TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

MODULE:1

Overview of IT Industry

1. **What is software? What is software engineering?**

Software :

- Software is a collection of instructions, data or programs used to operate computers and execute specific tasks.There are two main types of software system softwares and application softwares.

- Software acts as an intermediary between the user and the computer hardware, enabling the hardware to perform tasks based on user commands.

- Software development involves writing code in programming languages, testing, and maintaining the software to ensure it functions correctly and efficiently.

- Software development involves writing code in programming languages, testing, and maintaining the software to ensure it functions correctly and efficiently.

Software engineering:

- Software engineering is a branch of computer science that involves the systematic design, development, testing, and maintenance of software applications.It applies engineering principles and practices to software development, ensuring that software is reliable, efficient, and meets user requirements.

- The software engineering process includes requirements analysis, system design, coding, testing, and maintenance. It may specialize in various roles such as systems engineering, application development, quality assurance, and project management.

- It addresses the challenges of building large, complex software systems, ensuring they are scalable, maintainable, and adaptable to changing needs.

2. Explain types of software.

- The various types of software are as follows:

  - System Software: This includes the operating systems (like Windows, macOS, Linux) and utility programs that manage and support the computer's basic functions. It acts as a bridge between the hardware and the user.

  - Application Software: These are programs designed for end-users to perform specife tasks such as word processing (Microsoft Word), web browsing (Google Chrome), and gaming (Fortnite).

  - Programming Software: This type includes tools that developers use to write, test, and maintain software programs. Examples are compilers, debuggers, and integrated development environments (IDEs) like Visual Studio.

  - Driver Software: These are specialized programs that allow the operating system to communicate with hardware devices like printers, graphics cards, and network adapters.

- Firmware: This is a specialized form of software that provides low-level control for a device's specific hardware. It is typically embedded directly into the hardware of devices like routers, cameras, and embedded systems.

3. **What is SDLC? Explain each phase of SDLC.**

- The Software Development Life Cycle (SDLC) is a structured process used for developing software applications. It is a systematic approach to software development that outlines the steps involved in planning, creating, testing, and deploying software.The main goal of SDLC is to produce high-quality software that meets or exceeds customer expectations, within time and cost estimates. It helps ensure a systematic, disciplined, and quantifiable approach to software development, reducing risks and improving efficiency.

- **There are six steps of phases there are as follows:**

- **Planning/requirement gathering:**

    - Define the project scope, objectives, and feasibility.
    - Identify resources, set timelines, and estimate costs. This phase involves creating a project plan and risk management plan.

- **Requirements Analysis:**

    - Gather and analyze the requirements from stakeholders.

    - Conduct interviews, surveys, and document analysis to understand what the users need. Create detailed requirement specifications that will guide the design and development.

- **Design:**

    - Create the architecture and design of the software.

    - Develop high-level design (HLD) and low-level design (LLD) documents. This includes designing the system architecture, database schema, user interfaces, and other components.

- **Development:**

    - Write the actual code based on the design documents.

    - Developers use programming languages and tools to build the software. This phase involves coding, unit testing, and integration of different modules.

- **Testing:**

    - Verify that the software works as intended and is free of defects.

    - Perform various types of testing such as unit testing, integration testing, system testing, and user acceptance testing (UAT). Identify and fix bugs to ensure the software meets the requirements.

- **Deployment:**

    - Release the software to the users.

    - Deploy the software to the production environment. This phase may involve installation, configuration, and user training. Ensure that the software is operational and accessible to the end-users.

- **Maintenance**:

    - Update and fix the software as needed.

    - Monitor the software for issues, provide updates, and make necessary enhancements. This phase ensures the software remains functional and relevant over time.

4. **What is DFD? Create a DFD diagram on Flipkart.**

- A Data Flow Diagram (DFD) is a graphical representation that maps out the flow of information within a system or process. It uses symbols like rectangles, circles, and arrows to show data inputs, outputs, storage points, and the routes between each destination.

- It helps in understanding how data moves through a system, making it easier to analyze and design systems.

- They typically include processes, data stores, data flows, and external entities.

