Name :- Rafj Hany Said
iD :- 21010504

1)

a)

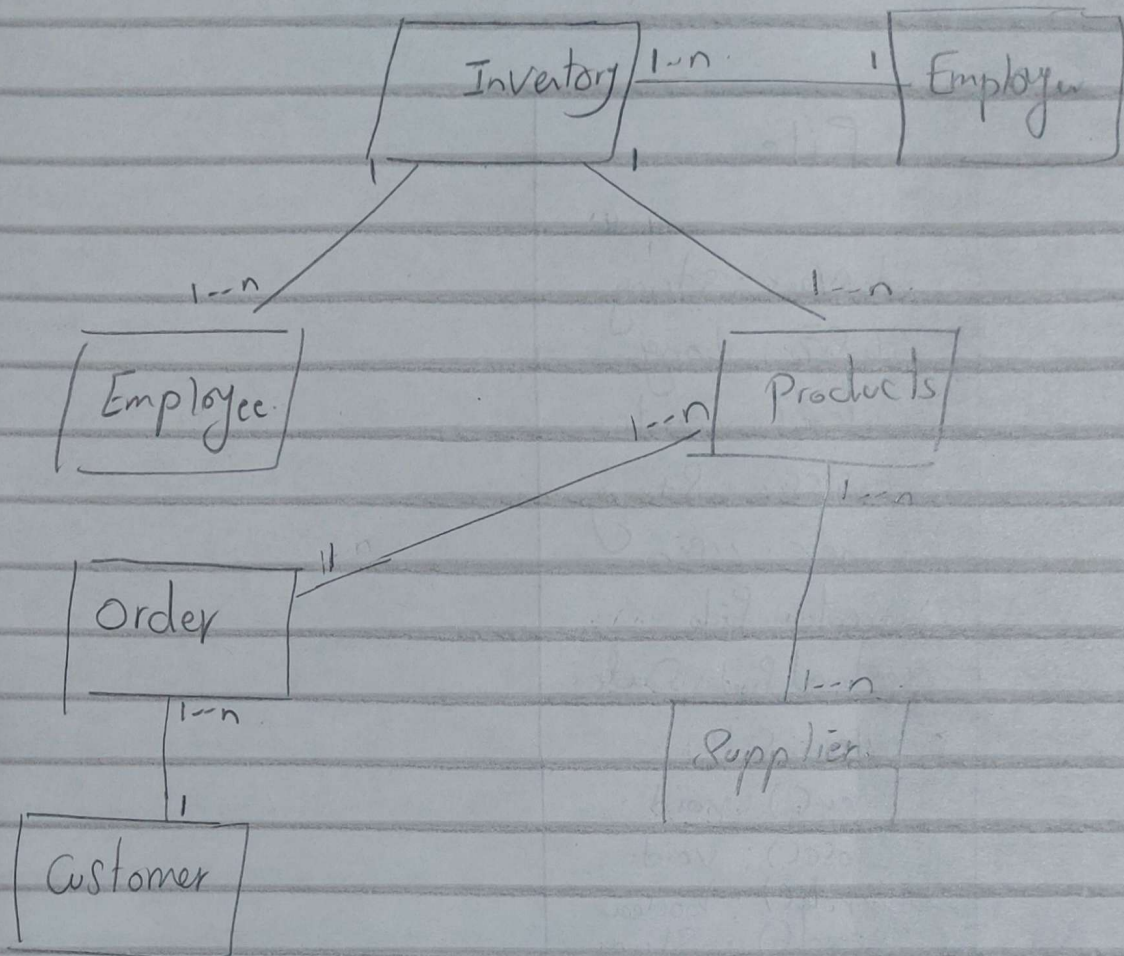| File |
| --- |
| - FileName : String.
- FileSize : Long.
- CreateAT : Date.
- Permission : String
- owner : User
- Parent : File
- Lastmodified : Date. |
| + getter, Setter.
+ open() : void
+ close() : void.
+ write() : boolean
+ read() : String
+ Lock() : |

b)

LoadJob
- File : File - Path : String. - buffer : Buffer
+ getter, Setter. + get Fib into memory() + return file () : void

c)

PlayingTrack
- TrackName : String - Content : Byte[] - Artist : String - duration : int.
+ getter, Setter. + Play () : Byte[] + Stop() void

