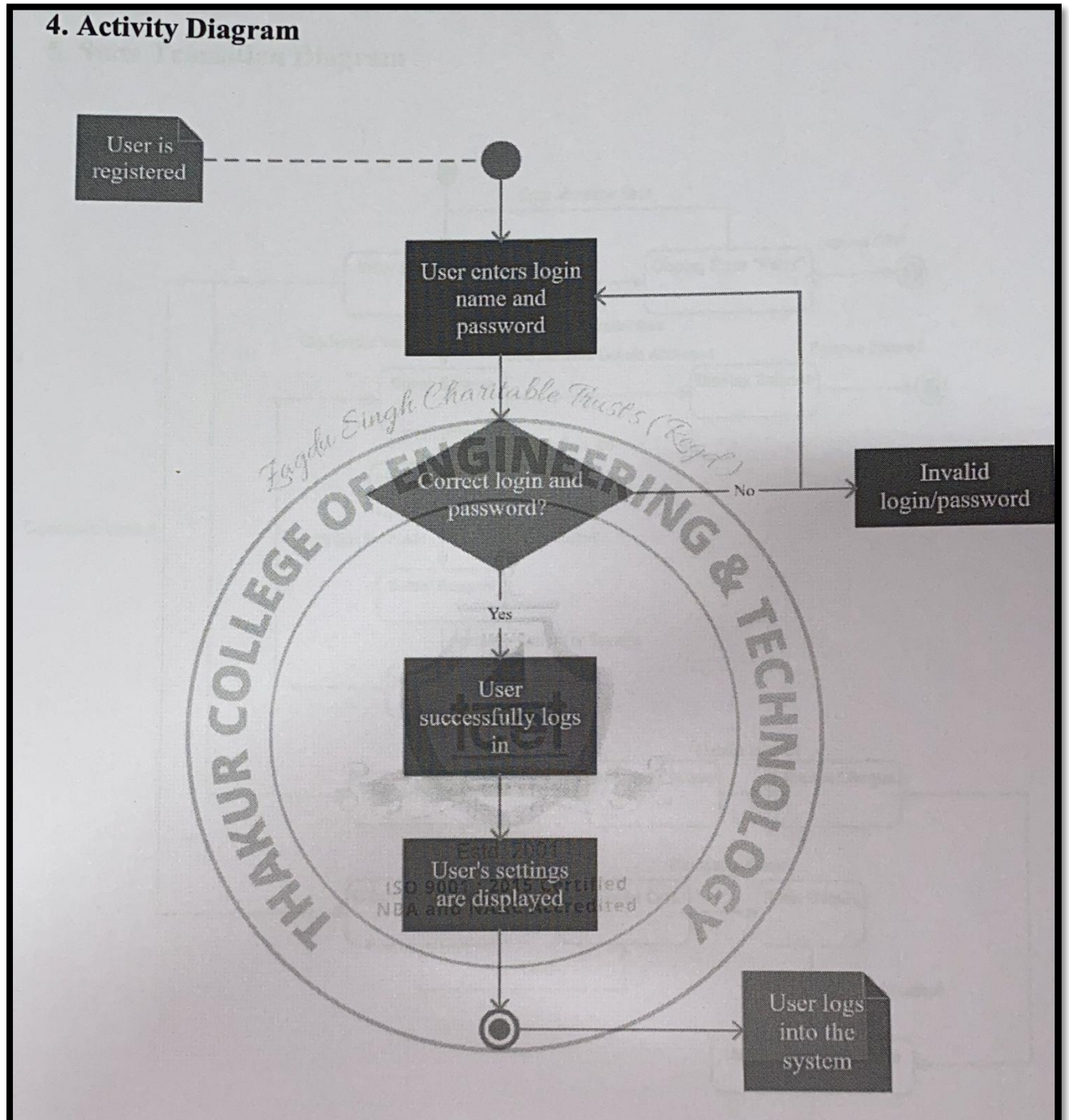


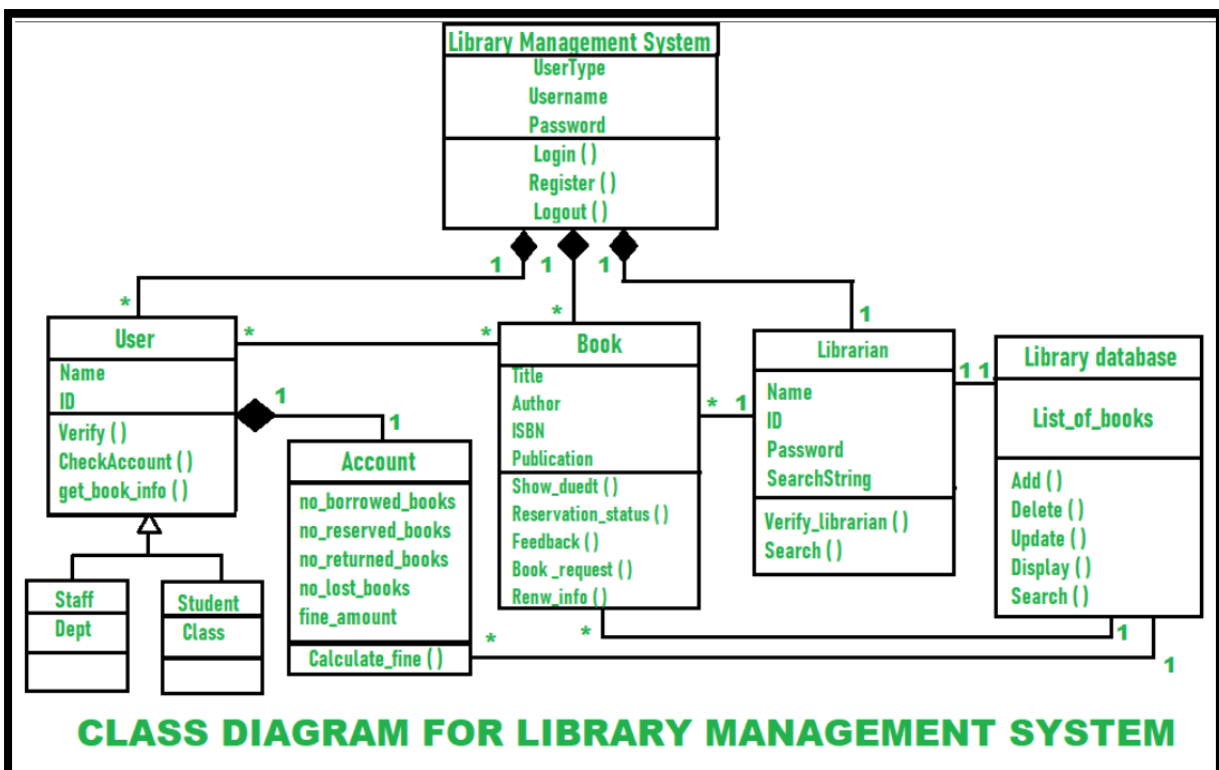
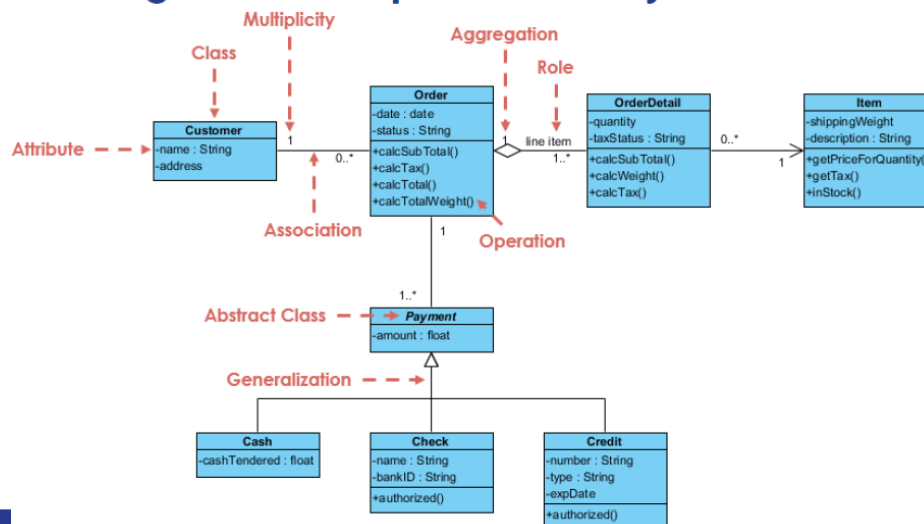
SE

