Functional & Non-Functional Requirements:

I List of Experiments:

1) Identify functional and Non-functional Requirements for

i) Online Ticket Reservations for Railways

(ii) Online Auction sales.

(i) Online Ticket Reservations for Railways:

Functional:

- Every online booking needs to be associated with an accou
- One account cannot be associated with multiple users.
- Search results should enable users to find the most recent and relevant booking options.
- System should enable users to block/pay for their tickets only when mandatory fields only in time boxed manner after tickets being added to cart.
- System should allow users to move to payment only when mandatory fields such as date, time, location has been mentioned.
- System should consider timezone synchronization when accepting bookings from different timezones.
- Booking confirmation should be sent to user to the specified contact details.

Non-functional:

- Use of captcha and encryption to avoid bots from booking tickets.
- Search results should populate within acceptable time limits.
- User should be helped approximately to fill in the mondatory fields, incase of invalid input.
- System should accept payments via different methods, like PayPal, wallets, cards, vouchers etc.
- System should visually confirm as well as send booking confirmation to the user's contact.

(ii) Online Auction Sales:

Functional Requirements:

- Only users with administrative login can access the control admin page and have full access right to the system.
- Basic account users can only view sales and bid.
- Basic account users are recognised by the strength of their bought items and sellers feedbacks.
- Seller account users are recognized and can only sell items and have to pay a subscription fee.
- Seller amount users are recognised by how many positive feedbacks and stars a seller has.
- Advanced account users can only sell, buy or trade items, advanced amount users are recognised by how well is their feedback from both sellers and buyers on the auction system, i.e. advanced account member that has 120 sales to his/her account with all positive feedback and

stars then this seller/trader is very good.

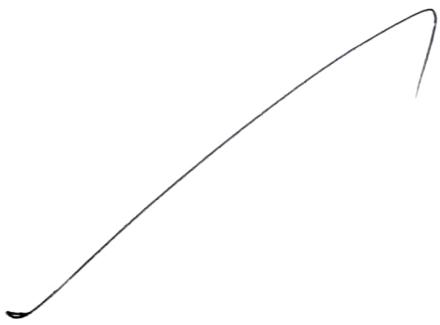
- Allow visitors to signup for account online, Login and Logout to the system options.
- New members have the option to upgrade to seller or advanced account.
- Admin user can create accounts and delete accounts.
- Basic account users can contact a seller privately.
- Sellers can create their own discussion forum for the item on sale.
- Enabling a remainder on an item for when its soon to finish.
- Users can pay online using a secure payment system.
- Sellers can have flash images or vedios to add to their adverstaisement.
- Guests can view and search for items but not be able to bid unless they signup for an account.
- Admin members can change sale status, Basic, seller and Advanced members can delete or edit their account/ profile.

Non-Functional Requirements:

- Ease of navigation around the site and use of simple colors and fronts on the system.
- All web pages that are generated throughout the website should be downloadable in no more than 10 seconds over 40KBps modern connection.
- Interface consistency through out the system.
- The response to a query shouldn't take a very long time

and not more than 10 seconds to load on screen.

- When a user sells or bids on the system they should be getting a conformation of what the member did, i.e. adding a new item on sale at the end of the adding to the system they should get a confirmation message to say its been added.



Usecase view:

* Use case diagrams model the functionality of the system.
* It contains of actors, usecases & relationships.

Usecases are

- Browse website
- Register details
- Provide/get login credentials.
- check availability
- fill the registration form to submit.
- Payment
- Issue Ticket.

Actors are

- Customer
- user
- agent
- Database administration.
- Bank administrator.

Relationships are

- Association
- Generalization.

Class Diagram View

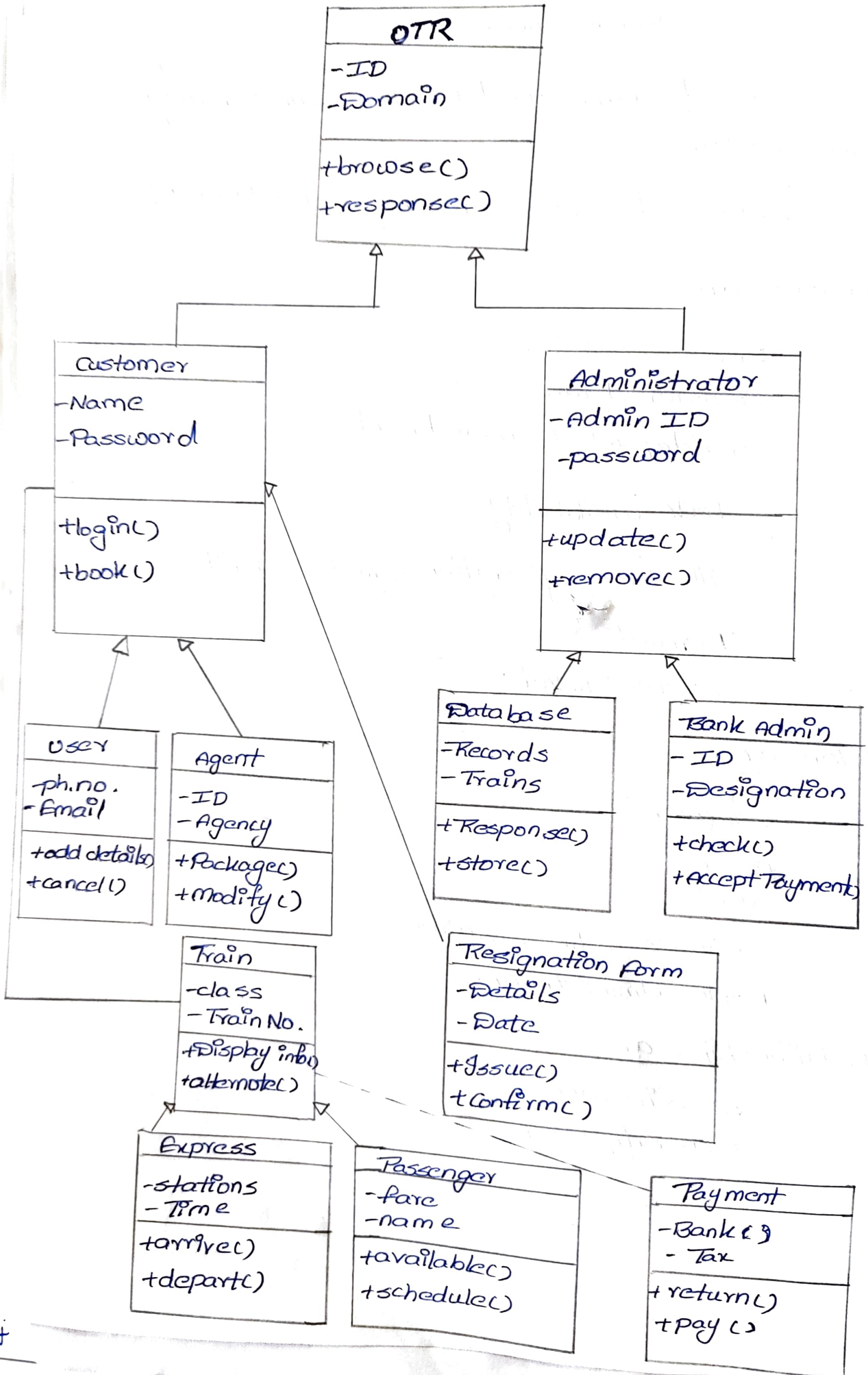
* class diagram describes the structure of a system by showing the system's classes, attributes, operations and relationships among them.

S.No	ClassName	Attributes	Operations.
1	OTR system	ID, domain	browse(), responses()
2	Customer	Name, password	login(), book()
3	Administrator	Admin ID, password	update(), remove()
4	User	ph.no, email	add details(), cancel()
5	Agent	ID, Agency	Package(), Modify()
6	Database	records, trains	response(), store()
7	Bank Admin	ID, Designation	check(), Accept payment()
8	Registration form	details, date	issue(), conform()
9	Train	class, Train no.	display info(), alternate()
10	Express	station, Time	arrive(), depart()
11	Passenger	fare, name	available(), schedule()
12	Payment	Bank, Tax	refund(), Pay()

Relationships are

Association, Generalization, Dependency.

Class Diagram



Sequence Diagram View

* It depicts the interaction between objects in a sequence order.

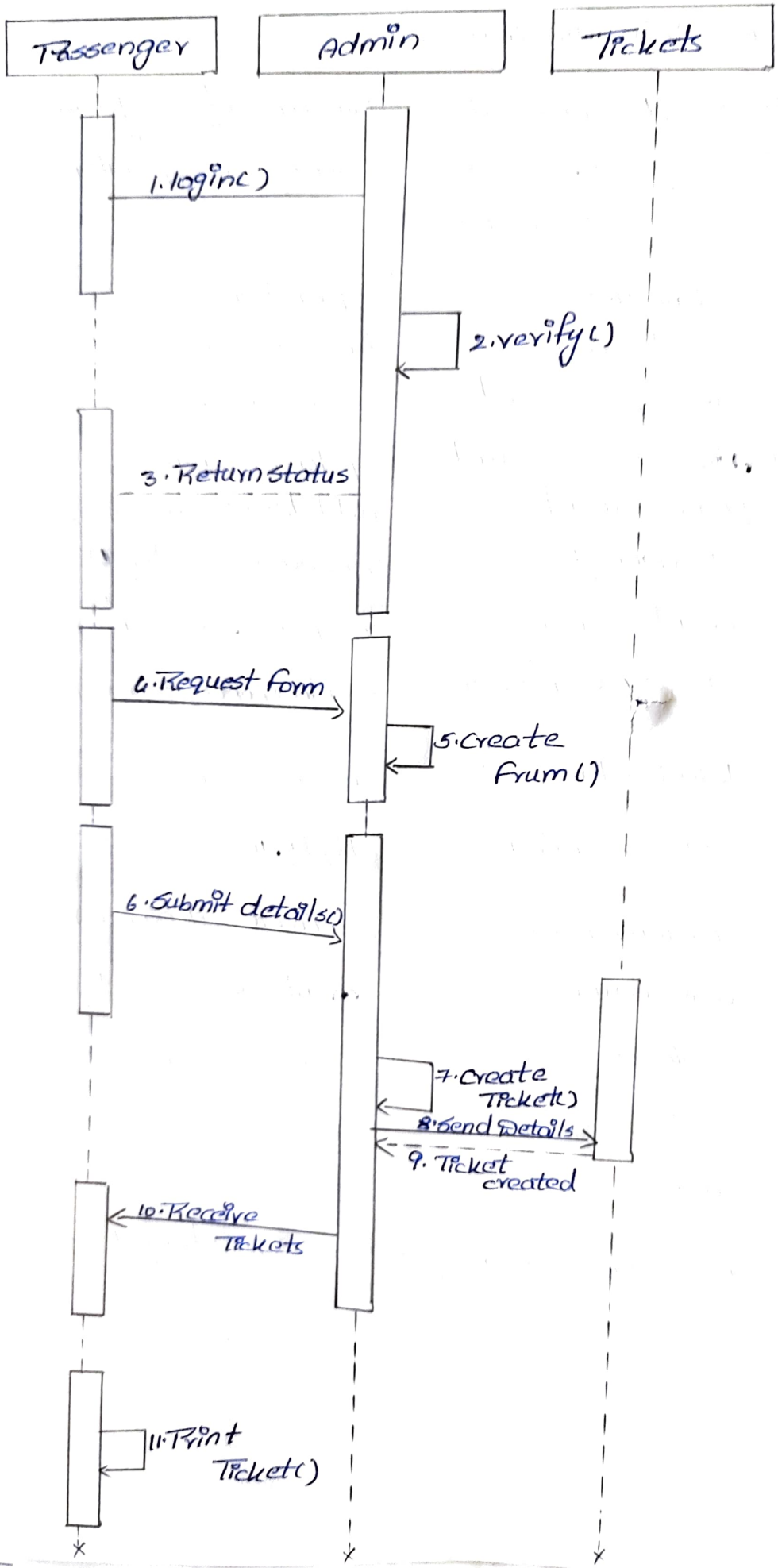
objects are

Passenger
Admin
Tickets.

Messages are

1) login()
2) verify()
3) Return Status()
4) Request form()
5) Create Form()
6) Submitdetails()
7) Create Ticket()
8) Send Details()
9) Ticket created()
10) Receive Ticket()
11) Print Ticket()

Sequence Diagram



Collaboration View:

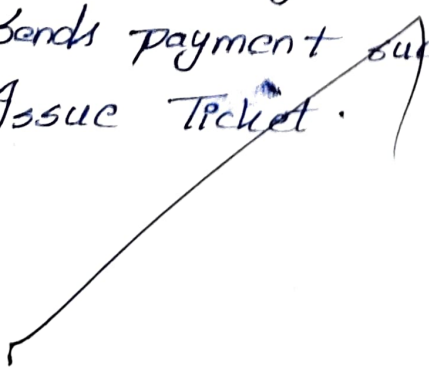
* It shows how various objects interact with each other.

Objects are

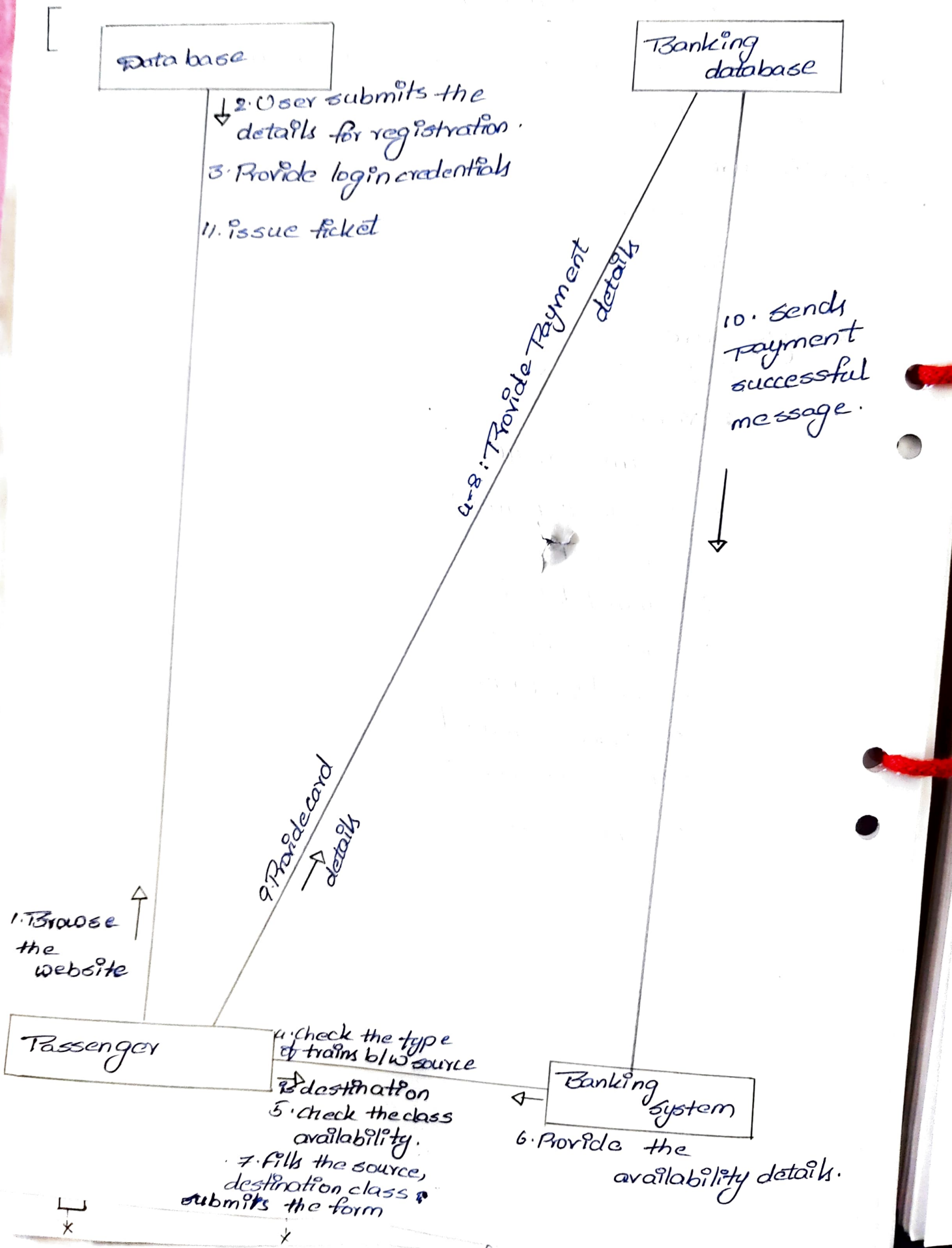
- Passenger
- Database
- Banking system
- Banking database.

Messages are

1) Browse the website.
2) User submits his details for registration.
3) Provide login credentials.
4) Check the types of trains between source and the destination.
5) Check the class availability.
6) Provide the availability Table.
7) Fills the source, destination, class & submits the form.
8) Provide card credentials / details.
9) Provide payment details.
10) Sends payment successful message.
11) Issue Ticket.



Collaboration Diagram



Statechart View

* It defines different states of objects during its lifetime and are changed by events.

States are

idle - Database is ready to back the tickets.

browse - Customer browses the websites & select the required website from ticket booking

Registration - Customer provides his details for registration into website.

Verification - Database verified the details.

Processing - Database processess the customer details. and provide login credentials.

check - Customer checks the availability of train, class.

Selecting - Customer submits regestration selects the type & class of tickets.

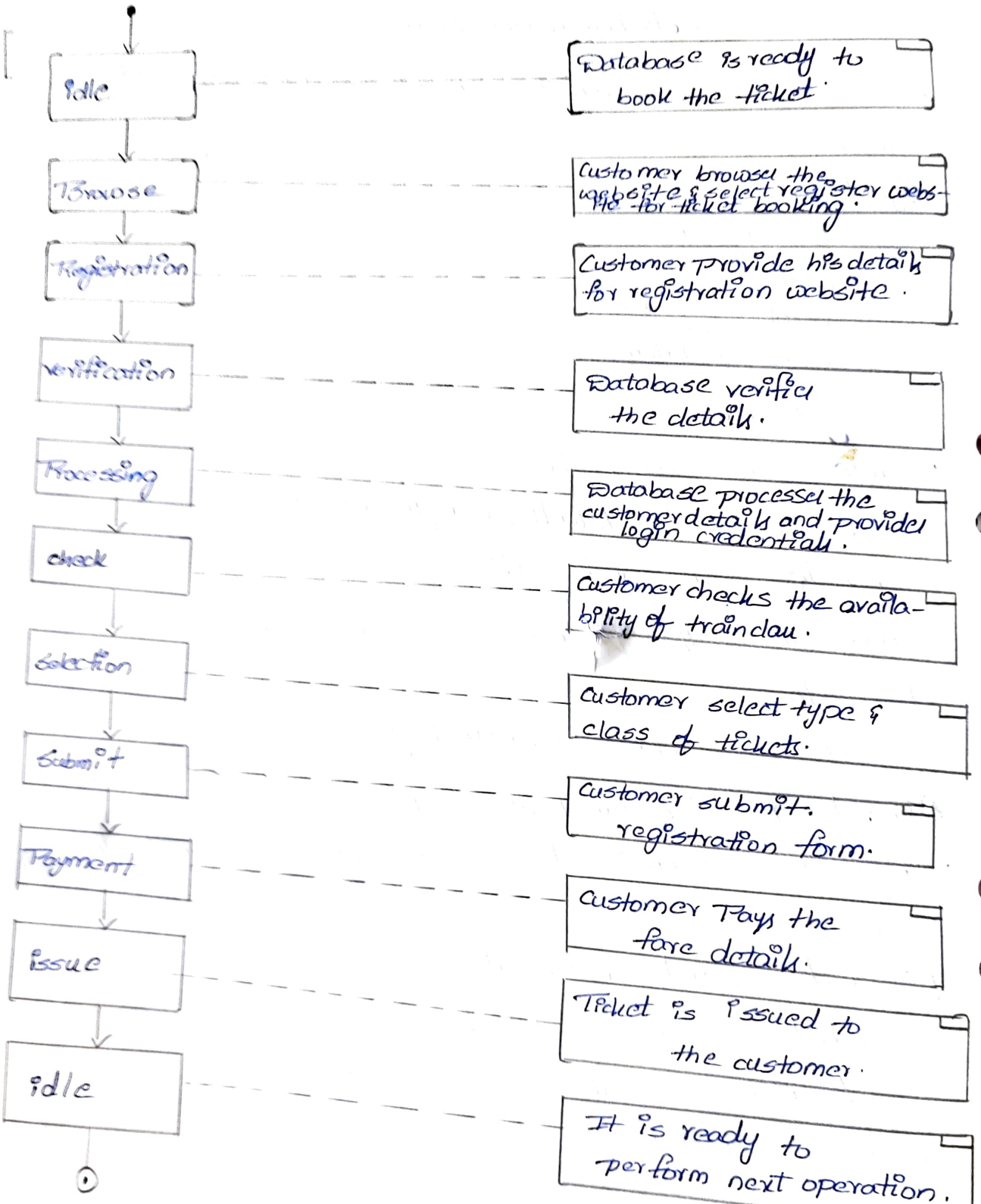
Submit - Customer submits registration from.

Payment - Customer pays the fare details.

Issue - Ticket is issued to the customer.

Idle - It is ready to perform another operation.

State Chart Diagram



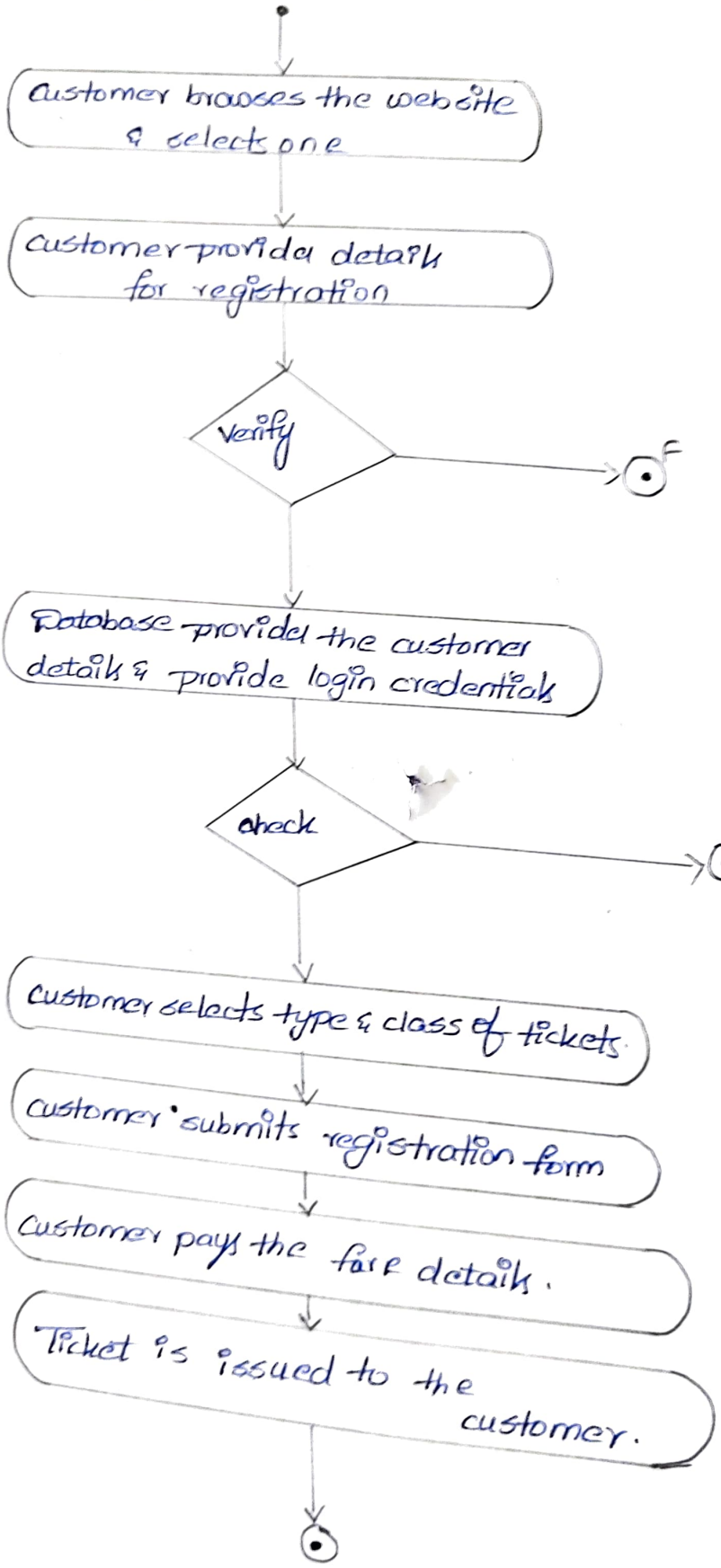
Activity View

It represents flow from one activity to another activity.