2)

```
        ┌──────────┐ 1-n            1 ┌──────────┐
        │Inventory │─────────────────│ Employer │
        └──────────┘                  └──────────┘
           1 │         │ 1
      1-n   /           \  1-n
    ┌──────────┐      ┌──────────┐
    │ Employee │      │ Products │
    └──────────┘      └──────────┘
                  1-n /      │ 1-n
              ┌──────────┐  │
       11-n   │  Order   │  │ 1-n
              └──────────┘  ┌──────────┐
                1-n │       │ Supplier │
                  1 │       └──────────┘
              ┌──────────┐
              │ Customer │
              └──────────┘
```

3)

Classes :-
- use design patterns.
- Solid principles
- inheriatance
- Encapsulation
- Polymorphism.
- Abstraction

Methods :-
- Naming Convention
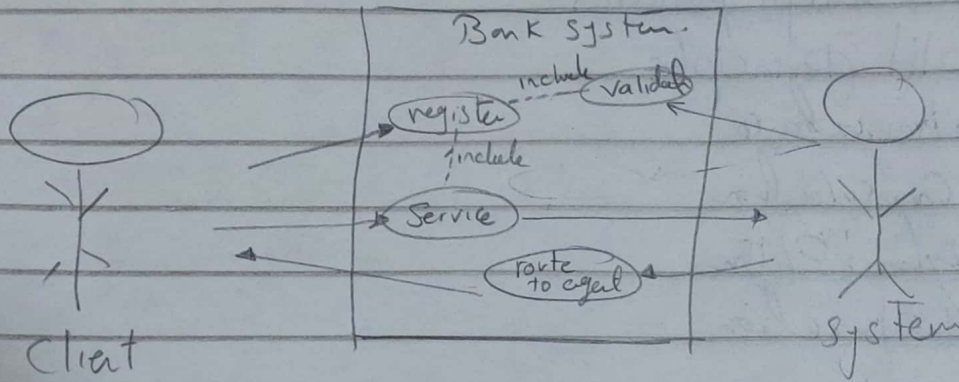- Single functionality
- Short
- not nested inclutation.

Variable :-
- not use magic number.
- boolea var not preferable.
- Naming Convension
- decrease life time, Scope.

1)

a) Bank

Use Case

Bank System

register — include — Validate

include

Service

route to agent

Client

System

Functional req.
- Authentication System
- System routing to agent
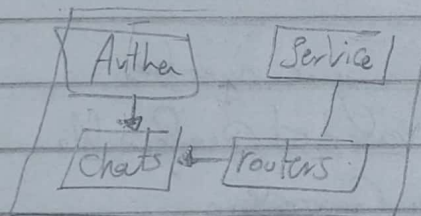- Chat between agent, Customer

non functional requirement
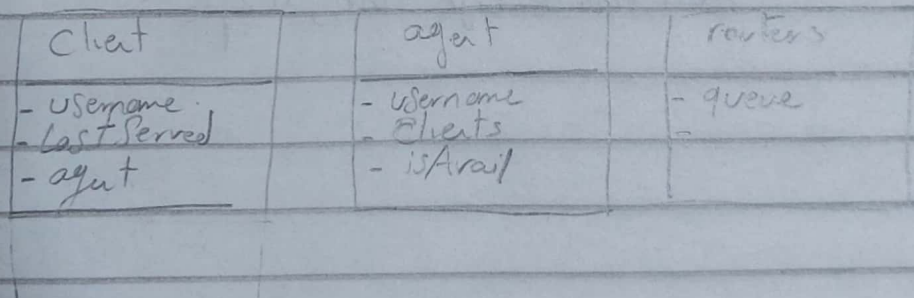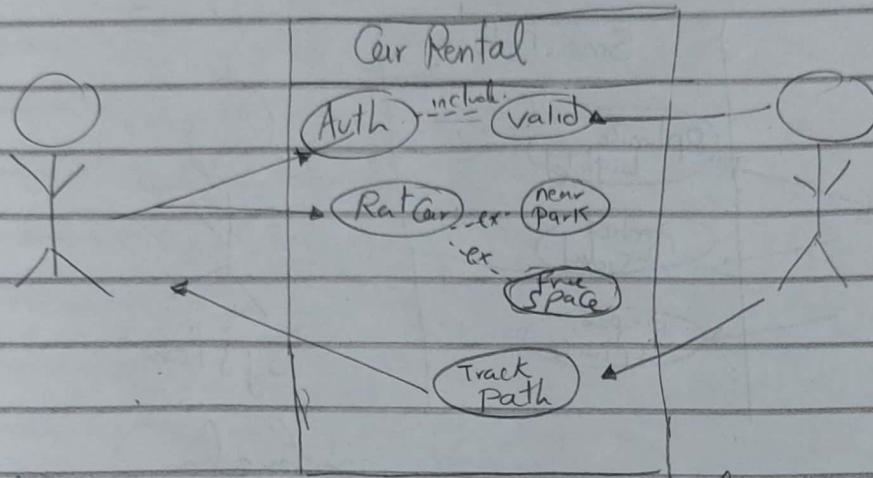- Security
- response time
- availability

