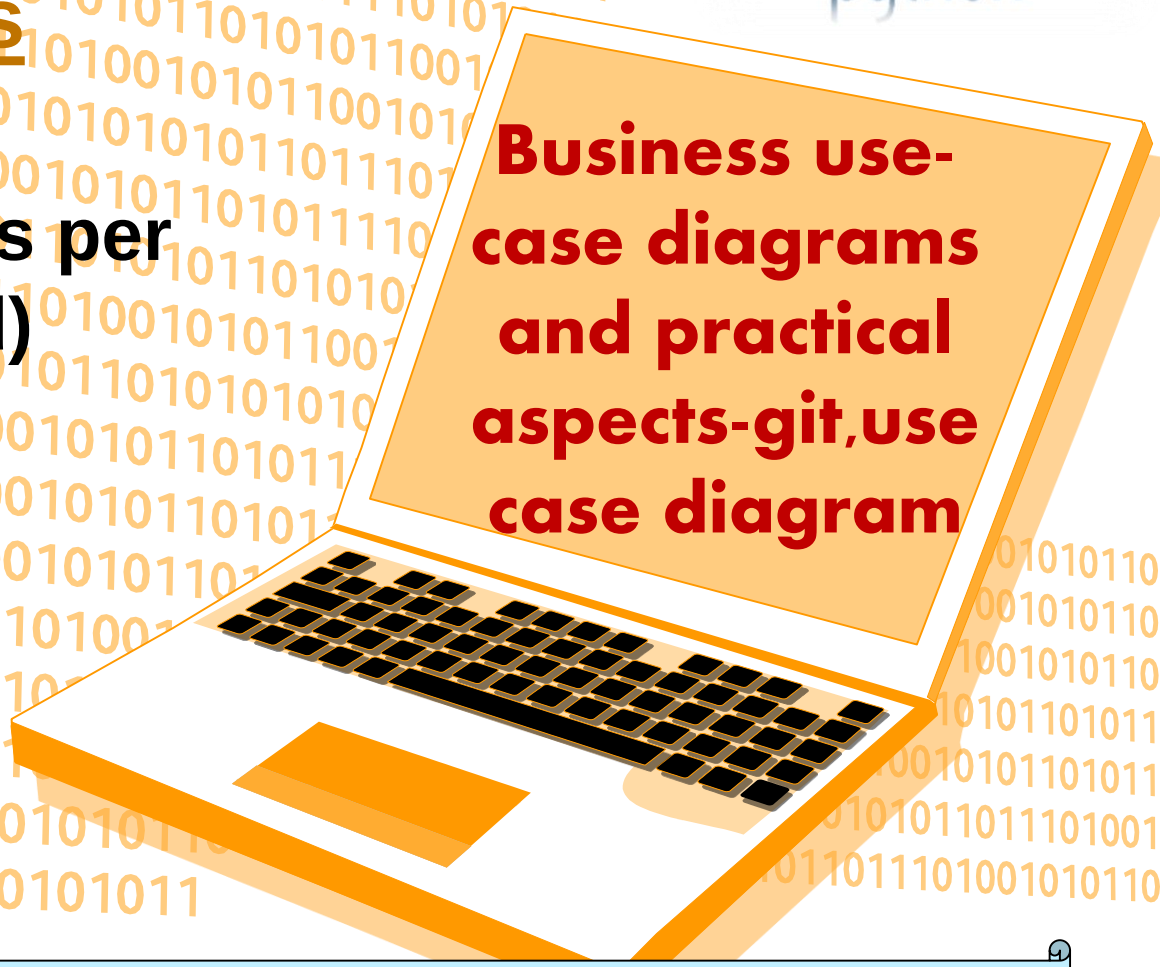
Chapter 9 :

Informatics Practices

Class XII (As per CBSE Board)



New Syllabus 2018-19

Visit : python.mykvs.in for regular updates

Business use-case diagrams

Business use case diagram describe how the business is being used by its customers. Activities that directly concern the customer.

A use case diagram is a representation of a user's interaction with the system where relationship is shown between the user and the different use cases in which the user is involved.

In UML standard, both business use case as well as business actor are not defined, so either we need to use some UML tool supporting those or create our own business modeling stereotypes.

Business use-case diagrams

What is a Use Case?

- Means ,how a business system interacts with its environment.
- Represents the activities that are performed by the users of the system.

What is an Actor?

It is a user or outside system which interacts with the system being designed in order to obtain some result from that interaction.

Business use-case diagrams

Use case diagrams shows what a system does from the view of an external observer. The main focus is on what a system does rather than how it does.

Use case diagrams are connected to scenarios. A **scenario** means what happens when anyone interacts with the system.

Here is a scenario for a point of sale terminal.

A customer reach to the shop to purchase an item.

The cashier/seller take the decision based on the item chosen by the customer and deliver that item receive money.

We want to write a use case for this scenario.

Step 1 Identify the actors

As we read the scenario, define those people or systems that are going to interact with the scenario.

Business use-case diagrams

Categories of Business Use Cases

Based on the activities in a business there are three categories of business use cases:

- **First, activities which are commercially important, known as business processes.**
- **Second, activities which are not commercially important, but to be performed anyhow to make the business work. E.g. security, administration, cleaning etc.**
- **Third, management work. Type of work that affects business use cases management.**

Business use-case diagrams

Business Use Cases Always Related to Business Actors-

Every business use case should have a relationship to or from a business actor. If business use-case model has business use cases which no one requests or use, it warns that something is wrong with the model.

