

Use case diagram

5 relationships type in a use case diagram

- Associate between actor and use case
- Generalization of an actor
- Extend between two use cases
- Include between two use cases
- Generalization of a use case

Feasibility Analysis

Business value: Tangible value can be quantified and measured easily → 2% reduction in OC
5% increase in sales

⇒ Intangible value - system provides important but hard to measure benefits.

- improved customer service
- better competitive position

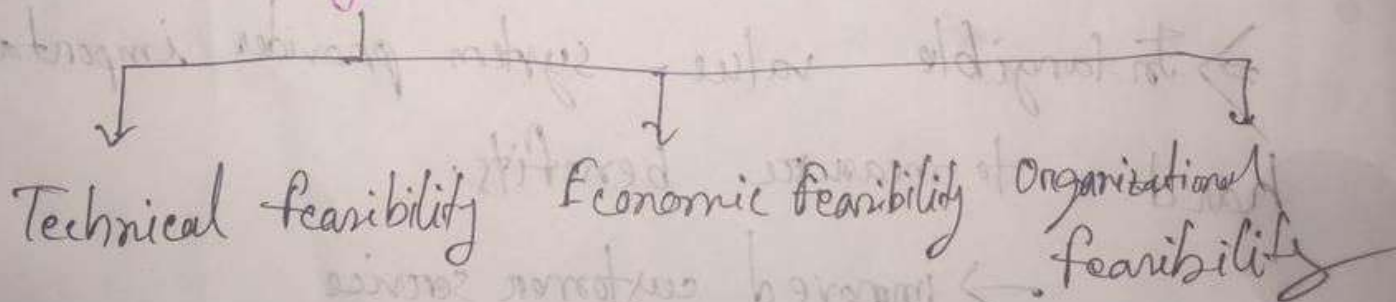
Feasibility analysis is used to aid in the decision of whether or not to proceed with the IS project.

As name suggest feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view.

→ Also identifies project risks

→ Can be revised throughout SDLC

Types



Technical Feasibility:

→ Current resources both hardware software along with required technology and analyzed to develop project.

Family with the application

" " " " Technology

Project size

Compatibility

Economic Feasibility (Study of cost and benefits of the project)

→ Perform cost benefit analysis

→ Identify cost and benefits

→ Assign values

→ Calculate cash flow and ROI

→ Development cost

→ Annual operation cost

→ Annual benefits

→ Intangible costs and benefit

Full - 1.5 40mm 50mm 300 0
C/A + M
C/A + A
O = 29%
Flow = 29%

(ROI) Return of Investment

Measure money received in return for money
Invested.

→ High ROI is desirable when benefits exceed costs.

→ Can be determined per year, or for either project completion period.

$$ROI = \frac{\text{Total } \cancel{\text{benefit}} - \text{cost}^{\text{Total}}}{\text{Total cost}}$$

Break - Even point :-

Length of time when returns will match amount
invested.

→ Greater time - Greater risk

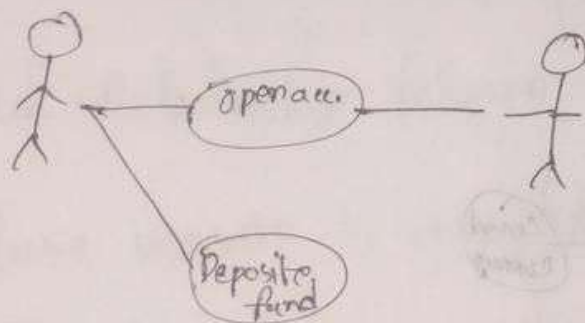
Organizational Feasibility

Perform Stakeholder Analysis

Stakeholder — any person, group or organization that can affect or will be effected by the system.