- Activity Diagram

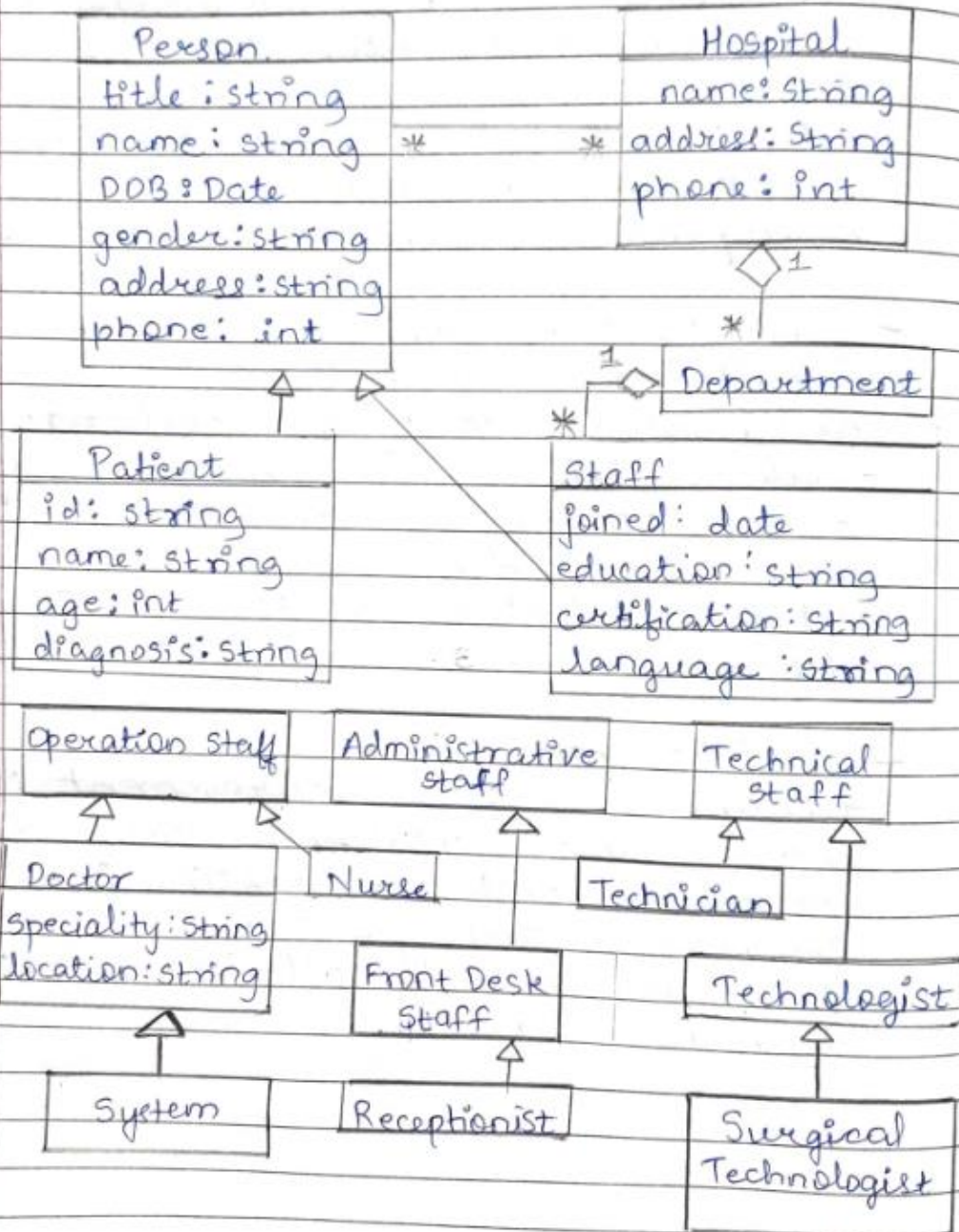


- Class Diagram

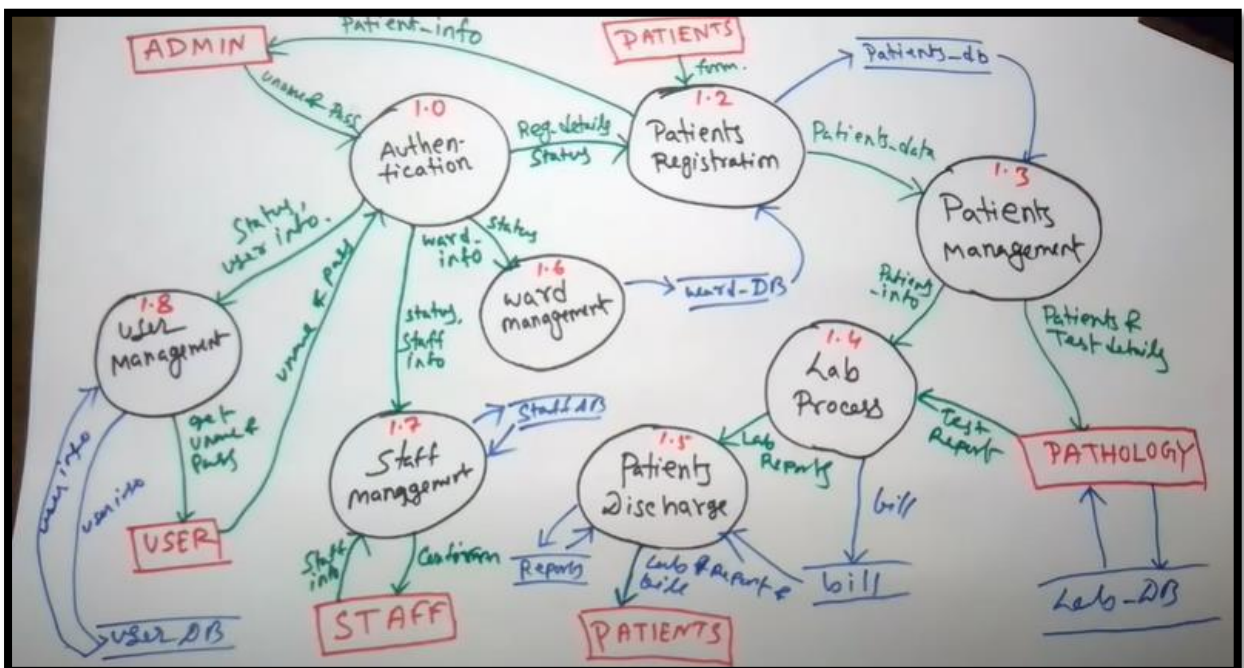
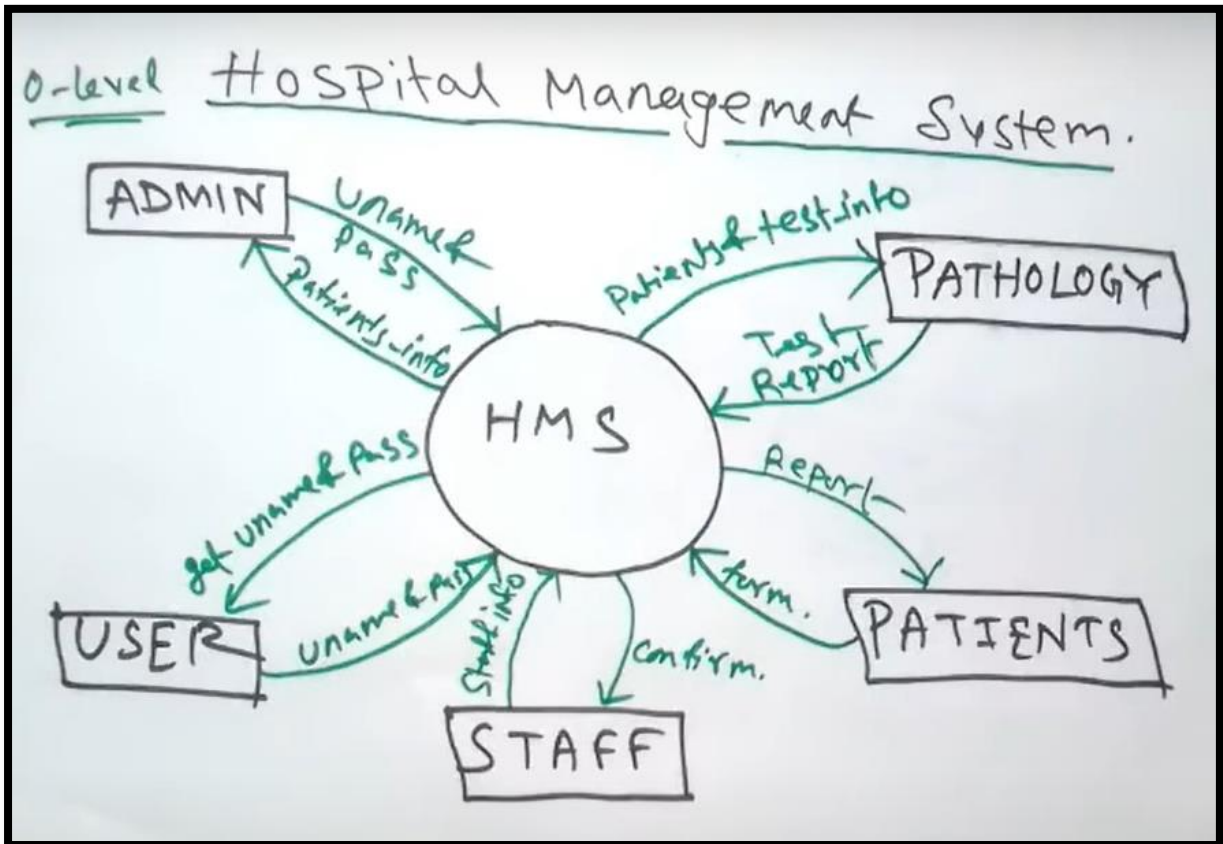
