1.What is SDLC ?

- SDLC stand for Software Development Life Cycle

- The life cycle defines a method for improving the quality of software and the all-around development process.

❖ SDLC Phases:

- Requirements Collection/Gathering
- Analysis
- Design
- Implementation
- Testing
- Maintenance

2. What is software testing?

- The process checks whether the actual software matches the expected requirements and ensures the software is bug-free.
- The purpose of software testing is to identify the errors, faults, or missing requirements in contrast to actual requirements
- Software testing can be divided into two steps

❖ Verification:

- It refers to the set of tasks that ensure that the software correctly implements a specific function.
- It means “Are we building the product right?”.

❖ Validation:

- It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.
- It means “Are we building the right product?”.

3. What is agile methodology ?

- Agile SDLC model is a combination of iterative and incremental process models
- Agile methodology is a project framework
- that breaks projects down into several dynamic phases, commonly known as sprints.
- These builds are provided in iterations.
- At the end of the iteration a working product is displayed to the customer
- There is no deadline for project completion

4. What is SRS?

- SRS stand for Software Requirement Specification
- SRS is a complete description
- of the behavior of the system to be developed.
- These requirements can be functional as well as non-functional depending upon the type of requirement
- a set of use cases that describe all of the interactions that
- the users will have with the software.
- Use cases are also known as functional requirements.
- Non-functional requirements are performance requirements, quality standards, or design Constraints

❖ Types of Requirements

- Customer Requirements
- Functional Requirements
- Non-Functional Requirements

5. What is oops ?

- OOP stnad for Object-Oriented Programming Language
- there are four basic Object-Oriented Programming Concepts

- Abstraction
- Inheritance
- Encapsulation
- Polymorphism

- Java, Python, and C++ are a few examples of Object-oriented programming languages
- Here are reasons why object-oriented programming language is preferred

- It reduces the complexity of the program by breaking it into objects and classes.
- The development of programs is relatively faster with the help of its various features
- The software is also easy to maintai and has simple updating processes.

6. Write Basic Concepts of oops ?

❖ Class:

- A class is a user-defined data type.

- It represents the set of properties or methods that are common to all objects of one type.
- A class is like a blueprint for an object.

❖ Objects:

- An object can be considered a real-life entity having a state and behaviour.
- Example:
- State: Name, Age, Gender, Address.
- Behavior: Reading, Writing, Running.

❖ Data Abstraction:

- Data abstraction is one of the most essential and important features of oops
- Data abstraction refers to providing only essential information about the data to the outside world
- and hiding the background details or implementation

- Encapsulation:
- Encapsulation is defined as the wrapping up of data under a single unit.

- In Encapsulation, the variables or data of a class are hidden from any other class
- the data in a class is hidden from other classes,
- so it is also known as data-hiding.

❖ Inheritance:

- Inheritance means that one class inherits the characteristics of
- another class.
- This is a very important concept of object-oriented programming
- this feature helps to reduce the code size.
- Inheritance describes the relationship between two classes
- its characteristics from a parent class and then add unique
- features of its own.

❖ Polymorphism:

- Polymorphism means “having many forms”.

- It allows different objects to respond to the same message in different
- ways, the response specific to the type of the object.
- That is a single function or an operator functioning in
- many ways different upon the usage is called polymorphism.

6. What is object ?

- An object can be considered a real-life entity having a state and behaviour.
- Example:
  - State: Name, Age, Gender, Address.
  - Behavior: Reading, Writing, Running.

7. What is class ?

- A class is a user-defined data type.
- It represents the set of properties or methods that are common to all objects of one type.
- A class is like a blueprint for an object.

8. What is encapsulation ?

- Encapsulation is defined as the wrapping up of data under a single unit.
- In Encapsulation, the variables or data of a class are hidden from any other class
- the data in a class is hidden from other classes,
- so it is also known as data-hiding.

9. What is inheritance ?

- Inheritance means that one class inherits the characteristics of another class.
- This is a very important concept of object-oriented programming
- this feature helps to reduce the code size.
- Inheritance describes the relationship between two classes
- its characteristics from a parent class and then add unique
- features of its own.

10. What is polymorphism ?

- Polymorphism means “having many forms”.
- It allows different objects to respond to the same message in different ways,
- the response specific to the type of the object.
- That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.