Project initiation:- involved creating and assessing goals and expectations for a new system

Use case diagram relation



→ Actor must be associated with at least one

use case

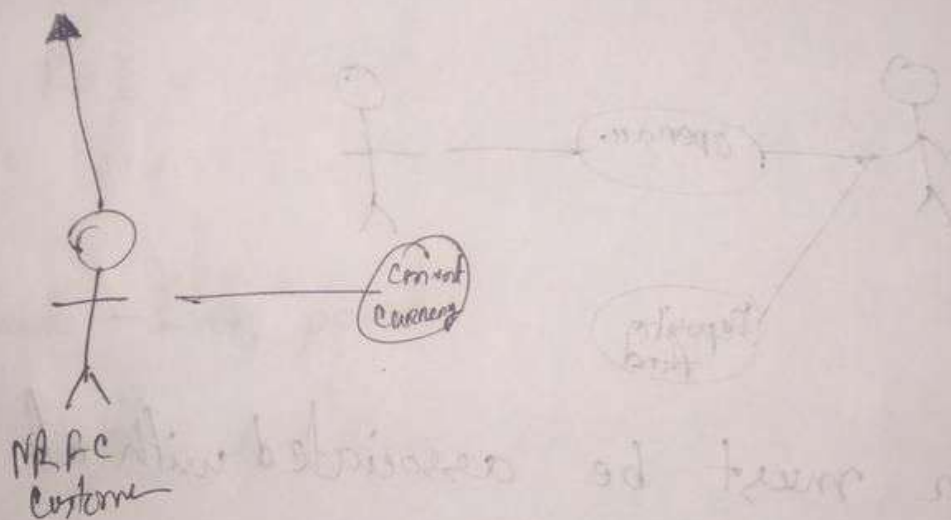
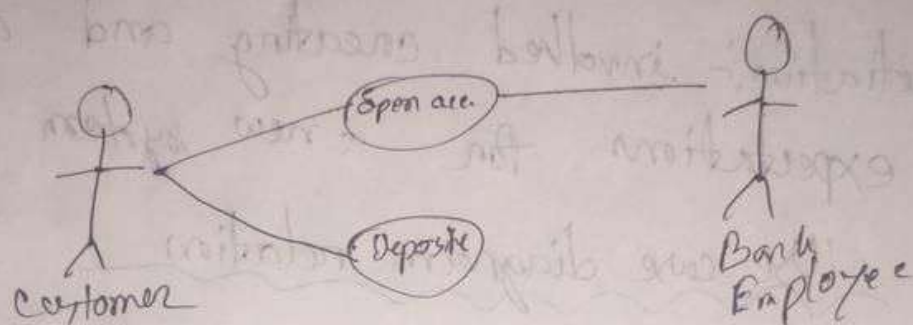
→ ... can be ... multiple use cases

→ Multiple actors can be associated with a single

use case.

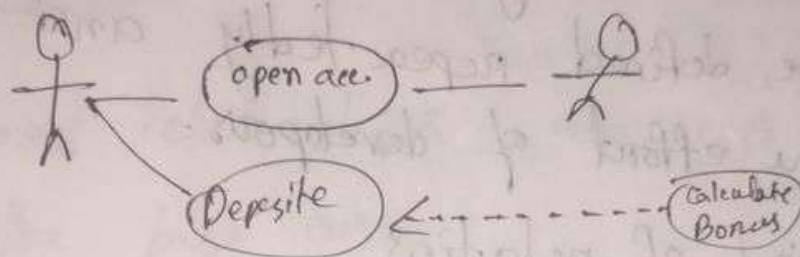
Generalization of an actor

One actor can inherit the role of other actor. The descendant inherit all the use cases of the ancestor.