Class Diagram Example: Order System



Q.3) Draw and explain class diagram for hospital management system.



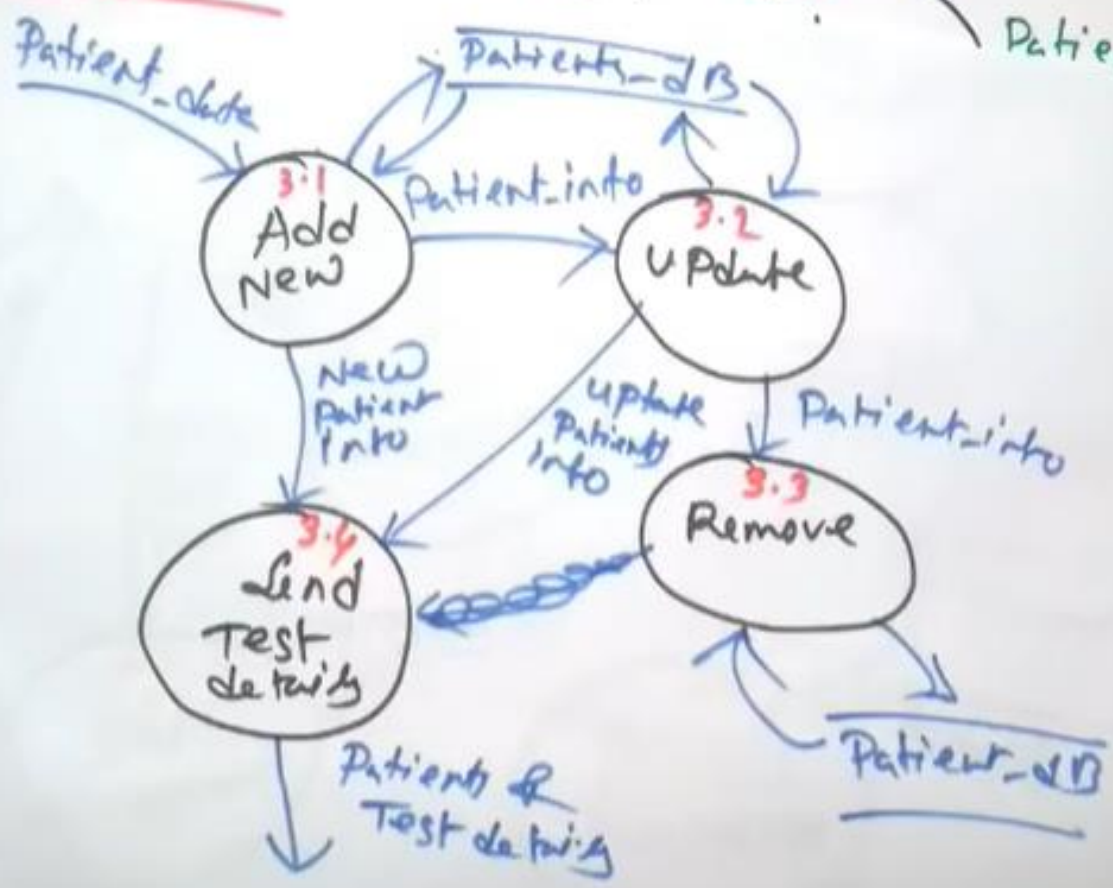
- DFD LEVEL (0,1,2)



