

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 moduled.



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. In case 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 does 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 highest bid for an item before the close of the auction is declared as the owner of the item. If the auctioner or the purchaser did not want to bid for the product there 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, usecases & 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

class Diagram View

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

S.N ^o	className	Attributes	Operations
1	register	name, age, email, phoneno.	create register()
2	goods	seller, goodname	
3	account		creat account(), confirm account(), update account, delete account()
4	user	name, email	login(), logout()
5	ongoing auctiongood	actual price, sellerid, date of sale	
6	auctiongoods	auction date, auction price	auction()
7	Payment	amount	pay()
8	bid	buyerid, buyerprice	bidding(), check bid()

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 logsins through the admin provided registration.
- 6) Buyer logsins 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.

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 activities 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.

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 logs in to the server through admin.
- 4) Seller logs in 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 notifies for a bid to seller.
- 10) Admin notifies for a bid to buyer.
- 11) Seller starts auction.
- 12) Buyer starts auction.
- 13) Seller says final price
- 14) Buyer makes a payment.

Component View

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

Components are

Certification

Registration of Good

Seller

Negotiation

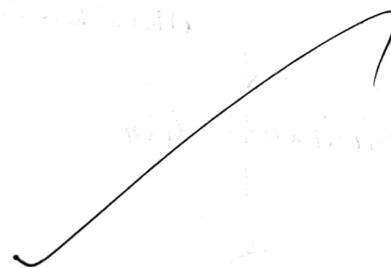
Management History, auction

Purchaser

Method of Bidding.

Relationship

Dependancy.



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

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.



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

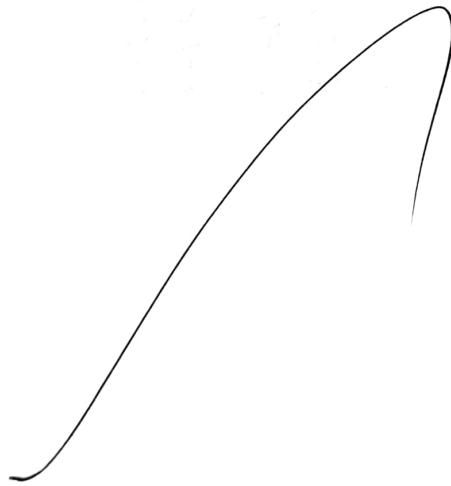
Indicate Lift position

Emergency

open/close door

Actors are

User



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	move up(), move down()
7	Cell button control	direction, audible tone	add request(), remove request()

Sequence View

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

Object are

Elevator

button

button control

hall button

call button

hall button control

call button control

door

message are

button press

illuminate

cancel illuminate

Hall call

fan on

direction

Floor

open

close.

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

Messager are

1: button press by user

1.1: illuminate

1.1.1: 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

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.

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 elevator moves to the next step up/down.

→ Elevator moves down if it is above the current floor.

→ Elevator moves up if it is below the current floor.

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 panel

outdoor panel

control system.

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;



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 appliance 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 appliance and to allow for communication between the different devices in a system, and for coordination among the various process running on such device. The system 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 if 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 appliance controller

Operate microwave

Operate sprinkler

Operate pet feeder

Logout.

Actors are

User

Relationships are

Association

class diagram view

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

SNO	className	Attributes	Operations
1	USER	name, email	login()
2	Useraccount	attribute	login()
3	admin	attribute	delete user(), add user(), delete appliance(), add appliance()
4	Home automation system	status boolean	login(), logout(), update(), select appliance().
5	Refrigerator	on, off	check status(), alert sensor(), set temperature(), operations()
6	air conditioner	on, off	set temperature(), check status(), get channel()
7	TV	on, off, channel, volume	get channel()
8	Security system	on, off	check status(), alert status()
9	Sensor	on, off, type	operations()

Sequence View

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

User, HACS, TV, microwave, refrigerator

Messages are

- 1) User logs 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 status of microwave.
- 9) microwave returns the status to HACS.
- 10) HACS display status of microwave to user.
- 11) User enter value of microwave to HACS
- 12) HACS displays updated value to microwave
- 13) microwave displays updated value to user.
- 14) User selects refrigerator.
- 15) HACS check status of refrigerator.
- 16) Refrigerator returns status to HACS.
- 17) HACS displays status to user
- 18) HACS set value for refrigerator.
- 19) HACS display value of refrigerator to user.

Collaboration View

* It shows how various objects interact with each other.

Objects are

TV, HACS, microwave, user, refrigerator

Messages are

- 1: login
- 1.1: verified
- 2: select TV
- 3: check status
- 3.1: return updated status
 - 3.1.1: display status
 - 3.1.2: check status
- 4.1: return status
- 4.1.1: display status
 - 4.1.1.1: return values
 - 4.1.1.2: set values
- 4.6.2.1: display updated values
- 5: select microwave
- 6: check status
- 6.1: return status
- 6.1.1: display status
- 6.1.2: display values
- 7: select refrigerator
- 8: set values

Component View

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

Components are

System Admin

HACS user

HACS appliance controller

microwave

sprinkler

pet feeder

Database

Relationships

Dependency,

Activity View:

It represents flow from one activity to another activity.

Activities are

- 1) User logs 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 device.
- 7) Trigger switcher module if no command is passed.
- 8) It sends either status update or user acknowledgement.

State chart view

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

States are

Idle

Valid

Display

Receive do/receiving status from appliance

Send do/signaling 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 status of the appliance is received.
- It is notified to user.
- The appliance working is displayed to show the user.
- User can change the working power of appliances based on need.

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.

