Software Testing Assignment

Module 1(Fundamental)

* What is SDLC

Software development life cycle is a structure imposed on the development of a software product that defines the process.

SDLC is essentially a series of steps, or phases, that provide a model for the development and life cycle management of an application or piece of software.

* What is agile methodology?

Agile SDLC model is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software products.

Each iteration typically lasts from about one of three weeks.

Agile thought process had started early in the software development and started becoming popular with time due toits flexibility and adaptability.

* What is SRS (Software Requirements Specification)

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

Use case are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements.

* What is oops?

Identifying objects and assigning responsibilities to these objects.

Objects communicate to other objects by sending messages.

An object is like a black box.

The internal details are hidden.

Messages are received by the methods of an object.

* Write basic concepts of oops?

- Object

- Class

- Encapsulation

- Inheritance

- Polymorphism

- Overriding

- Overloading

- Abstraction

* What is object

An object is referred as a instance/example for a class.

- An “object” is anything to which a concept applies.

- This is the basic unit of object-oriented programming (OOP)

- That is both data and function that operate on data are bundled as a unit called as object.

**Those two parts of an object**

Object=Data + Methods

* What is class

A class is blueprint for an object.

- A class reparents an abstraction of the object and abstracts the properties and behaviour of that object.

- An object is particular instance of class which has actual existence and there can be many objects (or instances) for a class.

- **Example**

Apple, Banana, Cherry, Mango <<--these are the object for fruit class.

* What is encapsulation

It is the process of wrapping up of the data into a single unit.

- Encapsulation is the practice of including is an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

- Encapsulation enables data hiding, hiding irrelevant information from user of a class and exposing only the relevant details required by the user.

* What is inheritance

Inheritance means that one class inherits the characteristics of another class, This is also called a “is a” relationship.

- One of the most useful aspects of object-oriented programming code is reusability.

- This is very important concept of object-oriented programming since this features help to reduce the code size.

* What is polymorphism

Polymorphism means “having many forms”

- It allows different object to respond to the same message in different ways, the response specific to the type of the object.

- The ability to change from is known as polymorphism.

- There are two types of polymorphism in Java

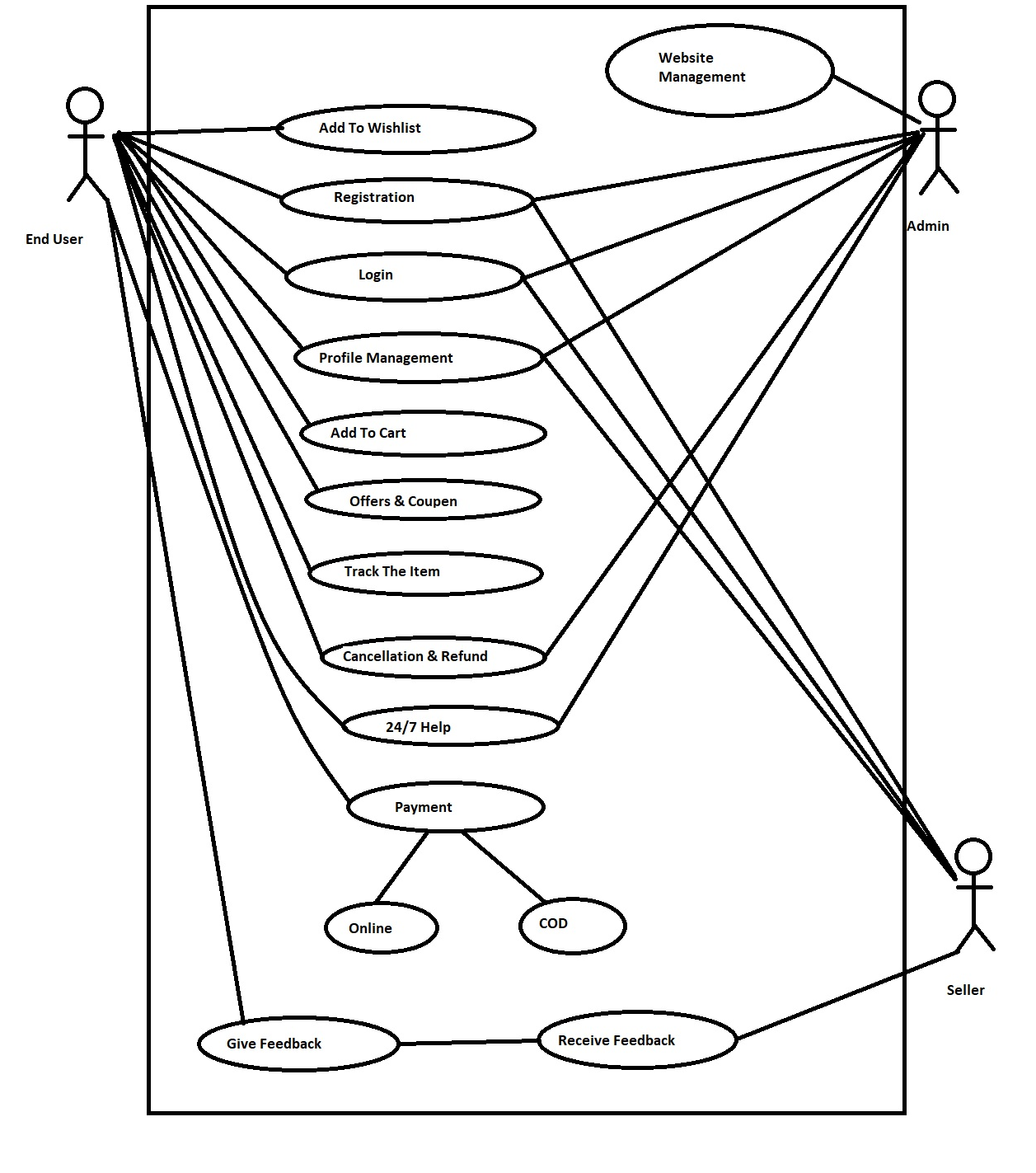
1.Compile time polymorphism (Overloading)

