OOSE notes - 123

Object Oriented Software Engineering (Pokhara University)

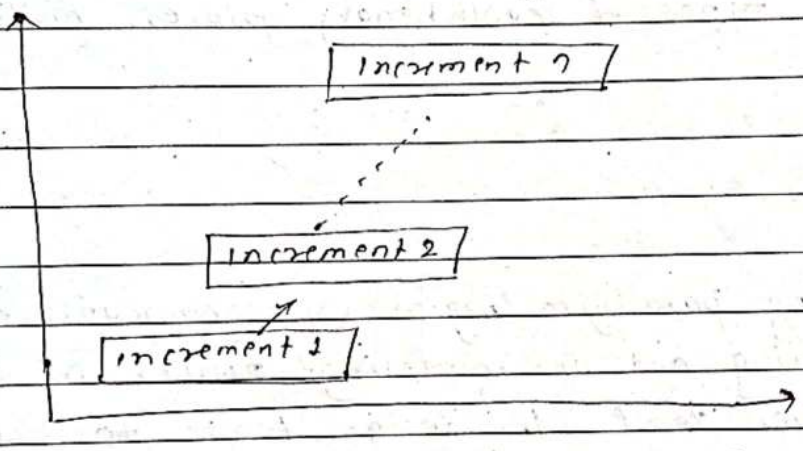
28/1/20

Date
Page

Perspective process Model.

a) RAD (Rapid Application Development)

The incremental model or RAD model combines elements of linear and parallel process flow. These incremental dev. model is particularly useful when completing a project deadline that has been established. If the core product is well received, additional staffing / requirements can be added to implement the next increment ie. increments can be planned to manage technical risks.



RAD model will be implemented when

1. Developing the application in a very short period of time.
2. Customer / client clearly define their requirements.
3. When the staffing is less or multiple projects are running in parallel.
4. very less reusable components.

b) Evolutionary process model

These models are iterative which is suitable for the new system where there is no clear idea of the initial requirements, inputs and outputs parameter. This produces an increasingly more complete version of the software.

This process model are appropriate when

1. No clear idea about the products for both customer and developers.
2. Good communication between the customer and developers.
3. Sufficient time for the s/w development.
4. Very less chances of outsourcing and availability of reusable components.

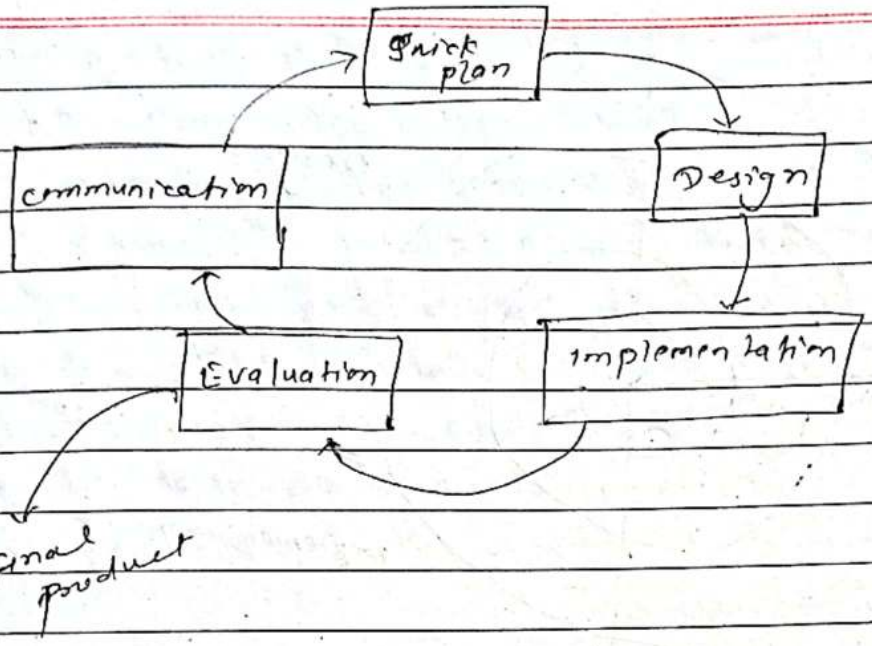
There are two types of evolutionary process model

(1) prototyping
(2) spiral model

Prototyping

— The prototyping paradigm begins with communication and the planning and the modeling occurs in the next iteration. After that the design leads to the creation of the first prototype which is deployed and evaluated by stakeholders/customers/users, who provide feedback to further refine the requirements.

Upon the customers/users feedback, the requirements are redefined and the cycle continues until the cust/user accepts the prototype.



2 week

Assignment

Explain the specialized process models and describe about

(i) component based development

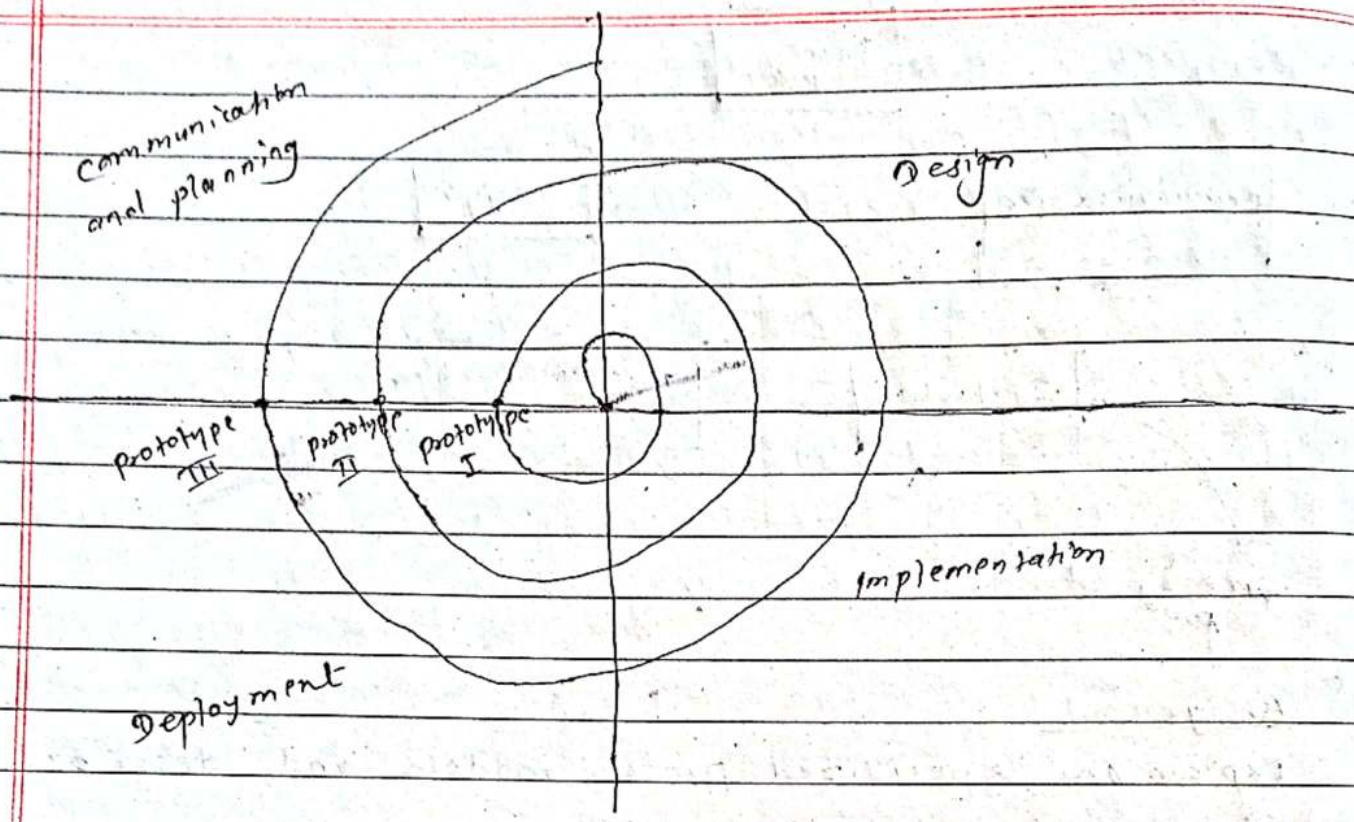
(ii) Formal method model

(iii) Aspect oriented software models/ development.

spiral model

spiral model is the combination of waterfall model and prototyping model which is used for rappid development of increasingly more complete version of the S/W. This is the realistic approach to the development of the large scale systems and softwares.

The spiral model uses prototyping as a risk reduction mechanism and also enables a prototype at each evolutionary level.



Unified process and UML

UML stands for Unified Modeling Language. which is a standard language for writing software and blue prints. UML is used to visualize, specify, construct and documents the artifacts of a s/w sys. which is used for modeling the systems from Enterprise information sys to distributed web based systems and even real time embedded system.

The UML method generally uses the concept of

(i) Unified - It combines the preceding methods of object oriented.

(ii) Modeling - it is used for visually modeling the system.

(iii) Language - It offers syntax through which the above model can be constructed using any programming language.

Date
Page

The UML diagrams are used for

(i) visualizing the components
(ii) specifying the models
(iii) constructing the models.
(iv) proper documentation of a particular s/w.

Inside the UML diagram.. there are

(I) static or structural diagram
(II) Behavioural diagram

(I) class diagram
(2) object diagram
(3) component diagram
(4) Deployment diagram

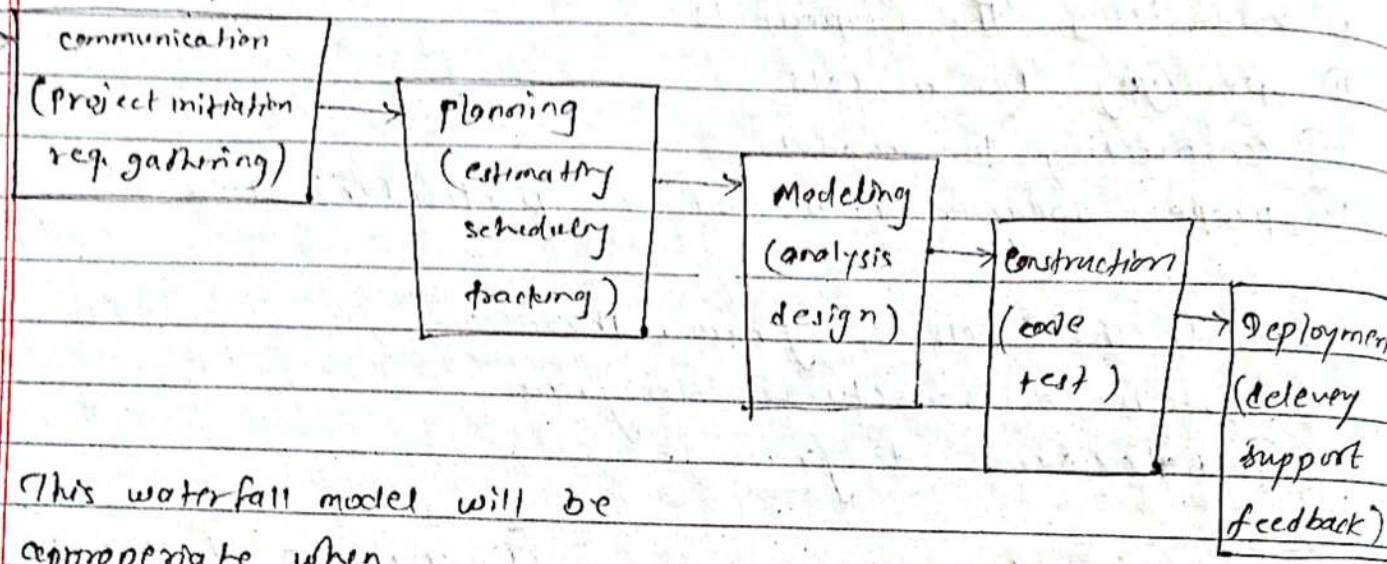
(1) Use case diagram
(2) Interaction diagram
  ↳ a) sequential diagram
  b) collaboration diagram
(3) Activity diagram
(4) state chart diagram

2080/01/21

Date
Page

11 Perspective process models

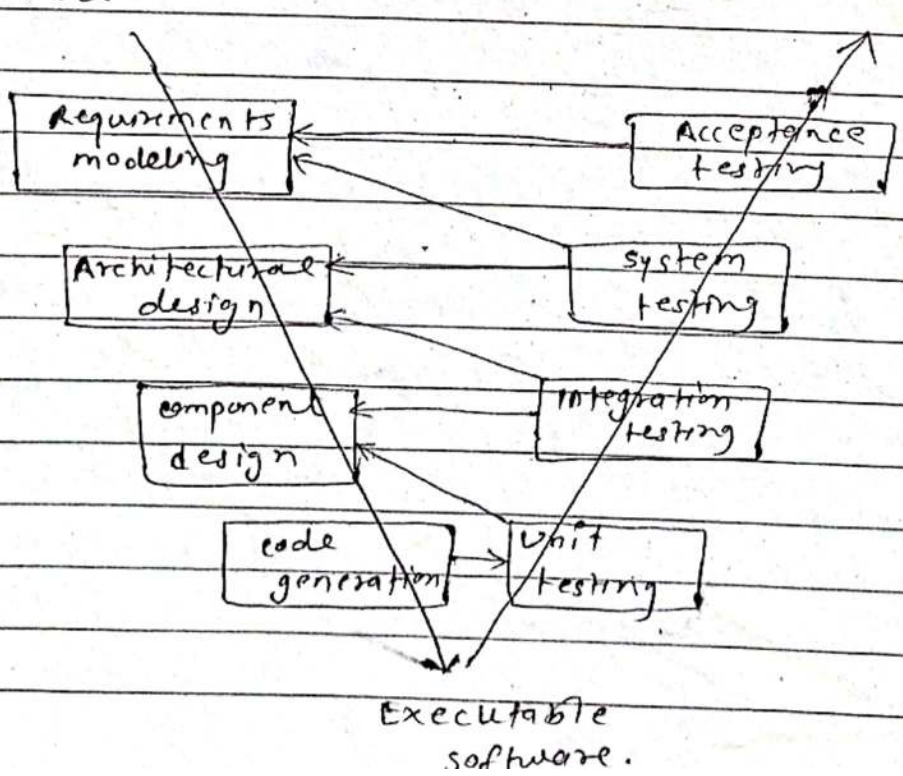
a) waterfall model



This waterfall model will be appropriate when

1. User requirements are clearly defined.
2. Less no. of staffing working on a project in the organization.
3. very less reusable components
4. sufficient time for the s/w development.
5. Less comm? bet^n customer & developer.