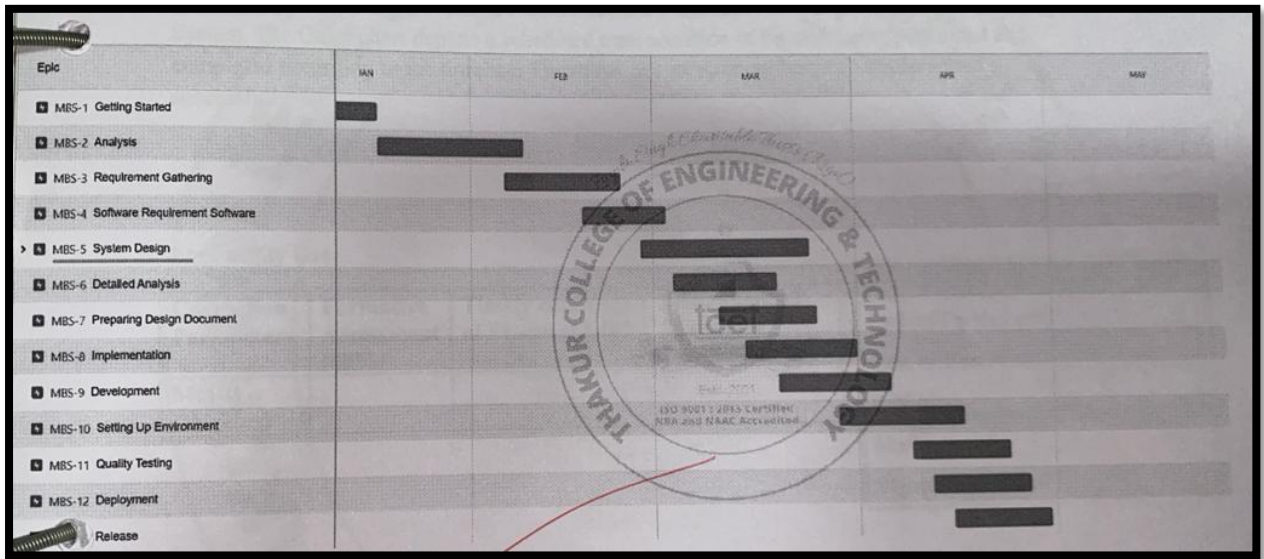
2nd Level

2.0

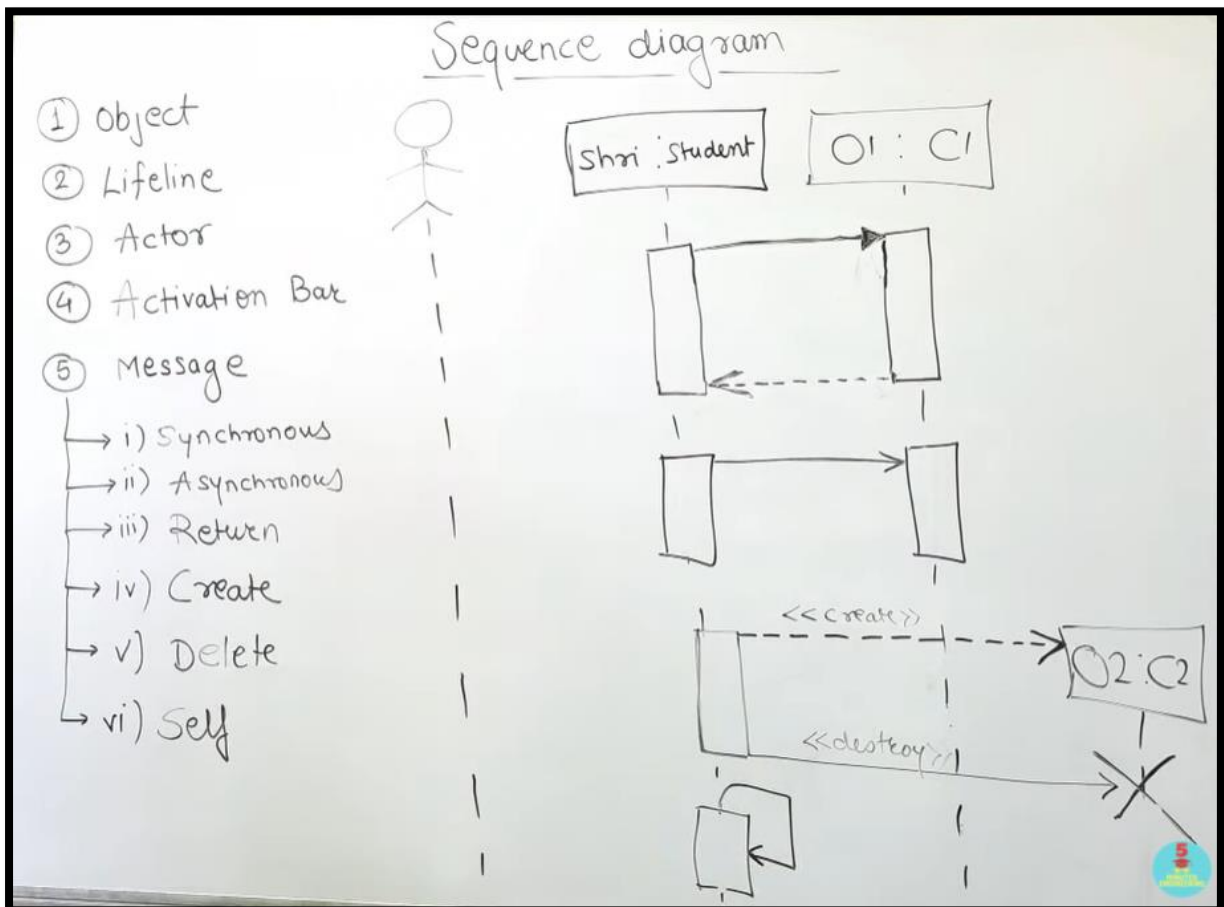
Patients

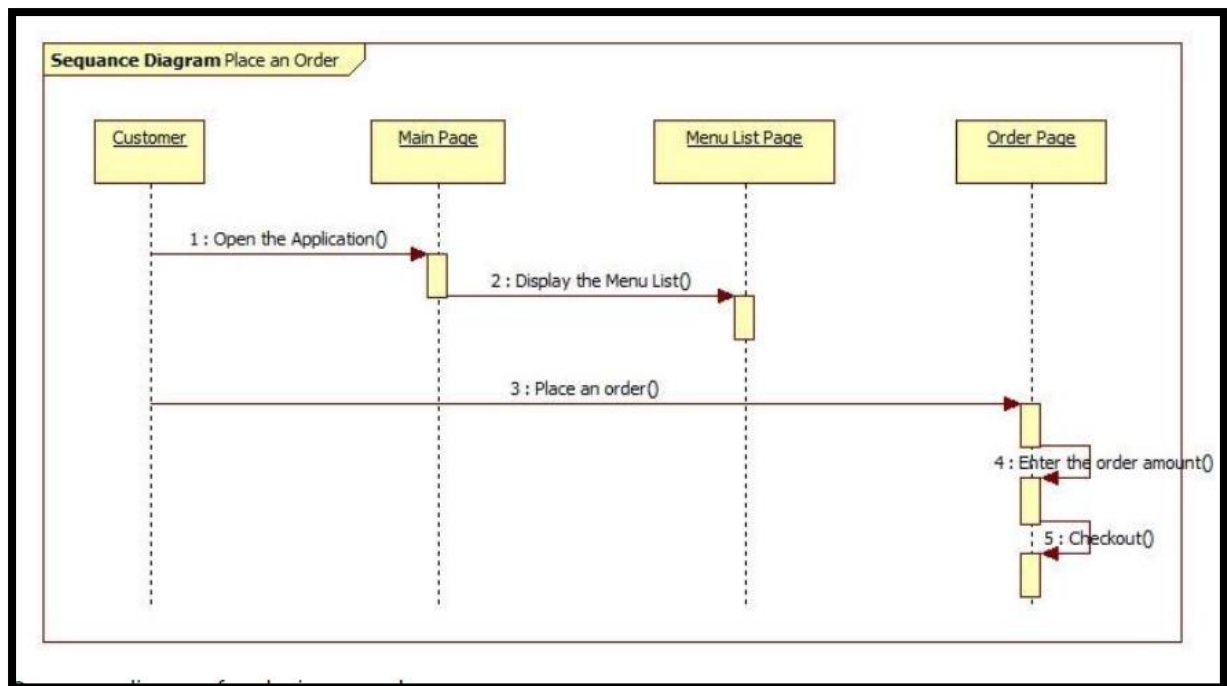


- TimeLine Chart

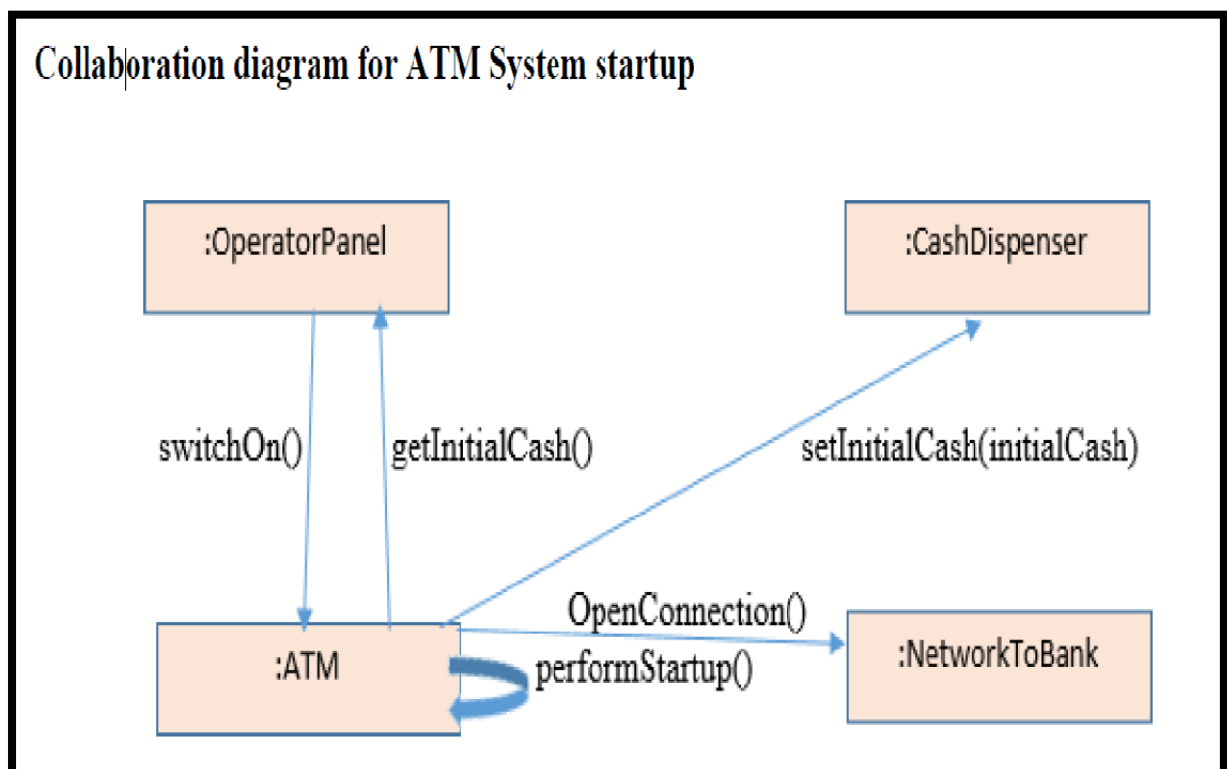


- Sequence diagram

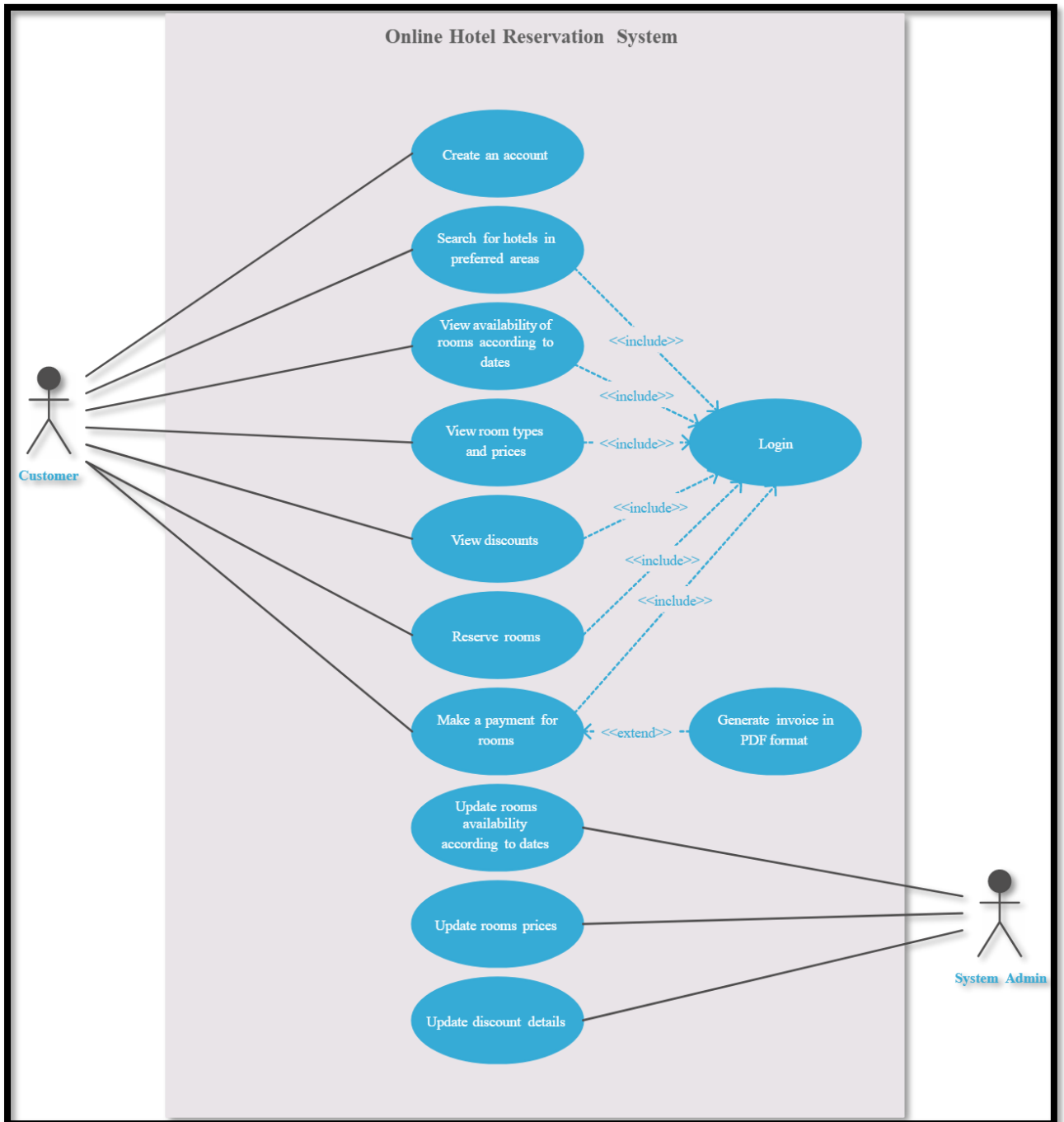