- They typically include processes, data stores, data flows, and external entities.
- DFDs can be hierarchical, starting from a high-level overview (Level 0) and breaking down into more detailed levels (Level 1, Level 2, etc.).

- DFDs are useful for both technical and non-technical audiences, making them a popular tool in software engineering and business analysis.

- **Types of DFD:**

1. Context-Level DFD (Level 0 DFD):

- **Definition**: It is the simplest, high-level diagram that provides an overview of the entire system.

- **Purpose**: Shows the system as a single process with interactions from external entities (e.g., users, customers, or external systems).

- **Example**: A "Customer" interacts with an "Online Shopping System" to place an order, and the system responds with confirmation.

Key Feature: No detailed breakdown of internal processes.

2. Level 1 DFD:

- **Definition**: This diagram breaks down the main process from the Level 0 DFD into sub-processes to provide more detail.

- **Purpose:** Helps to understand how data flows between different processes within the system.

- **Example**: In an "Online Shopping System," Level 1 DFD might show sub-processes like "Search Products," "Place Order," and "Process Payment."

Key Feature: Detailed view of how the system works internally.

3. Level 2 (and more detailed levels):

- **Definition:** These diagrams further break down sub-processes from Level 1 into even more detailed steps.

- **Purpose:** Provide in-depth understanding of complex processes within the system.

- **Example**: "Process Payment" could be broken down into "Verify Payment" and "Process Transaction."

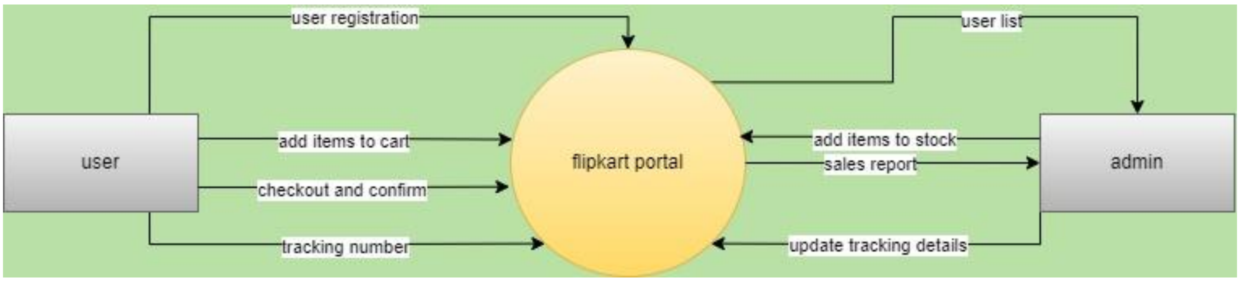
--- DFD ON FLIPKART:

➢ **LEVEL 0 DFD:**

Components of the DFD:

1. **External Entities:**
   - **User:** Represents the customer who interacts with the system.
   - **Admin:** Represents the administrative staff who manage the system.
2. **Main Process:**
   - **Flipkart Portal:** The central process/system that handles all the operations.

3. **Data Flows:**
    - **User to Flipkart Portal:**
        - **User Registration:** Flow of data when a user registers on the portal.
        - **Add Items to Cart:** When a user adds items to their shopping cart.
        - **Checkout and Confirm:** When a user proceeds to checkout and confirms their purchase.
        - **Tracking Number:** The system provides a tracking number to the user post-purchase.
    - **Flipkart Portal to Admin:**
        - **Add Items to Stock:** Admin adds items to the system's stock.
        - **Sales Report:** The system provides sales reports to the admin.
        - **Update Tracking Details:** Admin updates tracking details in the system.
        - **User List:** The admin has access to the list of registered users.



LEVEL 0 DFD.

