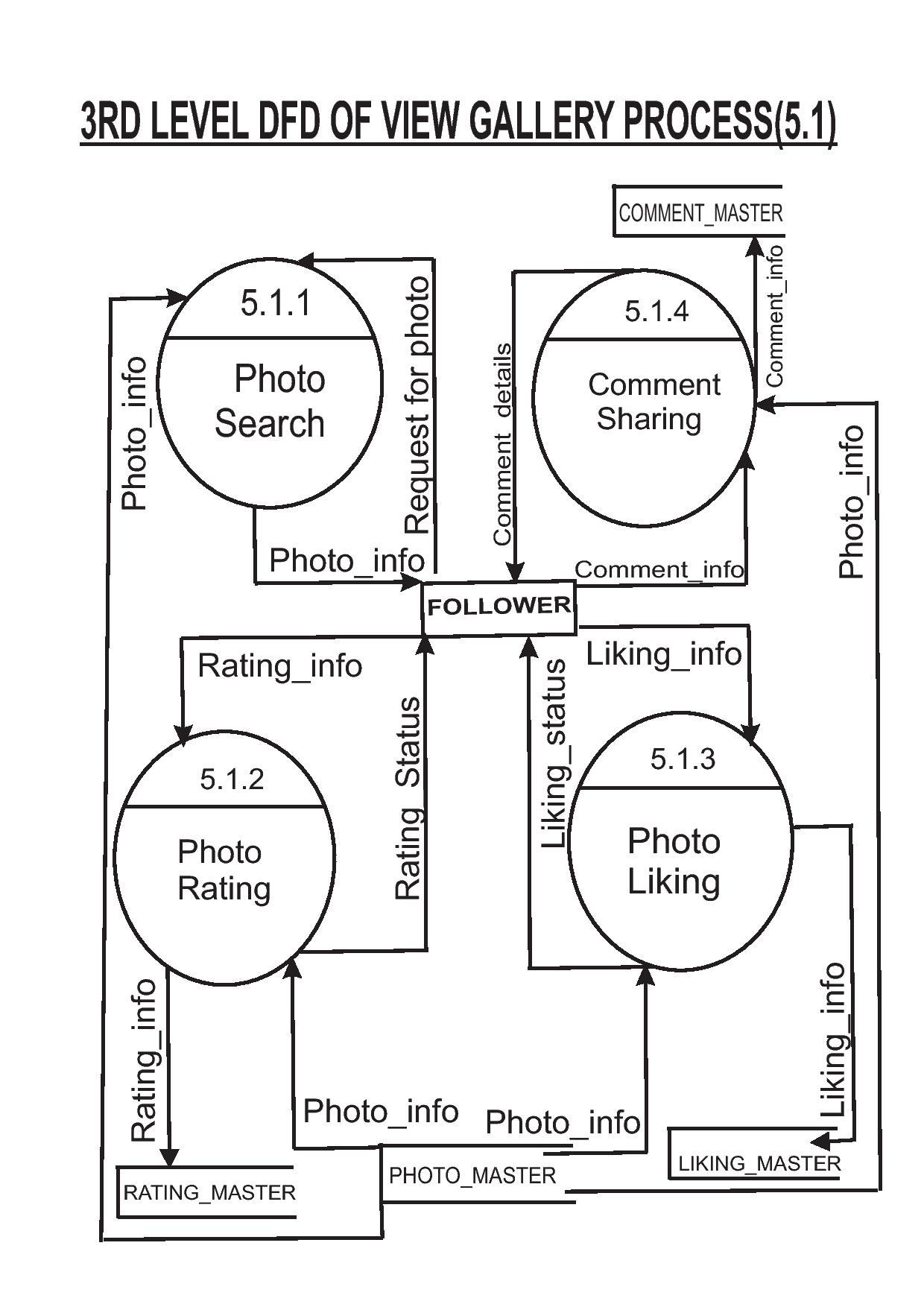


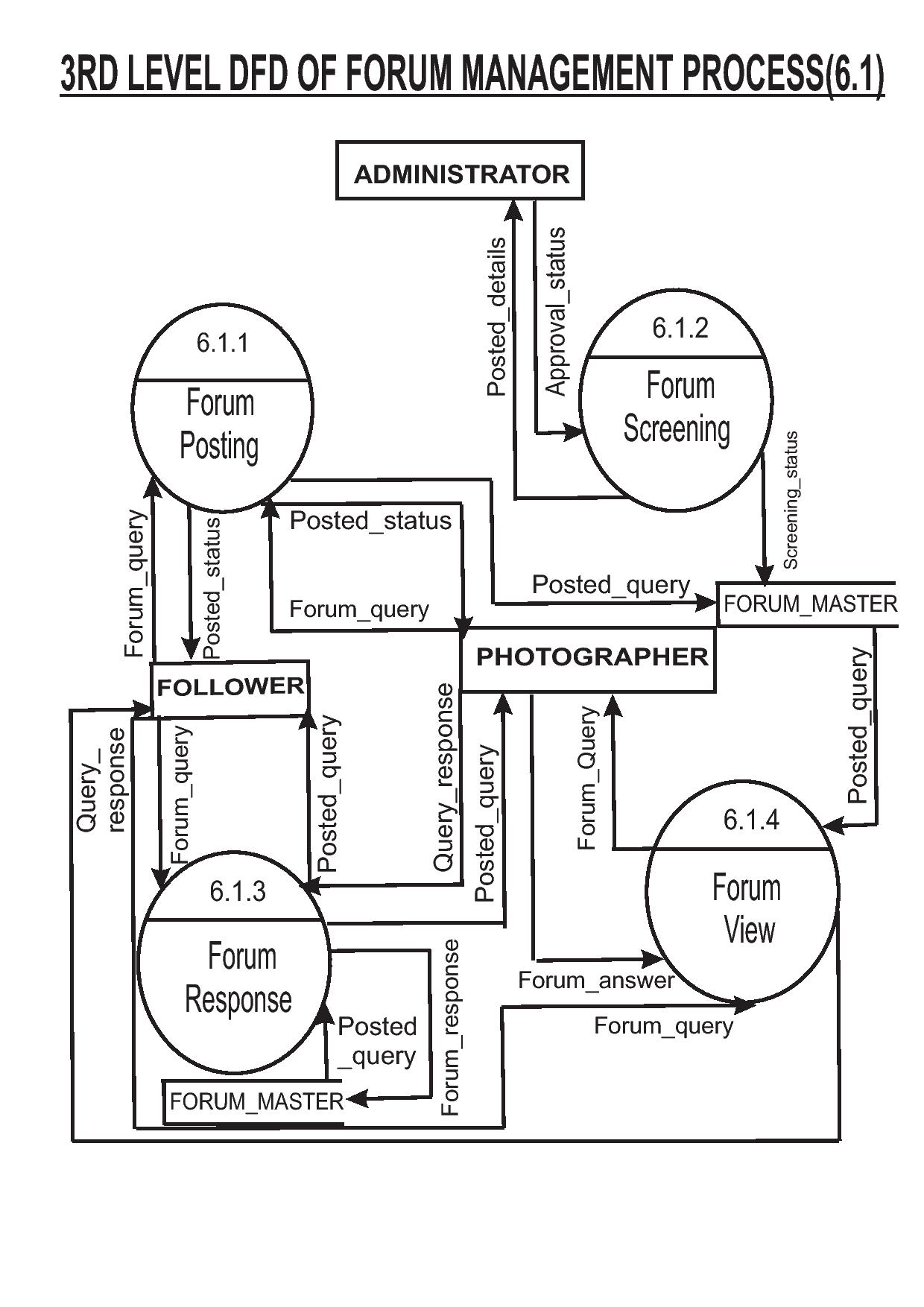
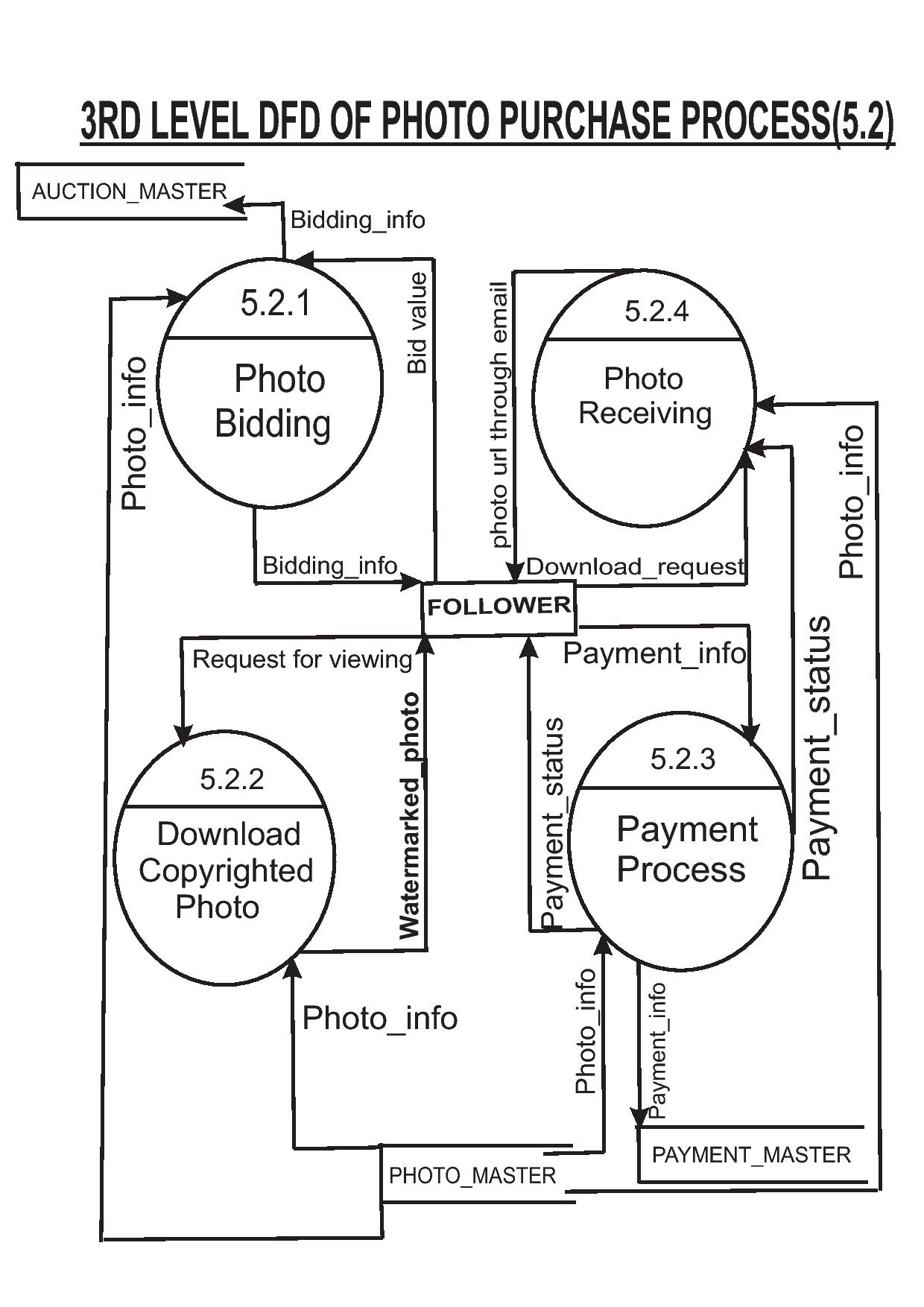


