## 1.ER Diagram

Entity Relationship (ER) diagram is a graphical representation of entities and their relationships to each other. It is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. It displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. It is created based on three basic concepts: entities, attributes and relationships. It contains different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

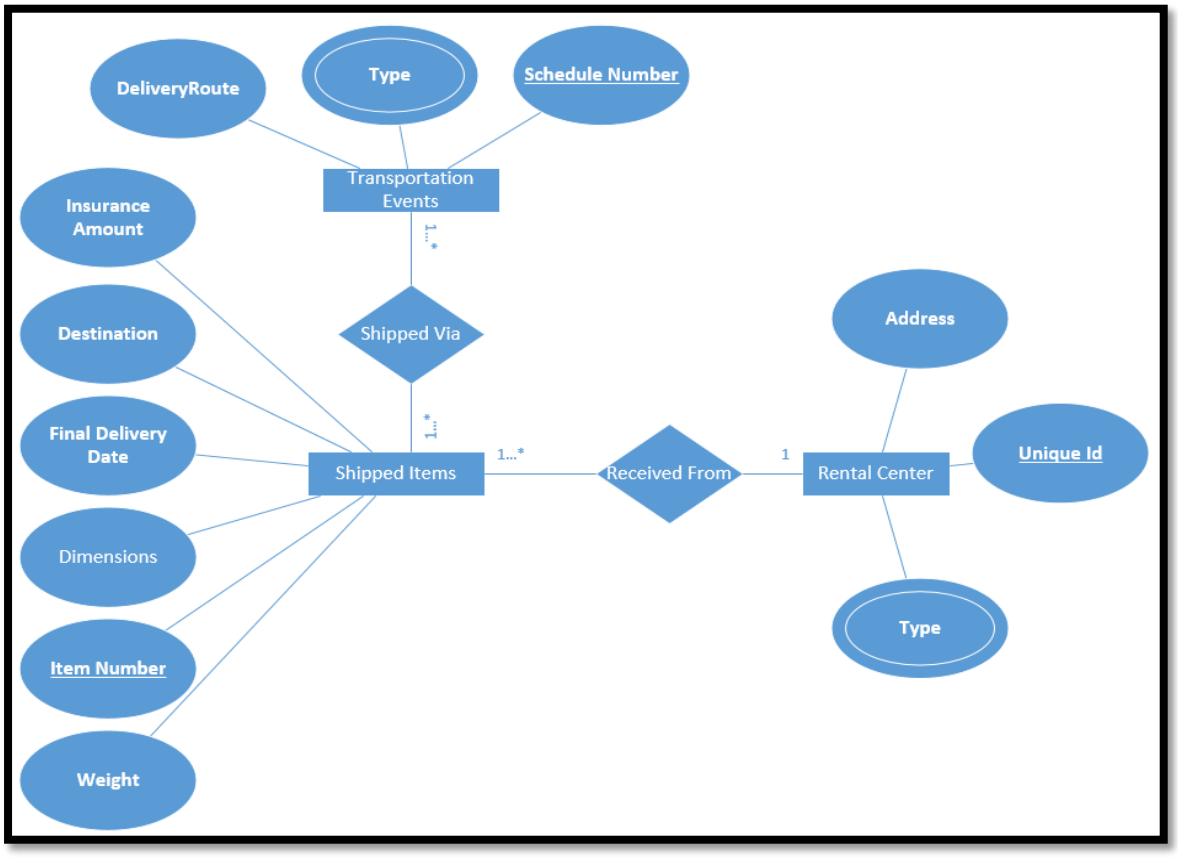
**Practical Question:**

UPS (Unitied Parcel Services) prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute.

Please create an Entity Relationship diagram that captures this information about the UPS

system. Be certain to indicate identifiers and cardinality constraints.

**Solution:**



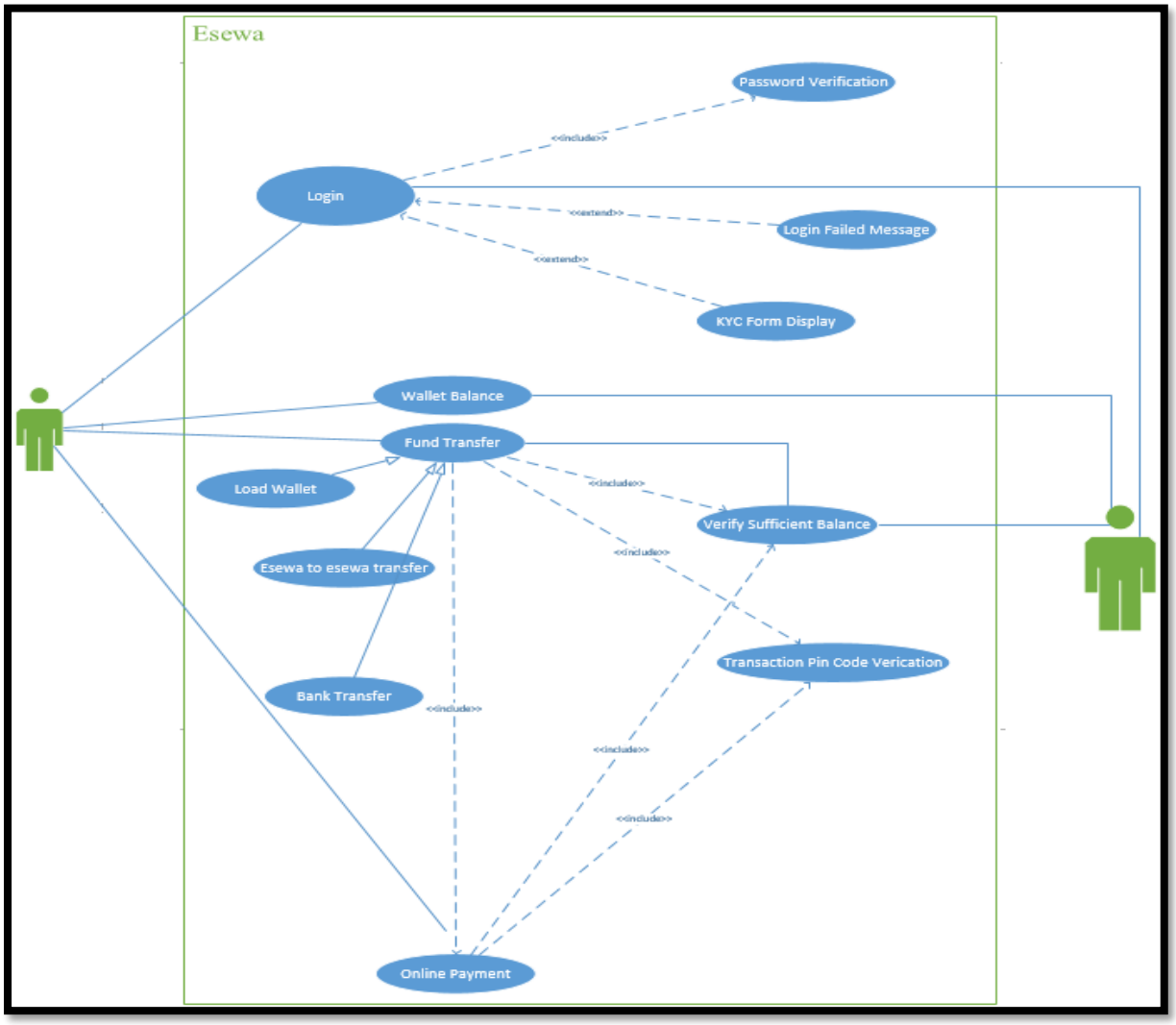
## 2.Use Case Diagram:

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems

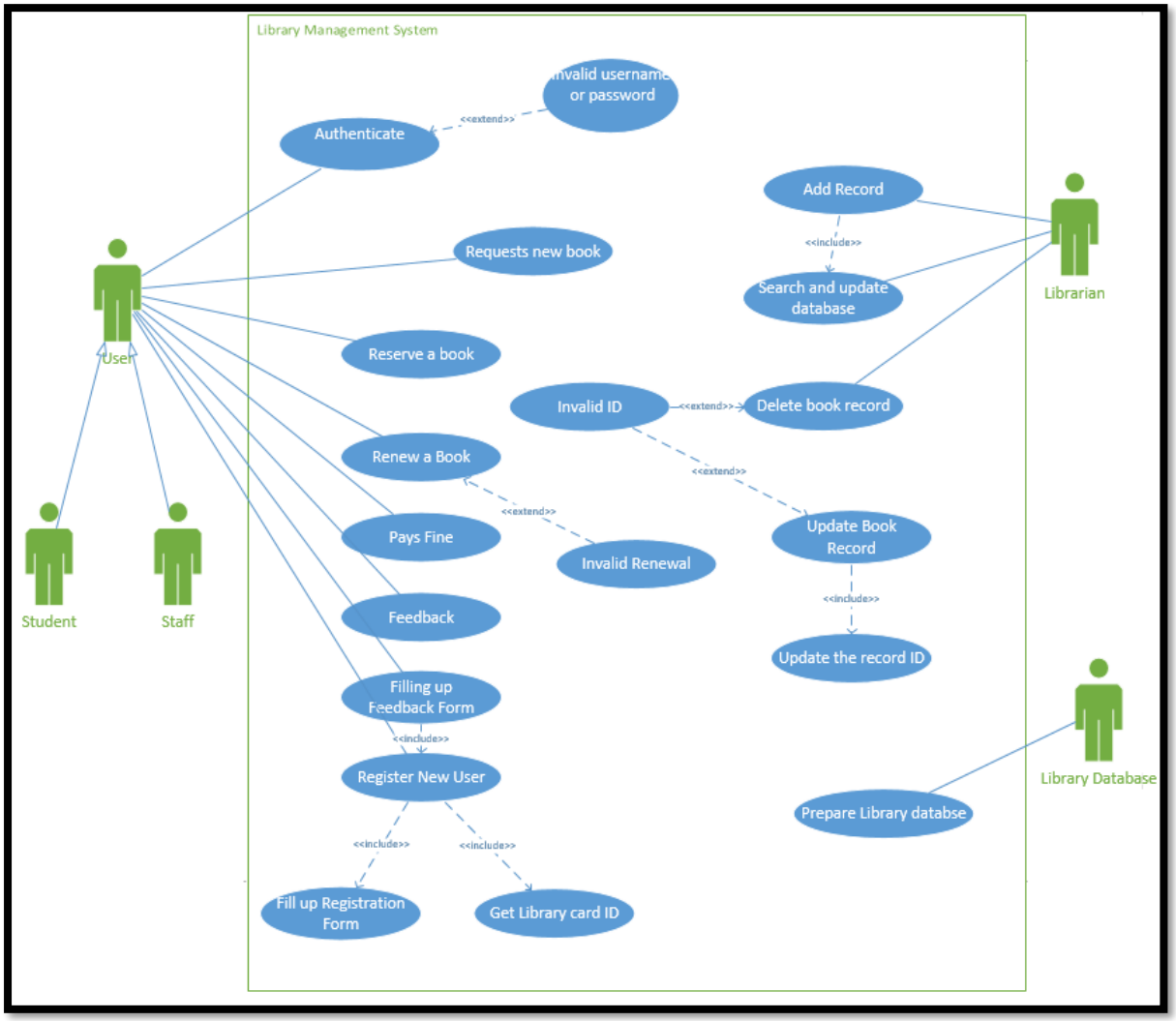
Use case diagram uses

- Represent the goals of systems and users.
- Specify the context a system should be viewed in.
- Specify system requirements.
- Provide a model for the flow of events when it comes to user interactions.
- Provide an outside view of a system.
- Show's external and internal influences on a system.

### a.Use Case diagram of Esewa:



**b.Use Case diagram of Library Management System:**



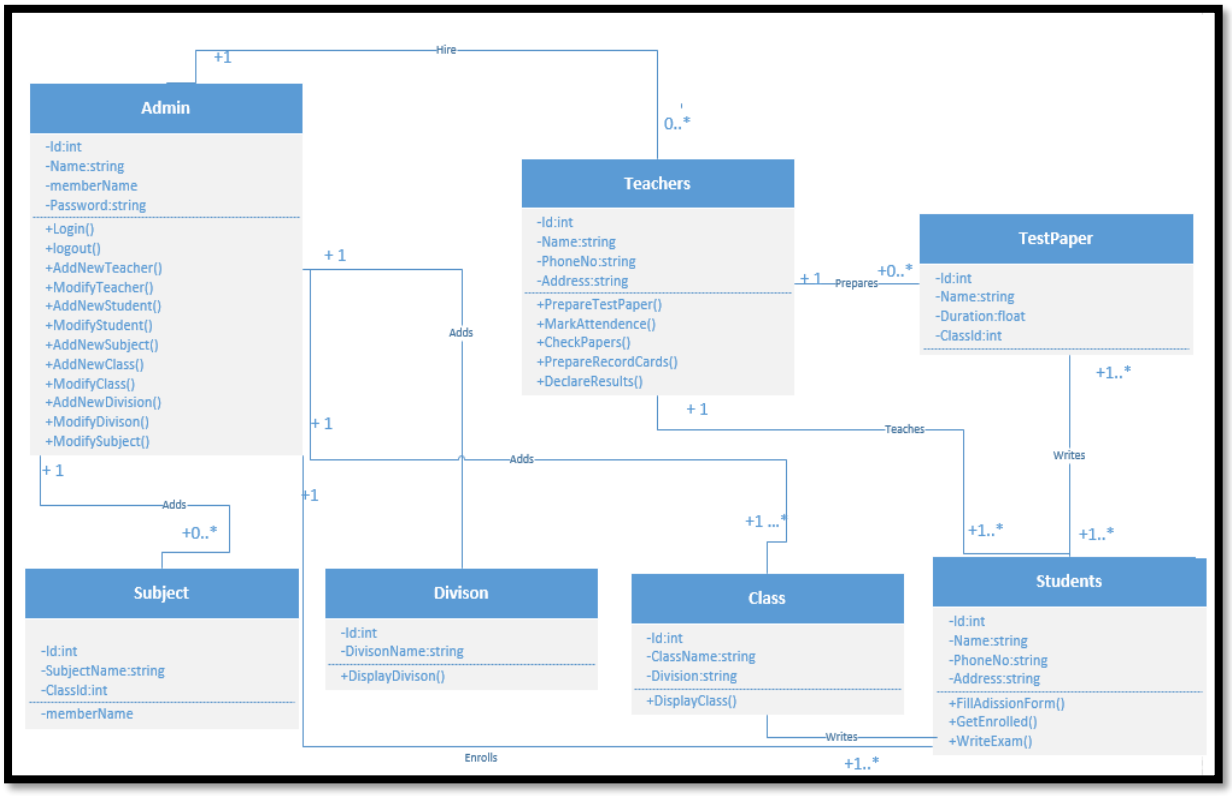

**3.Class Diagarm:**

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. It is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

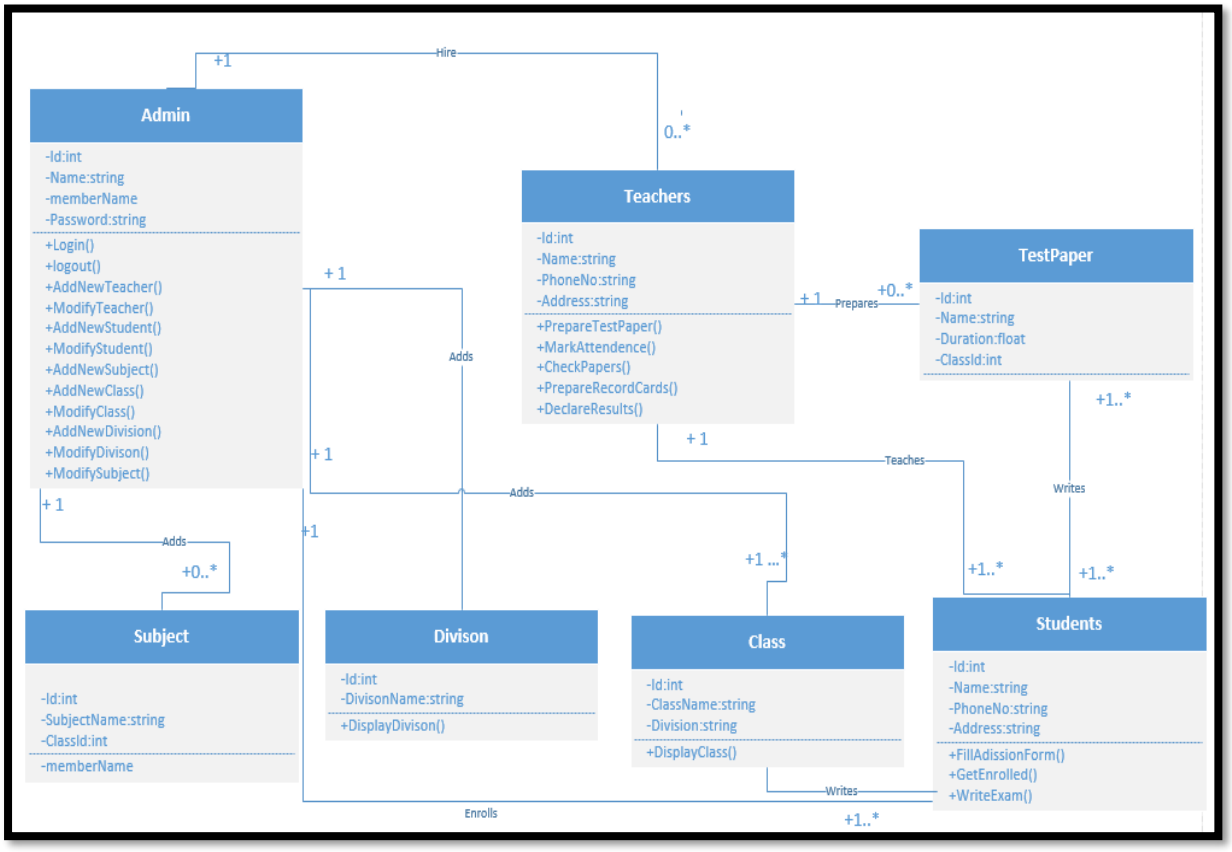
In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