2. LEVEL 01 DFD ON FLIPKART:

- **External Entities:**

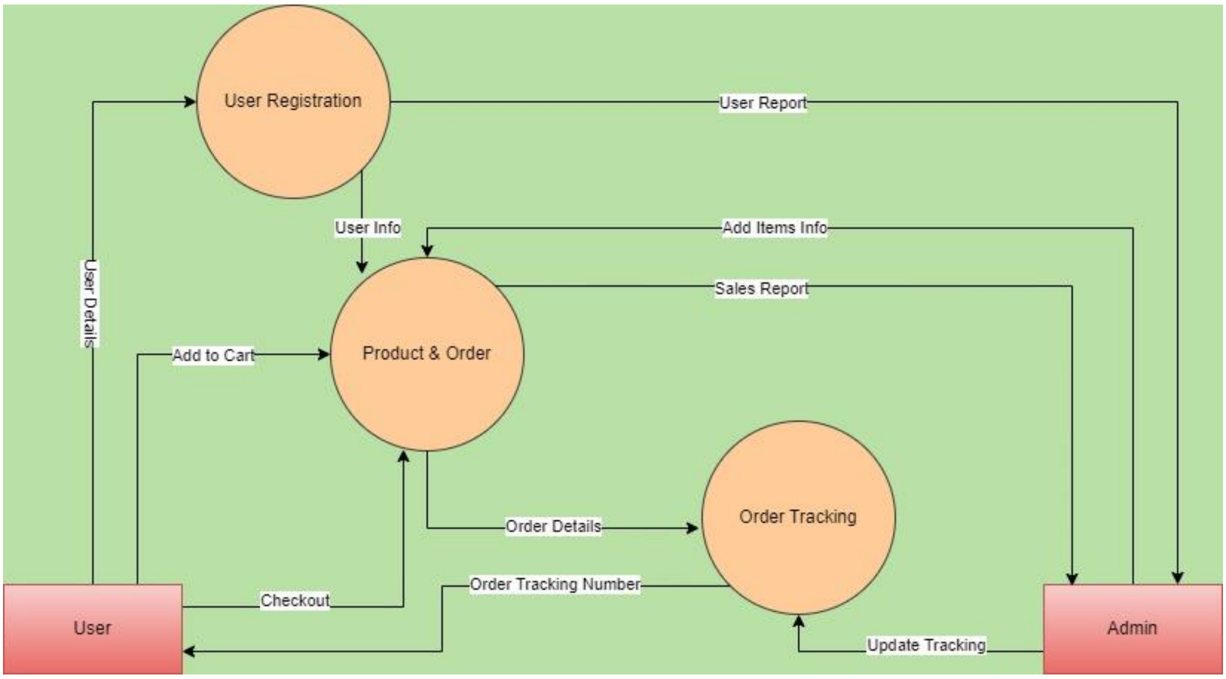
    - **User:** Represents the customer interacting with the system.
    - **Admin:** Represents the administrative staff managing the system.

- **Processes:**

  - **User Registration:** Handles the registration of new users and the management of user information.
  - **Product & Order:** Manages the products, the process of adding items to the cart, and the checkout process.
  - **Order Tracking:** Handles the tracking of orders and updating of tracking information.

- **Data Flows:**

  - **User Registration Process:**
    - **User Details** flow from the User to the User Registration process.
    - The **User Info** generated from this process flows to both the Product & Order process and is used to generate a **User Report** for the Admin.
  - **Product & Order Process:**
    - **Add to Cart:** User sends data to add items to the cart.
    - **Checkout:** User completes the purchase, triggering the checkout process.
    - **Order Details** are sent from this process to the Order Tracking process.
    - **Sales Report** is generated and sent to the Admin.
  - **Order Tracking Process:**
    - **Order Tracking Number:** Sent to the User after the order details are processed.
    - **Update Tracking:** Admin updates the tracking information, which is then used to update the User.
  - **Admin Interaction:**
    - **User Report:** Admin receives reports on user activities.
    - **Add Items Info:** Admin adds items to the stock via the Product & Order process.
    - **Update Tracking:** Admin updates order tracking information in the system.



LEVEL 1 DFD.

