**Module -1(fundamental)**

1. **What is SDLC?**

* **SDLC means a Software Development Life Cycle.**
* **SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.**
* **There are a number of different development models.**

1. **What is software testing?**

* **Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.**
* **Test execution is only a part of testing, but not all of the testing activities**
* **It can also be stated as the process of validating and verifying that a software program or application or product.**

1. **What is agile methodology?**

* **Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.**
* **Agile Methods break the product into small incremental builds.**

1. **What is SRS**

* **Software requirement specification**
* **A software requirements specification (SRS) is a complete description as the behavior of the system to be developed.**
* **It includes a set of use cases that describe all of the interaction that the users will have with the software.**

**User cases are also known as functional requirement. In addition to use cases, the SRS also contains nonfunctional requirements.**

1. **What is Oops**

* **Identifying objects and assigning responsibilities to these objects.**
* **Objects communicate to other objects by sending messages.**
* **Message are received by the methods of an object**
* **An object is like a black box.**
* **The internal details are hidden.**

1. **Write basis concepts of Oops**

* **Object**
* **Class**
* **Encapsulation**
* **Inheritance**
* **Polymorphism**
* **Overriding**
* **Overloading**
* **Abstraction**

1. **What is object**

* **An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.**
* **An” object” is anything to which a concept applies.**
* **This is the basic unit of object oriented programming (OOP).**

1. **What is class**

* **Class is a collection of data member (variable) and member function (process, methods) with its behaviors.**
* **This doesn’t actually define any dada, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object**

1. **What is encapsulation**

* **Data hiding: wrapping up of data into single unit private your data member or member or member function.**
* **Encapsulation is the practice of including in an object everything it need hidden from other object. The internal state is usually not accessible by other object.**

**10) what is inheritance**

* **Properties of parent, class extends into child class.**
* **Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship**

**11) what is polymorphism**

* **Ability to take one name having different forms**
* **: many forms or multiple forms**
* **: there are mainly 2 type**
* **compile time (method overloading)**
* **run time (method overriding)**

**12 write SDLC phases with basic introduction**

* **SDCL phases**
* **Requirement collection**
* **Analysis**
* **Design**
* **Implementation**
* **Testing**
* **Maintenance**

**1) Requirement collection**

**Features**

**Usage scenarios**

**Requirements will change**

**Plan for change**

**Early prototyping**

**2) analysis phase**

**The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.**

**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.**

**3) design phase**

**Design architecture document**

**Implementation plan**

**Critical priority analysis**

**Performance analysis**

**Test plan**

**4) implementation phase**

**In the implementation phase, the team builds the components either from scratch or by composition.**

**For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline.**

* **Implementation- code**
* **Critical error removal**

**5) testing phase**

**Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.**

**It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.**

**6) maintenance phase**

**Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.**

**The developing organization or team will have some mechanism to document and track defects and deficiencies.**

**13 explain phases of the waterfall model**

* **Requirement collection**
* **Analysis**
* **Design**
* **Implementation**
* **Testing**
* **maintenance**

**14 write phases of spiral model**

* **Planning**
* **Risk analysis**
* **Engineering**
* **Customer evaluation**

**15) write agile manifesto principal**

* **Meeting client requirement**
* **Taking changes**
* **Pull off feedback**
* **Support teamwork**
* **Face to face talk**
* **Measure progress**
* **Direct on active members**
* **Self-organized**
* **Sustainable development process**
* **Technical excellence**
* **Adjust strategies**
* **Monitor the product cycle**

**16) explain working methodology of agile model and also write pros and cons.**

* **Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.**
  + - **pros**
* **Is a very realistic approach to software development**
* **Promotes teamwork and cross training.**
* **Little or no planning required**
* **Easy to manage**
* **Gives flexibility to developers**
  + - **Cons**
* **Not suitable for handling complex dependencies**
* **More risk of sustainability, maintainability and extensibility**
* **An overall plan, an agile leader and agile PM practice is a must without**

