C.M.SURGIMED

Submitted By:

1.Dixit Sathwara

2.Hiren Panchal

3.Taxil Parekh

As a part of submission of Web Application Project Semester-5

External Guide
Mr.Hiren Agrawal

Internal Guide
Dr.Sanjay B Sonar

Navgujarat College of Computer Applications
Gujarat University
Ashram Road, Ahmedabad

October-2022

C M SURGIMED
04 R KANTILALESTATE, OPP MAYUR SOAP FACTORY,
TAVDIPURA,DUDHESHWAR, SHAHIBAUGH,
AHMEDABAD– 380004 (Guj.)
GST No: 24AKMPA3848G1Z1

Date: 20/06/2022

To Sir,

C. M. SURGIMED,

04 R Kantilal Estate,

Opp Mayur Soap Factory,

Tavdipura,Dudheshwar,

Ahmedabad-380004

The Director,

Navgujarat collage of computer

Applications

Ashram Road

Ahmedabad

SUBJECT: Confirmation Letter for Project Allocation

Dear Sir,

We hereby confirm that the students from Navgujarat Collage of Computer Application of Sem-5 are granted the permission to develop the project for the duration of one year. We will provide the necessary information regarding the project. The project detailed are as under.

Project Title: C M SURGIMED

Technology: php

Student Details: (1) Dixit Sathwara

(2) Hiren Panchal

(3) Taxil Parekh

REGARDS,

HIREN AGRAWAL

(M): +91-9033458586 +91- 9106609995

Navgujarat College of Computer Applications

Navgujarat Campus, Ashram Road Ahmedabad
www.navgujaratbca.com 079-27542859

Certificate of Completion

This is to certify that Software Development Project "C.M.SURGIMED" is submitted to Gujarat University by:

1 Dixit Sathwara
2 Hiren Panchal
3 Taxil Parekh

They have successfully completed the project for the duration of six months (SEM-V) towards the partial fulfilment for the award of "Bachelors in Computer Applications [B.C.A.] in the year 2022.
This is the original work carried out under our guidance and supervision. We further certify that the details are best to our knowledge and believe that the matter presented in this project is Bonafied. We appreciate their enthusiasm and dedication towards work submitted.
Best wishes for their future endeavors.

(Internal Guide)	(Hon. Director)	(External Examiner)

Acknowledgement

We express our heart gratitude to number of people who extended their full support and co-operation in developing this project first, We would like take this opportunity to thank our College **"NAVGUJARAT COLLEGE OF COMPUTER APPLICATION"** for giving us this opportunity and a platform for discovering and developing our potential , This kind of experience that we have received while making this project report is so immense the narrating that in few word is difficult.

After putting in such a hard work we have realized that takes to work in shop and do a project. Our shop **"C.M.SURGIMED"** and colleagues have been great source of help without them we were unable to do this project. Therefore, our project as if small drop water of sea. We have learnt many think from being a part of concept family.

After that our heartiest thank is our internal guide as well as respected faculty for entrusting upon us the responsible and acting as ray of light in darkness. We find our self-short of world to describe our feeling for the role he played of friend, a philosopher and guide, whenever we were in need.

Lastly, we are thankful to our parents without blessing, Love and Support. We are unable to traverse through this most significant stage of life and also, we would Like to take this opportunity to express our regards to all friends and faculties who have helped us directly or indirectly during the executing of the project. We are privileged and thankful to all in bringing our errors and shortcoming.

Prepared By:

Dixit Sathwara

Hiren Panchal

Taxil Parekh

INDEX

1. ORGANIZATION PROFILE

Company Name	C.M.SURGIMED
Company Address	04 R kantilal Estate, opp Mayur Soap Factory, Tavdipura, Dudheshwar, Shahibaugh Ahmedabad-380004
Contact Person	Mr.Hiren Agrawal
Contact Number	9033458586

2. COMPANY PROFILE

Project Tital	C.M.SURGIMED
Numbers Of Members	3
Name of Members(Roll no)	Dixit Sathwara(96) Hiren Panchal(66) Taxil Parekh(69)
Project Duration	1 year
Internal Guide	Dr. Sanjay B Sonar
External Guide	Mr.Hiren Agrawal

3.ANALYSIS

❖ EXISTING SYSTEM:-

- Current system customer are placed order through phone calls, messages or face-to-face communication. In current system customer are not think about products are available or not.

❖ PROBLEM WITH EXISTING SYSTEM:-

- Current system is totally works manually.
- The existing system is based on phone call or face-to-face communication.
- Current system very hard to operate and maintain.
- The paper based work so the records are lose some time.

❖ PROPOSED SYSTEM (NEW SYSTEM):-

- The Online shopping web application is easy to customer because customer are purchase items in stay in home through our computer. This new system customer are view variety of products and what's products are unavailable(finished) and what's products are available.

❖ ADVANTAGE OF NEW SYSTEM:-

- Effective communication between admin and customer.
- Payment system are available.
- Home delivery is available.
- Customer aware about a products and see that what's product available or not.
- Product is nice or not given feedback.
- View a product review.

❖ TOOLS AND TECHNOLOGY:-

1) TECHNOLOGY:-

1. **Frontend**
    - **PHP**
    - **Html, Css, js, Boostrap**

2. **Backend**
    - **My SQL**

2) TOOLS:-

- **Sublime Text**
- **Visual Studio Code**
- **Draw.io**
- **Microsoft Power Point**
- **Microsoft Word**
- **SQL Server**

4. SYSTEM FLOW DIAGRAM

System flow diagram is a basically a graphical and sequential representation of the major steps involved in a systematic process.

❖ SYMBOLS OF SYSTEM FLOW DIAGRAM:-

- **Start/End point**

An oval represents a start or end point
The terminator symbol marks the starting or ending point of the system. It usually contains the word “Start” or “End”.

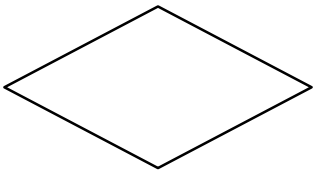
- **Arrows**

A line is a connector that shows relationships between the representative shapes.

- **Process**

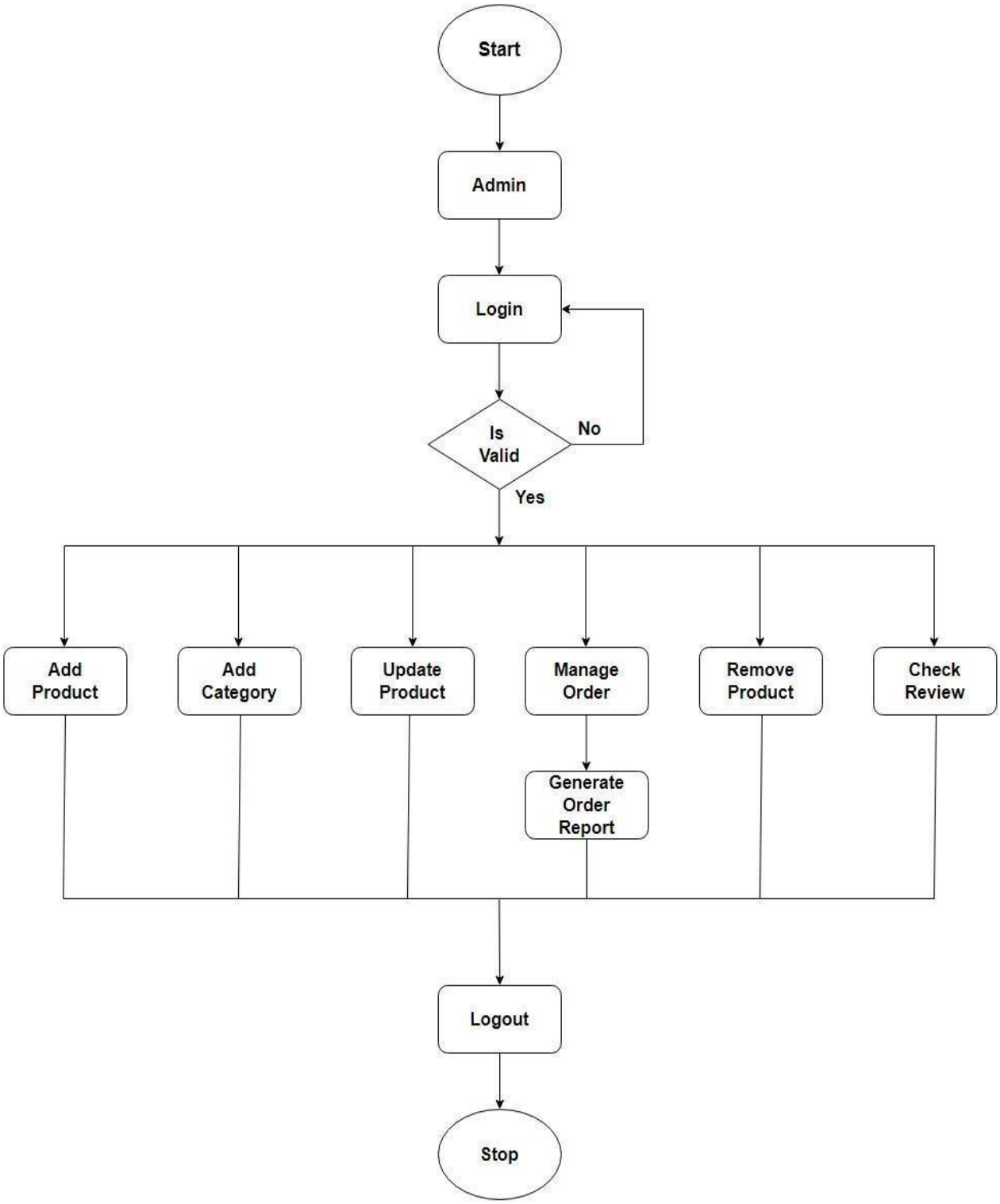
A rectangle represents a process.

- **Decision**

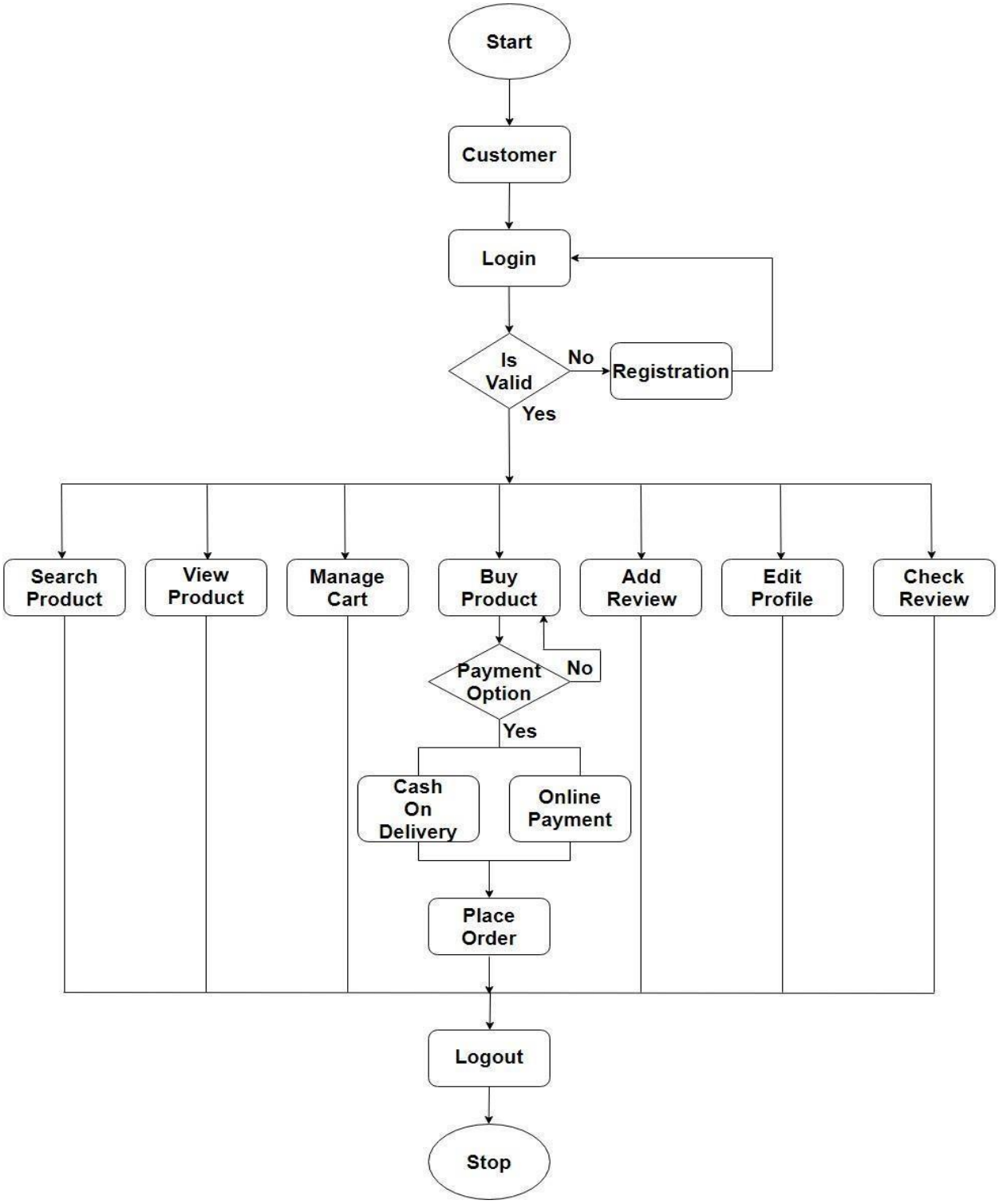


A diamond indicates a decision. It appoints where the outcome of a decision dictates the next step. There can be multiple outcomes, but often there are just two – yes and no or true – false, then branches to different parts of the system flow diagram.

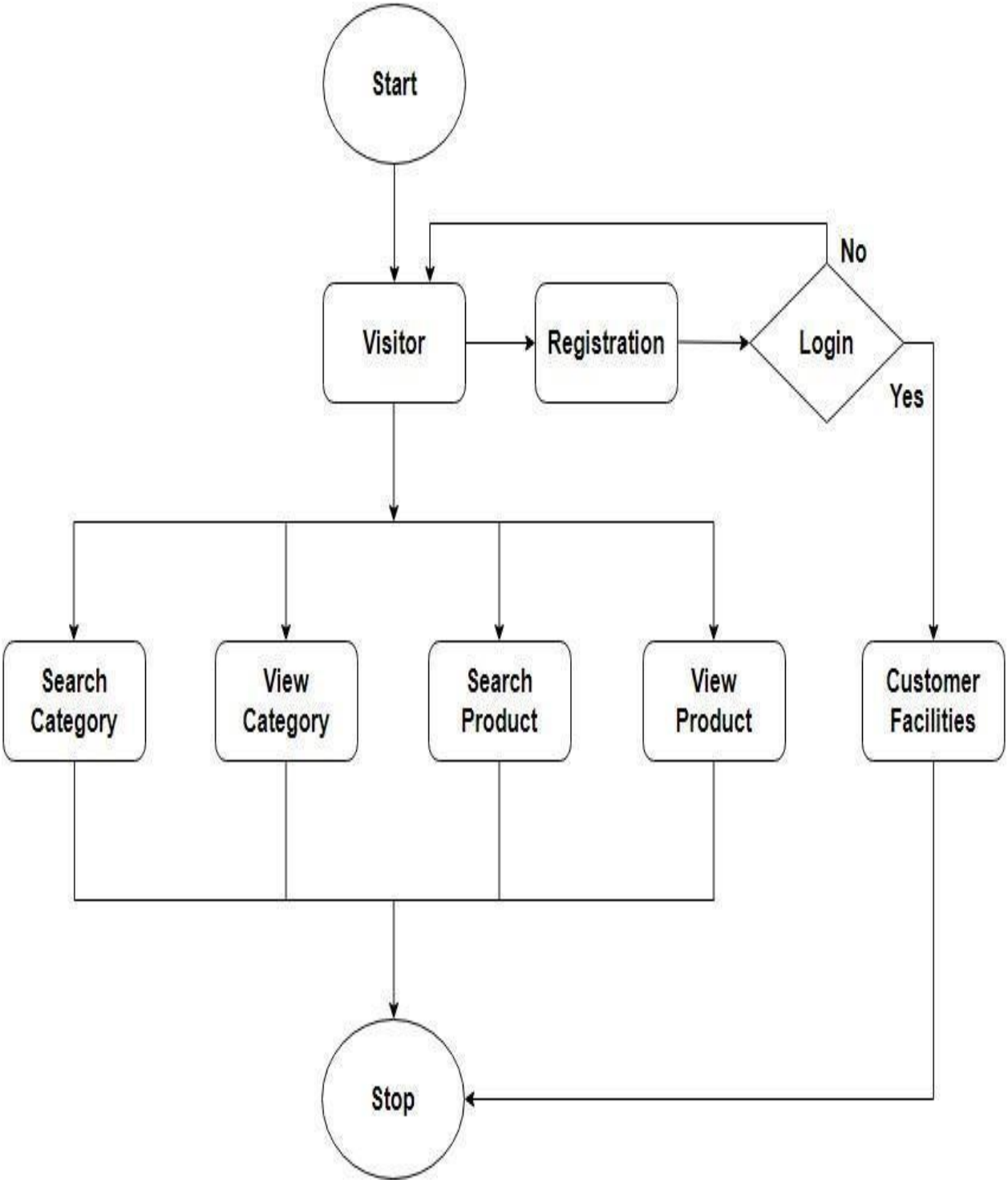
❖ ADMIN SYSTEM FLOW DIAGRAM:-



❖ CUSTOMER SYSTEM FLOW DIAGRAM:-



❖ VISITOR SYSTEM FLOW DIAGRAM:-



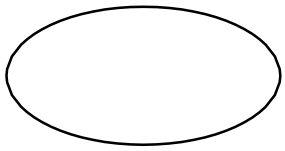
5. UML DIAGRAM

USE CASE DIAGRAM

- Use case is a set of scenarios that describing an interaction between a user and a sys-tem.
- Use case diagram displays the relationship among actors and use cases.
- The two main components of a use case diagram are use cases and actors.
- An actor represents user or another system that will interact with the system.
- Use case is an external view of the system that represents s some actions the user mightperform in order to complete a task.

❖ SYMBOLS OF USE CASE DIAGRAM:-

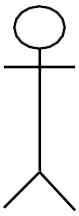
- **Use Case**



Use Case

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

- **Actors**



Actors are the users of a system. When one system is the actor of another system label theactor system with the actor stereotype.

- **System**

Draw your system's boundaries using a rectangle that contains use cases, Placeactor outside the system's boundaries.