11. Write SDLC phases with basic introduction

- SDLC stand for Software Development Life Cycle
- The life cycle defines a method for improving the quality of software and the all-around development process.
- SDLC Phases
- that describes how to develop, maintain, replace, and enhance specific software.
- The life cycle defines a method for improving the quality of software and the all-around development process.

❖ Requirements Collection/Gathering:

- Although requirements may be documented in written form, they may
- be incomplete, unambiguous, or even incorrect.
- Requirements definitions usually consist of natural
- language, supplemented by (e.g., UML) diagrams
- and tables.
- Requirements are both type Functional and Non-functional
  - Functional : like Update the database on the server
  - Non-functional: are constraints on the system or the development process.

❖ Analysis:

- The analysis phase defines the requirements of the system,
- This phase defines the problem that the customer is trying to solve.
- The deliverable result at the end of this phase is a
- requirement document.

❖ Design:

- Design Architecture Document
- Implementation Plan
- Critical Priority Analysis
- Performance Analysis
- Test Plan
- Depending on the project subject, the design phase products include
- dioramas, flow-charts, sketches, site trees, HTML screen.

❖ Implementation

- the project begins to take shape. The construction of the final product is the focus of this stage.
- The construction of the final product
- Implementation - Code
- Critical Error Removal
- The implementation phase deals with issues of quality, performance,
- baselines, libraries, and debugging.

❖ Testing:

- The testing phase is a separate phase which is performed by a different
- team after the implementation is completed.
- many types of testing are :

- Regression Testing
- Internal Testing
- Unit Testing
- Application Testing
- Stress Testing

❖ Maintenance:

- Maintenance is the process of changing a system after it has been deployed
- types of maintenace
  - Corrective maintenance: identifying and repairing defects

- Adaptive maintenance: adapting the existing solution to the new platforms
- Perfective Maintenance: implementing the new requirements In a spiral lifecycle

12. Explain Phases of the waterfall model ?

- waterfall model as know as classical model
- The waterfall is unrealistic for many reasons
- Requirements must be “frozen” to early in the life cycle
- Requirements are validated too late

- when to use:
- Waterfall Model is Used for short term project
- We can use this model when requirements are fixed
- We can use this model when project requirements are not ambiguous
- We can use this Model if Project Requirement is very well documented, clear and fixed
- Technology is understood and is not dynamic.
- Product definition is stable.

❖ Pros

- Simple and easy to understand and use
- Easy to manage due to the rigidity
- Easy to arrange tasks.
- Phases are processed and completed one at a time.

❖ Cons

- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- No working software is produced until late during the life cycle.
- It is difficult to measure progress within stages.

13. Write phases of spiral model.

- Spiral Model is very widely used
- phases of spiral model:

1.Planning:The first phase of the Spiral Model is the planning phase a plan is created for the next iteration of the spiral.

2.Risk Analysis: In the risk analysis phase,

the risks associated with the project are identified and evaluated.

3.Engineering: In the engineering phase,

the software is developed based on the requirements gathered in the previous iteration.

4.Evaluation:it meets the customer's requirements and

if it is of high quality.

5. Planning: The next iteration of the spiral begins

with a new planning phase, based on the results of the evaluation.

14. Write agile manifesto principles.

- Agile methodology is a project management framework
- The Agile Manifesto is a document that focuses on four
- values and 12 principles for Agile software development

1.Our highest priority is to satisfy

the customer through the early and continuous delivery of valuable software.

2.Welcome changing requirements,

even late in development. Agile processes harness change for the customer’s competitive advantage.

3.Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4.Business people and developers must work together daily throughout the project.

5.Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6.The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7.Working software is the primary measure of progress.

8.Agile processes promote sustainable development.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9.Continuous attention to technical excellence and good design enhances agility.

10.Simplicity–the art of maximizing the amount of work not done–is essential.

15. Explain working methodology of agile model and also write pros and cons.

- Agile Model:

➢ Agile Model is combination of iterative and incremental model

➢ Each iteration typically lasts from about one to three weeks.

➢ Agile Methods break the product into small incremental builds.