There are three main reasons for structuring the business use-case model:

- **To make it easier to understand.**
- **To reuse parts of workflows which are shared among many business use cases.**
- **To make easier to maintain.**

Business use-case diagrams

Characteristics of a Good Business Use-Case Model

- It must perform all the activities within the business.
- It must conform to the business for which these are designed.
- Every activity must be relate to at least one use case.
- Should have balance between the number of use cases and the size of the use cases:
- Less use cases make the model easier to understand where as More use cases may make the model difficult to understand.
- Large use cases may be complex and difficult to understand.
- Each use case must be unique.
- It should give a good comprehensive picture of the organization.

Business use-case diagrams

Elements in use case diagrams

Actor

An actor represents a role that an outsider takes on when interacting with the business system.

Association

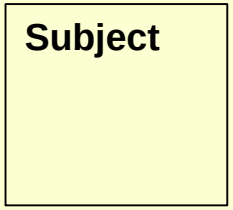
It is relationship between an actor and a business use case

Business Use Case

It describes the interaction between an actor and a business system

Subject

It describes a business system that has one or more business use cases attached to it.



Business use-case diagrams

How to Draw a Use Case Diagram?

Following are the steps to draw use case diagram.

- Identify the Actors of the system.
- Identify all roles played by the users relevant to the system.
- Identify the users/task required to achieve the goals.
- Create use cases for every goal.
- Structure the use cases.
- Prioritize/review/estimate/validate the users.

Business use-case diagrams

Structuring/relationship type of Use Cases

- **<<include>> Use Case**

The time to use the <<include>> relationship is after you have completed the first cut description of all your main Use Cases

- **<<extend>> Use Case**

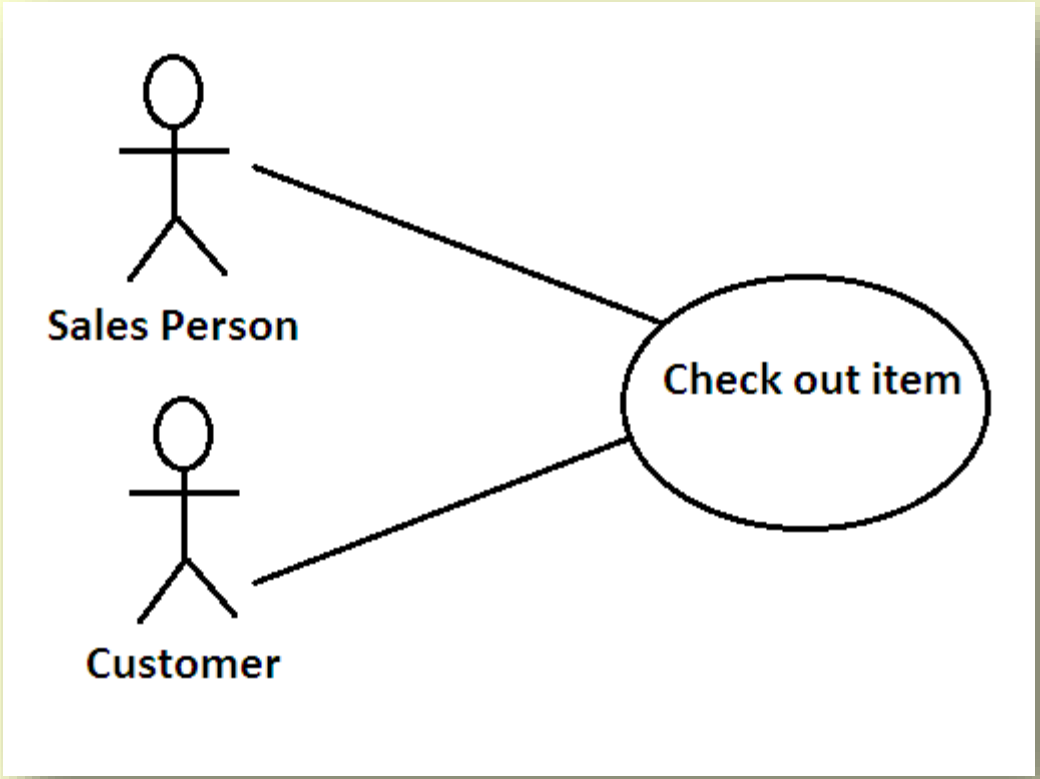
The <<extend>> use case accomplishes this by conceptually inserting additional action sequences into the base use-case sequence.

- **Abstract and generalized Use Case**

The general use case is abstract. It can not be instantiated, as it contains incomplete information. The title of an abstract use case is shown in italics.

