

1) Buy a camera from Amazon.com

Objects and Behavior:

### **Customer**

Data: Name, Phone\_Number, Email\_Address, Mailing\_address, User\_name, Password, billing\_address

Behavior: Browse, Login, Sign\_up, Select\_camera, Add\_to\_cart, Checkout, pay\_the\_bill, Read\_reviews, Compare\_items, Continue\_shopping,

### **Order**

Data: Order\_name, Order\_Number, Reviews, Cost

Behavior: Review\_order, Return\_order, Replace\_order, Confirm\_Order, Cancel\_order.

### **Card**

Data: Card\_name, Card\_number, CVV, Verification\_code.

Behavior: Get\_card\_details, Review\_card\_details, Send\_Verification\_code

### **Sequence of flow – Invoke objects with behavior**

Customer.browse\_amazon.com()

Customer.Select\_camera()

Customer.Read\_reviews(order)

Customer.Compare\_items()

Customer.Add\_to\_cart(Order)

If Account\_already\_exists:

Boolean Authenticated = Customer.login\_to\_account(User\_Name, Password)

If Authenticated:

Goto Continue\_Shopping

Else:

Display "Try Again"

Else:

Customer.Signup(Email\_Address, Phone\_Number, Address, Password)

End

End

Boolean Continue\_shopping = Customer.Continue\_Shopping

If Continue\_Shopping:

Go\_to\_mainpage

Else:

Customer.checkout(Order)

Order.Review\_order(Order)

Boolean Confirm\_order = Order.Confirm\_order(Order)

Boolean Pay\_later = Customer.Pay\_later(Order)

Boolean Cancel\_Order = Customer.Cancel\_Order(Order)

If Confirm\_Order:

Card.get\_card\_details(Card\_number, CVV, Billing\_Address, Expirydate)

Card.Review\_card\_details()

Card.Send\_Verification\_Code()

Customer.Pay\_the\_bill()

End

If Pay\_later:

Go\_to\_mainpage

End

If Cancel\_Order:

Order.Cancel\_Order(Order)

End

2) Design a platform for buying tickets of local events.

Objects and Behavior:

**Customer:**

Data: Name, Phone\_Number, Email

Behavior: Browse, Select\_event, Select\_no\_of\_tickets, Checkout, pay\_the\_bill, Read\_reviews, Select\_Category, Select\_date, Select\_time, Select\_ticket\_class, Cancel\_tickets

**Card**

Data: Card\_name, Card\_number, CVV, Verification\_code.

Behavior: Get\_card\_details, Review\_card\_details, Send\_Verification\_code

**Event**

Data: Name, Available\_Timings, Available\_Date, Location, Category

Behavior: Review\_Event, Print\_Tickets, Send\_confirmationmail, Refund\_Ticket\_Amount

**Ticket**

Data: No\_of\_tickets, Cost\_of\_ticket, Class, Final\_Cost, Refund\_cost

Behavior: Refund\_If\_Cancelled, Cancel, Buy

**Sequence of flow – Invoke objects with behavior**

Customer.Browse()

Customer.Select\_Category()

Customer.Select\_event()

Customer.Read\_reviews()

Event Event\_Time = Customer. Select\_Time()

Event Event\_Date = Customer.Select\_Date()

If Event\_time != Event.Available\_Timings OR Event\_Date != Event.AvailableDate

Display "Tickets not available"

Else

Customer.Select\_ticket\_class()

Ticket.no\_of\_tickets = Customer.Select\_no\_of\_tickets()

End

If Ticket.no\_of\_tickets >= 1

Customer.Checkout()

Ticket.Final\_Cost = Ticket.no\_of\_tickets \* Ticket.Cost\_of\_ticket

Else

Display "Select the tickets"

End

If Ticket.Final\_Cost > 0.00

Card.get\_card\_details(Card\_number, CVV, Billing\_Address, Expirydate)

Card.Review\_card\_details()

Card.Send\_Verification\_Code()

Customer.Pay\_the\_bill()

Ticket.Print\_tickets()

Event.Send\_Confirmationmail(email\_id) ##

Else

Ticket.Print\_tickets()

*//for free tickets//*

Event.Send\_Confirmationmail(email\_id)

Boolean Cancel\_tickets = Customer.Cancel\_tickets(Ticket)

Boolean Refund = Event.Refund\_If\_Cancelled()

If Cancel\_Tickets:

If Refund:

Ticket.Cancel\_ticket(ticket)

Event.Refund\_Ticket\_Amount(Refund\_Cost)

Else:

Ticket.Cancel\_Ticket(ticket)

End

End

3) Design an app to book a doctor's appointment using your medical insurance provider.