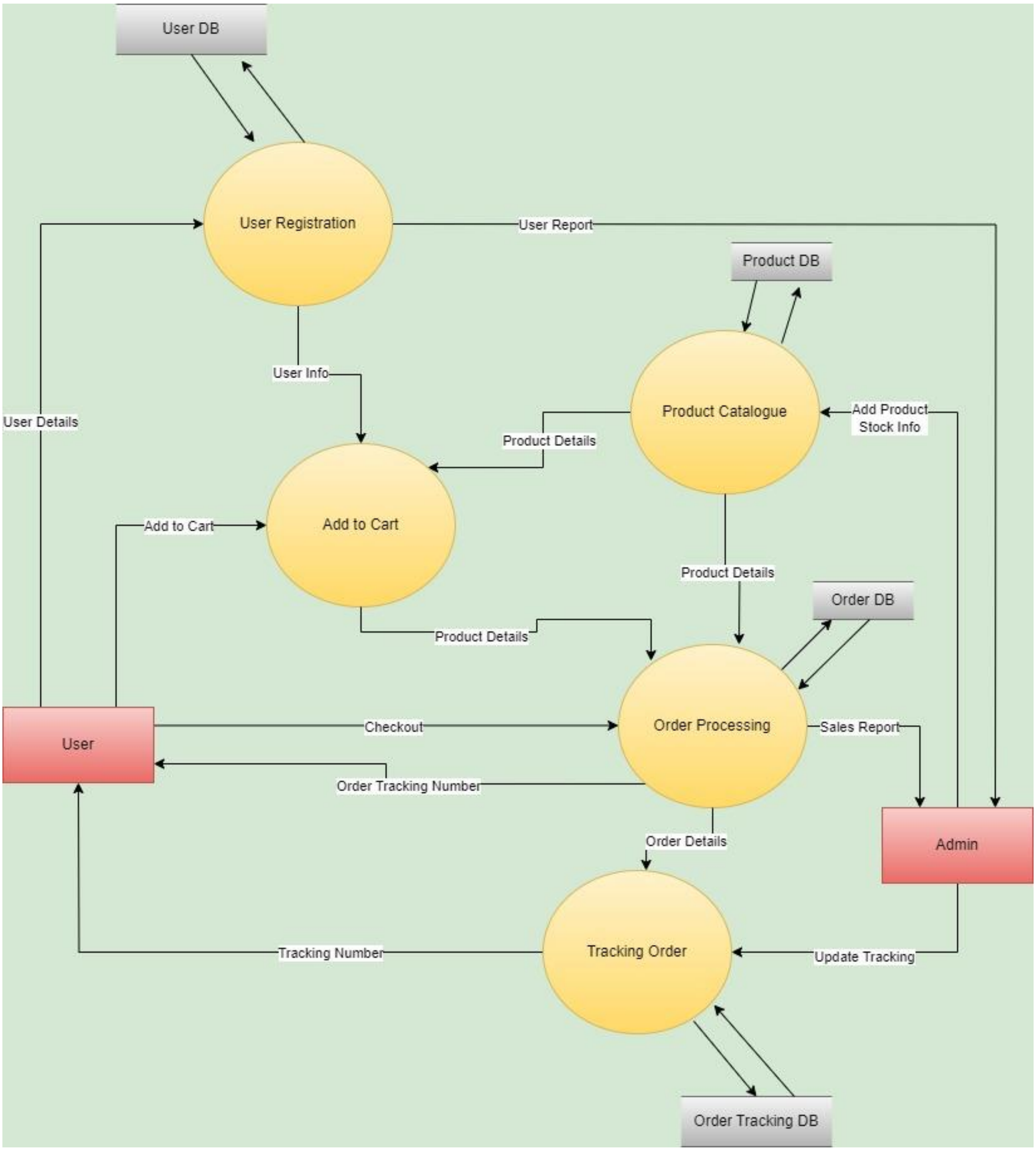
4. LEVEL 2 DFD ON FLIPKART:

- **External Entities:**
  - **User:** The customer interacting with the system.
  - **Admin:** The administrative user who manages the system.
- **Processes:**
  - **User Registration:** Manages user registration and user information.
  - **Product Catalogue:** Handles the management of product details and stock information.
  - **Add to Cart:** Manages the process of users adding items to their shopping cart.
  - **Order Processing:** Manages the checkout process and order processing.
  - **Tracking Order:** Handles the tracking of orders and updating tracking details.
- **Data Stores:**
  - **User DB (Database):** Stores user details.