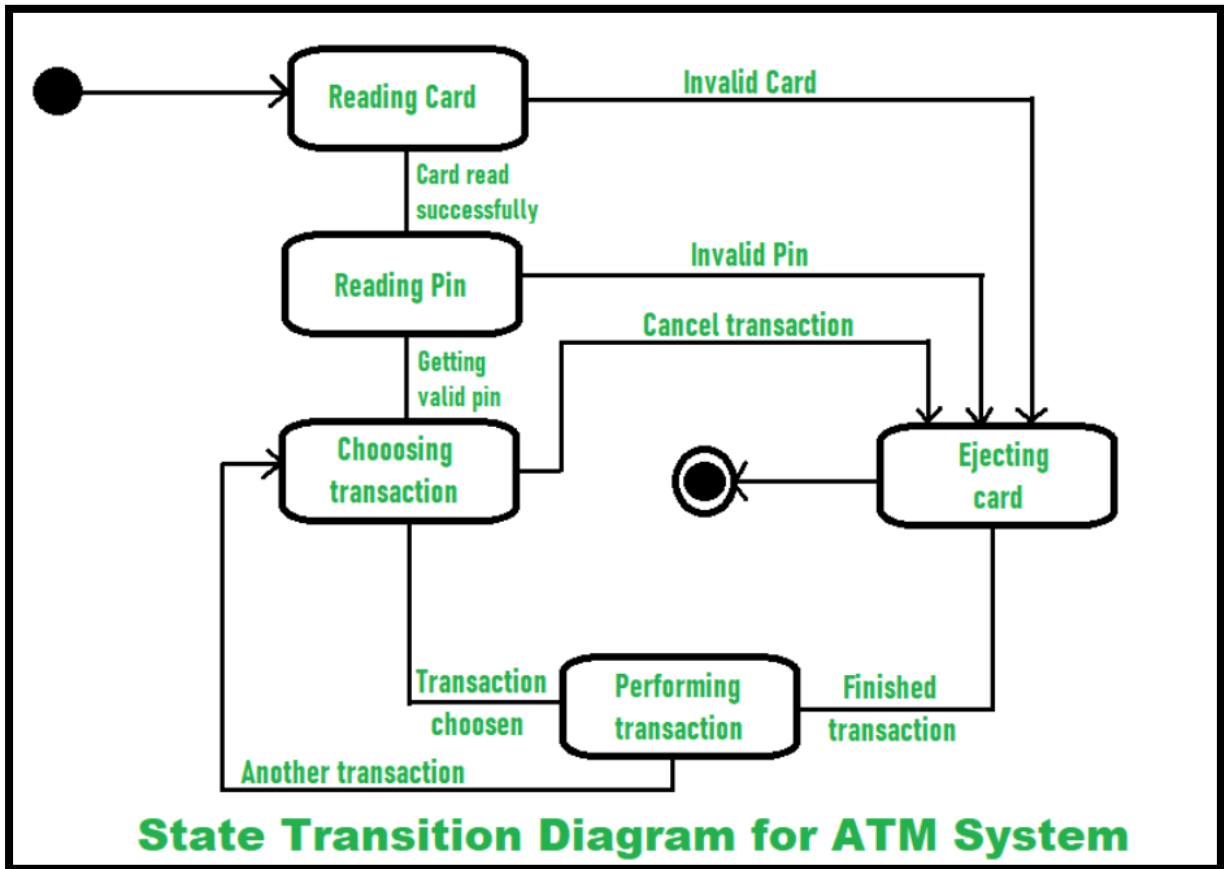
- Collaboration diagram



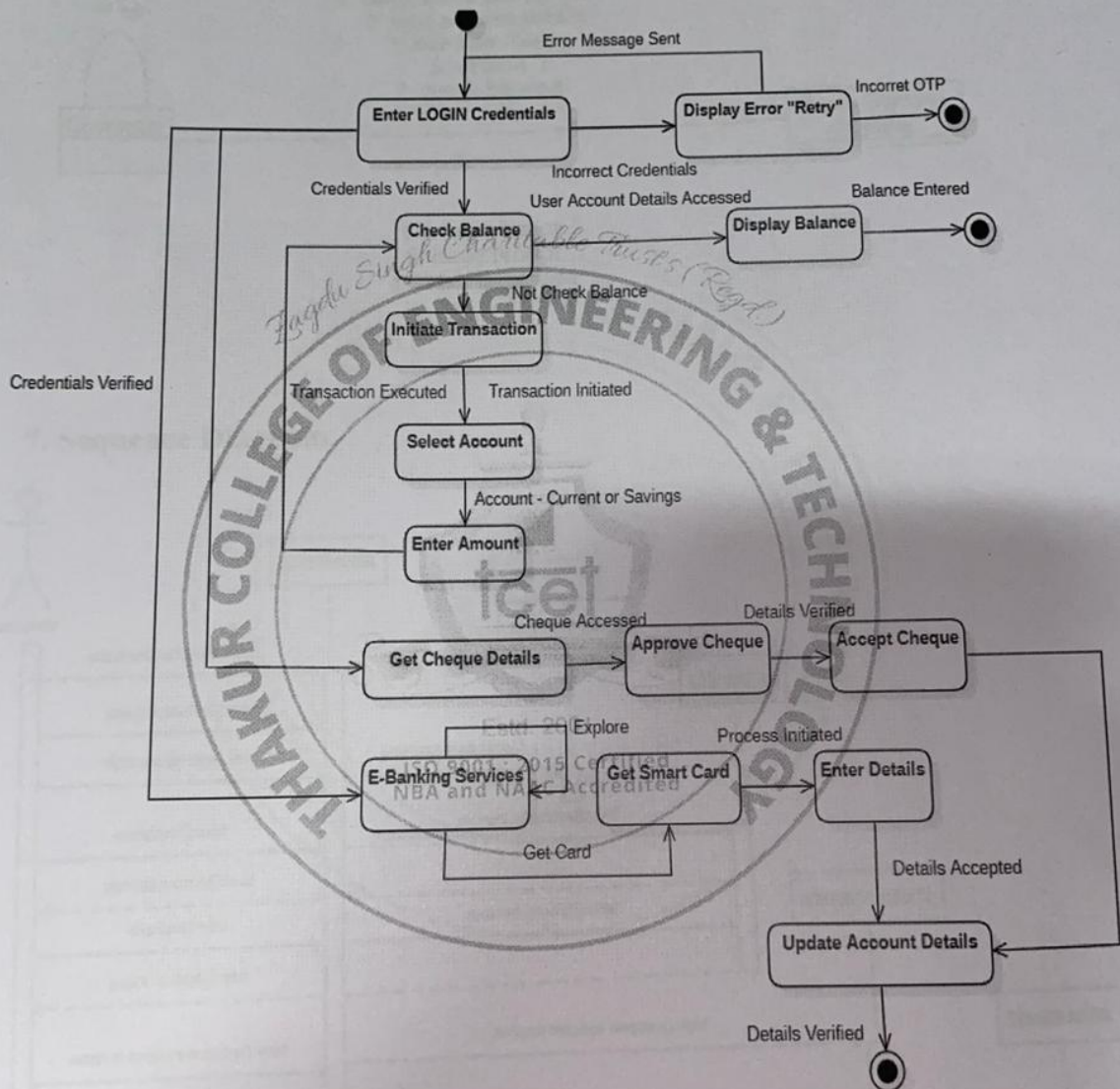
- UML Use Case



- State Diagram



5. State Transition Diagram



- Product and Project Matrix
- Waterfall Model
- Spiral
- Agile

- Dev Ops

SCM – Software Configuration
Management

SCI – Software Configuration Items

Version and Variants

Software versioning is the process of assigning either unique version names or unique version numbers to unique states of computer software.

