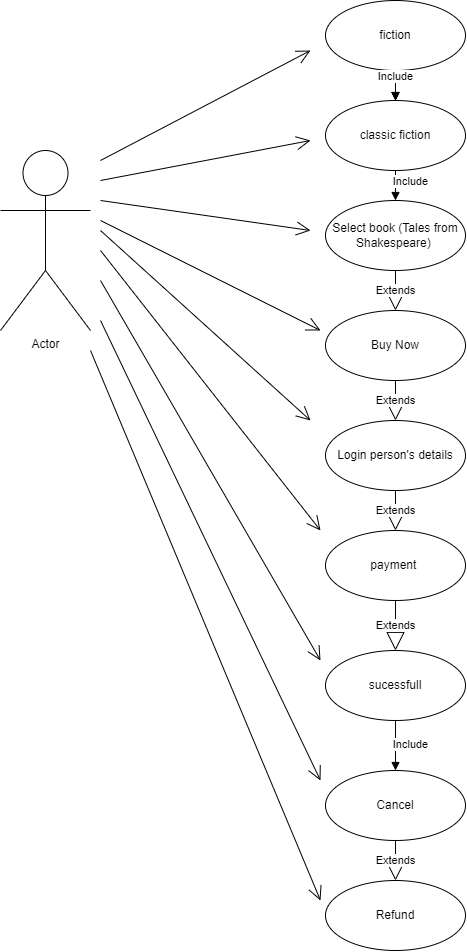
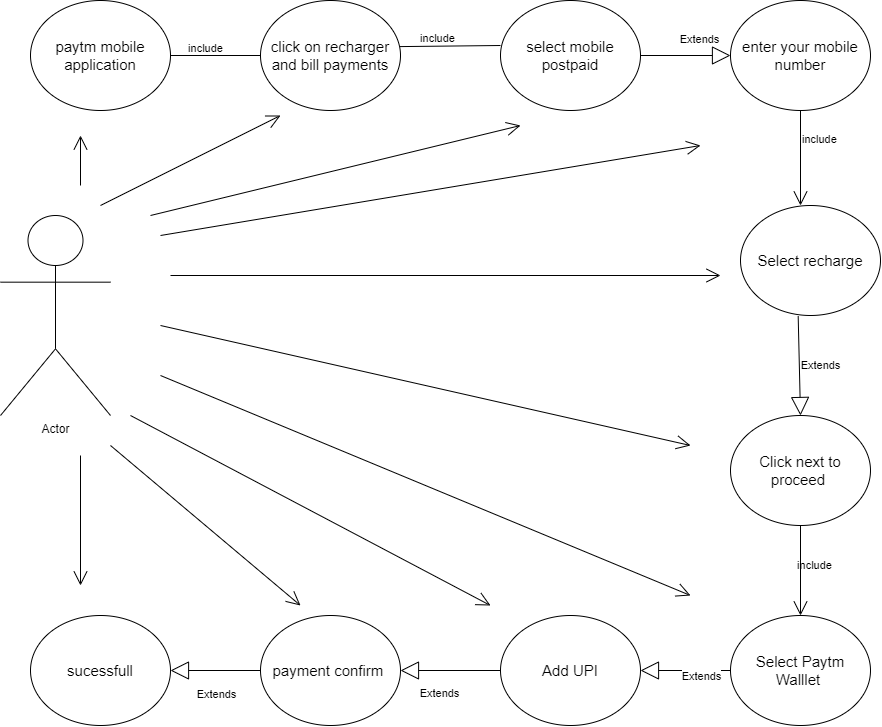
* What is SDLC?
* SDLC means Software Development Life Cycle.
* A Software Development Life Cycle is essentially phases, that provide a model for the development and lifecycle management of an application or piece of software.
* What is software testing?
* Software testing is the process used to identify the correctness, completeness and quality of developed computer software.
* Software testing is a process used to find any existing bugs in the software with the help of executing a program of application.
* What is agile methodology?
* Agile means continuous incremental improvement through small and frequent releases.
* Agile methodology is a combination of iterative and increment model.
* The Agile methodology is a way to manage a project by breaking it up into several phases.
* It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle though a process of planning, executing, and evaluating.
* What is SRS?
* SRS means Software Requirement Specification.
* SRS is a complete description of an application which is to be developed.
* A software requirement specification is a document that describes what the software will do and how it will be expected to perform.
* SRS contains use case diagram that describes all the interaction users will have with the software application.
* It also describes the functionality the product needs to fulfil all stakeholder (Business, users) needs.
* What is OOPS?
* OOP means Object Oriented Programming.
* Object oriented programing is way of writing the program is organized way, objects are like a black box where data are hidden.
* Object oriented programming provides you framework to place the data and the relevant functions together in the same object.
* Write the basic concepts of OOP.
* The basic concept of Object Oriented Programming is:

1. Class
2. Object
3. Inheritance
4. Polymorphism
   * + - Over ridding
       - Over loading
5. Encapsulation
6. Abstraction

* What is Object?
* Everything in the world is an Object.
* functionality Object gives the permission to access of class.
* An object is a class instance that allows programmers to use variables and methods from inside the class.
* Objects represent a real world entity that you’re trying to deal with.
* What is Class?
* Class is a collection of data member and member function.
* A class can be understood as a template or a blueprint, which contains some values, known as behaviours or methods.
* You can declare a class only once.
* Class is a logical entity.
* A class is a way to arrange data and behaviour information. It is a template that must be implemented by its objects.
* What is encapsulation?
* Encapsulation is the practice of including in an object everything it needs hidden from other object.
* The process wrapping the data in a single unit.
* The internal state is usually not accessible by other objects.
* Encapsulation is placing the data and the functions that work on that data in the same place.
* What is Inheritance?
* Making class from existing class.
* Deriving the attribute of some other class.
* What is Polymorphism?
* One name multiple form
* Same name of function with same parameter but definition will be different.
* Draw Use case on Online book shopping



* Draw a Use Case on Online bill payment system(Paytm)



* Write SDLC phases with basic introduction.
* 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.
* The life cycle defines a methodology for improving the quality of software and overall development process.
* SDLC Phases
* Requirements
* Analysis
* Design
* Implementation
* Testing
* Maintenance
* Requirements:
* requirements may be documented in written form, they may be incomplete, or even incorrect.
* Requirements will change

Two types of requirement:

* Functional
* Non function
* Analysis:
* The analysis phase defines the requirements of the system.
* The deliverable result at the end of this phase is a requirement document.
* This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.
* Design:
* The Design team can now expand upon the information established in the requirement document.
* Implementation:
* In the implementation phase, the team also creates the components from scratch or by composition.
* Implementation – Code
* Testing:
* 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.
* Type of Testing:
* Regression Testing
* Internal testing
* Unit testing
* Application testing
* Stress testing
* The testing phase is a separate phase which is performed by a different team after the implementation is completed.
* Maintenance:
* Software maintenance is also one of the phases in the System Development Life Cycle (SDLC), as it applies to software development.
* The maintenance phase is the phase which comes after deployment of the software into the field.
* Types of maintenance:
* Corrective maintenance
* Adaptive maintenance
* Perfective maintenance