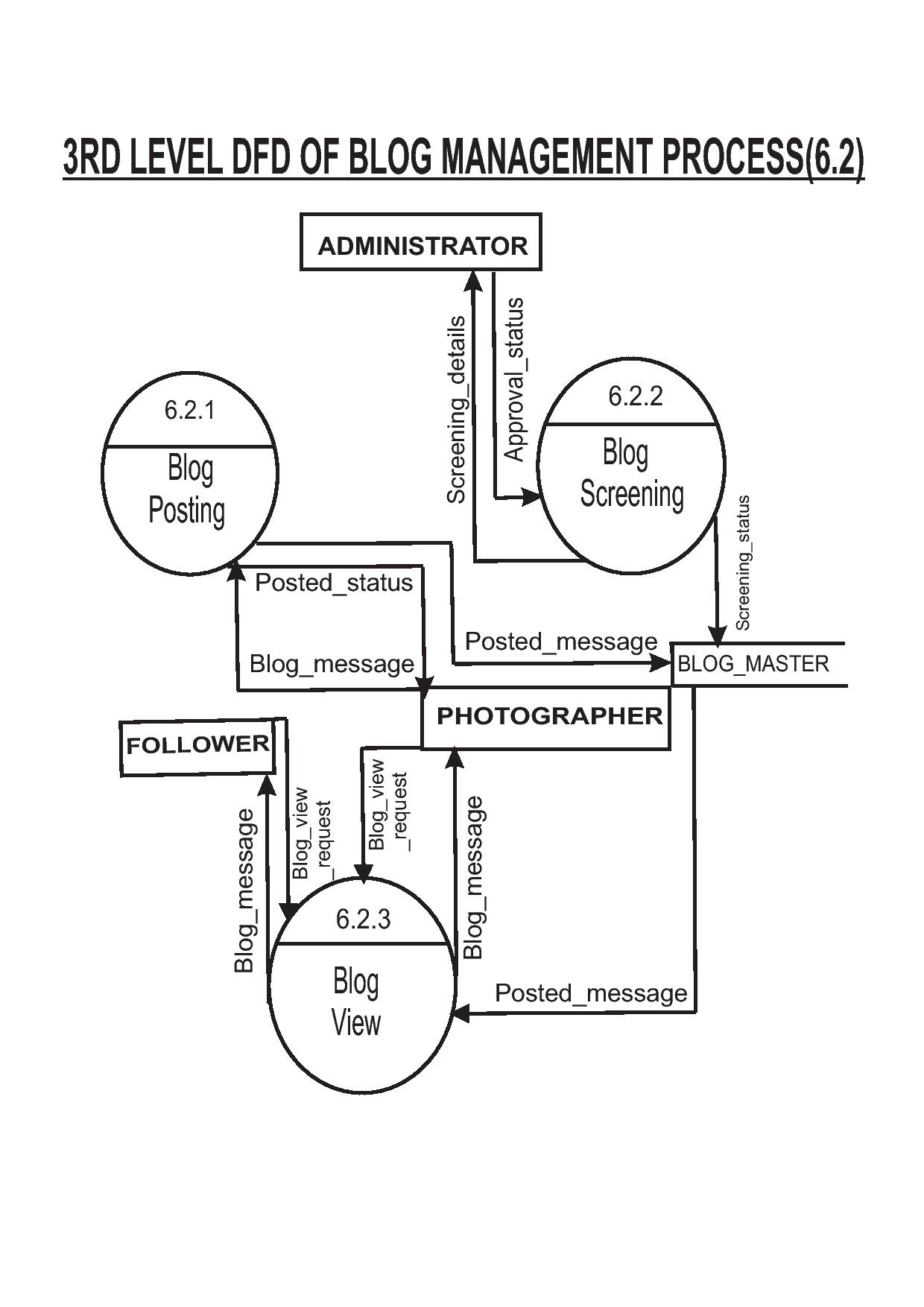


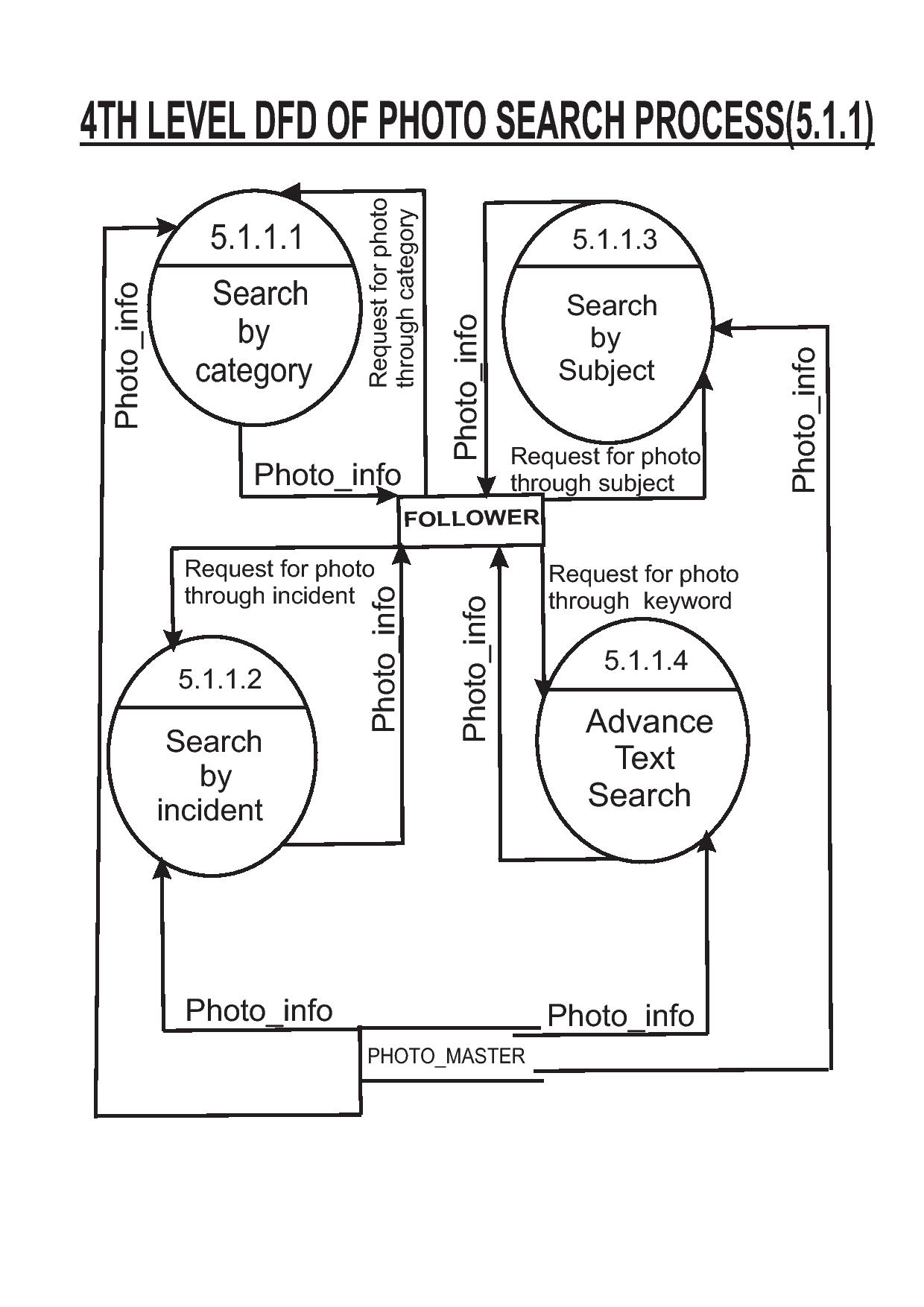










****

**Use Case Diagram**

In software engineering, a use case diagram in the Unified Modeling Language

(UML) is a type of behavioural diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in