- **Relationships**

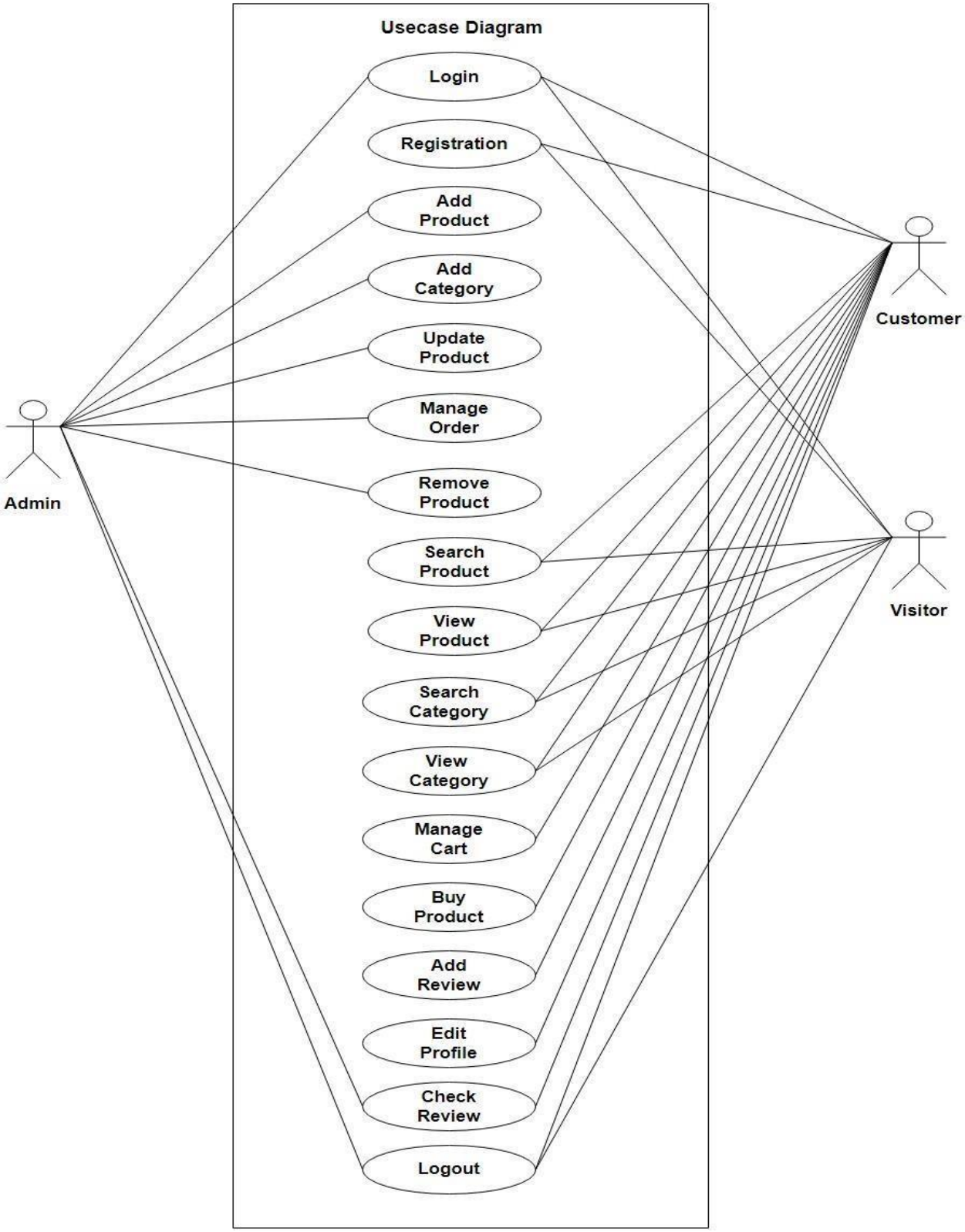
<<include>>

<<extend>>

Illustrate relationships between an actor and use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends". A "uses" relationship indicates that one use case is needed by another in order To perform a task.

An "extends" relationship indicates alternative options under a certain use Case.

❖ USE-CASE DIAGRAM:-



SEQUENCE DIAGRAMS

- A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next.
- Sequence diagrams establish the roles of objects and help provide essential information todetermine class responsibilities and interfaces.
- This type of diagram is best used during early analysis phases in design because they are simple and easy to comprehend. Sequence diagrams are normally associated with use cases.

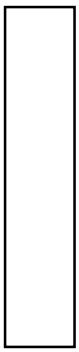
❖ SYMBOLS OF SEQUENCE DIAGRAM:-

- **Class roles and participants**

Component

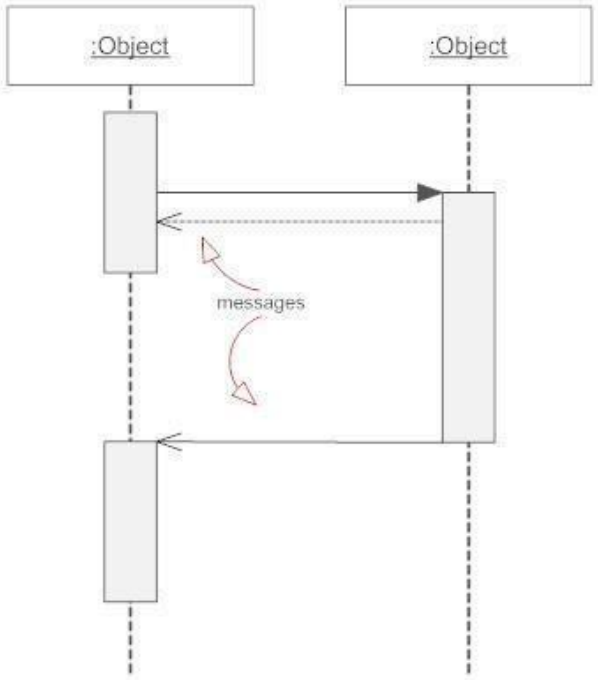
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

- **Activation or Execution Occurrence**



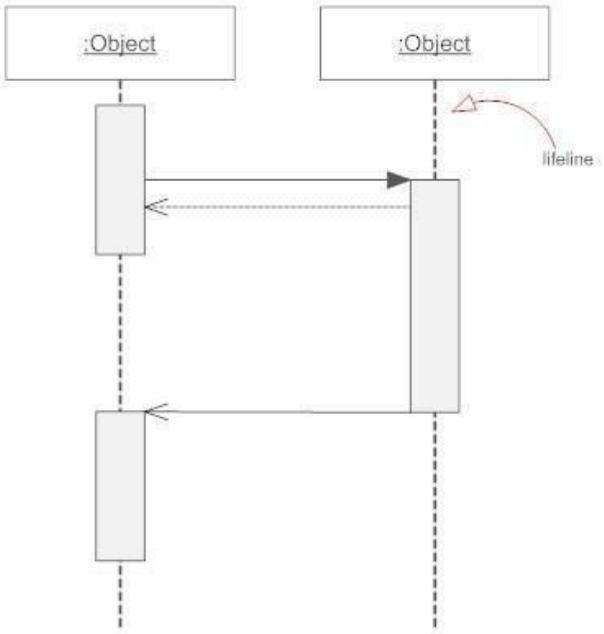
Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message,use a thin gray rectangle placed vertically on its lifeline.

- **Messages**



Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

- **Lifelines**



Lifelines are vertical dashed lines that indicate the object's presence over Time

Types of Messages in Sequence Diagram

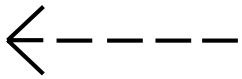
- **Synchronous Message**

A synchronous message requires a response before the interaction can continue.
It's usually drawn using a line with a solid arrowhead pointing from one object to another.

- **Asynchronous Message**

An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sanding other messages to other objects within the system.
The arrowhead used to show this type of message is a line arrow.

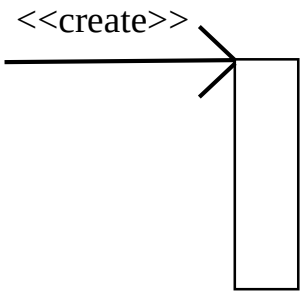
- **Reply or Return Message**



A reply messages is drawn with a dotted line and an open arrowhead pointing back to the original lifetime.
A message an object sends to itself, usually shown as a shaped arrow pointing back to itself.
This is a message that creates a new object.

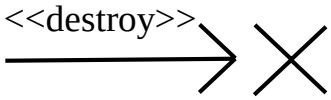
.

- **Create Message**



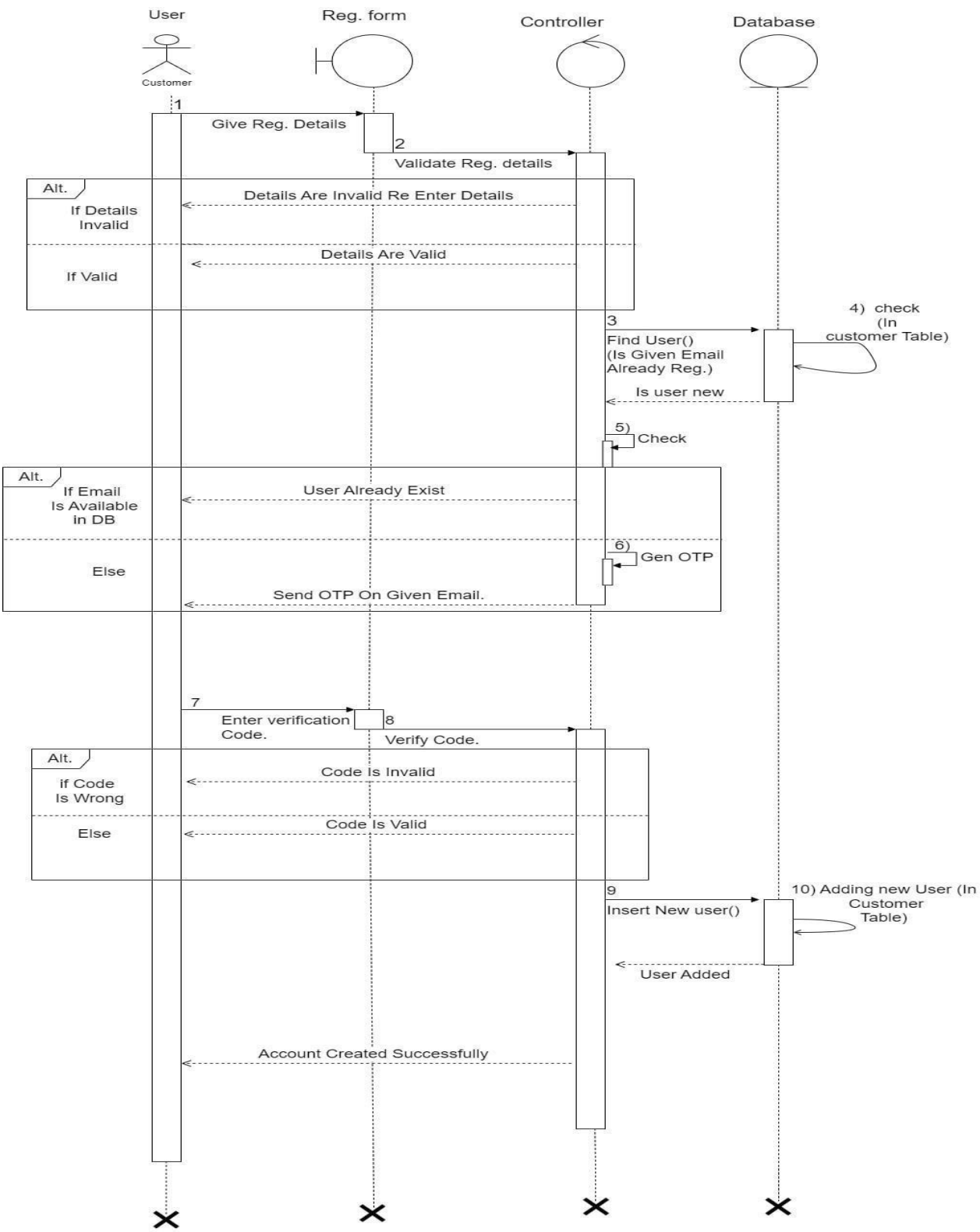
A message represent an interaction between objects, or between an object and the environment.
A message can be an event, a triggered operation, or a primitive operation.
In the meta-model, a message defines a specific type of communication.

- **Delete Message**

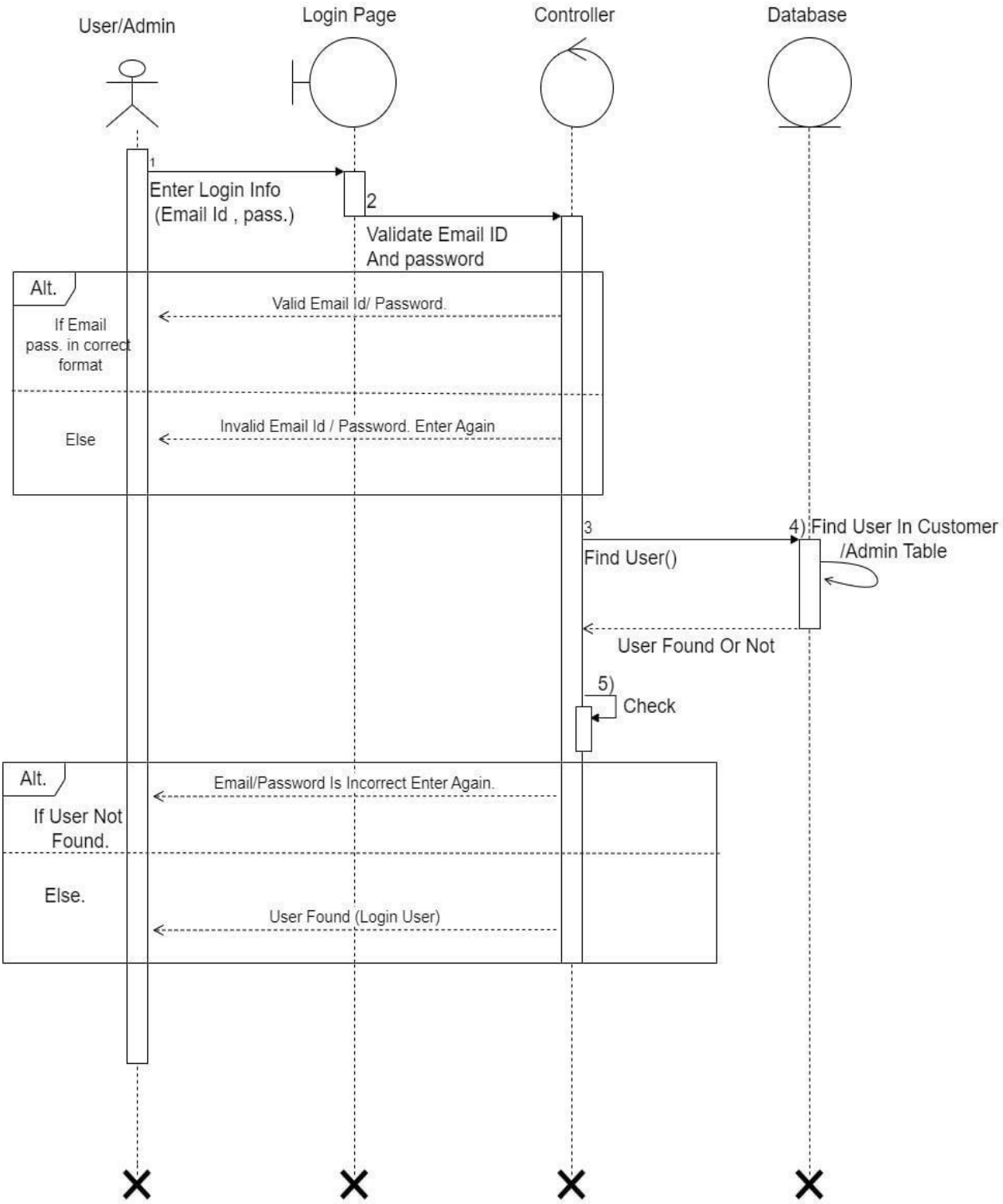


If you delete a create message, the lifeline extends vertically to the top of the interaction frame.
This is a message that destroys an object.
It can be shown by an arrow with an x at the end.