* Explain phases of the waterfall model.
* As the name Waterfall model implies this model’s process of downward mechanism is similar to that waterfall.
* The whole process is divided into sequential stages, and it is imperative to complete each phase successfully in order to move onto the next one.
* Waterfall model phases:
* Requirements
* Analysis
* Design
* Implementation
* Testing
* Maintenance
* Requirement
* Requirements must be documented, clear and fixed. they must be “frozen” to early in the life cycle. Requirements are validated too late. There are no ambiguous requirements.
* Analysis
* All requirements of the project are analysed and documented in a specification document and a feasibility analysis is done to check if these requirements are valid.
* Design
* The system design is prepared which specifies hardware and system requirements. It helps define the overall system architecture.
* Implementation
* In this phases the source code is written as per requirements. The system is developed in small program called units, after which units are integrated.
* Testing
* Testers check the program for all possible defects, by running test cases either manually or by automation. The client is involved in the testing phases as well, in order to ensure all requirements are met. All flaws and bugs detected during this phase are fixed to ensure Quality Assurance.
* Maintenance
* The next step is to provide support and maintenance for the software, making sure it runs smoothly. If the client and users come across errors/defects/bugs during use, fixing them is the main purpose of this stage.
* Write phases of spiral model.
* The most important feature is its ability to manage unknown risks after the project has started.
* Bohme’s Spiral model has phases:
* Planning: determination of objectives, alternatives and contraction
* Design: analysis of alternatives and identification/resolution of risks
* Construct: development of the “next level” product
* Evaluation: assessment of the result of construct
* Explain working methodology of agile model and write pros and cons.
* It involves constant collaboration with stakeholders and continuous improvement at every stage.
* Agile model is combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
* The agile methodology is a way to manage a project by breaking it up into several phases.
* These builds are provided in iterations.
* Each iteration typically lasts from about one to three weeks.
* Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
* At the end of the iteration a working product is displayed to the customer and important stakeholders.

Pros

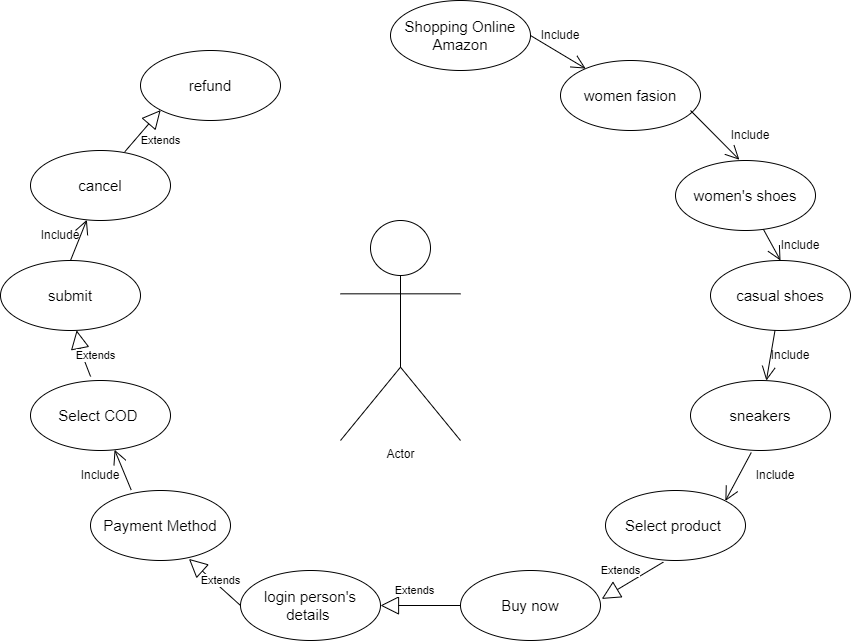
There are many benefits to using an agile methodology strategy.

* Timely delivery
* Adaptability
* Promote teamwork and cross training
* Suitable to fixed or changing requirements
* Gives flexibility to developers
* Ease of collaboration
* Increased performance improvement
* Transparency
* Continuous improvement
* Higher profits
* Less preparatory work

Corns

There are some disadvantages from using the agile methodology style of project management.

* Transfer difficulties
* Not suitable for handling complex dependencies
* More risk of sustainability, maintainability and extensibility
* Variable goals
* Lack of documentation
* Less documented improvement
* Depends heavily on customer interaction
* Goal focus shifting
* Less predictability
* Draw Use Case on Online shopping product using COD.



* Draw Use Case on online shopping product using payment gateway.