terms of actor and their goals (represented as use cases). The main purpose of a use case diagram is to show what system functions are performed for which actors. Roles of the actors in the system can be depicted.

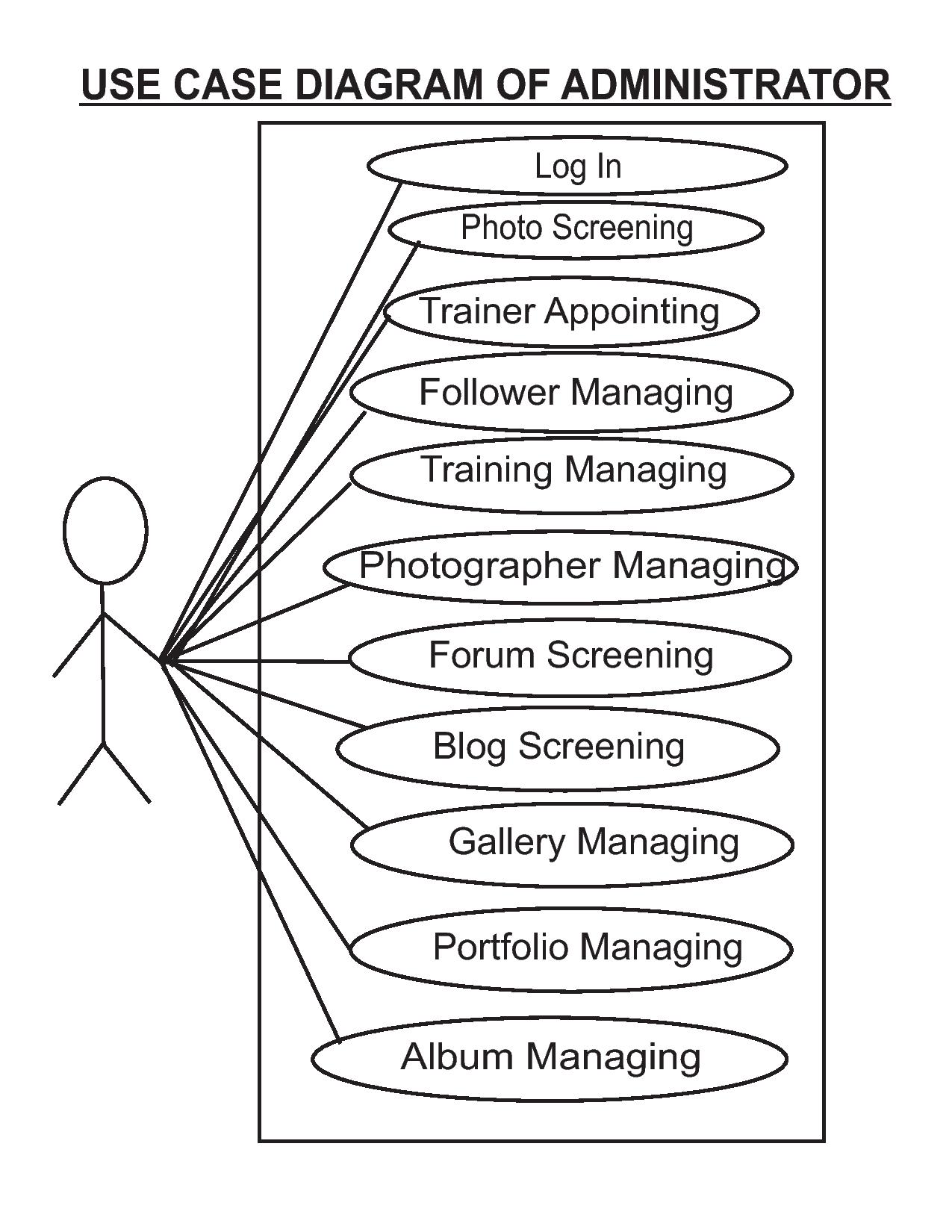
A use case is a set of scenarios describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases.