Activities are

1) Customer browses the website and select one.
2) Customer provides details for registration.
3) Database processes the customer details and provide login credentials.
4) Customer selects type and class of tickets.
5) Customer submits registration form.
6) Customer pays the fare details.
7) Ticket is issued to the customer.

Activity Diagram



Component View

* There are used to visualize the organization and relationships among components.

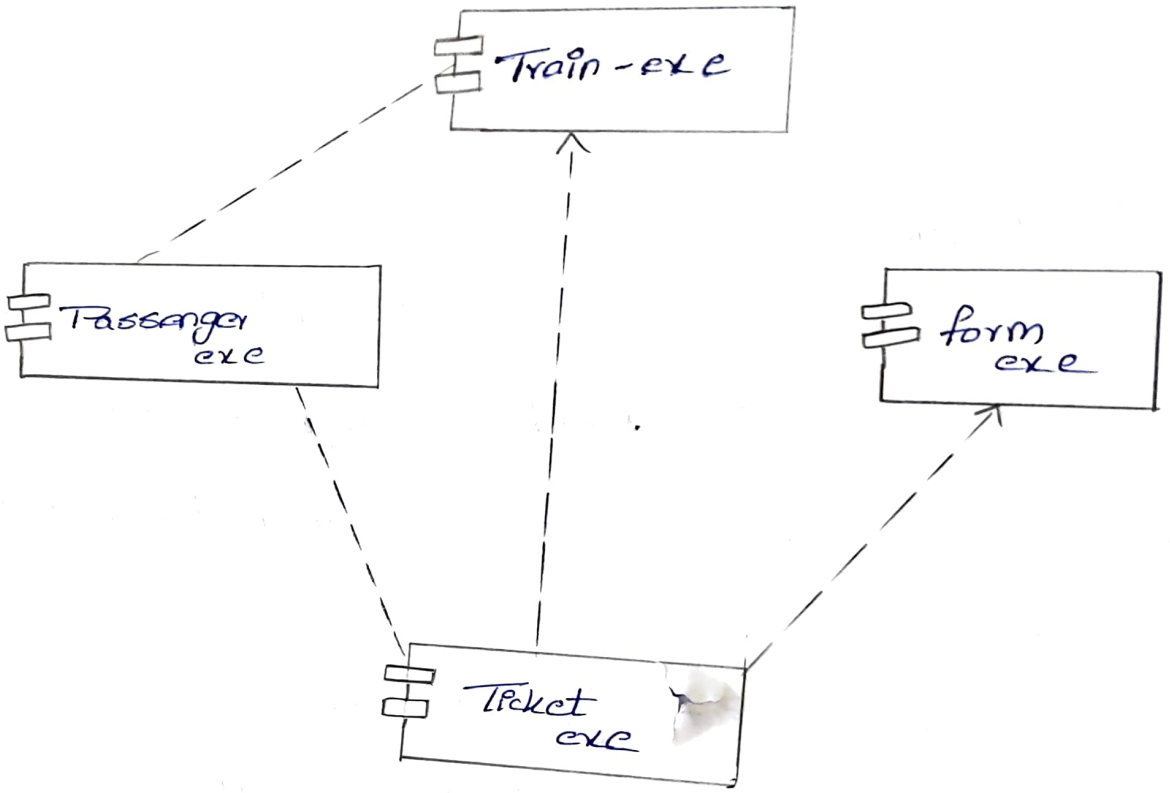
Components are

- Train
- Ticket
- Passenger
- Form

Relationships are

- Association
- Dependency

Collaboration Diagram



Deployment view

* It represents the deployment view of the system.
* It consists of nodes which are used to display the application.

Nodes are

Railway Reservation Database

Ticket

OTR system

Passenger

Travel agent

Payment

Train

Relationships are

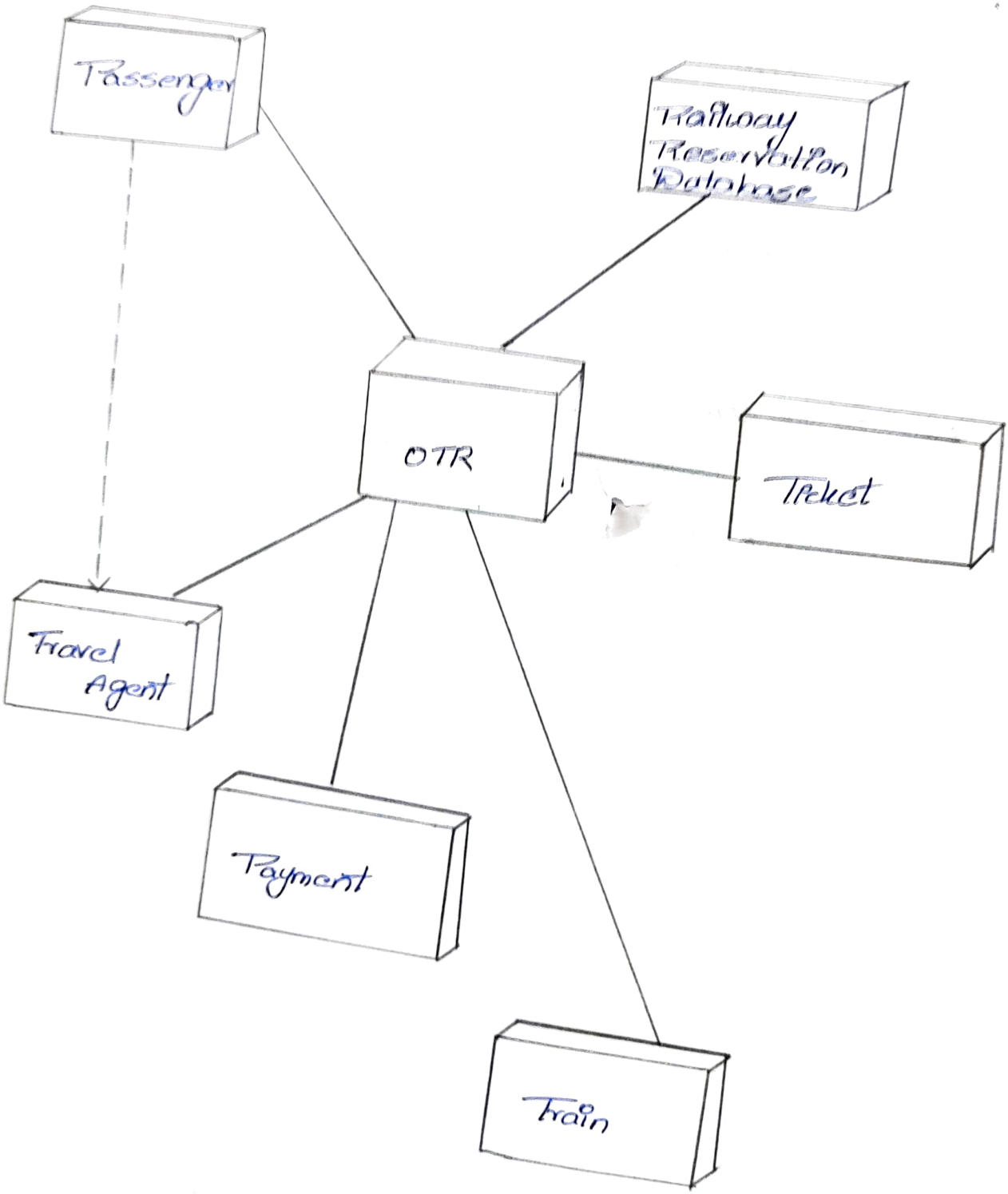
Association

Dependency

Result:

Thus all the news of the online Ticket Reservation system have been modelled and this is mapped to following Pro's and Con's.

Deployment Diagram



Case Study-2:

Aim To model all the value of point of state system (POS)

Problem Statement

A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and barcode scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant, that is even if remote services and temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDA's touch-screens.

Use Case View :

* Use case diagrams model the functionalities of the system
* It contains of actors, usecases & relationships

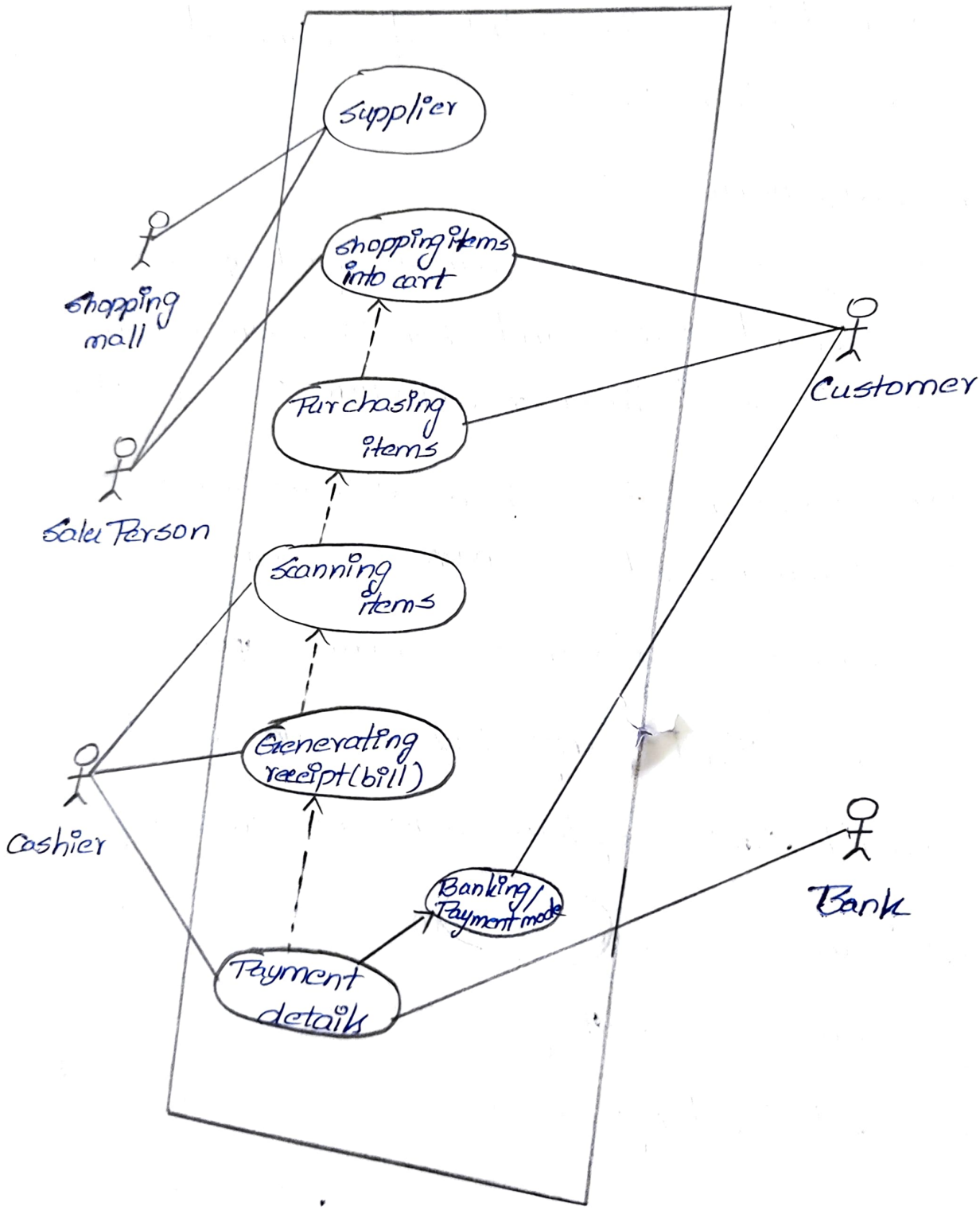
Use cases are

- Supplier
- shopping items into cart
- Purchasing items
- Scanning items
- Generating receipt
- Payment details
- Banking/ Payment mode.

Actors are

- Shopping mall
- Sale person
- Cashier customer
- Bank.

UseCase Diagram



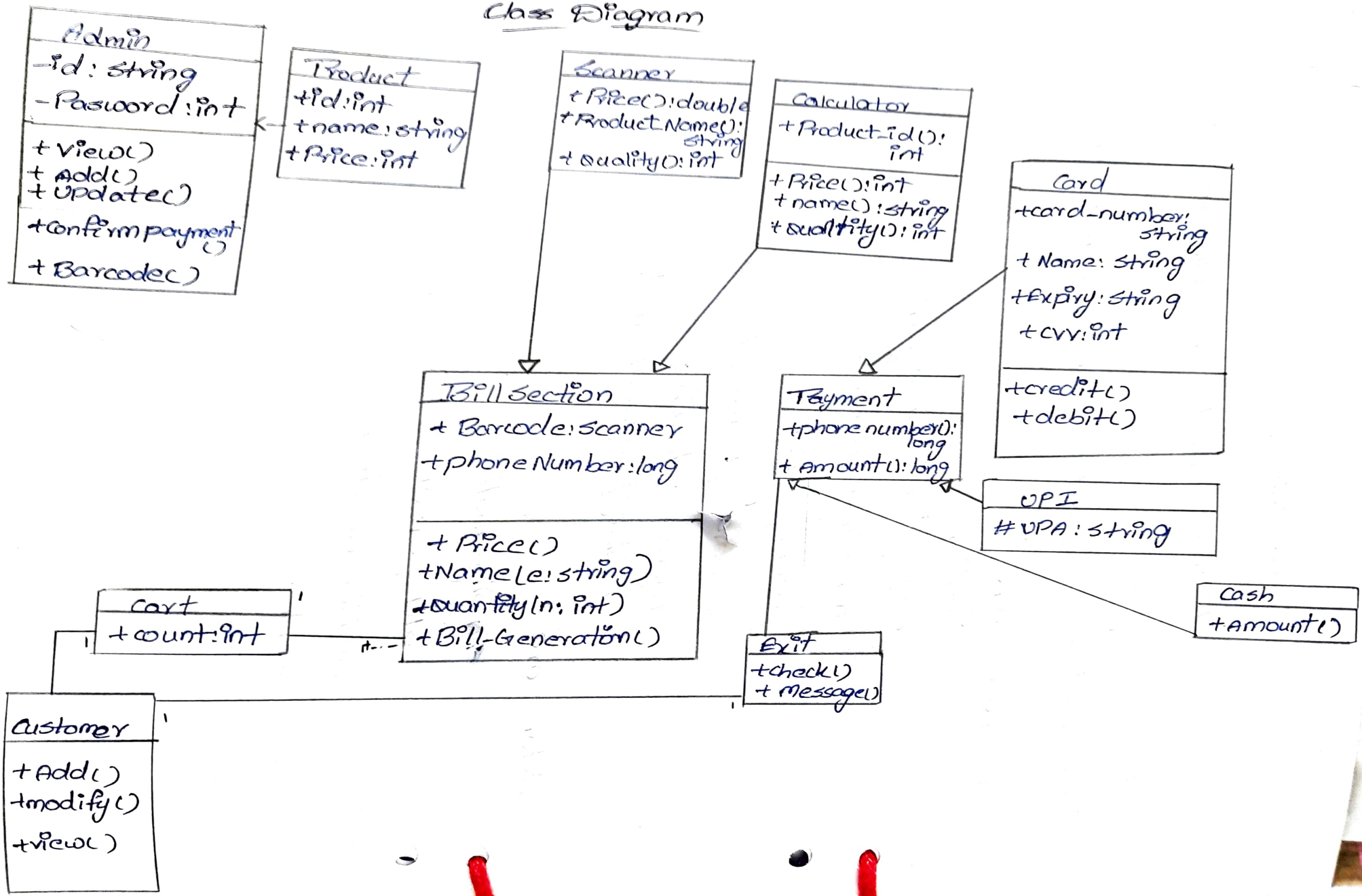
Class Diagram View :

* class diagram describe the structure of a system by showing the system's clases, attributes, operations & relationships among them.

S.No	Class Name	Attributes	Operations.
1	Supplier class	branch, Address	placingorder, getting order
2	Bill class	date: int, amount: float	display date(), display amount()
3	Purchase class		cart data(), billgeneration(), item scanner().
4	Customer class	name, address, ph.no.	ask Item(), search Item(), feed back()
5	shopping mall class		
6	Receipt class	no. of items	name(), address() add amt(), add discount()
7	Scanner class	no. of items	barcode scanner()
8	Shopping cart class	item list: array	-
9.	Sales person class	name, id	
10.	payment class	amount, cal. amount, get balance, add text	
11.	Item class	name, item, weight, manufactu re	
12	cash class	dominations	dominations counter
13	Credit card class	card-name	do-expiry
14	Debit card class	card-number	do-expiry.
15	UPI class	Id_int	Scanner() receipt()

Relationships are Association, Generalization, Dependency.

Class Diagram



Sequence Diagram View

* It depicts the interaction between objects in a sequence order.

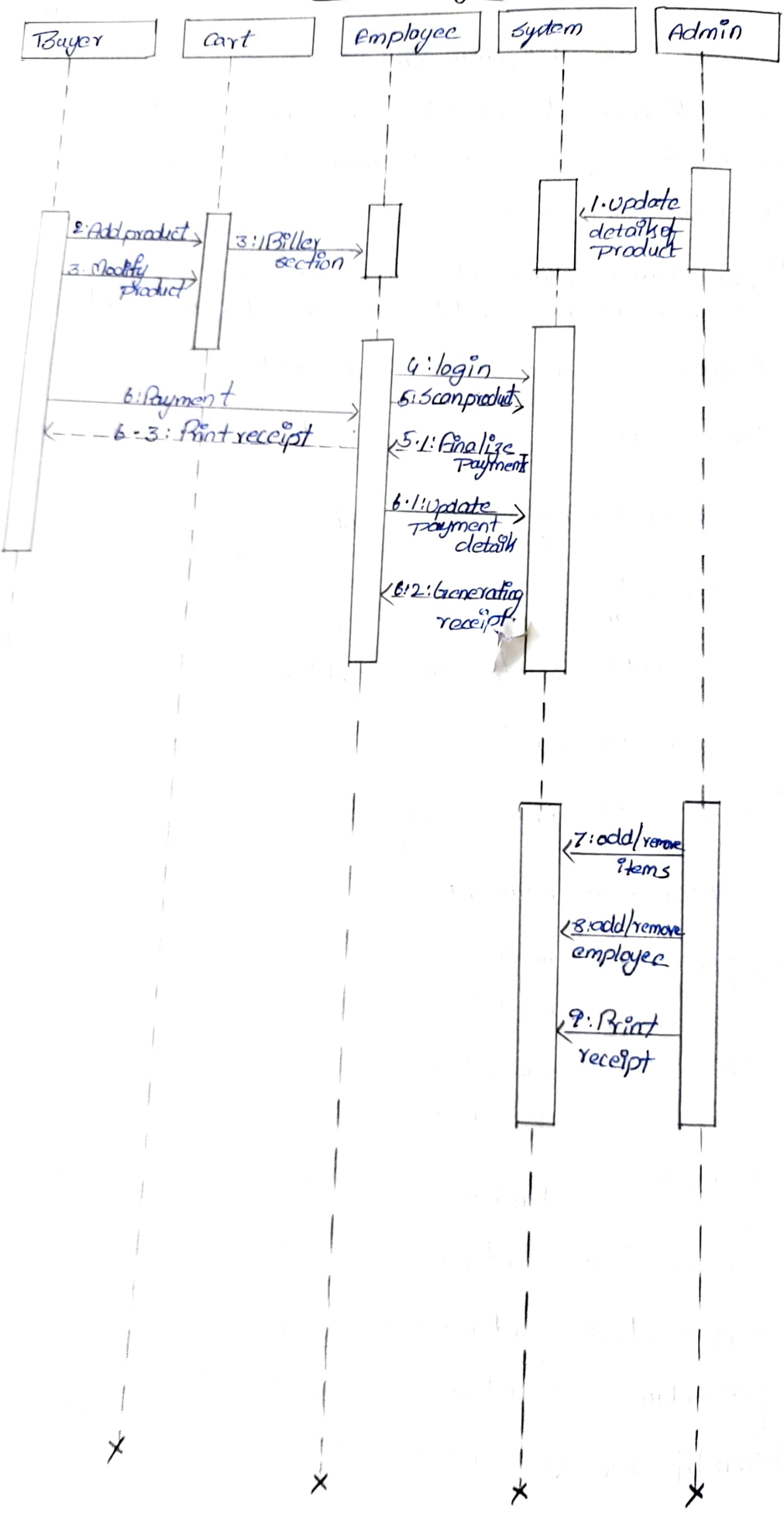
Objects are

- Buyer
- cart
- Employee
- System
- Admin

Messages are

1) Update details of product
2) Add products.
3) Modify products.
4) 3.1) Biller section.
4) Login
5) Scan products
5.1) Finalize payment
6) Payment
6.1) Update permanent details.
6.2) Generating Receipt
6.3) Print Receipt
7) add / remove items
8) add / remove employee
9) Print report

Sequence Diagram



Collaboration View

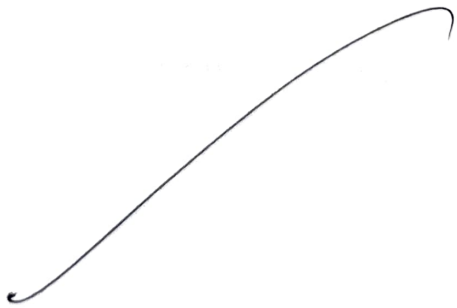
It shows how various objects interact with each other