Arc desig

Layered Architecture

Auther | Service

Chats ← routers

class Digram

Client	agent	routers
- username	- username	- queue
- Last Served	- Clients	-
- agent	- isAvail	

b) Car Rental Service.

UseCase

Car Rental

Auth ---includ--- Valid

RatCar --ex-- near park

--ex-- free space

Track Path

functional	non functional.
- get nearest Parking	- Security
- rent Car.	- response time
- Tracking Path.	
- Notification	

Architecture design

Distributed Architecture. (descenteralized database)

Class diagram.

User	Parking	Reservatn
- usernany.	- Location.	
- paymetinfo	- trackes aval	

c) Smart home manger :-

use Case

Smart Home

Optimize Lighty

Predict Resident

Prepare appliance

User

System
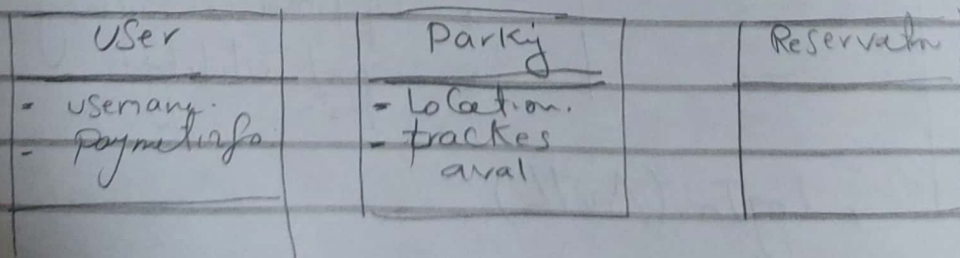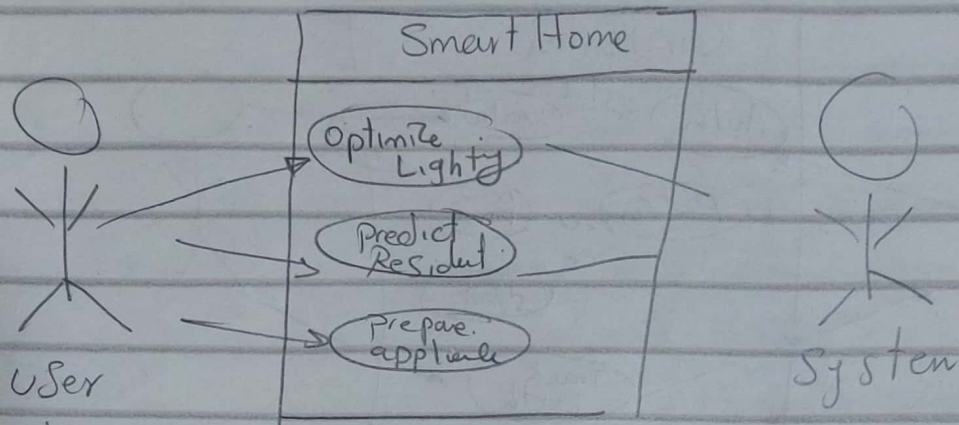
functin
- Control appliance.
- optimize Lighty
- Usage utility

non funcln
- Security
- Response time
- depadability
- Availity

Architecture
   micro Service architecture.

classes

  ⌐> user
  └─▷ Service. ──▷ Contole appliance
        └▷ optimize Lighty
        └▷ report utility
  └─▷ LogIn.(Auth)