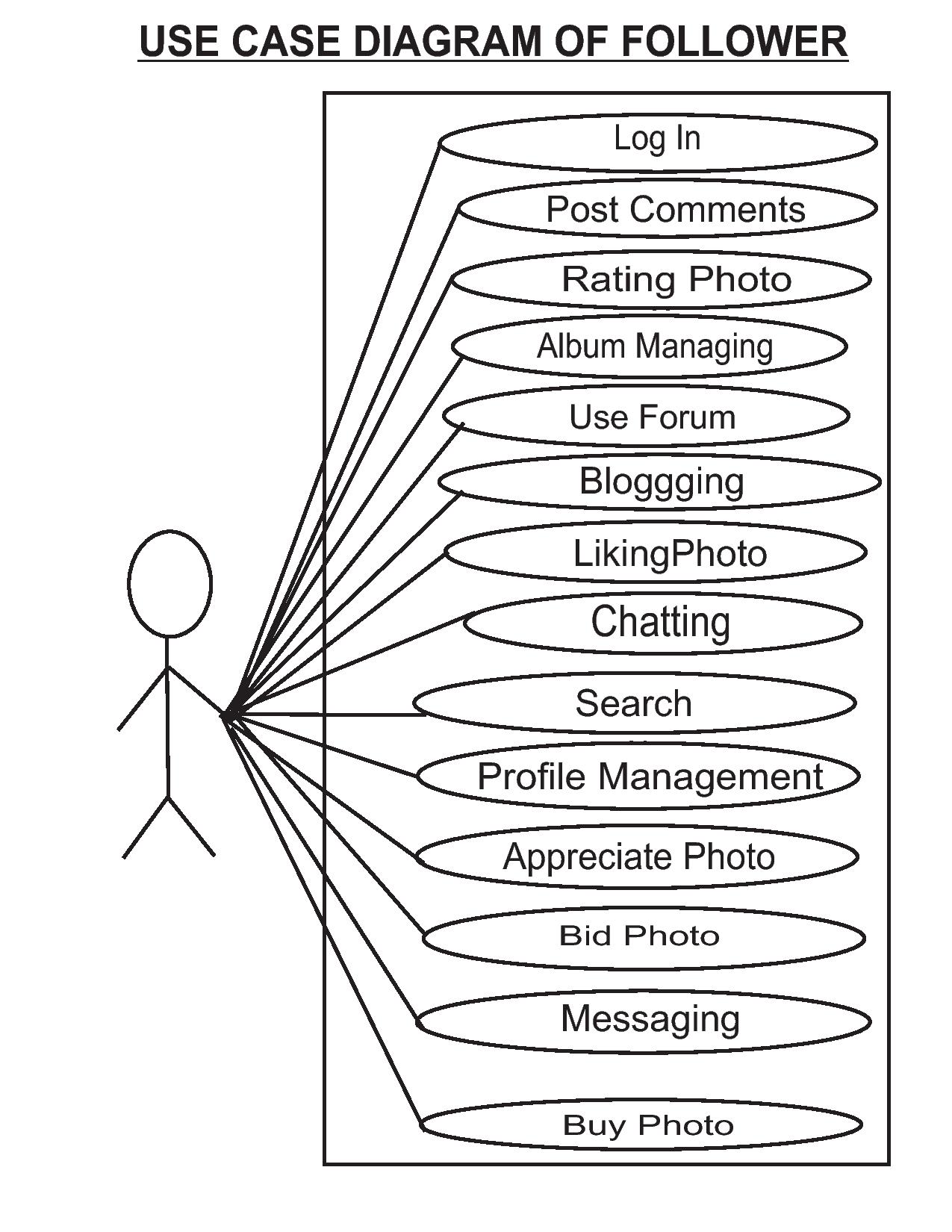
**System**

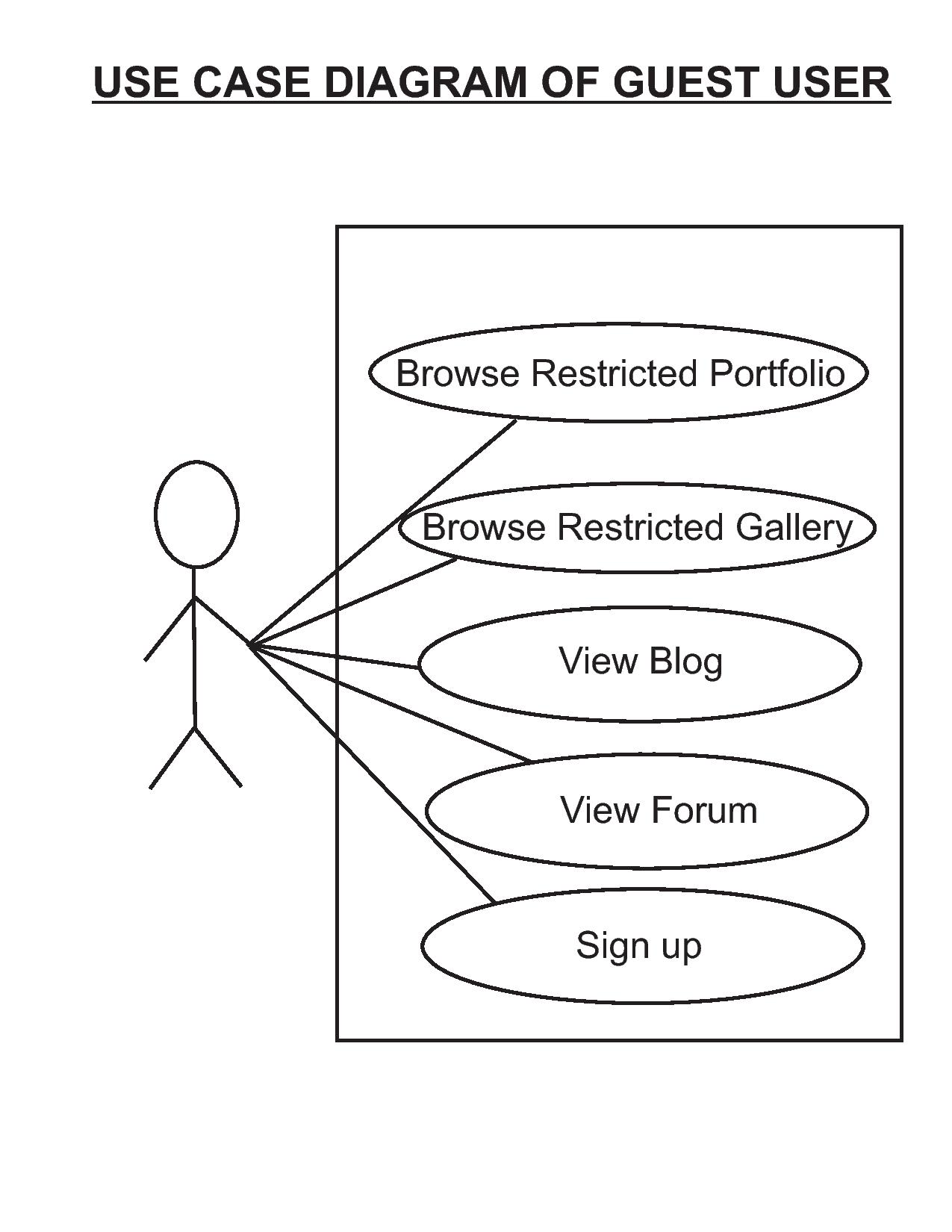
**Boundary**

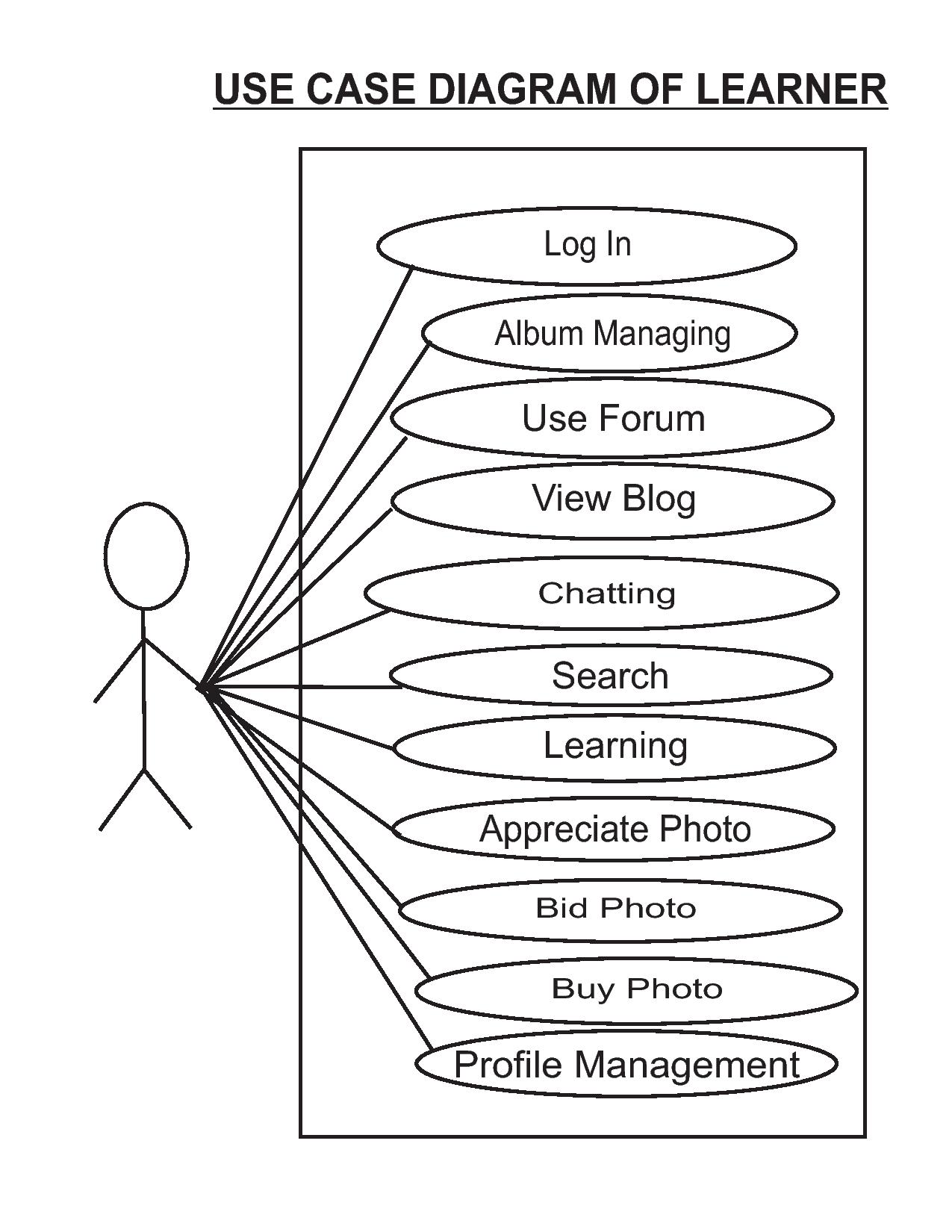
actor

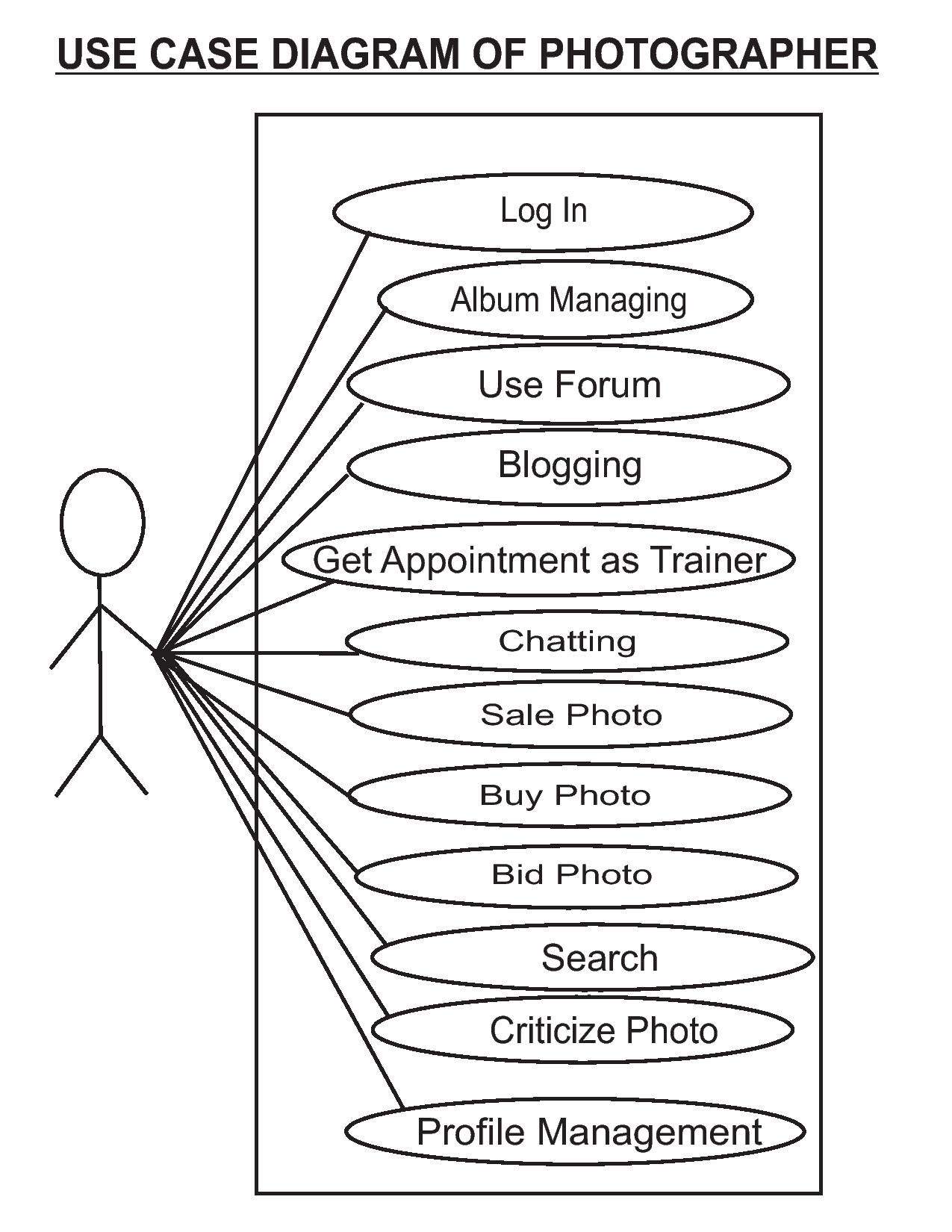
**Association**

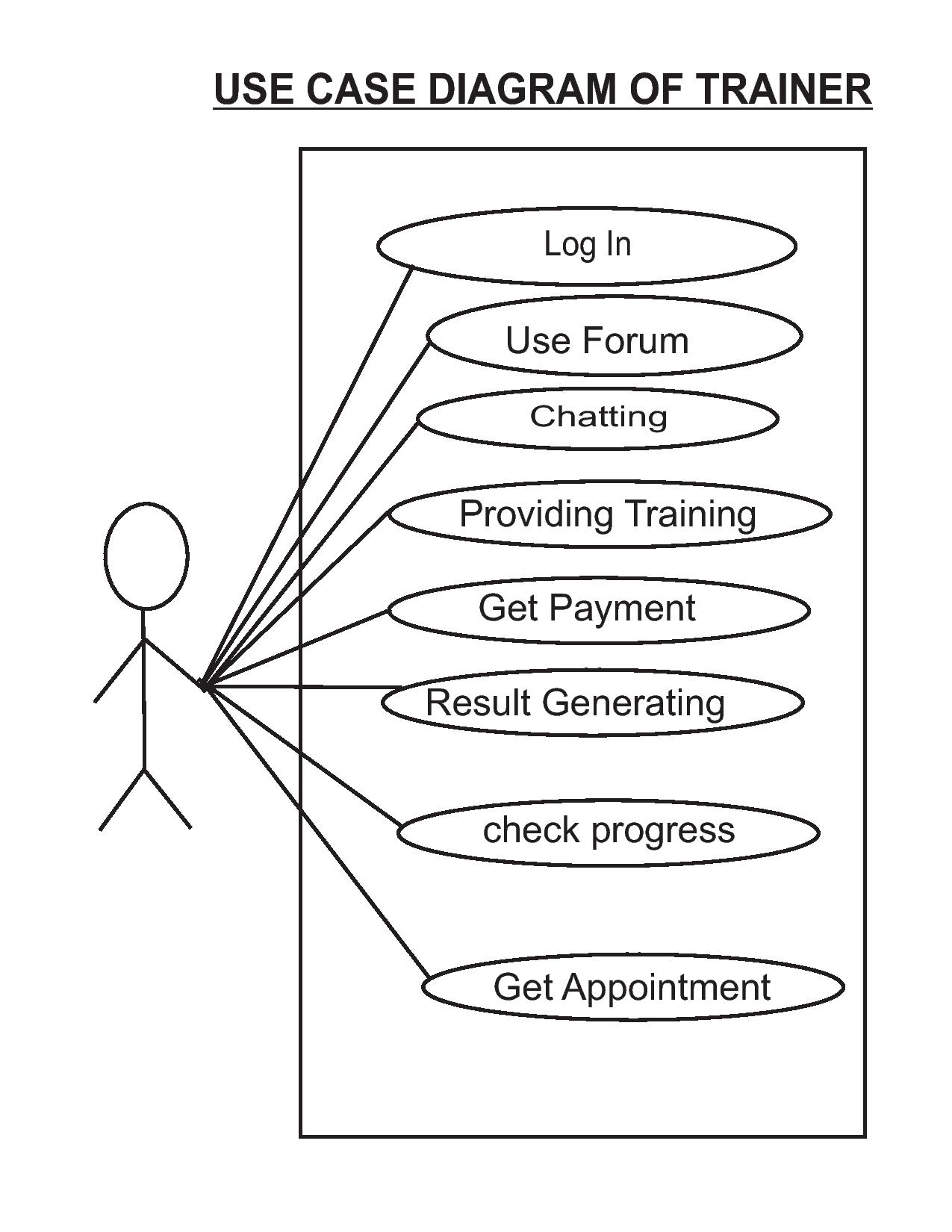












**Database Design**

**TABLE NAME: ADMIN (PRIMARY KEY: ADM\_ID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| ADM\_ID | VARCHAR2(20) | NOT NULL | Primary key |
| NM | VARCHAR2(30) | NOT NULL | name |
| LOGINID | VARCHAR2(20) | NOT NULL ,UNIQUE | Logging id |
| PASSWORD | VARCHAR2(15) | NOT NULL | password |
| EMAIL | VARCHAR2(30) | NOT NULL | Email id |

**TABLE NAME: ROYALTY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | Reference MEMBER MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | Reference ALBUM |
| NO\_VIEWER | NUMBER | NOT NULL | Viewer count |
| AMT | NUMBER | NOT NULL | Amount for Royalty |

**TABLE NAME: MEM\_INST**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) |  | Member id |
| INST\_AREA | VARCHAR2(30) | NOT NULL | Interest domain |

**TABLE NAME: BLOG (PRIMARY KEY: BLGID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| BLGID | VARCHAR2(20) | NOT NULL ,PRIMARY KEY |  |
| POST\_TEXT | VARCHAR2(2500) | NOT NULL |  |
| BDT | TIMESTAMP(6) | NOT NULL |  |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| SRCEEN\_ST | VARCHAR2(1) | NOT NULL |  |

**TBLE NAME: FORUN\_ANS (PRIMARY KEY: FANSID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| FANSID | VARCHAR2(20) | NOT NULL | PRIMARY KEY |
| FID | VARCHAR2(20) | NOT NULL ,FOREIGN KEY | REFERENCE FORUM\_QRY |
| ANSWER | VARCHAR2(500) | NOT NULL |  |