Extend Relationship between two use cases

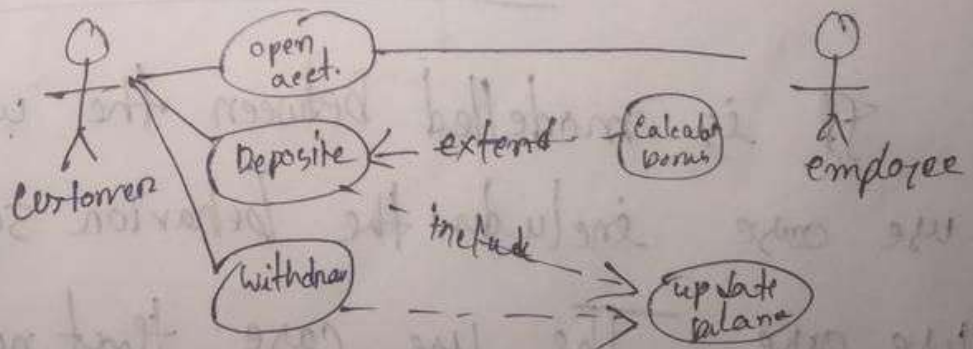
- It extend the base use case
- depend on the base use case
- optional
- must be meaningful



Include Relationship between two use cases

→ base use case is incomplete without a included use case.

→ The include use case is mandatory and not optional.



Use case Relationship

Use case relationship models the dependency between the use cases in the interaction model of a system.

→ Establishing relationships between the use cases allow reusing those use case which needs to be defined repeatedly and this reduces the effort of developers.

kind of relations

→ Include

→ extend

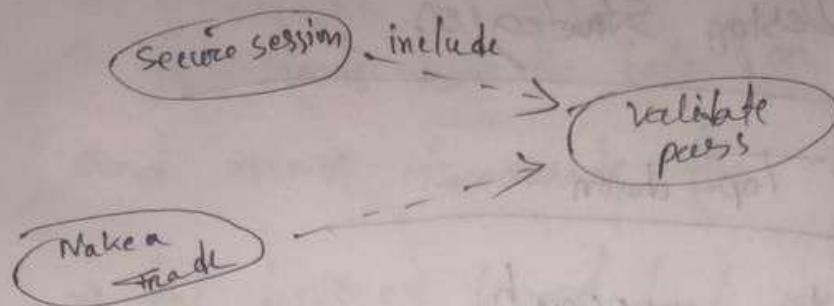
→ Generalization

Include use case

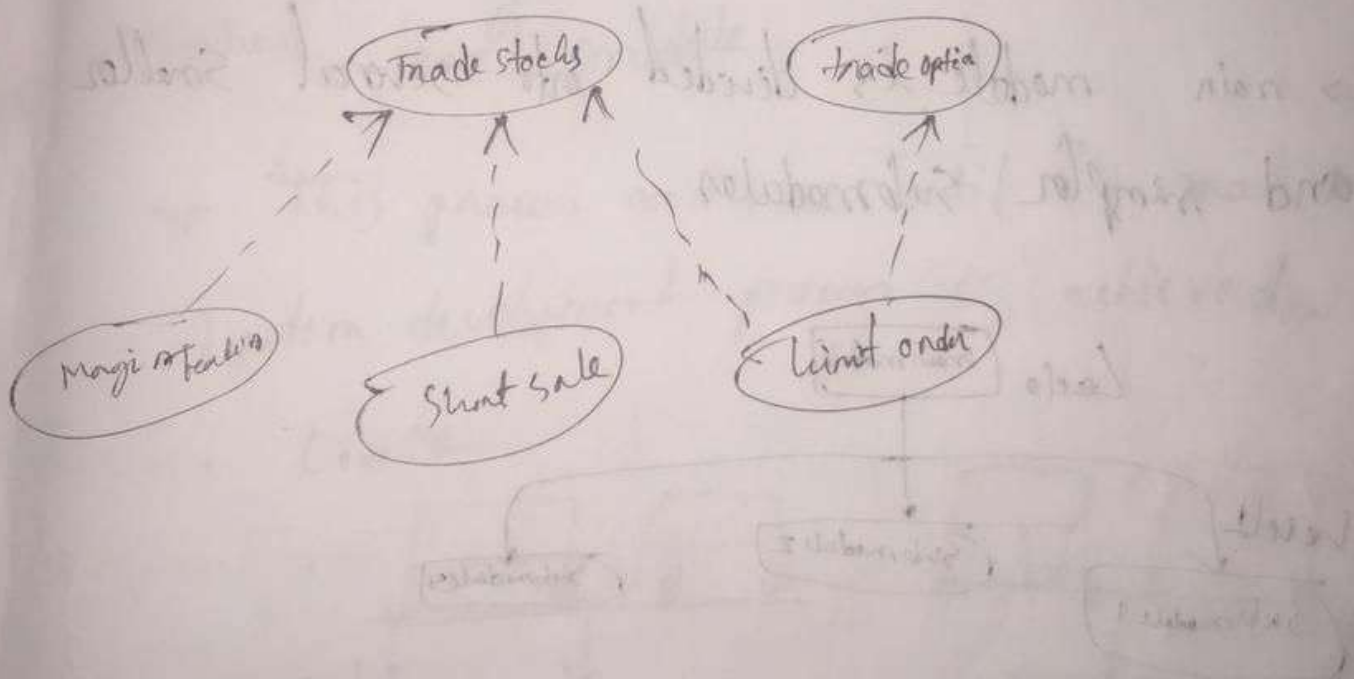
It is modelled between the use cases when a use case includes the behavior sequence of another use case. The use case that need to be described repeatedly for a complex and a large system are