b) V model

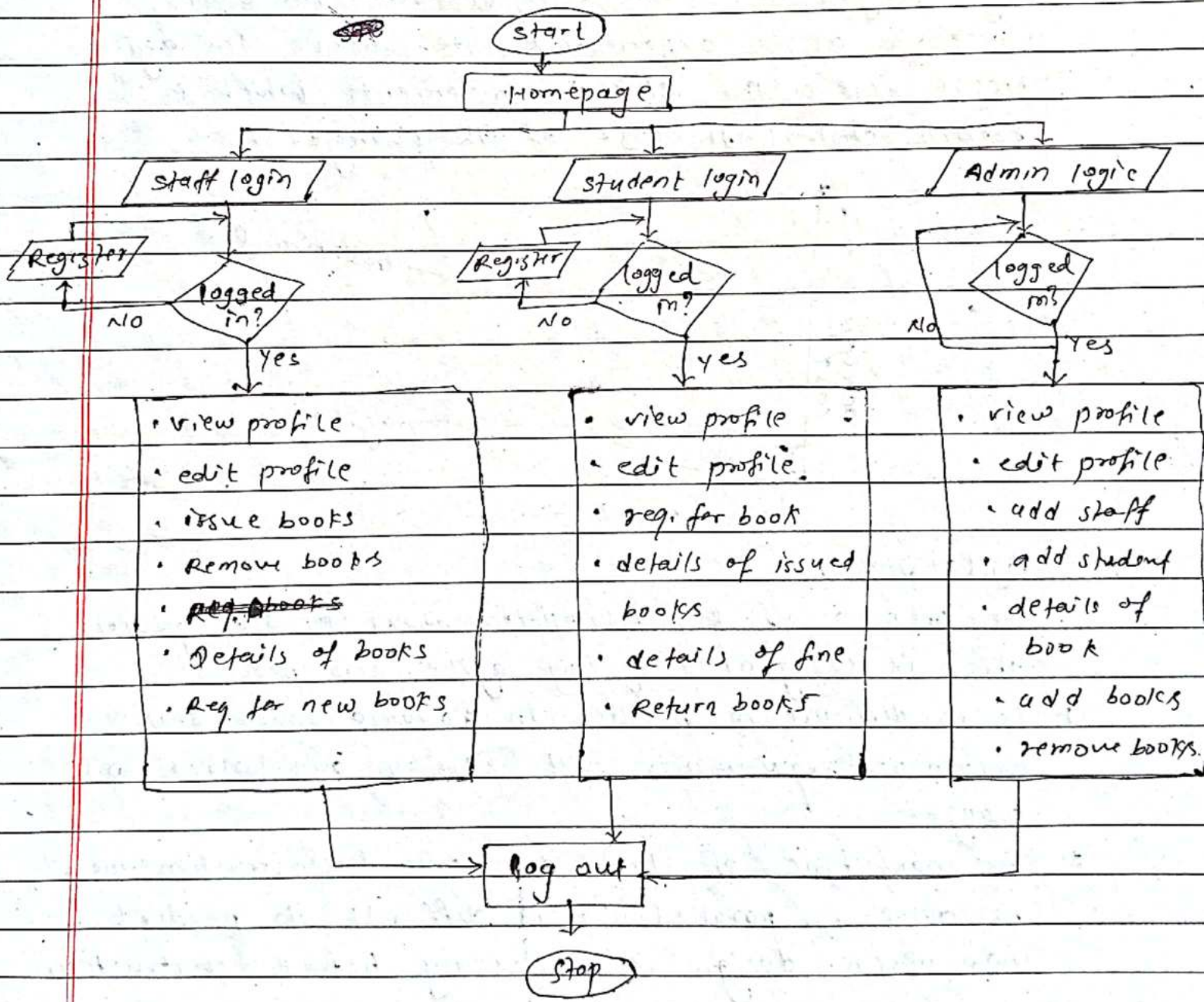


Date
Page

How to create a system flowchart in 5 steps :-

1. Define the scope of your project
2. Identify tasks in chronological order
3. Organize tasks by type and symbol
4. Draw your flowchart
5. Confirm and refine your flowchart.

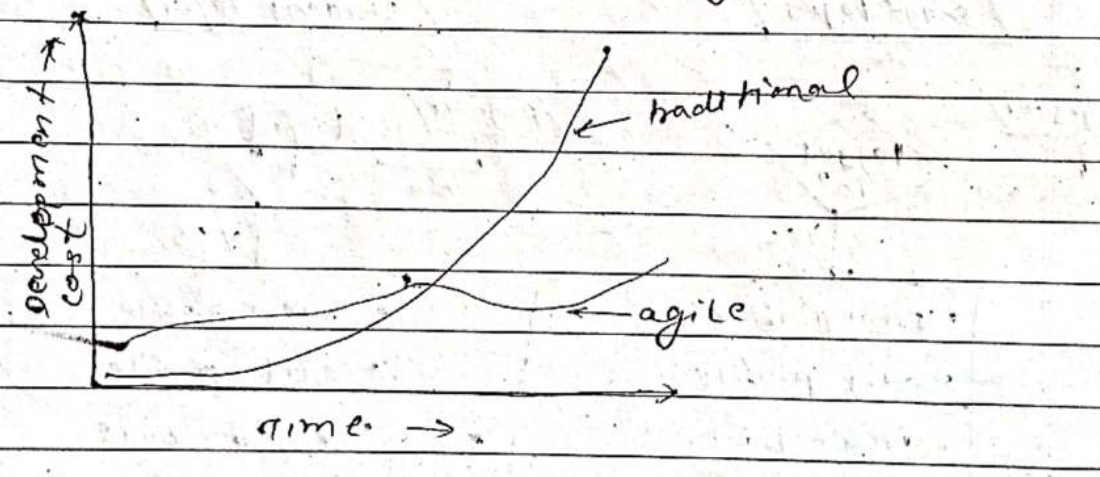
library mgmt. sys



Date
Page

Agile development

- The basic framework activities such as comm", planning, modeling, construction and deployment. The same framework also can used for agile dev. but they transform into the minimum tasks that pushes the proj. team toward const" and delevery.
- An agile process reduces the cost of change in a particular s/w development. Traditionaly if there is any changes to be made in the s/w, the cost increases at an exponential rate. Hence the agile process release the s/w in increments which in return; control the cost of the change.



Agile process :-

There are a no. of key assumption about a s/w project which is categorized by any agile s/w process.

1. It is difficult to predict in advance which s/w components/ requirements will persist and which will change.
2. For many types of s/w, design and construction should be done in parallel. It is difficult to predict how much design is necessary before construction

which is used to prove the design.

3. Analysis, design, construction and testing are not as predictable

An agile s/w process adapts incrementally with the help of a regular customer feedback i.e. there are executable prototypes / s/w increments delivered in a short period of time to adapt. The change of pace done by the customer feedback. Also iterative approach enables the customer to evaluate the s/w increment, provide necessary feedback and influence the process adaptation for the better s/w dev.

Agility principle

The agile alliance define the 12 agility principles which are:

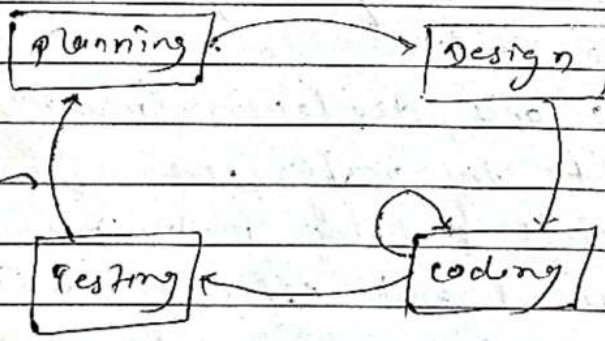
1. Our heighest priority is to satisfy the customer thru early and continuous delivery of valu s/w.
2. The change in the req. must be wellomed even late in the dev.
3. Deliver working s/w frequently with a preference to the shorter time scale.
4. Business people and developers must work together dailly throughout the project.
5. Must build project around motivated individuals.
6. The most efficient and effective method of conveying information within a development team is face-to-face conversation.

7. Working s/w is the primary measure of progress.
8. Agile process must maintain sustainable development
9. Continuous attention to technical excellence and good design enhances the agility.
10. Simplicity is essential.
11. The best architecture, requirements and design emmerge from self organizing teams.
12. At regular intervals, the team reflects on how to become more effective. Then tuens and adjust its behaviour accordingly.

Agile process models:-

1. Extreme programming (XP)

- It is the most widely used approach to agile s/w development which uses an object oriented approach as its preferred development paradigm and encompases a set of rules and practise that occur within the context of four framework activities. which are planning, design, coding, and testing.

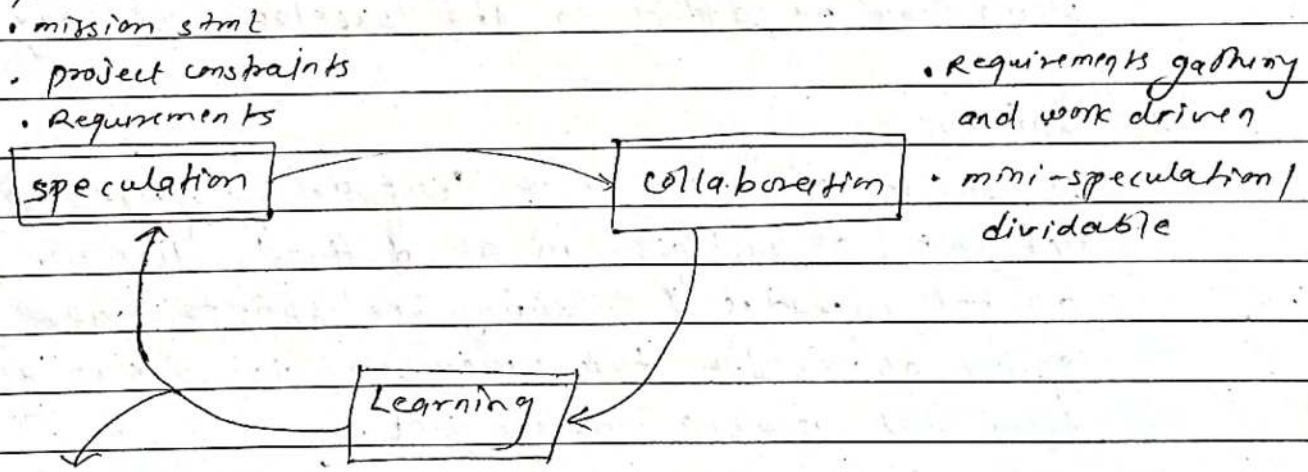


2. Adaptive software development. (ASD)

— ASD lifecycle incorporates three phases which are speculation, collaboration and learning.

During speculation, the project is initiated and adaptive cycle planning is conducted which uses project initiation information like mission statement (objectives), project constraints (delivery dates or user descriptions) and basic requirements (software increments).

Similarly while collaborating, all the requirements gathered will be completed in a minievent and learning incorporates what components are implemented, testing to be done and technical reviews for the deployment of the project.



Date
Page

8. Scrum

- Scrum is an agile s/w dev. method developed in the early 90s which uses agile platform for its development activities and incorporates the framework such as requirements, analysis, design, evolution, and delivery. Scrum emphesis the use of a s/w process pattern that have proven effective for project with tight ~~datebase~~ deadline, changing requirements and business criticality.

Some of the components used in a development are:

i) Backlogs

- It is a priotized list of requirements or features that provide business value for the customer. Also items can be added to the backlock at any time.

ii) sprints

- It consists of work units that are required to achieve a backlogs in a defined time. changes are not introduced during the sprints cause the sprint allows team members to work in a short term but stable environment.