| FADT | TIMESTAMP(6) | NOT NULL |  |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE MEMBER\_MASTER |
| SRCEEN\_ST | VARCHAR2(1) | NOT NULL |  |

**TABLE NAME: MEM\_MASTER (PRIMARY KEY: MEM\_ID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | NOT NULL | Primary kay |
| NAME | VARCHAR2(30) | NOT NULL |  |
| DOB | DATE | NOT NULL |  |
| GENDER | VARCHAR2(1) | NOT NULL |  |
| ADDR | VARCHAR2(100) | NOT NULL | Address |
| COUNTRY | VARCHAR2(30) | NOT NULL |  |
| CONTACT | VARCHAR2(15) | NOT NULL |  |
| EMAIL | VARCHAR2(1) | NOT NULL |  |
| PROF\_IMG | VARCHAR2(100) | NOT NULL | Profile image |
| PROF\_PROFILE | VARCHAR2(500) | NOT NULL | Professional profile |
| JOB\_PROFILE | VARCHAR2(500) | NOT NULL | Job description |
| MEM\_LVL | VARCHAR2(10) | NOT NULL | Member level |
| MEM\_ST | VARCHAR2(1) | NOT NULL | Interest area |
| BANK | VARCHAR2(30) | NOT NULL |  |
| IFC | VARCHAR2(15) | NOT NULL |  |
| BRANCH | VARCHAR2(50) | NOT NULL |  |
| ACC\_NO | VARCHAR2(20) | NOT NULL |  |
| LOG\_ID | VARCHAR2(20) | NOT NULL |  |
| PSWD | VARCHAR2(15) | NOT NULL |  |

**TABLE NAME: FORUM\_QRY (PRIMARY KEY: FID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| FID | VARCHAR2(20) | NOT NULL,PRIMARY KEY |  |
| QUESTION | VARCHAR2(500) | NOT NULL |  |
| FDT | TIMESTAMP(6) | NOT NULL |  |
| MEM\_ID | VARCHAR2(20) | NOT NULL ,FOREIGN KEY | REF MEMBER\_MASTER |
| FTYPE | VARCHAR2(2) | NOT NULL |  |
| SRCEEN\_ST | VARCHAR2(1) | NOT NULL |  |

**TABLE NAME: BLOG\_CMNT (PRIMARY KEY: BLG\_CMNTID )**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| BLG\_CMNTID | VARCHAR2(20) | NOT NULL ,PRIMARY KEY |  |
| BLGID | VARCHAR2(20) | FOREIGN KEY | REFERENCE BLOG |
| CMNT\_TEXT | VARCHAR2(500) | NOT NULL |  |