Objects are

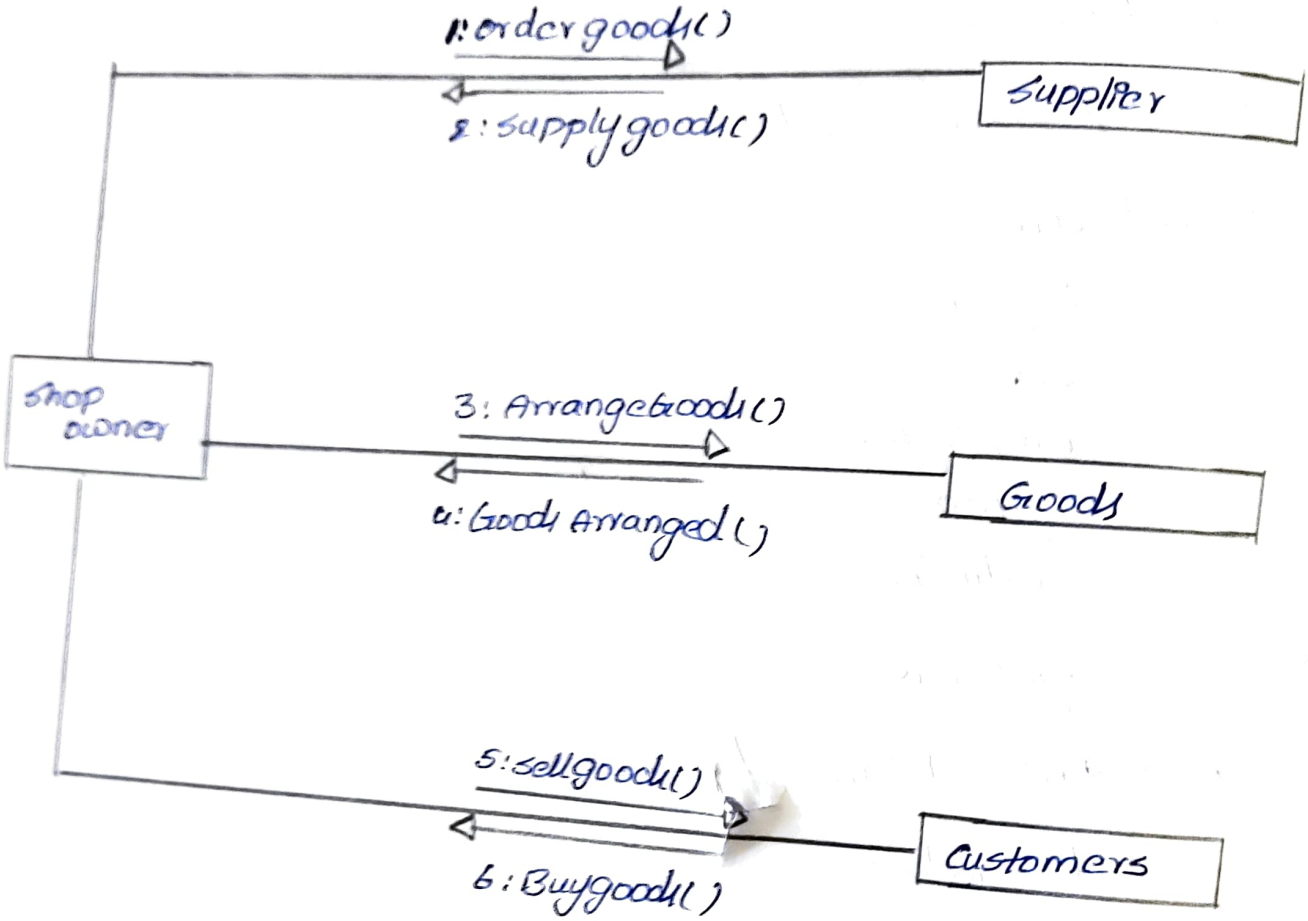
- Shopowner
- Supplier
- Goods
- Customer

Messages are

1) order goods()
2) supply goods()
3) Arrange goods()
4) Goods Arranged()
5) Sell goods()
6) Buy goods()



Collaboration Diagram



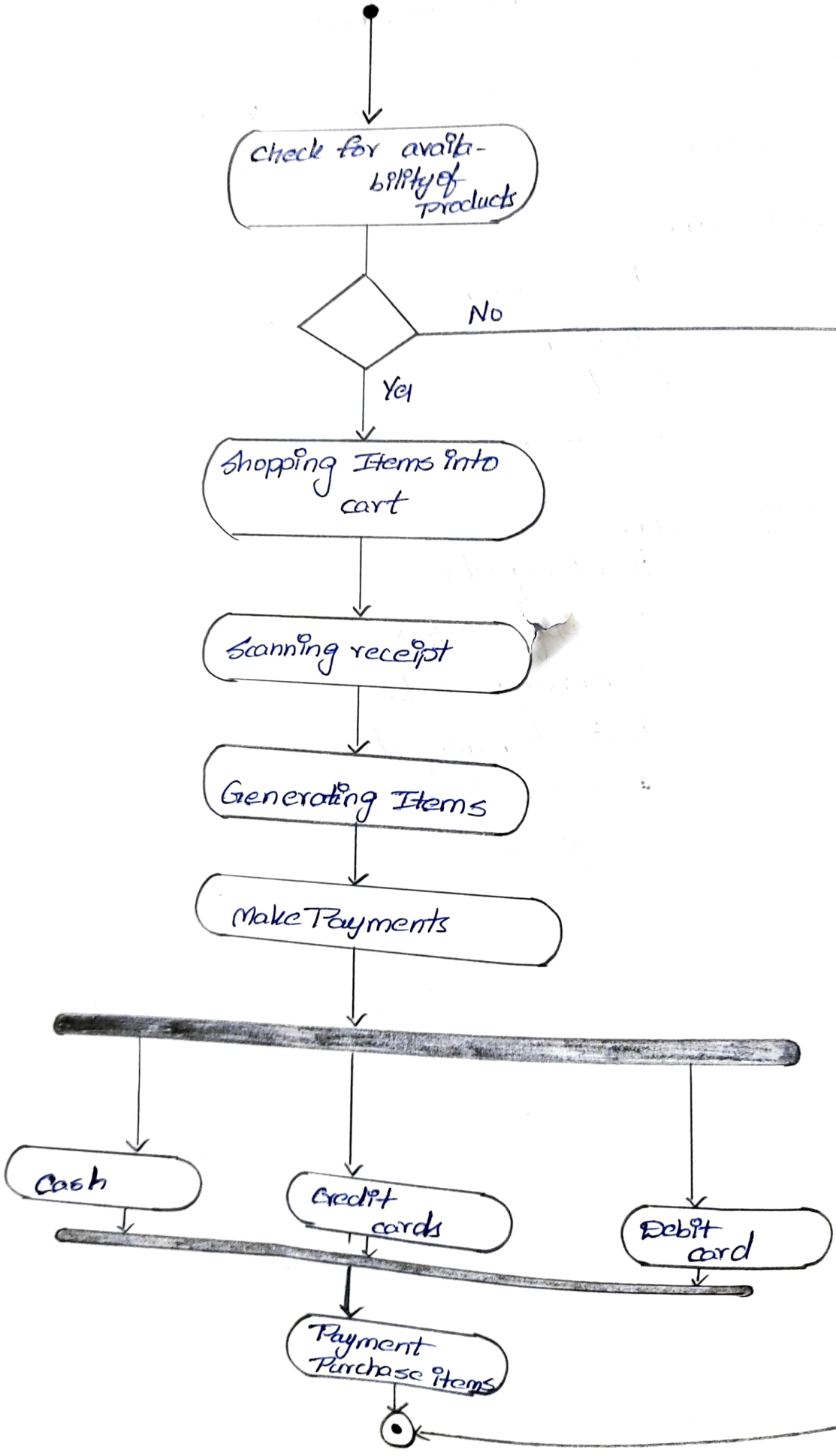
Activity View

It represents flow from one activity to another activity.

Activities are

1) Customer checking for availability of products.
2) Shopping items are added into cart.
3) Scanning the items for receipt.
4) Generating the receipt for the items
5) Customer makes payments.
6) The options under payment is through cash, credit card, debit card.
7) The items are being purchased.

Activity Diagram



Component View

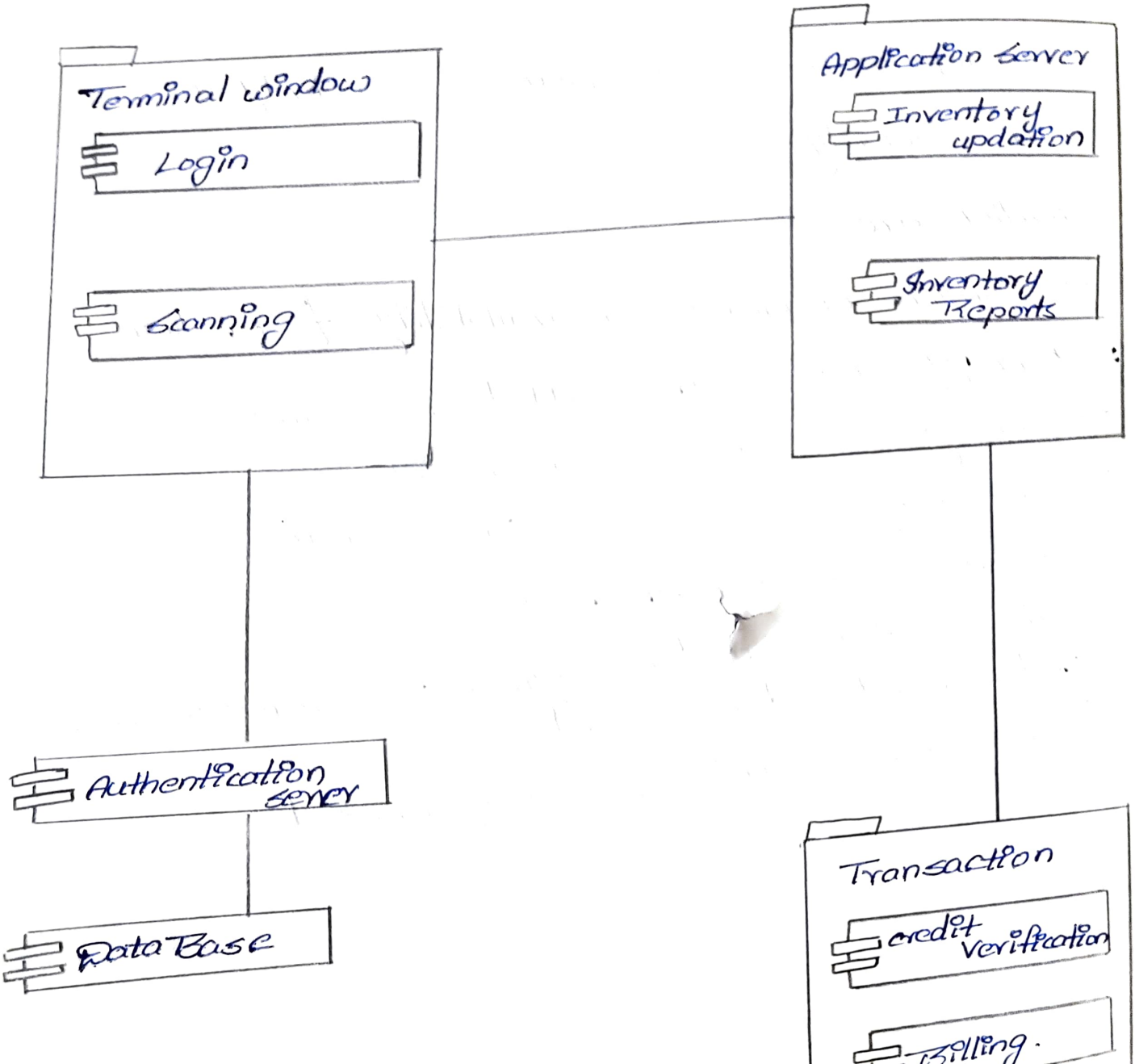
Component diagram also known as UML component diagram, describes the organization and wiring of the physical components in a system.

Component diagrams can be used to -

- model the
  - components of a system
  - executable of an application
  - system's source code.
  - database schema

Components in the component diagram are Terminal window, Application server, database and transaction.

Component Diagram



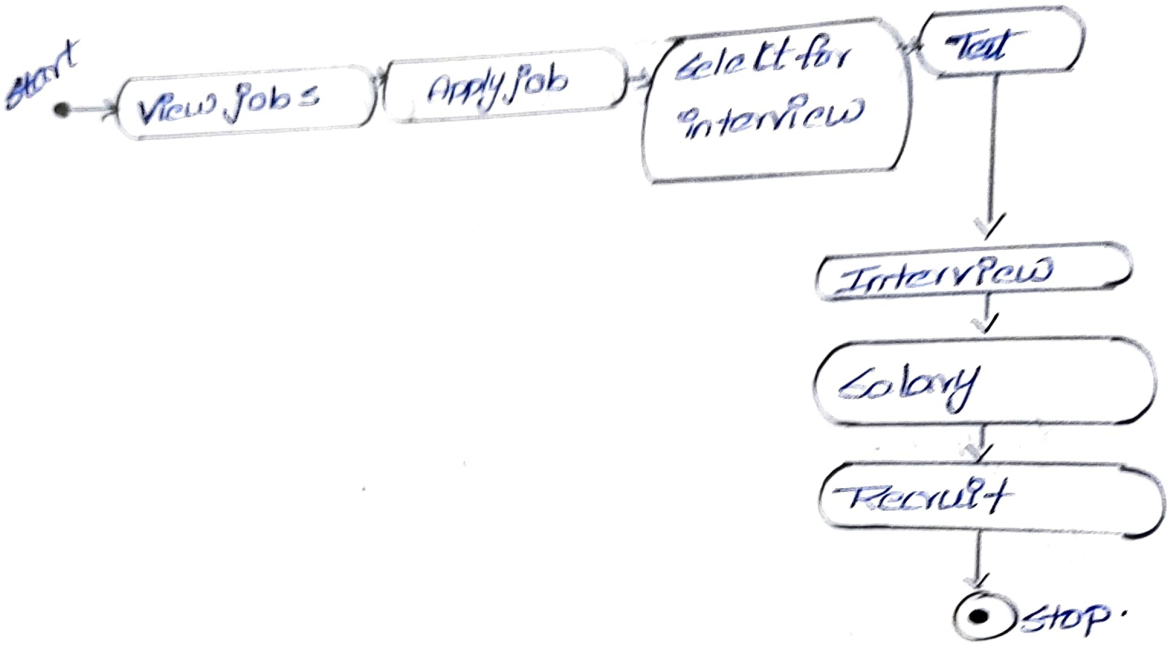
State Chart View

* It defined different states of objects during its lifetime and are changed by events.

States are

1) View jobs
2) Apply job
3) Select for interview
4) Test
5) Interview
6) Salary.
7) Recruit.

Statechart Diagram



Deployment View

* It represents the deployment view of the system.
* It consists of nodes which are used to display the application.

Nodes are

Printer

Terminal

Barcode Scanner

Application Server

Database.

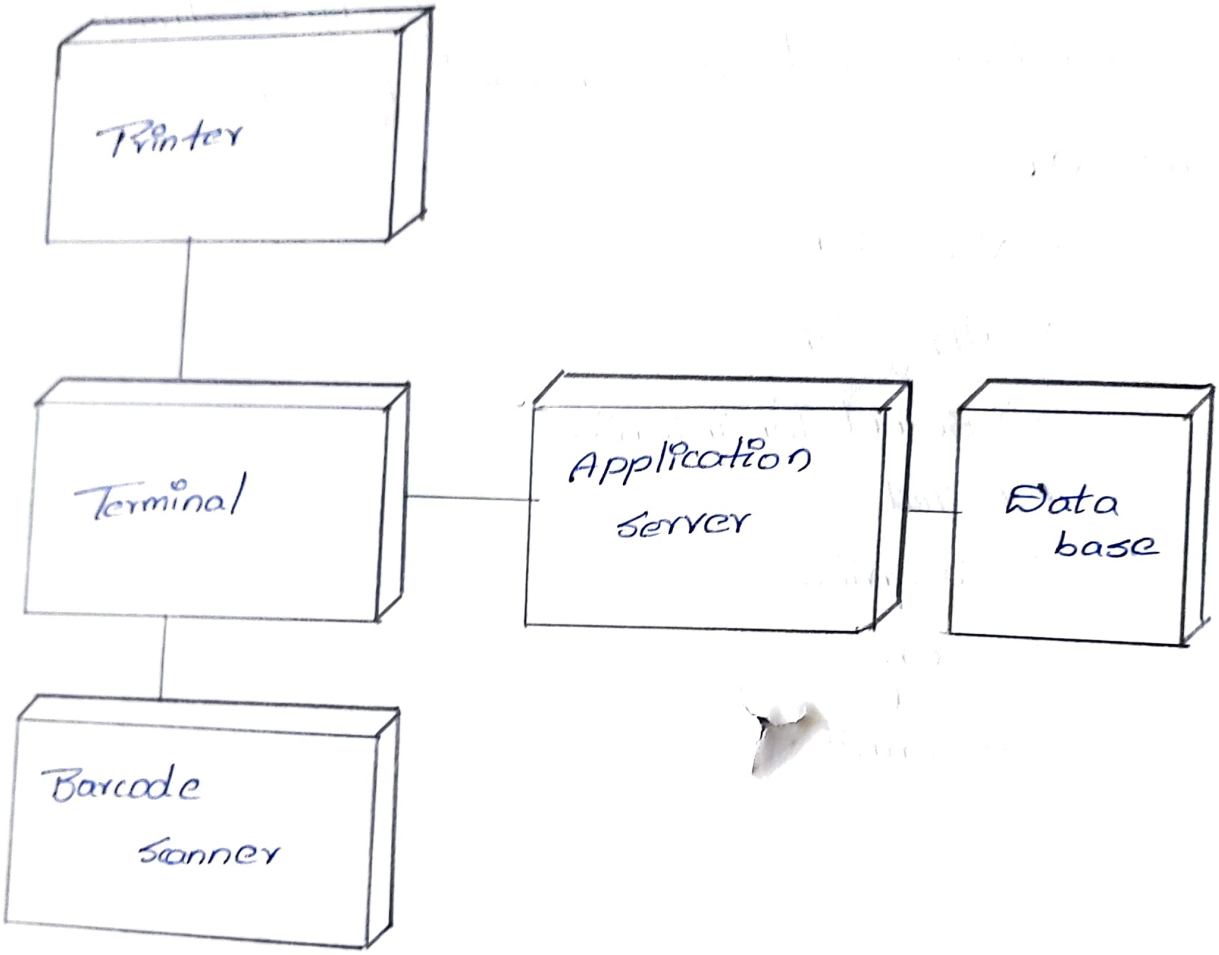
Relationships are

dependency.

Result:

Thus all value of point of state system(POS) are modelled.

Deployment Diagram



Case Study -3.

Aim To mode the 'Recruitment procedure' for software industry.

Problem statement

In the software industry the recruitment Procedure is the basic thing that goes in the hand with the requirement as specified by the technical management team HR gives first an advertisement in leading NewsPapers, Journals, weeklies and websites. The job seekers can apply for it through by Post or e-mail to the company. The Techical skill and the experience of the candidates are reviewed and the short listed candidates are called for the interview. There may be different rounds for interview like written Test, techical interview, and HR interview. After the successful completion of all rounds of interview the selected candidates names are displayed.

Meanwhile HR gives all the details about the salary, working hours, terms and conditions and the retirement benifit to the candidate.

UseCase View :

* Usecase diagrams model the functionality of the system.
* It consists of actors, usecases & relationships.

Usecases are

- Login
- Register
- Send interview details
- Attend test
- Select talented applicant
- Send appointment letter.

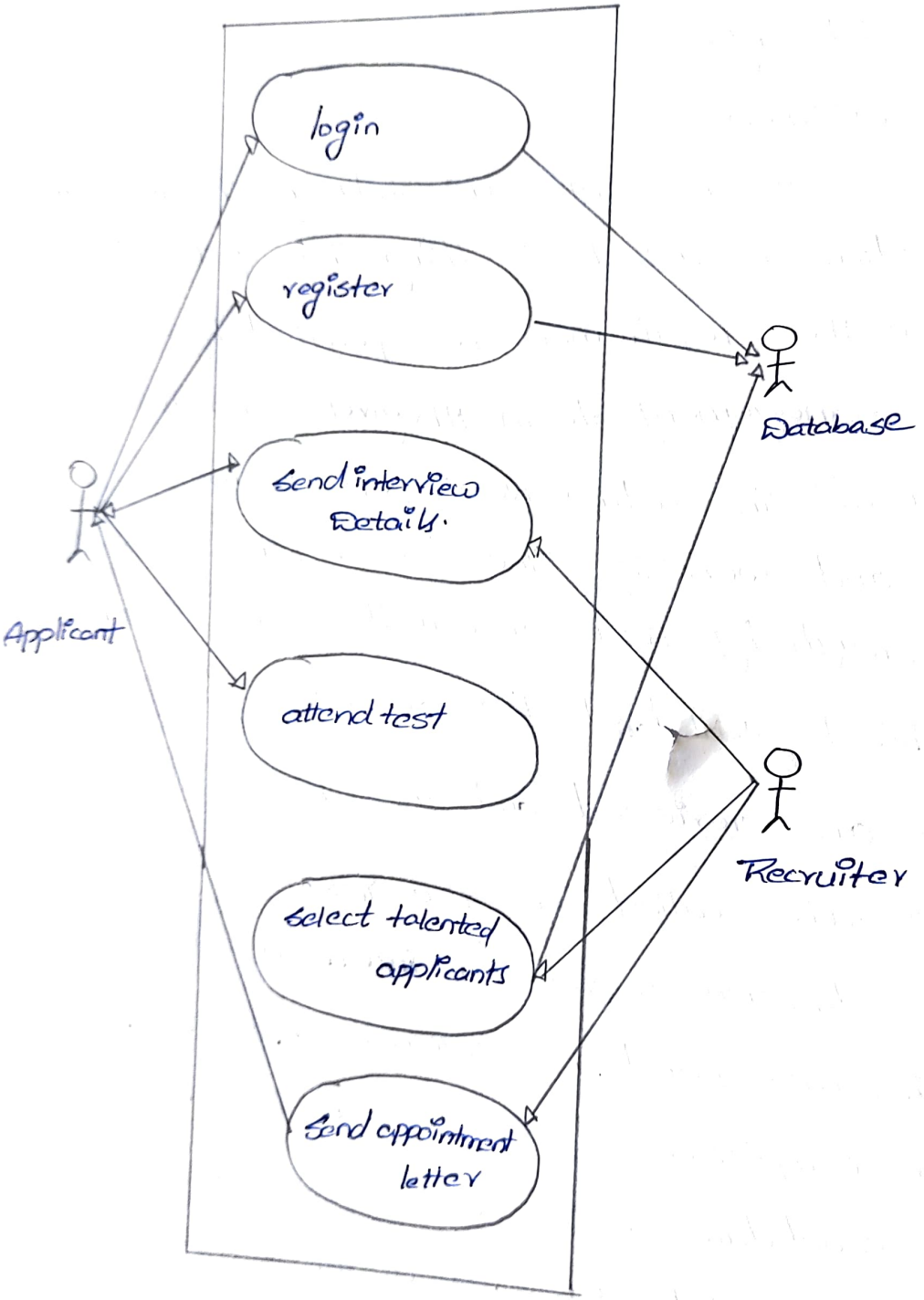
Actors are

- Applicant
- Database administrator
- Recruiter.

Relationships are

- Association
- Generalization

Use case Diagram

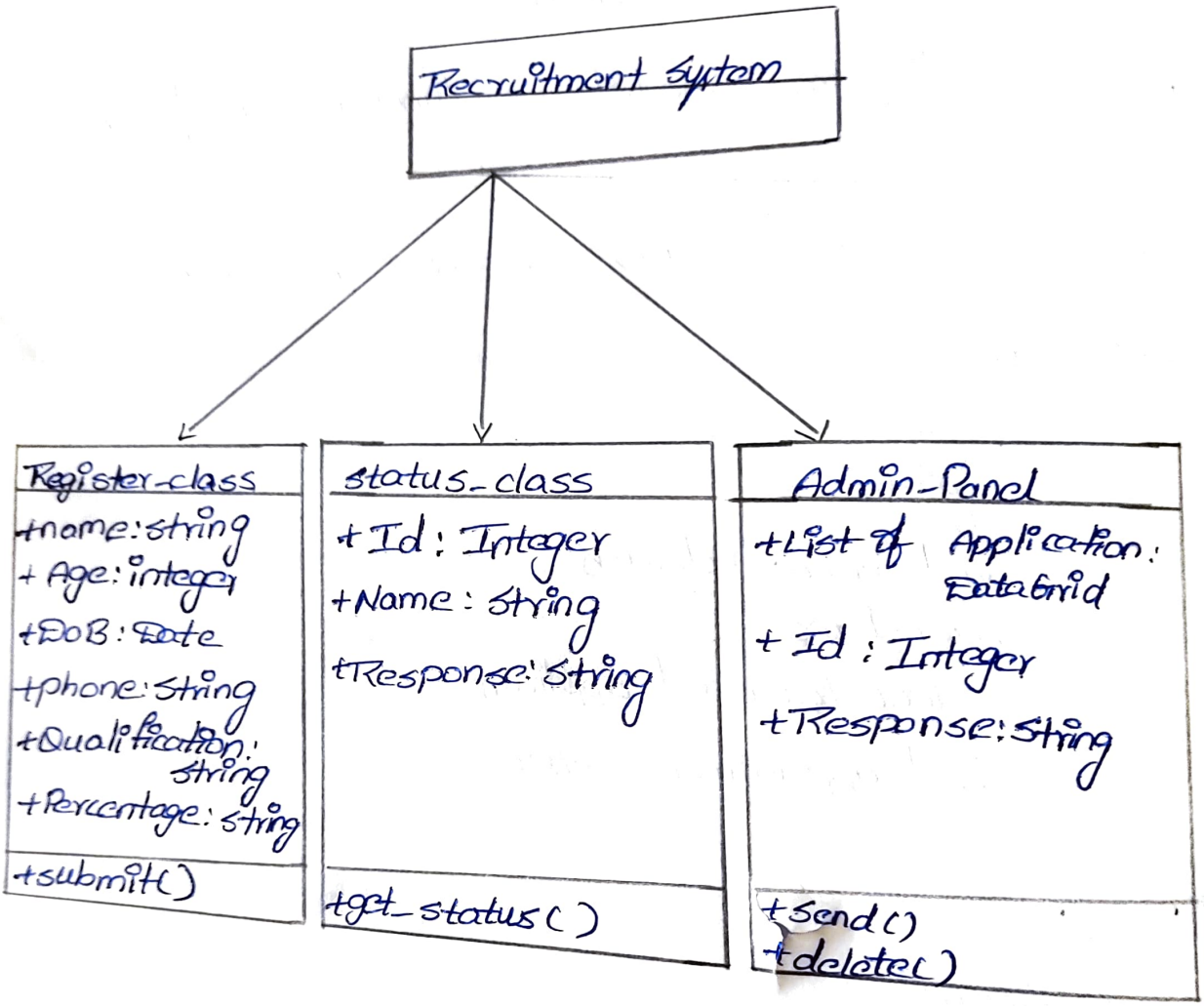


Class Diagram View

*Class diagram describes the structure of a system by showing the system's classes, attributes, operations and relationships among them.