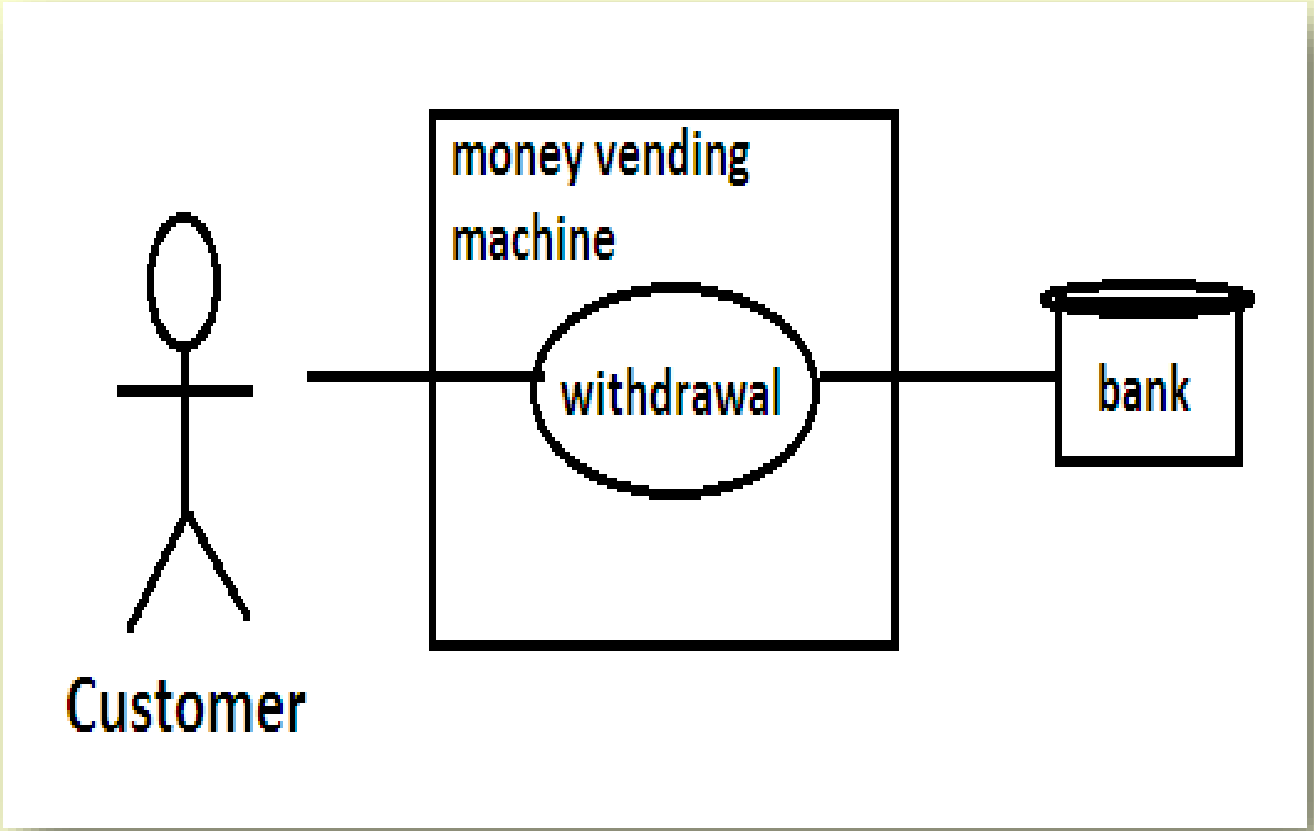
Business use-case diagrams

E.g. Use case diagram



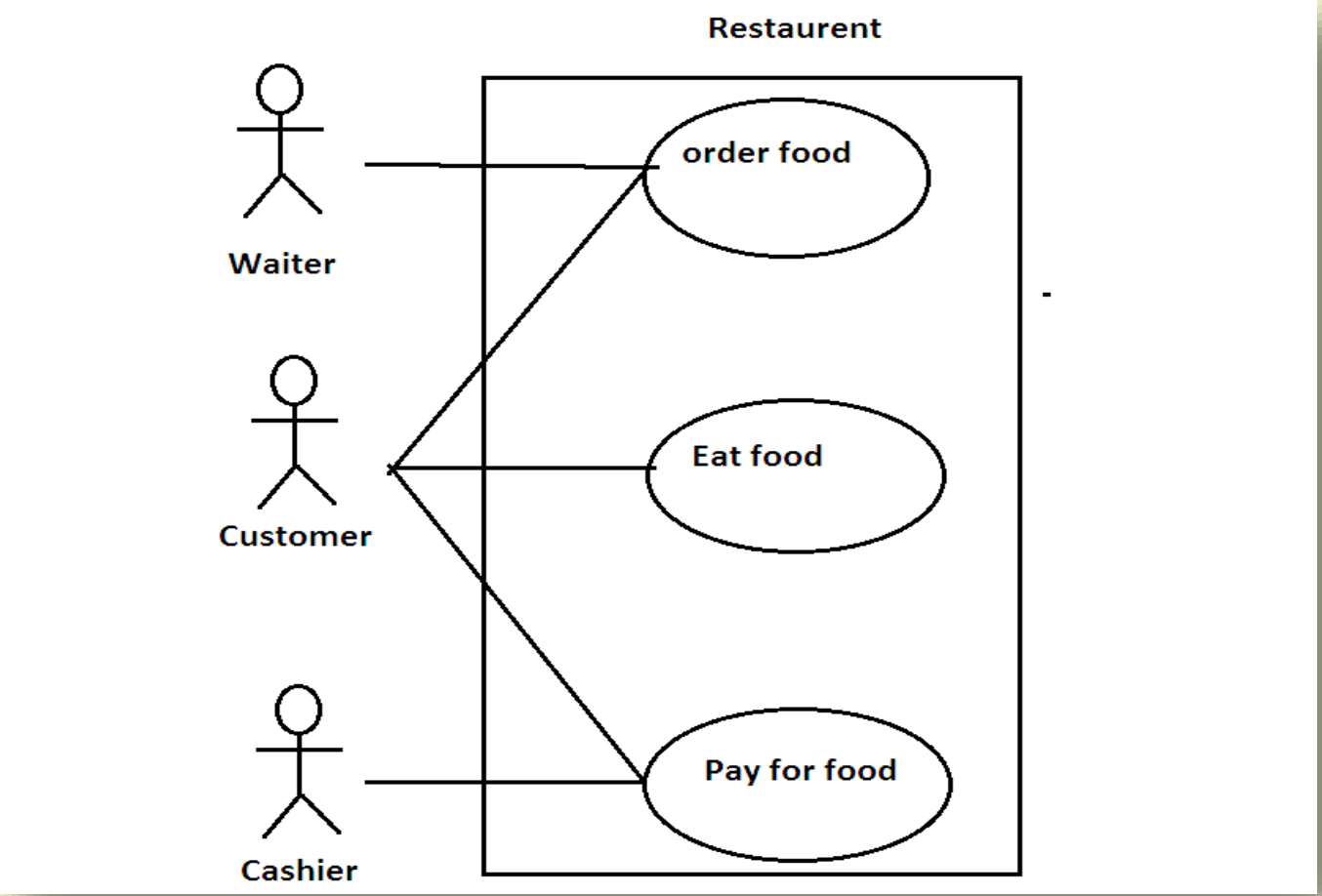
Business use-case diagrams

E.g. Use case diagram



Business use-case diagrams

E.g. business use case diagram



Business use-case diagrams

Version control system

Version Control System (VCS) can be considered as a kind of database. It helps us to save a snapshot of the complete project at any point of time. Through it project files can be tracked along with who made the change and why the changes were made. Later on when if required to take a look at an older snapshot/version, VCS shows how exactly it differed from the previous one.

When our project is tracked by VCS, any addition/deletion/modification in files of our project will be automatically detected and recorded by it.

Version Control System also know as:

- Source Control Management System
- Revision Control System
- Configuration Management System

Business use-case diagrams

Features of VCS

- Maintain separate track record for each team-members of the project.
- Easy to compare and merge codes of different branches.
- Easy to trace changes in code to find the version that introduced a bug
- Simple to compare versions to resolve conflicts in code during merging
- Revert changes made to code to any state from its history.

Business use-case diagrams

Top Version Control Systems

- GIT
- CVS
- SVN
- Assembla
- Mercurial
- Bazaar

Global Information Tracker

GIT

Git is currently the most popular distributed version control system.

It originates from the Linux kernel development and it was founded in 2005 by Linus Torvalds. Nowadays it is being used by many popular open source projects, like,Android,Eclipse developer teams, as well as many commercial organizations.

Basically it was written in the programming language C, but later on Git has been re-implemented in other languages, e.g., Java, Ruby and Python.