| BDT | TIMESTAMP(6) | NOT NULL |  |
| MEM\_ID | VARCHAR2(20) | FOREIGN EY | REFERENCE MEMBER\_MASTER |
| SRCEEN\_ST | VARCHAR2(1) | NOT NULL |  |

**TABLE NAME: PORTFOLIO**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| AWARD | VARCHAR2(200) | NOT NULL |  |
| CV | VARCHAR2(500) | NOT NULL |  |
| USEDIN | VARCHAR2(100) | NOT NULL |  |

**TABLE NAME: ALBUM (PRIMARY KEY: PIC\_ID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| PIC\_ID | VARCHAR2(20) | NOT NULL,PRIMARY KEY |  |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY |  |
| PHOTO\_FILE | VARCHAR2(100) | NOT NULL |  |
| PHOTO\_DOMAIN | VARCHAR2(50) | NOT NULL |  |
| PHOTO\_DES | VARCHAR2(1000) | NOT NULL |  |
| EQUI\_DES | VARCHAR2(500) | NOT NULL |  |
| EQUI\_IMG | VARCHAR2(100) | NOT NULL |  |
| SRCEEN\_ST | VARCHAR2(1) | NOT NULL |  |
| UPLOAD\_TIME | TIMESTAMP(6) |  |  |

**TABLE NAME: CHAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID1 | VARCHAR2(20) | FOREIGN KEY | REFERENCE MAMBER\_MASTER |
| MEM\_ID2 | VARCHAR2(20) | FOREIGN KEY | REFERENCE MAMBER\_MASTER |
| CHATDT | TIMESTAMP(6) | NOT NULL |  |
| CHAT\_TXT | VARCHAR2(500) | NOT NULL |  |

**TABLE NAME: LERNER (PRIMARY KEY: LID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| LID | VARCHAR2(20) | NOT NULL ,PRIMARY KEY |  |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| TRAINER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBAER\_MASTRER |
| TRAINING\_DOMIAN | VARCHAR2(100) | NOT NULL |  |
| VALID\_FORM | VARCHAR2(30) | NOT NULL | DURATION START |
| VALID\_TO | VARCHAR2(30) | NOT NULL | DURATION END |
| CREDIT | VARCHAR2(30) | NOT NULL |  |
| CUR\_LVL | VARCHAR2(30) | NOT NULL |  |

