**1) What is SDLC**

1. 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.

**2 what is Software testing?**

1. A Software Testing is a process used to identify the Correctness, Completeness, and equality of developed computer software.

**3) what is agile methodology?**

A agile model believes that every project need to be handled differentlyand the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes to deliver specific features for release.

**4) What is SRS**

A. A software requirements specification is a complete description of the behaviour of the system to be developed.

**5) write SDLC phases with basic introduction**

A 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 .

**Requirement Gathering**

* Features
* Usage scenarious
* Requirements will change!
* User and business needs change during the project
* Build constant feedback into the project plan
* Plan for change

**Three types of problem scanarise:**

1. **Lack of clarity :** It is hard to write documents that are both preciseand easy to read.
2. **Requirement confusion** : Functional and Non- functional Requirements tend to be intertwined .
3. **Requirements Amalgamation** : Several different requirements may be expressed together .

**Analysis Phase**

* The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
* The deliverable result at the end of this phase is a requirement document.
* Ideally, this document states in a clear and precise fashion what is to be built.
* The analysis represents the “ What” phase.

**Design phase:**

* Design Architecture Document
* Implementation plan
* Critical Priority Analysis
* Performance Analysis
* Test Plan

**Implementation Phase:**

* In the implementation phase, the team builds the components either document from design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.
* Implementation –code
* Critical Error Removal

**Testing Phase**

* Configuration and version management
* Updating all analysis, design and user documentation
* Repeatable, automated tests enable evolution and refactoring
* The developing organization or team will have some mechanism to document and track defects and deficiencies.

**Maintenance**

Is the process of changing a system after it has been deployed.

1. **Corrective maintenance:** identifying and repairing defects.
2. **Adaptive maintenance:** adapting the existing solution to the new platforms.
3. **Perfective Maintenance:** implementing the new on decides the utility and value of the software at a particular level of quality outweighs the impact of the known defects and deficiencies.

**6)Explain phases of the waterfall model**

* The classical software life cycle the software development as a step by –step “Waterfall” between the various development phases.
* Requirements must be ”frozen” to early in the life product.
* The project is short.

**Pros:**

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model.
* Clearly defined stages.
* Easy to arrange tasks.
* Process and results are well documented.

**Cons**

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Poor model for long and ongoing projects.
* Not a good model for complex and object-oriented projects.

7) **Write phases of spiral mode**

* Spiral model is very widely in the software as it is in synch with the natural development process of any product learning with maturity and also involves minimum risk for the customer as well as the development firms.
* For medium to high risk projects.
* Requirements are complex and need evaluation to get clarity.

**Pros**

* Changing requirements can be accommodated.
* Allows for extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.

**Cons:**

* Management is more complex.
* End of project may not be known early.
* Not suitable for small or low risk projects and could be expensive for small projects.
* Process is complex.
* Spiral may go indefinitely.

8) Explain working methodology of agile modal and also write pros and cons.

A. Agile SDLC model is a combination of iterative and incremental process models with focus on process adeptability and customer satisfaction by rapid delivery of working software product.

**Prose:**

* Is a very realistic approach to software development
* Promotes tem work and cross training?
* Functionality can be developed rapidly and demonstrated.
* Suitable for fixed or changing requirements

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.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

9) Write agile manifesto principles

* Individual interaction
* Working software
* Customer collaboration
* Responding to change

10) What is oops

* Object oriented programming systems
* Black box testing

11) What is object

* Its an instances of class.

12) What is encapsulation

* Data hiding
* Wrapping up of data into single unit private your data member and member function.

13) What is inheritance

* Properties of parent class extends into child class.
* Properties of super class extends, into subclass.