Global Information Tracker

Advantages of using GIT

- ***Performance**:Git has good performance among other VCS. Committing, branching, merging all are optimized for a better performance than other systems.*
- ***Security**: It secure our codes.Git handles security with cryptographic method SHA-1.*
- ***Branching Model**: we can have multiple local branches which are independent of each other. So there is less friction, **context switching** (switch back and forth to new commit, code and back)*
- ***Staging Area**: Git has an intermediate stage called as "index" or "staging area" where commits can be formatted and modified before completing the commit.*
- ***Distributed**: Distributed means that the repository or the complete code base is mirrored onto the developers system so that he can work on it only.*
- ***Open Source:** Being open source invites the developers from all over the world to contribute to the software and make it more powerful*

Disadvantages of using GIT

- *Git is less preferred for handling extremely large files or frequently changing binary files .*
- *GIT does not support 'commits' across multiple branches or tags.*

Global Information Tracker

Git terminology

- **Branch – A named pointer to a commit.**
- **Commit - Creates a new commit object in the Git repository.**
- **HEAD - Pointing to the existing checked out branch (selected branch)**
- **Index - Alternative term for the staging area**
- **Repository – It contains the history, the different versions over time and all different branches and tags.**
- **Revision - A version of the source code**
- **Staging area - Place to store changes in the working tree before the commit**
- **Tag-Points to a commit that uniquely identifies a version of repository**
- **URL-Location of the repository.fetchurl for fetch data from other repository and pushurl to push data to other repository**
- **Working tree - Contains the set of working files for the repository**