➢ At the end of the iteration a working product is displayed to the customer

➢ There is no deadline for project completion

❖ Pros:

➢ Is a very realistic approach to software development

➢ Promotes teamwork and cross training.

➢ Functionality can be developed rapidly

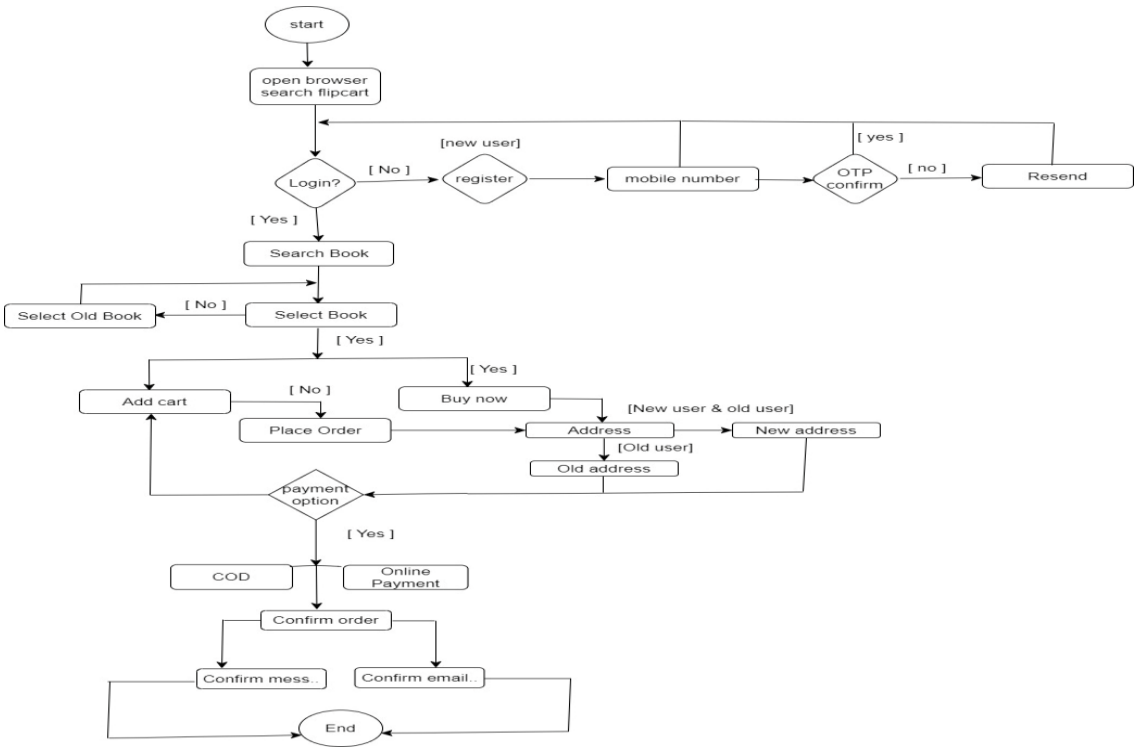
➢ Suitable for fixed or changing requirements

- Little or no planning required
- Easy to manage
- Gives flexibility to developers

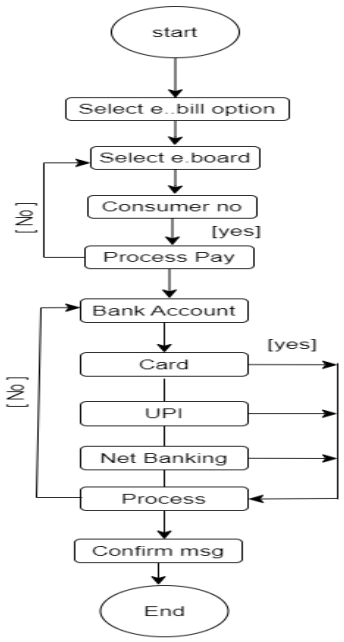
❖ Cons:

- Not suitable for handling complex dependencies.
- Strict delivery management dictates
- Transfer of technology to new team members