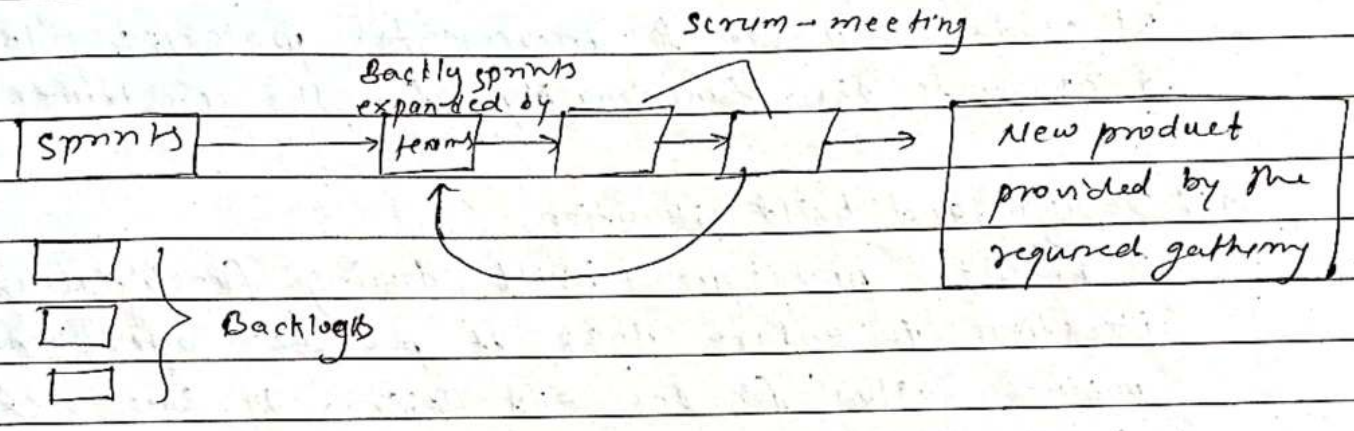
iii) scrum meetings

- These are the short meetings (roughly about 15 mins) held daily by the scrum team. The scrum meeting helps the team to uncover potential problems as early as possible. Also these daily meetings leads to knowledge socialization and

promoting a self organizing team structure.

iv) Demos

- It delivers the software increment to the customer so that the functionality that has been implemented can be demonstrated and evaluated by the customer.



4. Dynamic system development model (DSDM)

- DSDM provides a framework for building and maintaining systems which meet the tight time constraints thru the use of incremental prototyping in a controlled project environment. This DSDM can also be combined with extreme programming to provide a combination approach.

The DSDM lifecycle uses three different iterative cycle preceeded by two additional lifecycle activities.

i) Feasibility study

It establish the basic business requirements and constraints associated with the application to be built and then asses whether the application is a viable candidate for the DSDM process.

ii) Business study
It establishes the functional and the information requirements that will allow the application to provide business value.

iii) Functional model iteration
It produces a set of incremental prototypes that demonstrate the functionality for the customer.

iv) Design and build iteration
It revisits prototypes built during functional model iteration to ensure that it provides operational business value for the end users. In some cases functional model iteration and design build iteration occurs concurrently.

v) Implementation
It provides the operational environment of the design model using any programming language.

2080/12/28

Date
Page

5. Feature Driven Development (FDD)

FDD was used as a practice process model for the development of the s/w by using object oriented approach. After that some changes were made and improved using agile processwhere this model can properly used for medium scale and large scale projects. This model is applied to adopt the following philosophy:

i) Emphesizes on group members on the some FDD team.

ii) Manages problem and project complexity using feature based decomposition followed by the integration of s/w increments.

iii) Communication of technical detail using verbal, graphical, and text based means.

iv) FDD emphesizes software quality assurence by encouraging an incremental development strategy, design and code inspection, software quality assurence audits and the collection of the matrix.

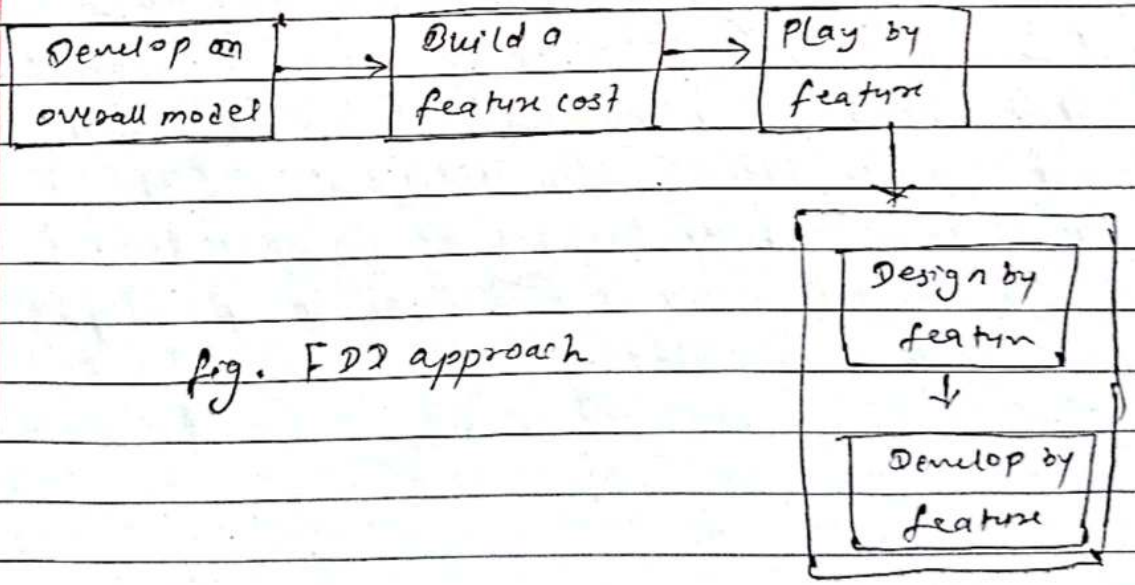


fig. FDD approach

Date
Page

6. Lean Software Development (LSD)

LSD adapts the principle of Lean manufacturing in the world of s/w engineering. Lean principles are eliminate waste, build quality, create knowledge, ~~default~~ defer commitment, deliver fast, respect people, optimize all.

Ch. 3 Software Modeling

2080/11/8

Date
Page

software modeling

Some principles in the SW engineering practice that has a single overriding goal, deliverable on time, high quality, operational software are given below.

1. Divide and conquer
2. Understanding the use of abstraction
3. Strive for consistency.
4. Focus on the transfer of information
5. Build software that exhibits effective modularity.
6. Look for patterns
7. When possible represent the problem and its solution from a number of different perspectives.
8. Remember that someone will maintain the software.

Requirement engineering

RE tasks are conducted to establish a solid foundation for design and constructions which are defined for the generic software process. Following are the steps in requirement engineering

1. Inception
The task that defines the scope and nature of the problem/project to be solved.

2. Elicitiation
A task that helps stakeholders define what is required that is problem of scope, problem of understanding, and problems of volatility.

3. Eloboration
The information obtained from customer during inception and elicitiation is expanded and refined during elaboration ie. this tasks focous on developing a refined requirement that identifies the behaviour and function of a software.

4. Negotiation
What are the priorities in a project, what are the essential components, and when it is required.

5. Specification
The term specification means different things to different people ie. the specification can be a written document or a set of graphical models or a formal mathematical model or a prototype or any combination of these.

Date
Page

Developing the requirement model

The intend of the analysis model is to provide a description of the required informational, functional, and behavioural domains of a computer based system. The model changes dynamically as we learned about building a system and understanding what does the stakeholder needs.

There are many different ways to look for the requirements for a computer based sys.

1. scenario based elements

The system is described from the user's point of view using a scenario based approach. For eg. basic use-cases and their corresponding use case diagram and activity diagram.

Use case

A use case diagram is a visual representation of a system functionalities and actors involved on it. ie. It helps software developers to understand the system requirements and design the sys. architecture more effectively.

Use case diagrams provides a clear and consized way to visualize the system functionality and helping SW developers to identify user requirements and design the system accordingly.

Some benifits of the use case diagram are:

1. clear understanding of the sys functionality and user interaction.

2. Efficient gathering of system requirements.
3. Effective communication among stakeholders.
4. Efficient testing of the system.
5. Improve project management.

Components of the use-case diagram:

1. system → The process and steps taken to reach the endgoal and including the necessary functional requirements and their anticipated behaviours.

2. Actors → An entity that interacts with the system to perform a specific function or a task is known as actor. We have two types of actors:
   i) primary actor → It initiate a use case
   ii) supporting actor → It provides supports to the primary actor.

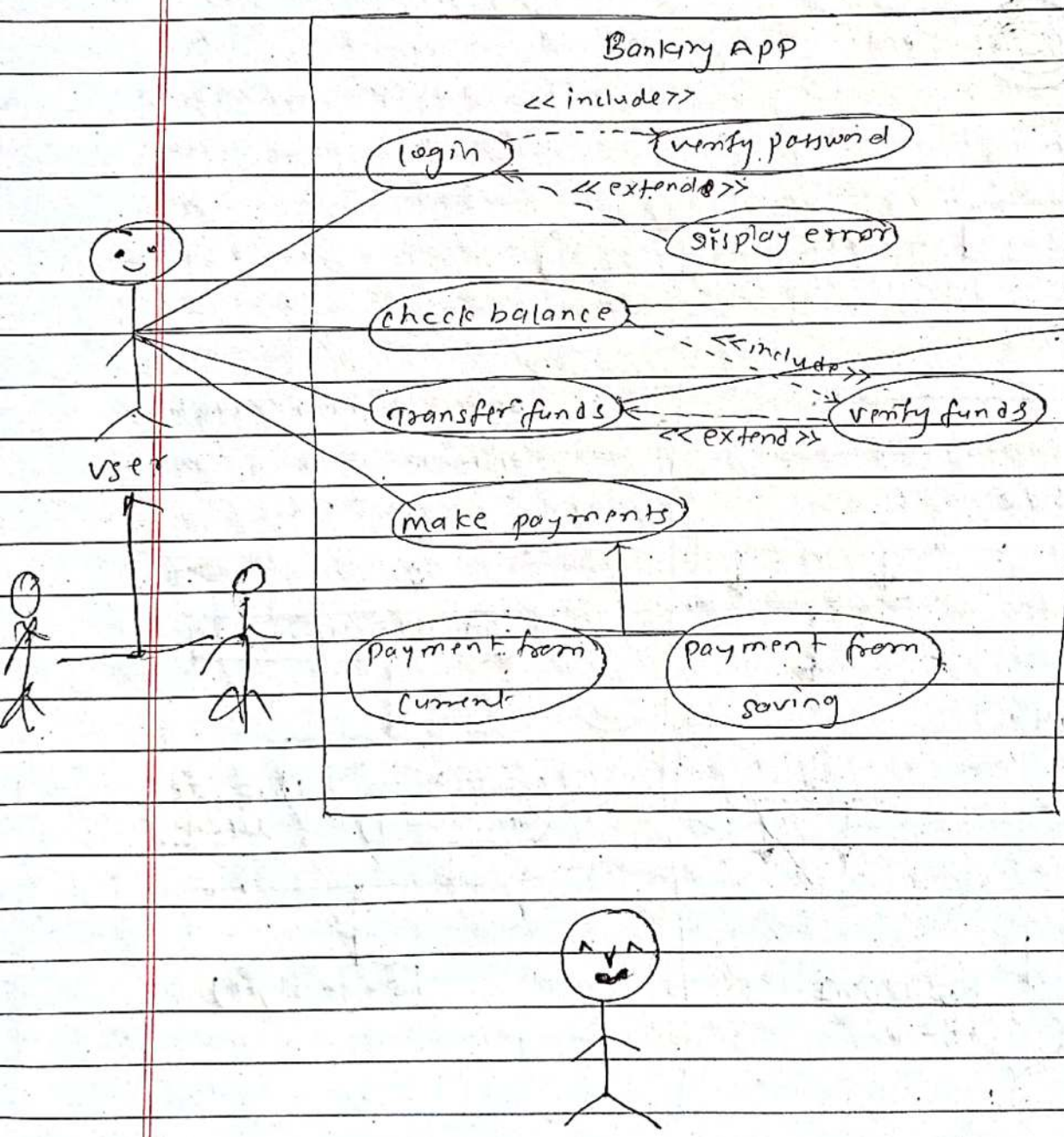
3. Use-case → use-case defins the system functions that a particular user will perform to solve the(a) divided module.

4. Relationship and association → A relationship between a actor and a use-case indicates the actor interacts with the use-case. i)

ii) Include → A relationship between two use cases, where one use case includes the functionality of another use case.

iii) Extend → A relationship bet^n two use cases where one use case extends the functionality of another use case

iv) Generalization → A relationship bet^n two use cases where one use case is a more general version of the other.



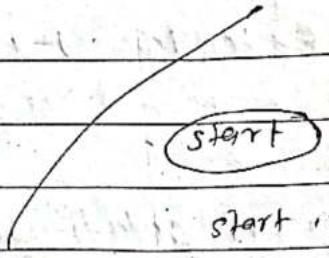
2080/2/18

Date
Page

Activity diagram.

1. start node

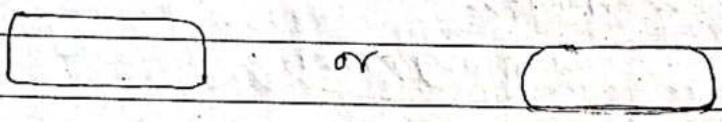
start node



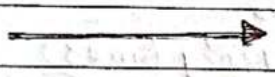
start in flowchart.