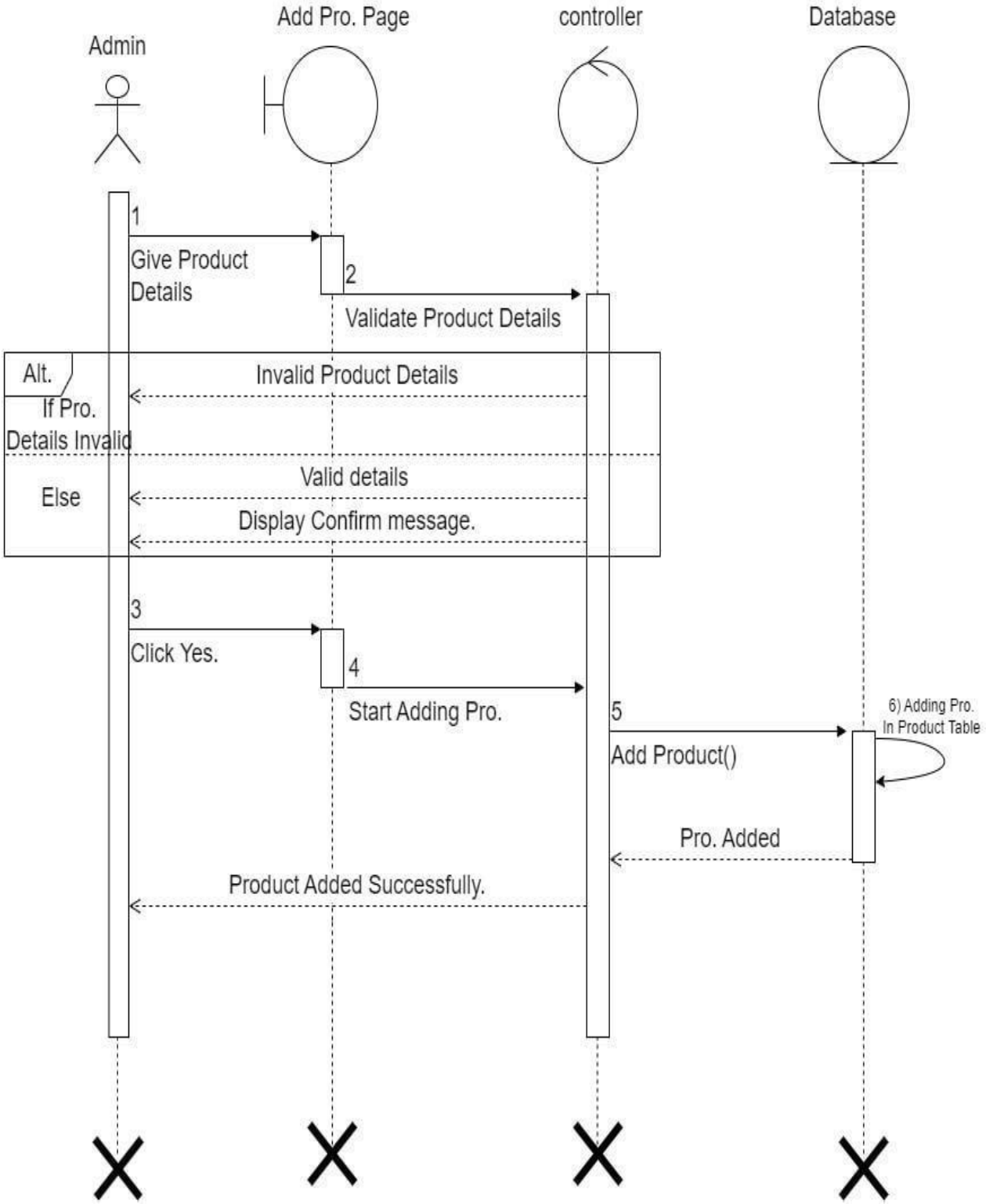
❖ REGISTRATION SEQUENCE DIAGRAM:-



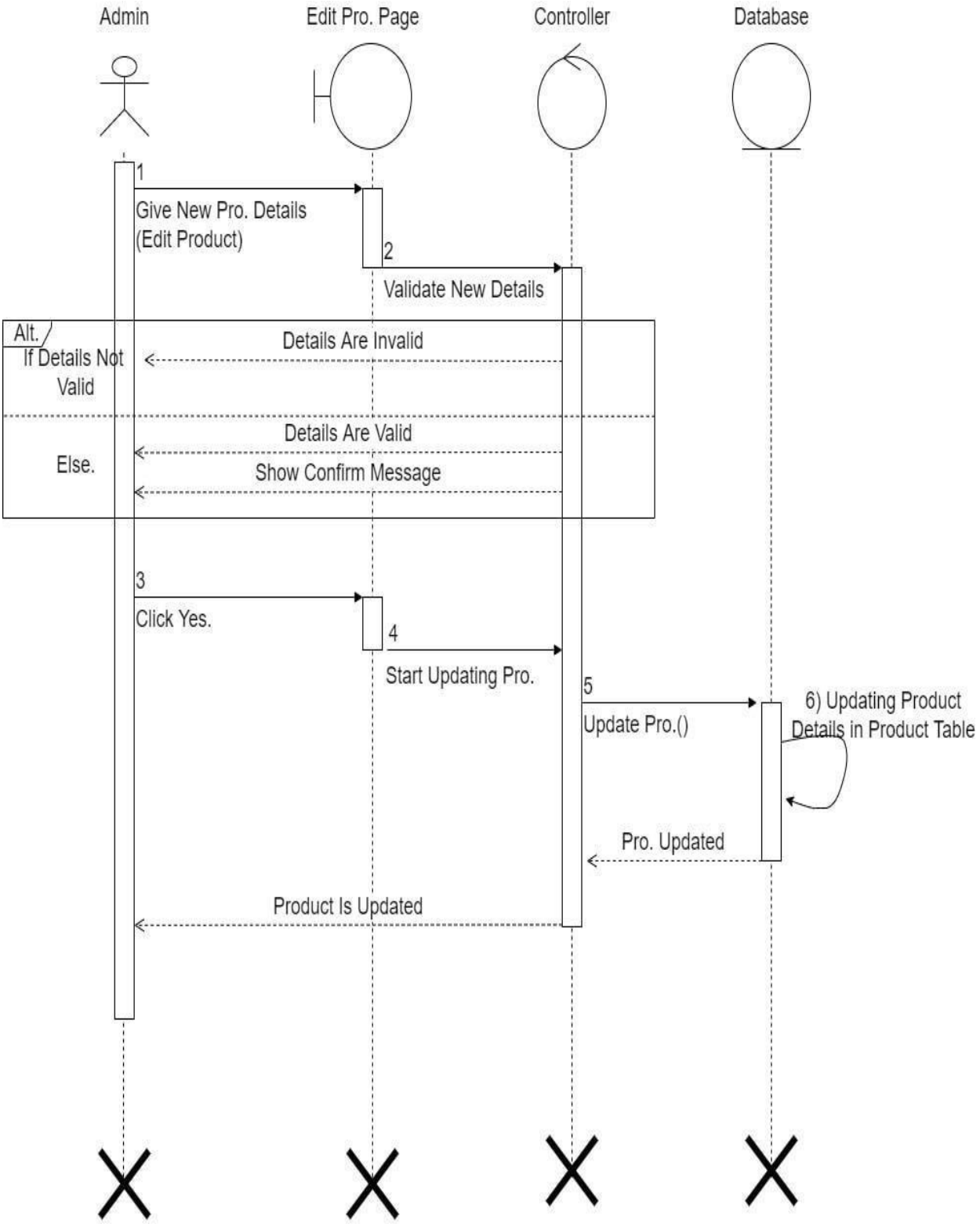
❖ LOGIN SEQUENCE DIAGRAM:-



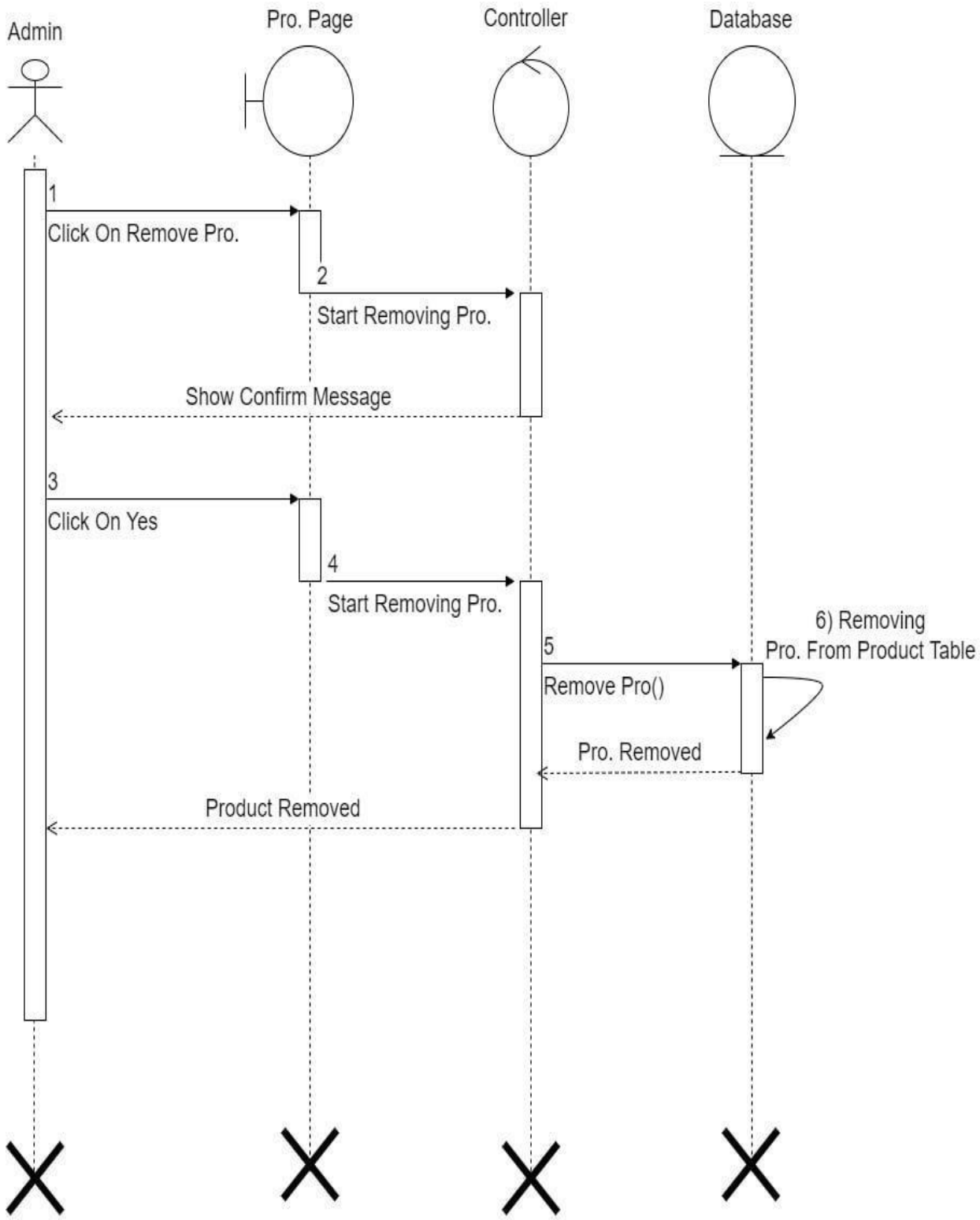
❖ ADMIN ADD PRODUCT SEQUENCE DIAGRAM:-



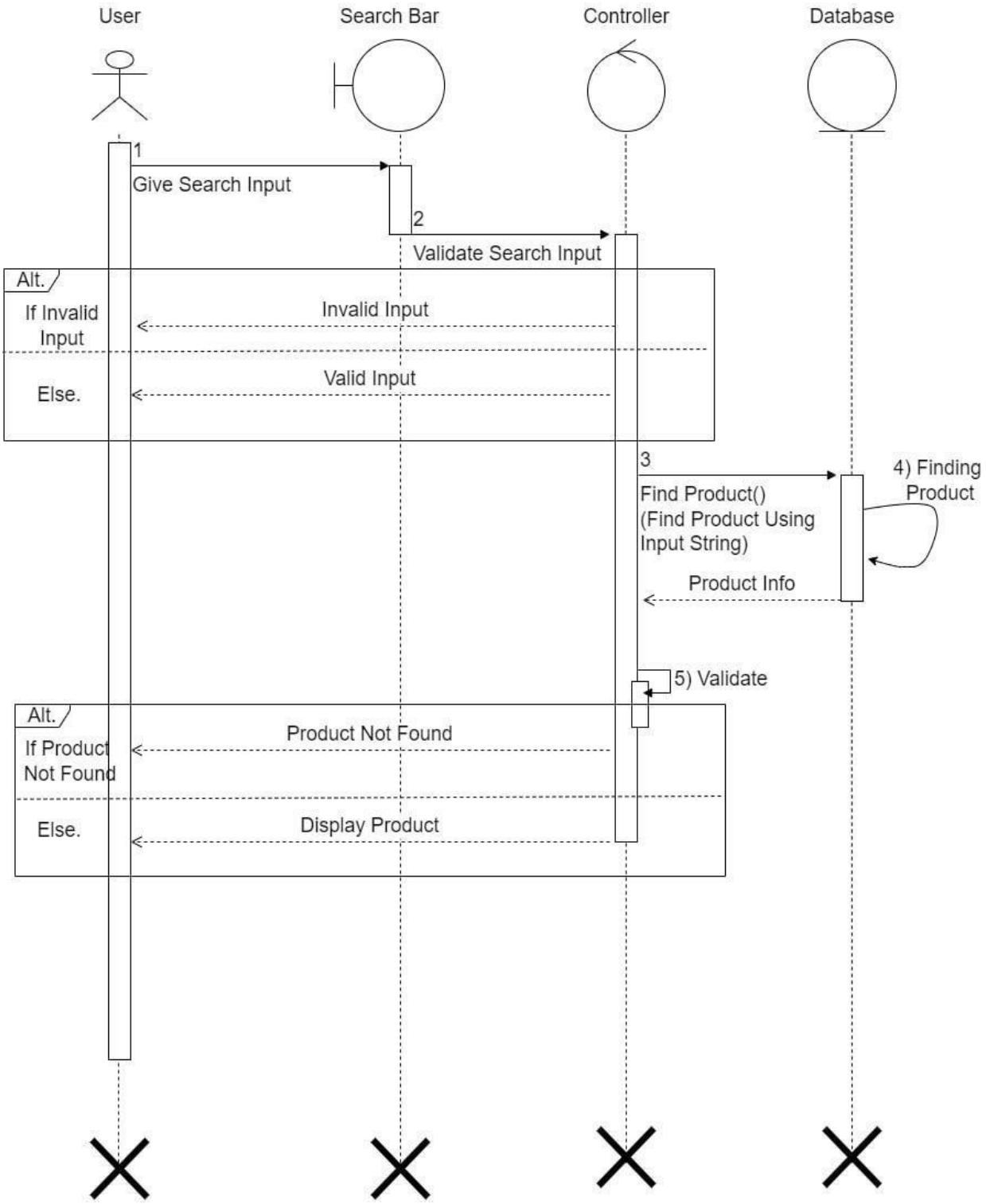
❖ ADMIN UPDATE PRODUCT SEQUENCE DIAGRAM:-



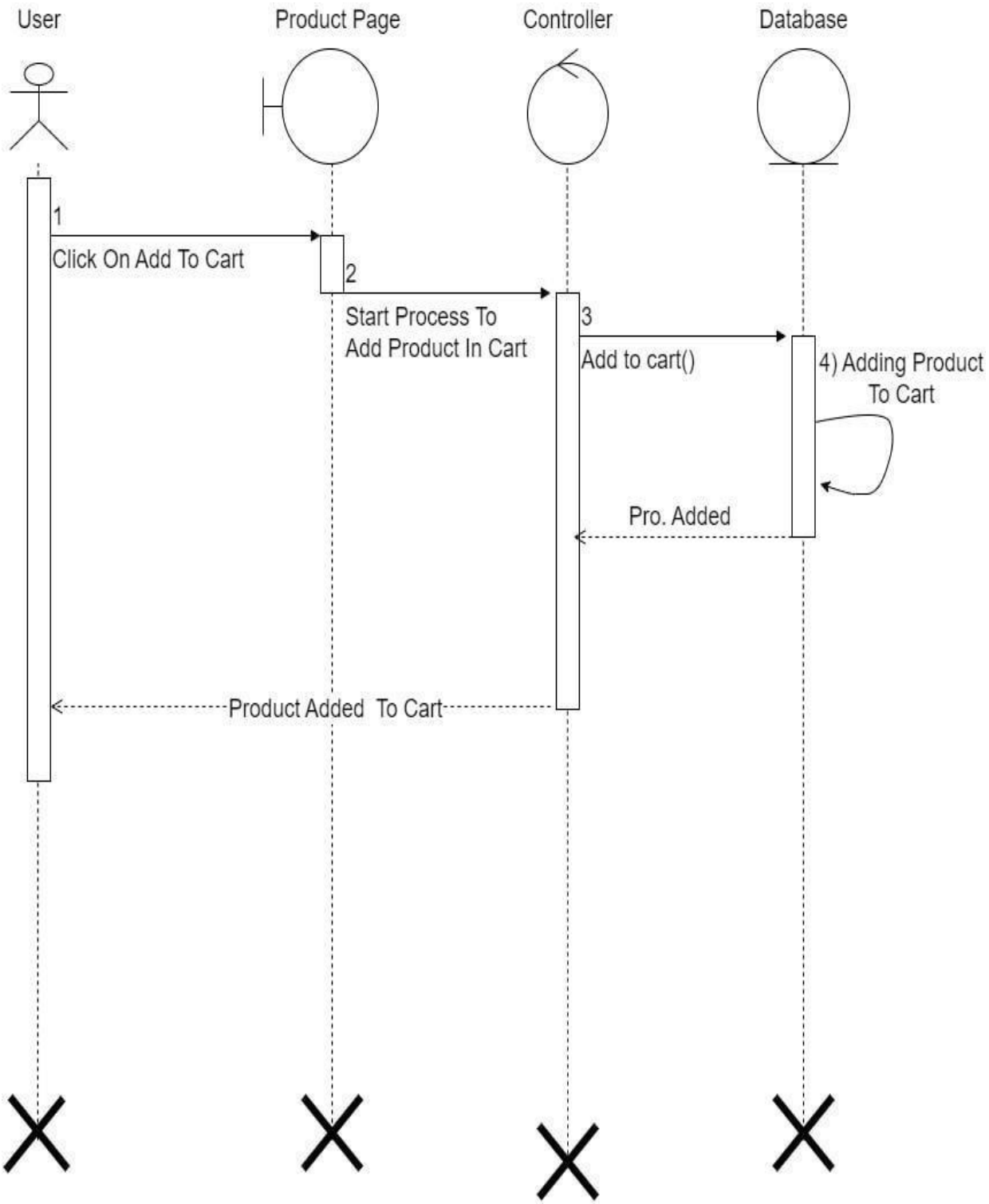
❖ ADMIN REMOVE PRODUCT SEQUENCE DIAGRAM:-



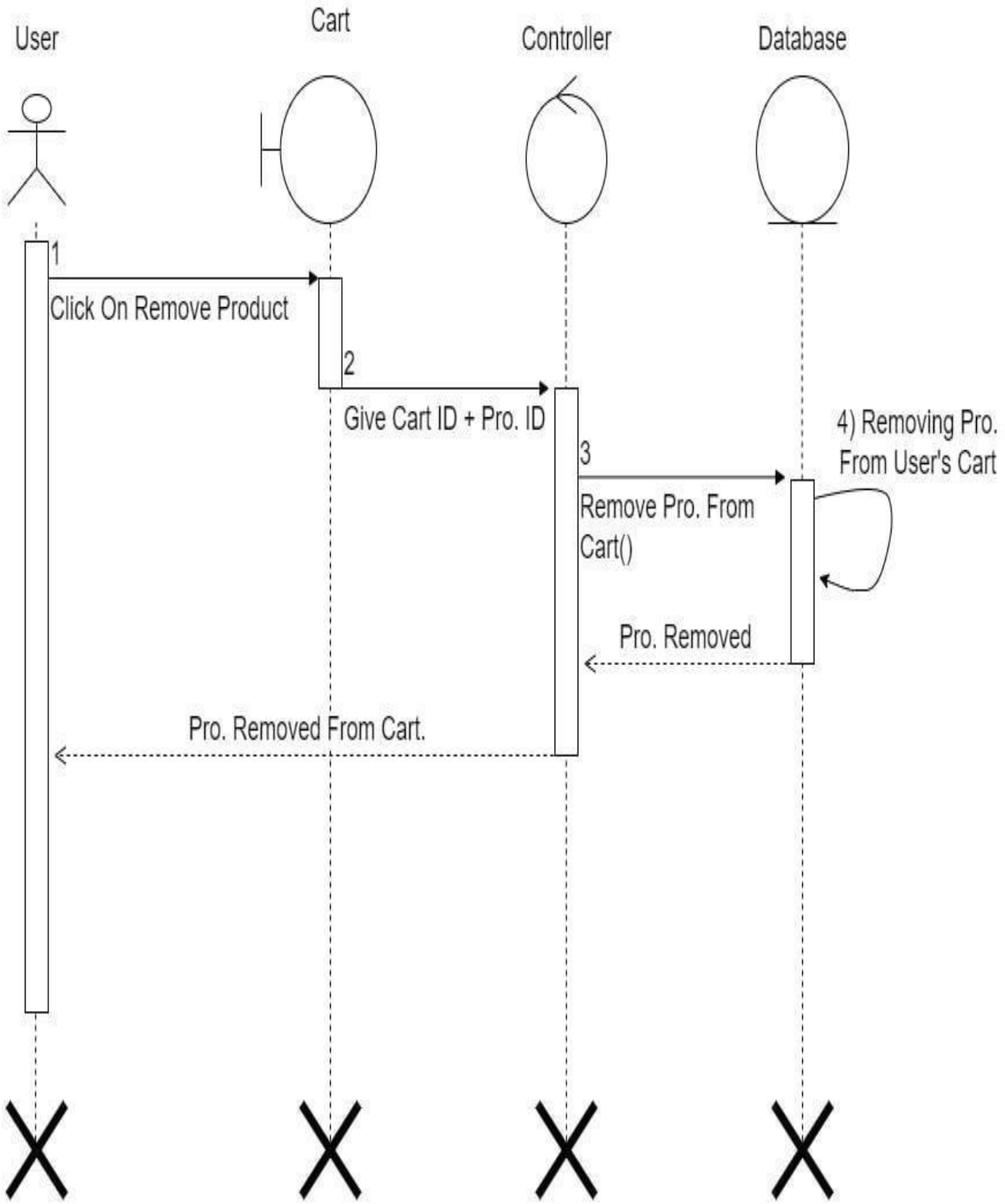
❖ SEARCH PRODUCT SEQUENCE DIAGRAM:-



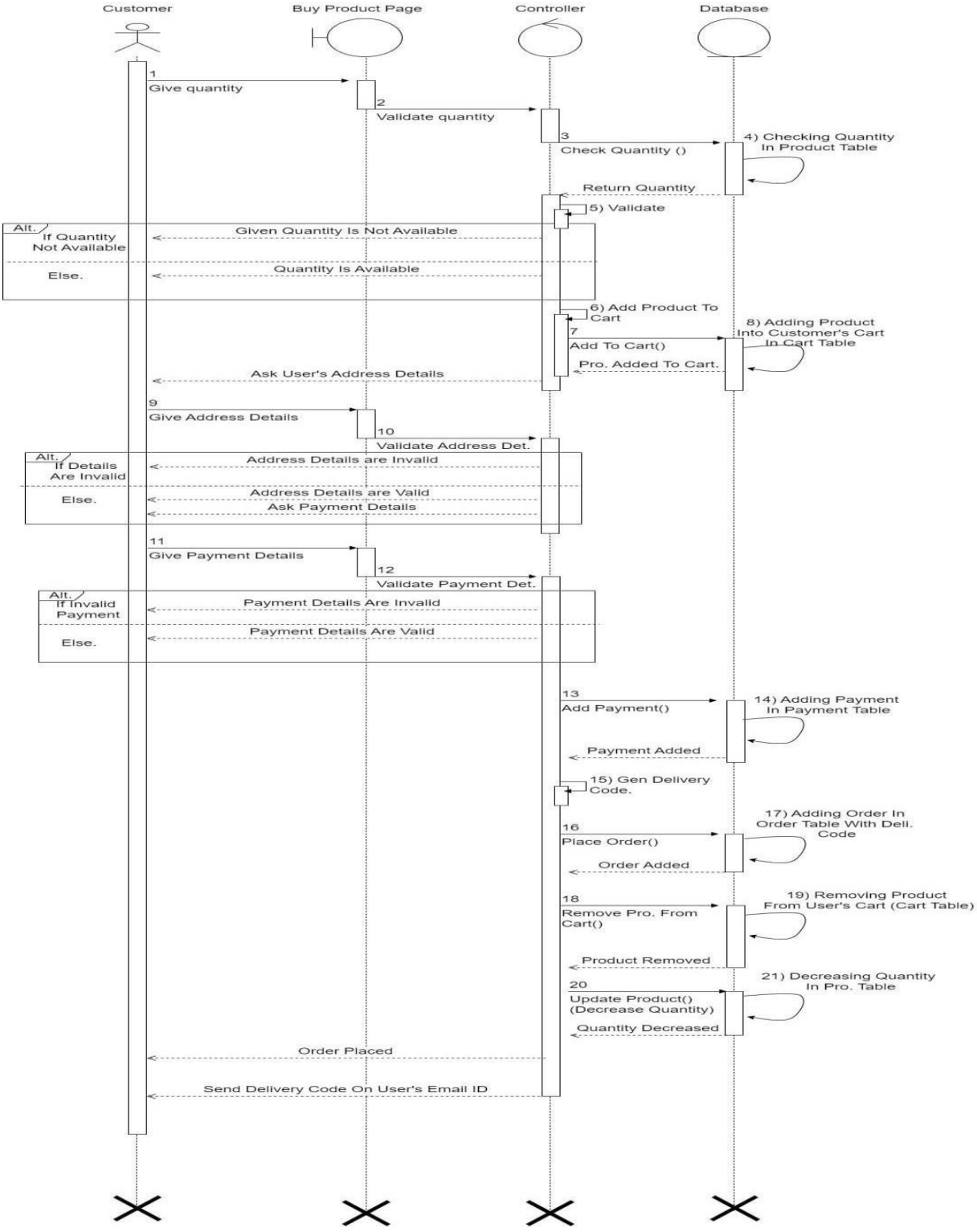
❖ ADD PRODUCT TO CART SEQUENCE DIAGRAM:-