## a.Class Diagram of School Management System:




## b.Class Diagram of Delivery System:



**4.Data Flow Diagarm:**
A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. The data-flow diagram is part of the structured-analysis modeling tools.

**Types of DFD:**
DFD is of two types:

1. **Logical DFD:**Logical data flow diagram mainly focuses on the system process. It illustrates how data flows in the system. Logical DFD is used in various organizations for the smooth running of system. Like in a Banking software system, it is used to describe how data is moved from one entity to another.
2. **Physical DFD:**Physical data flow diagram shows how the data flow is actually implemented in the system. Physical DFD is more specific and close to implementation.

**Components of Data Flow Diagram:**

Following are the components of the data flow diagram that are used to represent source, destination, storage and flow of data.

- **Entities:**
  Entities include source and destination of the data. Entities are represented by rectangle with their corresponding names.
- **Process:**
  The tasks performed on the data is known as process. Process is represented by circle. Somewhere round edge rectangles are also used to represent process.
- **Data Storage:**
  Data storage includes the database of the system. It is represented by rectangle with both smaller sides missing or in other words within two parallel lines.
- **Data Flow:**
  The movement of data in the system is known as data flow. It is represented with the help of arrow. The tail of the arrow is source and the head of the arrow is destination.

DFD can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond.

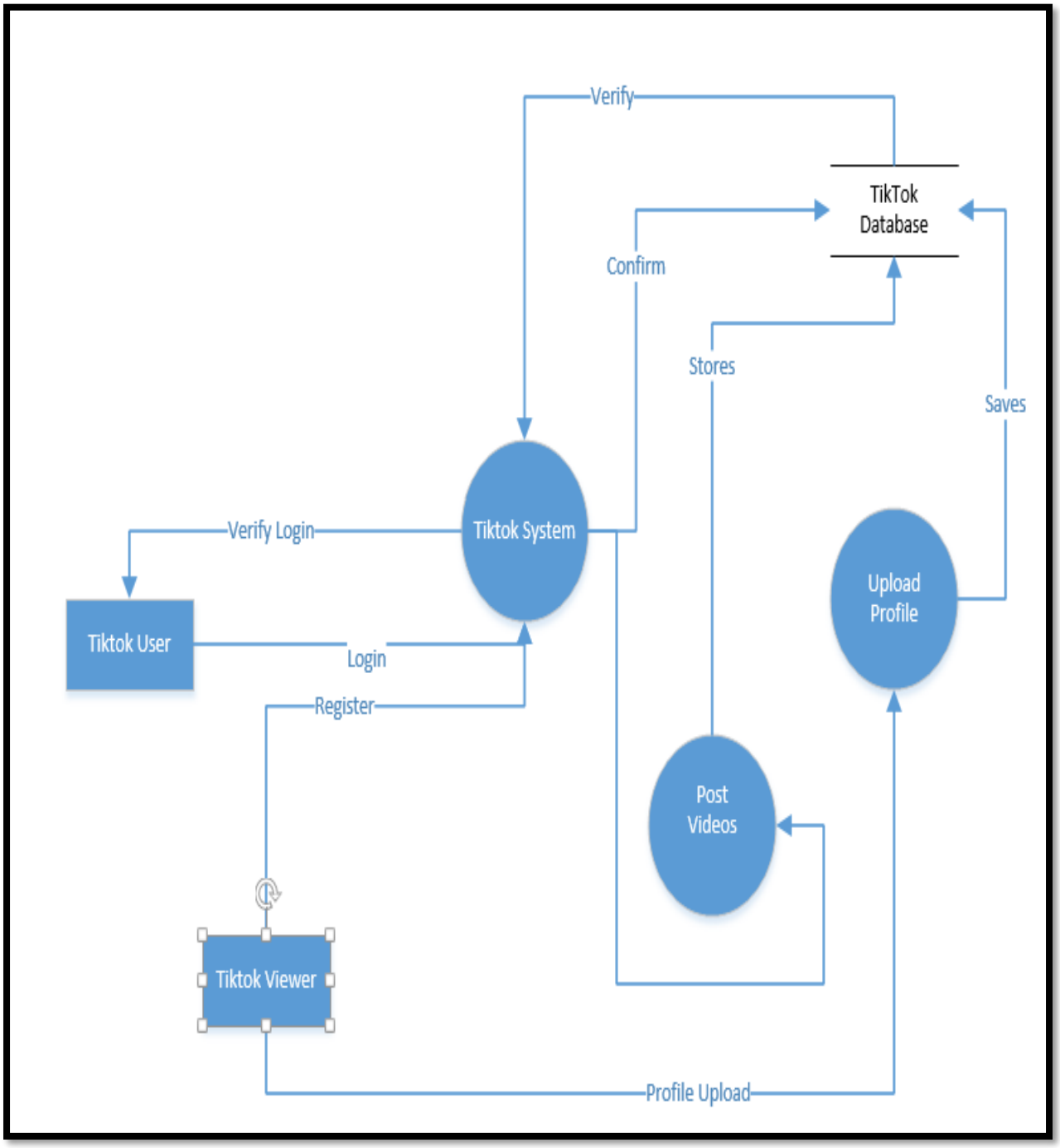
**0-level DFD:**
It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