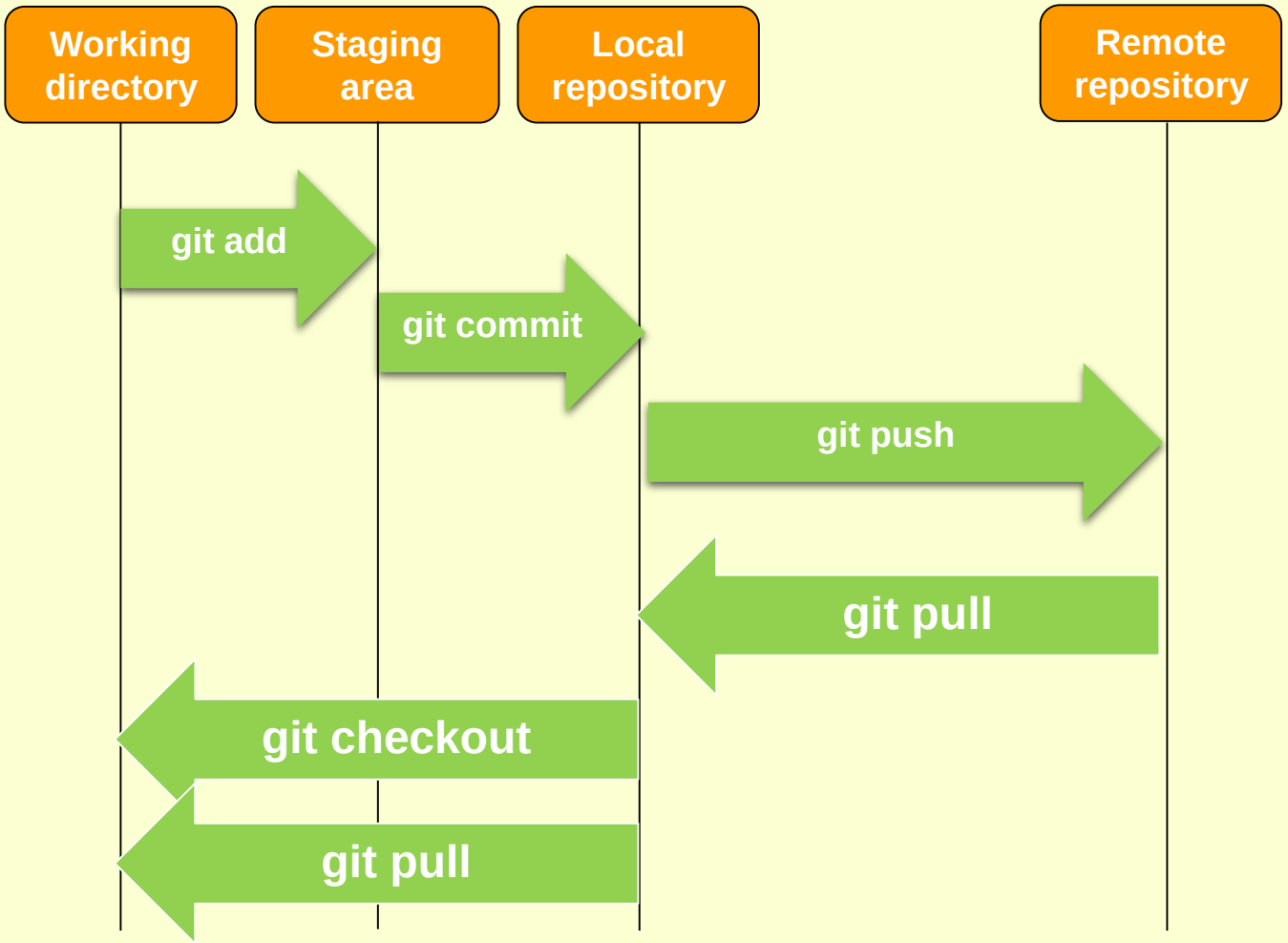
Global Information Tracker

Install git for Windows

- Download git from https://gitforwindows.org/
- Double click the downloaded file to install git
- Accept all the defaults

Global Information Tracker

How to work in git/working structure of git



Global Information Tracker

Some of the basic operations in Git are:

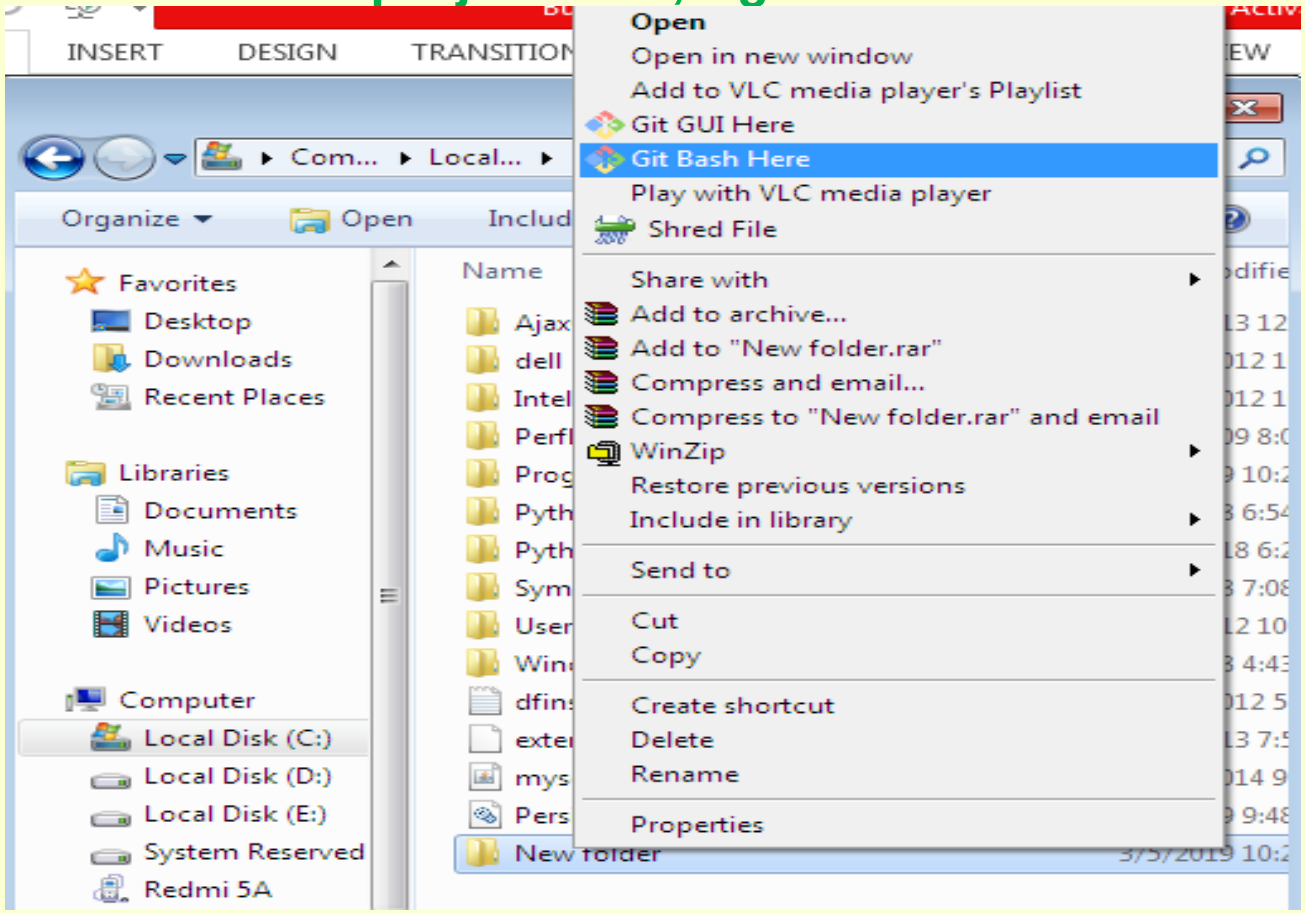
- Initialize -
- Add
- Commit
- Pull
- Push

Some advanced Git operations are:

- Branching
- Merging
- Rebasing

Global Information Tracker

After installing Git in Windows system, just open folder/directory where we want to store all our project files; right click and select 'Git Bash here'.