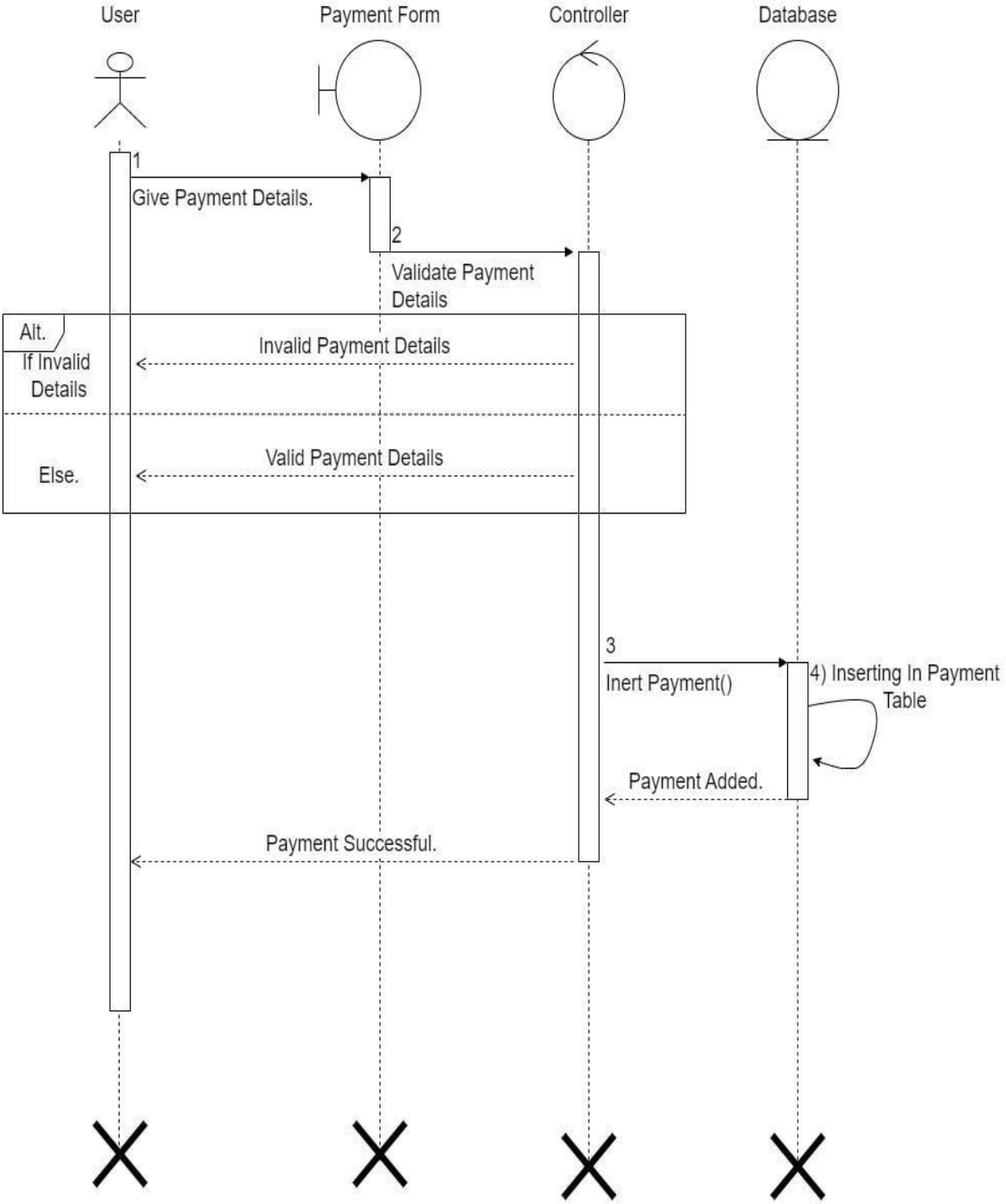
❖ REMOVE PRODUCT FROM CART SEQUENCE DIAGRAM:-



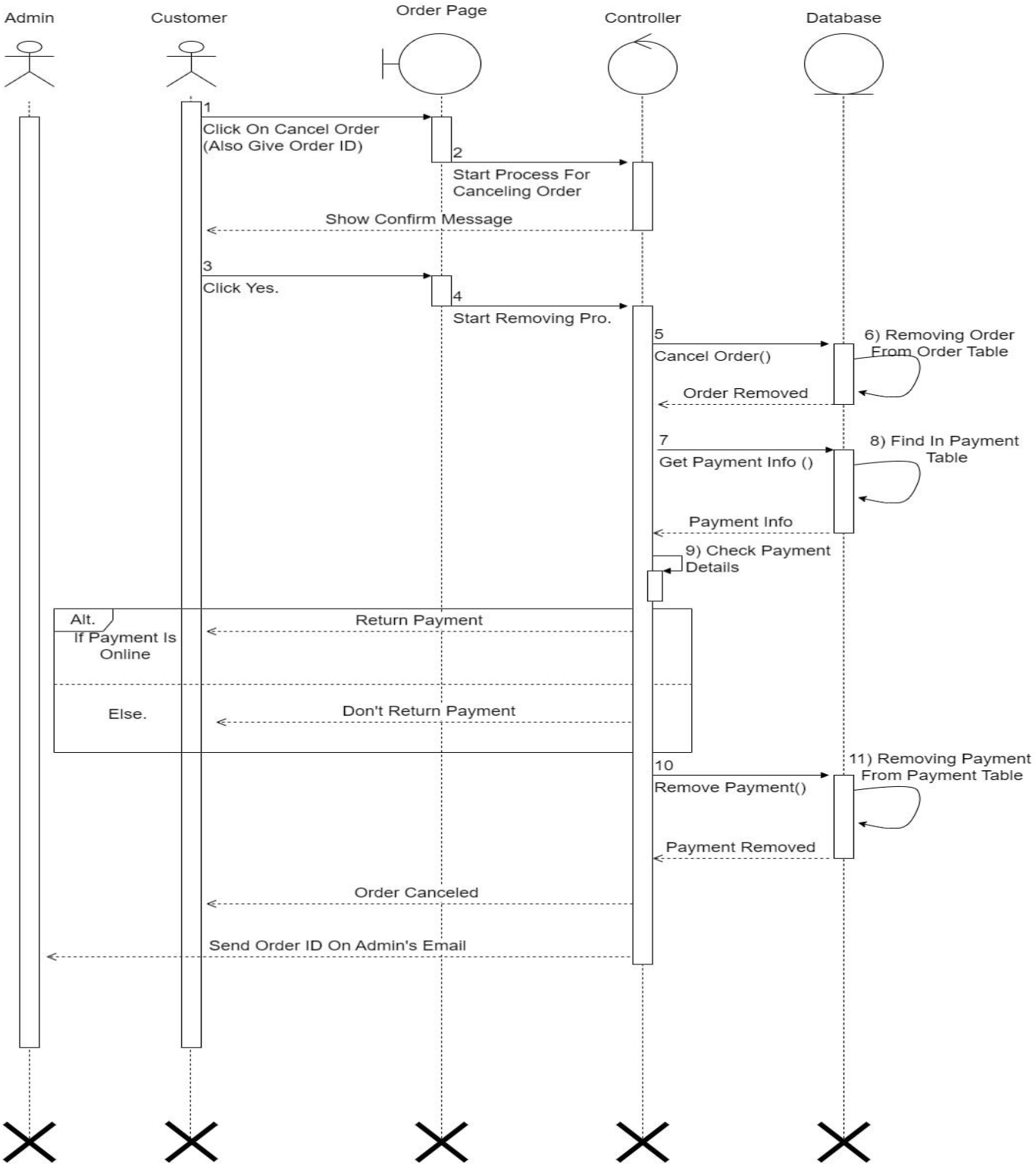
❖ BUY PRODUCT SEQUENCE DIAGRAM:-



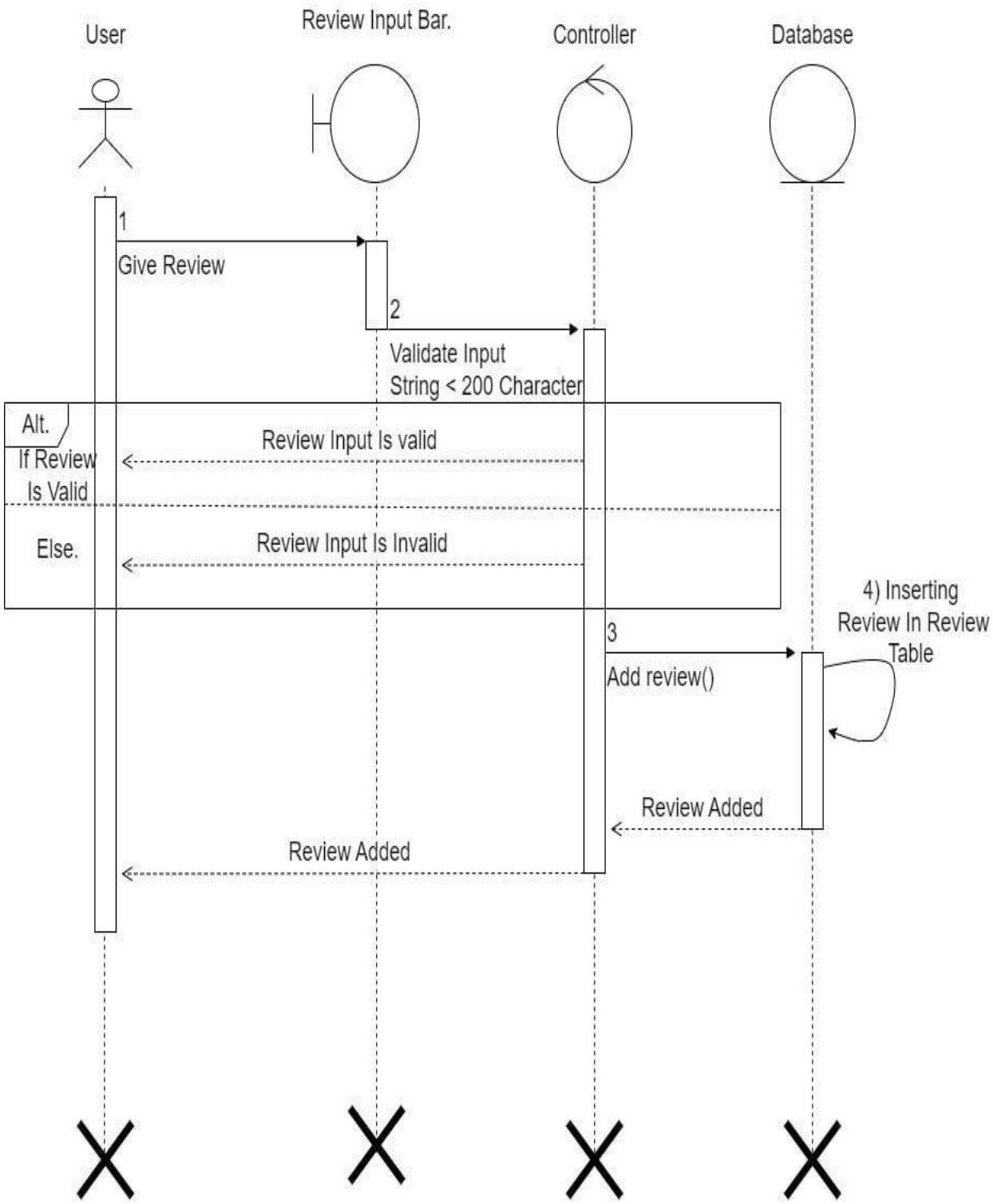
❖ PAYMENT SEQUENCE DIAGRAM:-



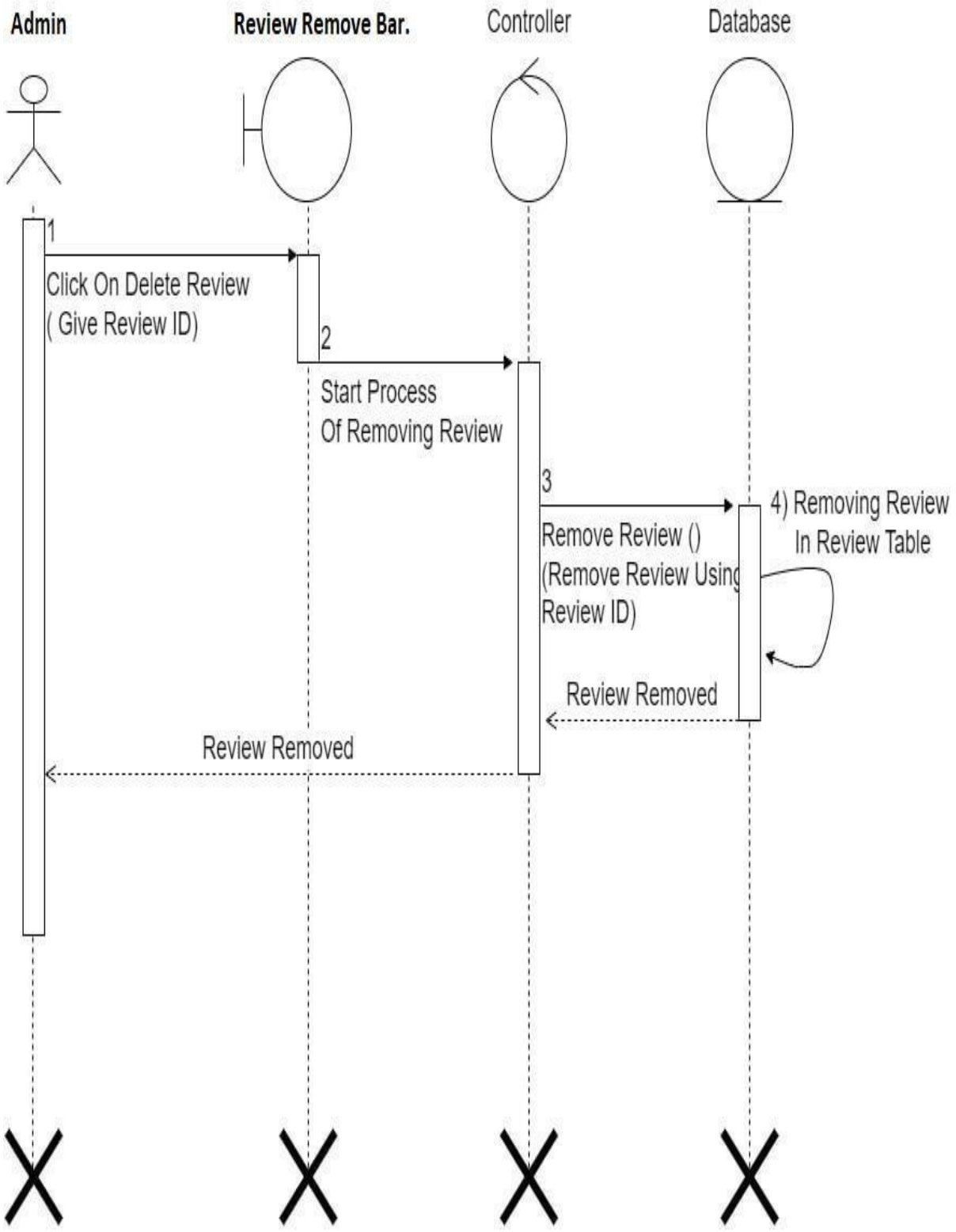
❖ CANCEL ORDER SEQUENCE DIAGRAM:-



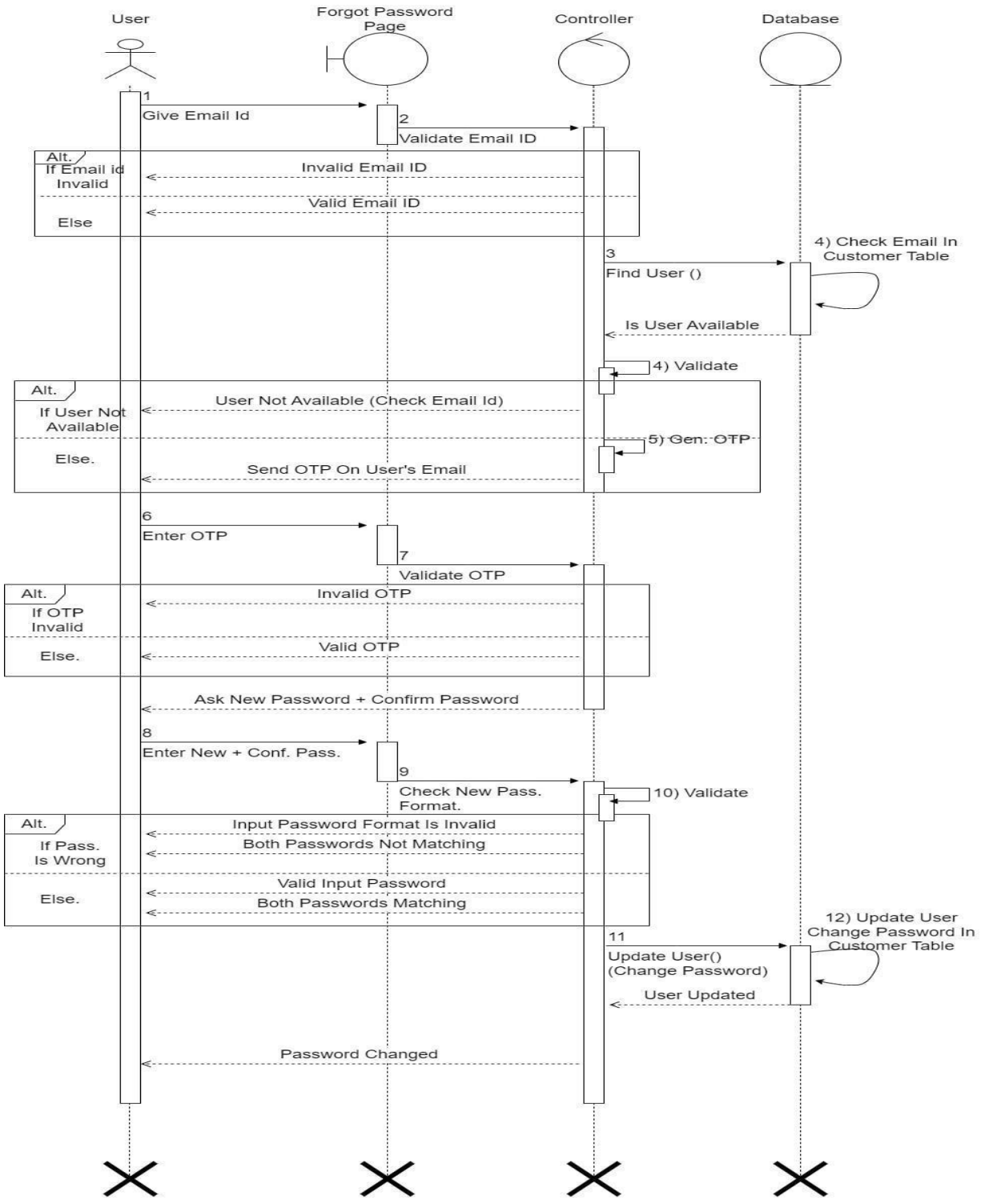
❖ ADD REVIEW SEQUENCE DIAGRAM:-



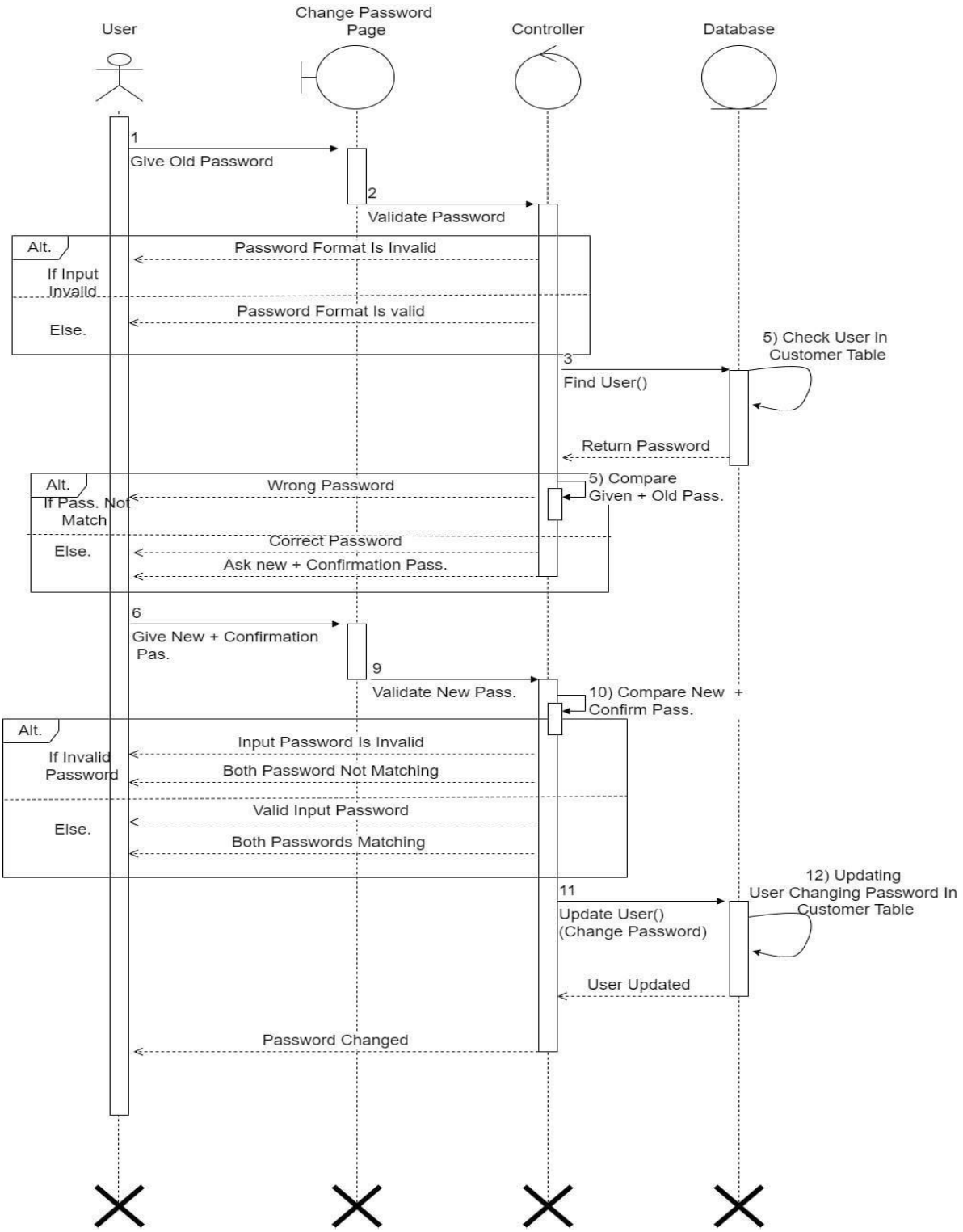
❖ REMOVE REVIEW SEQUENCE DIAGRAM:-



❖ FORGOT PASSWORD SEQUENCE DIAGRAM:-



❖ CHANGE PASSWORD SEQUENCE DIAGRAM:-



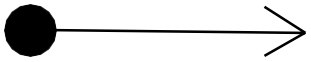
ACTIVITY DIAGRAMS

- Activity diagrams are graphical representation of step wise activities and actions withSupport for choice, iteration and concurrency.
- Activity diagram can be used to describe the business and operational step – by – stepWorkflows of components in a system.
- Activity diagrams are typically used for business process modeling, for modeling thelogic captured by a single usage scenario, or for modeling the detailed logic.
- Although UML activity diagrams could potentially model the internal logic of a com-plex operation it would be far better to simply rewrite the operation so that it is simpleenough that you don't require an activity diagram.

❖ SYMBOLS OF ACTIVITY DIAGRAM

- **Initial State or Start Point**

  A small filled circle followed by an arrow represents the initial action state or the start pointfor any activity diagram. For activity diagram using swim lanes, make sure the start point isplaced in the top left corner of the first column.

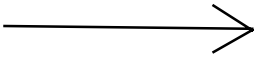


- **Activity or Action State**