2. Final activity node.

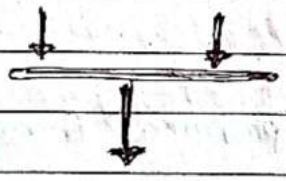
3. Activity



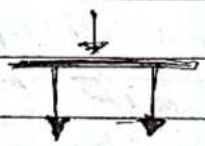
4. Control flow



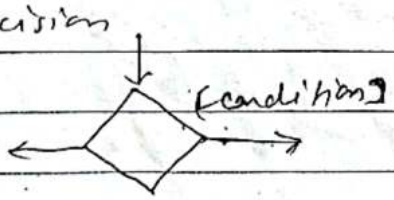
5. Join node



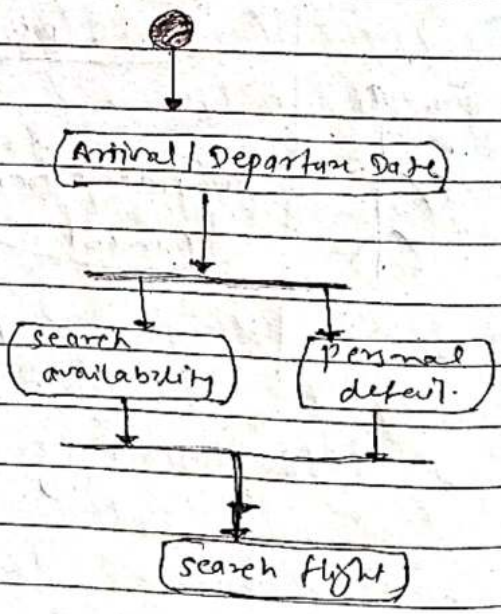
6. Fork node



7. Decision

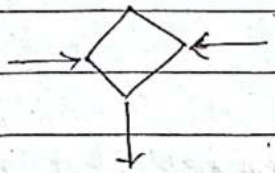


online flight reservation system.

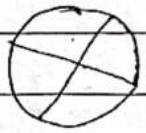


Date
Page

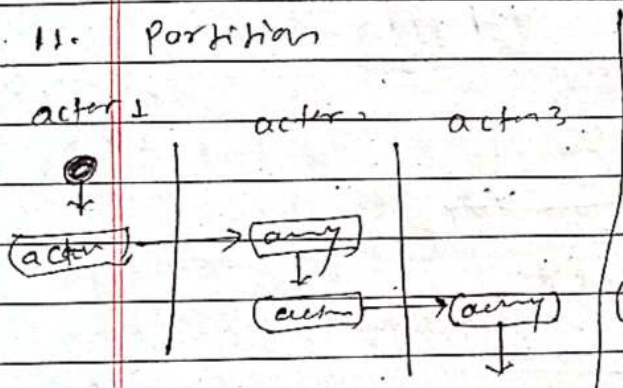
9. merge node



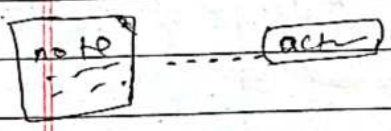
10. Final flow node.



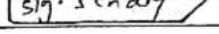
11. Partition



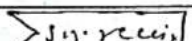
12. note



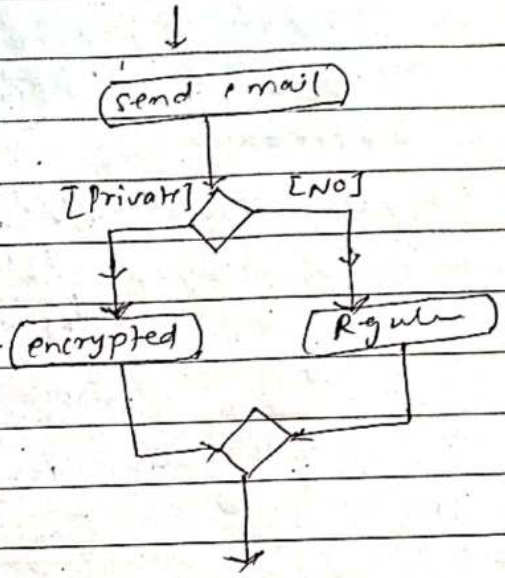
13. signal sending



14. signal receipt

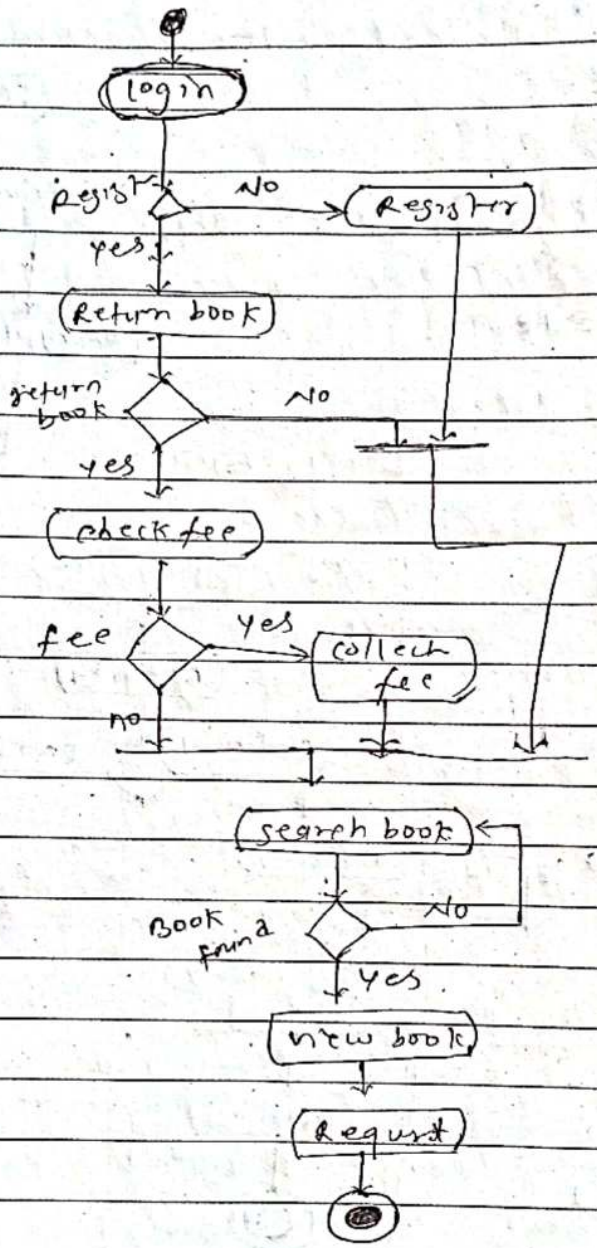


sending email.



Date
Page

Borrowing a book



Date
Page

2. Data modeling

class diagram + object diagram

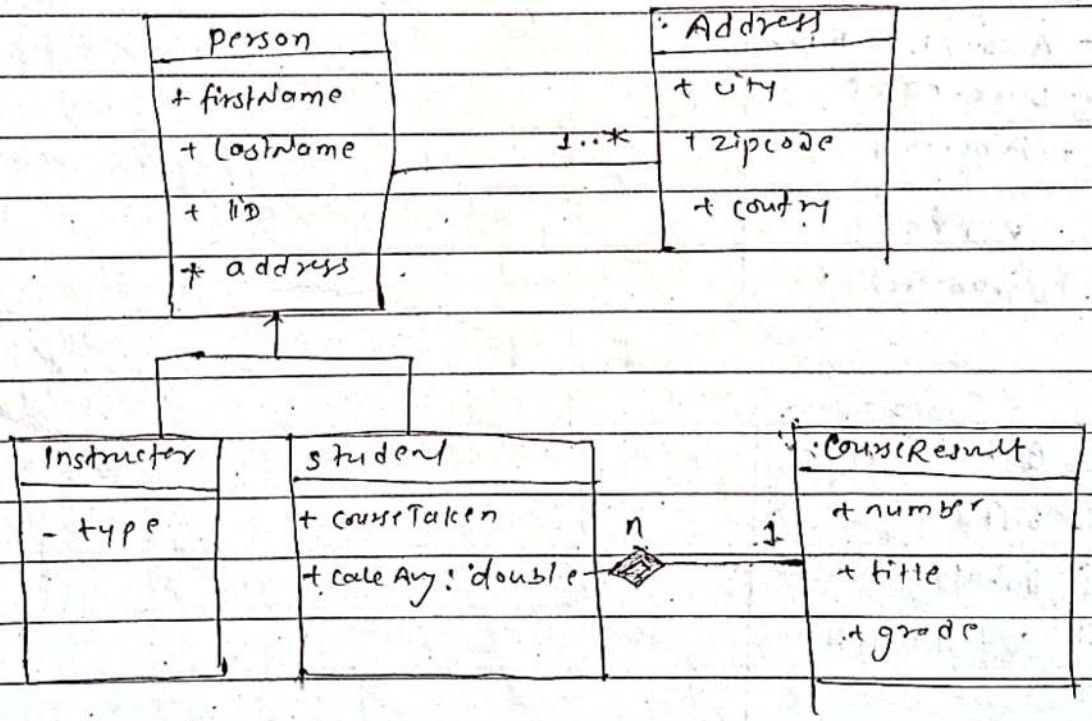
class diagram

1. class name
2. class attributes
3. class operation
4. Relationship (←)

+ public
# protected
- private

- Association (loosely bind) / derived
- Aggregation (belongs to) ◇→
- generalization (inheritance) ↑
- Dependency ····→
- Composition ◆→