2.Runtime polymorphism (Overriding)

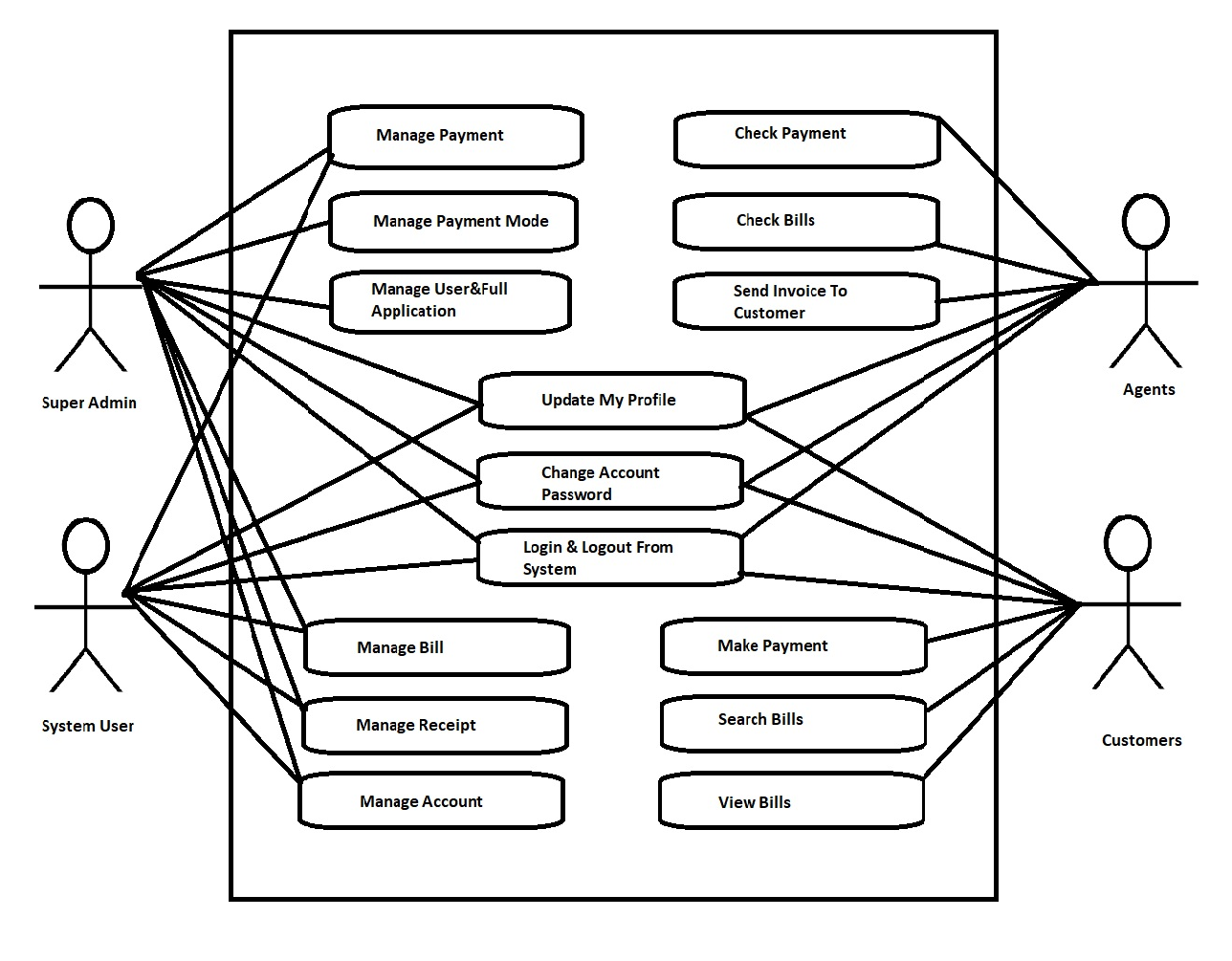
Overloading-method name same working different but within a single class.