S.NO	Class Name	Attributes	operations
1	Recruitment system	—	—
2	Register_class	Name, age, DOB, phone, Qualification, Percentage	Submit()
3	Status_class	Id, Name, Response	get_submit()
4	Admin_panel	List of Applications, Id, Response	send(), delete()

Class Diagram



Sequence View

* It depicts the interaction between objects in a schem

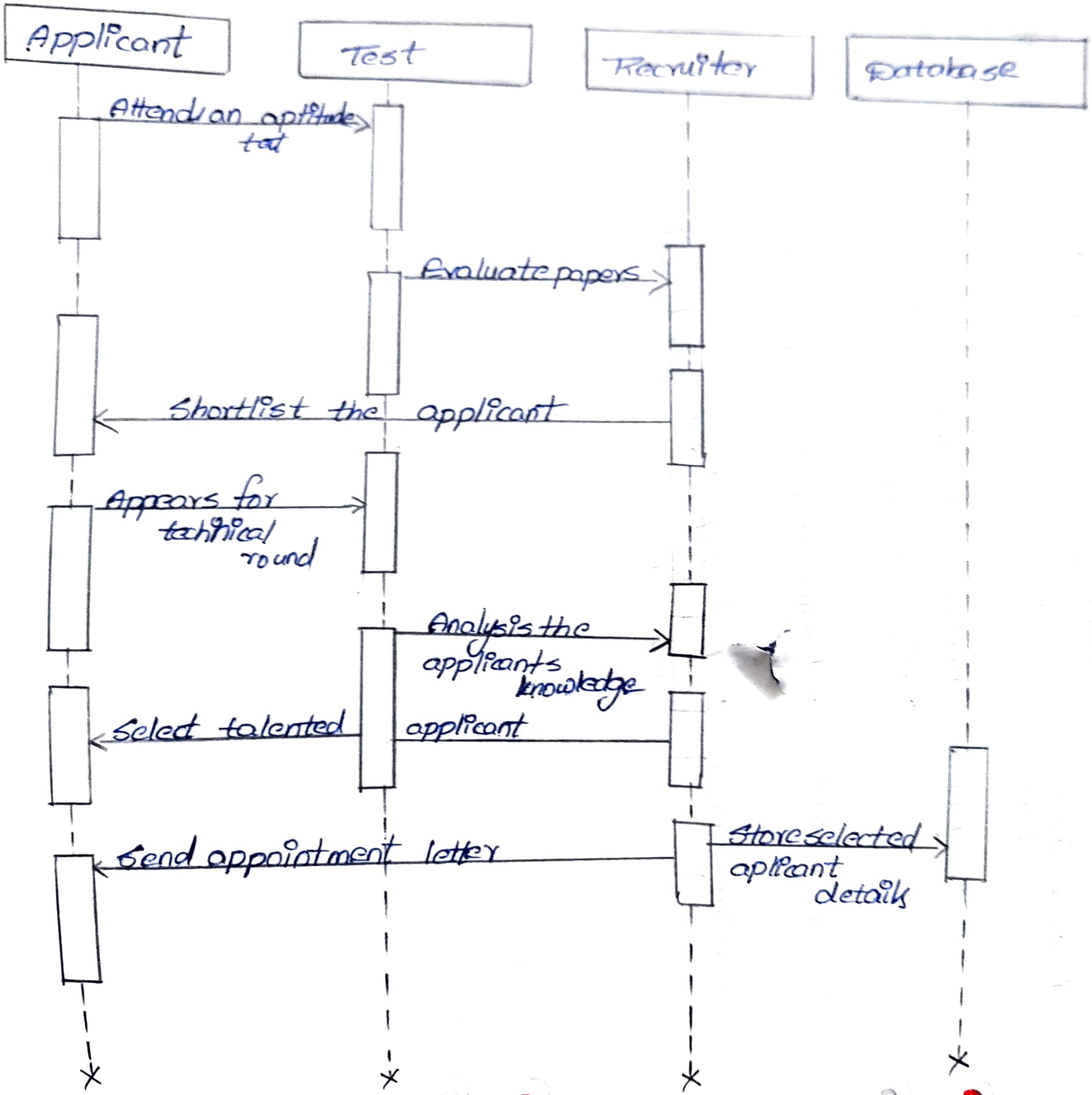
order

Objects are
- Applicant
- Test
- Recruiter
- Database

Messages are

1) Attends aptitude test
2) Evaluate papers
3) Shortlist the applicant
4) Appears for technical round
5) Analysis of applicants knowledge
6) Select talented applicant
7) Store selected applicant details.
8) Send appointment letter.

Sequence Diagram



State chart View

* It defines different states of objects during the its lifetime and are changed by event.

States are

Login - The applicant first login into the database.

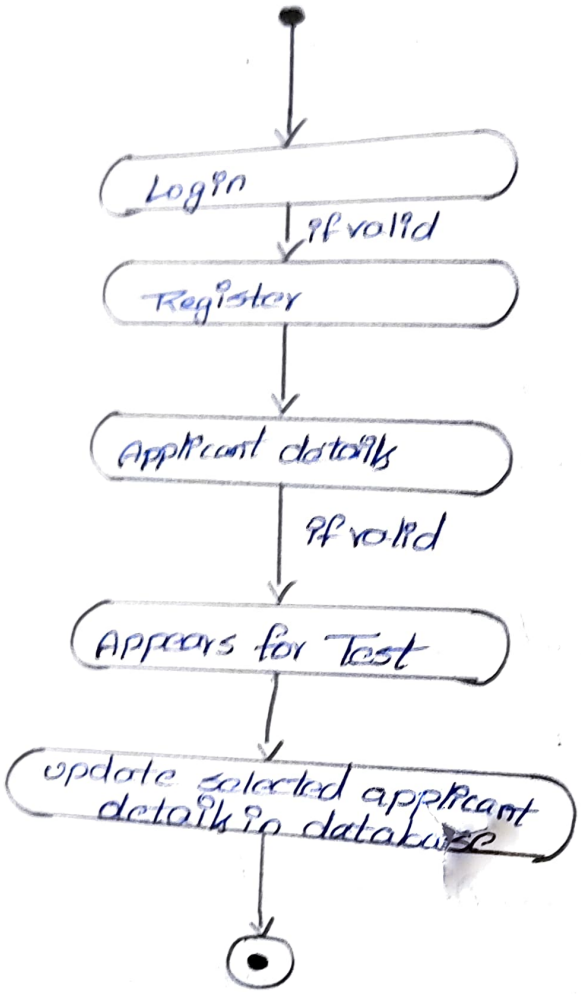
Register - If the login is valid then register into the database.

Applicant details - Submit the details while registering.

Appears for test - If applicant details are valid then the applicant can appear for test.

→ After the submission of applicant the update for the selected applicant details in the database.

Statechart Diagram



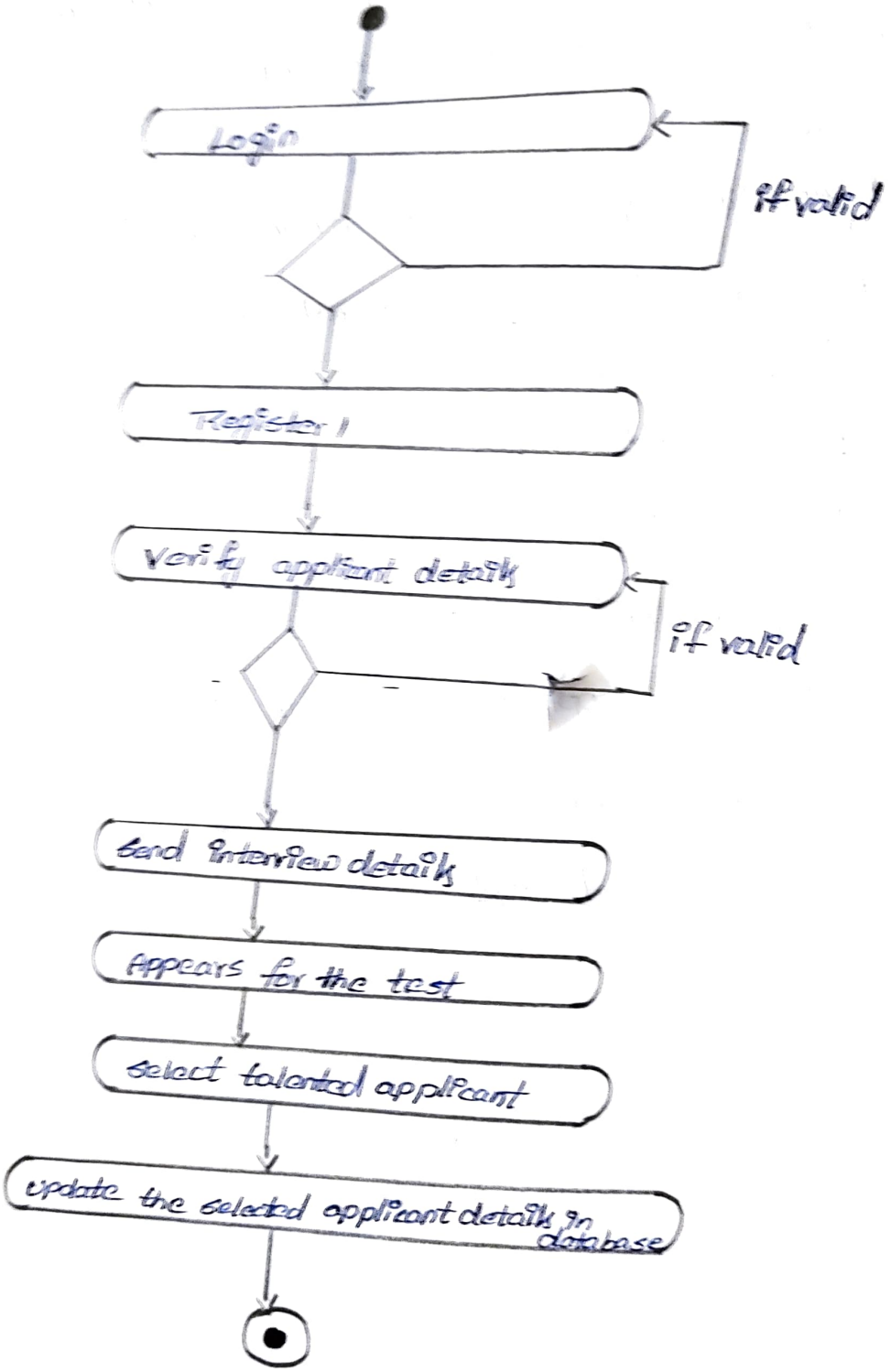
Activity View

It represents the flow from one activity to another activity

Activities are

1) Applicant logins into the database
2) If it is valid then user registers into the database.
3) Verify the user details.
4) If they are valid send interview details to the applicant.
5) The applicant appears for the test.
6) Based on test the talented applicant is selected.
7) Finally, the selected applicant details are updated into the data base.

Activity Diagram



Component View:

These are used to visualize the organizations and relationships among components.

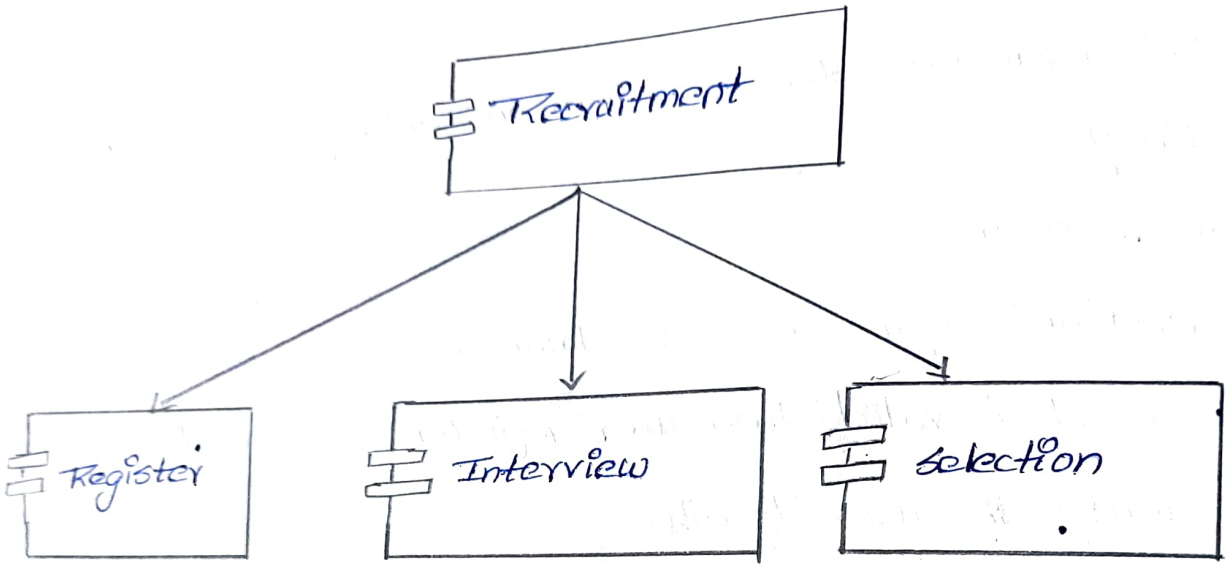
Components are

- Recruitment
- Register
- Interview
- Selection

Relationships are

- Association

Component Diagram



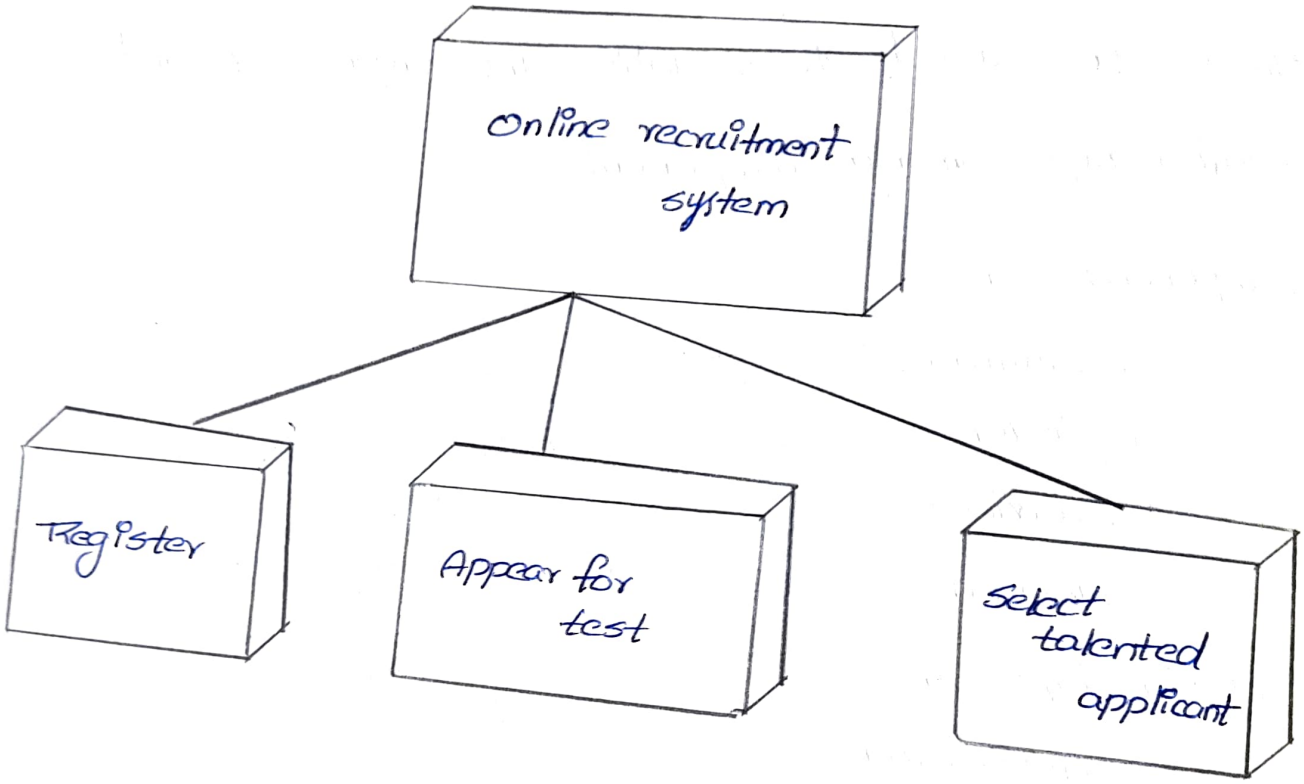
Deployment View:

* It represents the deployment view of the system.
* It consists of nodes which are used to display the application.

Nodes are

Online recruitment system
Register
Appears for test
Select talented applications/applicants.

Deployment Diagram



Collaboration View

It shows how various objects interact with each other.

Objects are

- Job seeker
- Job Recruiting Agency
- Employer

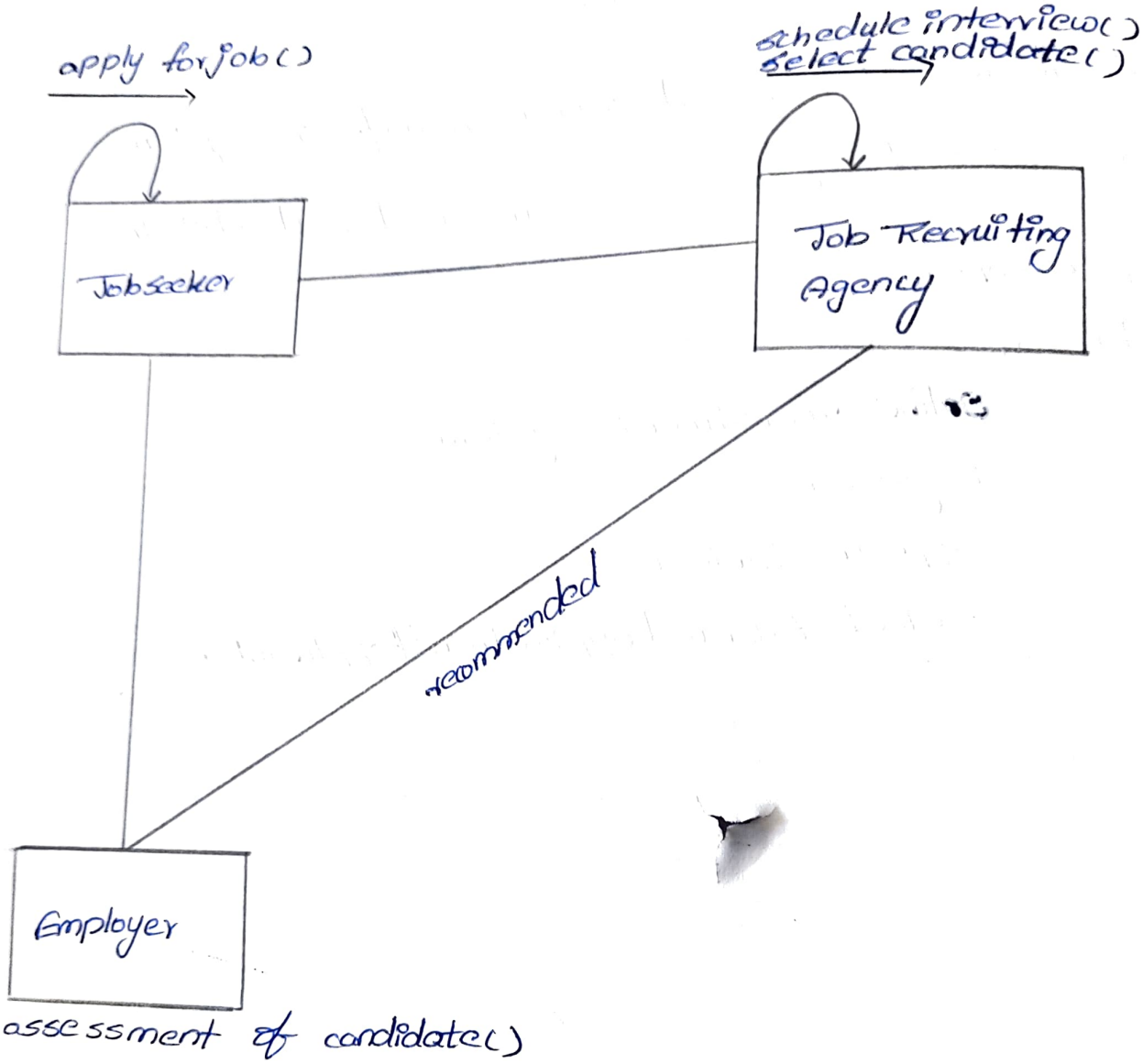
Messages are

- apply for job()
- schedule interview()
- select candidate()
- assessment of candidate()

Result:

The all values of recruitment procedure are duled.

Collaboration Diagram



Case Study - 4

Aim

To model all the values of online Auction sales.

Problem Statement

The online auction system is a design about a website where sellers collect and prepare a list of items they want to sell and place it on the website for visualizing. To accomplish this purpose the user has to access the site. Incase it's a new user he has to register, Purchaser's login and select items they want to buy and keep building for it's interacting with the purchase and seller through message doc this. There is no need for customers to interact with sellers because every time the purchasers bid the details will be updated in the database. The purchasers making the heighest bid for an item before the close of the auction is declared as the owner of the item. If the auctioner or the purchaser didnot want to bid for the product their is fixed cutout price mentioned for every product. He can pay that amount directly and own the product.

Use Case View

* Use case diagrams model the functionality of the system.
* It consists of actors, use cases & relationships.

Use cases are

Display Products in an Auction

Add products to Lot
Delete Lot

Display Minimum Bid that can be placed on the Lot

Placed Bid on the Lot

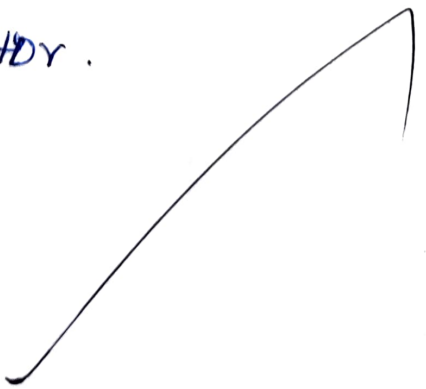
Withdraw Bid

Display List of Bid placed on the Auction
Display Bid details.