**1-level DFD:**
In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

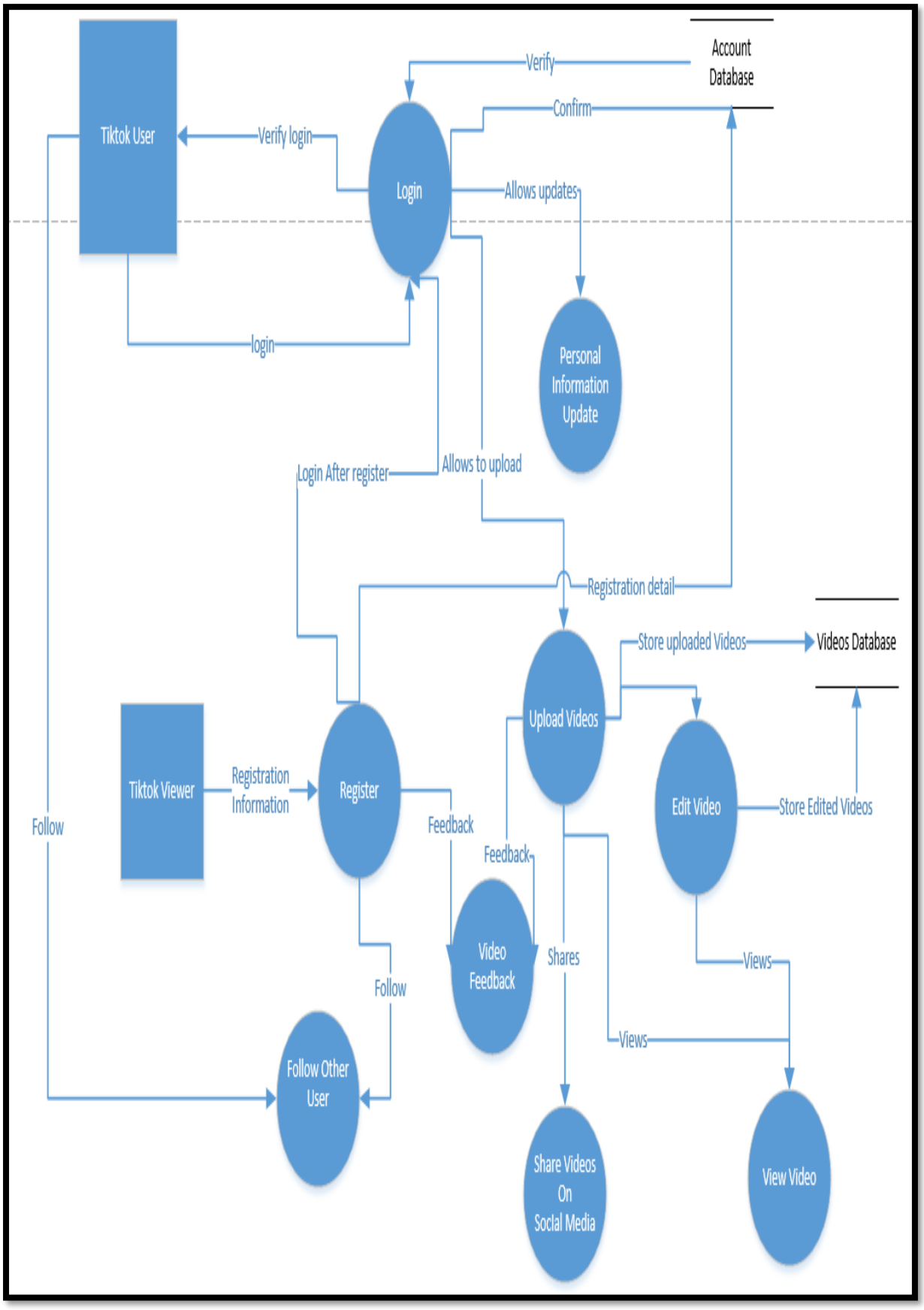
**2-level DFD:**
2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