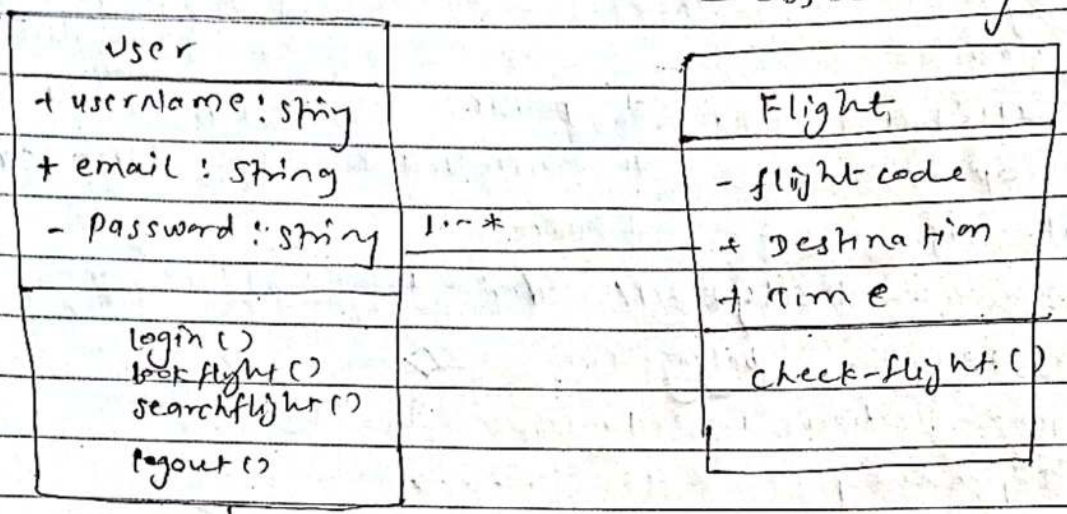
no need to make use case diagram

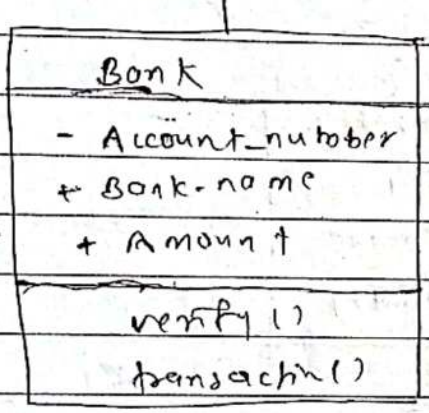


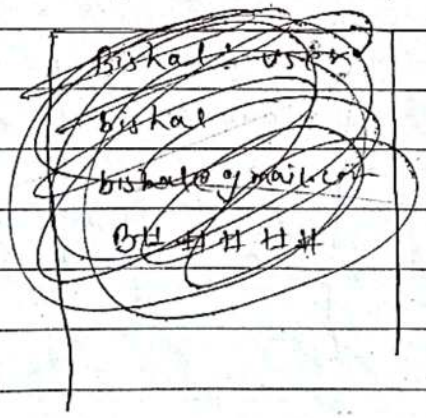
object diagram

Date
Page

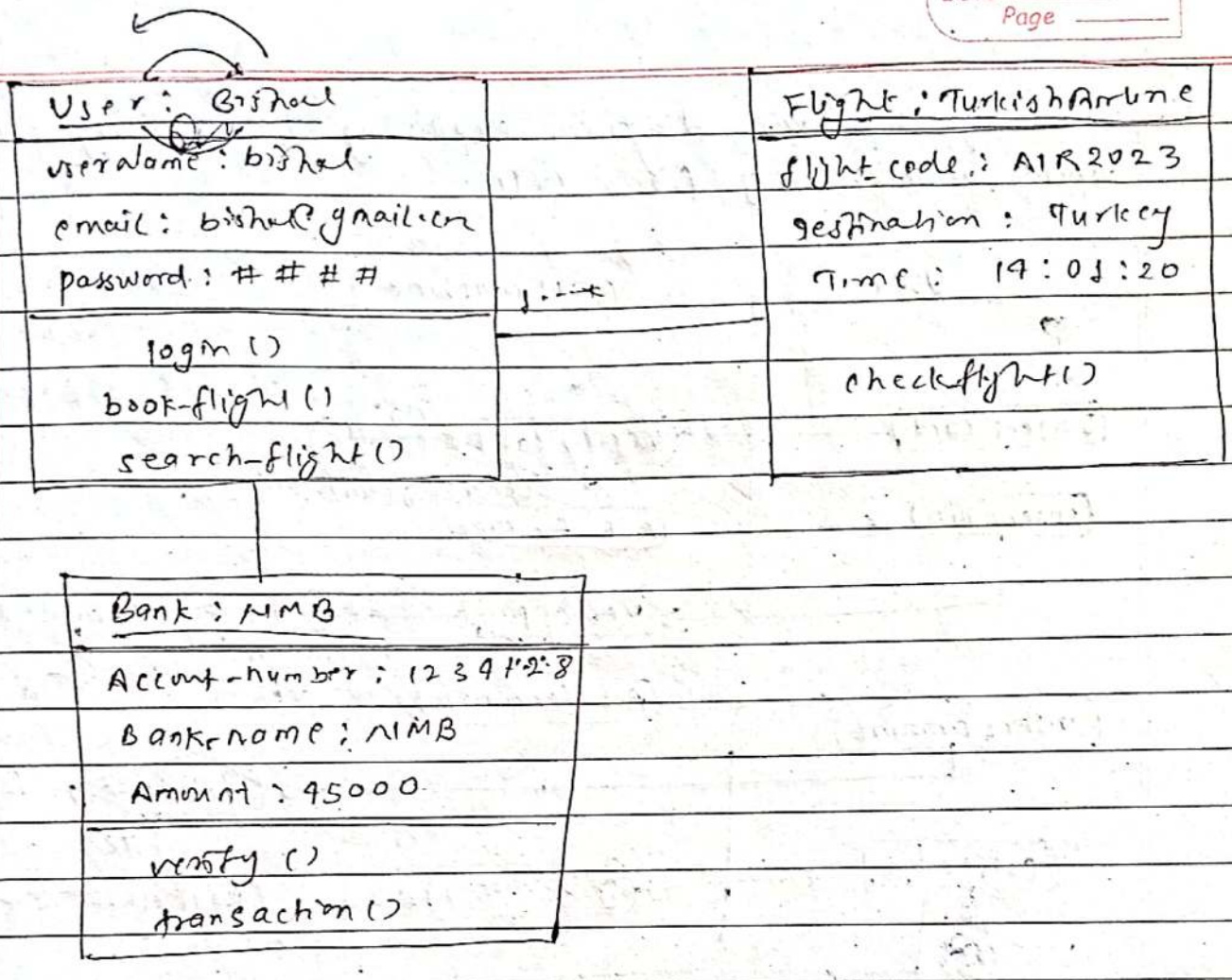
Q. Online Reservation system को - class diagram and
- object diagram.







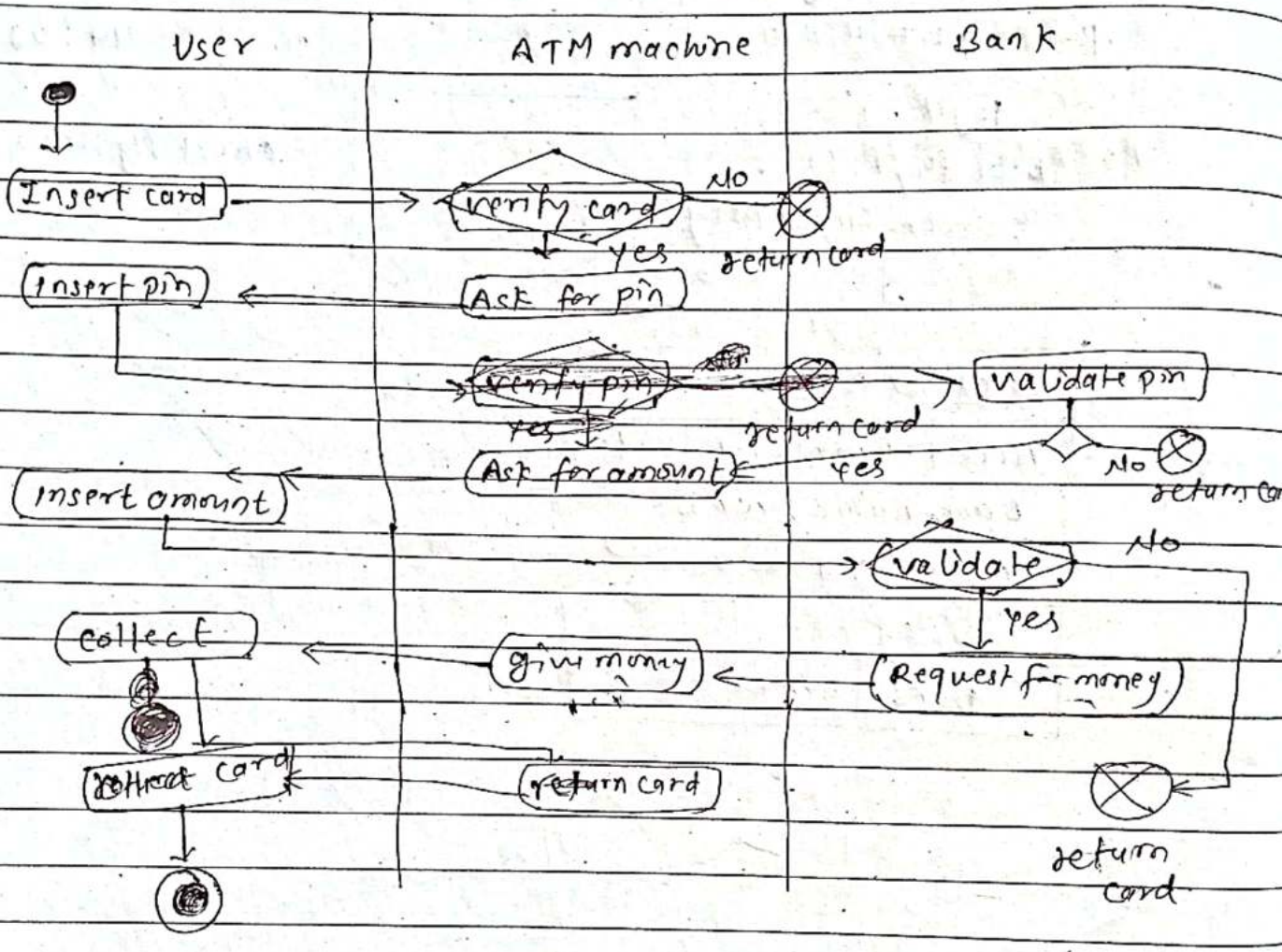
Date
Page



2080/21 22

Date
Page

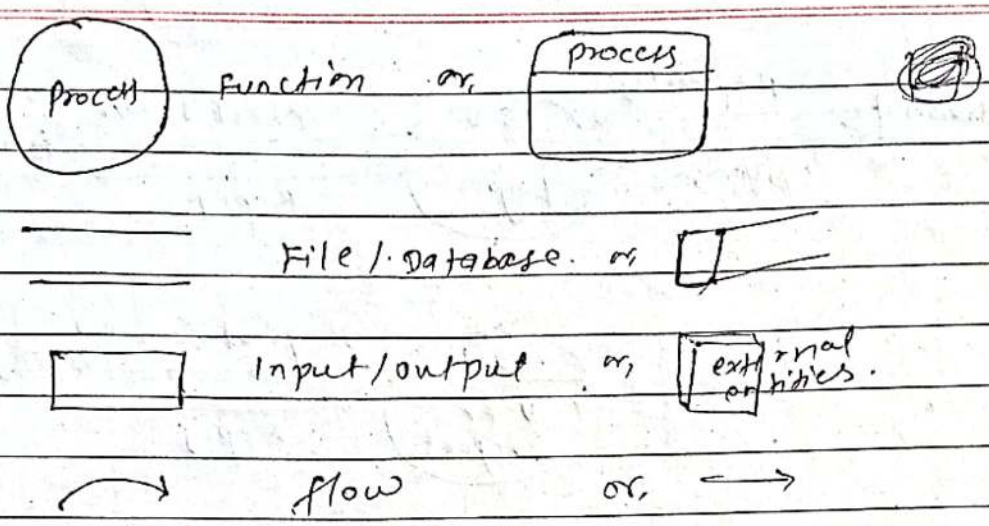
9. Draw a swimlane diagram explaining the activity performed while withdrawing from ATM.



Flow oriented modeling

2080/12/27

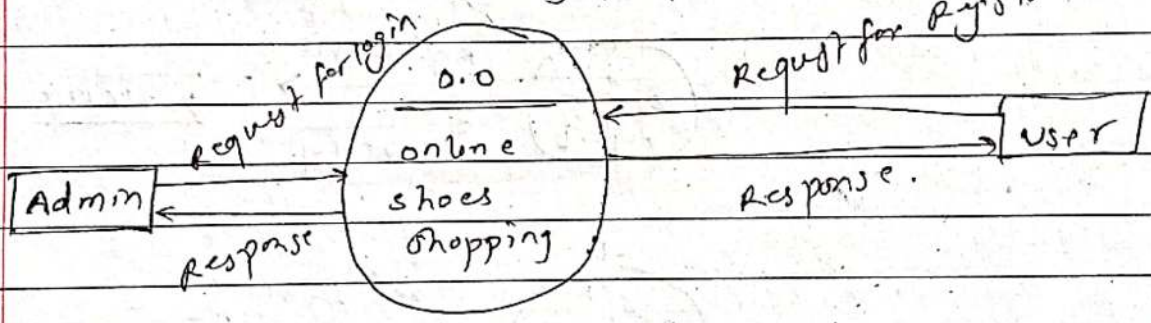
Date
Page



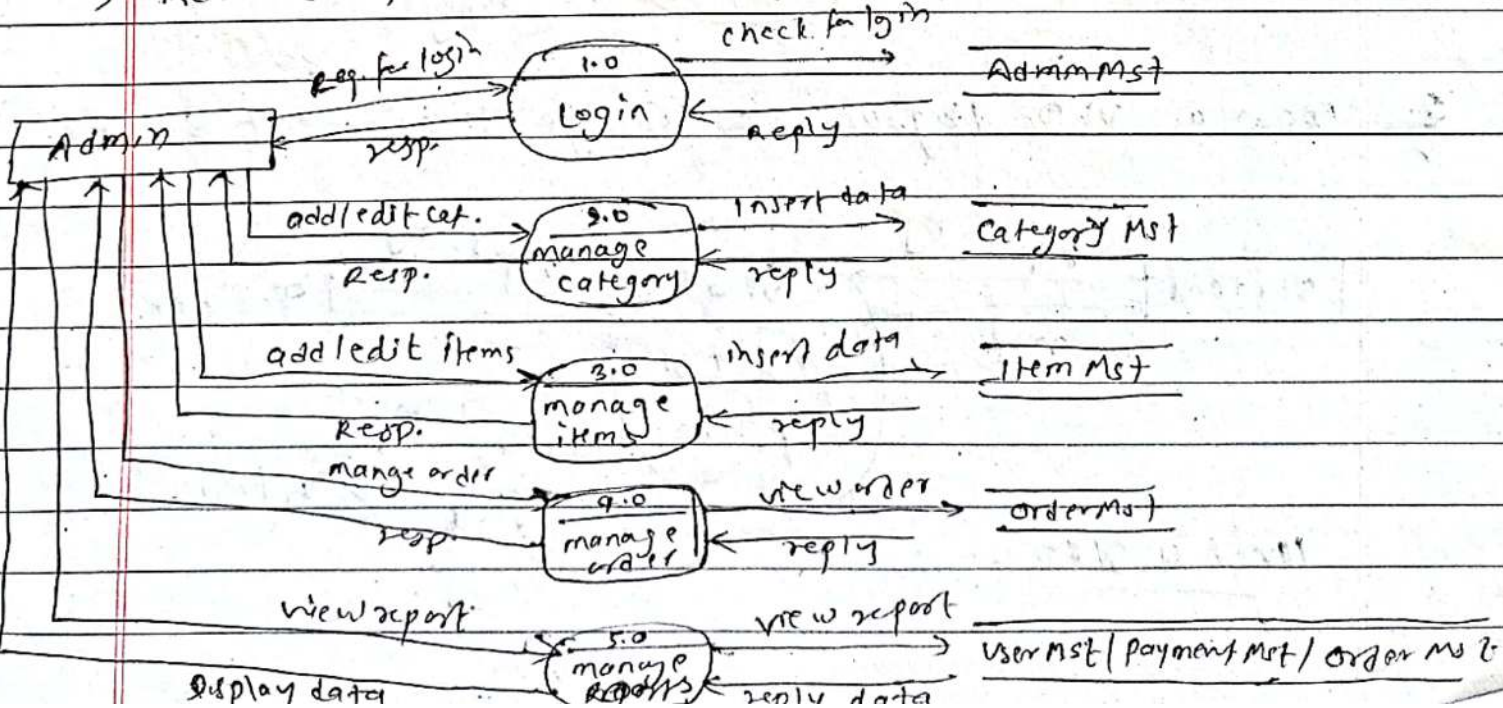
DFD (Data flow diagram)