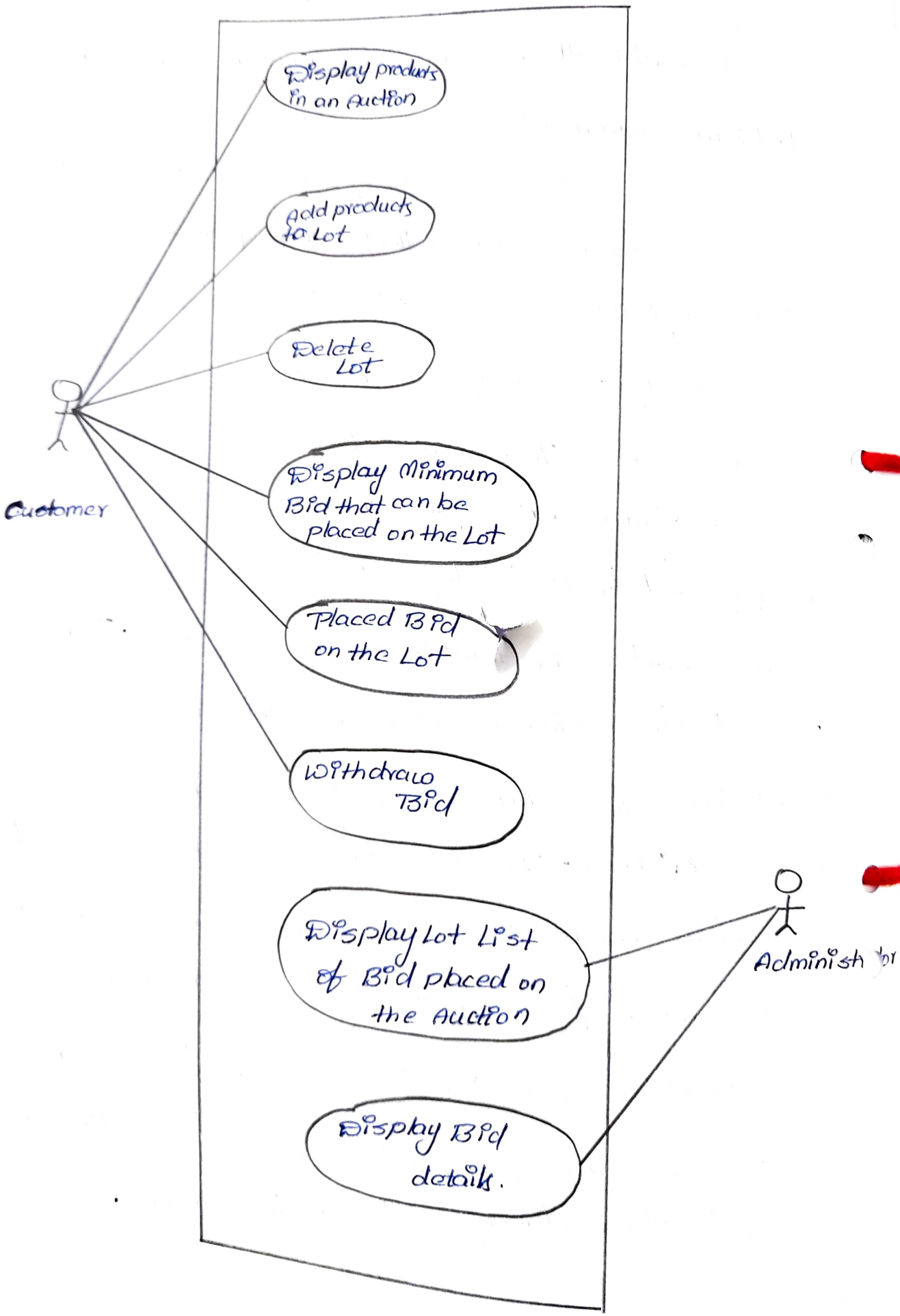
Actors are

Customer

Administrator.



Use Case Diagram

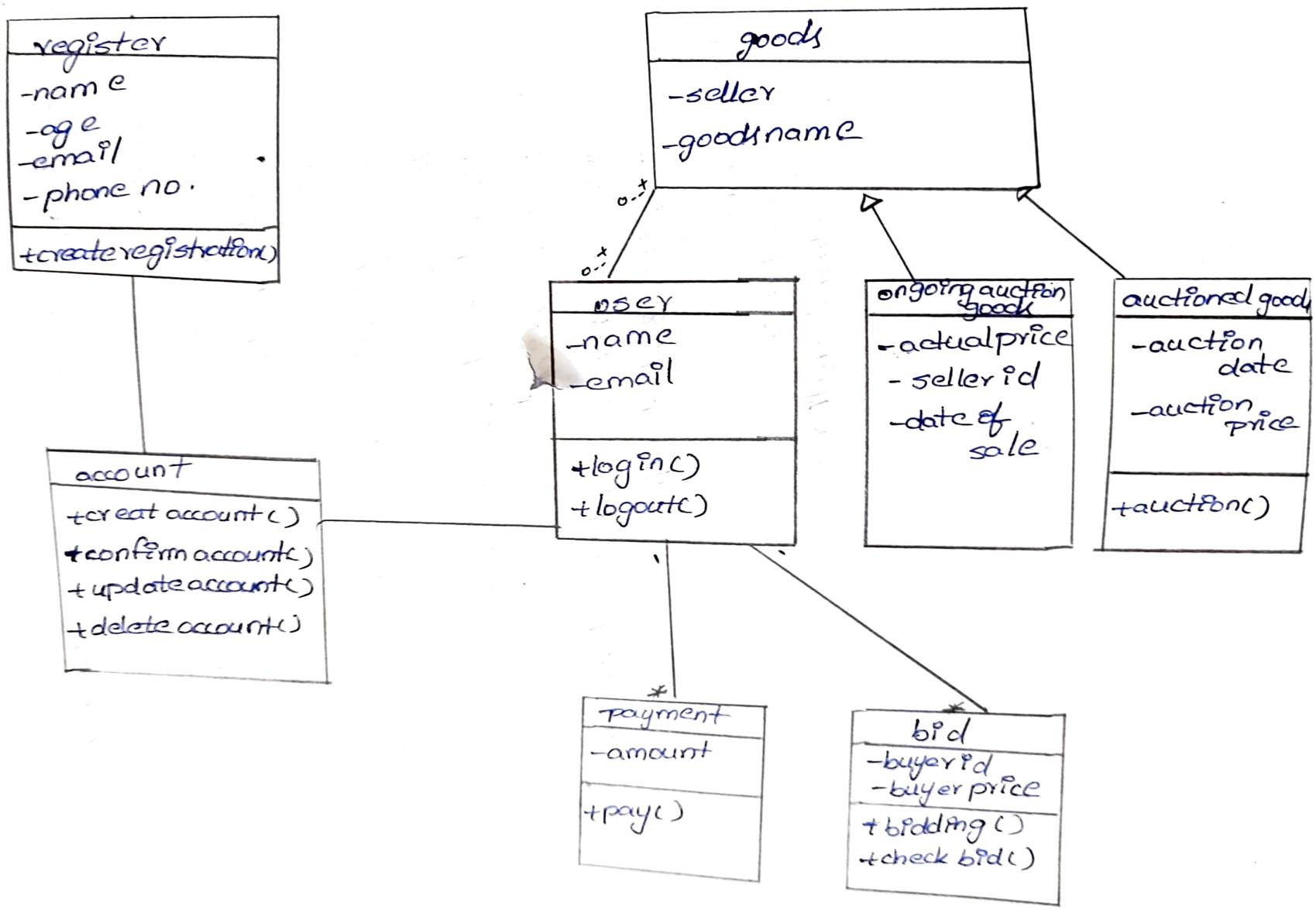


Class Diagram View

* class diagram describes the structure of a system by showing the system's classes, attributes, operations and relationships among them.

S.No	className	Attributes	Operations
1	register	name, age, email, phoneno.	create register()
2	goods	seller, goodsname	
3	account		creatacount(), confirm account(), update account, delete account() login(), logout()
4	user	name, email	
5	ongoing auctiongoods	actual price, sellerid, date of sale	
6	auctiongoods	auctiondate, auction price	auction()
7	Payment	amount	pay()
8	bid	buyerid, buyerprice	bidding(), checkbid().

Class Diagram



Sequence View

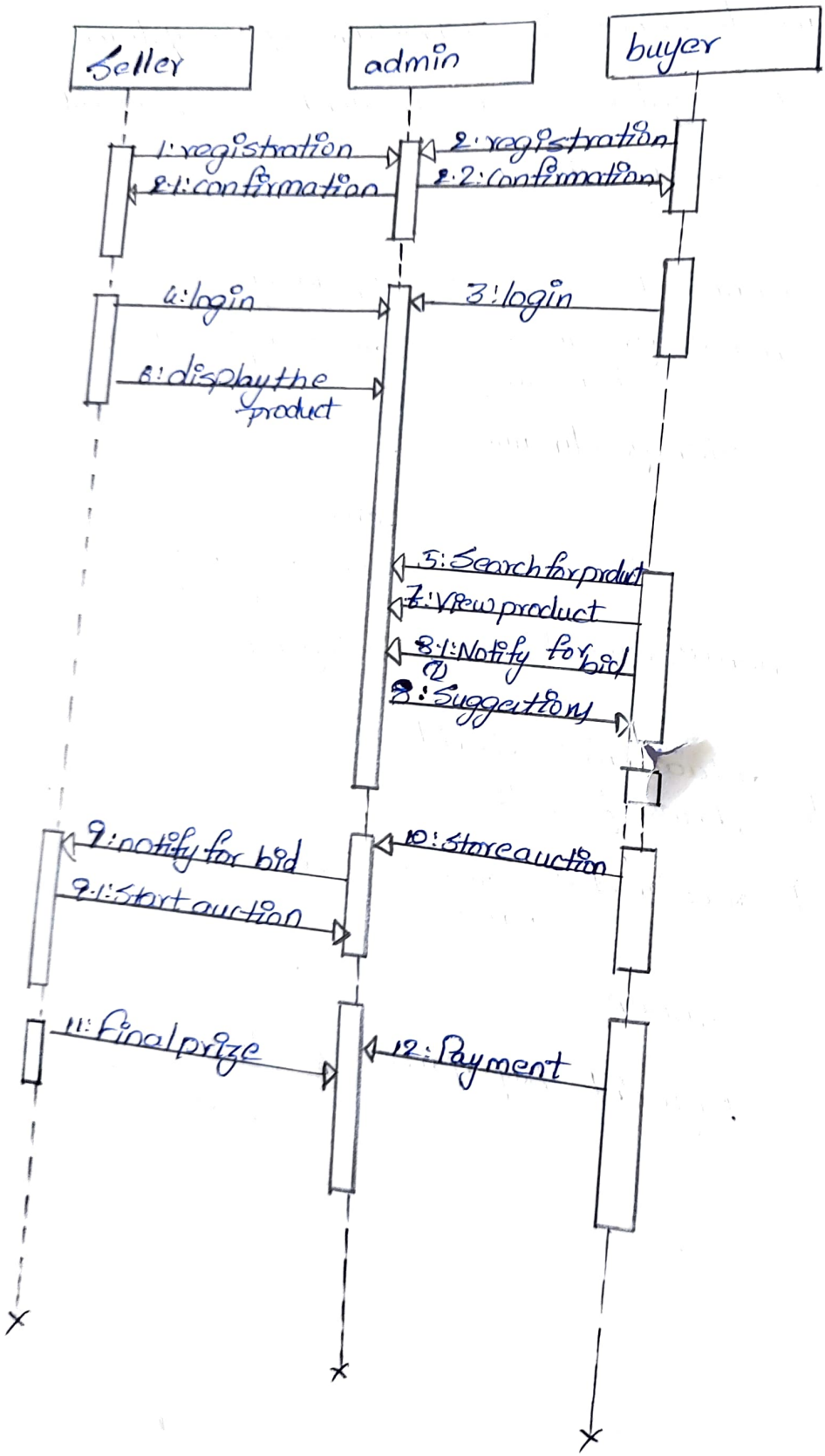
* It depicts the interaction between object in a sequence order

Objects are
Seller, admin, Buyer

Messages are

1) Seller registers to admin
2) Buyer registers to admin
3) Admin confirms registration of buyer.
4) Admin confirms registration of seller.
5) Seller logins through the admin provided registration.
6) Buyer logins through the admin provided registration.
7) Buyer searches for the product through admin.
8) Seller displays the product through admin.
9) Buyer views the product shown by admin.
10) Suggestions are given to buyer by admin.
11) Notifications of bid for the product searched to buyer
12) Notification of bid for the product searched to seller.
13) Auction is started between Admin and seller.
14) Auction is started between buyer and Admin.
15) Final price is shown to admin by seller.
16) Buyer pays the payment.

Sequence Diagram

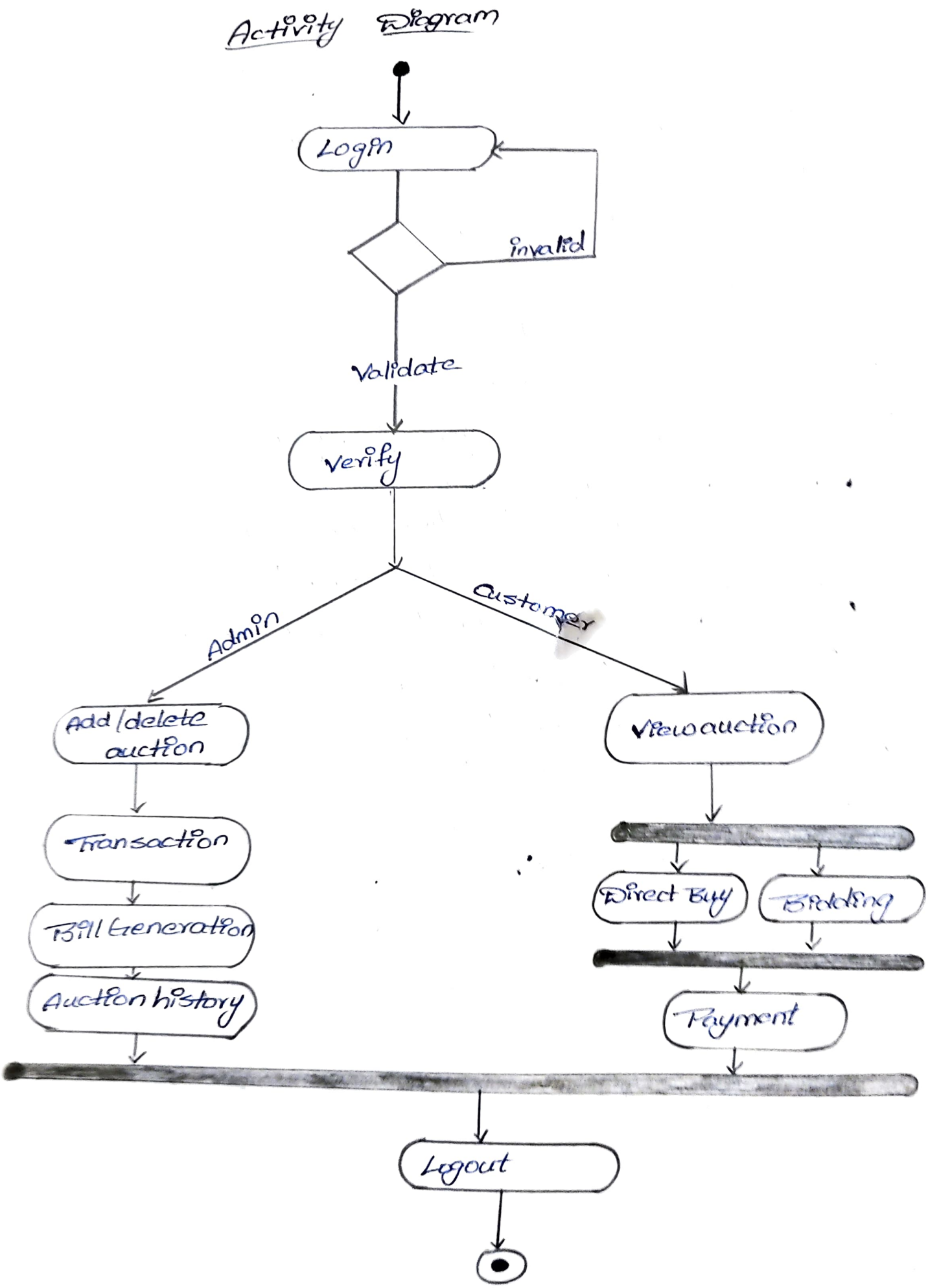


Activity View

It represents flow from one activity to another activity.

Activities are

1) Login - buyer/seller
2) If it is invalid again login
3) If valid verify the login details.
4) If the login person is admin a set of activates can be done.
5) If the login person is admin then the admin can add/delete auction.
6) If login person is customer a set of activities are done.
7) Transactions can also be done by the admin.
8) Bill generation of the transaction.
9) Auction history can be viewed.
10) If login person is customer then they can view auction.
11) While viewing they can direct Buy or bid the article.
12) Either way payment is done.
13) Customer logs out.


Activity Diagram
Login
invalid
Validate
verify
Admin
Customer
Add/delete auction
Transaction
Bill Generation
Auction history
View auction
Direct Buy
Bidding
Payment
Logout

Collaboration View

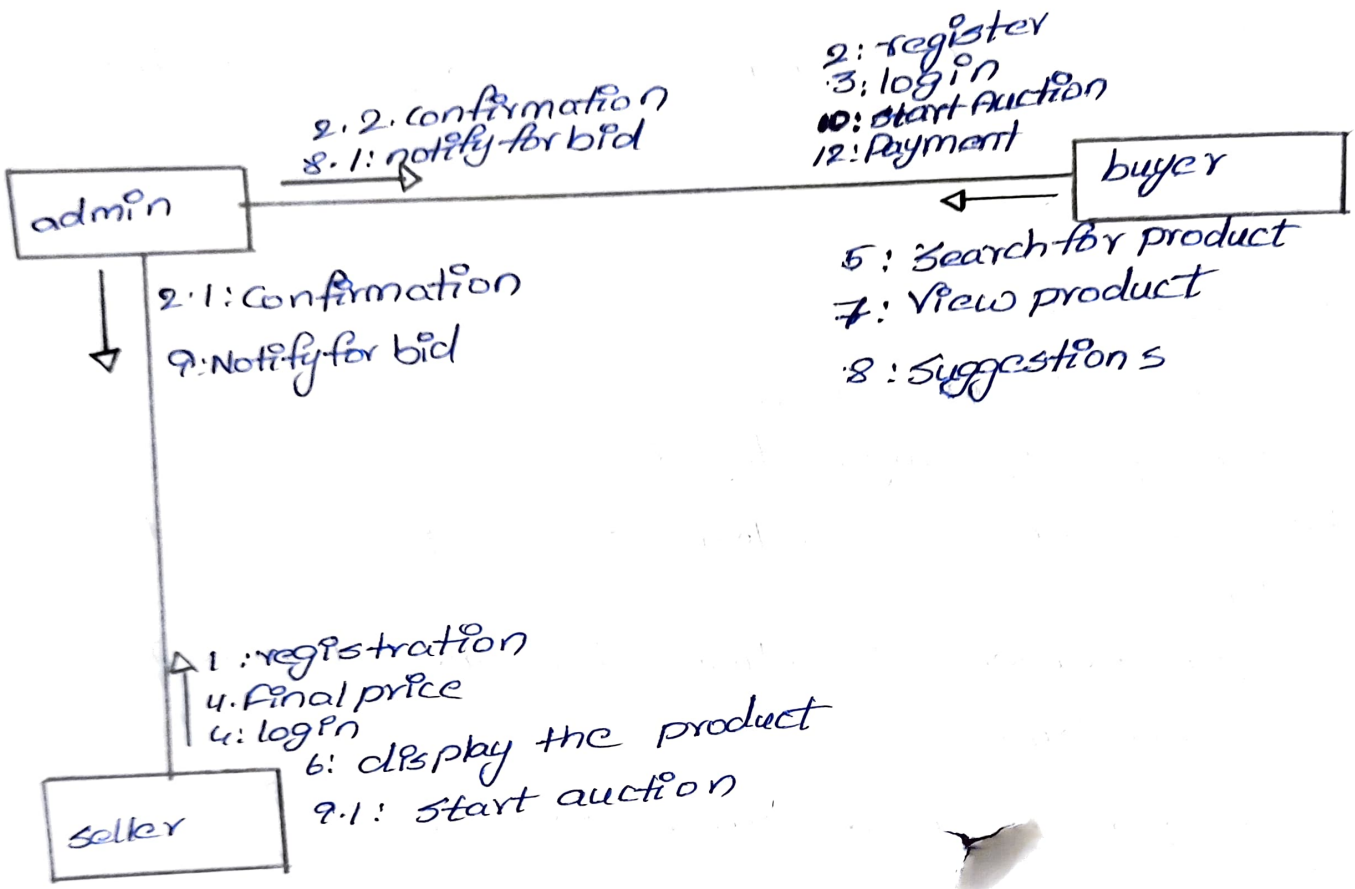
* It shows how various objects interacts with each other

objects are
admin, buyer, seller

Messages are

1) Registration of seller to admin
2) Registration of buyer to admin
2.1) Admin sends confirmation to seller
2.2) Admin sends confirmation to buyer
3) Buyer logins to the server through admin.
4) Seller logins to the server through admin.
5) Buyer searches for the product.
6) Seller displays the product
7) Buyer views the product
8) Buyer gets suggestions based on viewed items.
9) Admin notifys for a bid to seller.
10) Admin notifys for a bid to buyer.
11) Seller starts auction.
12) Buyer starts auction.
13) Seller says final price
14) Buyer makes a payment.

Collaboration diagram



Component View

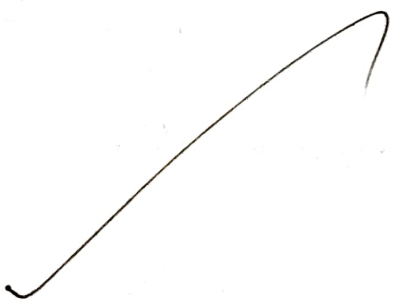
These are used to visualize the organizations and relationships among components.

Components are

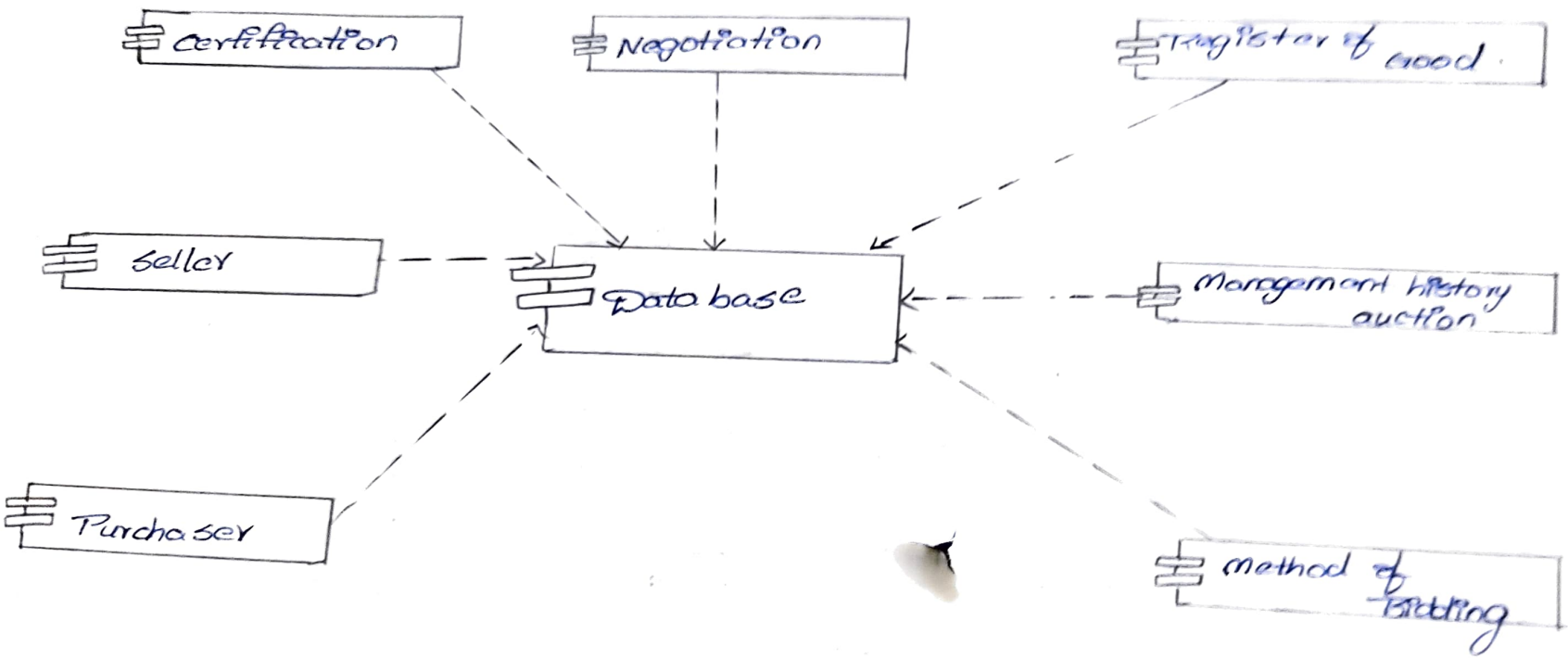
- Certification
- Registration of Good
- Seller
- Negotiation
- Management History auction
- Purchaser
- Method of Bidding.

