

Usecase view:

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

Use cases 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.

No	className	Attributes	Operations
1	OTR system	ID, domain	browse(), response()
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	Packaged(), 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.

Sequence Diagram View

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

Objects are

Passenger

Admin

Tickets

Messager are

- 1) login()
- 2) verify()
- 3) Return status()
- 4) Request Form()
- 5) Create Form()
- 6) Submit details()
- 7) Create Ticket()
- 8) Send Details()
- 9) Ticket created()
- 10) Receive Ticket()
- 11) Print Ticket()

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.

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 website & select the required website from ticket booking

Registration - customer provides his details for registration into website.

Verification - Database verifies the details.

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

Check - Customer checks the availability of train, class.

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

Submit - Customer submits registration form.

Payment - Customer pays the fare details.

Issue - Ticket is issued to the customer.

Idle - It is ready to perform another operation.

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.

Component View

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

Components are

Train

Ticket

Passenger

Form

Relationships are

Association

Dependency

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 needs of the online Ticket Reservation system have been modelled and this is mapped to following Rio's and cons.

Aim To model all the values of point of sale 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 service 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.

User 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 .

Class Diagram View :

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

S.No	ClassName	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(), bill generation(), 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(), add rev()
7	Scanner class	no. of items	add amt(), add discount()
8	Shopping cart class	itemlist: array	barcode scanner()
9	Sales person class	name, id	-
10	Payment class	amount, cal. amount, get balance, add test	
11	Item class	name, item, weight, manufacturer	
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.

Sequence Diagram View

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

Objects are

Buyer

cart

Employee

System

Admin

Message 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

Collaboration View

It shows how various objects interact with each other

Objects are

shopowner

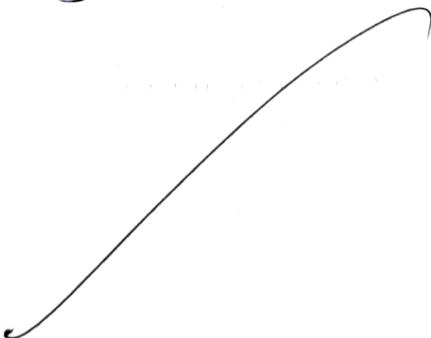
Supplier

Goods

Customer

Messages are

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



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.

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.

State chart View

* It defines 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.

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 values of point of sale system (POS) are modelled.



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

Problem statement

In the software industry the recruitment Procedure is the basic thing that go 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 Technical skills 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, technical 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 benefit 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

Class Diagram View

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

SNo	ClassName	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()

Sequence View

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

Objects are

Applicant

Test

Recruiter

Database

Messages are

- 1) Attend 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.

Statechart 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.

Activity View

It represents the flow from one activity to another activity

Activities are

- 1) Applicant logs 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.

Component View:

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

Components are

Recruitment

Register

Interview

Selection

Relationships are

Association

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.

