1. What is software? What is software engineering?

Software is set of instructions, data or programs used to operate computers and execute specific tasks. Software is a generic term used to refer to application, and programs that run on a device.

Software is responsible for directing all computer-related devices and instructing them regarding what and how the task is to be performed. However, the software is made up of binary language (composed of ones and zeros), and for a programmer writing the binary code would be a slow and tedious task. Therefore, software programmers write the software program in various human-readable languages such as Java, Python, C#, etc. and later use the source code

What is software engineering?

Definition: Software engineering is a detailed study of engineering to the design, development and maintenance of software. Software engineering was introduced to address the issues of low-quality software projects. Problems arise when a software generally exceeds timelines, budgets, and reduced levels of quality.
Description: A software product is judged by how easily it can be used by the end-user and the features it offers to the user. An application must score in the following areas:-

1) Operational: -This tells how good a software works on operations like budget , usability, efficiency, correctness ,functionality , dependability , security and safety.

2) Transitional: - Transitional is important when an application is shifted from one platform to another. So, portability, reusability and adaptability come in this area.

3) Maintenance: - This specifies how good a software works in the changing environment. Modularity, maintainability, flexibility and scalability come in maintenance part.

MODULE: 1 (SDLC)

Software Development Lifecycle or SDLC is a series of stages in software engineering to develop proposed software application, such as:

1) Communication

2) Requirement Gathering

3) Feasibility Study

4) System Analysis

5) Software Design

6) Coding

7) Testing

8) Integration

9) Implementation

10) Operations and maintenance

11) Disposition

2. Explain types of software?

- **Application software:** The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

- **System software:** These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the system software include the firmware, computer language translators and system utilities. best example of system software; it manages all the other computer programs. Other examples of

- **Driver software:** Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

- **Middleware:** The term middleware describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

- **Programming software:** Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

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

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond.

Requirements Gathering

Gathering Requirement and Analysis | Second Step In SDLC | BetsolThe second step of SDLC is gathering maximum information from the client requirements for the product. Discuss each detail and specification of the product with the customer.

Analysis

The development team will then analyze the requirements keeping the design and code of the software in mind. Further, investigating the validity and possibility of incorporating these requirements into the software system. The main goal of this stage is that everyone understands even the minute detail of the requirement. Hardware, operating systems, programming, and security are to name the few requirements.

Design

Design | Third Step In SDLC | BetsolIn the design phase (3rd step of SDLC), the program developer scrutinizes whether the prepared software suffices all the requirements of the end-user. Additionally, if the project is feasible for the customer technologically, practically, and financially. Once the developer decides on the best design approach, he then selects the program languages like Oracle, Java, etc., that will suit the software.

Once the design specification is prepared, all the stakeholders will review this plan and provide their feedback and suggestions. It is absolutely mandatory to collect and incorporate stakeholder’s input in the document, as a small mistake can lead to cost overrun.

Implementation

Implementation | Fourth Step In SDLC | BetsolTime to code! It means translating the design to a computer-legible language. In this fourth stage of SDLC, the tasks are divided into modules or units and assigned to various developers. The developers will then start building the entire system by writing code using the programming languages they chose. This stage is considered to be one of the longest in SDLC. The developers need certain predefined coding guidelines, and programming tools like interpreters, compilers, debugger to implement the code.

The developers can show the work done to the business analysts in case if any modifications or enhancements required.

Testing

Testing | Fifth Step In SDLC | BetsolOnce the developers build the software, then it is deployed in the testing environment. Then the testing team tests the functionality of the entire system. In this fifth phase of SDLC, the testing is done to ensure that the entire application works according to the customer requirements.

After testing, the QA and testing team might find some bugs or defects and communicate the same with the developers. The development team then fixes the bugs and send it to QA for a re-test. This process goes on until the software is stable, bug-free and working according to the business requirements of that system.