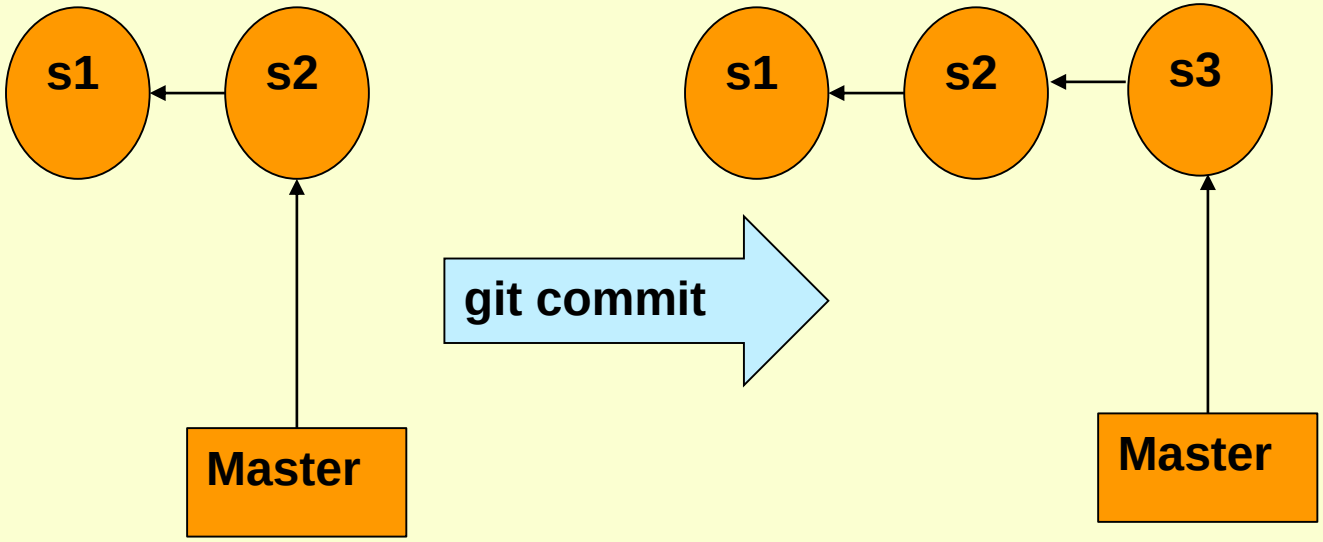
It will open up Git Bash terminal where we can enter commands to perform various Git operations.

Global Information Tracker

Initialize - git init command creates an empty Git repository/re-initializes an existing one. It creates a .git directory with sub directories and template files.
Type **git init** in git bash of selected directory.
git status command lists all the modified files which are ready to be added to the local repository.
Now type **git status** , it will display the existing status.
Now make some changes in the selected directory ,like create a.txt file in it.
Now again type **git status**, it will display that untracked files are there like a.txt.
git add command to add any new or modified files to the index. **(git add –A** to add all the files of current directory**)**
Now type git add a.txt and then git status, it will display changes occurred.

Global Information Tracker

git commit - It refers to recording snapshots of the repository at a given time. git commit always changes master points to latest commit.



In above diagram after git commit command master point will move to s3 snapshot from s2 snapshot.

Global Information Tracker

git commit - before using git commit command please configure git system first.

like

$ git config --global user.email “abc199@gmail.com"

$ git config --global user.email “abc“

Now apply git commit command like

git commit -m "adding one file"

The above command will commit the changes in one file in the local repository.

Before using git pull or push we have to set the central repository using command like given below.

$ git remote add origin https://abc.com/a.git

Now type the following command

$ git push origin master

Above command will transfer commits from our local repository to remote repository.

$ git pull origin master

It is opposite to git pull command.it will transfer commits of remote repository to local repository.

Note – GUI Version of git can also be used

Global Information Tracker

Branching

Branches in Git are pointers to a specific commit. Git generally prefers to keep its branches as lightweight as possible.

There are basically two types of branches viz. local branches and remote tracking branches.