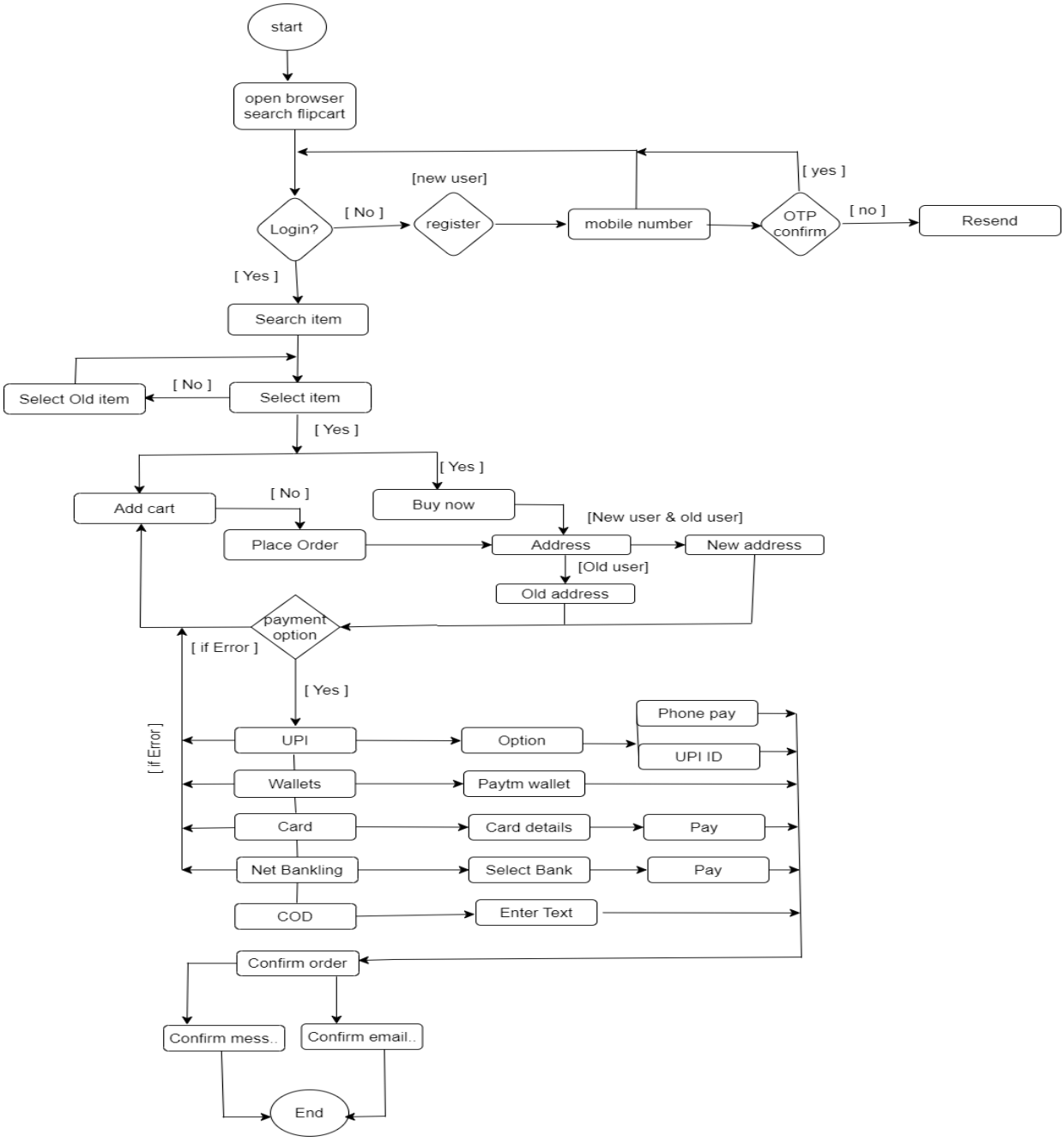
16. Draw Usecase on Online book shopping