Deployment

Development | Sixth Step In SDLC | BetsolThe sixth phase of SDLC: Once the testing is done, and the product is ready for deployment, it is released for customers to use. The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software. Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

Maintenance

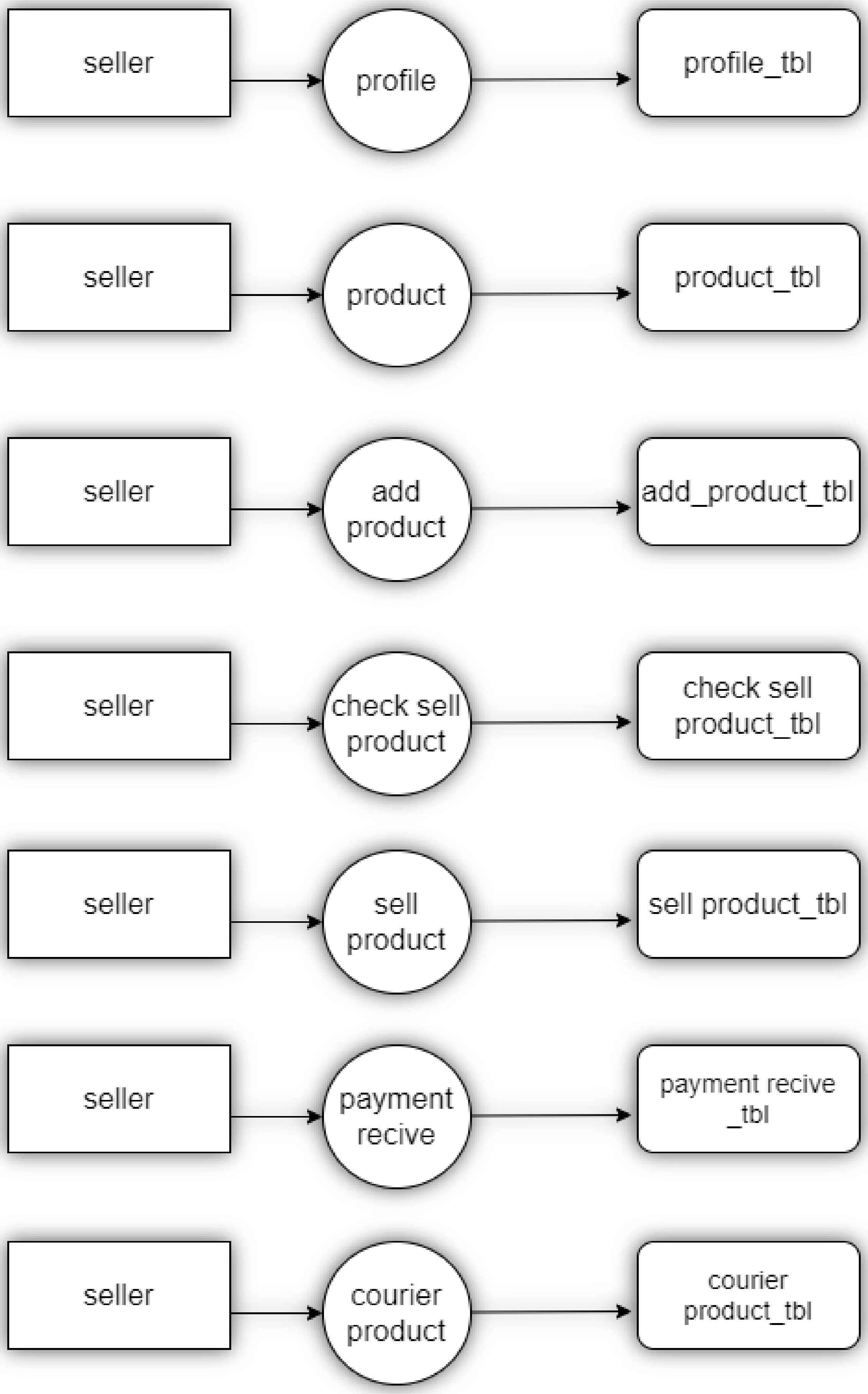
Maintenance | Seventh Step In SDLC | BetsolThe actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. Maintenance is the seventh phase of SDLC where the developed product is taken care of. According to the changing user end environment or technology, the software is updated timely.