modelled and included in the other use cases when required.

→ Similar to subroutines



Extend Relationships : adds up behaviour square to the base case.



Generalization

Design Strategies

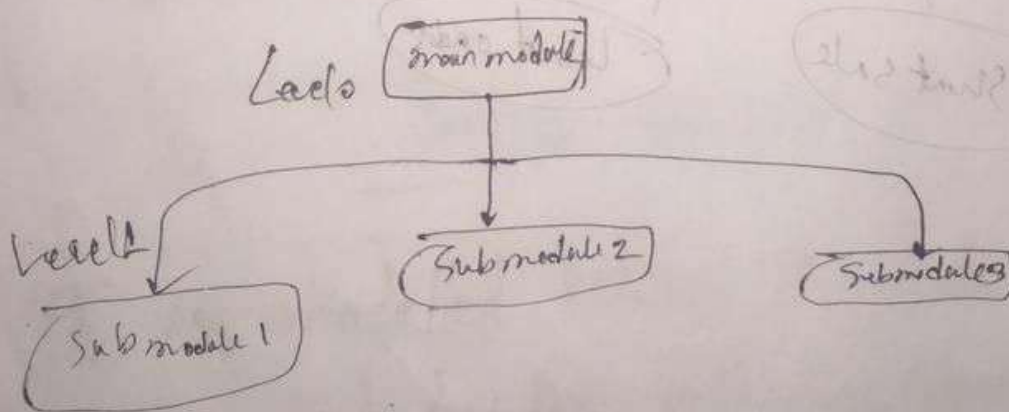
Top-down

→ use modular approach

→ It starts from the top on the highest

level module

→ main module is divided into several smaller and simpler submodules



Bottom up

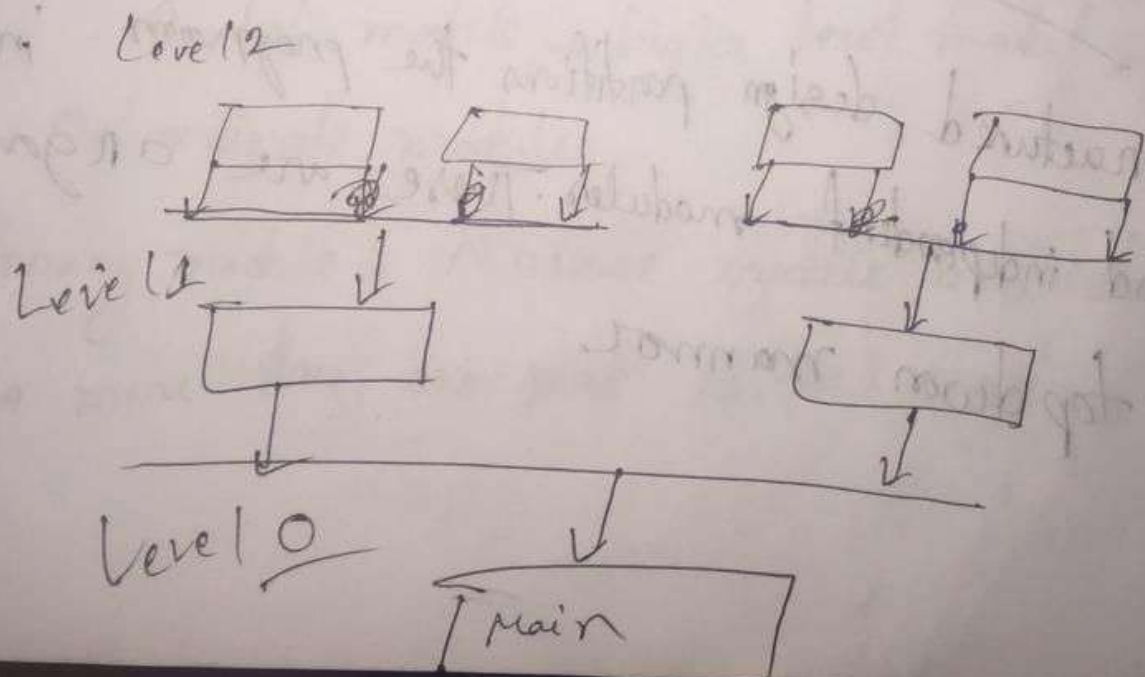
It follows the modular approach to develop the design of the system

→ It starts from the bottom on the most basic level and moves forwards the highest level.

→ the basic or lowest level are identified

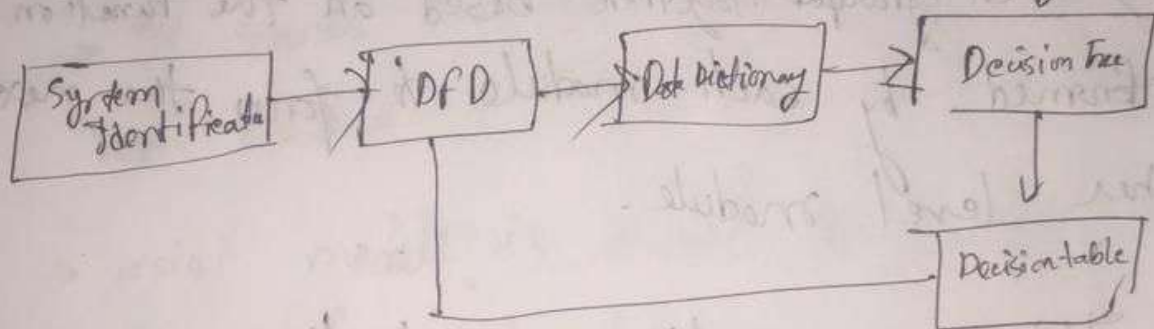
→ then grouped together based on the function performed by each module to form the next-higher level module.

→ This process continues until the main module of system development process is achieved.



Structured Design

→ data flow based methodology that helps in identifying the input and output of the developing system. The main objective of Structured design is to minimize the complexity and increase the modularity of a program.



Modelizzazione

Structured design partitions the program into small and independent modules. These are organized in top down manner.

to minimize the complexity and to manage the problem by sub dividing it into smaller segments.

Advantages:

- critical interfaces are tested first
- provide abstraction
- allows multiple programmers to work simultaneously
- allows code reuse.

Structure chart

→ recommended tool for designing a modular top down system.

Control module, → higher level module, called Subordinate modules.

Library module: reusable module and can be invoked from more than one point in the chart.