  An action state represents the non-interruptible action of objects. You can draw an actionstate in Smart Draw using a rectangle with rounded corners.

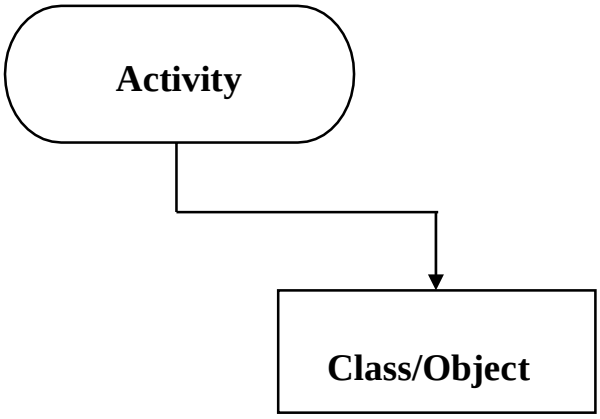
- **Action Flow**

  Action flows, also called edges and paths, illustrate the transitions from one action state toanother. They are usually drawn with an arrowed line.



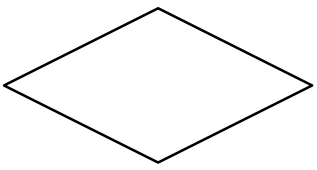
- **Object Flow**

  Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. Anobject flow arrow from an object to an action indicates that the action state uses the object.



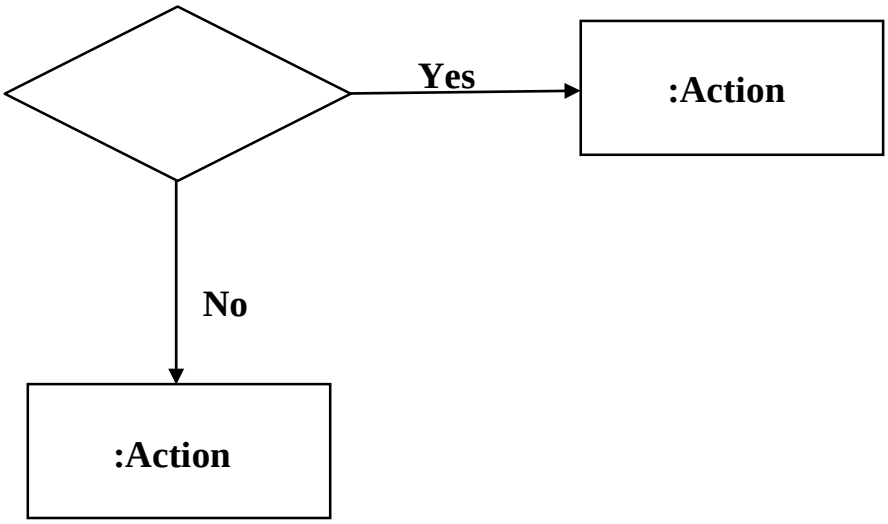
- **Decisions and Branching**

  A diamond represents a decision with alternate paths. When an activity requires a decisionprior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



- **Guards**

In UML, guards are a statement written next to a decision diamond that must be true beforemoving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.

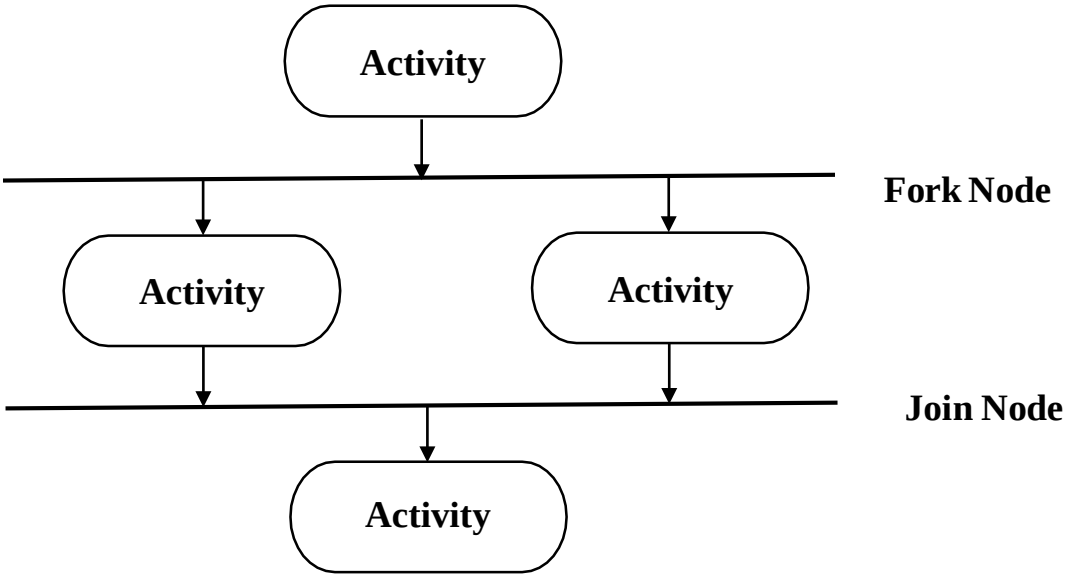


- **Synchronization**

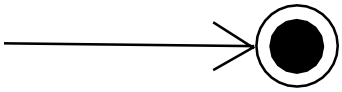
A fork node is used to split a single incoming flow into multiple concurrent flows. It isrepresented as a straight, slightly thicker line in an activity diagram.

A join node joins multiple concurrent flows back into a single outgoing flow.

A fork and join mode used together are often referred to as synchronization.

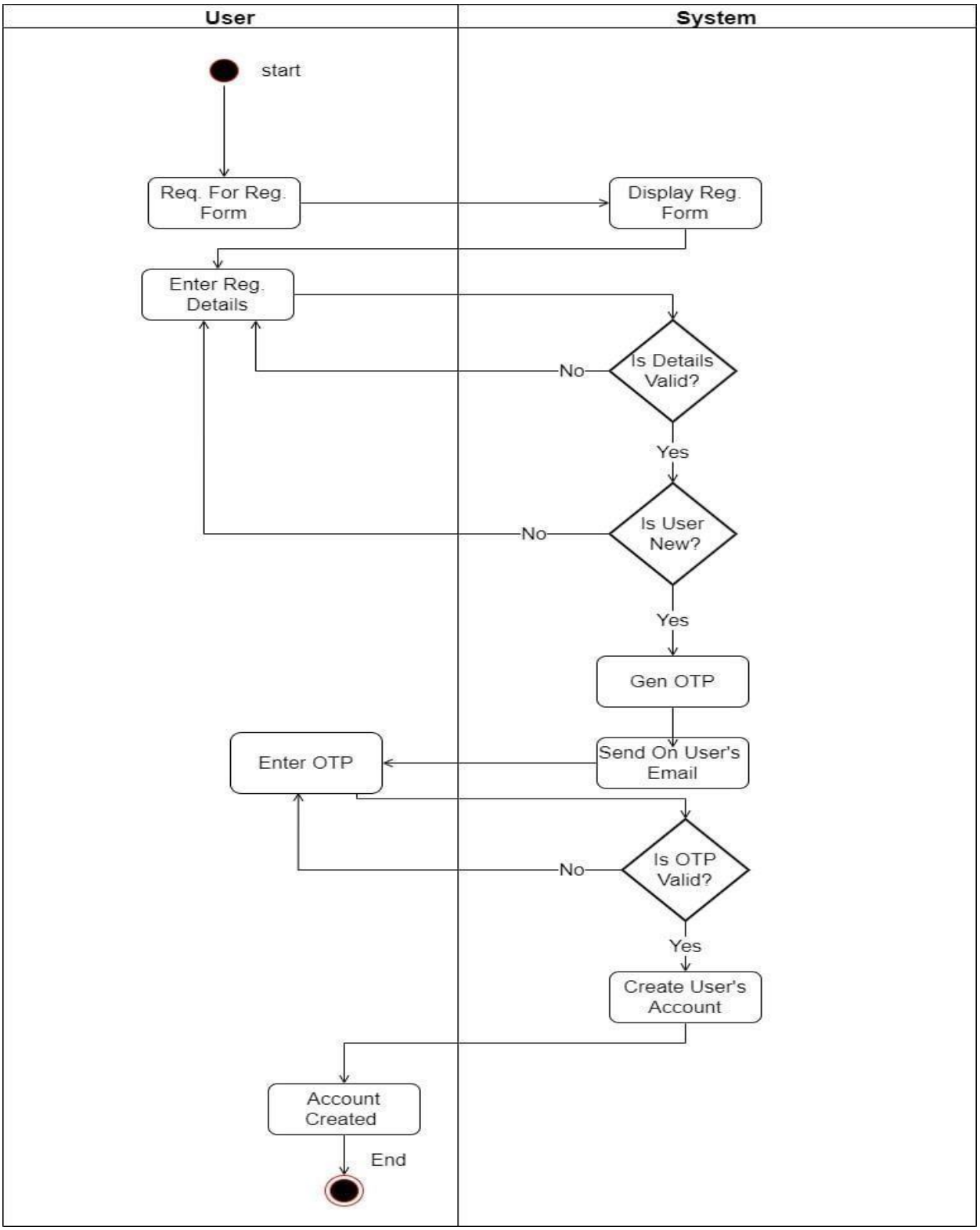


- **Final State or End Point**

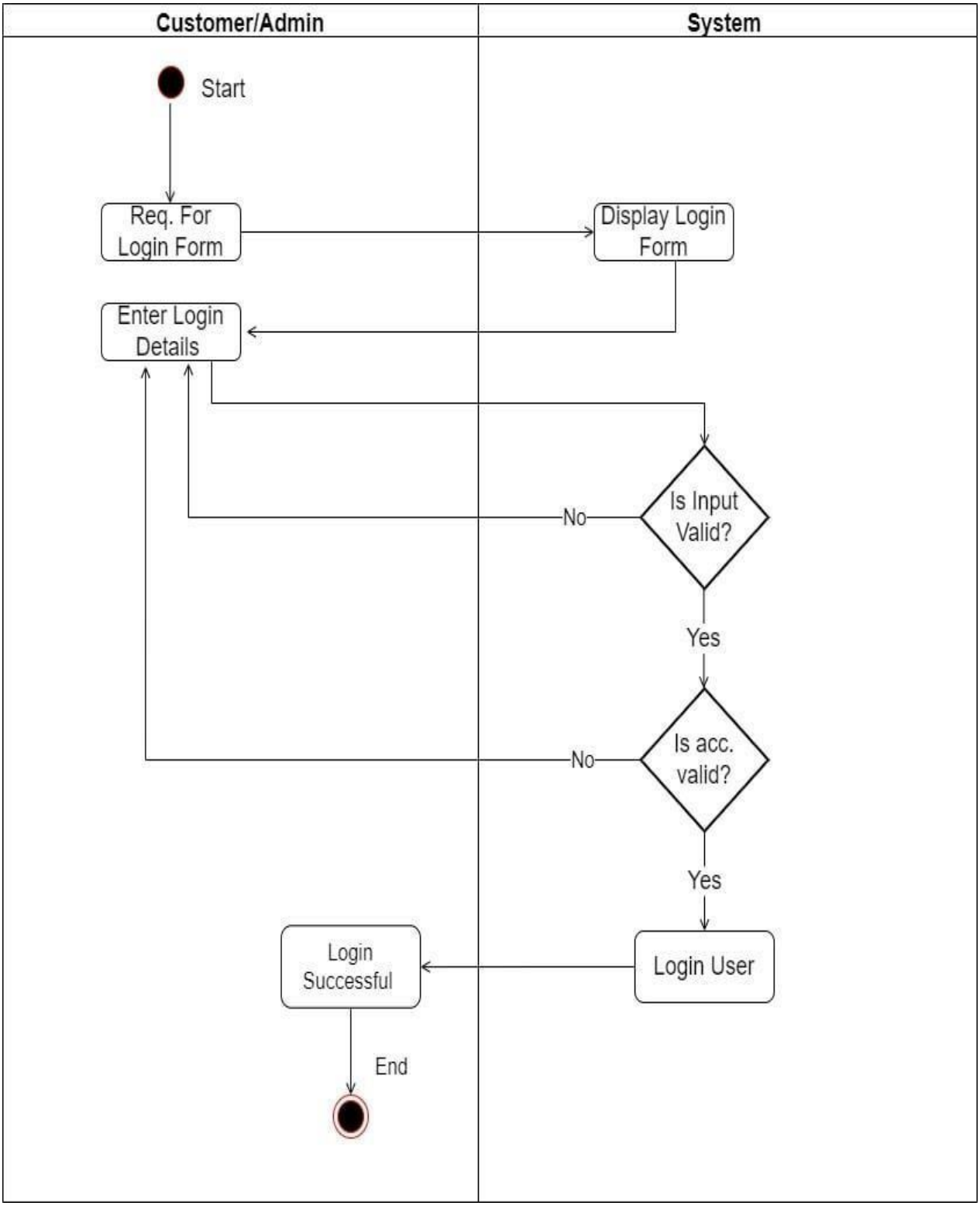


An arrow pointing to a filled circle nested inside another circle represents the final actionstate.