→ level 0 DFD (CFD) (context flow diagram)

eg. online shoes shopping

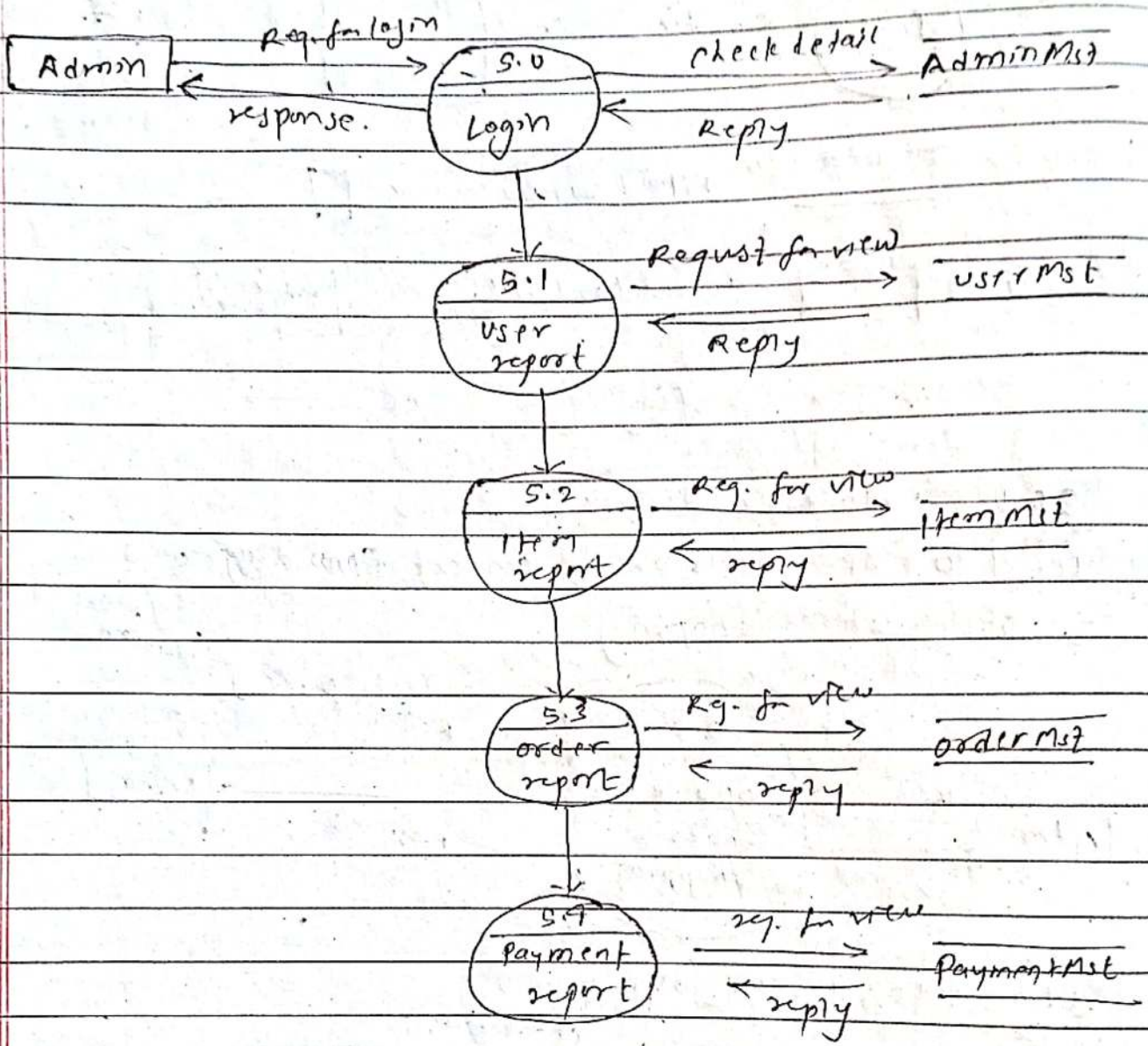


→ Admin side, level 1 DFD.

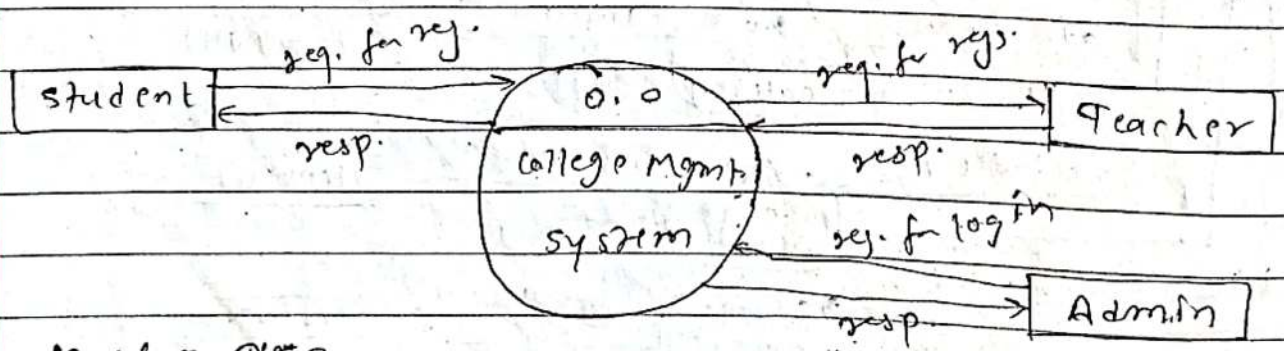


Date Page

2nd level Admin DFD - (5.0) (manage reports)



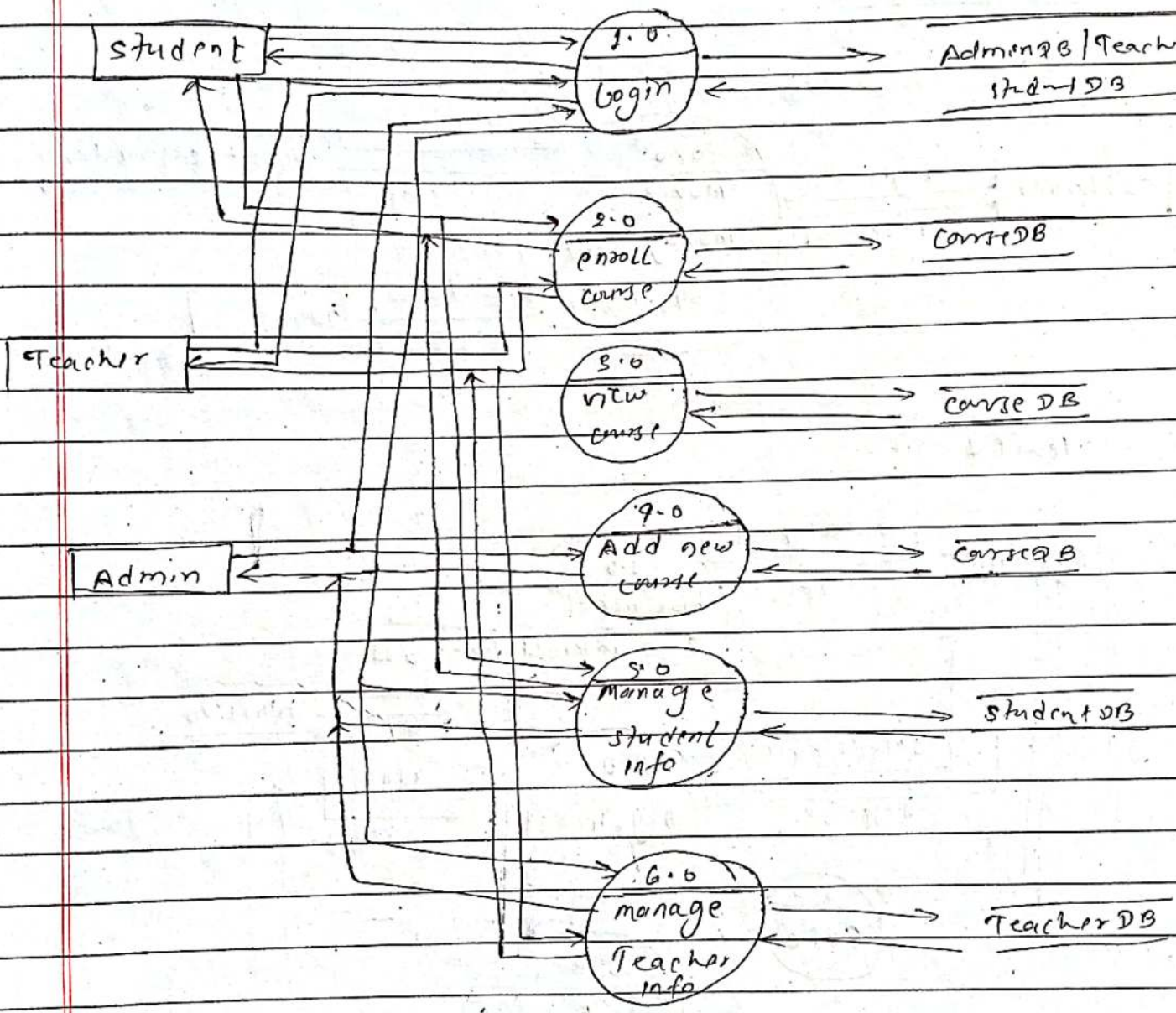
Q. Draw a DFD diagram for College Management system.



level 0 DFD.

Date
Page

level 1 DFD

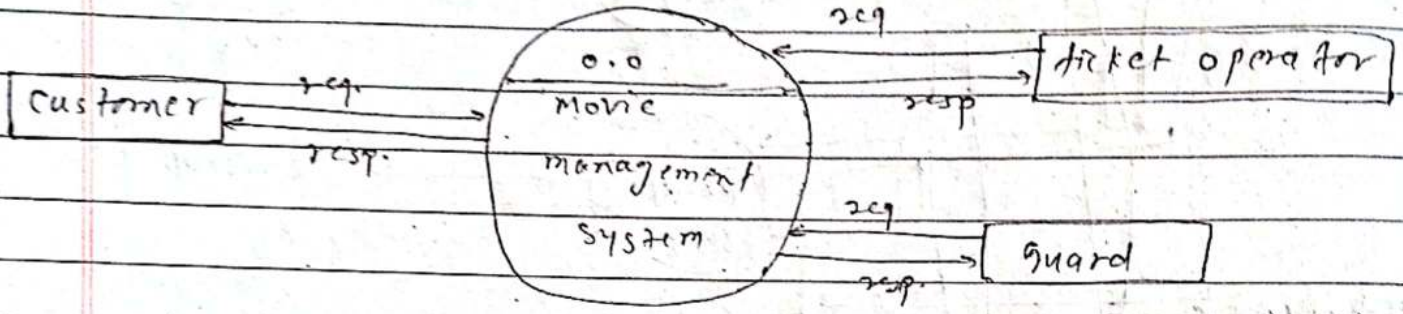


2015 spring
2016 fall

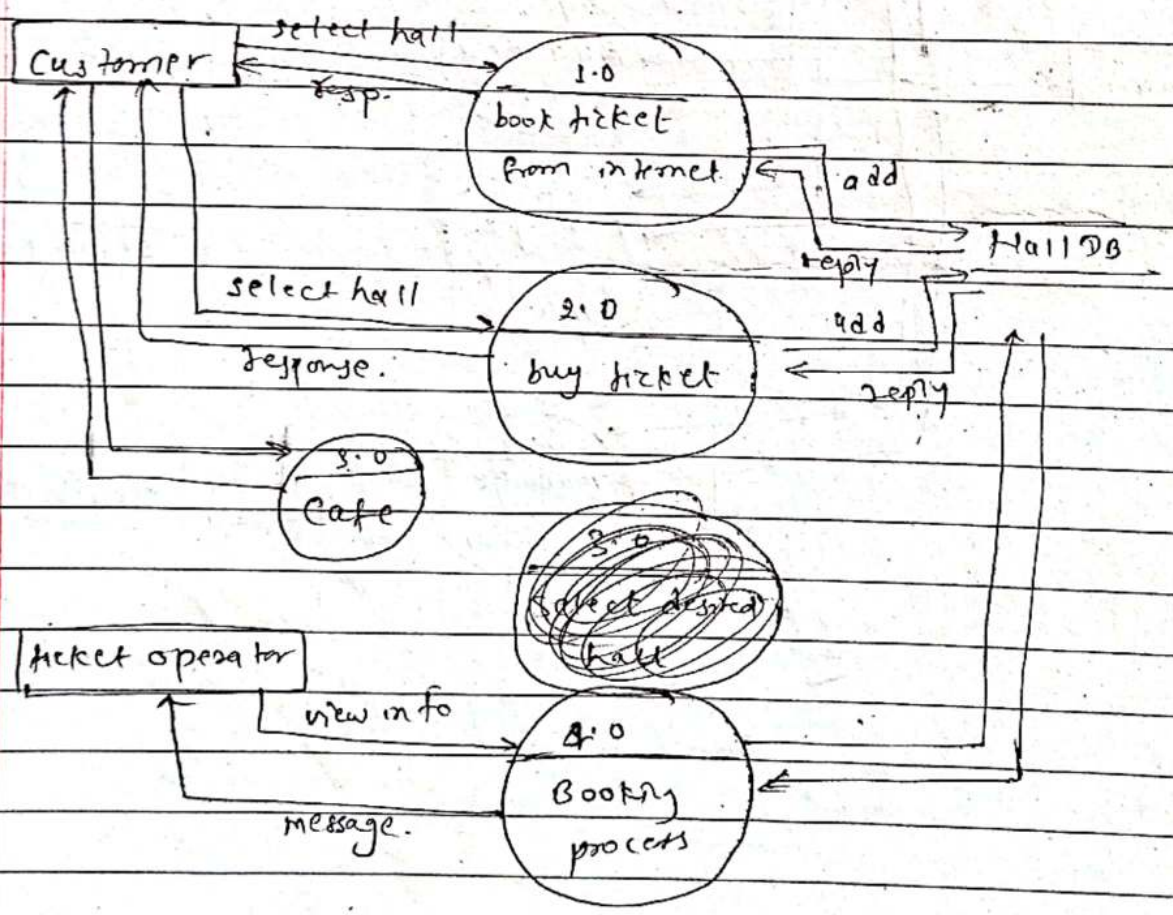
Date
Page

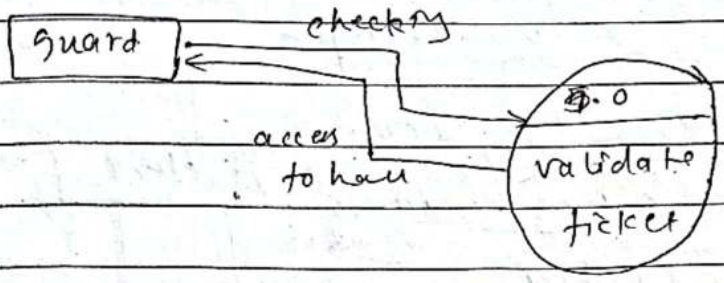
2015 spring. 3.(a)