Two different approach

tutorial's point -- live in class room

System audit

tutorial's point

Activity diagram

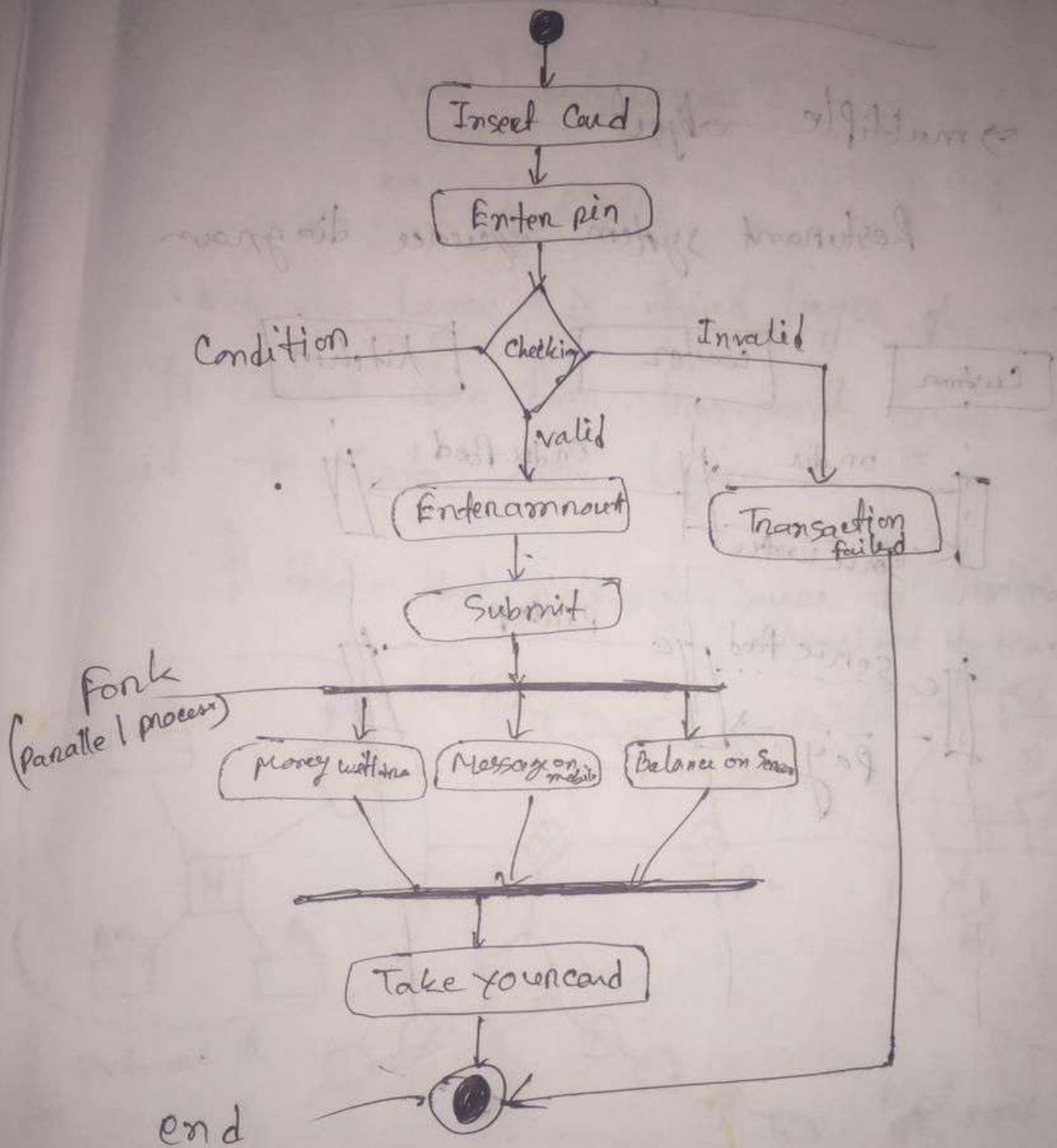
→ It comes under the behavior diagram of the UML.

→ dynamic aspects

→ flow of actions

→ The activity diagram is a flowchart to represent the flow of control among the activity in a system

ATM money withdrawal



Sequence diagram

multiple object

Restaurant system Sequence diagram