## a.Tiktok:

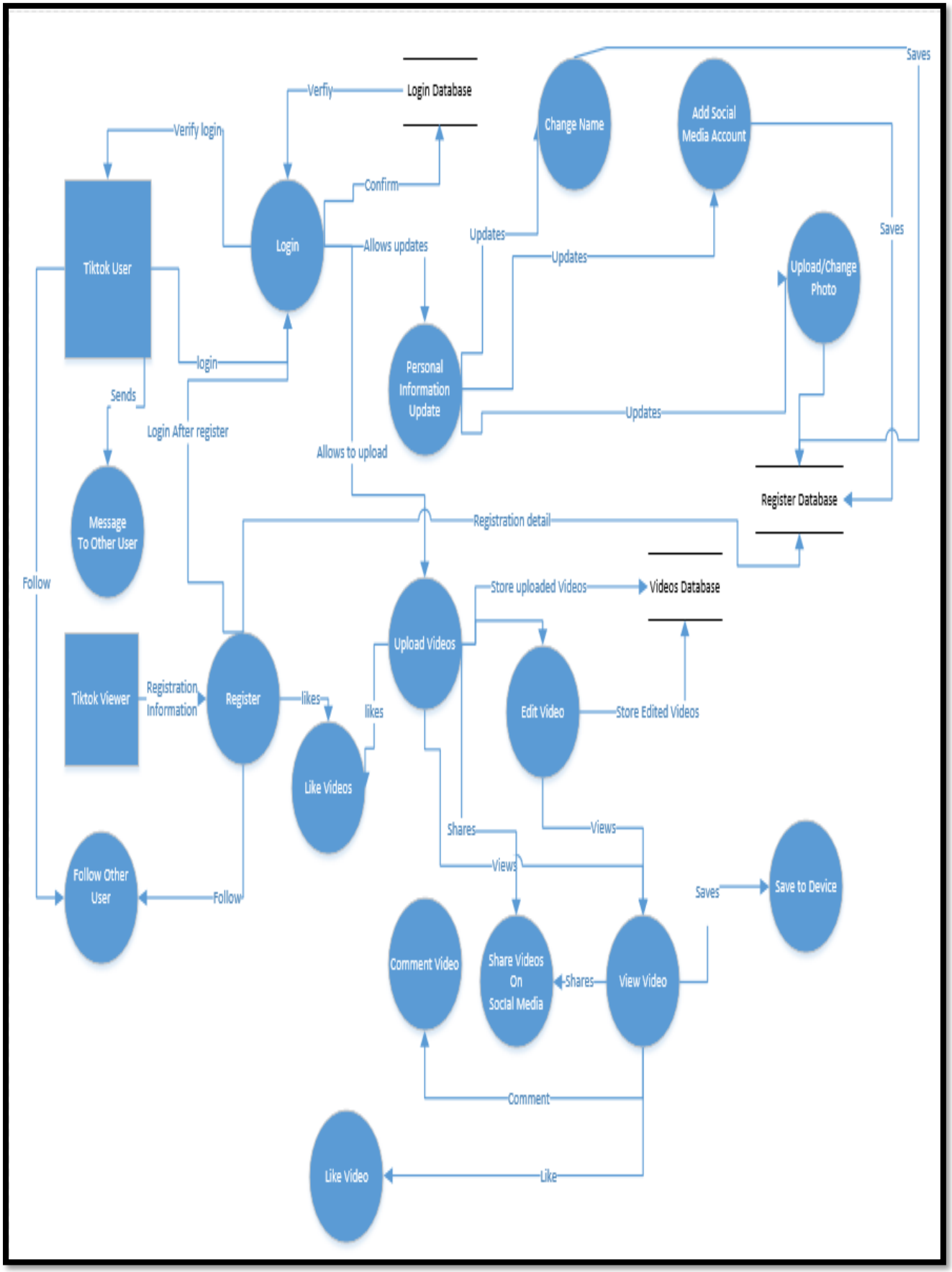
**Level 0 diagram of Tiktok:**



**Level 1 DFD of Tiktok:**

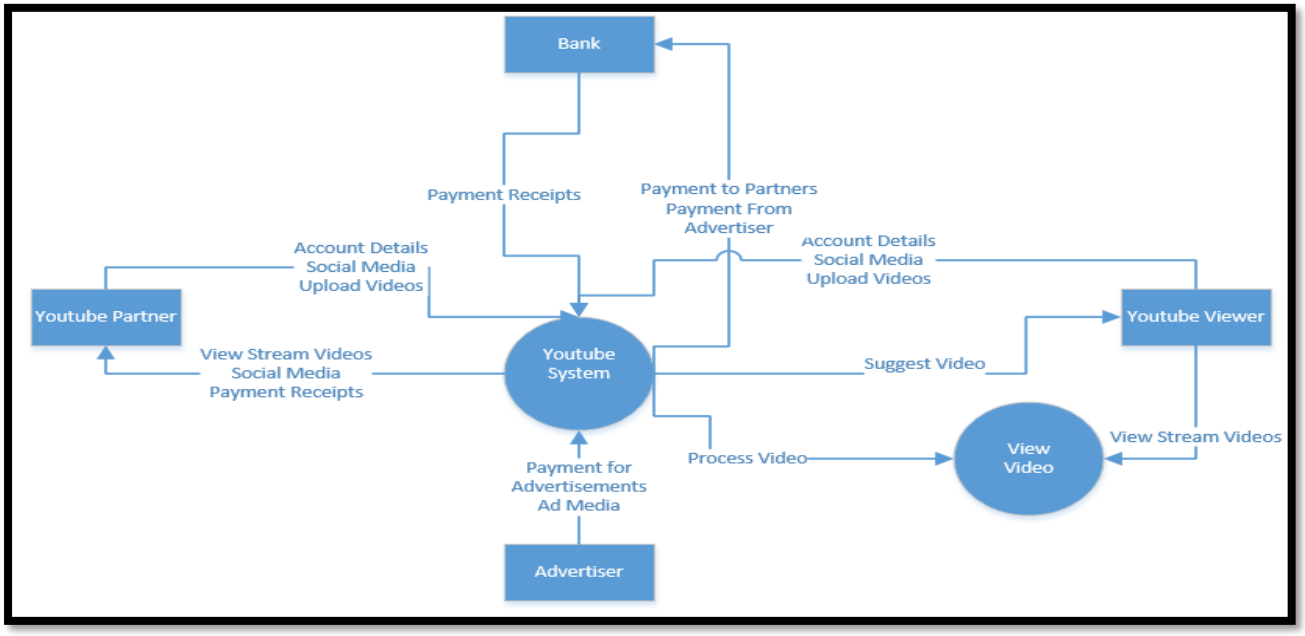


**Level 2 DFD of Tiktok:**

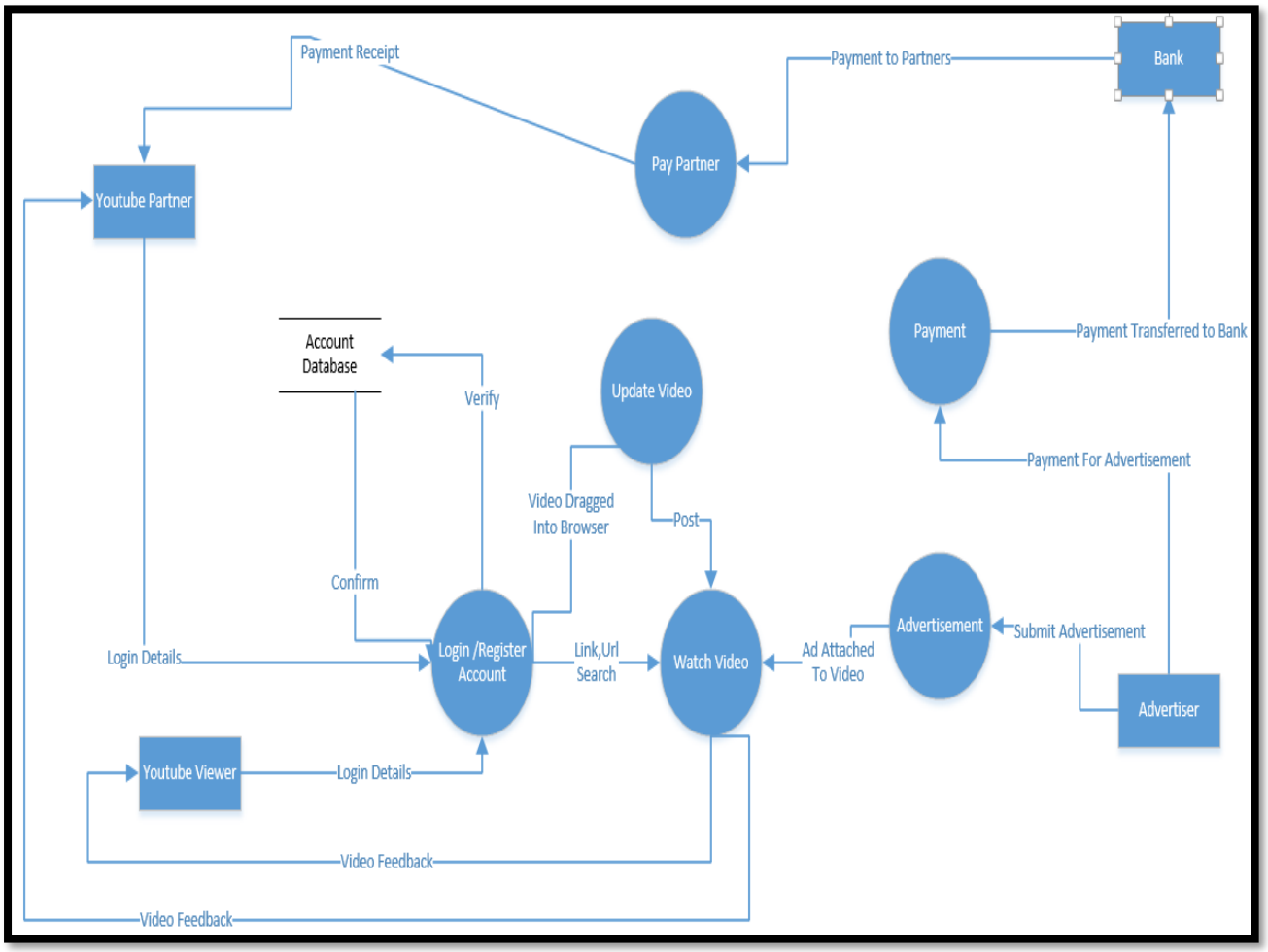


## b.Youtube:

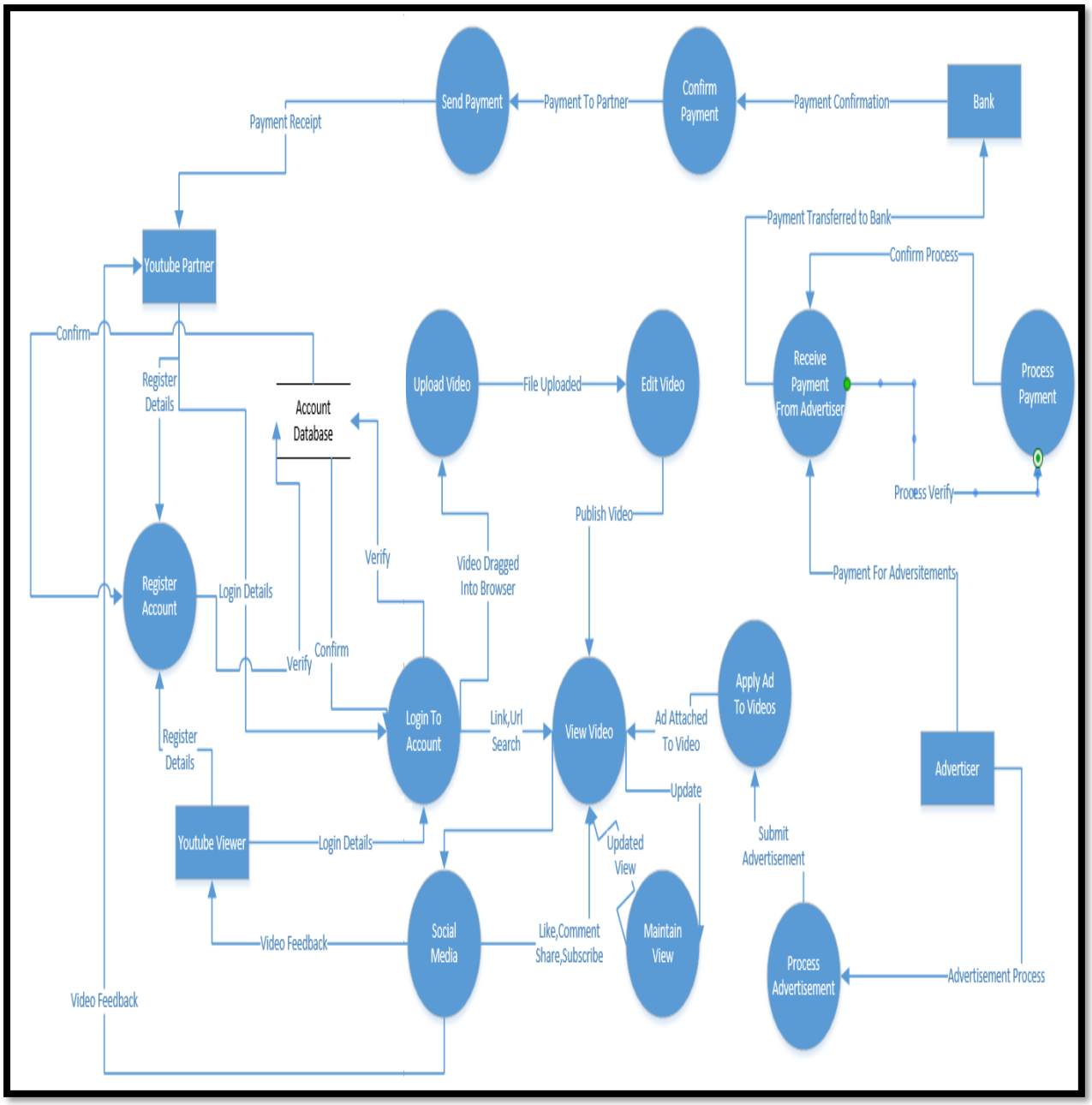
### Level 0 DFD of Youtube:




### Level 1 diagarm of Youtube:



**Level 2 diagram of Youtube:**




## 5.Break Even Point Analysis

Break-even point analysis is a measurement system that calculates the margin of safety by comparing the amount of revenues or units that must be sold to cover fixed and variable costs associated with making the sales. The purpose of the break-even analysis formula is to calculate the amount of sales that equates revenues to expenses and the amount of excess revenues, also known as profits, after the fixed and variable costs are met.