**TABLE NAME: EXALBUM (PRIMARY KEY: LID )**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| LID | VARCHAR2(20) | FOREIGN KEY | REFERENCE LEARNER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| EXALDT | TIMESTAMP(6) |  |  |
| TRAINER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| EXPRT\_RT | VARCHAR2(10) |  |  |
| EXPRT\_CMNT | VARCHAR2(100) |  |  |

**TABLE NAME: TRAINER\_CHARG**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| EXPERTICE\_LVL | VARCHAR2(30) | NOT NULL | EXPERTICE LEVEL |
| AMT | NUMBER | NOT NULL | AMMOUNT |

**TABLE NAME: LIKING**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE MEMBER\_MASTER |
| CRITC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE MEMBER\_MASTER |
| CRITC\_CMNT | VARCHAR2(500) |  |  |
| CRITCDIS\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE MEMBER\_MASTER |
| LDDT | TIMESTAMP(6) |  | LIKING DATE |
| LKST | VARCHAR2(10) |  | LIKING STATUS |

**TABLE NAME: LEARNING\_MASTER (PRIMARY KEY: LEARNING\_ID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| LEARNING\_ID | VARCHAR2(20) | NOT NULL ,PRIMARY KEY |  |
| DOMIAN | VARCHAR2(500) | NOT NULL |  |
| DURATION | VARCHAR2(50) | NOT NULL |  |
| FEES | NUMBER |  |  |
| ELG\_CRIT | VARCHAR2(200) | NOT NULL | ELIGIBILITY CRITERIA |

**TABLE NAME: AUCTION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| MINBIDVAL | NUMBER | NOT NULL |  |
| MAXBIDVAL | NUMBER | NOT NULL |  |
| START\_TIME | DATE | NOT NULL |  |
| END\_TIME | DATE | NOT NULL |  |

**TABLE NAME: BID\_TAB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| BIDDER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| BIDDT | DATE | NOT NULL |  |
| BID\_AMT | NUMBER | NOT NULL |  |

**TABLE NAME: PHOTO\_SALE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REFMEMBER\_MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| PRICE | NUMBER | NOT NULL |  |

**TABLE NAME: CHARGES (PRIMARY KEY: CRG\_ID)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| CRG\_ID | VARCHAR2(20) | NOT NULL ,PRIMARY KEY |  |
| AM | NUMBER | NOT NULL | AMMOUNT |
| CRG\_TYP | VARCHAR2(50) | NOT NULL |  |

**TABLE NAME: LOGGEDIN\_USER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| MEM\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| LOGIN\_DATE | DATE | NOT NULL |  |
| LOGIN\_TIME | TIMESTAMP(6) | NOT NULL |  |

**TABLE NAME: LERNER\_PAYMNT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| LID | VARCHAR2(20) | FOREIGN KEY | REFERENCE LEARNER |
| LEARNING\_ID | VARCHAR2(20) | FOREIGN KEY | REF LEARNING\_MASTER |
| AMT | NUMBER | NOT NULL | AMMOUNT |
| CARD\_TYP | VARCHAR2(50) | NOT NULL |  |
| CARD\_NO | VARCHAR2(50) | NOT NULL |  |
| CARD\_EXPDT | DATE | NOT NULL |  |
| CARD\_HOLDER | VARCHAR2(50) | NOT NULL |  |

**TABLE NAME: TRAINER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| TRAINER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| DATE\_APP | DATE |  |  |
| TRAINING\_DOMAIN | VARCHAR2(30) | NOT NULL |  |
| OFFER\_DATE | DATE | NOT NULL |  |
| VALID\_UPTO | DATE |  |  |
| EXPERTICE\_LVL | VARCHAR2(30) | NOT NULL |  |
| OFFER\_PRD | VARCHAR2(10) | NOT NULL |  |
| ACCEPTENCE\_ST | VARCHAR2(1) |  |  |

**TABLE NAME: TRAINER\_LEARNER**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| TRAINER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| LID | VARCHAR2(20) | FOREIGN KEY | REF LERNER\_MASTER |
| DATE\_ALLOT | DATE | NOT NULL |  |
| TRAINING\_DOMAIN | VARCHAR2(30) | NOT NULL |  |
| VALID\_UPTO | DATE | NOT NULL |  |
| EXPERTICE\_LVL | VARCHAR2(30) | NOT NULL |  |

**TABLE NAME: TRAINING\_PYMNT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| TRAINER\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| DATE\_ALLOT | DATE | NOT NULL |  |
| TRAINING\_DOMAIN | VARCHAR2(30) | NOT NULL |  |
| VALID\_UPTO | DATE | NOT NULL |  |
| NO\_LEARNER | NUMBER | NOT NULL |  |
| TOT\_AMT | NUMBER | NOT NULL |  |
| PAY\_RELDT | DATE |  |  |

**TABLE NAME: PURCHES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **CONSTRAINT** | **DESCRIPTION** |
| SL\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| BY\_ID | VARCHAR2(20) | FOREIGN KEY | REF MEMBER\_MASTER |
| PIC\_ID | VARCHAR2(20) | FOREIGN KEY | REFERENCE ALBUM |
| AMT\_PAID | NUMBER | NOT NULL |  |
| PAY\_DT | DATE | NOT NULL |  |
| CARD\_NO | VARCHAR2(20) | NOT NULL |  |
| CARD\_EXPDT | DATE | NOT NULL |  |
| CARD\_TYP | VARCHAR2(20) | NOT NULL |  |
| CARD\_HOLDER | VARCHAR2(50) | NOT NULL |  |
| SL\_AMT | NUMBER | NOT NULL |  |
| PYMNT\_RELDT | DATE | NOTT NULL |  |

**Process Involved**

* **Registration process:** It accepts user name from the guest or new member. It checks whether the user name is already in use. If so, the process prompts user to give anther valid user name. Registration process is sub divided into three process-
  1. **Photographer registration** where photographer has to submit his portfolio, job profile, photos for gallery and album .This process deducts Rs 5000 as registration charge for Photographer.
  2. **Follower registration** is the second process where other members have to submit their information and a registration fees of Rs2000.
  3. **Payment process** is actually deals with payment related validation and generates a transaction id and status for successful payment transaction.

It automatically generates member id and deposit id and encrypts the password. It stores member id along with all related information to the relevant table. On successful completion of registration process it redirects new member to his homepage.

* **Log in process:** It deals with two sub process:
* **Authentication:** It accepts user id and password from the user. The password is encrypted and matched with the encrypted password stored table. This process also checks user status (whether he has already unsubscribed or not). It redirects valid administrative users and members to their respective home page.
* **Login Recovery:** If a member forgets his password this process helps him by generating a new password. Here the member has to give his email id and member id .This process then finds the member id and status and verifies. If match is found, it automatically generates a new password and send it to the member’s corresponding email address and prompts him to change his password immediately for safety.
* **Photographer zone:** It deals with the management of all the activities of the photographer available in this system.
  + **Photo Upload:** Photographer can upload photo for gallery, album, auction or sale.
  + **Gallery Management:** From various photos, photographer can select 10 photos for gallery.
  + **Photo Sale:** Photographer chooses his best photos for sailing through this zone.
  + **Portfolio Management:** With his exclusive photos and work experience and job profile photographer here make his stunning portfolio.
  + **Photo Auction:** Photographer chooses his best photos for auction through this zone..
* **Follower Zone:** It deals with the management of all the activities of the non- photographer available in this system.
* **Upload Photo:** Follower can upload photo for gallery, album.
* **Profile Management:** allows Follower edit and update their personal information
* **Photo Purchase:** Follower can get exclusive photos of various experts through direct sale or auction
* **Photo Bidding:** Follower can bid for a photo in auction pole.
* **Download Copyrighted Photo:** Follower can view and download watermarked photo for free.
* **Payment Process:** Follower can pay for their selected photo by secure payment process.
* **Photo Receiving Process:** After successful payment follower can get their purchased photo URL through email.
* **View Gallery:** In the photo gallery, Member can perform some following task:
* **Search Photo:** They can search their desired photo by
* **Search by category:** ProvidingSpecific Category.
* **Search by subject:** Providing Specific Subject.
* **Search by incident:** Providing Specific Incident.
* **Advance Text Search:** Providing Specific Keyword.
* **Comment Sharing:** Member can post their opinion for photos.
* **Photo Liking:** Member can like photos.
* **Photo Rating:** Member can give rating for some photos.
* **Learning Zone:** It provides 3 ways to purchase a product, namely direct purchase, purchase through wish list and purchase through cart.
* **Trainer appointment:** order id, customer id, product id, price, order date, order status, order delivery date and product Status.
* **Learner registration:** A customer can add and keep product items in his wish list for future reference. He can wish list.
* **Tips and Techniques:** Here learner can get expert’s advice tip and photographic techniques.
* **Check Progress:** Trainer can evaluate learner’s progress and assign respective expertise level
* **Student Forum:** With the help of student forum learner can get expert’s answers to their various questions
* **Group Discussion:** Learner can engage in group discussion with other learners as well as experts.
* **Concept Sharing:** It deals with the management of all the products available in this system.
* **Forum Management:** Through this process member can access forum facility.
* **Blog management:** Through this process member can access blog facility.
* **Chatting:** Through this process member can access chatting facility.
* **Messaging:** Through this process member can access message posting facility.