**which it will not work.**

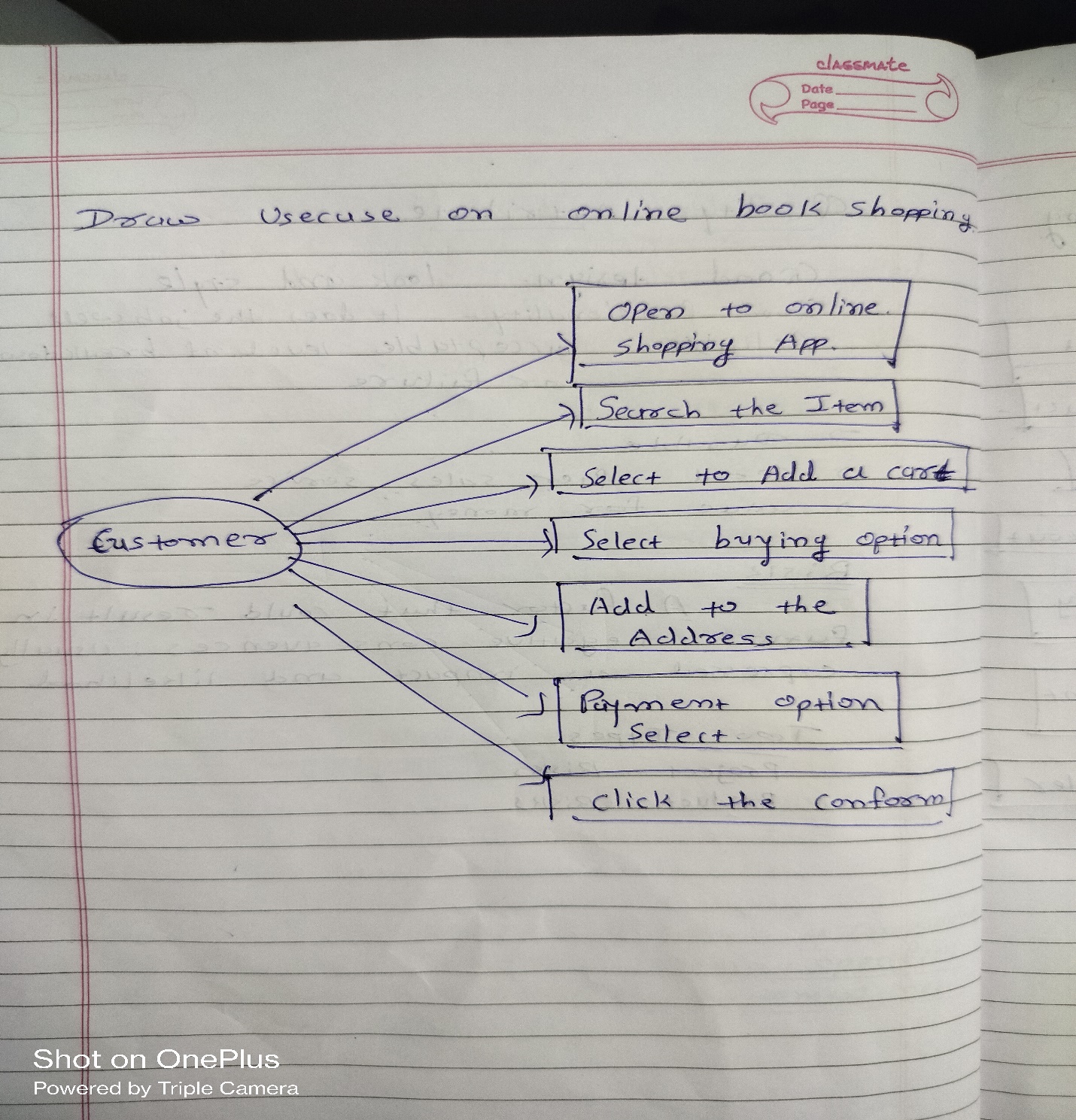
* **Transfer of technology to new team members may be quite challenging**