  - **Product DB:** Stores product details and stock information.
  - **Order DB:** Stores order details.
  - **Order Tracking DB:** Stores tracking information for orders.

- **Data Flows:**

  - **User Registration Process:**
    - **User Details** flow from the User to the User Registration process and are stored in the **User DB**.
    - **User Info** is sent to the **Add to Cart** process and used to generate a **User Report** for the Admin.
  - **Product Catalogue Process:**
    - **Product Details** flow into the process from the **Product DB**.
    - **Add Product, Stock Info:** Admin updates product and stock details, which are stored in the **Product DB**.
    - **Product Details** flow to both the **Add to Cart** and **Order Processing** processes.
  - **Add to Cart Process:**
    - **Add to Cart:** User adds items to the cart.
    - **Product Details** flow from the Product Catalogue to the cart.
    - **Checkout:** User checks out, and the details are sent to the **Order Processing** process.
  - **Order Processing Process:**
    - **Product Details** and **Order Details** are processed and stored in the **Order DB**.
    - **Order Tracking Number:** Generated and sent to the user.
    - **Sales Report:** Generated and sent to the Admin.
  - **Tracking Order Process:**
    - **Order Details** flow from the **Order Processing** process.
    - **Order Tracking Number** is sent to the user.
    - **Update Tracking:** Admin updates the tracking details, which are stored in the **Order Tracking DB**.
    - **Tracking Number** is also sent back to the User.


User DB
User Registration
User Report
Product DB
User Info
User Details
Product Catalogue
Add Product
Stock Info
Product Details
Add to Cart
Add to Cart
Product Details
Order DB
Product Details
User
Checkout
Order Processing
Sales Report
Order Tracking Number
Admin
Order Details
Tracking Order
Tracking Number
Update Tracking
Order Tracking DB

LEVEL 2 DFD.

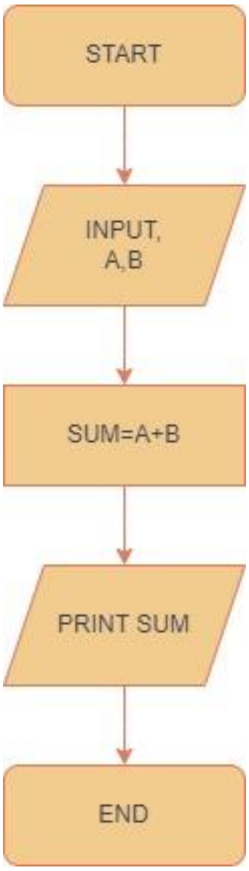
5. **What is Flow chart? Create a flowchart to make addition of two numbers.**

- A flowchart is a visual representation of the steps in a process, system, or algorithm. It uses various shapes like rectangles, diamonds, ovals, and arrows to illustrate the sequence of actions or decisions needed to complete a task.
- Flowcharts help in understanding and documenting processes, making them easier to analyze and improve.

- Components: Common symbols include in flow charts are:

  - **Ovals**: Indicate the start and end points.
  - **Rectangles:** Represent process steps or actions.
  - **Diamonds:** Show decision points.
  - **Arrows:** Indicate the flow of the process.

- They are widely used in business, engineering, education, and many other fields to map out processes, plan projects, and document workflows.



Here's the logic for a flowchart that performs the addition of two numbers:

1. **Start**: The process begins.

2. **Input Numbers:** Get the first number and the second number from the user.

3. **Add the Two Numbers:** Perform the addition of the two numbers

4. **Display the Result:** Output the sum of the two numbers.

5. **End:** The process ends

6.What is Use case Diagram? Create a use-case on bill payment on paytm.

- A use case diagram is a type of Unified Modeling Language (UML) diagram that visually represents the interactions between users (or "actors") and a system to achieve specific goals.

- It helps in understanding the functional requirements of a system from a user's perspective, making it easier to communicate with stakeholders and guide the development process.