Version – Windows 7,8,10

Variant – Linux, Mac Os, Windows

Given me one senerio supposed i have erp project and you are develper. initially i have given something to develop it and after few months i bring some changes to it, on which condition weather you have to accept those changes or not, tell me those condition.

Ans: if it is cost effective, initially i have given some cost details now if these changes will will cross those cost limit then i will not accept those changes and if it not crossing cost limit then i will accept those changes

git init → \$ git init Demo

git add →

To add one file

\$ git add Filename

To add more than one file

\$ git add*

git commit → \$ git commit -m " Commit Message"

git push → \$ git push [variable name] master eg. origin master

git checkout → \$ git checkout

\$ git checkout <branchname>

git clone git clone <https://name-of-the-repository-link>

git log → \$ git log

git status → \$ git status

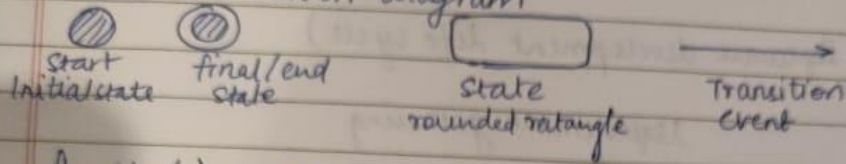
git pull → \$ git pull URL

git merge → \$ git merge BranchName

Verification → Reviews, walk through & Inspection

types System testing - recoverability, Performance, Regression, non-functional

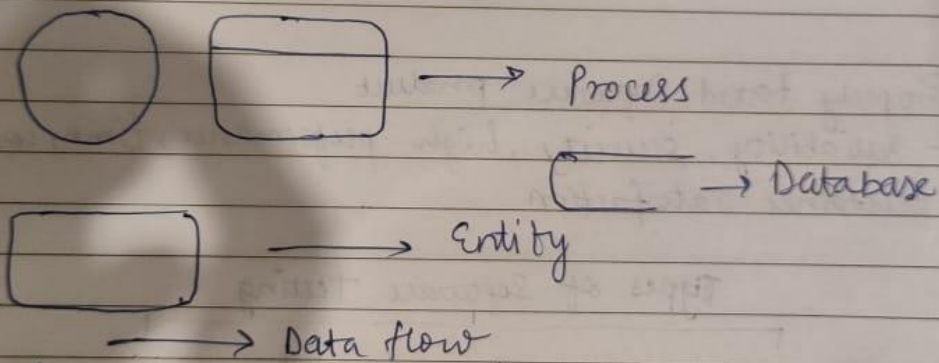
State Transition diagram



function() : eg: prompt_user()
↳ entry or exist actions
represents

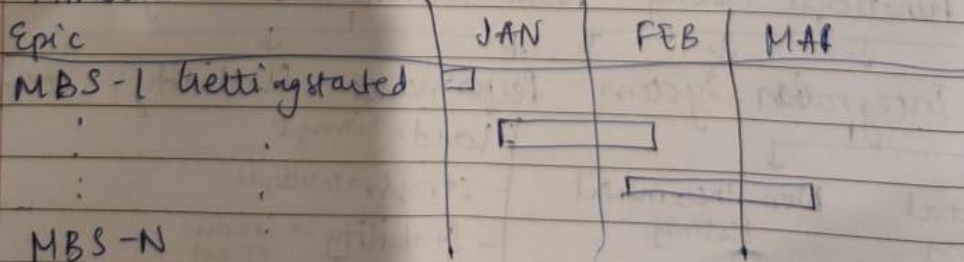
DFD (Data Flow diagram)

Level 0 :



No. of Entity will remain same in all level
Only processes expand / break / increase

Timeline chart

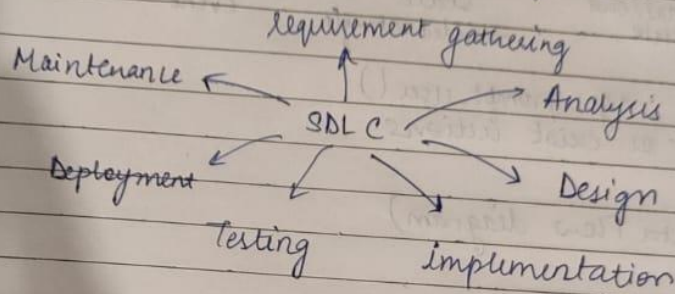


SOFTWARE ENGINEERING

Software Testing

- Detect bug in software
- To meet end user expectation
- To achieve end users expectation

SDLC (Software development life cycle)