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

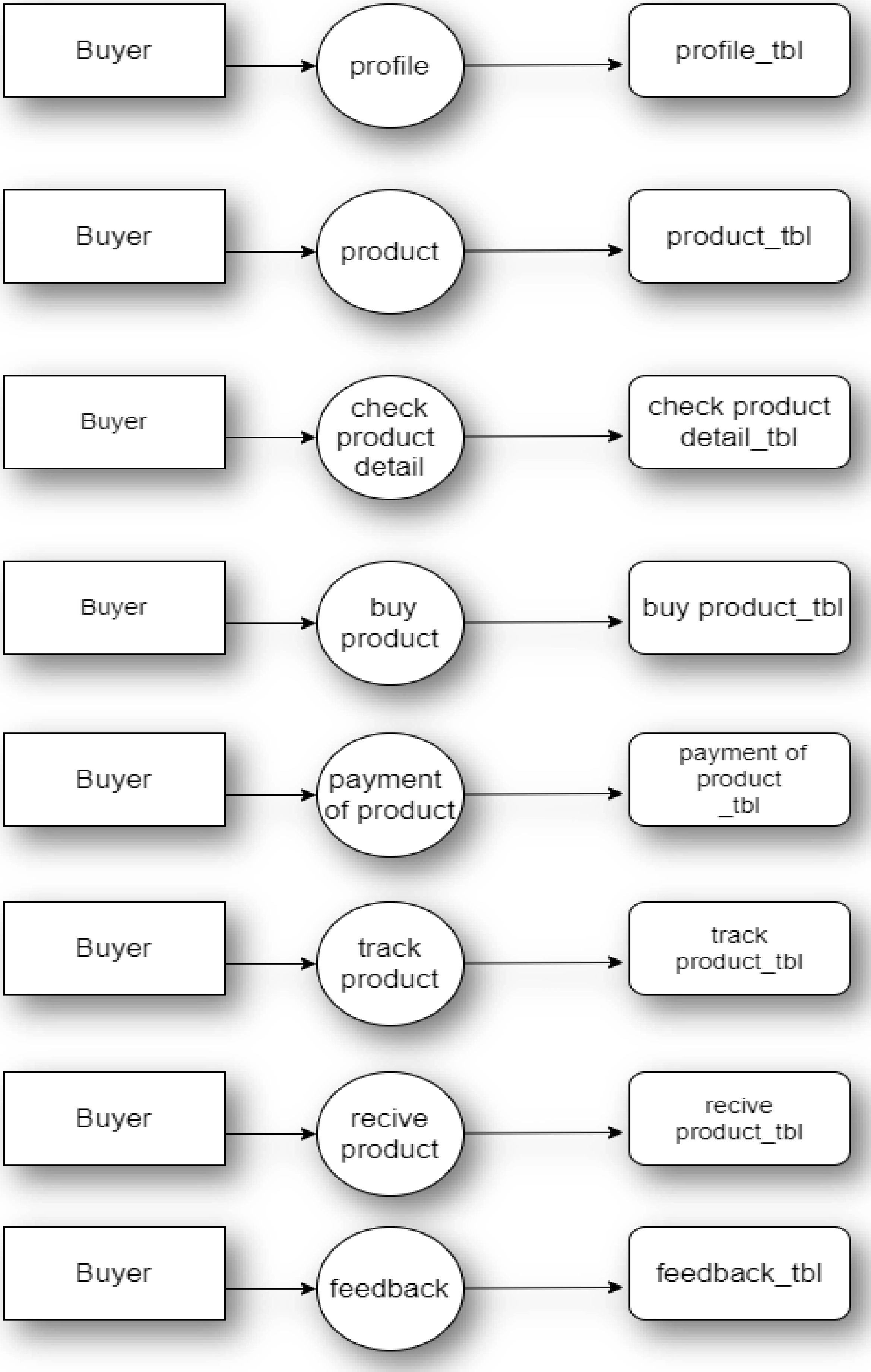
A data flow diagram is a graphical or visual representation using a standardized set of symbols and notation to describe a business operation through data movement.

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart.