**a.Question:**

Assuming monetary benefits of an information system at $85,000 per year, one-time costs of $75,000, recurring costs of $35,000 per year, a discount rate of 12 percent, and a five-year time horizon, calculate the net present value of these costs and benefits of an information system. Also calculate the overall return on investment of the project and then present a break-even analysis. At what point does breakeven occur?

**Solution:**

Given:

Benefits per year(Y) = $85,000

One time cost = $75,000

Recurring costs per year= $35,000

Discount rate (i)= 12 %

For present value, $PV_n = Y \times \left[\frac{1}{(1+i)^n}\right]$

For benefit;

In 0 year, $PV_0 = \$0 \times \left[\frac{1}{(1+.12)^0}\right] = \$0$

In 1st year, $PV_1 = \$85{,}000 \times \left[\frac{1}{(1+.12)^1}\right] = \$75892.86$

In 2nd year, $PV_2 = \$85{,}000 \times \left[\frac{1}{(1+.12)^2}\right] = \$67761.48$

In 3rd year, $PV_3 = \$85{,}000 \times \left[\frac{1}{(1+.12)^3}\right] = \$60501.32$

In 4th year, $PV_4 = \$85{,}000 \times \left[\frac{1}{(1+.12)^4}\right] = \$54019.04$

In 5th year, $PV_5 = \$85{,}000 \times \left[\frac{1}{(1+.12)^5}\right] = \$48231.28$

NPV of Benefit= $PV_0+PV_1+PV_2+ PV_3+PV_4+ PV_5 = \$306405.98$

For Recurring Cost;

In 0 year, $PV_0 = \$0 \times \left[\frac{1}{(1+.12)^0}\right] = \$0$

In 1st year, $PV_1 = \$35{,}000 \times \left[\frac{1}{(1+.12)^1}\right] = \$31250$

In 2nd year, $PV_2 = \$35{,}000 \times \left[\frac{1}{(1+.12)^2}\right] = \$27901.79$

In 3rd year, $PV_3 = \$35{,}000 \times \left[\frac{1}{(1+.12)^3}\right] = \$24912.31$

In 4th year, $PV_4 = \$35{,}000 \times \left[\frac{1}{(1+.12)^4}\right] = \$22243.13$

In 5th year, $PV_5 = \$35{,}000 \times \left[\frac{1}{(1+.12)^5}\right] = \$19859.94$

Over all cost= NPV of Recurring Cost = Onetime cost+$PV_0$+$PV_1$+$PV_2$+ $PV_3$+$PV_4$+ $PV_5$

=$201167.17

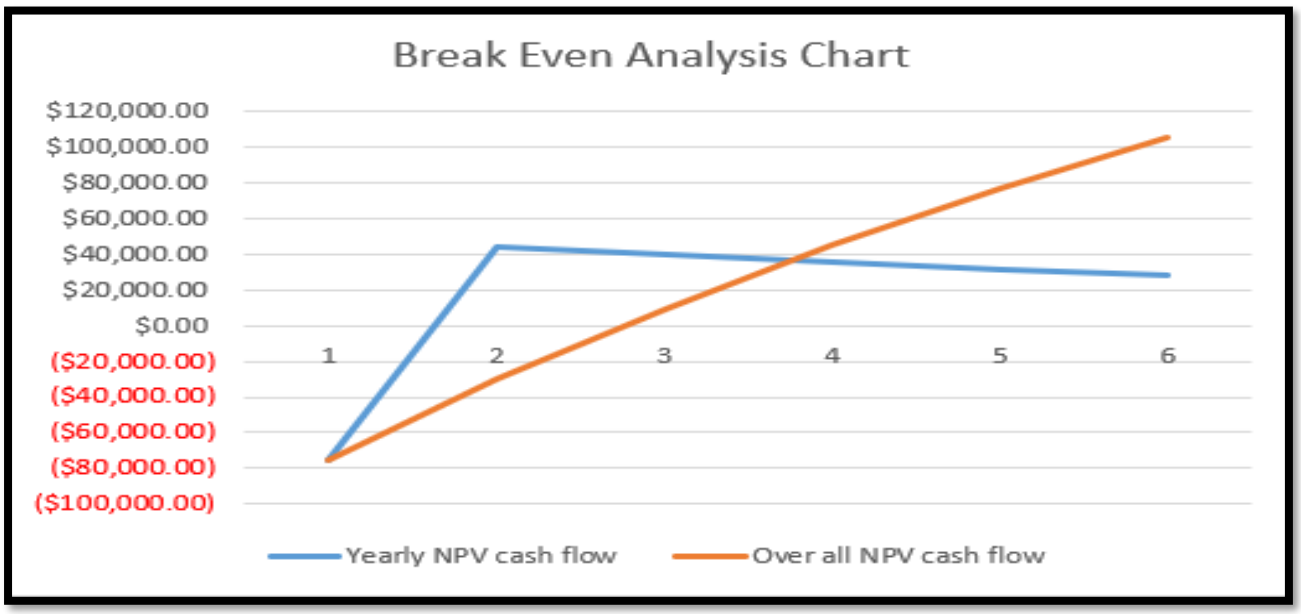
Over all NPV=NPV of Benefit – NPV of Recurring Cost=$306405.98 –$201167.17

=$105238.81

Over all ROI = Over All NPV / Over All Costs =$105238.81/$201167.17=0.52

**Break Even Analysis:**

| | Year of project | | | | | | |
|---|---|---|---|---|---|---|---|
| Years | Year 0 | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Total |
| Net economic benefit | $0.00 | $85,000.00 | $85,000.00 | $85,000.00 | $85,000.00 | $85,000.00 | |
| Discoutn rate (12%) | 1.000 | 0.893 | 0.797 | 0.712 | 0.636 | 0.567 | |
| PV of Benefit | $0.00 | $75,892.86 | $67,761.48 | $60,501.32 | $54,019.04 | $48,231.28 | **$306,405.98** |
| | | | | | | | |
| NPV of Benefit | $0.00 | $75,892.86 | $143,654.34 | $204,155.66 | $258,174.69 | $306,405.98 | **$306,405.98** |
| | | | | | | | |
| One Time Cost | ($75,000) | | | | | | |
| | | | | | | | |
| Recurring Cost | $0 | ($35,000) | ($35,000) | ($35,000) | ($35,000) | ($35,000) | |
| Discount Rate (12%) | 1.000 | 0.893 | 0.797 | 0.712 | 0.636 | 0.567 | |
| PV of Recurring Cost | $0.00 | ($31,250.00) | ($27,901.79) | ($24,912.31) | ($22,243.13) | ($19,859.94) | ($126,167.17) |
| | | | | | | | |
| NPV of Recurring cost | ($75,000.00) | ($106,250.00) | ($134,151.79) | ($159,064.09) | ($181,307.23) | ($201,167.17) | ($201,167.17) |
| ovel all NPV | | | | | | | $105,238.81 |
| Over all ROI=Over all NPV / NPV of all cost | | | | | | | 0.52 |
| | | | | | | | |
| Break Even Analysis | | | | | | | |
| Yearly NPV cash flow | ($75,000.00) | $44,642.86 | $39,859.69 | $35,589.01 | $31,775.90 | $28,371.34 | |
| Over all NPV cash flow | ($75,000.00) | ($30,357.14) | $9,502.55 | $45,091.56 | $76,867.47 | $105,238.81 | |



**6.Pert Chart:**

PERT (Program Evaluation and Review Technique) chart is a project management tool used to schedule, organize, and coordinate tasks within a project. It is a method to analyze the tasks involved in completing a given project, especially the time needed to complete each task and to identify the minimum time needed to complete the total project. It is a project management tool that provides a graphical representation of a project's timeline. It breaks down the individual tasks of a project for analysis.