level 0 DFD

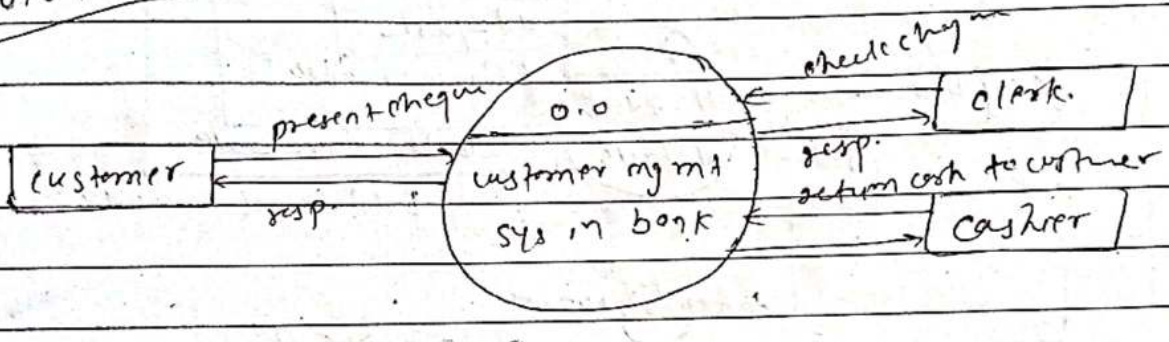


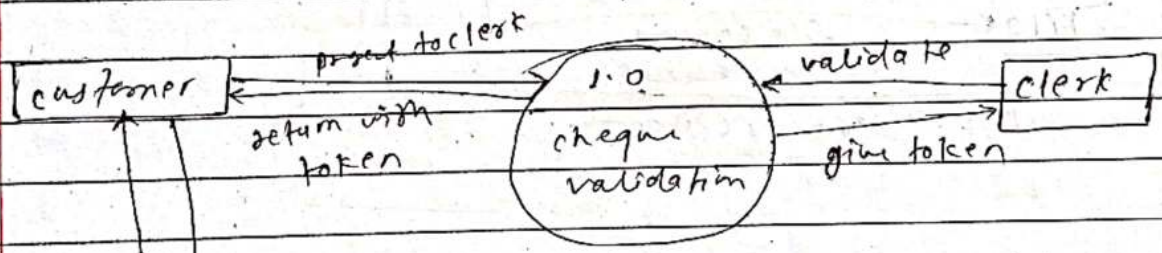
level 1 DFD

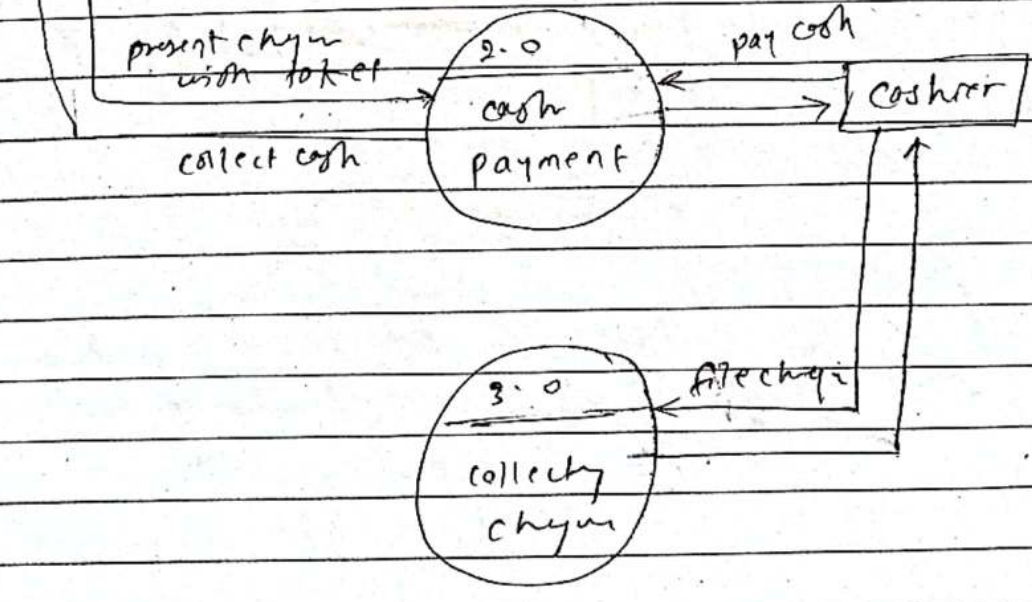




2016 Fall





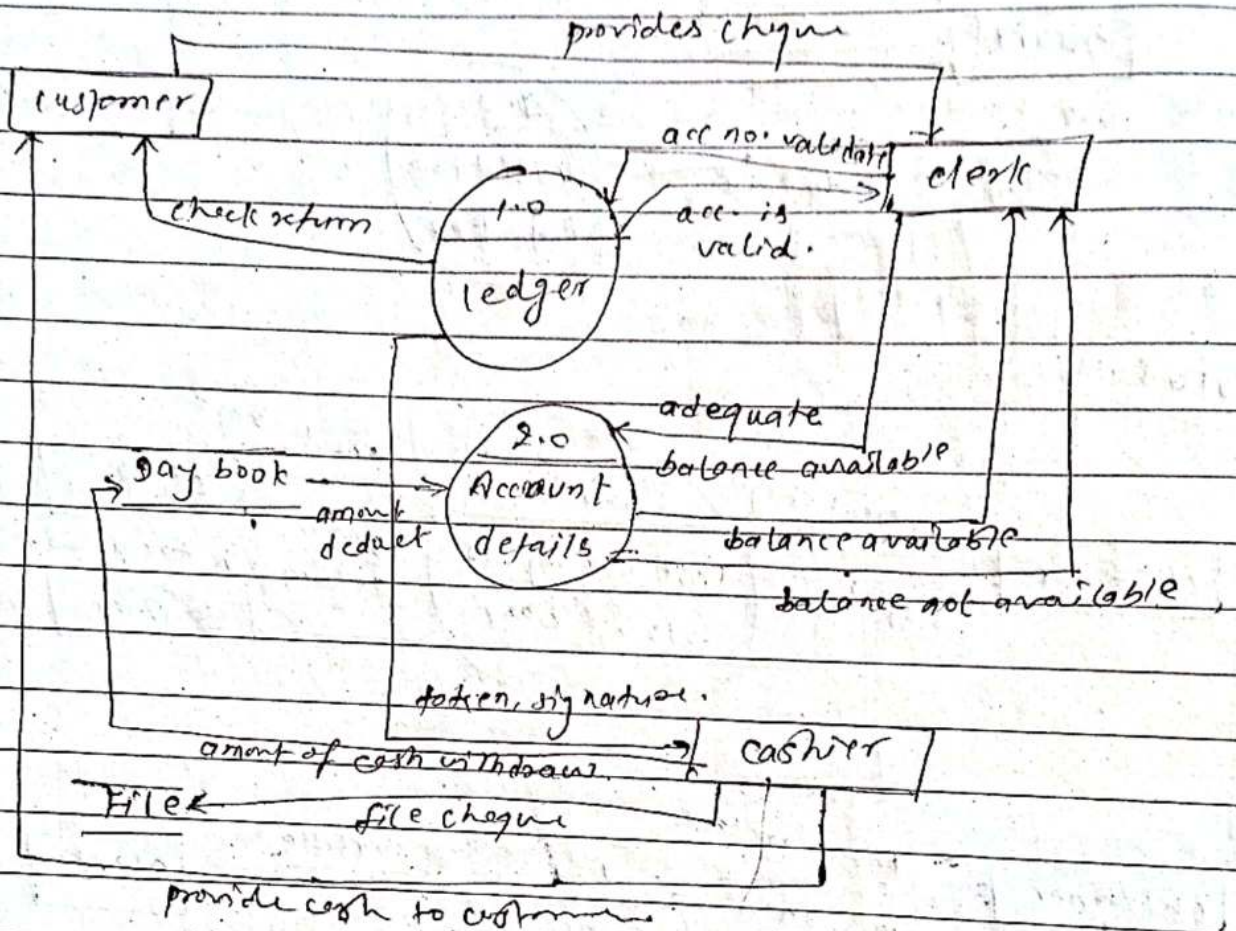


2080/21/25

Date
Page

2016 Spring

7(a)



Date
Page

→ Behavioural modeling

~~Sequence diagram~~

→ The behavioural model indicates how the softwares will respond to external events. To create such model we have to follow the following steps:

1. Evaluate all the use cases to fully understand the sequence of interaction within the system.
2. Identify events that drive the interaction sequence and understand how these events relate to specific objects.
3. Create a sequence for each of the use case.
4. Build a state diagram for the system.
5. Review the behavioural model to verify accuracy and consistency.

(#) Sequence diagram

→ It is an interaction diagram that emphesizes the time ordering of mesages. It shows a set of objects and the messages sent or received by those objects.

Graphically a sequence diagram is a table that shows objects arranged along the x-axis and messages in increasing order time along the y-axis.

Components of sequence diagram :-

1. Lifeline → represents an individual participant in the interaction. (objects)
2. Activations → A thin rectangular structure on life line represents the period during which an element is performing an operation.

3. Call messages – A msg defines a particular comm" bet" life lines of an interaction.

4. return message – It is a kind of msg that represents the pass of information back to the calling messages.

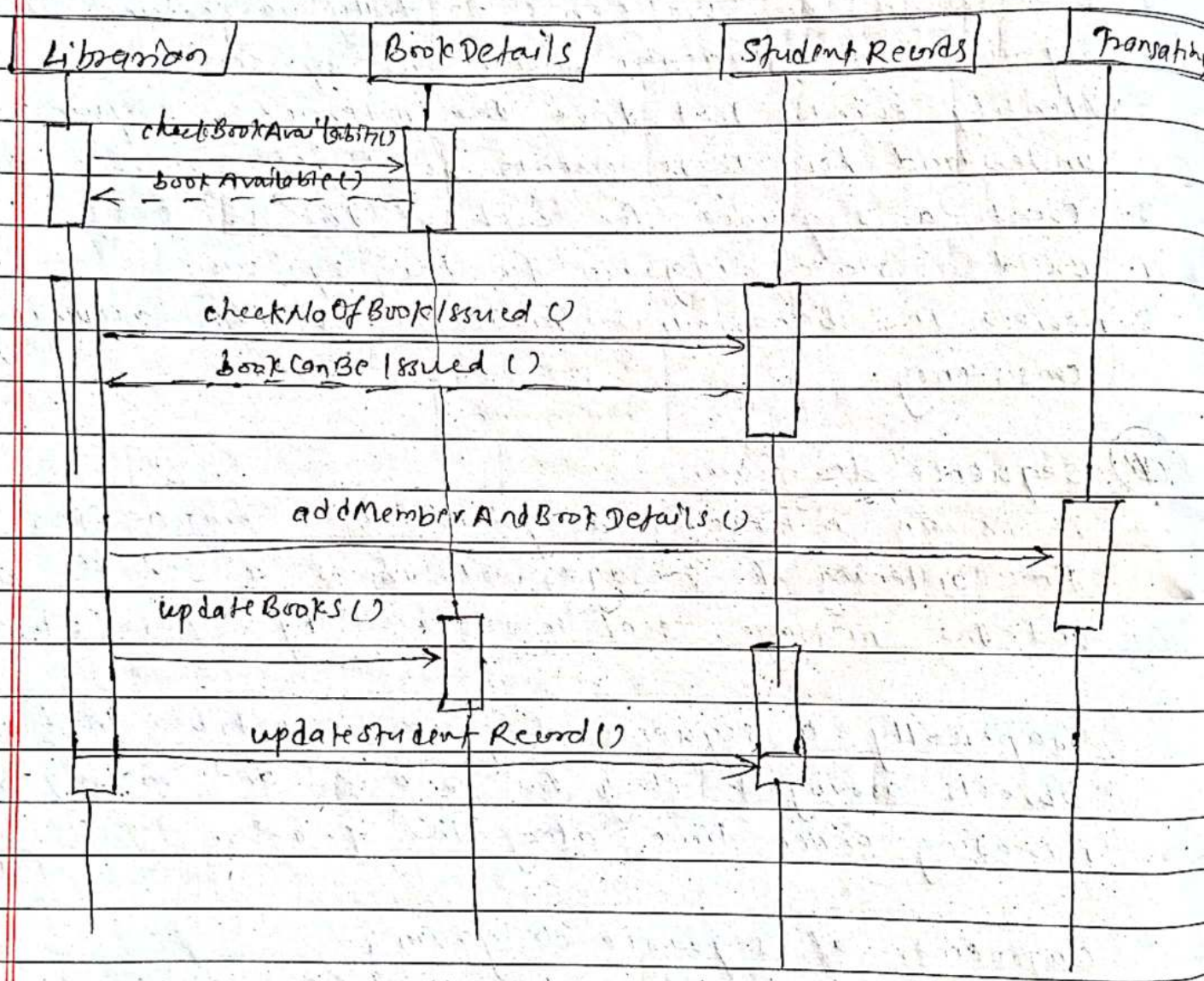


fig. sequence diagram for borrowing a book from Librarian.