- The components of use case diagram are:

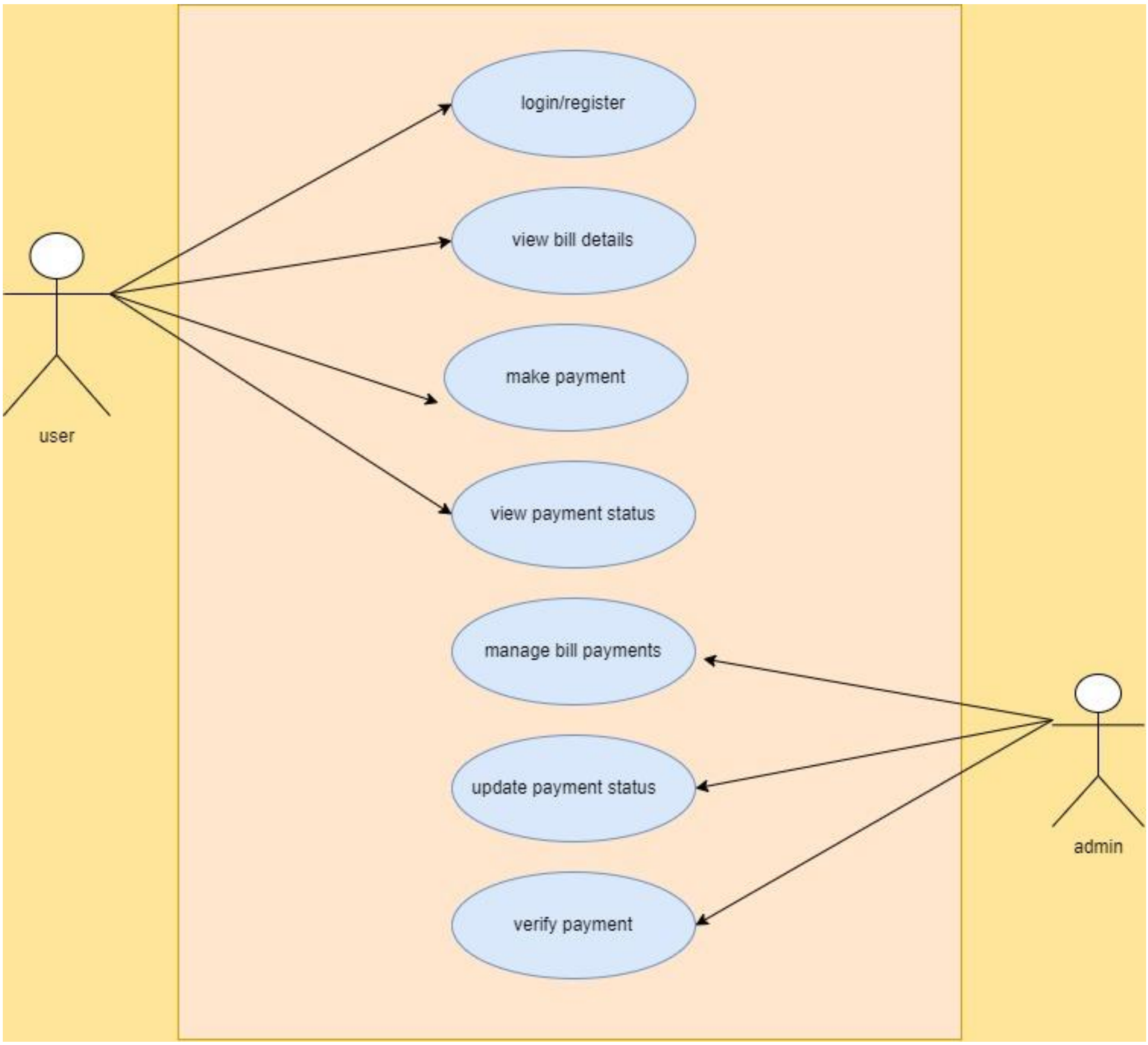
- **Actors:** External entities that interact with the system, such as users or other systems.

- **Use Cases**: Specific actions or services the system performs, represented by ovals.

- **System Boundary**: A rectangle that defines the scope of the system, enclosing all the use cases.

- Use case diagrams also show the relationships between actors and use cases, such as associations, generalizations, and dependencies.

- Use case diagrams provide a high-level view of the system's functionality and are useful for both technical and nontechnical audiences.



Use Case diagram on bill payment