Relationship

- Dependancy.



Component Diagram



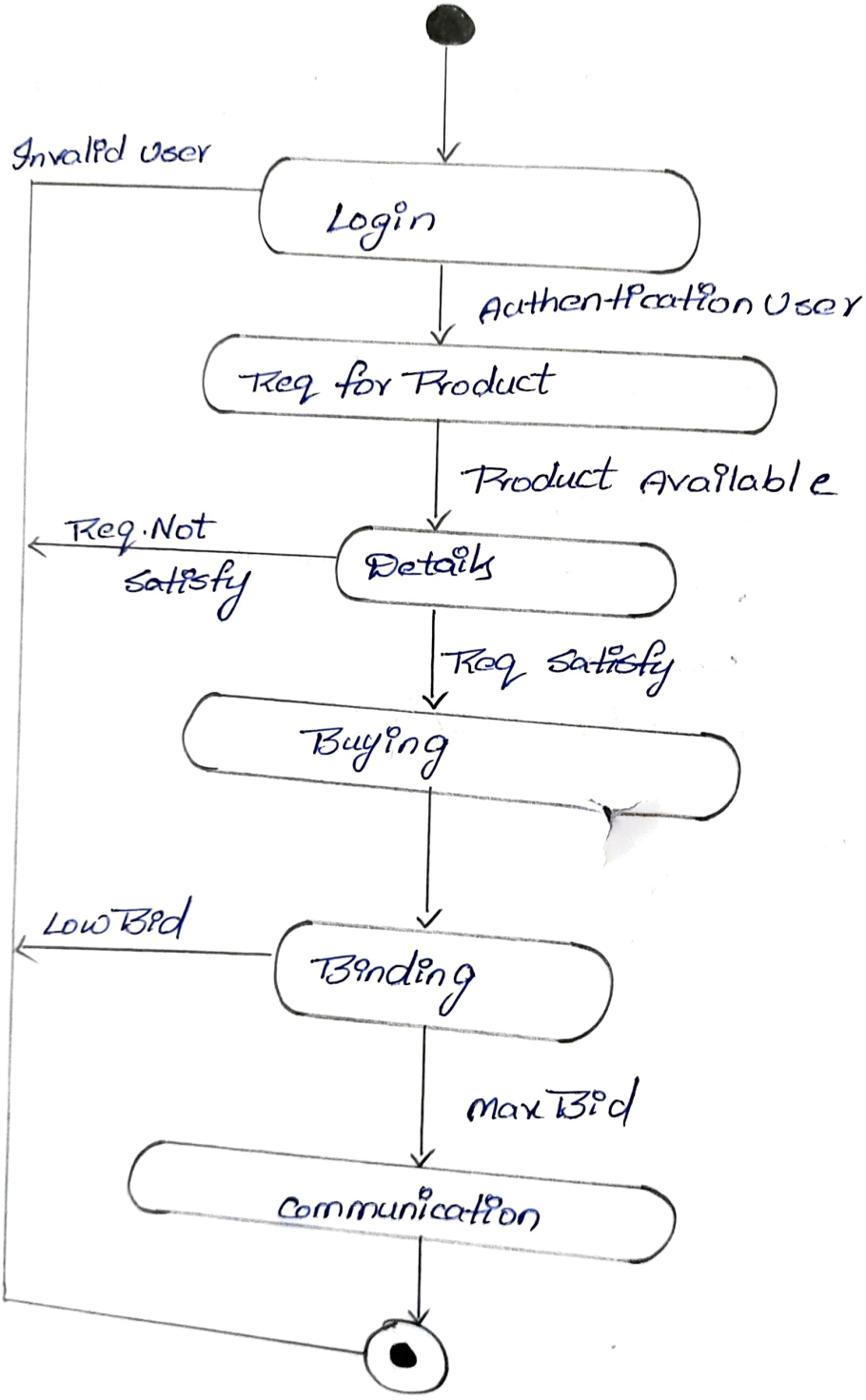
Statechart View :

* It defines different states of objects during its life time and are changed by events.

States are

- Login
- Req for Product
- Details
- Buying
- Bidding
- Communication

State chart Diagram



Deployment View:

* It represents the deployment view of the system
* It consists of nodes which are used to display the application

Nodes are

Webserver

Admin

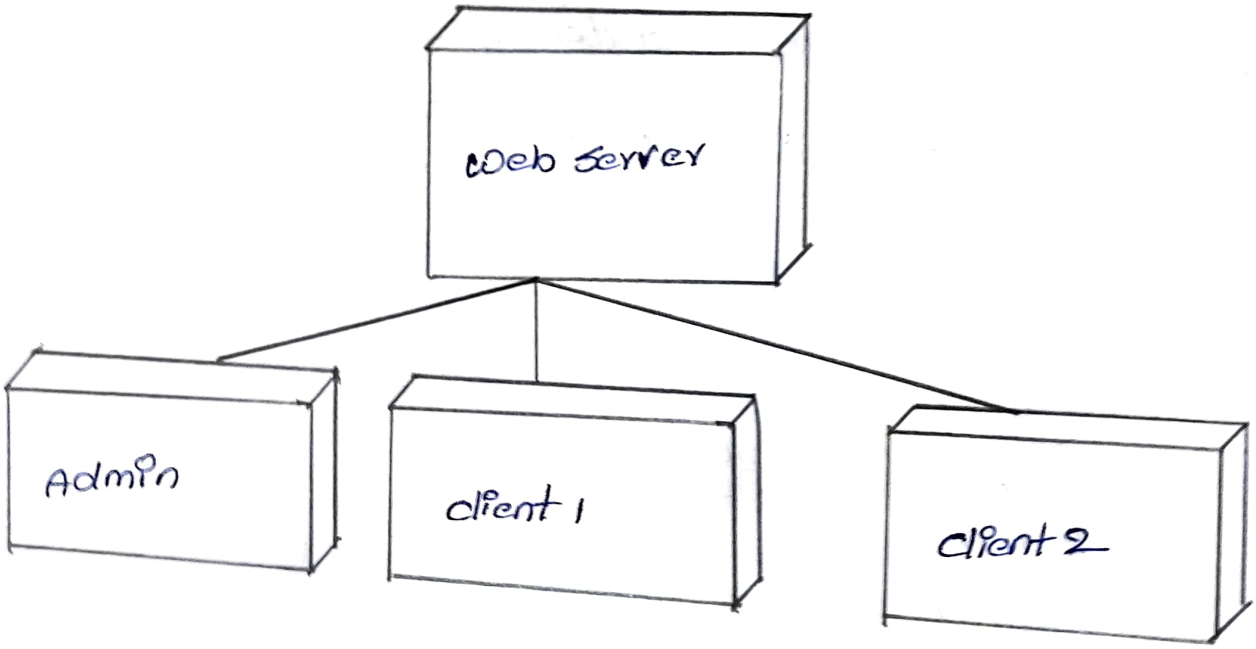
client1

client2

Result:

Thus all the values of online Auction sales are modelled.

Deployment Diagram



Case Study-5:

Aim To model two floor elevator simulator.

Problem Statement The elevator has the basic function that all elevator systems have, such as moving up and down, open and close doors, and of course pick up passengers. The elevator is supposed to be used in a building having floors numbered from 1 to max floor, where the first floor except for lobby. There are car call button in the car corresponding to each floor. For every floor except for the top floor and the lobby there are two hall call buttons for the passengers to call for going up and down. There is only one down hall call button at the top floor & one up hall call button in the lobby. When the cars stop at a floor, the doors are opened and the car lantern indicating the current direction of car. The car moves fast btw floors, but it should be able to slow down early enough to stop at a desired floor. When an elevator has no requests, it remains at its current floor with its door closed.

In order to the certificate system safety, emergency break will be triggered & the car will be forced to stop under any unsafe conditions.

Use case View

* Use case diagrams model the functionality of the system.
* It contains of actors, usecases & relationships

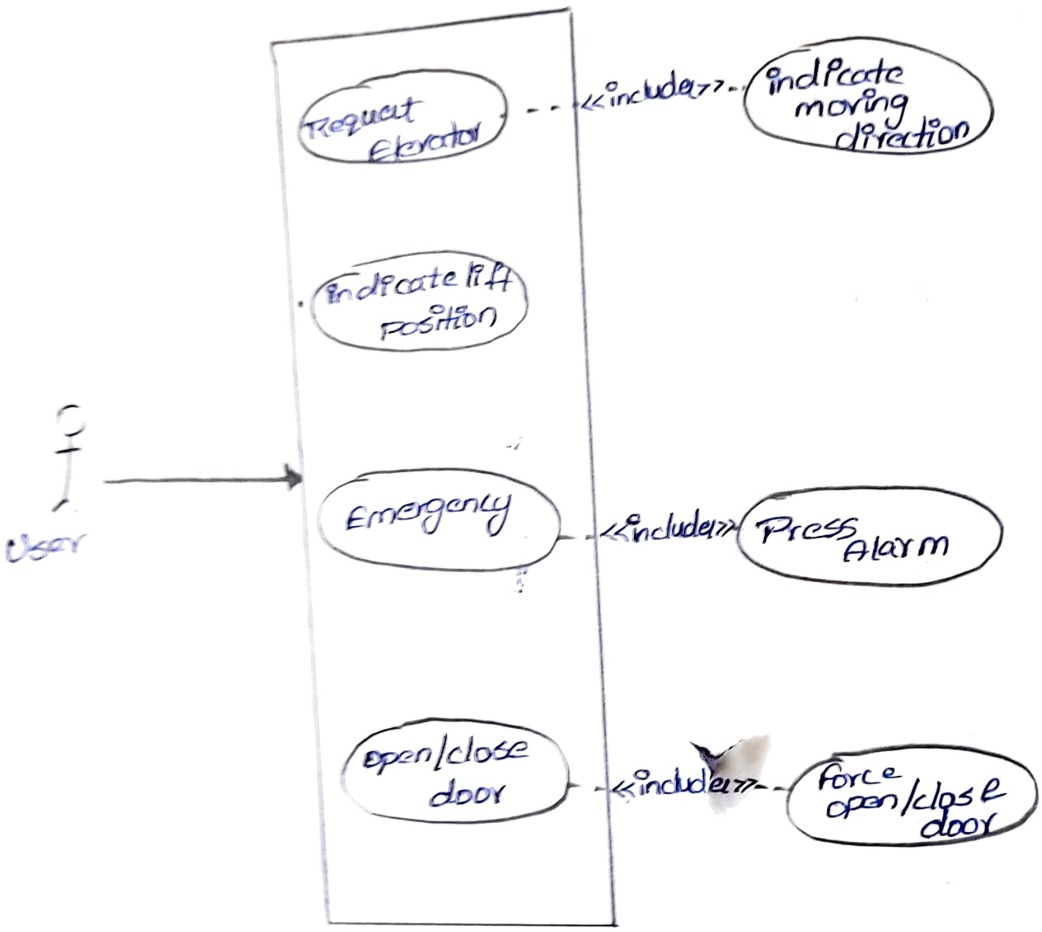
Use cases are

- Requests Elevator
- Indicate Lift position
- Emergency
- open/close door

Actors are

- User

Use Case Diagram

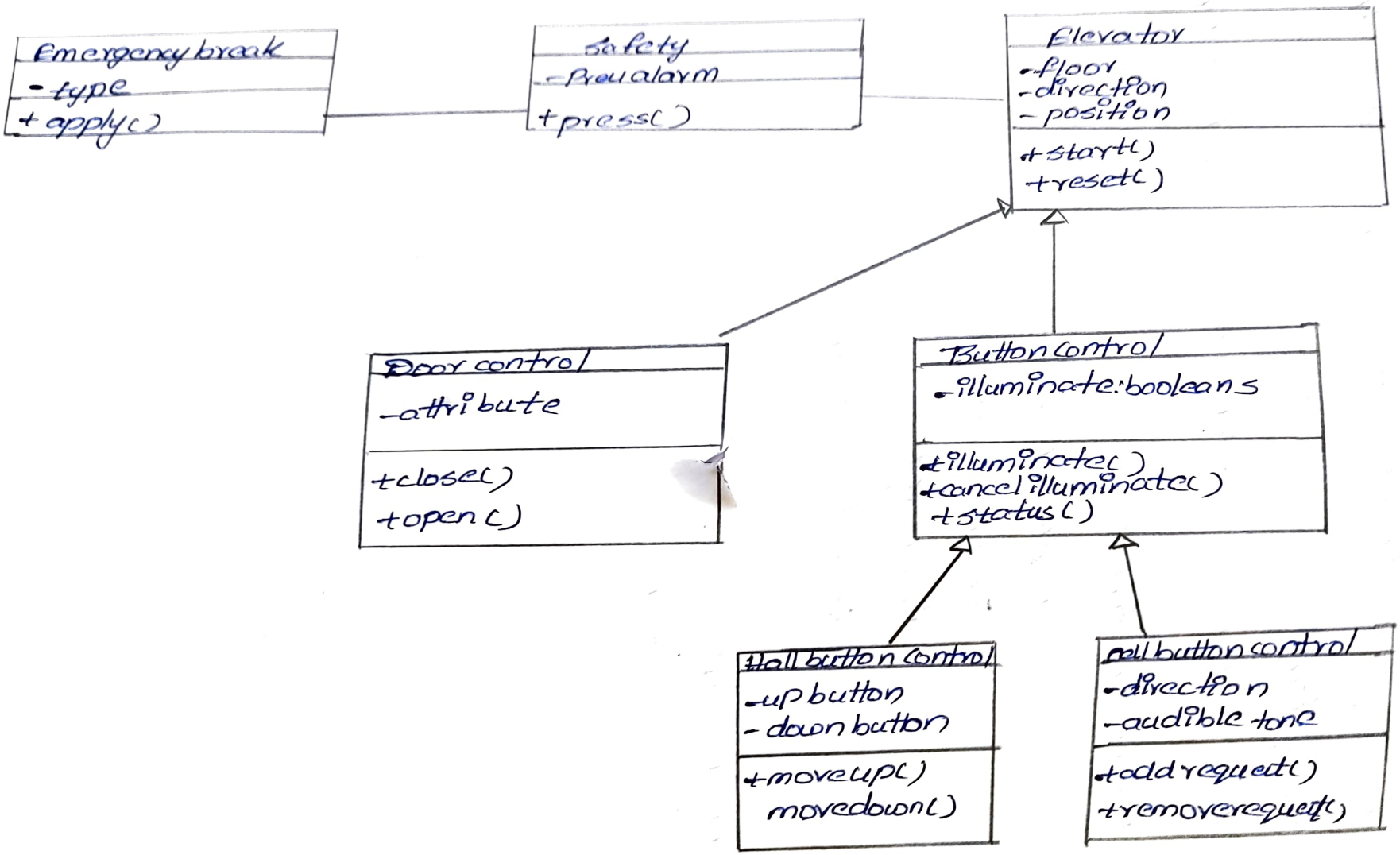


Class Diagram View

* class diagram describes the structure of a system by showing the systems' classes, attributes, operations and relationships among them.

S.NO	Class Name	Attributes	Operations
1	Emergency break	type	apply()
2	Safety	Pressalarm	press()
3	Elevator	floor, direction, position	start(), reset()
4	Door control	attribute	close(), open()
5	Button control	illuminate	illuminate(), cancel-illuminate(), status()
6	Hallbutton control	upbutton, down button	moveup(), move down()
7	Cell button control	direction, audible tone	add request(), remove request()

Class Diagram



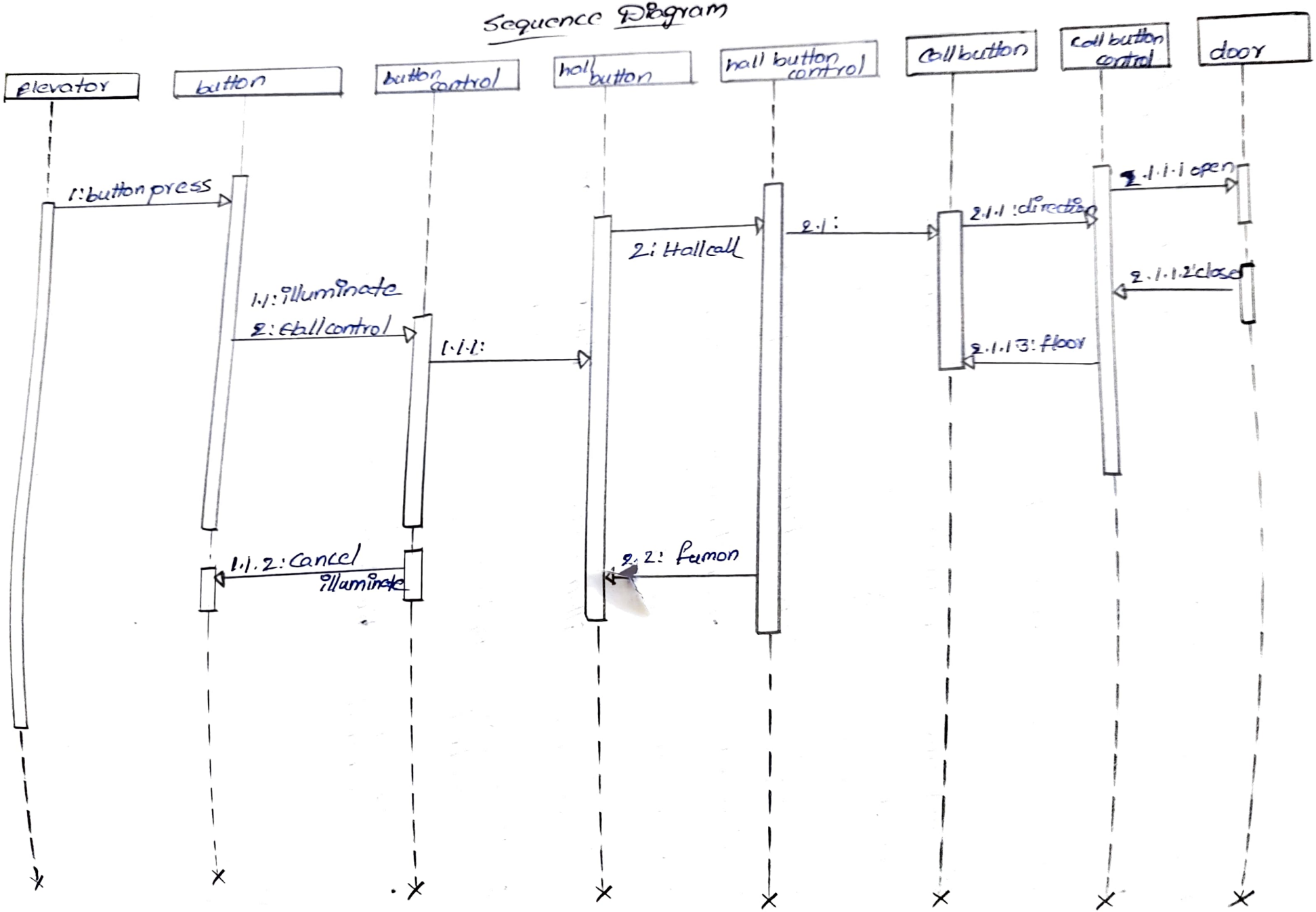
Sequence View :

* It depicts the interaction between objects in a sequence order

objects are
- Elevator
- button
- button control
- hall button
- call button
- hall button control
- call button control
- door

messages are
- button press
- illuminate
- cancelilluminate
- Hall call
- fumon
- direction
- floor
- open
- close.