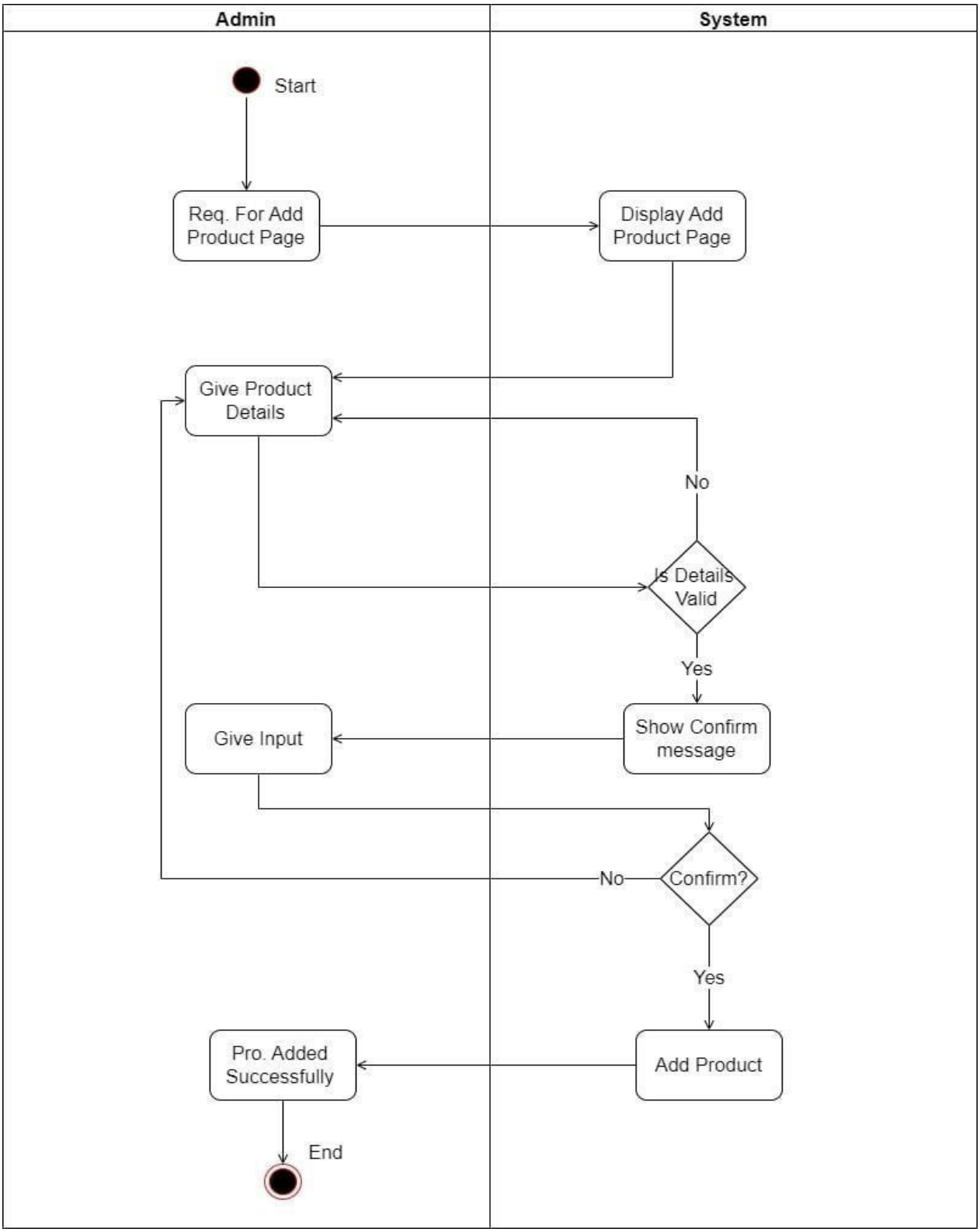
❖ REGISTRATION ACTIVITY DIAGRAM:-



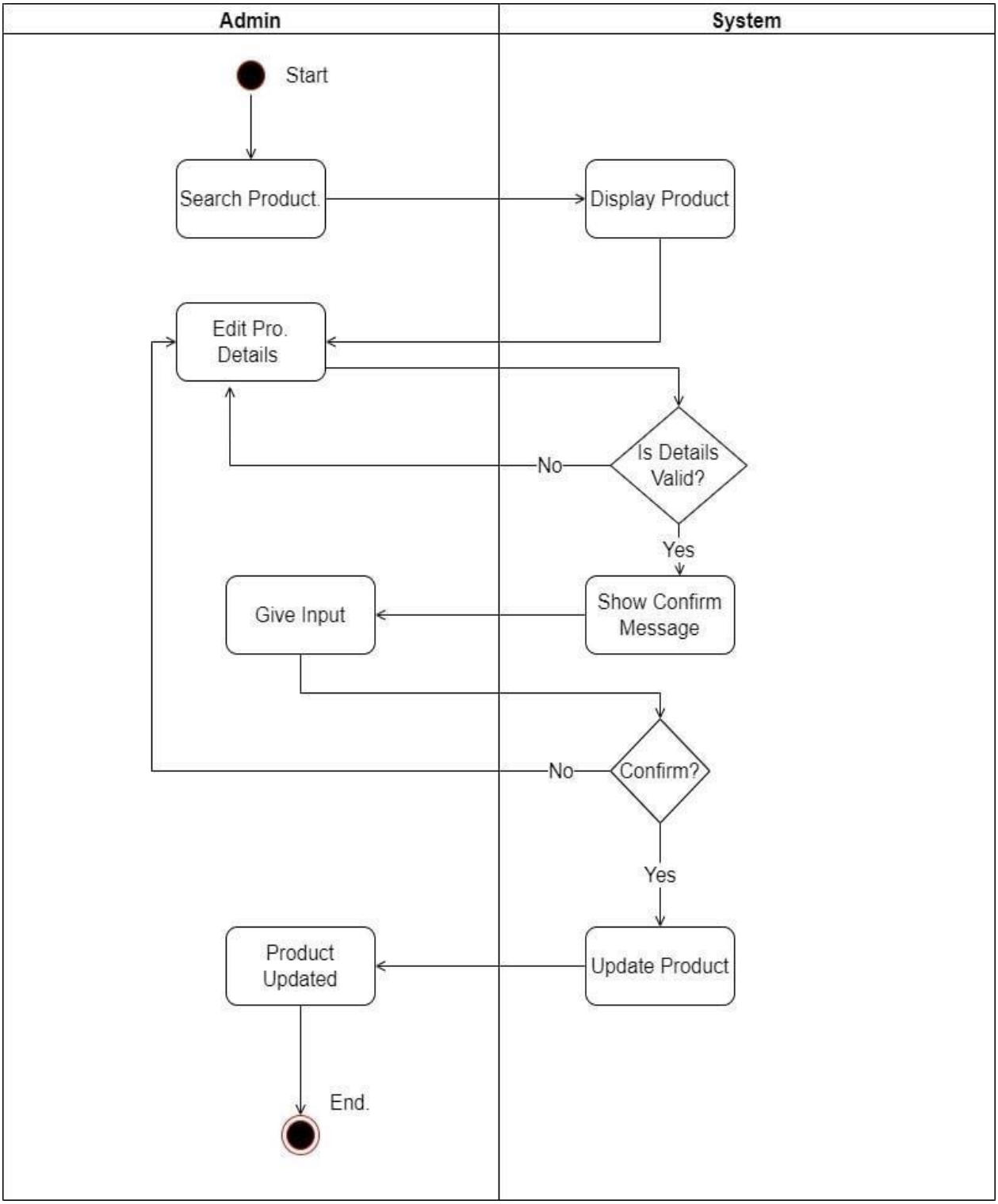
❖ LOGIN ACTIVITY DIAGRAM:-



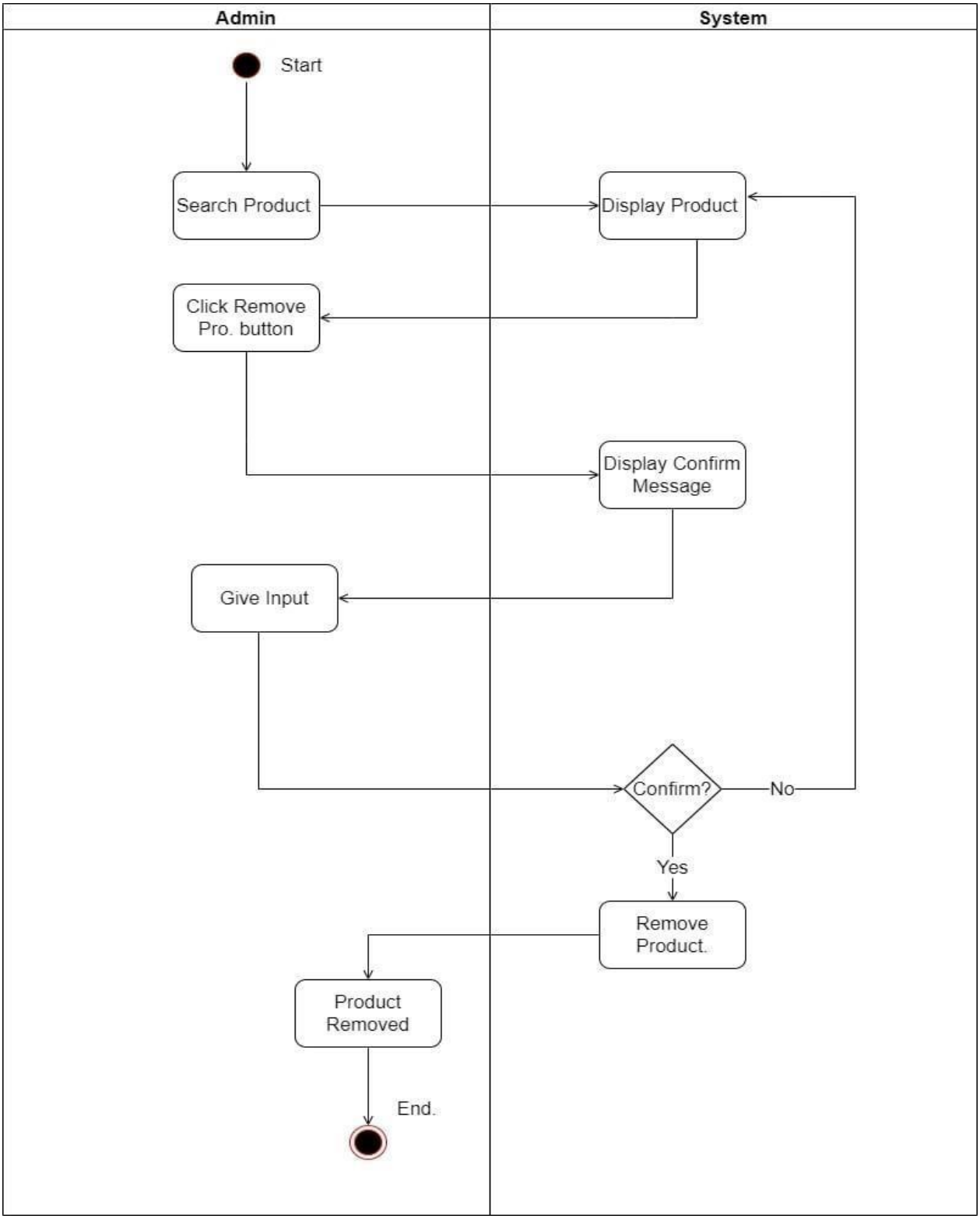
❖ ADMIN ADD PRODUCT ACTIVITY DIAGRAM:-



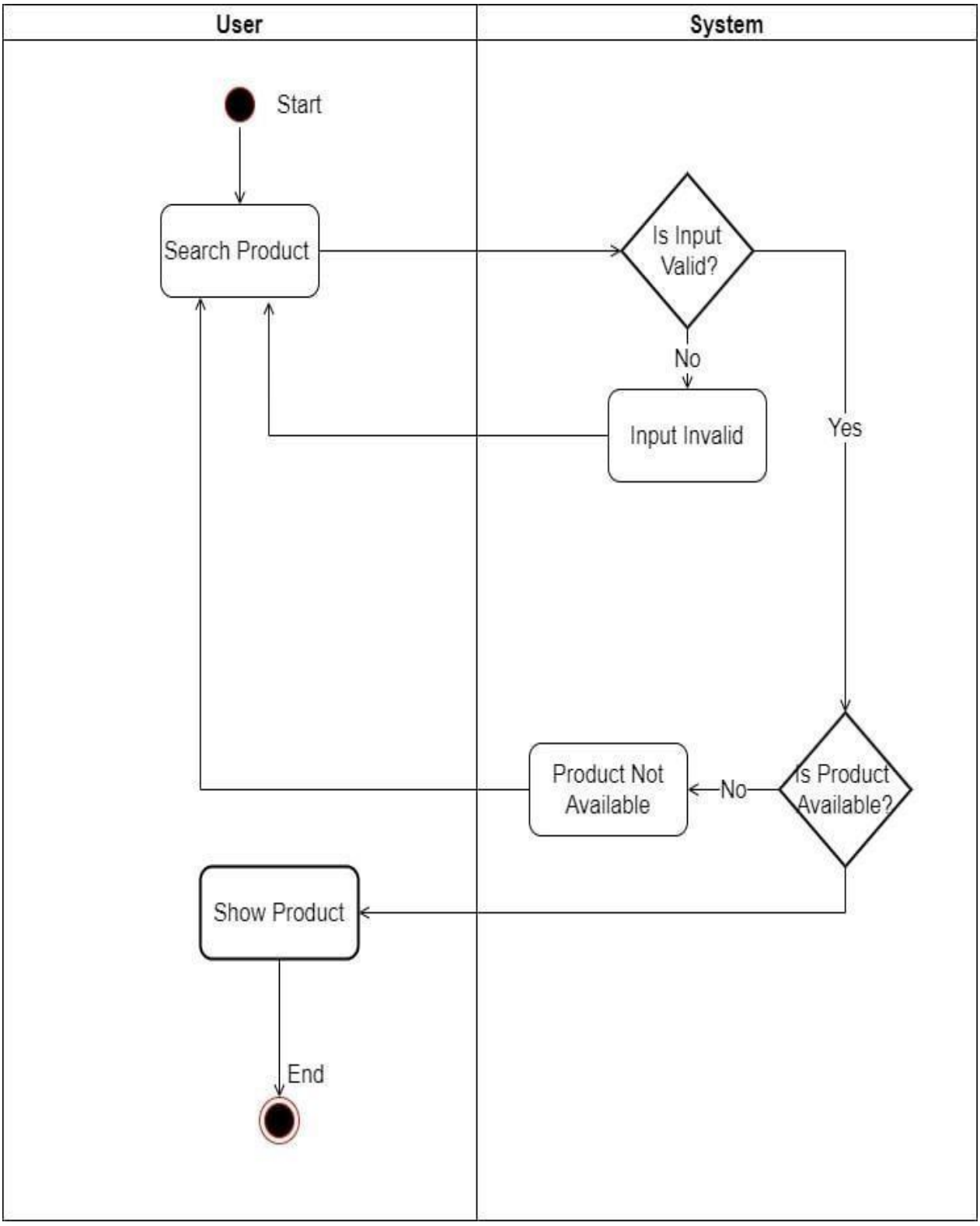
❖ ADMIN UPDATE PRODUCT ACTIVITY DIAGRAM:-



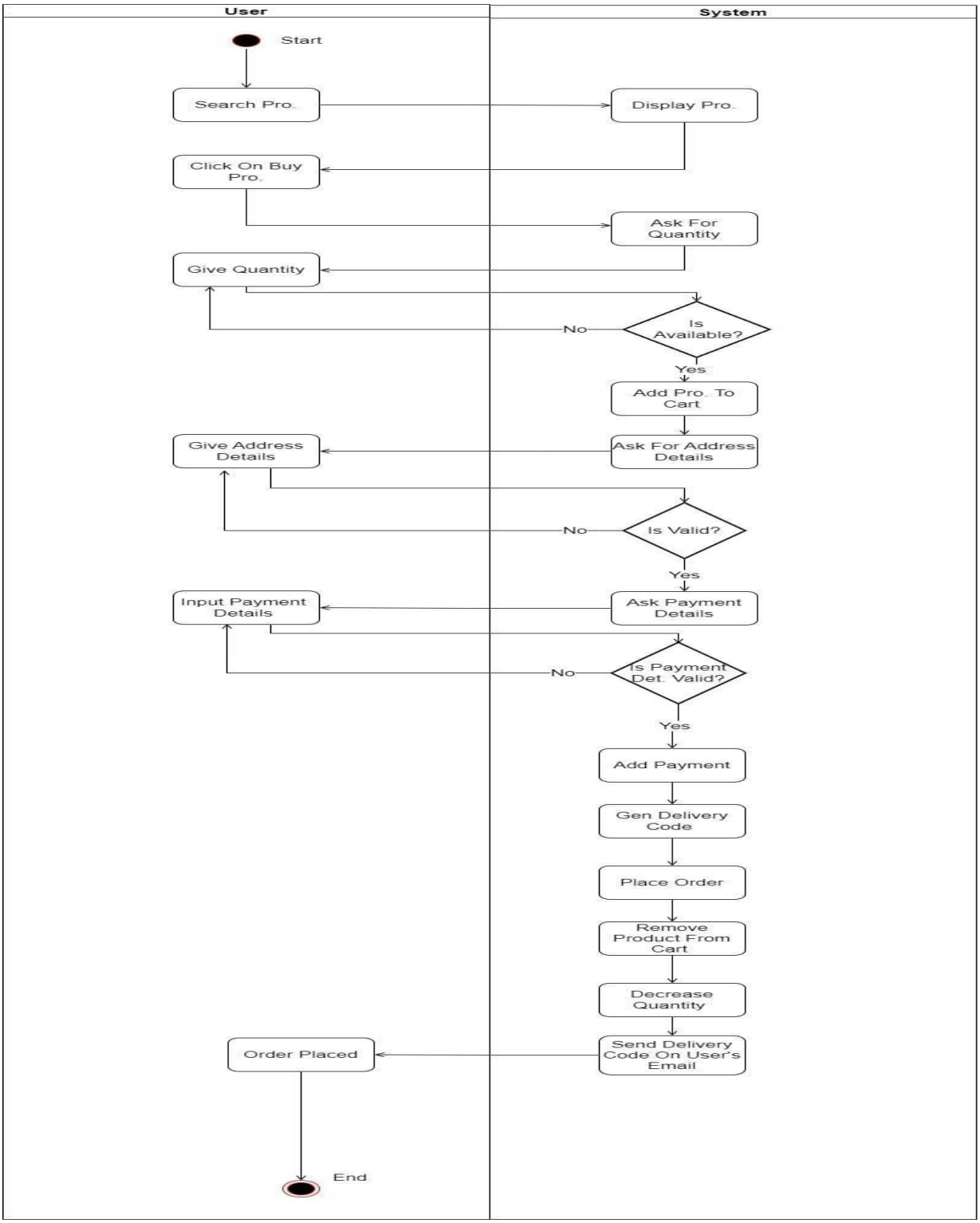
❖ ADMIN REMOVE PRODUCT ACTIVITY DIAGRAM:-