**due to lack of documentation**

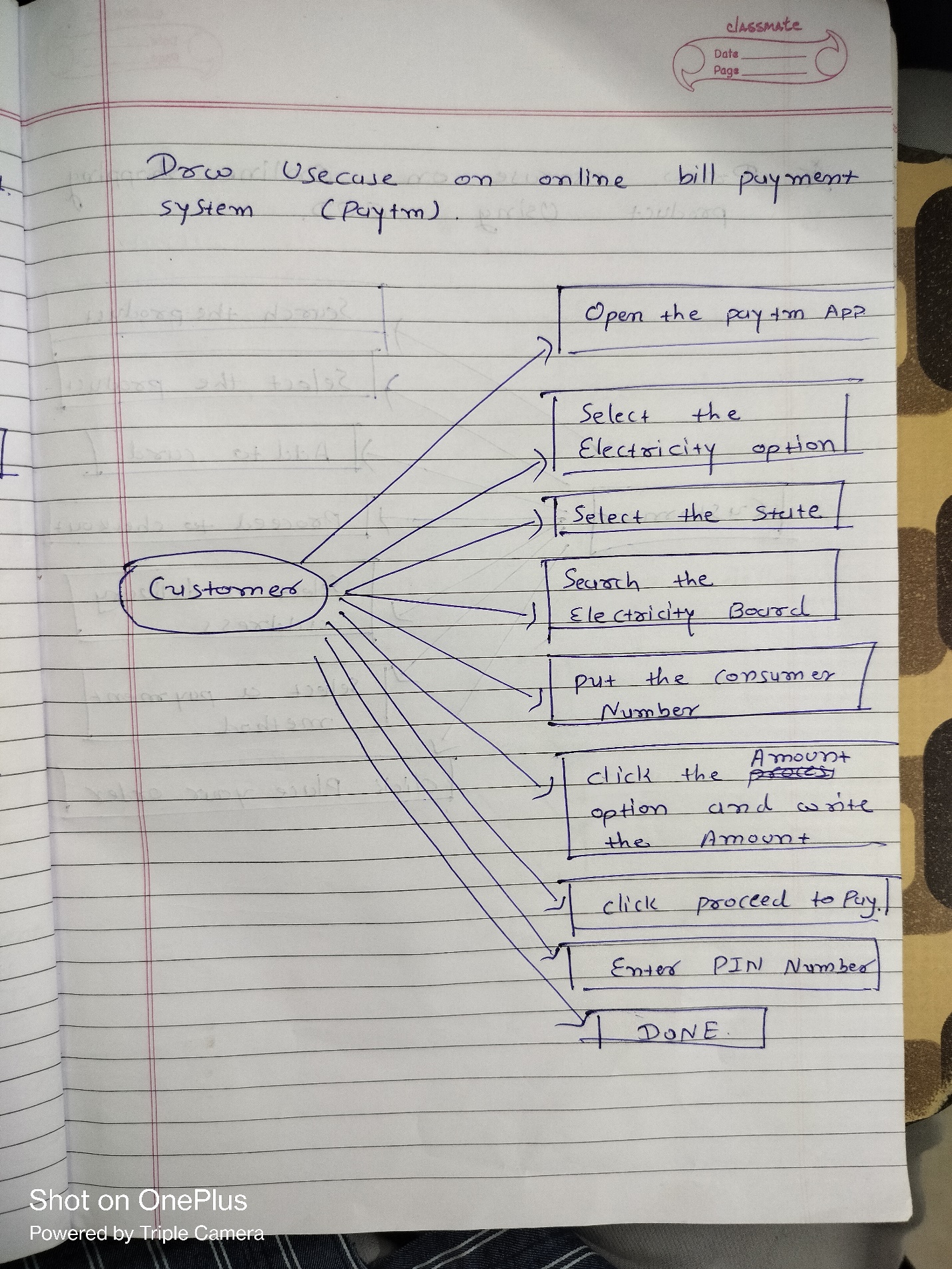
* **There is very high individual dependency, since there is minimum**