A PERT chart is a visual project management tool that's useful for mapping out project tasks and planning the overall project schedule. PERT charts are tools used to plan tasks within a project - making it easier to schedule and coordinate team members accomplishing the work

## Questions:

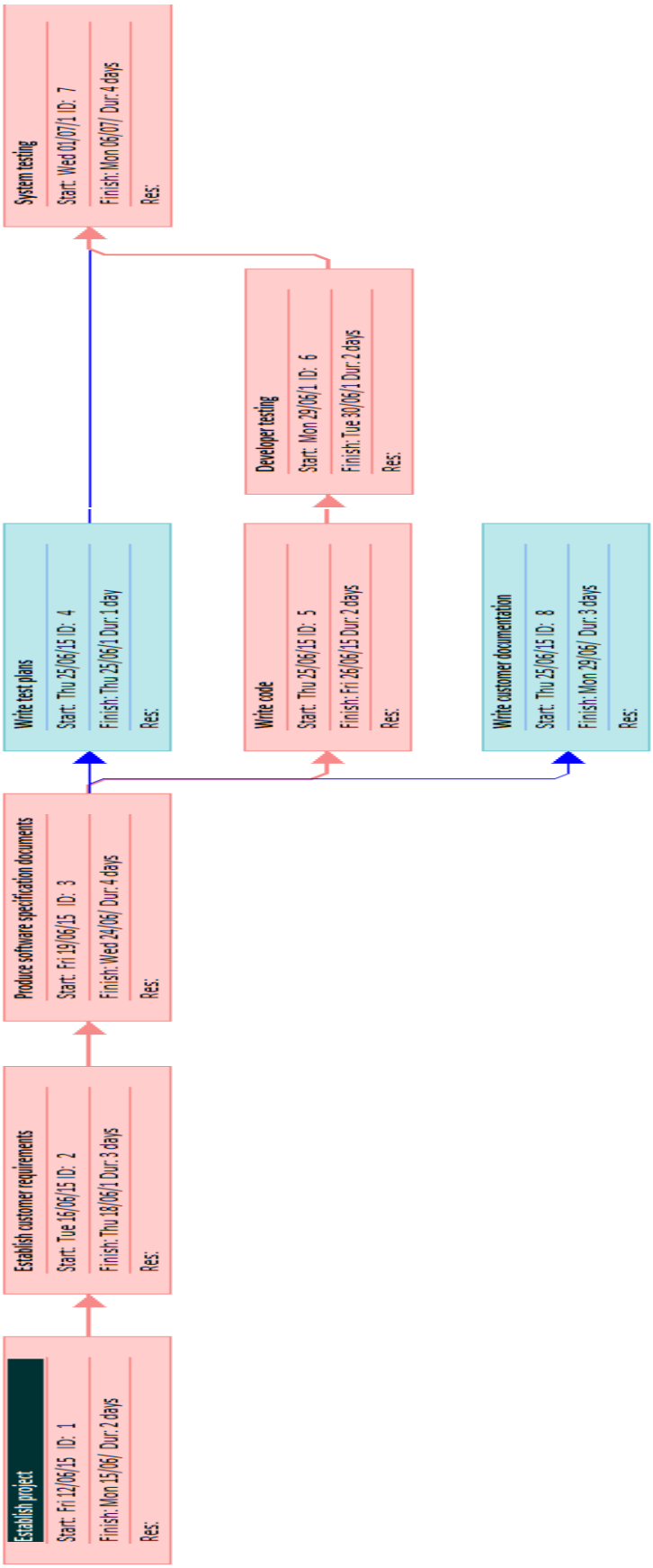
**a.Draw the PERT chart according to the following table which shows the tasks, dependencies, and estimated times for a software development project.**

| Project start date: 12 June 2015 | | | |
|---|---|---|---|
| Task Identifier | Task Description | Predecessor Task(s) | Time (days) |
| 1 | Establish project | - | 2 |
| 2 | Establish customer requirements | 1 | 3 |
| 3 | Produce software specification documents | 2 | 4 |
| 4 | Write test plans | 3 | 1 |
| 5 | Write code | 3 | 2 |
| 6 | Developer testing | 5 | 2 |
| 7 | System testing | 4, 6 | 4 |
| 8 | Write customer documentation | 3 | 3 |

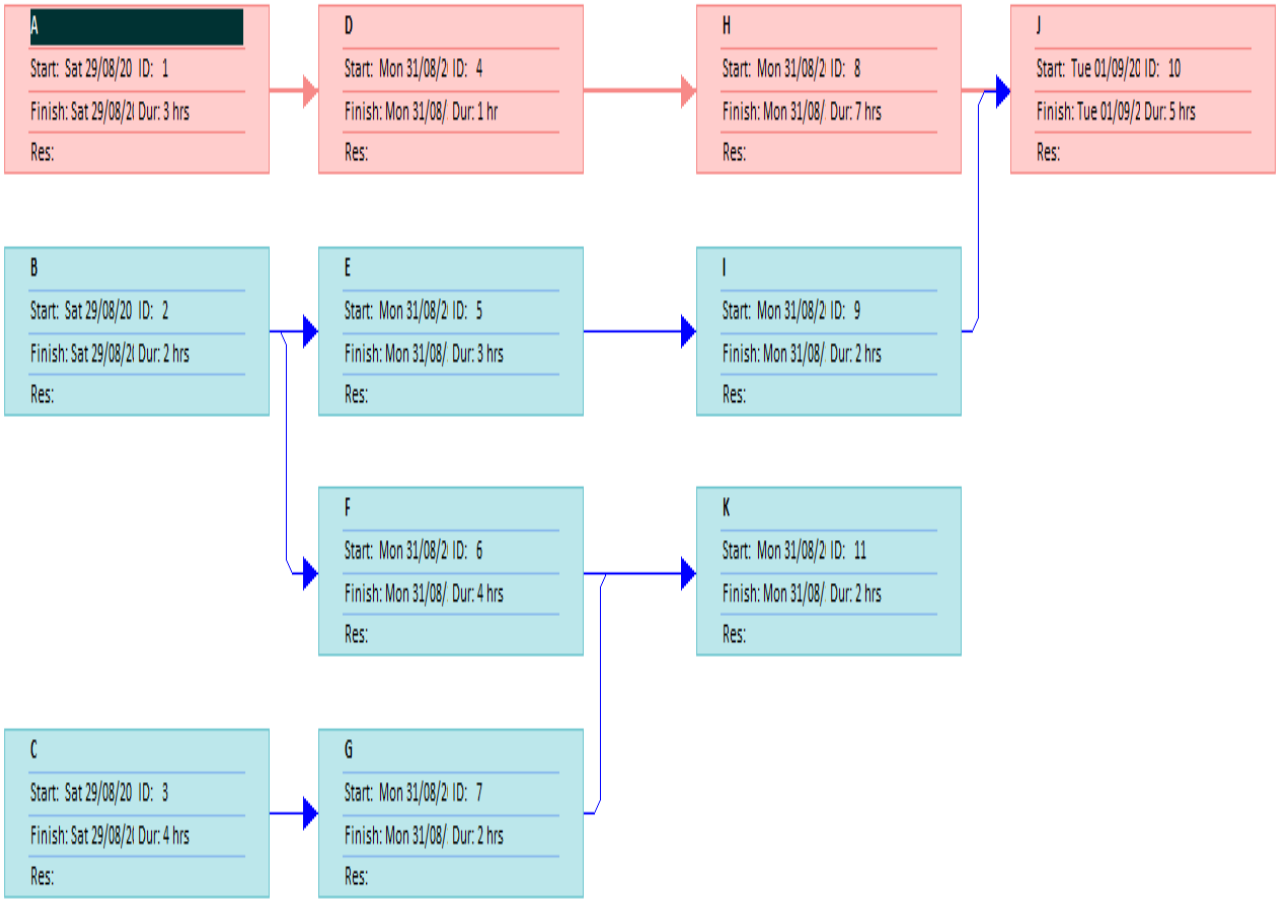
**b.Construct a network for a project whose activities and their predecessor relationship are given in table below.**

| Activity | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Predecessor | - | - | - | A | B | B | C | D | E | H, I | F, G |
| Time taken (hr) | 3 | 2 | 4 | 1 | 3 | 4 | 2 | 7 | 2 | 5 | 2 |