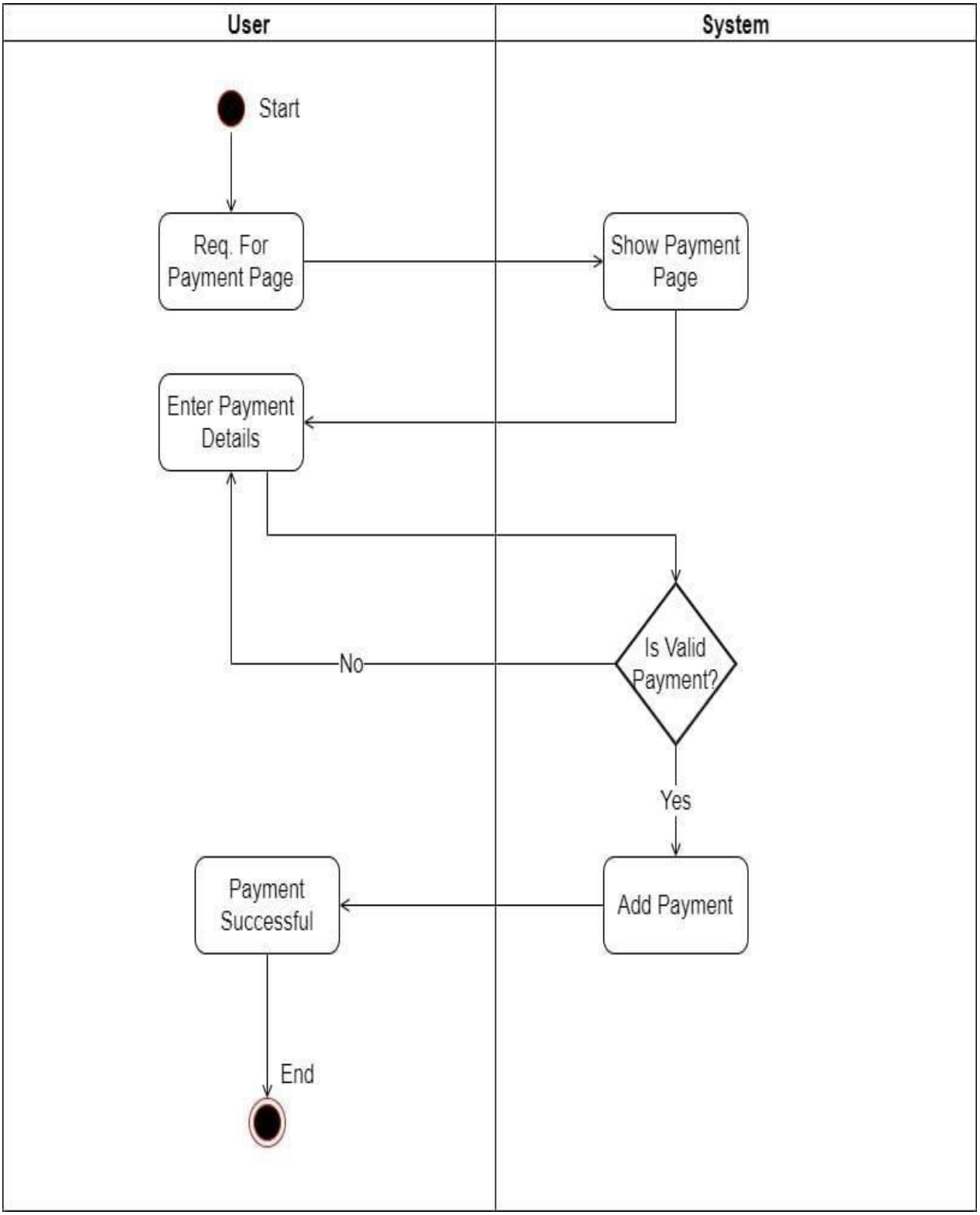
❖ SEARCH PRODUCT ACTIVITY DIAGRAM:-



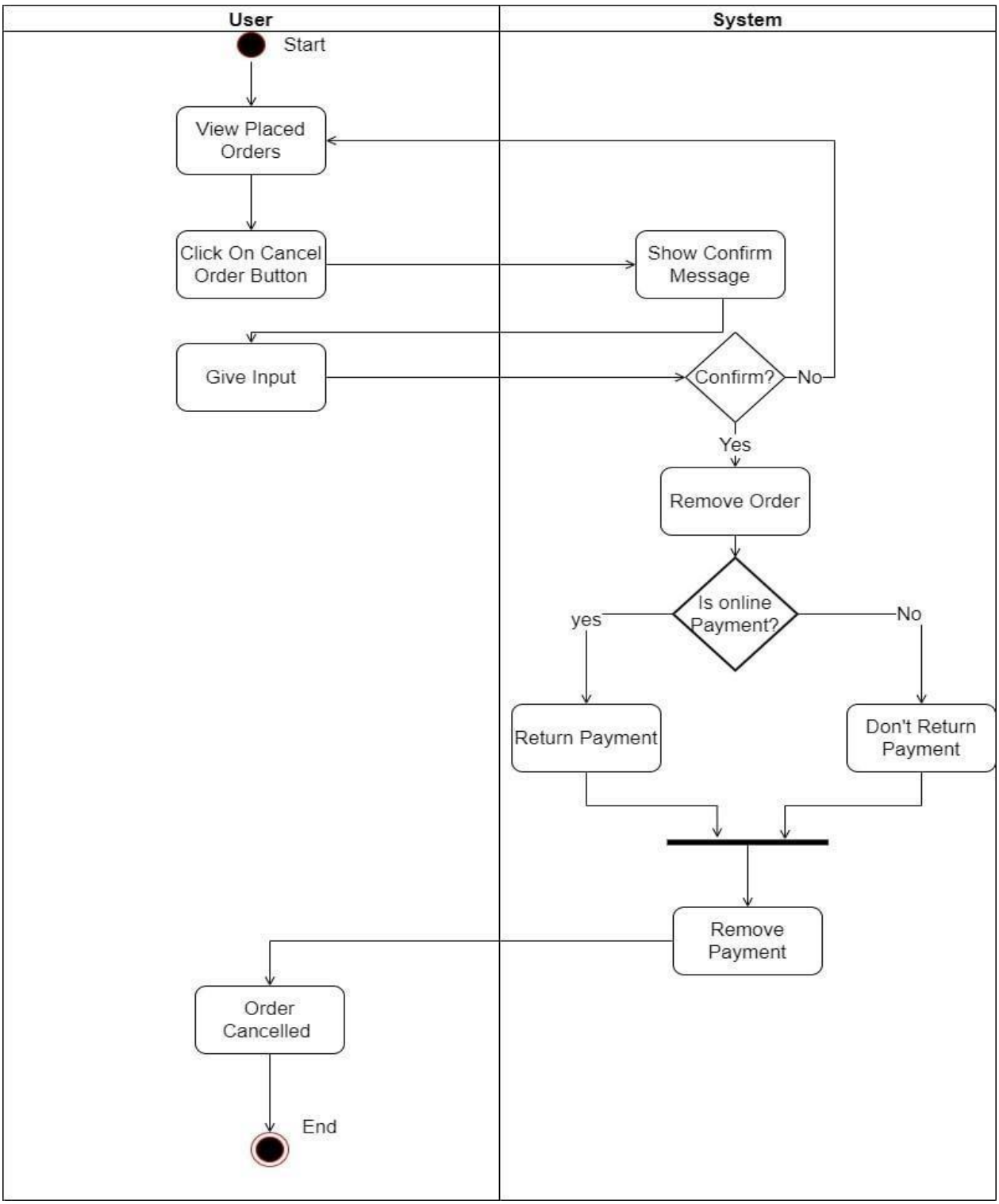
❖ BUY PRODUCT ACTIVITY DIAGRAM:-



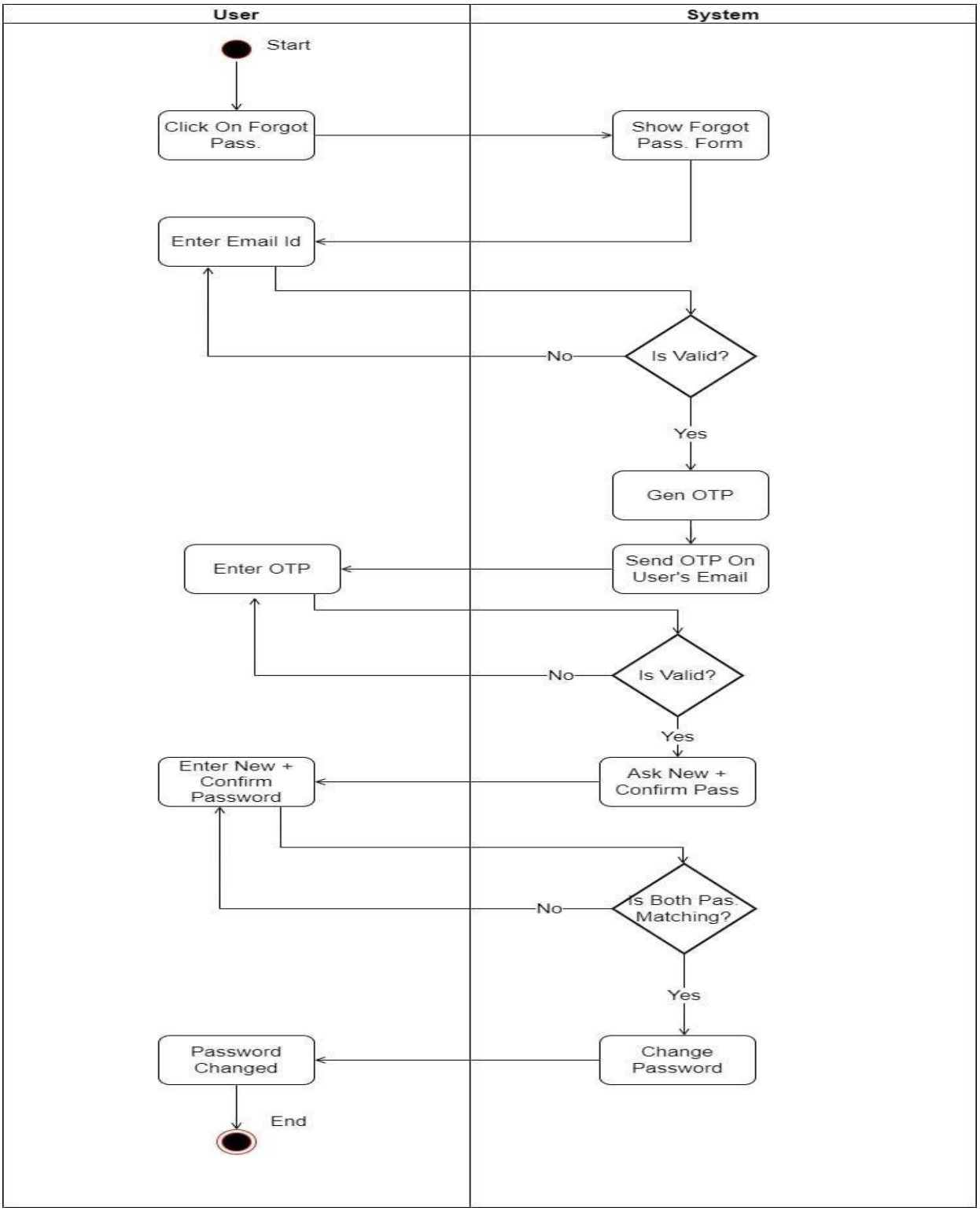
❖ PAYMENT ACTIVITY DIAGRAM:-



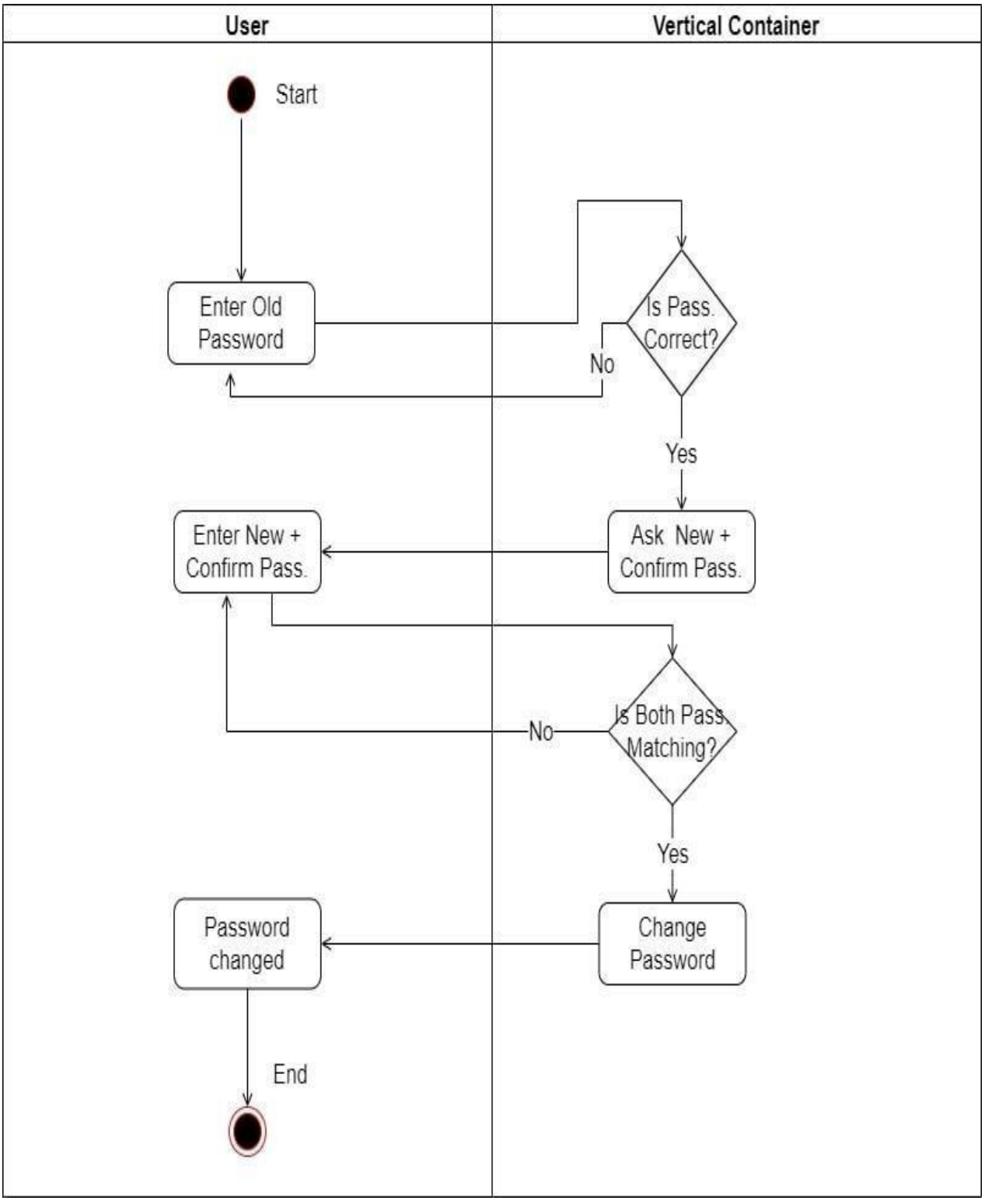
❖ CANCEL ORDER ACTIVITY DIAGRAM:-



❖ FORGOT PASSWORD ACTIVITY DIAGRAM:-



❖ CHANGE PASSWORD ACTIVITY DIAGRAM:-



CLASS DIAGRAMS

- A class diagram is a picture for describing generic descriptions of possible systems.
- Class diagrams and collaboration diagrams are alternate representations of object models.
- Class diagrams contain classes and object diagrams contain objects, but it is possible to mix classes and objects when dealing with various kinds of metadata, so the separation is not rigid we applied that concepts over here.
- Class diagrams contain icons representing classes, interfaces, and their relationships.

❖ SYMBOLS OF CLASS DIAGRAM

- **Aggregation**

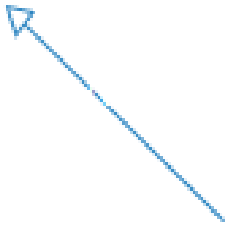
Aggregation is a special type of association in which objects are assembled or configuredtogether to create a more complex object. An aggregation describes a group of objects andhow you interact with them.

- **Dependency**

Dependency relationship is a relationship in which one element, the client,

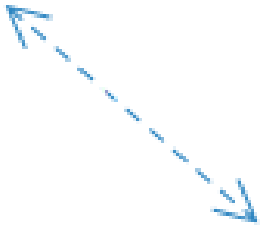
uses or depends on another element, the supplier.

- **Composition**



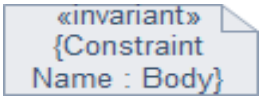
Composition represents whole-part relationships and is a form of aggregation.

- **Generalization**



Generalization is a relationship in which one model element (the child) is based on anothermodel element (the parent).

- **Association**

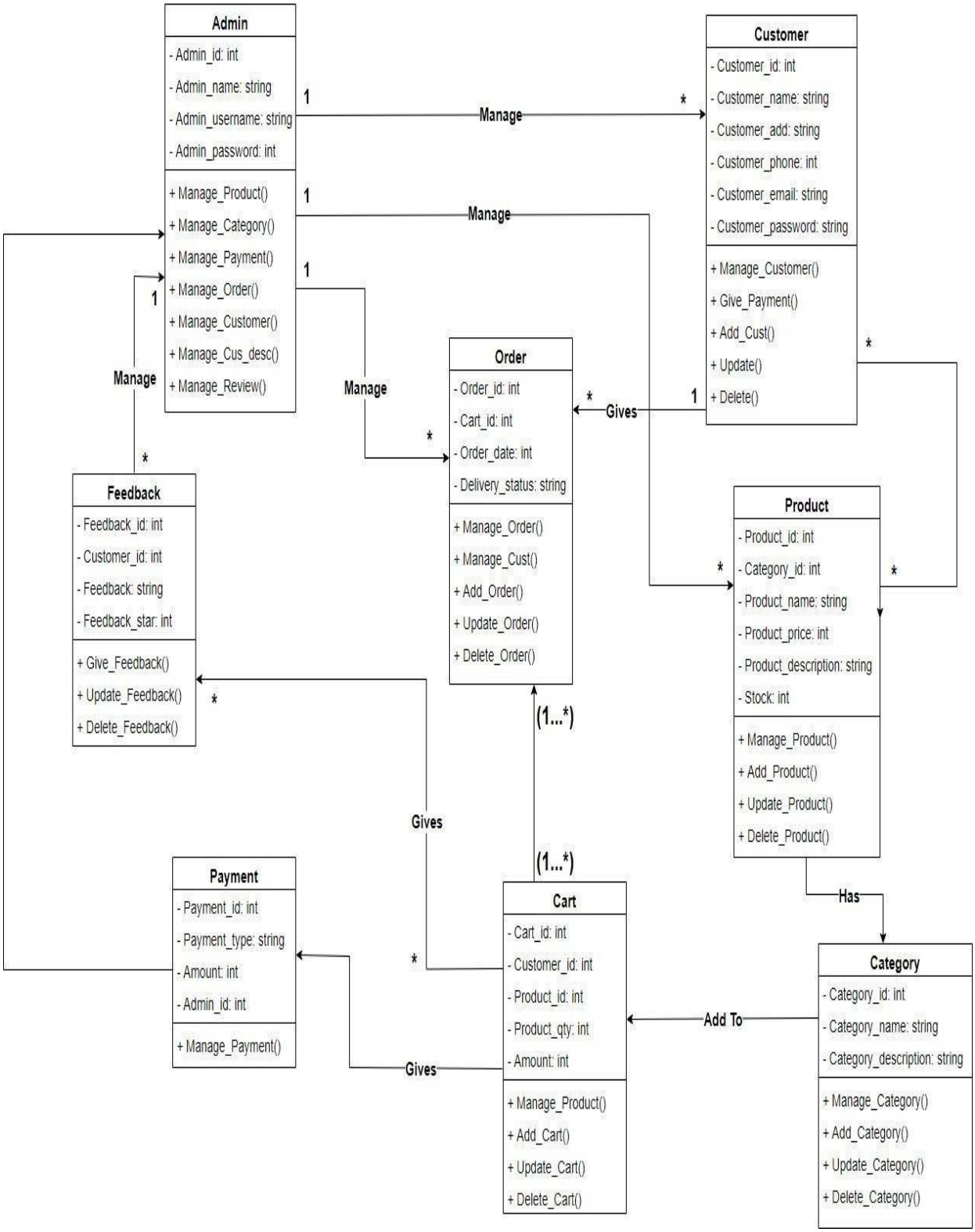


Association is a relationship between two classifiers, such as classes or use cases, that de-scribes the reasons for the relationship and the rules that govern the relationship.