**documentation generated.**

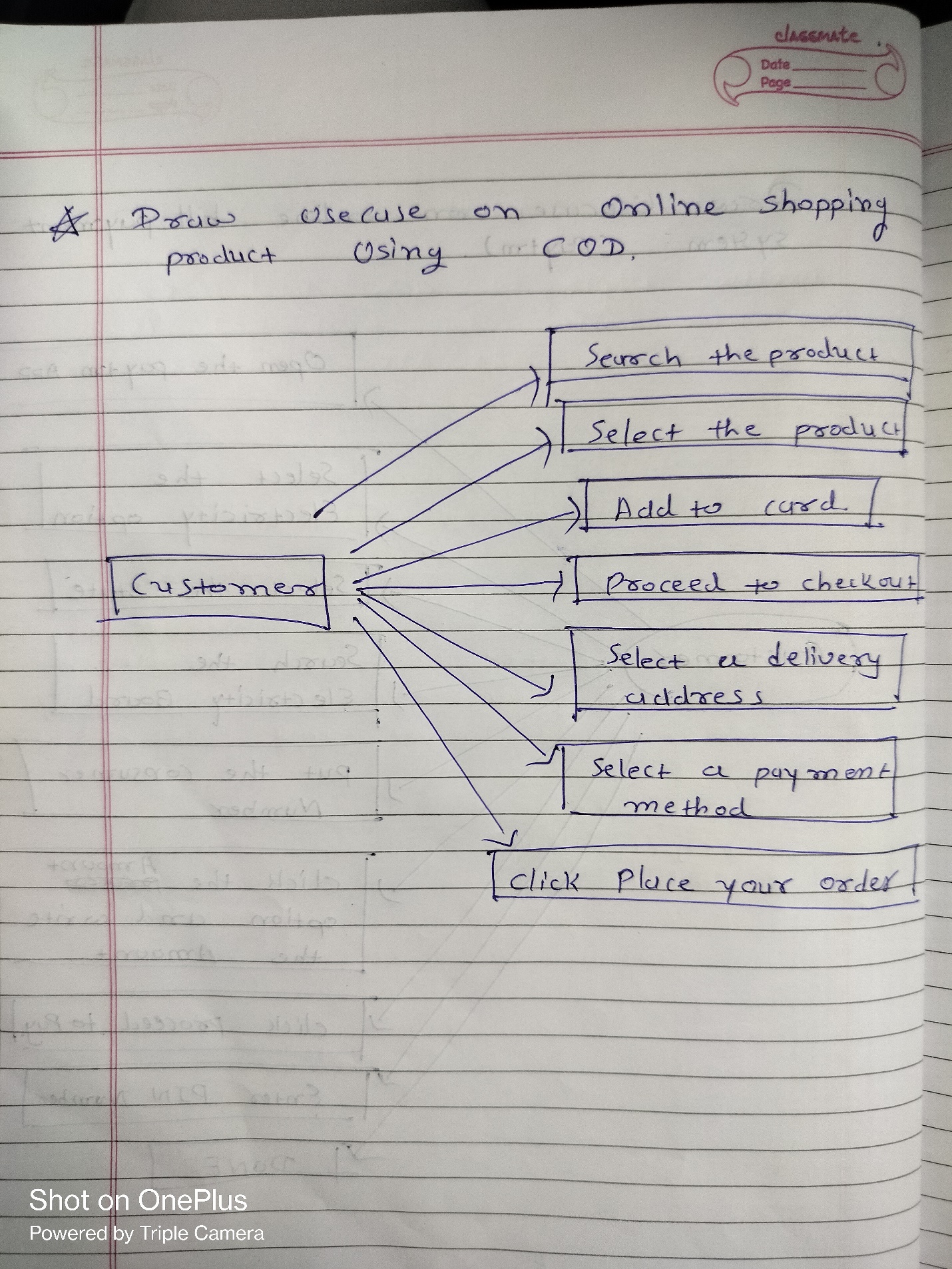
**17) Draw usecase on online book shopping**