Overriding-method name same working also same, but both are in different class here both class are in parent child relation.

* What is RDBMS- Topic pending in class
* What is SQL- Topic pending in class
* Write SQL commands- Topic pending in class
* Draw use case on online book shopping



* Draw 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. These are a number of different development models.

**SDLC Phases**

1.Requirements collection / Gathering

- Establish customer needs

- Features

- Usage scenarios

- user and business needs change during the

Project.

* Functional requirements is describe system

service or function.

* Non functional requirements are

constraints on the system or the

development process.

* Non functional requirements may be more

critical than functional requirements.

2.Analysis Phase

- The analysis phase defines the

requirements of the system, independent

of how these requirements will be

accomplished.

* Model and specify the requirements.

“What"

3.Design

- Design architecture document

- Implementation plan

- Critical priority analysis

- Modal and specify a solution.

“Why”

4.Implementation

- In the implementation phase, the team

builds the components either from scratch

or by composition.

* Construct a solution in software.

5.Testing

- Quality is a distinguishing attribute of a

including the degree of excellence.

* A customer satisfied with quality of a

product will remain loyal and wait for new

functionality in the next version.

- Regression testing

- Internal testing

- Unit testing

- Application testing

- Stress testing

- Validate the solution against the

requirements.

6.Maintenance

- Repair defects and adapts the solution to

the new requirements.

* Software maintenance is one of the

activities in software engineering, and is

the process of enhancing and optimizing

deployed software , as well as fixing defects.

* Explain phases of the waterfall modal

The waterfall is unrealistic for many reasons, especially;

-Requirements must be “frozen” to early in

the life cycle.

-Requirements are validated too late.

**Waterfall model phases**

1.Requirements- The first phase involves understanding what needs to design and what is its function, purpose ,etc.

2.System design- The requirements specifications from the first phase are studied in this phase and system design is prepared.

3.Implementation- With inputs from system design, the system is first developed in small programs called unit, which are integrated into the next phase.

4.Integration and testing- All the units developed in the implementation phase are integrated into system after testing of each unit.

5.Deployment of system- Once the functional and non-functional testing is done , the product is deployed in the customer environment or released into the market.

6.Maintenance- This step occurs after installation and involves making modifications to the system or an individual component to alter attributes or improve performance.

* Write phases of spiral model

Spiral model is very widely used in the software industry as it is in synch with the natural development process of any product

i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms.

**Spiral model phase**

1.Planning

2.Risk analysis

3.Engineering

4.Customer evaluation

* Write agile manifesto principles

**Individuals and interactions-** in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

**Working software**-Demo working software is considered the best means of communication with the customer to understand their requirements, instead of just depending on documentation.