- **Constraint**

Constraint is an extension mechanism that enables you to refine the semantics of a UMLmodel element.

❖ CLASS DIAGRAM:-



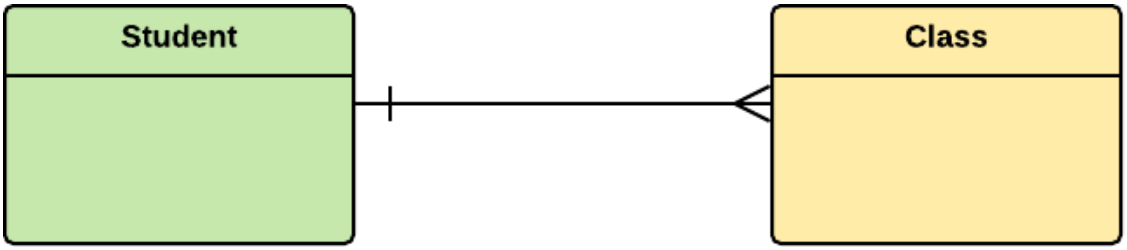
8. E-R DIAGRAM

ENTITY-RELATIONSHIP DIAGRAM (ERD) displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings makethis model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

❖ SYMBOLS OF E-R DIAGRAM

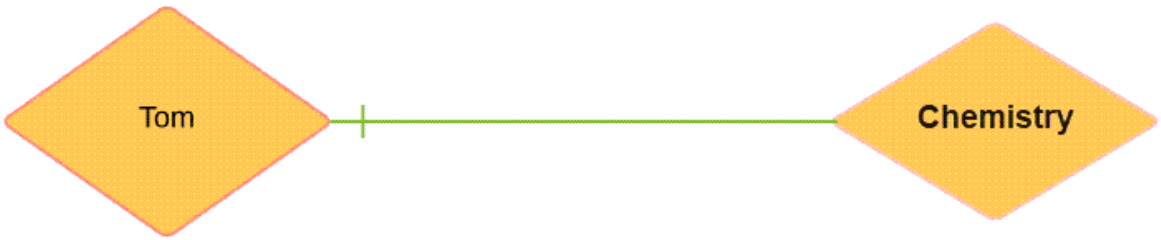
- **Entity Set**

  An entity set is a group of similar kind of entities. It may contain entities with attribute sharing similar values. Entities are represented by their properties, which also called attributes. All attributes have their separate values. For example, a student entity may have a name, age,class, as attributes.



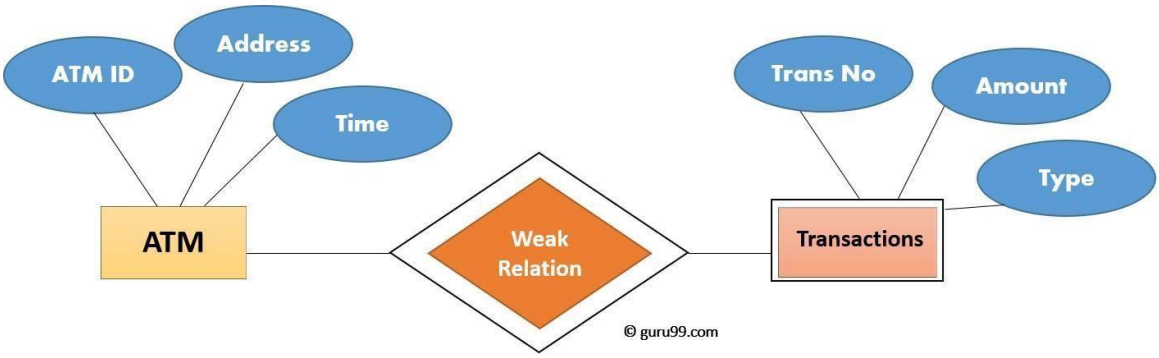
- **Relationship**

  Relationship is nothing but an association among two or more entities. E.g., Tom works inthe Chemistry department.



- **Weak Entities**

  A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need tohave participation.

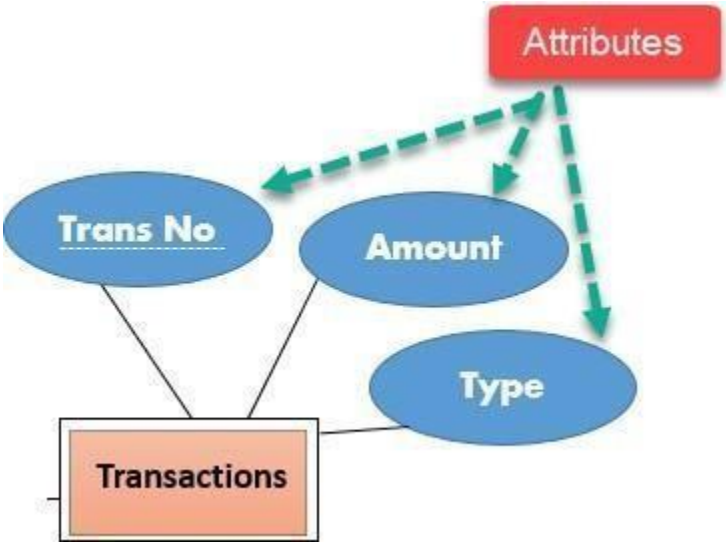


- **Attributes**

  It is a single-valued property of either an entity-type or a relationship-type.

  For example, a lecture might have attributes: time, date, duration, place, etc.

  An attribute is represented by an Ellipse

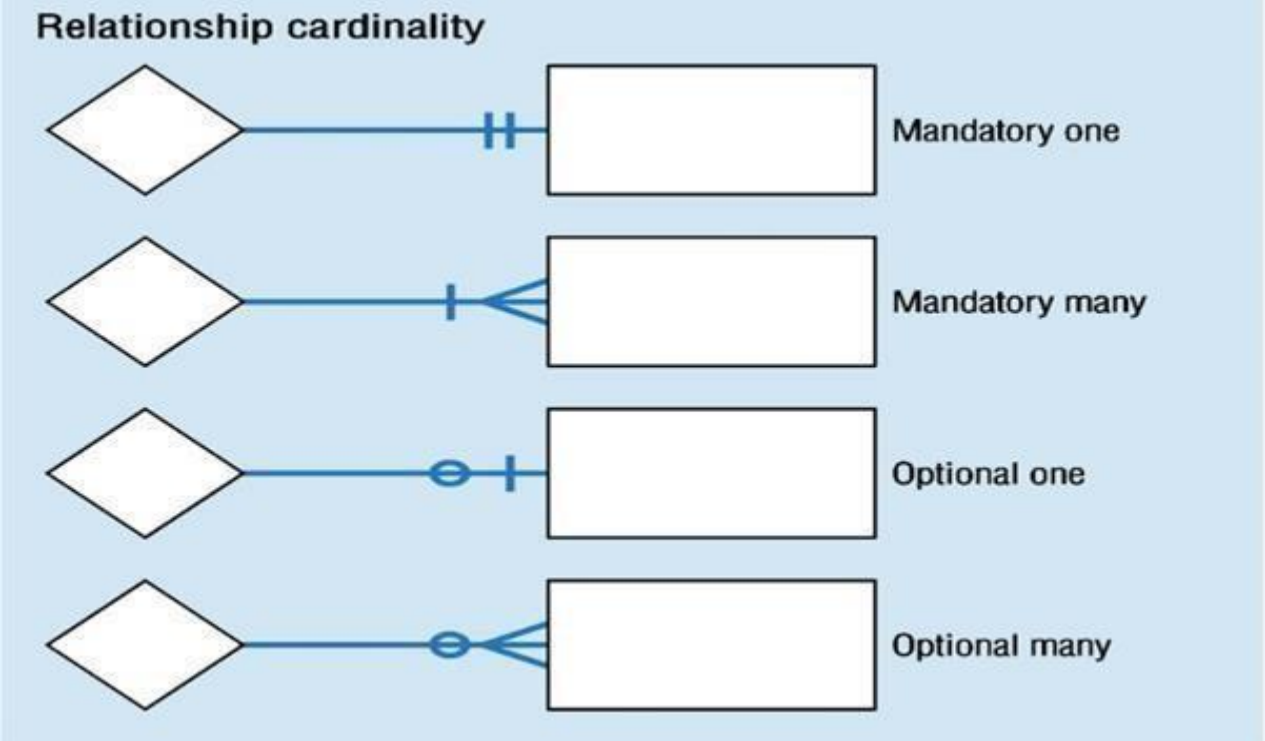


- **Cardinality**

  Defines the numerical attributes of the relationship between two entities or entity sets.

  Different types of cardinal relationships are:

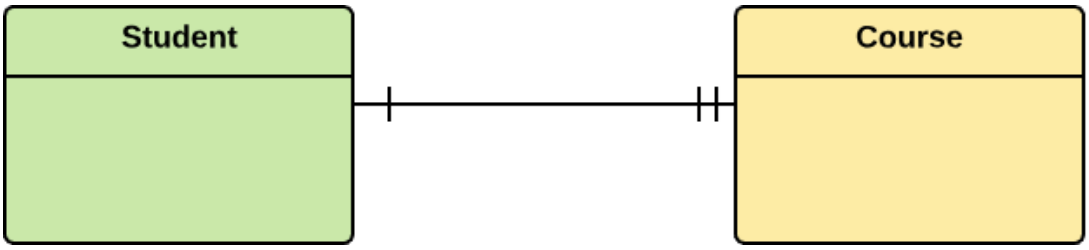
  - One-to-One Relationships
  - One-to-Many Relationships
  - Many-to-One Relationships
  - Many-to-Many Relationships



1. One-to-One:

One entity from entity set X can be associated with at most one entity of entity set Y and viceversa.

Example: One student can register for numerous courses. However, all those courses have asingle line back to that one student.



2. One-to-Many:

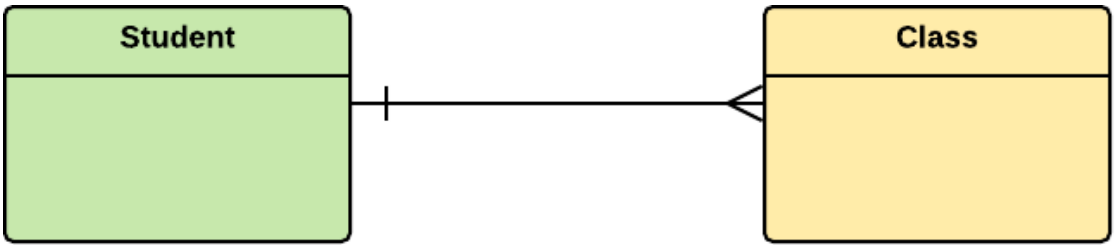
One entity from entity set X can be associated with multiple entities of entity set Y, but anentity from entity set Y can be associated with at least one entity.

For example, one class is consisting of multiple students.

3. Many-to-One:

More than one entity from entity set X can be associated with at most one entity of entity set Y. However, an entity from entity set Y may or may not be associated with more than oneentity from entity set X.

For example, many students belong to the same class.



4. Many-to-Many:

One entity from X can be associated with more than one entity from Y and vice versa.

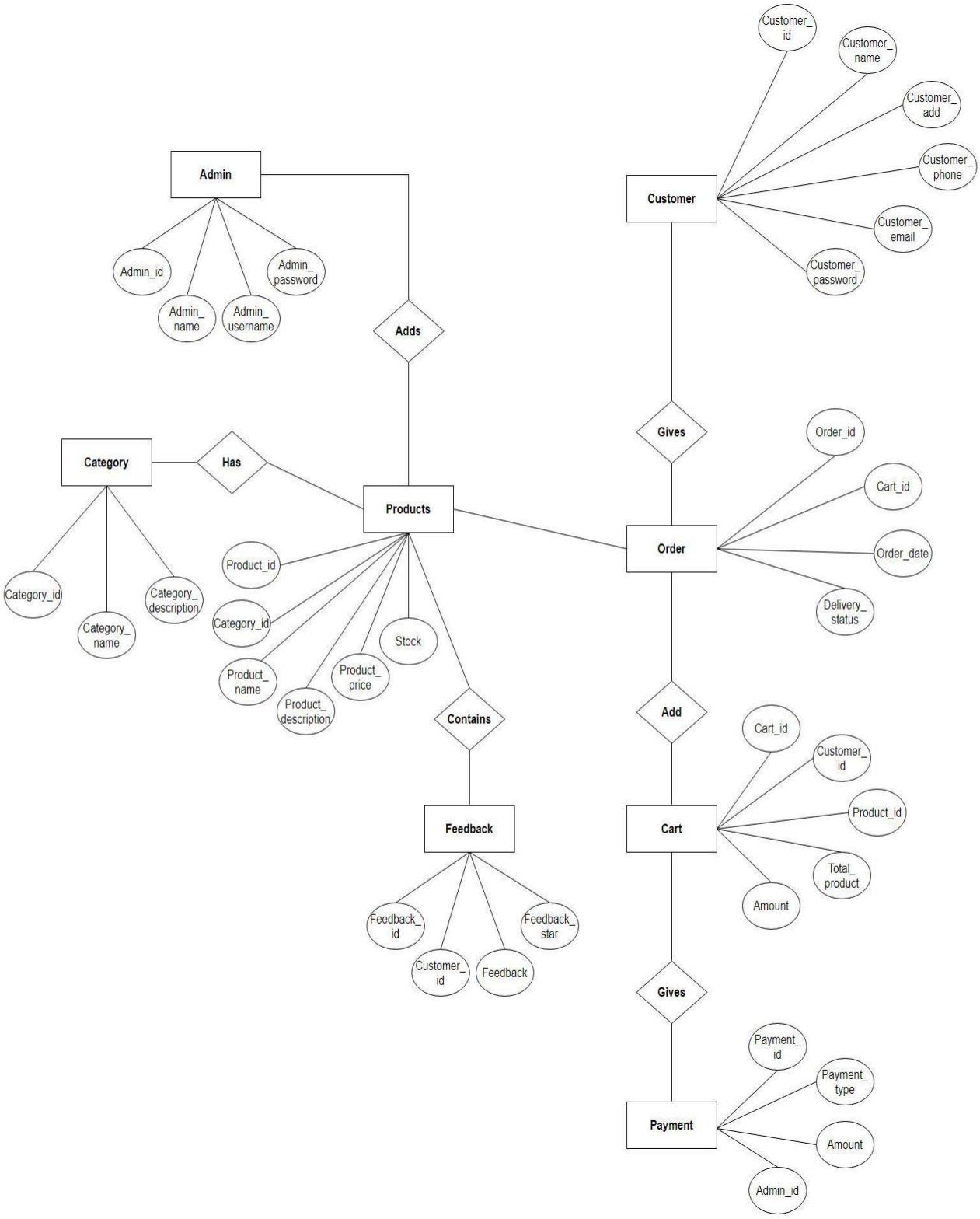
For example, Students as a group are associated with multiple faculty members, and facultymembers can be associated with multiple students.

- **ER-Diagram Notations**

  ER- Diagram is a visual representation of data that describe how data is related to each other.

  - **Rectangles:** This symbol represent entity types
  - **Ellipses :** Symbol represent attributes
  - **Diamonds:** This symbol represents relationship types
  - **Lines:** It links attributes to entity types and entity types with other relationship types
  - **Primary key:** attributes are underlined
  - **Double Ellipses:** Represent multi-valued attributes

❖ E R DIAGRAM:-


Admin
Admin_id
Admin_ name
Admin_ username
Admin_ password
Adds
Customer
Customer_ id
Customer_ name
Customer_ add
Customer_ phone
Customer_ email
Customer_ password
Gives
Category
Has
Category_id
Category_ name
Category_ description
Products
Product_id
Category_id
Product_ name
Product_ price
Product_ description
Stock
Contains
Order
Order_id
Cart_id
Order_date
Delivery_ status
Add
Feedback
Feedback_ id
Customer_ id
Feedback
Feedback_ star
Cart
Cart_id
Customer_ id
Product_id
Total_ product
Amount
Gives
Payment
Payment_ id
Payment_ type
Amount
Admin_id

9. DATA DICTIONARY

- A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to referto them.

- A first step in analyzing a system of objects with which users interacts is to identify each object and its relationship to other objects. This process is calleddata modeling and results in a picture of object-relationships.

- After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes relationship), the type of data (such as text or image or binary value) is described, possible predefined values are listed, and a brief textual descriptionis provided. This collection can be organized for reference into a book calleda data-dictionary.

- When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and basically what the data item means in real-world terms.

❖ A DATA DICTIONARY CONTAINS

- The definitions of all schema objects in the database. (Tables, Views, Indexes, Clusters, Synonyms, Sequences, Procedures, Functions, Packages, Triggers, and soon)
- How much space has been allocated for, and it's currently used by, the schema-objects.
- Integrity constraint-information

.

- The names of My SQL-users.
- Privileges and roles each user has been-granted.
- Auditing information, such as who accessed or updated various schema-objects.
- Other general database information.

❖ **ADMIN:-**

Filed Name	Datatype	Size	Constraint	Description
Admin_id	Int	20	Primary Key	Store the integer Value
Admin_name	Varchar	20	Not Null	Store the Character
Admin_Email	Varchar	20	Not Null	Store the Character
Admin_password	Varchar	20	Not Null	Store the Character

❖ **CUSTOMER:-**

Filed Name	Datatype	Size	Constraint	Description
Customer_id	Int	20	Primary Key	Store the integer Value
Customer_name	Varchar	20	Not Null	Store the Character
Customer_Address	Varchar	200	Not Null	Store the Character
Customer_Email	Varchar	20	Not Null	Store the Character
Customer_Phone_No	Int	10	Not Null	Store the number
Customer_password	Varchar	20	Not Null	Store the Character

❖ CART TABLE:-

Filed Name	Datatype	Size	Constraint	Description
Cart_id	Int	20	Primary Key	Store the integer Value
Customer_id	Int	20	Foreign Key	Store the integer Value
Product_id	Int	20	Foreign Key	Store the integer Value
Product_qty	Int	20	Not Null	Store the integer Value
Amount	Int	10	Not Null	Store the integer Value

❖ ORDER TABLE:-

Filed Name	Data-type	Size	Constraint	Description
Order_id	Int	20	Primary Key	Store the integer Value
Cart_id	Int	20	Foreign Key	Store the integer Value
Order_date	Date	10	Not Null	Store the Date
Delivery_status	Varchar	20	Not Null	Store the Character

❖ **PRODUCT TABLE:-**

Filed Name	Datatype	Size	Constraint	Description
Product_id	Int	20	Primary Key	Store the integer Value
Category_id	Int	20	Foreign Key	Store the integer Value
Product_name	Varchar	50	Not Null	Store the Character
Product_Price	Int	10	Not Null	Store the integer Value
Product_description	Varchar	100	Not Null	Store the Character
Stock	Int	100	Not Null	Store the integer Value

❖ CATEGORY TABLE:-

Filed Name	Datatype	Size	Constraint	Description
Category_id	Int	20	Primary Key	Store the integer Value
Category_name	Varchar	30	Not Null	Store the Character
Category_description	Varchar	100	Not Null	Store the Character

❖ PAYMENT TABLE:-

Filed Name	Datatype	Size	Constraint	Description
Payment_id	Int	20	Primary Key	Store the integer Value
Admin_id	Int	20	Foreign Key	Store the integer Value
Payment_type	Varchar	20	Not Null	Store the Character Value
Amount	Int	10	Not Null	Store the integer Value

❖ FEEDBACK TABLE:-

Filed Name	Datatype	Size	Constraint	Description
Feedback_id	Int	20	Primary Key	Store the integer Value
Customer_id	Int	20	Foreign Key	Store the integer Value
Feedback	Varchar	100		Store the Character
Feedback_Star	Int	5		Store the feedback star out of five