Date
Page

(2) State transition diagram (STD)

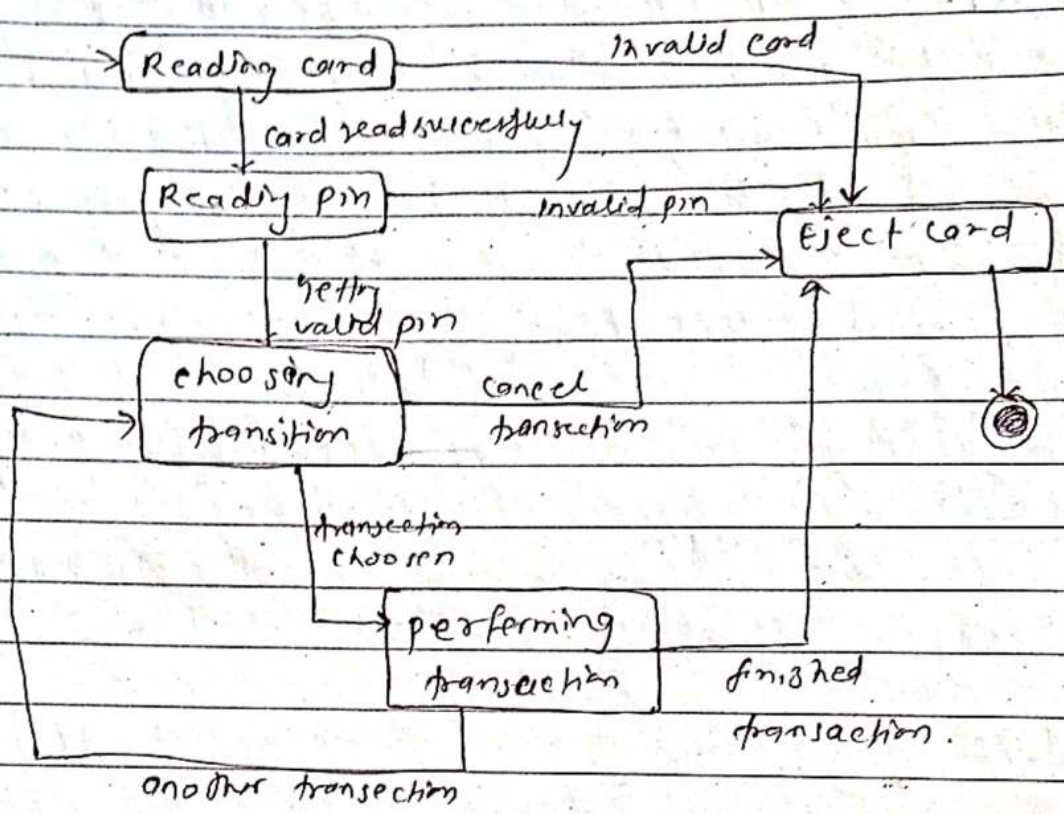
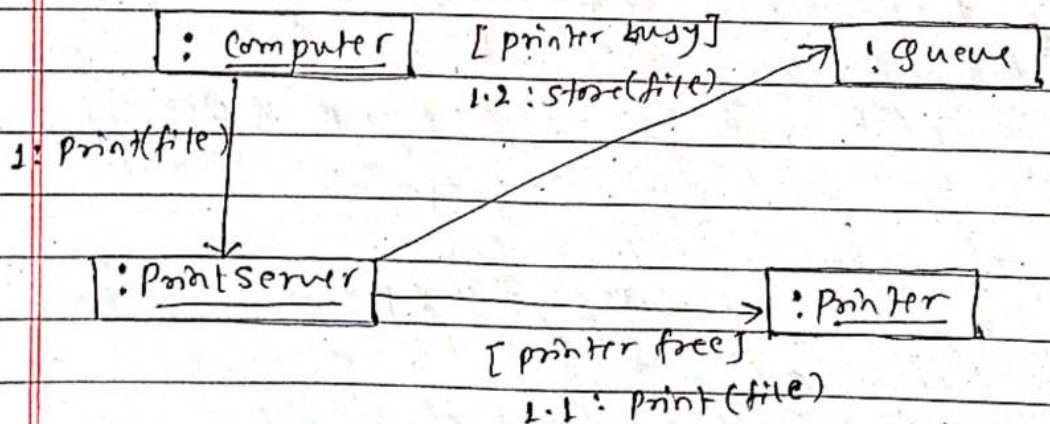


fig. State transition diagram for ATM system machine.

Collaboration and communication diagram:-



Date
Page

Design Process

McGlaughlin :- guide for the evaulation of a good design.

→ The design must implement all of the explicit requirements contained in the requirements model, and it must accommodate all of the implicit requirements desired by stakeholders.

→ The design must be a readable, understandable guide for those who generate code and for those who test and subsequently support the s/w.

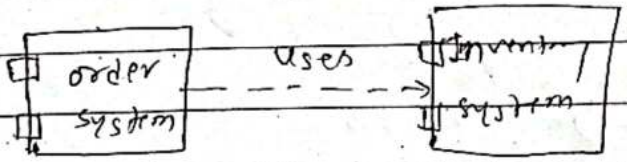
→ The design should provide a complete picture of the s/w, addressing the data, - - -

Design model

(1) Process dimension

(11) Abstraction dimension

component diagram



2080/3/9

Date
Page

The WS HH Principle →

- Why is the system being developed?
- What will be done?
- When will it be done?
- Who is responsible for a function?
- Where are they located organizationally?
- How will the job be done technically and managerially?
- How much of each resource is needed?

DFD, Activity, Sequence/State, Use case or callbss Diagram, ER diagram.
(1) (2) (3) (4)

Feasibility study. Mid Term.
↳ Technical, finance, time, resource, process of operation

The structure estimation model 2080/3/6

$$E = A + B \times (ev)^C$$

A, B & C are emprically derived constants
E → effort in person-month
EV → estimation ~~value~~ variable (LOC or FP)

COCOMO II Model
(Construction cost model)

Effort = $a(KLOC)^b$ PM
Time = $c(Effort)^d$ months
Number of people required = $\frac{\text{Effort applied}}{\text{development time}}$

Date
Page

Q.

software project	Effort schedule			
	a	b	c	d
organic	2.4	1.05	2.5	0.38
semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

organic software, 32000 LOC
Salary Rs. 15000 per month

32000 LOC = 32 KLOC

Effort = $2.4 \times (32)^{1.05}$ PM $\simeq$ 91.33 PM

Time = $2.5 \times (91.33)^{0.38}$ month $\simeq$ 13.899 months

cost = Time x Avg salary per month = 13.899 x 15000

= Rs. 208480.85

People req^{d} = $\frac{\text{(Effort applied)}}{\text{development time}}$

= 6.57

$\simeq$ 7 persons.

2080/3/20

Date
Page

Software Quality Assurance (SQA)

→ It is defined as a planned and systematic approach to the evaluation of the quality of software product standards, processes, and procedures.

The basic elements of SQA:

- Standards
- review and audit
- testing
- error collection and analysis.
- change management
- education and training
- vender management
- Security management
- Safety and risk management.

Measure of reliability and Availability. 2080/3/21

MTBF = MTTF + MTTR ← Mean Time To Repair

↑ MeanTime To Failure

↑

MeanTime Between Failure

$$\text{Availability} = \frac{MTTF}{MTTF + MTTR} \times 100\%$$

2080/3/23

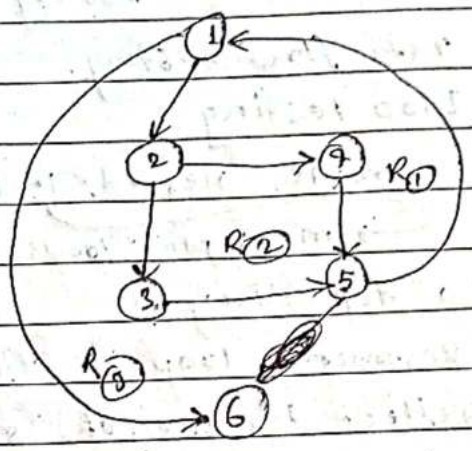
Date
Page

White Box Testing

a) Basic path testing

given code

```
(1)  while (n > y) {
(2)    if (n == 3), then
(3)      print ('- - -')
       else
(4)      print ('- - -')
(5)    n++
(6)  }
```



cyclometic complexity for a flow graph (G) is defined as

$V(G) = E - N + 2$

where, E is the no. of flow graph edges and N is the no. of flow graph nodes.

cyclometic complexity is also defined as $V(G) = P + 1$

where, P is the no. of predicate nodes in the graph.

No. of region $R = 3$

So, $V(G) = R = 3$

Paths

1 - 6

1 - 2 - 3 - 5 - 1 - 6

1 - 2 - 4 - 5 - 1 - 6.

For the above given path, that starts from a starting node to the exit node must have all the test cases define for each path. The error in each path are calculate and the most optimum path is ~~kept~~ unchanged and path with the most probability error is revisited for a better change.

b) control structure testing
c) data flow testing
d) Loop testing
simple, nested, concatinated
→ multiple loops are used simultaneously where there is no dependency.
unstructure loop → This class of loop are designed to reflect the use of structural programming construct.
The loop testing is a white box testing technique that focuses on the validity of loop constructs.

Black Box Testing

Black Box Testing tends to be applied during the latter stage of testing (integration & system testing).
(1) Equivalents partitioning
(2) Boundary value analysis
↳ a) Testing GUI
b) Testing of client–server architecture
c) Testing for real-time system

Software configuration management (SCM)

SCM is also called change management where a set of activities are designed in such a way that the change in the system can be properly managed by identifying the different products change, & establishing relationship among them and defining mechanism for the different product (versions) of the change.

Date
Page

SCM activities are developed to
(I) Identify change
(II) Control change
(III) Ensure that change is properly implemented
(IV) Report changes to others who may have a interest.

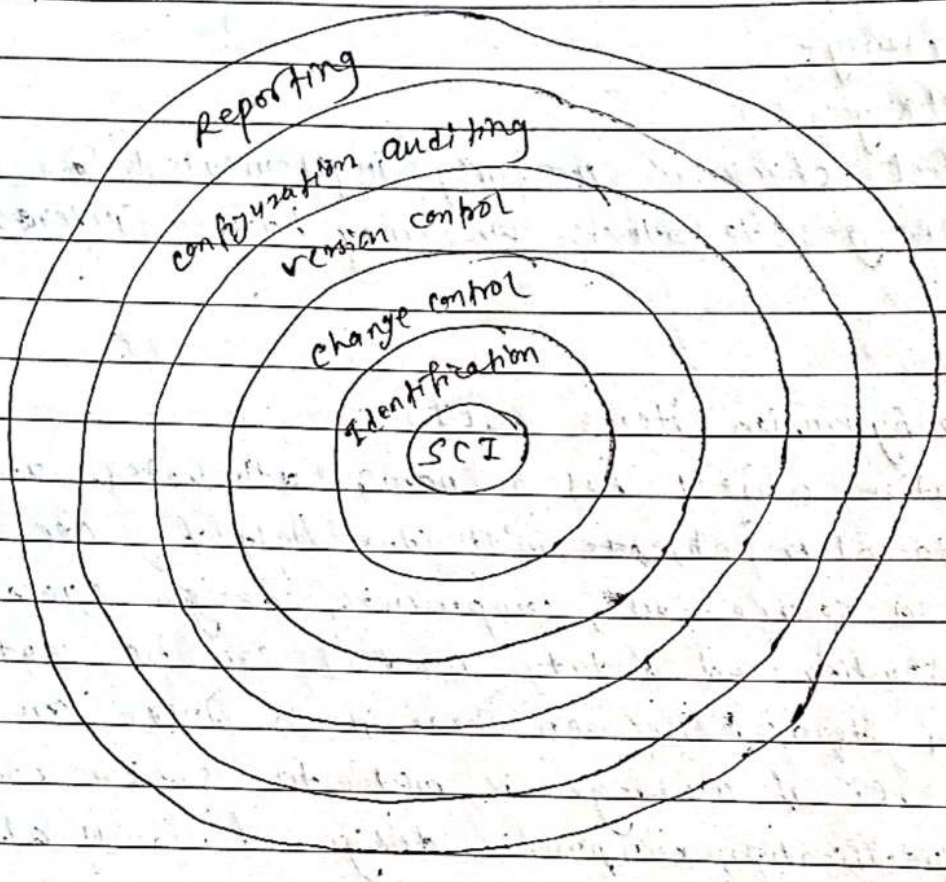
Baseline

Software configuration items (SCI)
A configuration object has a name, attributes, and are connected to other objects with the help of certain relations. For example, a source code, components, design specification, test specification and a data model can be software configuration items. Between these items there are certain relationship ie. if a change is made to source code then the test specification, components design etc. will also be changed.

SCM process
The main objectives of SCM are
(1) To identify all the items that collectively define the software configuration.
(2) To manage changes to one or more of these items.
(3) To facilitate the construction of different versions of an application.
(4) To ensure that the SW quality is maintained as the configuration evolves.

Date
Page

Layers of the SCM



20801312T

$$ROI = \frac{\Sigma \text{benefits} - \Sigma \text{costs}}{\Sigma \text{costs}} \times 100\%$$

where,

benefits = cost saving changes, reduced effort changes and the income that is generated from the market. and cost includes the direct SPI cost and indirect cost with quality control and change management.