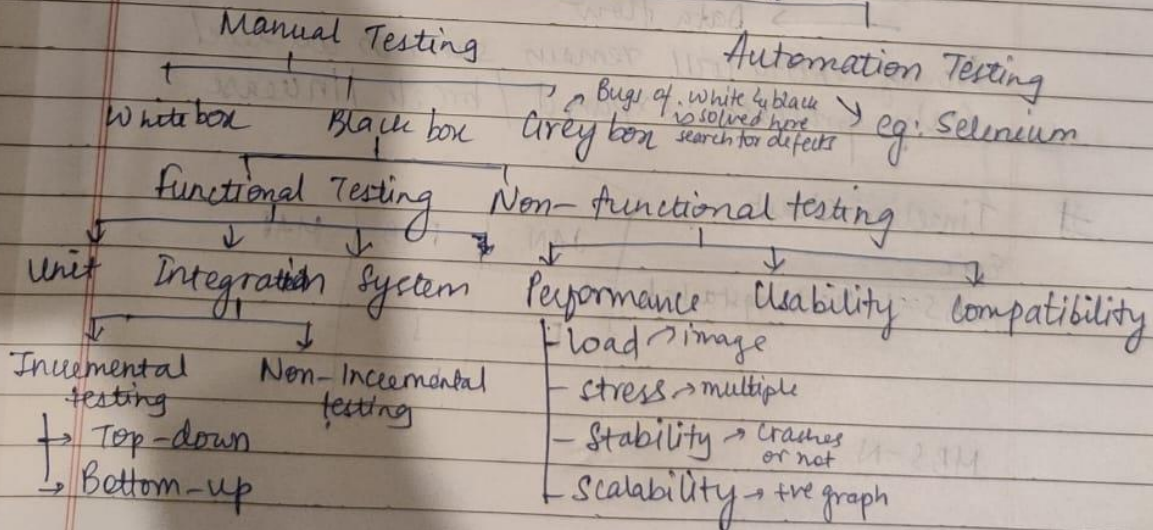


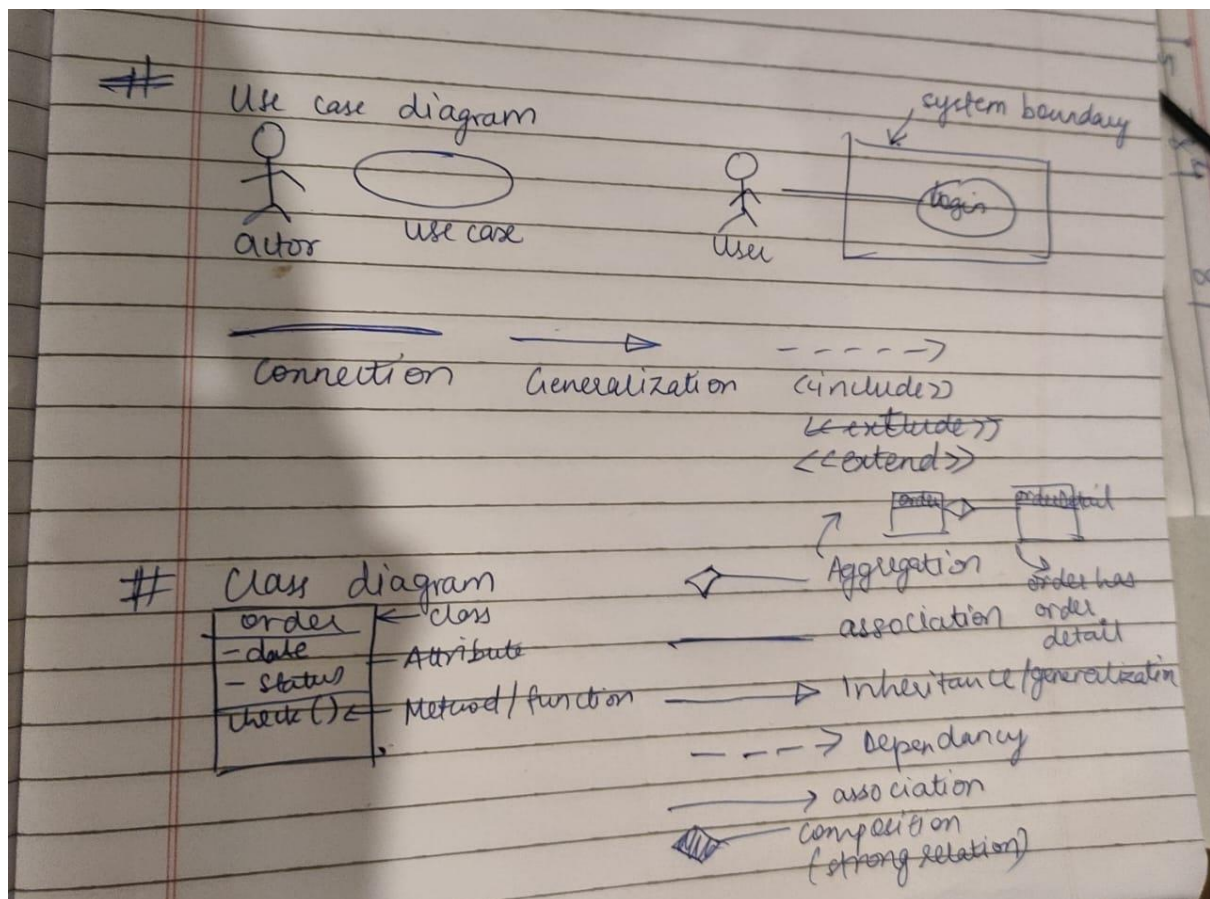
Properly tested Software product

- Reliability, security, high performance, time, cost, customer satisfaction

#

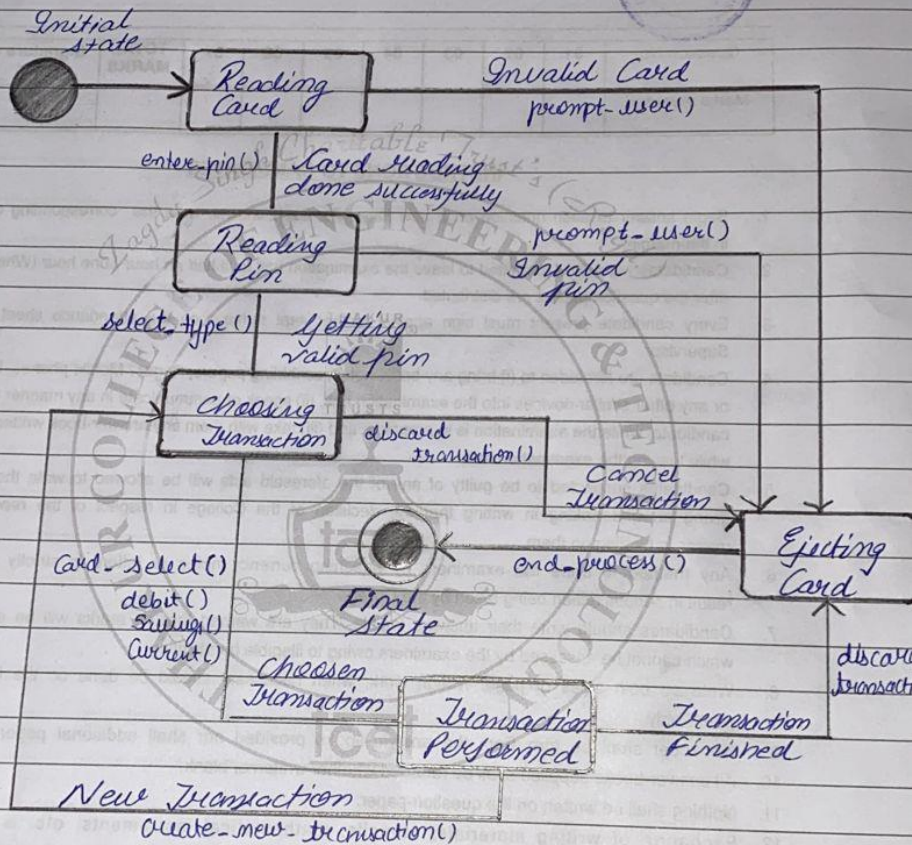
Types of Software Testing





Test case ID	Test Case Objective	Pre-requisites	Expected Result	Actual Result	Status
TC-1	Notification should appear	Face should be detected	A notification should be send to user	Notification is sent to the user	Pass
TC-2	View/Match Face	Database should have images	Face should be matched	Face is matched	Pass
TC-3	Capture Face	Good Quality Camera	Face should be captured	Face is captured	Pass
TC-4	Backup Records	There should be records in database	Records should be backed-up	Records are backed-up	Pass
TC-5	Update Details	Record of Criminal must exist	Records should be updated	Records are updated	Pass

Aim: State Transition Diagram on ATM Machine



When the user inserts the bank card or credit card in the ATM's card reader, the card read action is performed. If the card is not valid then machine will perform exit action. If the card is valid it will then ask for pin. Customer then enters the pin and then it is read. If incorrect pin is entered then it will perform ~~exit~~ exit action. After successful transaction, machine undergoes the exit action.