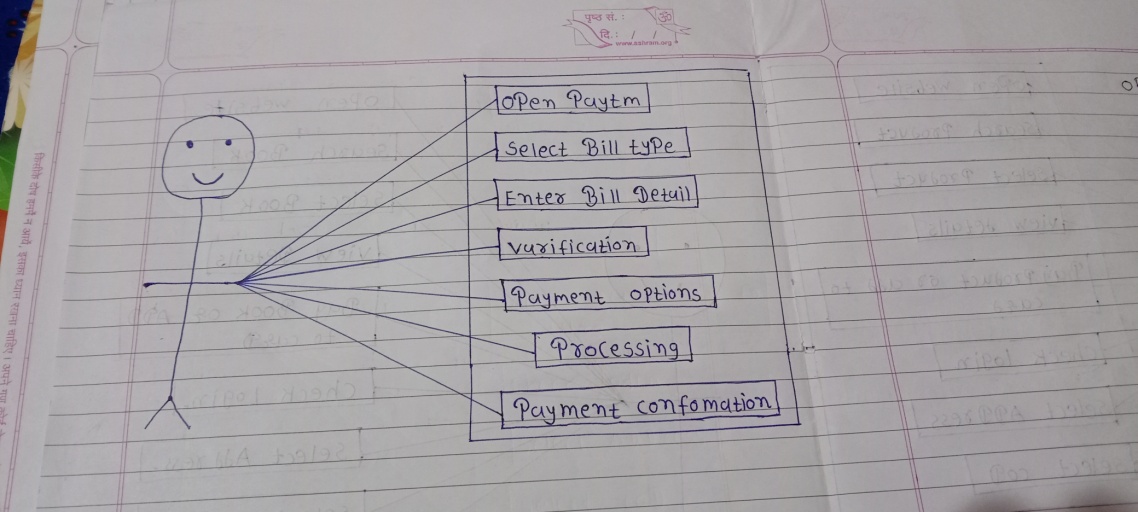
14) What is polymorphism

* Ability to take one name having many forms or multiple forms.