**a.Solution (should be viewed Horizontally)**



**b.Solution**



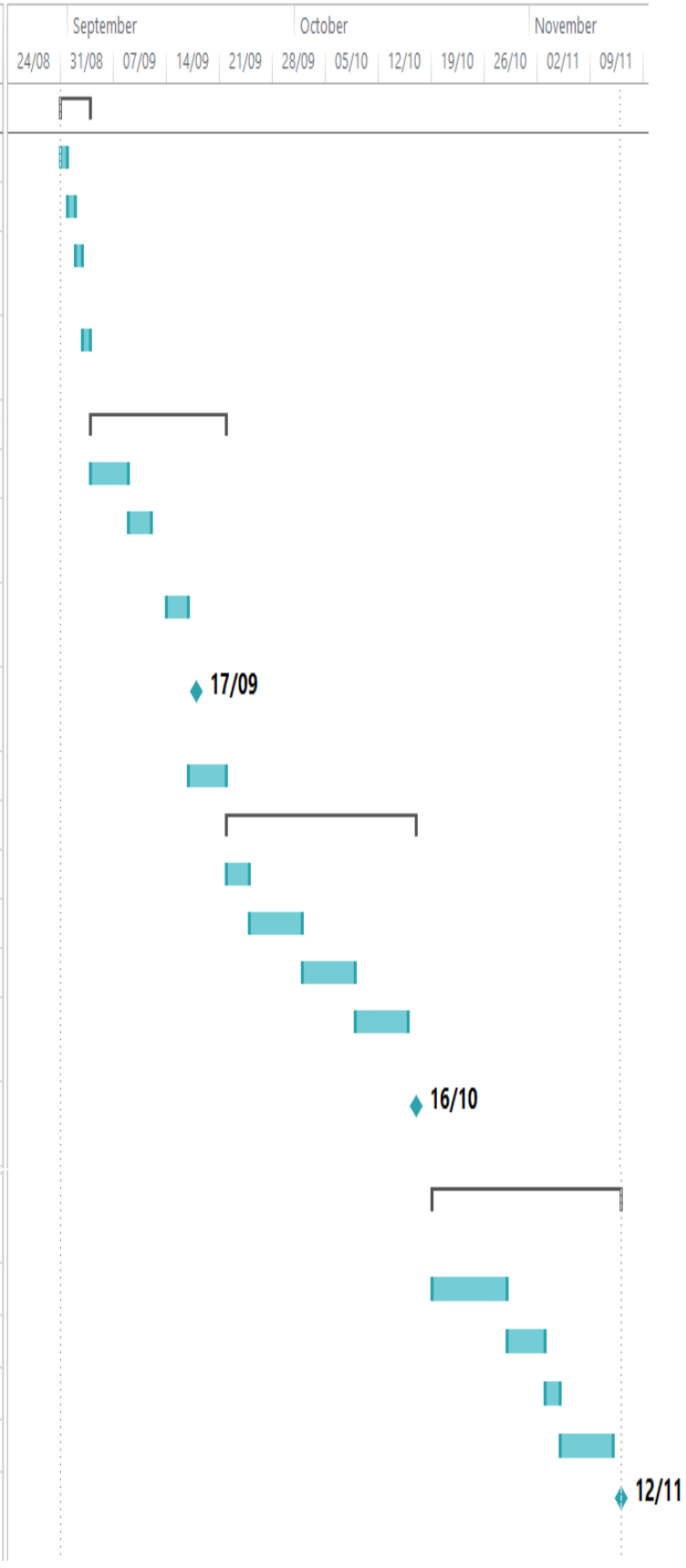

**7.Gantt Chart**

A Gantt chart is a type of horizontal bar chart commonly used in project management, which is a visual view of tasks scheduled overtime. It provides a graphical visualization of a schedule that helps to plan, coordinate, and track specific tasks (or elements) in a project. Gantt chart boils down multiple tasks and timelines into a single page. Using a Gantt chart allows all stakeholders to perceive the same schedule information, sets mutually understood expectations, and conducts their efforts according to the desired protocol. The Gantt chart tool provides a visual timeline for the start and end of tasks, making it clear how tasks are interrelated and perhaps rely on the completion of another before one can start.

A Gantt chart is a useful graphical tool which shows activities or tasks performed against time. It is also known as visual presentation of a project where the activities are broken down and displayed on a chart which makes it is easy to understand and interpret. Gantt chart is a useful tool in planning and scheduling the projects. It keeps the management updated as to when the project will get completed. It also keeps the management informed about any additional resources that are required, and manage dependencies between tasks.

**a.Solution**

| Task Name | Duration | Start | Finish |
|---|---|---|---|
| **PLANNING PHASE** | **4 days** | **Mon 31/08/2** | **Thu 03/09/20** |
| Select Team Memebe | 1 day | Mon 31/08/2( | Mon 31/08/20 |
| Define Problem | 1 day | Tue 01/09/20 | Tue 01/09/20 |
| Procedure Project Schedue | 1 day | Wed 02/09/20 | Wed 02/09/20 |
| Calculate Project Schedule | 1 day | Thu 03/09/20 | Thu 03/09/20 |
| **ANALYSIS PHASE** | **12 days** | **Fri 04/09/20** | **Mon 21/09/20** |
| Gather Information | 3 days | Fri 04/09/20 | Tue 08/09/20 |
| Define System Requirement | 3 days | Wed 09/09/20 | Fri 11/09/20 |
| Write the proposal Document | 3 days | Mon 14/09/20 | Wed 16/09/20 |
| Send proposal Document | 0 days | Thu 17/09/20 | Thu 17/09/20 |
| Draw logical DFD | 3 days | Thu 17/09/20 | Mon 21/09/20 |
| **DESIGN PHASE** | **19 days** | **Tue 22/09/20** | **Fri 16/10/20** |
| Design Currnet Proces | 3 days | Tue 22/09/20 | Thu 24/09/20 |
| Design New Process | 5 days | Fri 25/09/20 | Thu 01/10/20 |
| Design Data Base | 5 days | Fri 02/10/20 | Thu 08/10/20 |
| Design Interface and Forms | 5 days | Fri 09/10/20 | Thu 15/10/20 |
| Send Design Document | 0 days | Fri 16/10/20 | Fri 16/10/20 |
| **IMPLEMENTATION PHASE** | **19 days** | **Mon 19/10/20** | **Thu 12/11/20** |
| Do Coding | 8 days | Mon 19/10/2( | Wed 28/10/20 |
| Do Testing | 3 days | Thu 29/10/20 | Mon 02/11/20 |
| Do Acceptence Test | 2 days | Tue 03/11/20 | Wed 04/11/20 |
| Write Report Test | 5 days | Thu 05/11/20 | Wed 11/11/20 |
| Do Final Class Presentation | 0 days | Thu 12/11/20 | Thu 12/11/20 |



**b.Solution**

| Task Name | Duration | Start | Finish |
|---|---|---|---|
| **Feasibility report** | **4 days** | **Mon 01/03/2** | **Thu 04/03/21** |
| Technical | 1 day | Mon 01/03/21 | Mon 01/03/21 |
| Economical | 1 day | Tue 02/03/21 | Tue 02/03/21 |
| Operational | 1 day | Wed 03/03/21 | Wed 03/03/21 |
| Schedule | 1 day | Thu 04/03/21 | Thu 04/03/21 |
| **Framework pieces** | **5 days** | **Fri 05/03/21** | **Thu 11/03/21** |
| Performance | 1 day | Fri 05/03/21 | Fri 05/03/21 |
| Information | 1 day | Sat 06/03/21 | Sat 06/03/21 |
| Economy | 1 day | Sun 07/03/21 | Sun 07/03/21 |
| Control | 1 day | Mon 08/03/21 | Mon 08/03/21 |
| Efficiency and Service | 1 day | Fri 09/04/21 | Fri 09/04/21 |
| Investigation of (PI) | 2 days | Sat 10/04/21 | Mon 12/04/21 |
| Requirements of users | 2.5 days | Sat 13/03/21 | Tue 16/03/21 |
| Analysis of system needs | 4 days | Wed 17/03/21 | Mon 22/03/21 |
| Development and Documentation | 8 days | Tue 23/03/21 | Thu 01/04/21 |
| Cost Benefit Analysis | 2 days | Fri 02/04/21 | Mon 05/04/21 |
| Payback Analysis | 1 day | Mon 05/04/21 | Mon 05/04/21 |
| **Diagrams** | **15 days** | **Mon 05/04/2** | **Fri 23/04/21** |
| Pert chart | 2 days | Mon 05/04/21 | Tue 06/04/21 |
| Data Flow | 1 day | Tue 06/04/21 | Tue 06/04/21 |
| Level 0 Diagrams | 2 days | Wed 07/04/21 | Thu 08/04/21 |
| Level 1 Diagrams | 4 days | Fri 09/04/21 | Wed 14/04/21 |
| Data store | 8 days | Thu 15/04/21 | Mon 26/04/21 |
| All Entities Software | 6 days | Tue 27/04/21 | Tue 04/05/21 |
| Screen Shot Making | 2 days | Wed 05/05/21 | Thu 06/05/21 |