Sequence Diagram



Collaboration View

* It shows how various objects interacts with each other

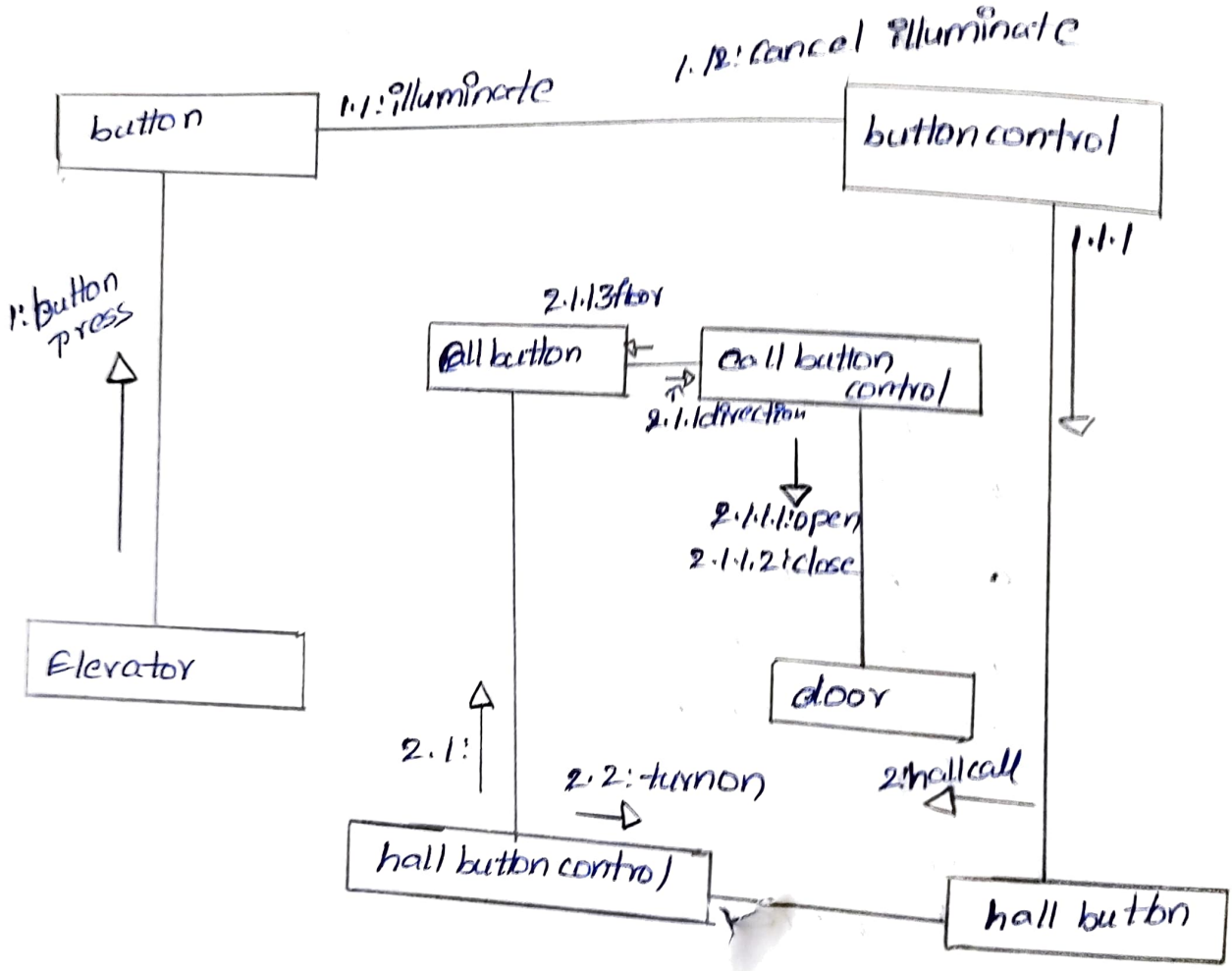
objects are

button
button control
hall button
hall button control
call button
call button control
elevator
door

Messages are

1: button press by user
1.1: illuminate
1.1.2: cancel illuminate
2: hallcall
2.1:
2.2: Turnon
2.1.1: direction
2.1.2.1: open
2.1.1.2: close
2.1.1.3: floor

Collaboration Diagram



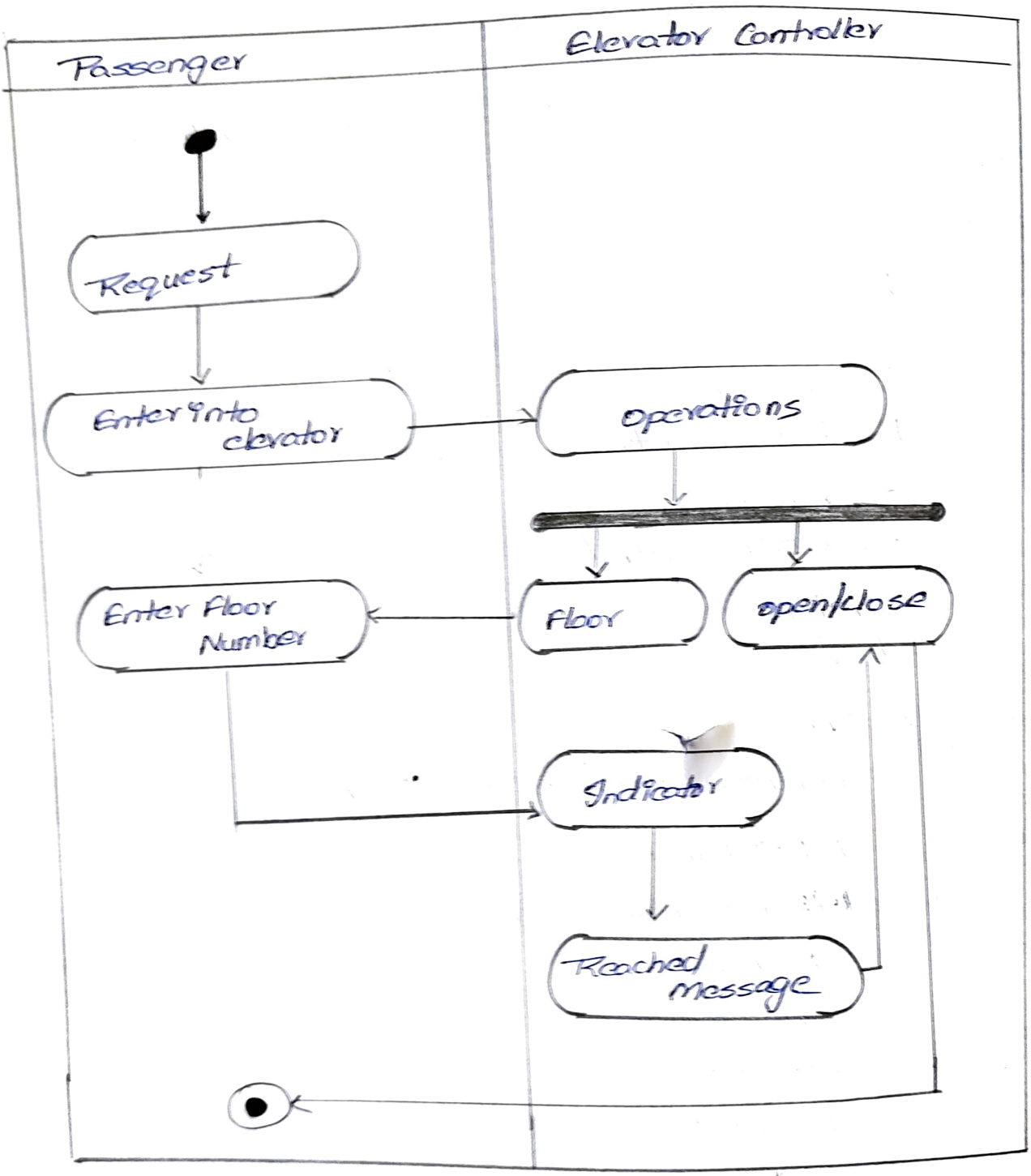
Activity View

It represents flow from one activity to another activity.

Activities are

- Request
- Enter into elevator
- operations
- Floor
- open/close
- Enter Floor Number
- Indicator
- Reached message.

Activity Diagram



State chart View

* It defines different states of objects during the lifetime & are changed by events.

States are

Idle - The initial position of elevator when the floor is chosen.

Moving up/down - Based on choice of floor the elevator moves up/down.

Stopping - The elevator stops at the designated floor.

Door opening - only after stopping the elevator door opens.

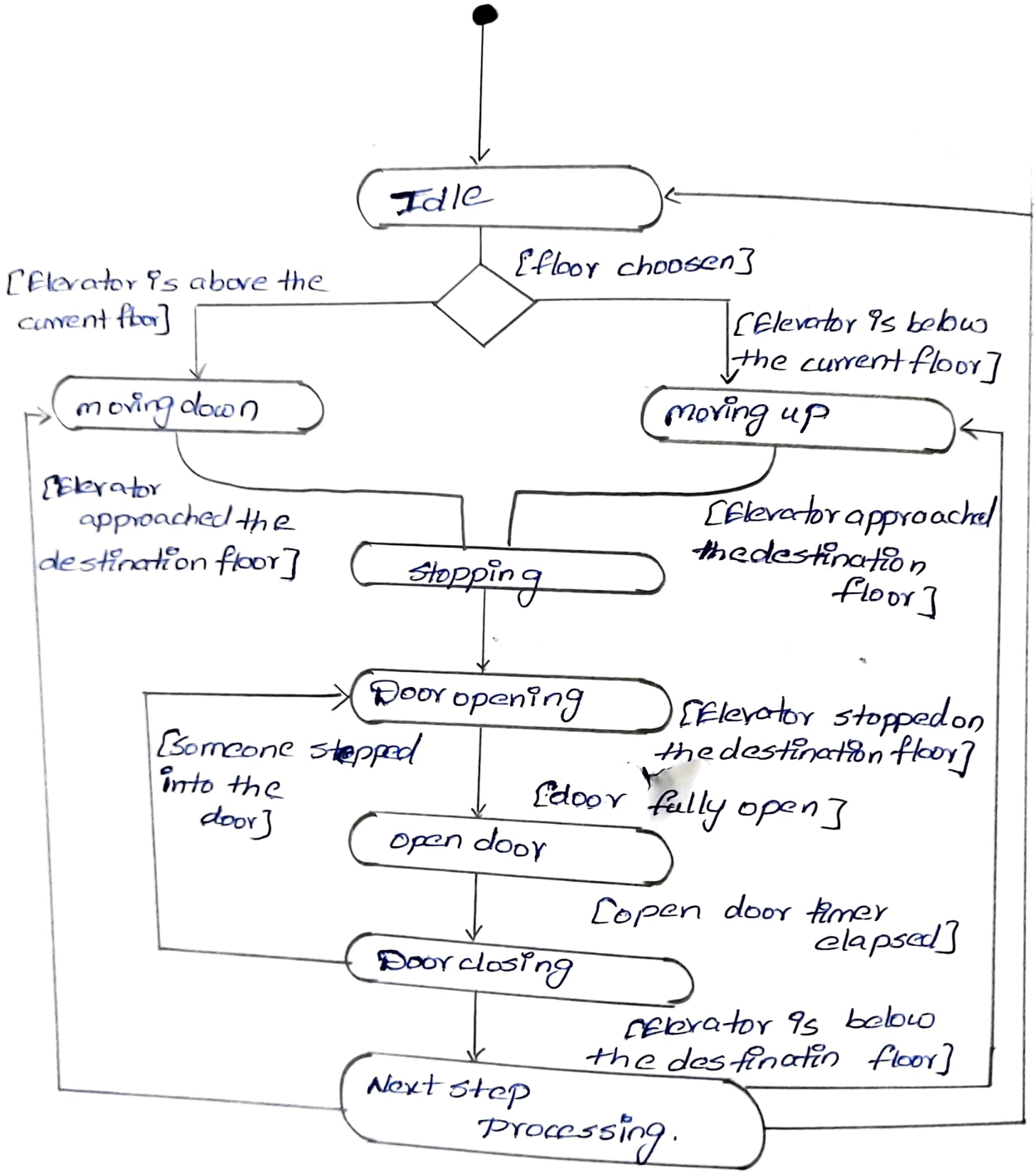
Door closing - After the person steps out the doors are closed after certain time.

Next step - Finally the elevation moves to the next step up/down.

->Elevator moves down if it is aboves the current floor.

->Elevator moves up if it is bellow the current floor.

State chart Diagram



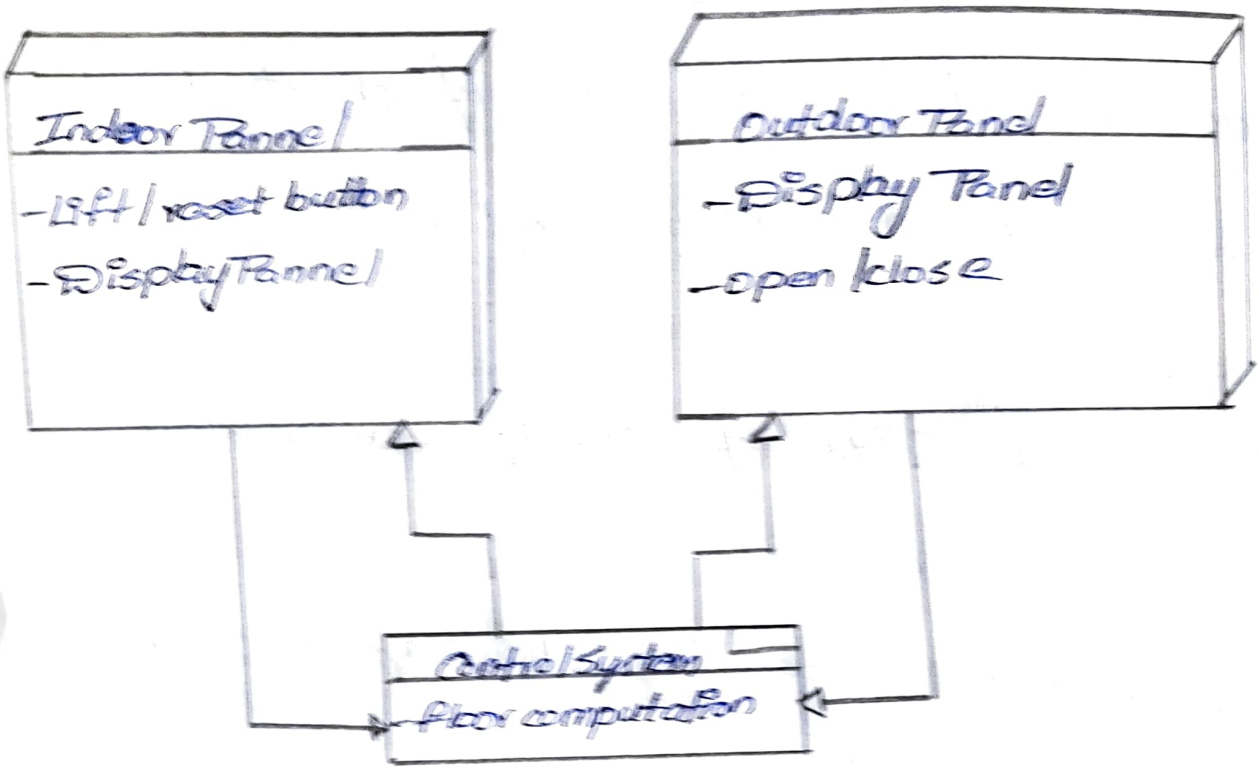
Deployment View

* It represents the deployment view of the system
* It consists of nodes which are used to display the elevator options.

Nodes are

- Indoor pannel
- outdoor pannel
- Control system.

Deployment Diagram



Component View

These are used to visualize the organizations and relationships among components.

Components are

- Elevator button
- Elevator
- Sensor
- Elevator control system
  - Door
  - Floor
  - Elevator button

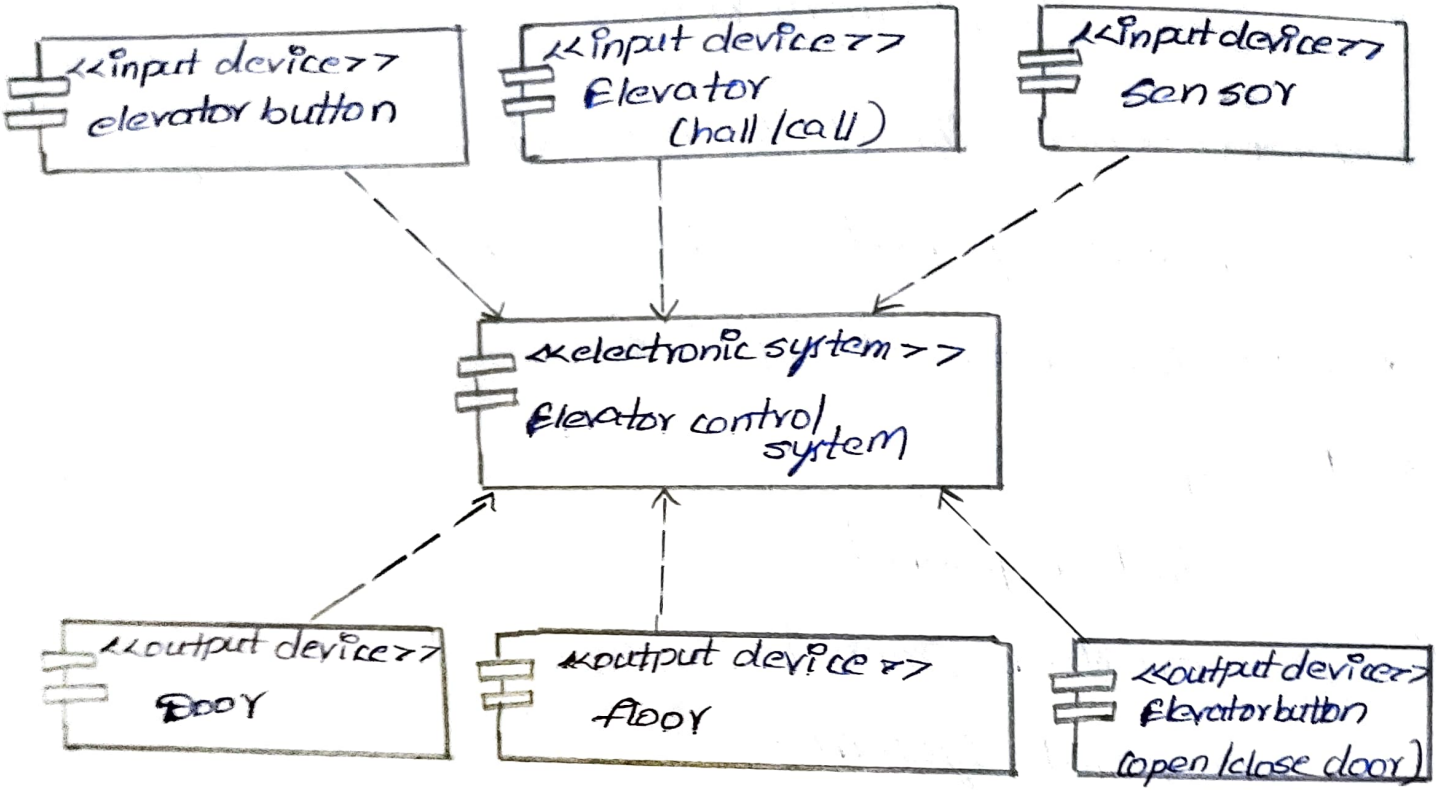
Relationship

Dependency:

Result:

Thus all the values of two step elevator system are modelled.

Component Diagram



Case Study-6:

Aim To model home appliance control system.

Problem Statement

A home appliance control system is a system which provides various services to remotely operate on home appliances, such as microwave oven, TV and garbage door etc through remote devices such as mobile phone, desktop and Palm-Top. A home appliance control system is a system which is controlled by a remote system such as a mobile phone or a palm-Top and at sometimes controls, monitor and coordinates home appliances such as air conditioner, microwave oven, garbage doors, TV set, VCR, audio controller, indoor/outdoor lights, water sprinkler, home security system, bath tub controller etc.

In order to activate home appliances and to allow for communication between the different devices in a system, and for coordination among the various process running on such devices. The systems administrator of the HACS, system has the ability to add a new remote device and configure it with HACS or delete an existing one when it is not used. Also the system administrator can create an account for a new user or delete existing account of it is no longer used.

Use Case View

* Use case diagrams model the functionality of the system.

* It contains of actors, usecases & relationships

Usecases are

Authentication
Select Appliance List
Select Appliance operations
Cancel Appliance operations
Get status
HACS applance controller
Operate microwave
operate sprinkler
operate pet feeder
Logout.

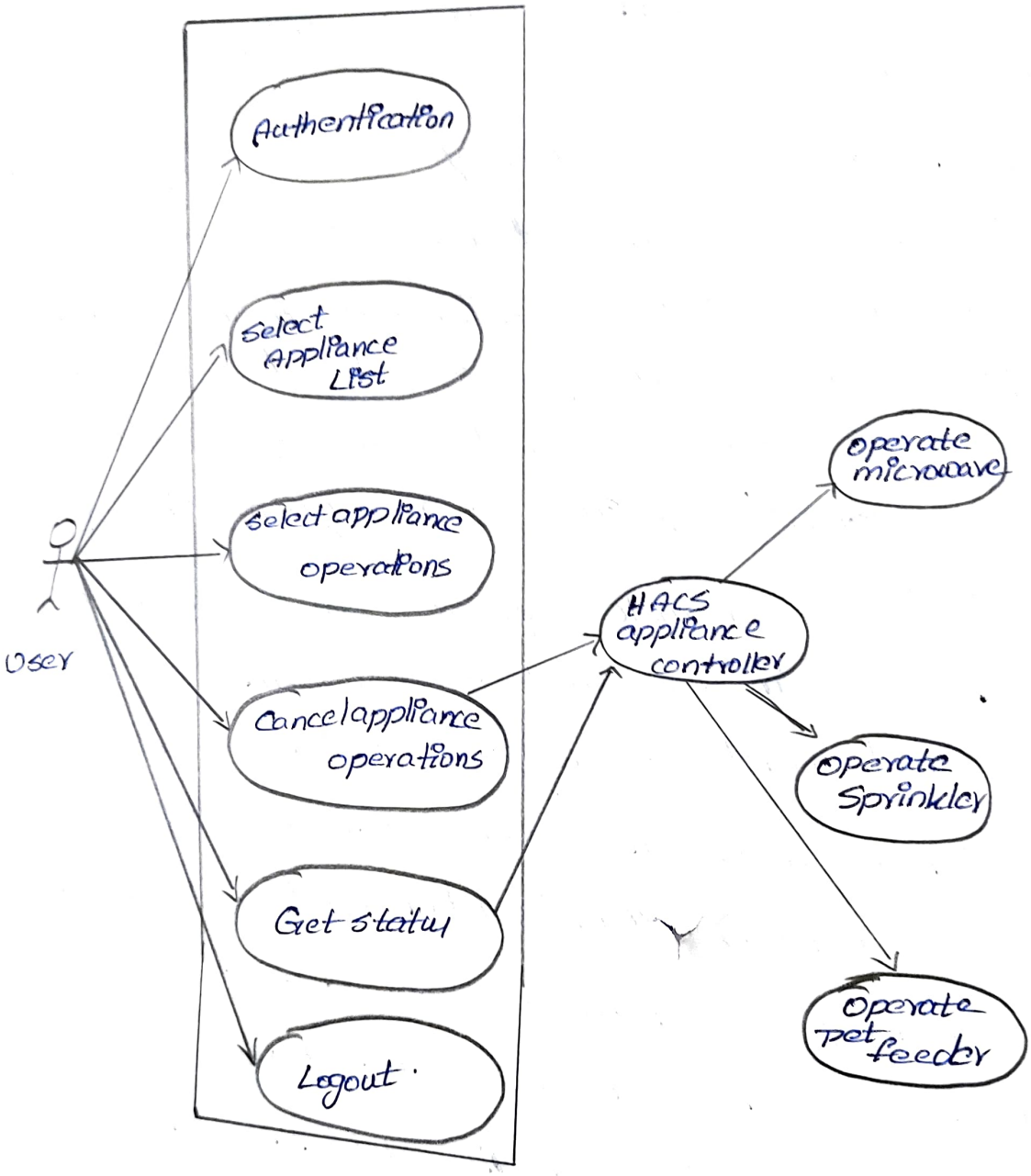
Actors are

User

Relationships are

Association

Use case Diagram

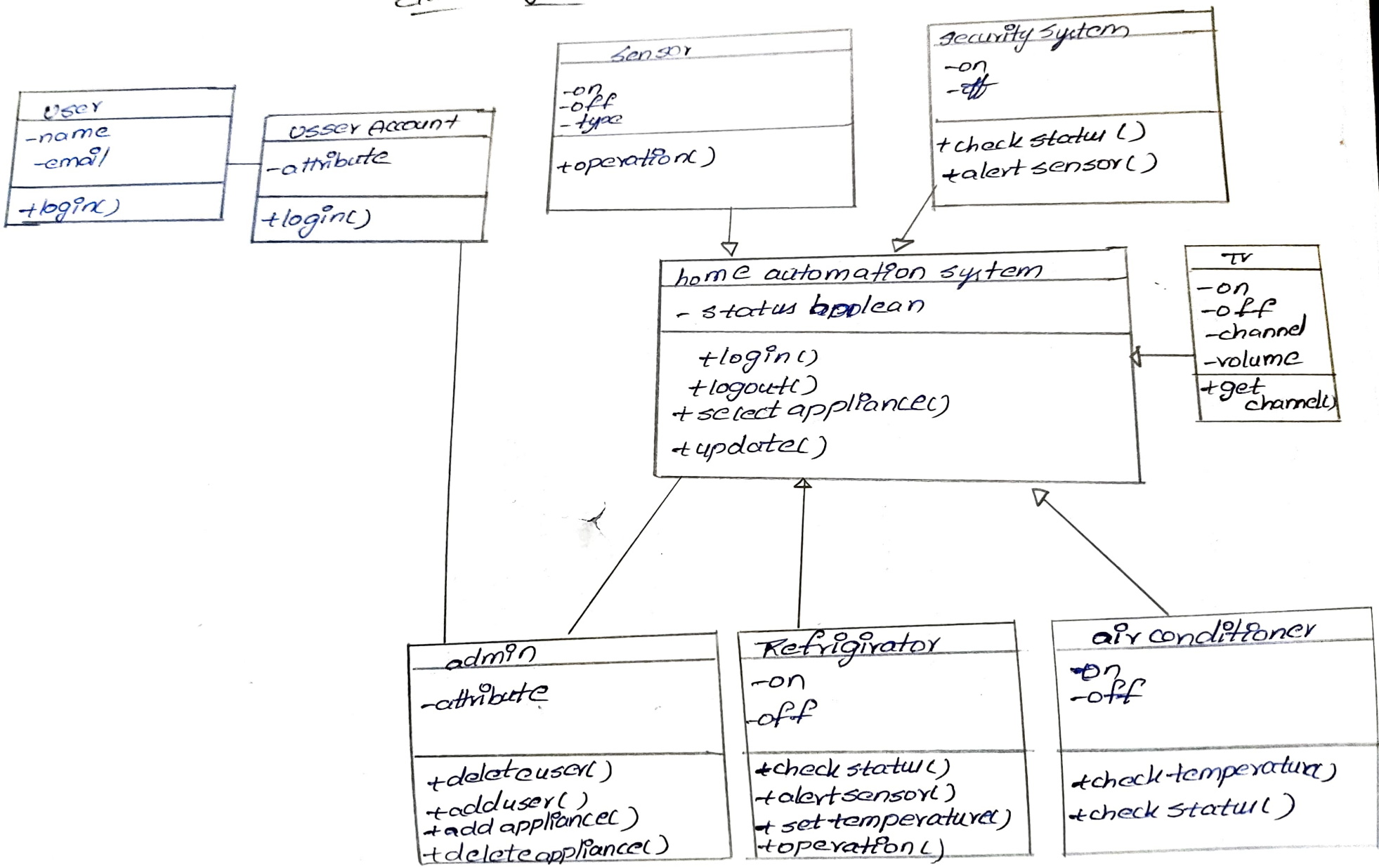


Class diagram View

* Class diagram describes the structure of a system by showing the system's classes, attributes, operations & relationships among them.