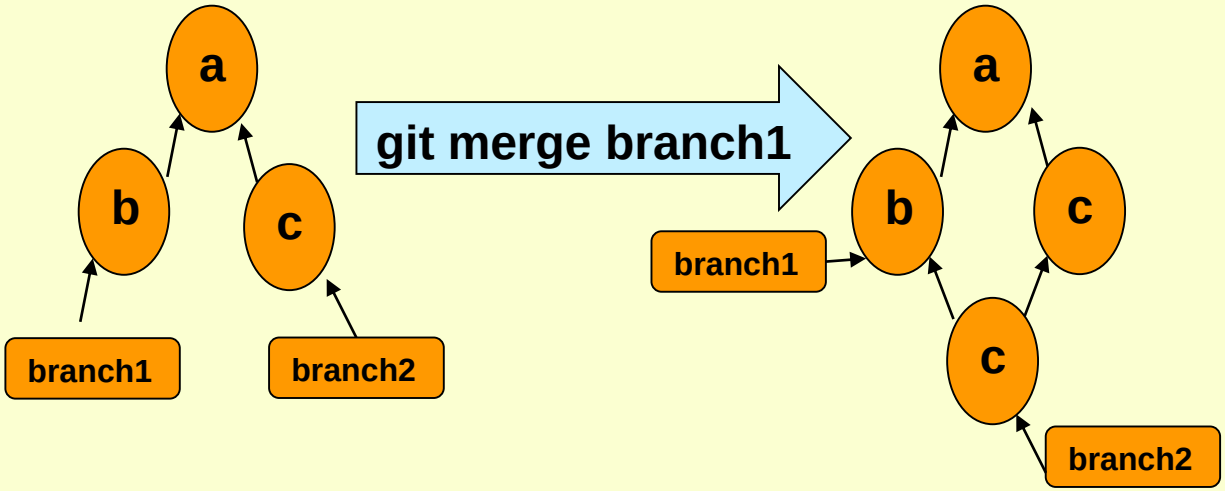
$ git branch myfirstbranch

The above command creates new branch named “myfirstbranch” and switched to new branch using below command.

$ git checkout myfirstbranch

Merging

Merging is the way through which we can combine the work of different branches together.



It is important to know that the branch name in the git merge command should be the branch we want to merge into the branch we are currently checking out. So, make sure that we are checked out in the destination branch.

Use case diagram (as case study of a software system)

A use case diagram is a representation of a user's interaction with the system where relationship is shown between the user and the different use cases in which the user is involved.

How to build a Use Case Diagram?

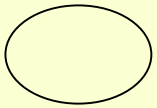
Following are the steps to draw use case diagram.

- Identify the Actors of the system.
- Identify all roles played by the users relevant to the system.
- Identify the users/task required to achieve the goals.
- Create use cases for every goal.
- Structure the use cases.
- Prioritize/review/estimate/validate the users.

Use case diagram (as case study of a software system)

Components of a Use Case Diagram?

Connector – Role play(system)
real system



use case – Capability

Connector – Interaction

Generalization – further parts of connector

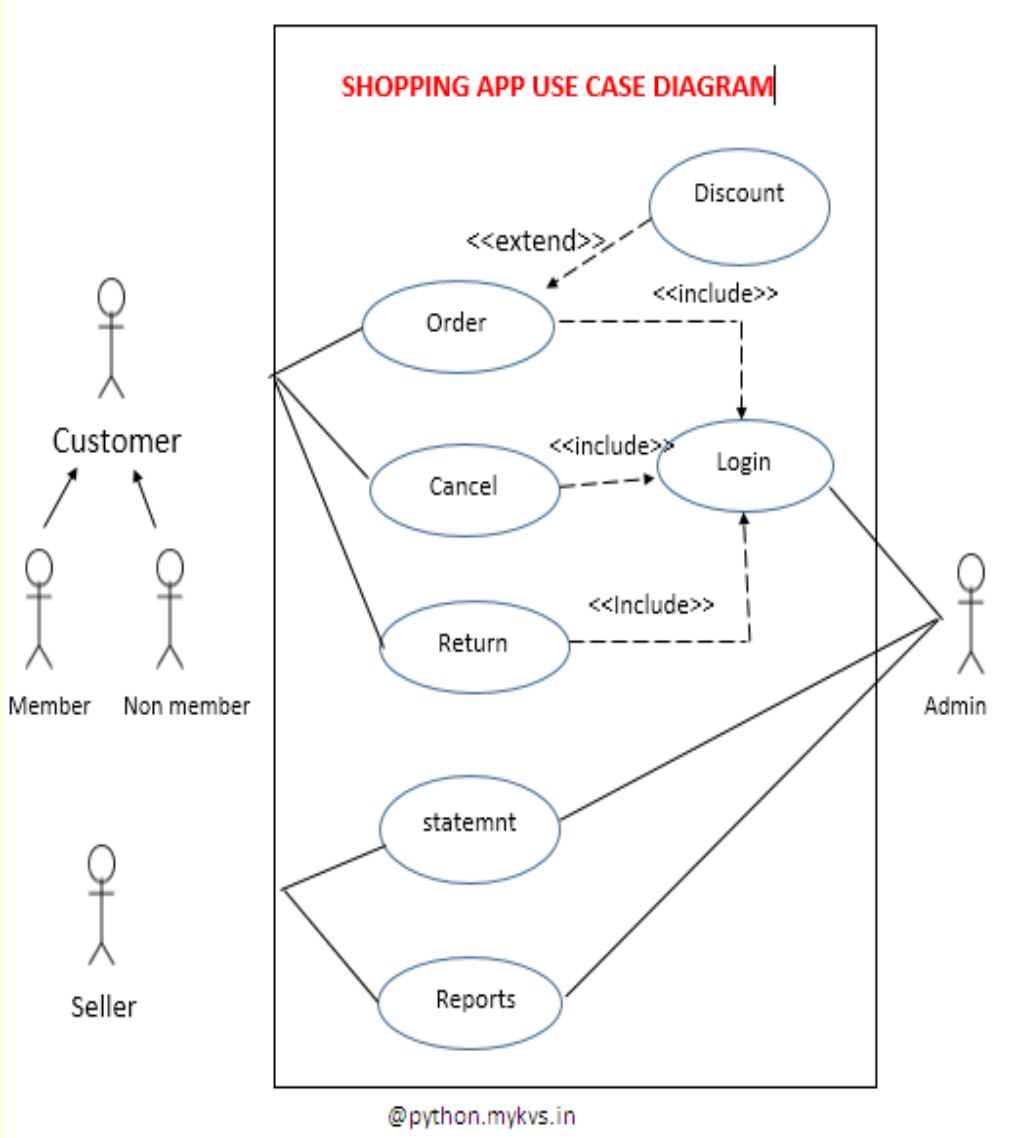
Stereo type – relationship

a. <<include>> It is implicit function ,to use any use case/capability we have to go through it.