**Methodology Adopted &System Implementation**

The linear sequential model is followed to develop the software; this model encompasses the following activities:

1. System/information modeling.

ii. System requirement analysis.

iii. System design.

iv. Code generation.

v. Testing & support.

All the above activities are described bellow:

First of all the problem should be clearly defined. Then it is required to study the system thoroughly in order to make a clear and transparent conception of what there is in the already existing system and what is to be done to make it computerized. In this purpose, the techniques adopted are the requirement elicitation and requirement analysis.

The requirements are gathered by studying different documents and by interviewing the related and authorized persons formally and informally. After analyzing the requirements a Software Requirement Specification (SRS) is written and it is produced to the appropriate authority of the system to get it approved.

After approval of the SRS the actual development work begins. The other facts and fictions of the system are elicited in order to tune the requirement criteria.

Then the analysis models are developed to represent the system pictorially/graphically. From these analysis models the system design is generated to represent the components, functions and behavior of the system. In this connection, DFD (Data Flow Diagram), ERD (Entity Relationship Diagram), Structure Chart etc are generated. Also the database is designed appropriately to make it compatible to the system. The front-end screens are first designed.

Then coding is started. After the coding is completed, different testing methods are adopted to trap the errors; White-box testing is to be performed to focus on the program-control-structure. This is done to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been exercised. Different types of runtime-errors are expected to be get trapped by this method of testing. Besides this black-box testing method is to be used for testing the developed project to validate the functional requirements without regard to the internal workings of a program. For this boundary-value-analysis technique is used, i.e. the testing of developed software at different boundary-level data is carried out. This ensures program’s ability to handle data at limits of acceptability, α-testing is conducted at the developer’s site by a customer,

Β-testing is conducted at one or more customer sites by the end-user of the software. The system is continuously debugged and tuned while conducting the testing process.

After conducting all the aforesaid procedures a user-manual is composed as a guide to the end-users. It includes security aspects, access rights, back up, controls etc. To implement the system the end-users are properly trained to handle the software.

For system-maintenance the end-users are advised to report the developer about any further erroneous result generated so far. To recover the system from severe crash of hardware or software the end-users are advised to take a daily back up at the end of everyday in a suitable secondary storage device, except the current hard disk.

**List of reports:**

## Administrator can get the following information as the output of the system:

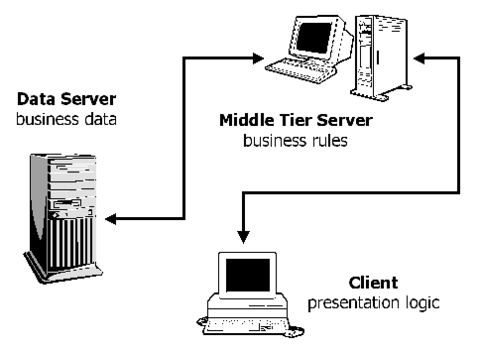
* Get information about Learners
* Get information about Photographer
* Get information about Follower
* Get information about Trainer
* Get information about Photo sale
* Get information about Photo Auction
* Get information about photo rating
* Report for royalty for photographers
* Report for photographer’s registration
* Report for Follower’s registration
* Report for Learner’s registration
* Report for Trainer’s registration
* Report for Trainer’s payment.

**Network architecture**

One of the primary reasons for the immense popularity of computers is the speed which they can accomplish specified tasks. However computer application is not always easy to use.

**Three-tier architecture**

In a three-tier architecture (also known as a multi-tier architecture), there are three or more interacting tiers, each with its own specific responsibilities (see Figure 4.3.1):



Three-Tier Architecture

* Tier 1: the client contains the presentation logic, including simple control and user input validation. This application is also known as a thin client.
* Tier 2: the middle tier is also known as the application server, which provides the business processes logic and the data access.
* Tier 3: the data server provides the business data.

Applied to web applications and distributed programming, the three logical tiers usually correspond to the physical separation between three types of devices or hosts:

* Browser or GUI Application
* Web Server or Application Server
* Database Server (often an RDBMS or Relational Database)

These are some of the advantages of three-tier architecture:

* 1. It is easier to modify or replace any tier without affecting the other tiers.
  2. Separating the application and database functionality means better load balancing.
  3. Adequate security policies can be enforced within the server tiers without hindering the clients.

**Security:**

Various security measure is implemented in different sphere of this system for protect all private data and information of the member of this system. These are as follows:-

**1) User authentication:** User verification is one of the strongest security measure that is implemented both in software as well as database label. System will check user’s information before secure navigation.

**2) Password recovery through email:** Member’s password can only be retrieved through system generated email to their corresponded email address.

**3) Encrypted password:** After choosing a password system will encrypted that password before storing it to the database.

**4) Watermarked photo:** After uploading exclusive photo of their actual resolution, system will automatically put watermarked to that photo for safety.

**5) SQL injection protection:** SQL injection is a technique where malicious users can inject SQL commands into SQL statements, via web page input. Injected SQL commands can alter SQL statement and compromises the security of a web application. The system is took protective measure for that specific activities.

**Further Enhancement of the Project:**

* Advertisement for different types of photography related equipments.
* Camera and accessory sailing and retailing.
* Vendor intervention as active member
* SMS verification at the time of login as Administrative user.
* SSL implementation.

**BIBLIOGRAPHY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BOOK | TOPIC | NAME | AUTHOR | PUBLISHER |
| 1 | Java | The Complete Reference | Herbert Schildt | Tata McGraw Hill |
| 2 | EJB | Ejb 3 In Action | [Debu Panda Reza R](http://www.flipkart.com/author/debu-panda-reza-rahman-derek-lane) | Dreamtech Press |
| 3 | JSP And Servlet | Java Server ProgrammingJava EE6 (J2EE 1.6), Black Book | [Kogent Learning Solutions Inc.](http://www.amazon.in/s/279-2739721-1030657?_encoding=UTF8&field-author=Kogent%20Learning%20Solutions%20Inc.&search-alias=stripbooks) | DreamTech |
| 5 | Tomcat | Professional Apache Tomcat 6 | Vivek Chopra, Amit Bakore | WROX |
| 6 | JavaScript | Professional JavaScript for Web Developers | Nicholas C. Zakas | WROX |
| 7 | AJAX | AJAX  The Complete Reference | Thomas Powell and Fritz Schneider | Tata McGraw Hills |
| 8 | HTML | HTML 5& XHTML: The Complete Reference | Thomas A.Powell | Tata McGraw Hills |
| 9 | CSS | Professional CSS Cascading Style Sheets for Web Design | Christopher Schmitt, Mark Trammell, | WROX |
| 10 | Database Programming | Oracle 10g  The Complete Reference | Kevin Loney, George Koch | Tata McGraw Hills |
| 12 | Database Programming | Database Systems Using Oracle | Nilesh Shah | Prentice Hall |
| 13 | SAD | Systems Analysis And Design | Igor Hawryszkiewycz | Prentice Hall |
| 14 | Software engineering | Software engineering | Pressman | Prentice Hall |