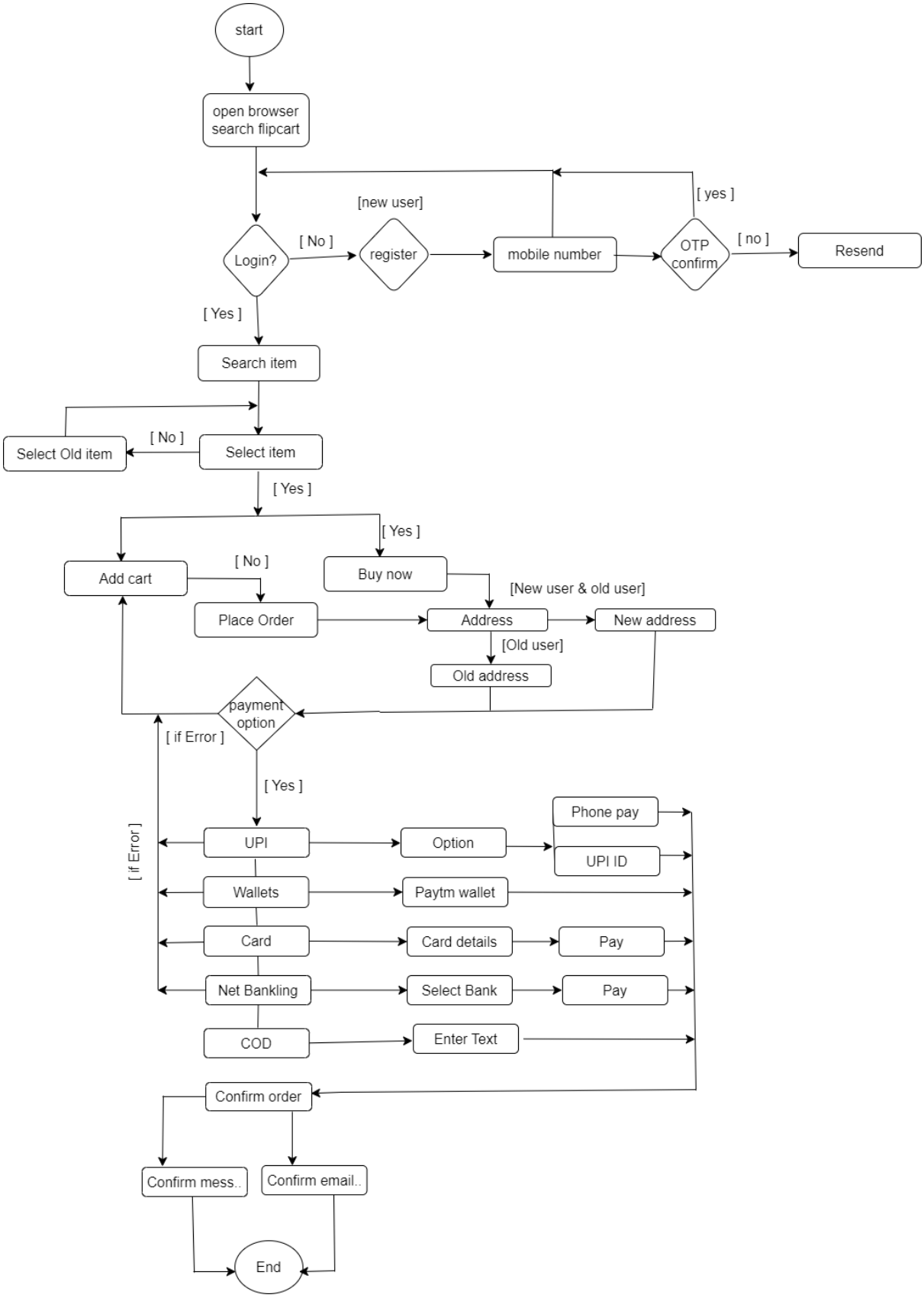
17. Draw Usecase on online bill payment system (paytm)



18. • Draw usecase on Online shopping product using COD.



19. Draw usecase on Online shopping product using payment gateway.


start
open browser search flipcart
Login?
[No]
[new user]
register
mobile number
OTP confirm
[yes]
[no]
Resend
[Yes]
Search item
Select item
[No]
Select Old item
[Yes]
Add cart
[No]
Buy now
[Yes]
Place Order
[New user & old user]
Address
New address
[Old user]
Old address
payment option
[if Error]
[Yes]
[if Error]
UPI
Option
Phone pay
UPI ID
Wallets
Paytm wallet
Card
Card details
Pay
Net Bankling
Select Bank
Pay
COD
Enter Text
Confirm order
Confirm mess..
Confirm email..
End