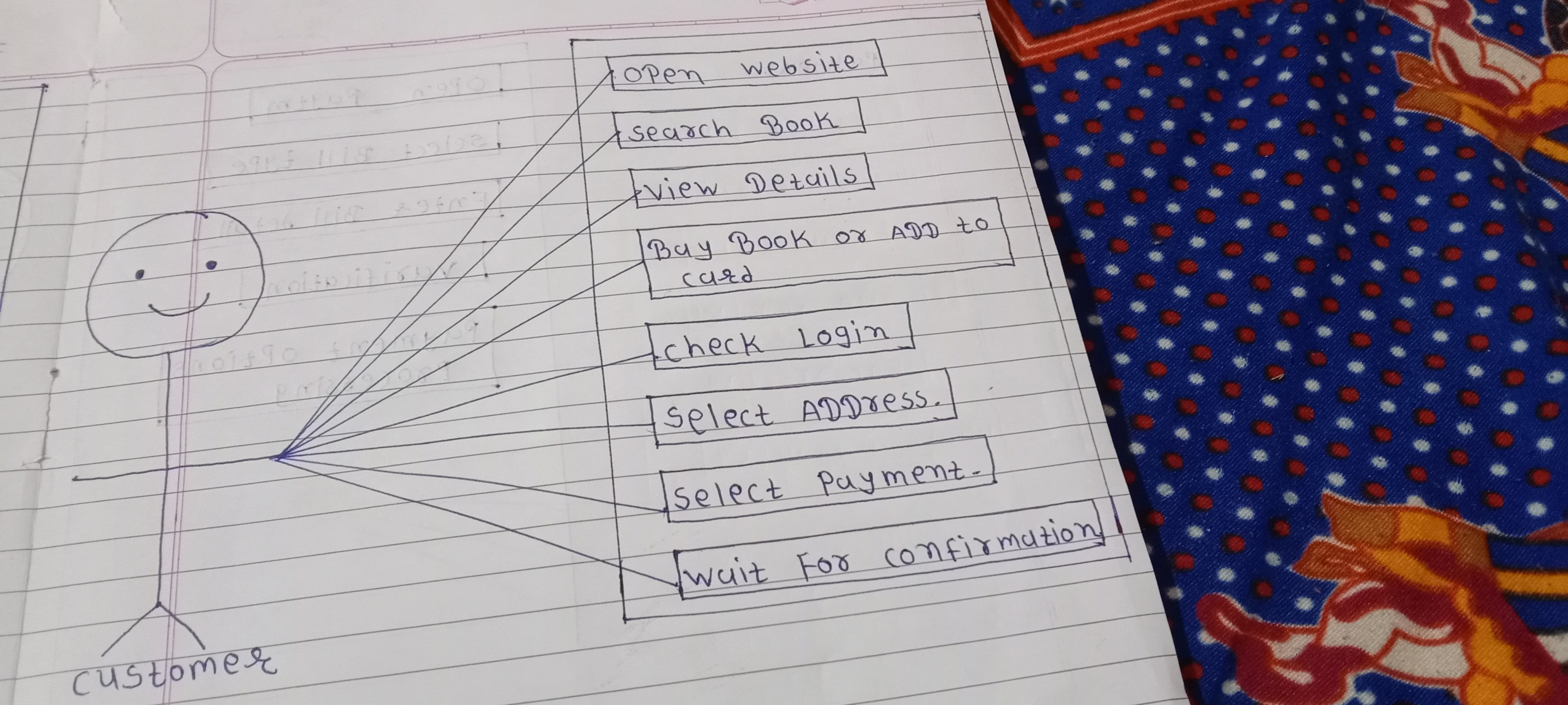
15) Write Basic Concepts of oops

* Oops: object oriented programming systems
* 1) Class: is an collection of data member and member function with its behaviours.
* 2) Objects: Its an instances of an class.
* 3) Encapsulation: data hiding , wrapping up of data into single unit private your data member and member function.
* 4) Inheritance: properties of parent class extends into child class.
* 5) Polymorphism: Ability to take one name having many forms or multiple forms
* 6) Abstraction: Data hiding , only essensial part should be display rest of the will be hide.

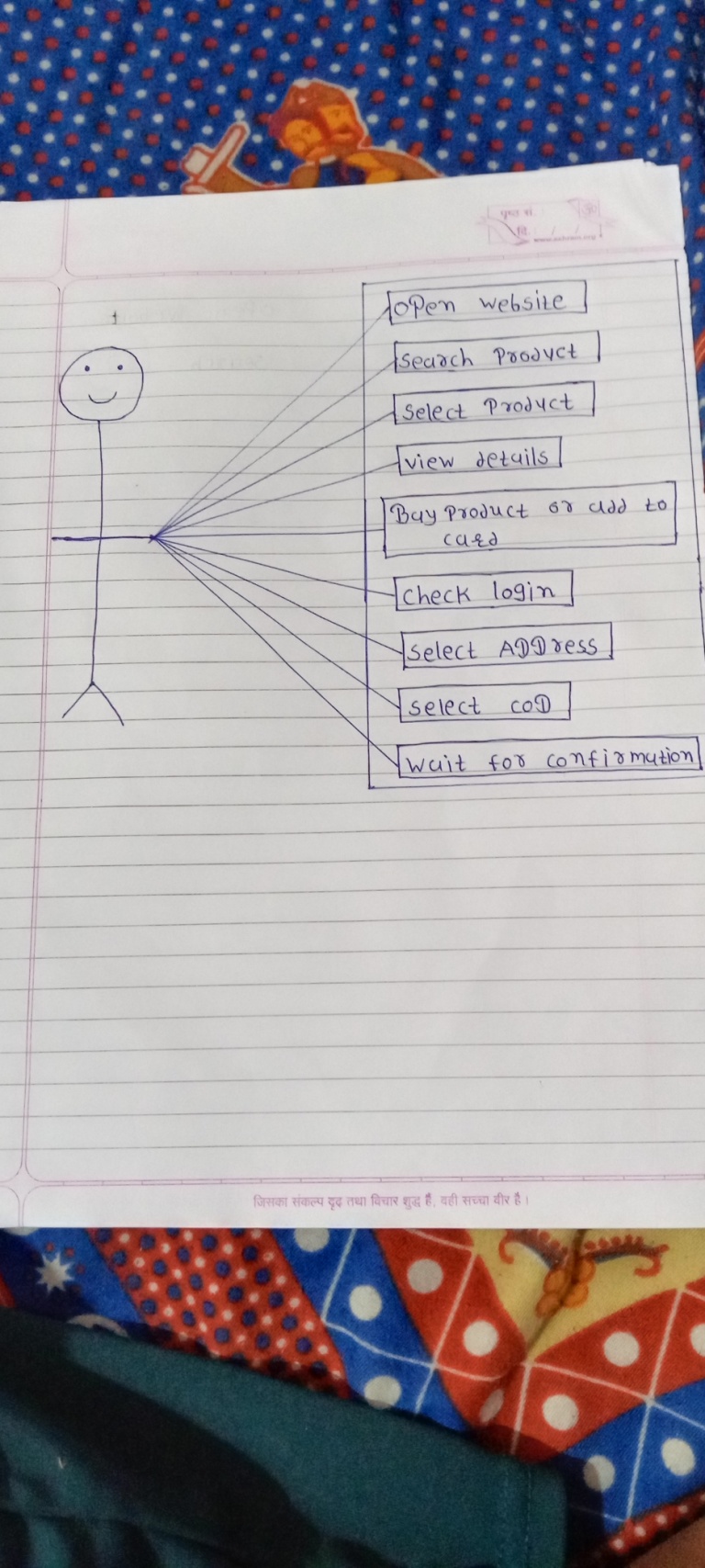
16) Draw Usecase on Online bill payment system