****

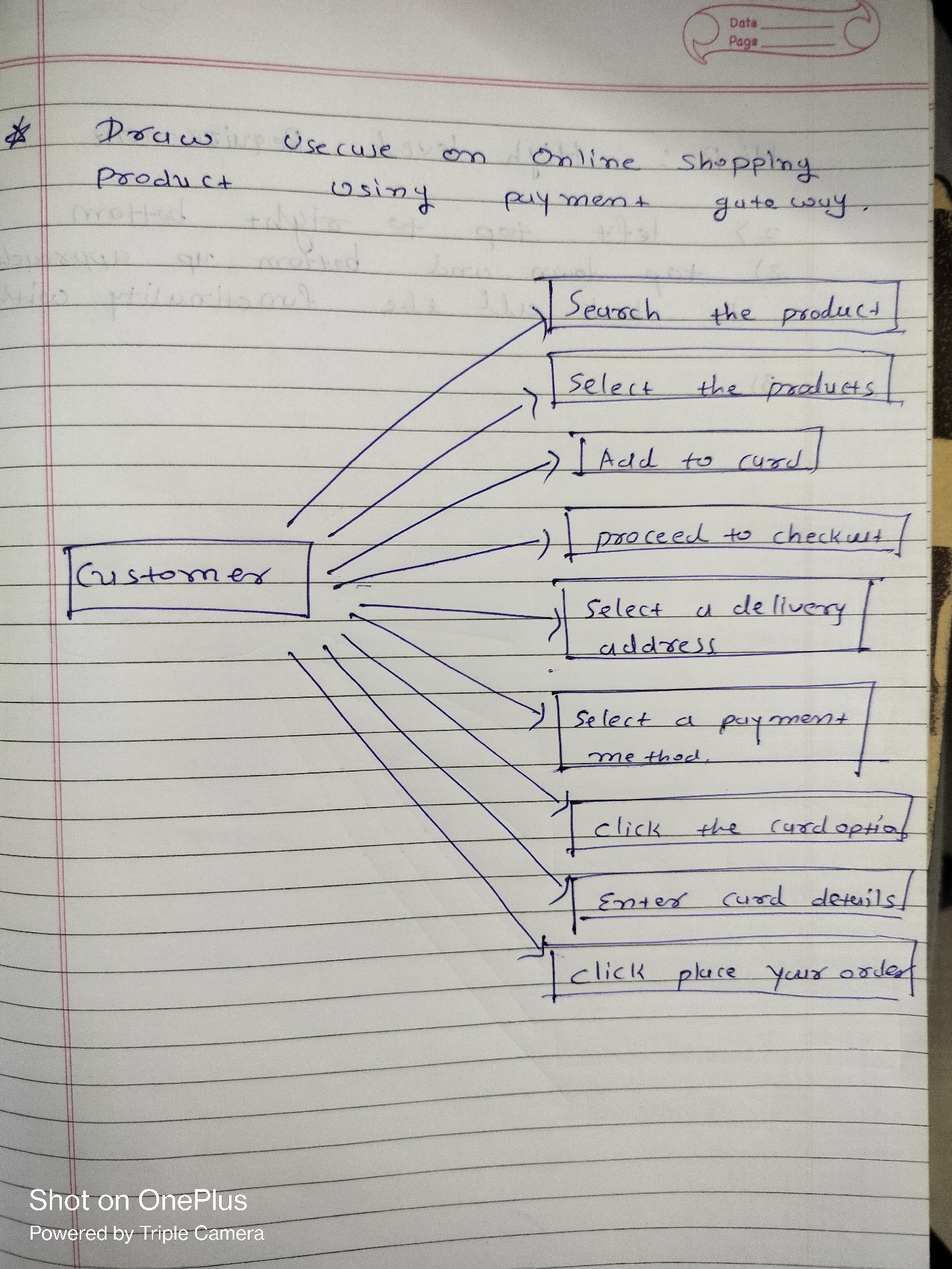
**18) draw usecase on online bill payment system (paytm)**

****

**19) draw usecase on online shopping product using cod**

****

**20) draw usecase on online shopping product using payment gateway.**

****