S.NO	Class Name	Attributes	Operations
1	User	name, email	login()
2	Useraccount	attribute	login()
3	admin	attribute	delete user(), adduser(), deleteappliance(), addappliance()
4	Home automation system	status boolean	login(), logout(), update(), select appliance().
5	Refrigerator	on, off	check status(), alertsensor(), set temperature(), operation()
6	air conditioner	on, off	set temperature(), check status(), getchannel()
7	TV	on, off, channel, volume	getchannel()
8	Security system	on, off	check status(), alert status()
9	Sensor	on, off, type	operation()

Class Diagram



Sequence View

* It depicts the interaction between objects in a sequence order

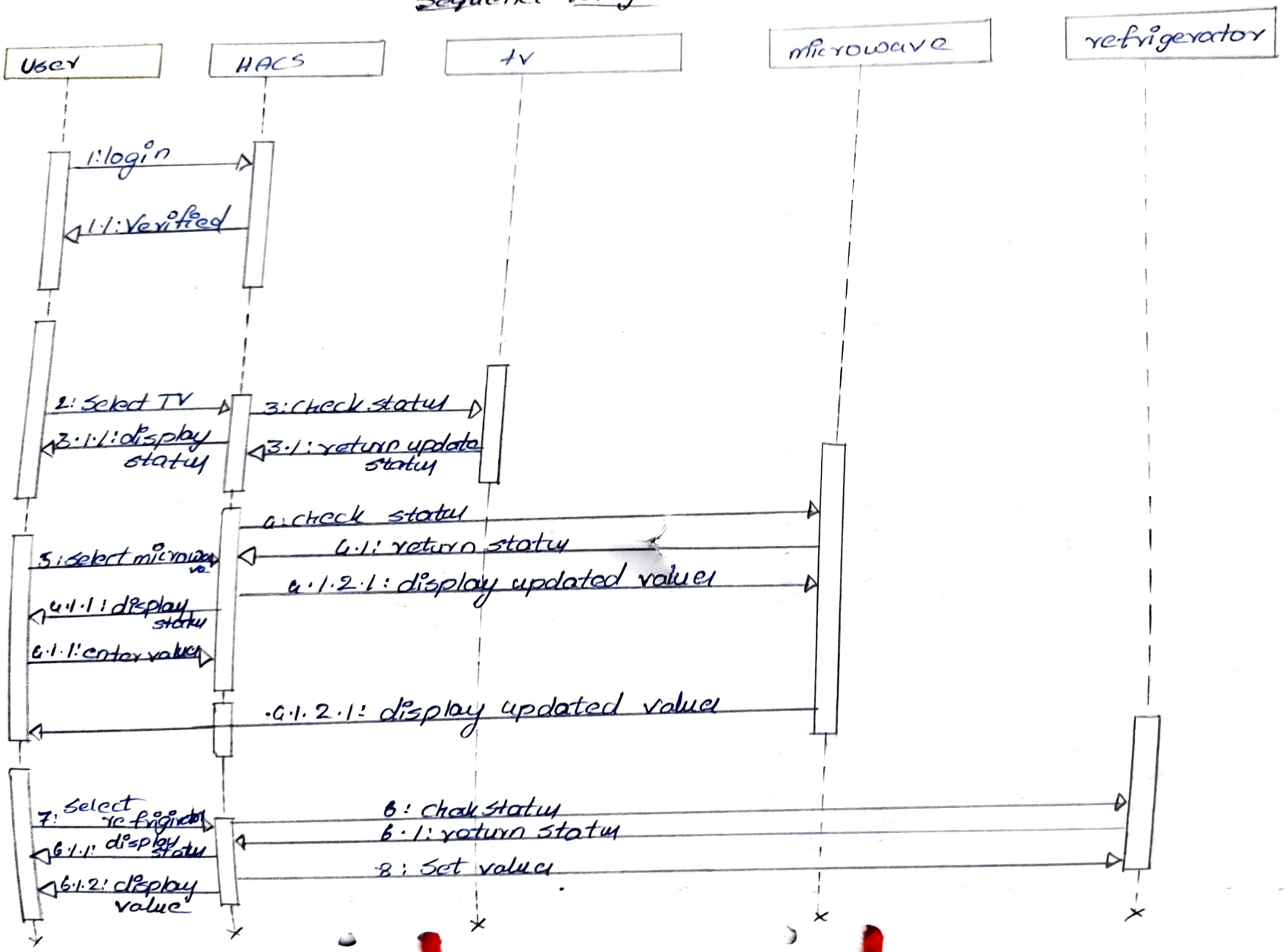
objects are

User, HACS, tv, microwave, refrigirator

Messages are

1) User logins in to HACS
2) HACS verifies the user
3) User selects TV
4) HACS checks the status of TV
5) TV returns the status of HACS.
6) HACS display TV status to user.
7) User select microwave.
8) HACS checks stacks of microwave.
9) microwave returns the status to HACS.
10) HACS display status of microwave to user.
11) User enter values of microwave to HACS
12) HACS displays updated values to microwave
13) microwave displays updated values to user.
14) User selects refrigirator.
15) HACS check status of regirator.
16) Regrigirator returns status to HACS.
17) HACS displays status to user
18) HACS set values for refrigirator.
19) HACS display values of refrigirator to user.

Sequence Diagram



Collaboration View

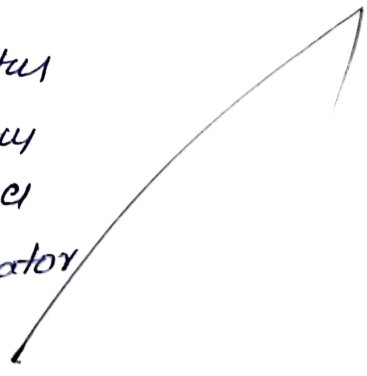
* It shows how various objects interact with each other.

Objects are

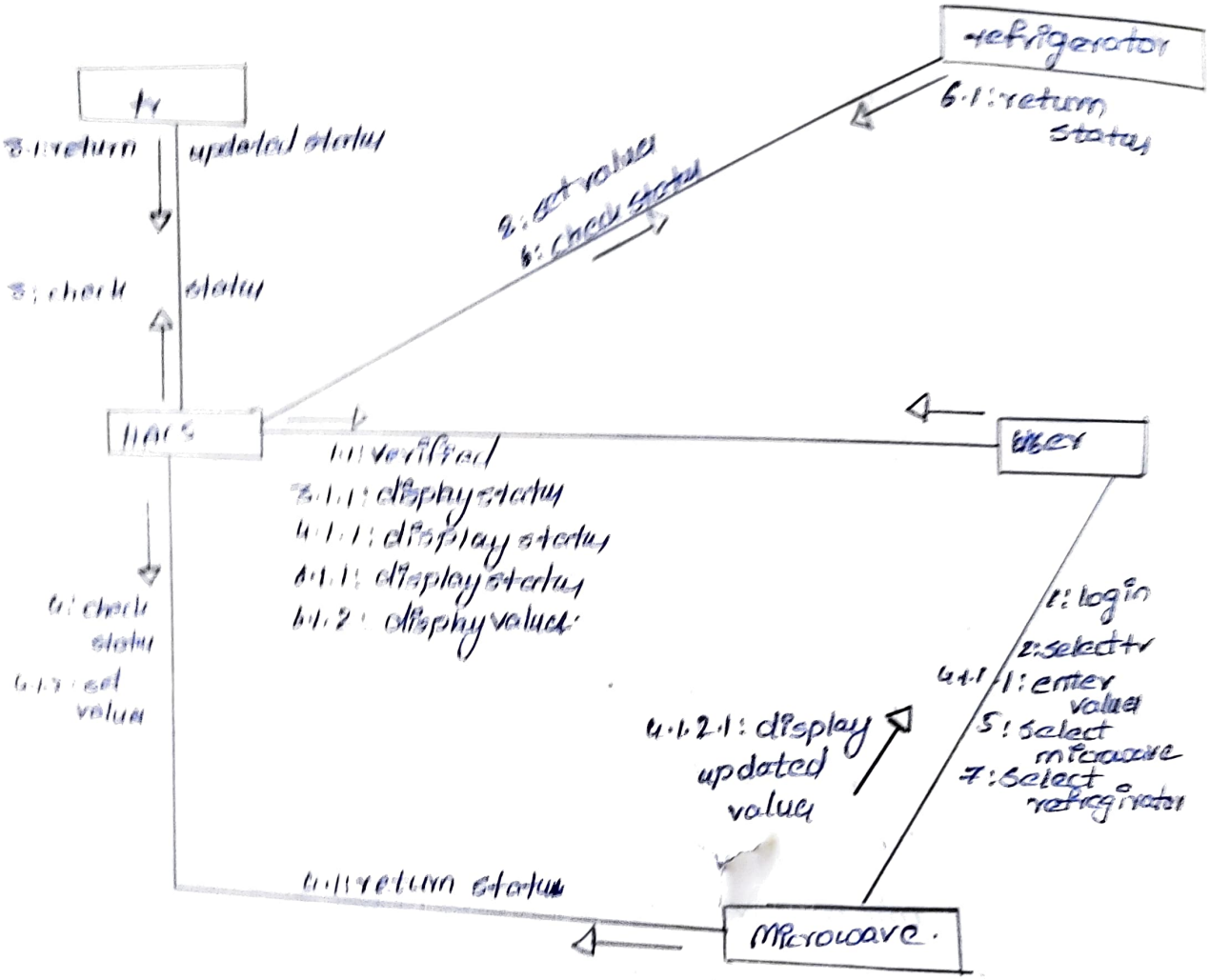
TV, HACS, microwave, user, refrigirator

Messages are

1: login
1.1: verified
2: select TV
3: check status
3.1: retur updated status
3.1.1: display status
4: check status
4.1: return status
4.1.1: display status
4.1.1.1: return value
4.1.2: set value
4.b.2.1: display updated value
5: select microwave
6: check status
6.1: return status
6.1.1: display status
6.1.2: display value
7: select refregirator
8: set value.



Collaboration Diagram



Component View

These are used to visualize the organization and relationships among components.

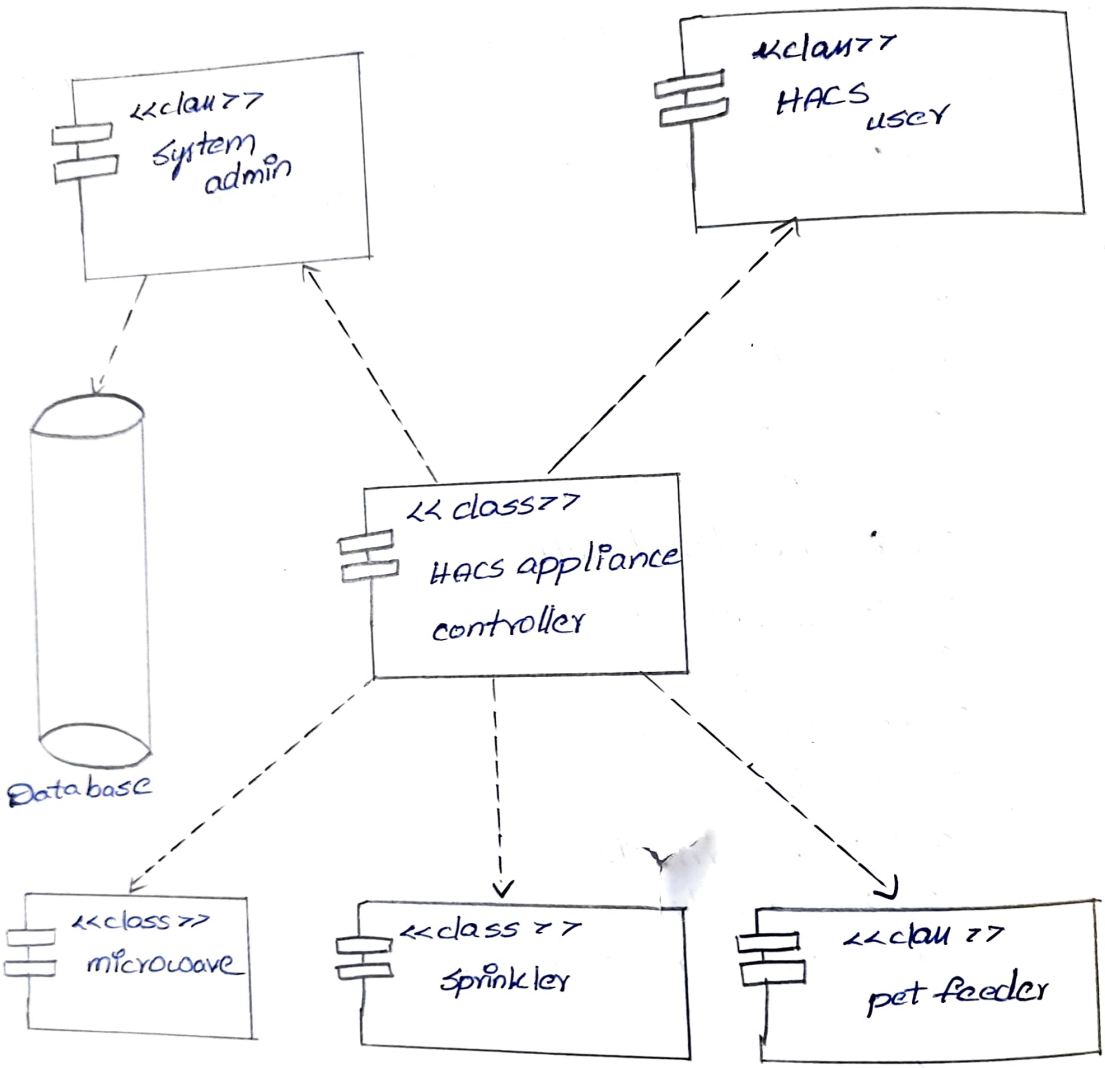
Components are

- Sugstem Admin
- HACS user
- HACS appliance controller
- microwave
- sprinkler
- pet feeder
- Database

Relationships

- Dependency,

Component Diagram



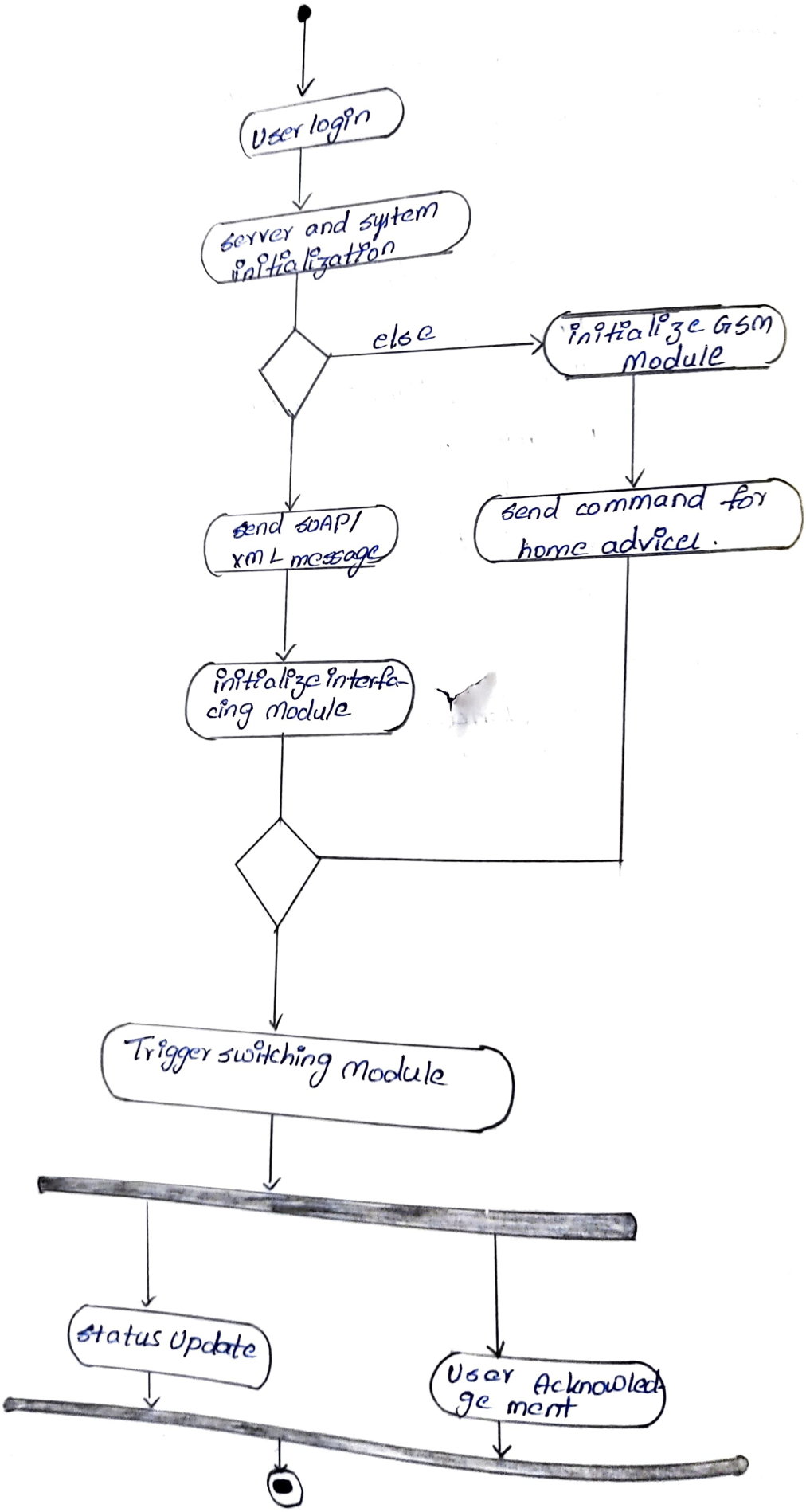
Activity View:

It represents flow from one activity to another activity.

Activities are

1) User logins into the server.
2) Server and system initialization begins.
3) If it is successful sends SOAP/XML message.
4) Else it initializes GSM module.
5) The XML message signals to initialize interfacing module.
6) The initialized GSM module sends commands to home devices.
7) Trigger switches modules if no command is passed.
8) It sends either status update or user Acknowledgement.

Activity Diagram



State chart View

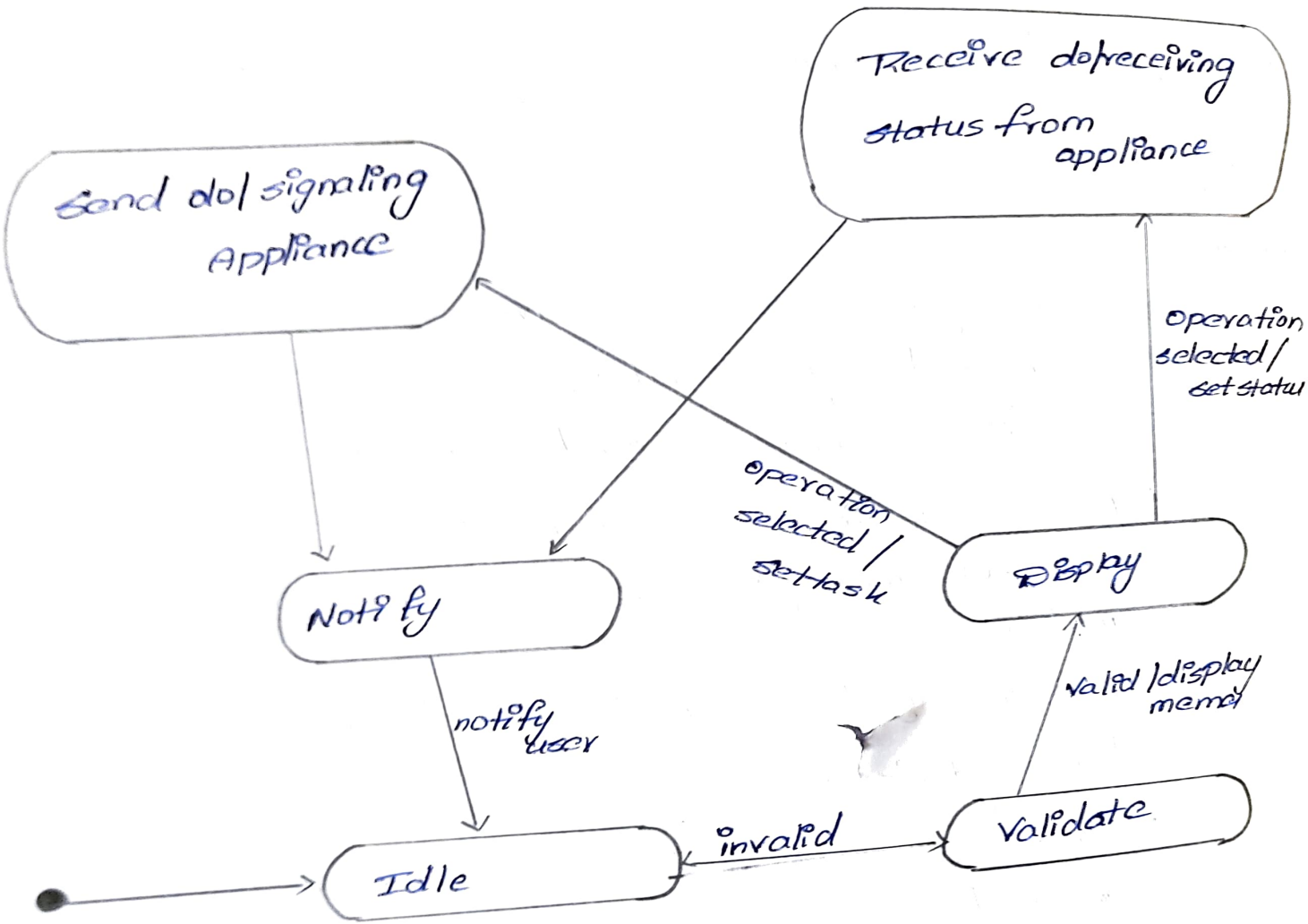
* It defines different states of objects during the its lifetime are charged by events.

States are

Idle
Valid
display
Recieve do/receiving status from appliance
Send do/ signelling appliance
Notify.

-> If the given command is valid do the command & display many.
-> If it is invalid, validate the command notify user.
->While displaying the operation selected the states of the appliance is received.
-> It is notified to user.
-> The appliances working is displayed to show the user.
->User can change the working power of appliances based on need.

Statechart Diagram



Deployment View

* It represents the deployment view of the system.
* It consists of nodes which are used to display the appliance options.

Nodes are

User

Appliance Control System

Result: Thus all the values of home appliance control system are modelled.

Deployment Diagram