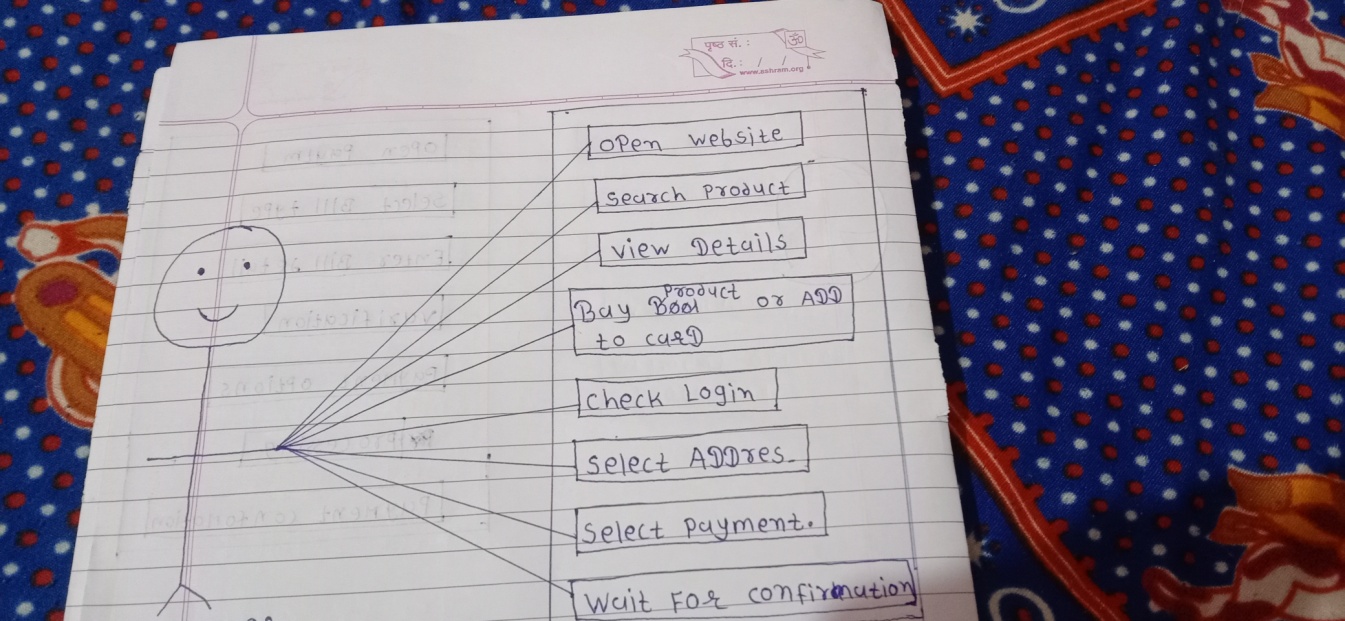
17) Draw Usecase on Online book shopping



18) Draw usecase on Online shopping product using COD



19) Draw usecase on online shopping product using payment gateway.



20) What is class

* Class is an collection of data member and member function with its behaviours.