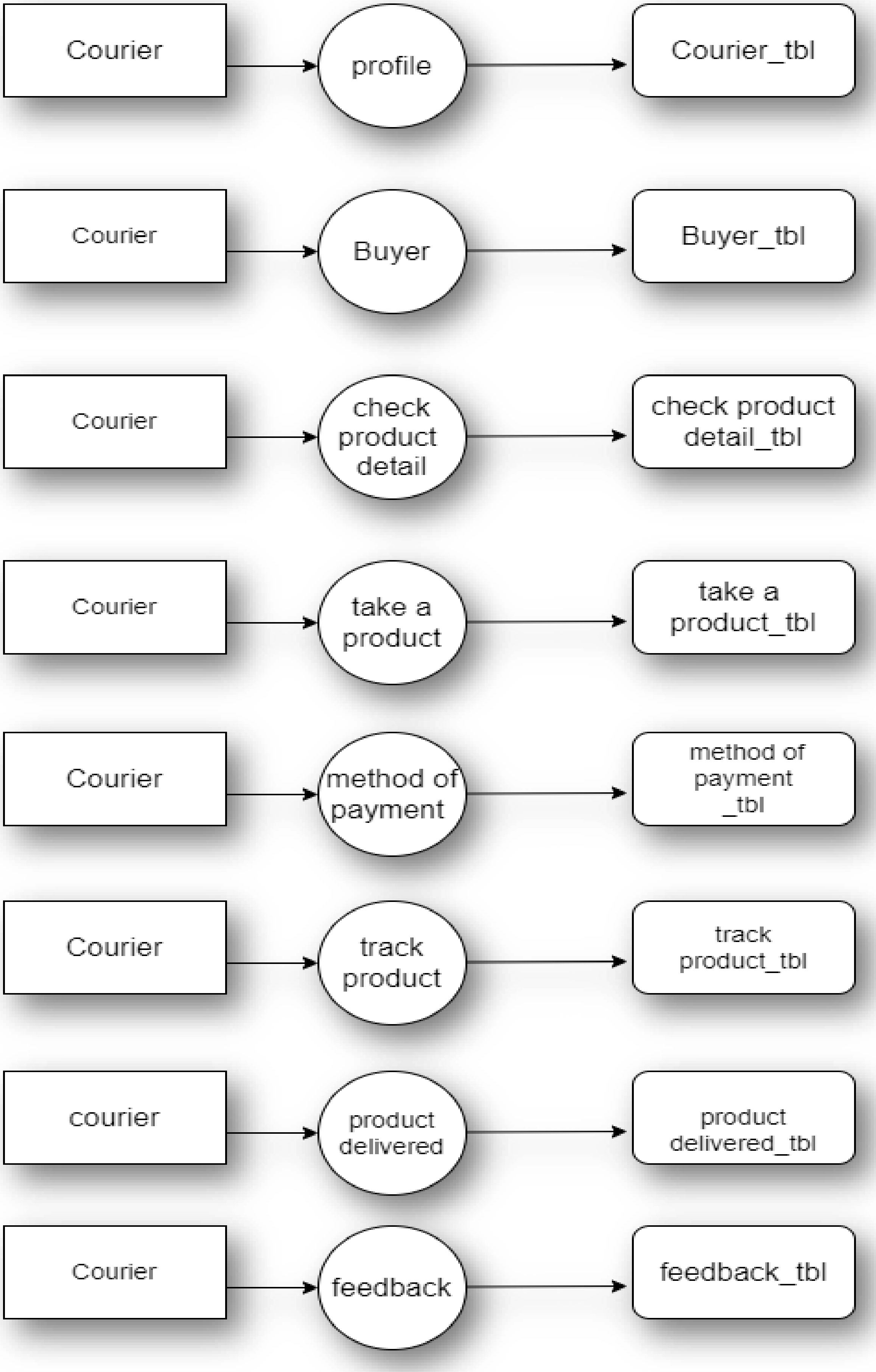
MODULE: 1 (SDLC)