b. <<extend>> It is explicit function and it is optional. means some capability to be used are optional

Use case diagram (as case study of a software system)

1. Shopping app use case diagram



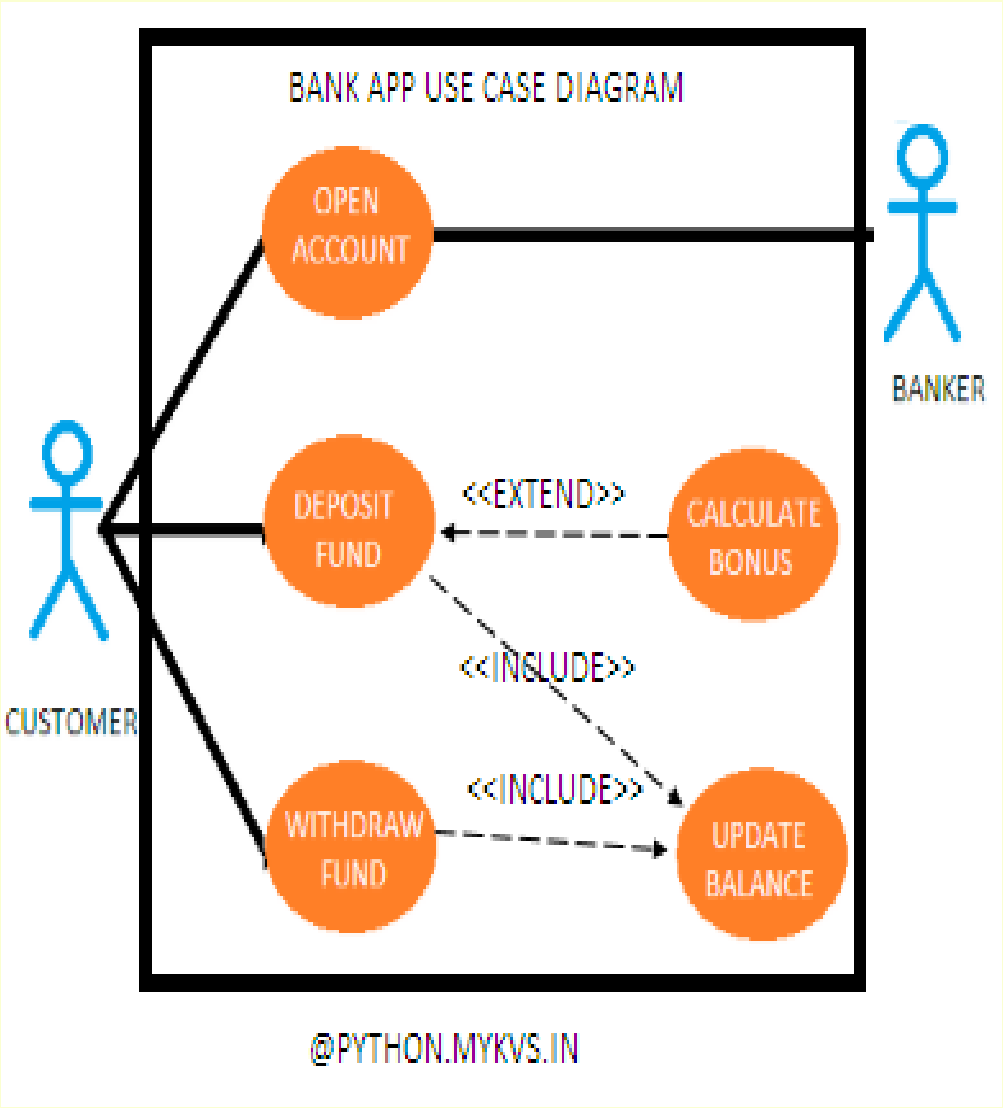
Steps

1. First draw actors on left side like customer(then generalized with member and non member) and seller
2. Then draw use cases like order,cancel,return for customer and stmt and reports for seller.
3. Assign connector(———) between actor and use cases
4. Make use case login, because order,cancel or return is possible after login that's why it is necessary (so mark it as include)
5. Draw discount as extended as it is optional.
6. Draw admin as actor and relate with login,stmt and reports.
7. Assign system boundary and name it as shopping app use case diagram

Note – design of use case diagram designed by two or more person may vary, because creativity of different person varies

Use case diagram (as case study of a software system)

2. Banking app use case diagram



Steps

1. First draw customer as actor
2. Then draw use cases open accunt,deposit fund,withdraw fund
3. Then connect use cases with customer
4. Then draw use case update bonus as <<include>> (because it necessary) from deposit fund and withdraw fund use cases and calculate bonus as optional (<<extend>>)
5. Then draw banker and connect it with open account(because for opening an account there must be banker
6. Then assign system boundary and assign name as bank app use case diagram

Note – design of use case diagram designed by two or more person may vary, because creativity of different person varies

Use case diagram (as case study of a software system)

There are many online website for use case design

- Creately :Online diagramming and collaboration. Several modeling languages supported
- Diagram.ly : Simpe and easy
- Draw.io
- Gliffy | Online Diagram and Flowchart Software
- Lucidchart
- Create UML diagrams online
- MxGraph (JavaScript based)

Other than above mentioned ,there are number of software/tools available on internet(can find through google) for use case design purpose.

Through little bit effort any one can draw use case design tool/ software.