**Customer collaboration-**As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

**Responding to change-** agile development is focused on quick response to change and continuous development.

* What is join? - Topic pending in class
* Write types of joins- Topic pending in class
* 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.

Agile modals break the product into small incremental builds.

Each iteration typically last from about one of three weeks.

**Pros**

- Is as very realistic approach to software development.

- Promotes teamwork and cross training.

- Functionality can be developed rapidly and demonstrated.

- Good model for environments that change steadily.

- Little or no planning are required.

- easy to manage.

- Gives flexibility to developers.

**Cons**

- Not suitable for handling complex dependencies.

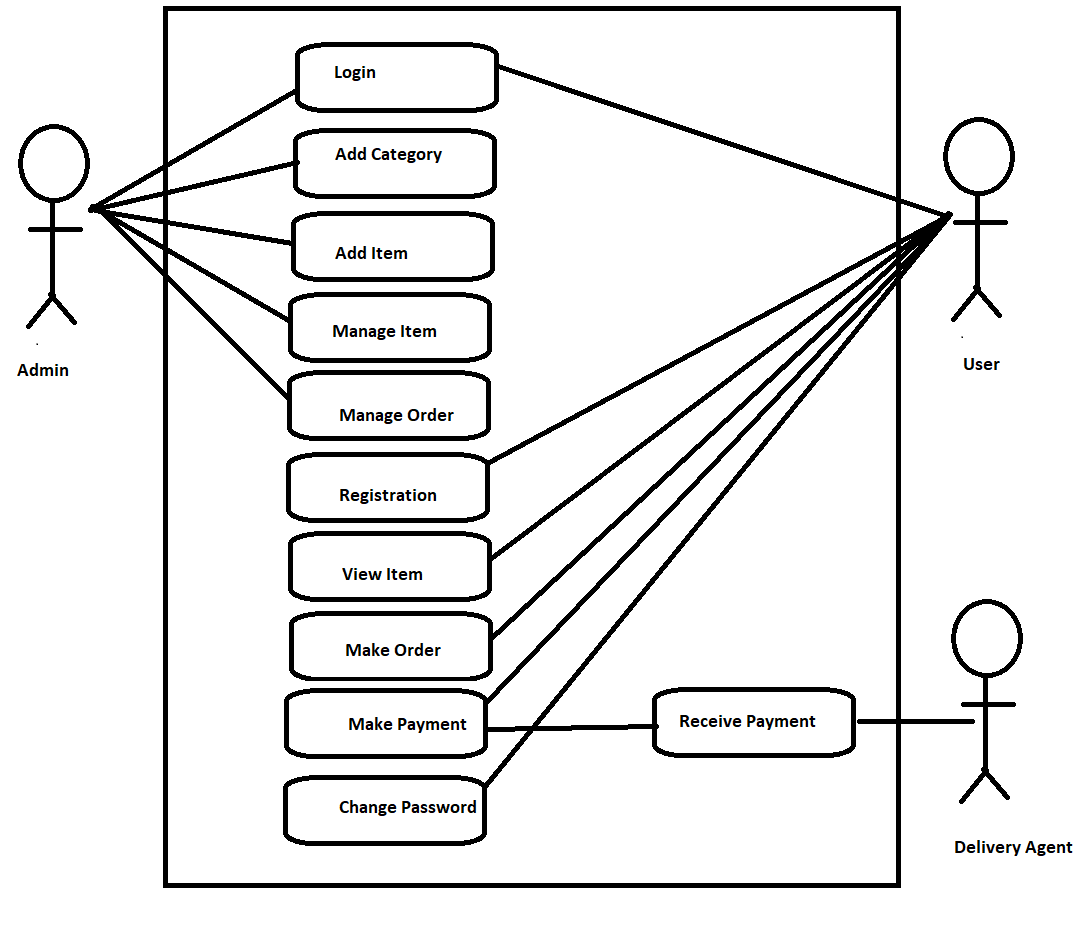
- More risk of sustainability, maintainability, and extensibility.

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

- There is very high individual dependency, since there is minimum documentation generated.

- Transfer to technology to new team members may be quite challenging due to lack of.

* Draw use case on online shopping product using COD



* Draw use case on online shopping product using Payment gateway