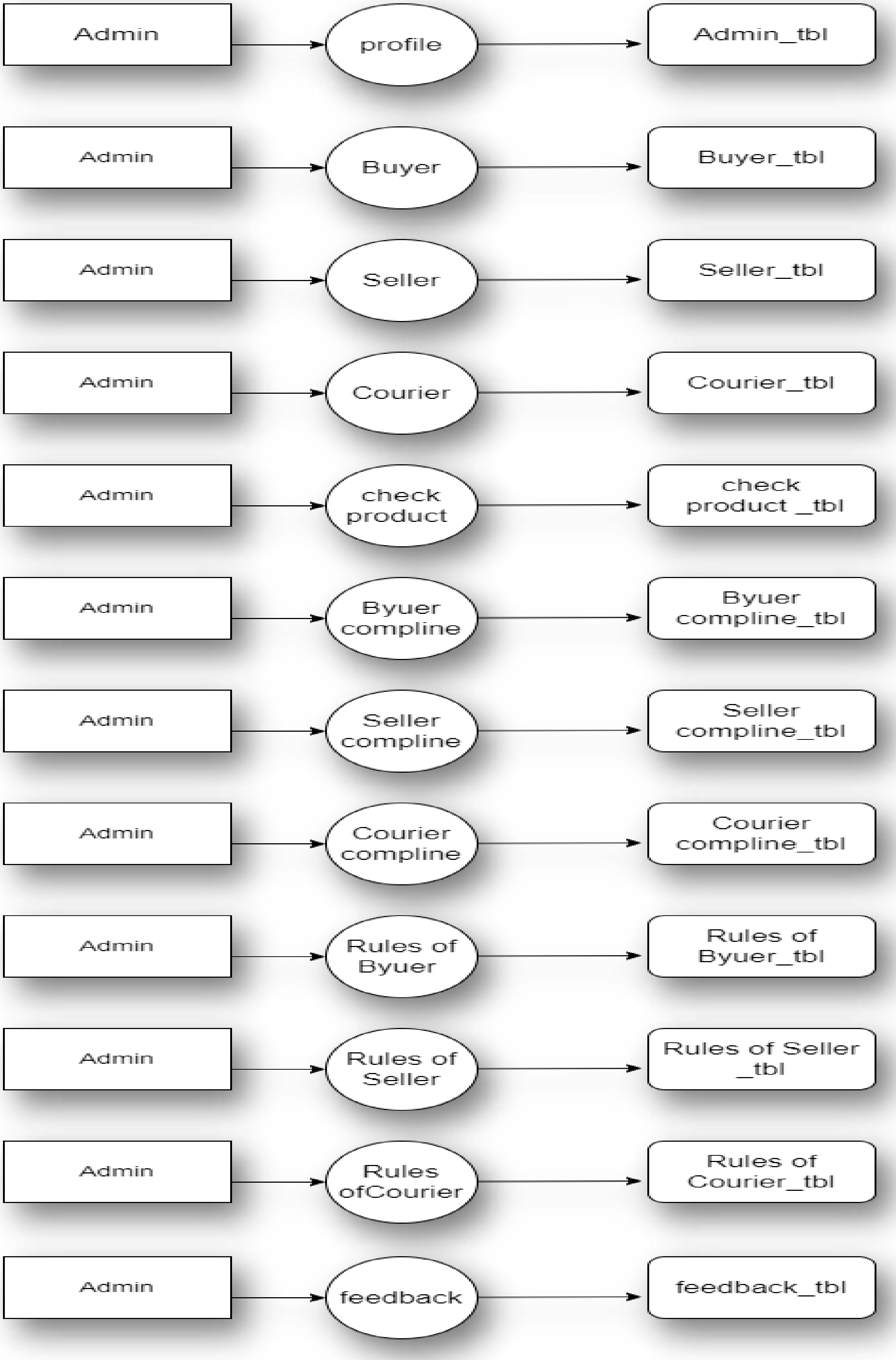
MODULE: 1 (SDLC)



MODULE: 1 (SDLC)



MODULE: 1 (SDLC)

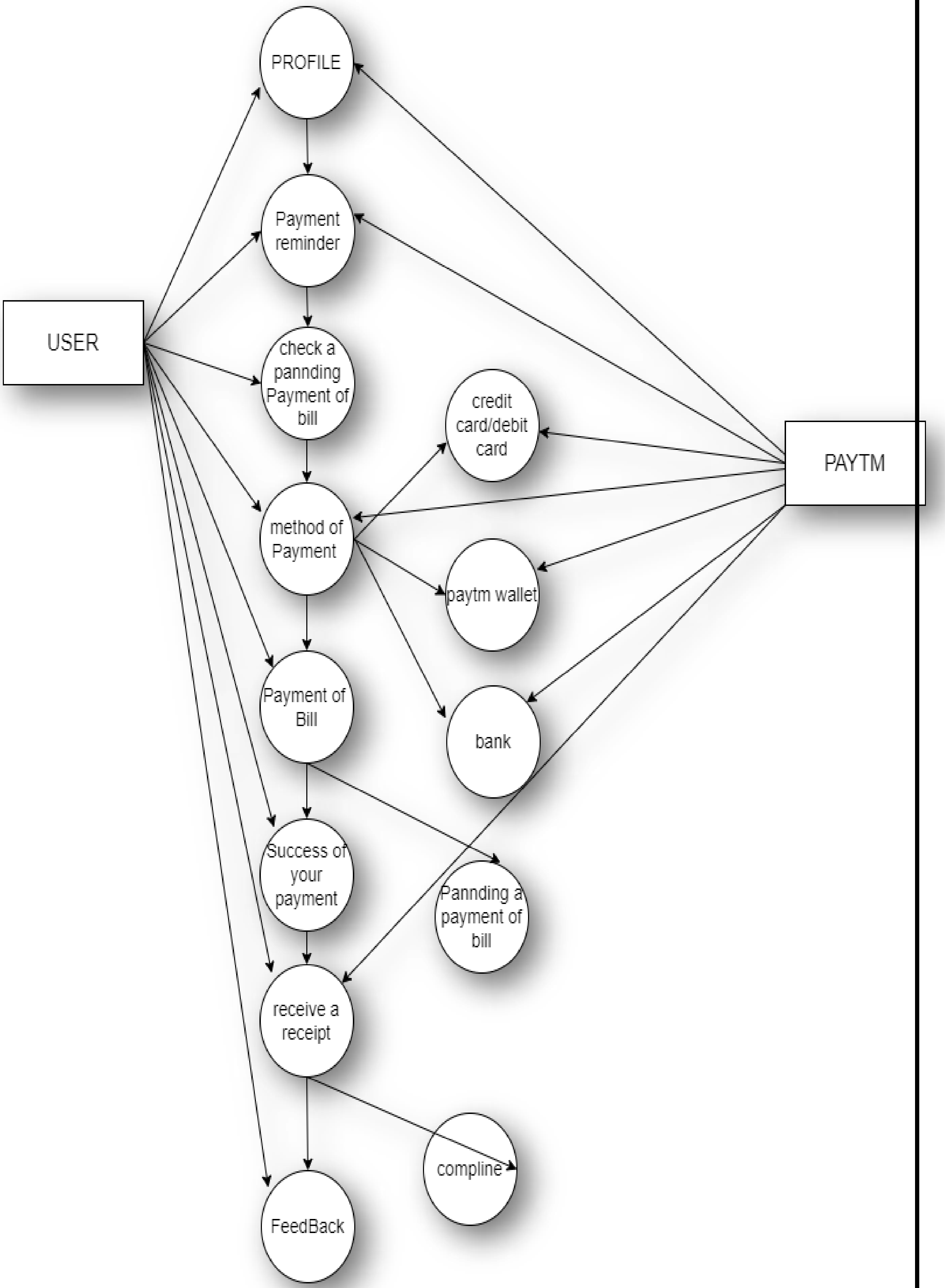


5. **What is Use case Diagram? Create a use-case on bill payment on paytm?**

It only summarizes some of the relationships between use cases, actors, and systems

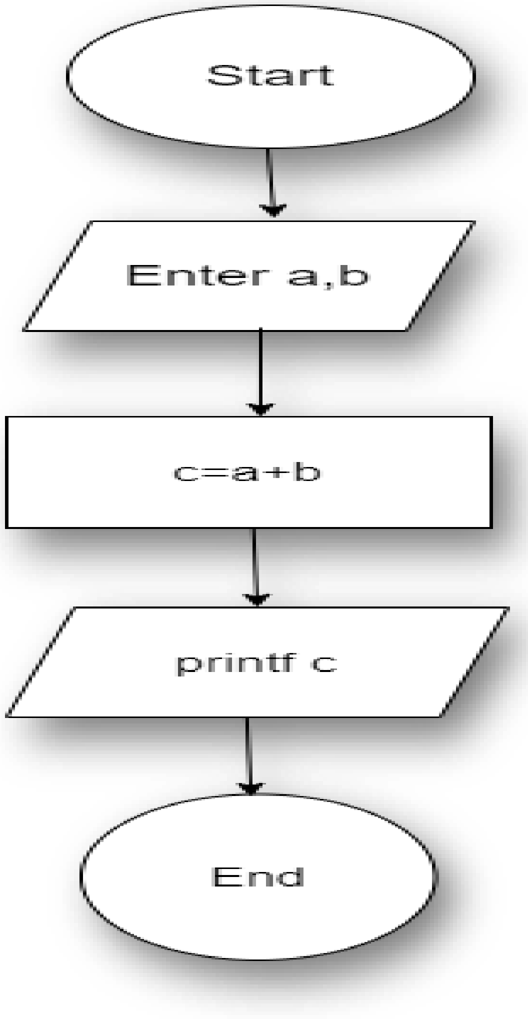
Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally

MODULE: 1 (SDLC)



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

A flowchart is a diagram that depicts a process system or computer algorithm. they are widely used in multiple fields to document study plan improve and communicate often complex process in clear , easy to understand diagrams.


Start
Enter a,b
c=a+b
printf c
End