Objects and Behavior:

### **Patient**

Data: Name, Phone\_Number, Email\_id, User\_Number, Password, Insurance\_ID, Appointment\_Date, Appointment\_Time, Emergency

Behavior: Select\_Doctor, Select\_date, Select\_time, Select\_Health\_Center, Sign\_up

### **Health\_center**

Data: Name, Location, Available\_time, Available\_Date, Doctor, Doctor's\_experience, Doctor's specialization,

Behavior: Send\_confirmationmail, Display\_Doctor\_details

### **App**

Data: Insurance\_plan, Amount\_left

Behavior: Live\_Chat, Secure\_Inbox, Customer\_Service, Order\_Prescription, Book\_appointment.Send\_ReminderMail, Send\_ReminderVoicemail

### **Sequence of flow – Invoke objects with behavior**

Patient.Sign\_up(Name, Phone\_number, Email\_id, Password, Insurance\_ID)

If Patient.Emergency:

Display "Please call 911"

Else

Patient.Select\_Health\_Center

Health\_Center.Display\_Doctor\_details(Doctor's\_name, Doctor's\_experience, Doctor's\_Specialization())

Patient.Select\_Doctor()

Appointment\_Date = Patient.Select\_date()

Appointment\_Time = Patient.Select\_time()

Patient.Select\_HealthCenter()

If Appointment\_Date != Healthcenter.Available\_Date OR Appointment\_Time != Healthcenter.Available\_Time

Display "Appointment not available"

Else:

App.Book\_appointment()

Health\_center.Send\_ConfirmationMail()

If Appointment\_Date - Current\_date == 1

App.Send\_ReminderMail()

App.Send\_ReminderVoicemail()

#### 4) Design a Job Searching Platform

Objects and Behavior:

##### **Applicant**

Data: Name, Email, Phone\_number, Resume, Password

Behavior: Sign\_up, Login, Apply\_For\_Job, Browse\_Jobs, Filter\_Jobs, RequestReferral, Select\_TypeOfRecruitment, Upload\_Resume, Enter\_Details()

##### **App**

Data: No\_of\_Jobs, Job\_Title, Job\_Location, Date\_posted, Job\_Description, Type\_Of\_Recruitment

Behavior: Send\_Payment\_confirmationMail, Send\_Notifications\_Of\_Jobs, Track\_Job\_Status, Set\_Job\_Interview  
Display\_Premium\_Benifts, Send\_Resume\_To\_Recruiters, Apply\_To\_Job

##### **Card**

Data: Card\_name, Card\_number, CVV, Verification\_code.

Behavior: Get\_card\_details, Review\_card\_details, Send\_Verification\_code

#### **Sequence of flow – Invoke objects with behavior**

If Account\_already\_exists:

Boolean Authenticated = Applicant.Login(User\_Name, Password)

Else:

Applicant.Sign\_up(Name, Phone\_number, Email, User\_Name, Password)

End

Applicant.Filter\_Jobs() // Engineering, Marketing, Analyst ... Location, Position

Applicant.Browse\_Jobs()

Applicant.Apply\_For\_Job()

Applicant.Request\_Referral()

Applicant.UploadResume()

Applicant.Enter\_Details()

App.Display\_Premium\_Benifts() // faster response

App.Type\_Of\_Recruitment = Applicant.Select\_TypeOfRecruitment()

If App.Type\_Of\_Recruitment == Premium

Card.get\_card\_details(Card\_number, CVV, Billing\_Address, Expirydate)

Card.Review\_card\_details()

Card.Send\_Verification\_Code()

Applicant.Pay\_the\_bill()

App.Send\_Payment\_confirmationMail()

App.Send\_Resume\_To\_Recruiters()

```
        Apo.Track_Job_Status()
        Job.Set_Job_Interview
Else:
    App.Send_Resume_To_Recruiters()

App.Apply_To_Job()
```

## 5) Order Pizza from Dominos

Objects and Behavior:

### **Customer:**

Data: Name, address, Phone\_Number, Email

Behavior: Sign\_Up\_For\_Dominos, Select\_Shopping\_Option, Pay\_the\_bill,  
Select\_Pizza, Select\_pizza\_size, Select\_store, Select\_Delivery\_Time,  
Select\_Delivery\_Location, Select\_Toppings, Select\_Pizzabase

### **App:**

Data: Delivery, Carryout, Stores\_nearby, Payment\_method, Customized\_Pizza

Behavior: Send\_Confirmation\_Mail, Display\_Shopping\_Options, Display\_Pizzas  
Get\_Customer\_Details, Display\_Store\_Nearby

### **Card**

Data: Card\_name, Card\_number, CVV, Verification\_code.

Behavior: Get\_Card\_details, Review\_card\_details, Send\_Verification\_code

### **Sequence of flow – Invoke objects with behavior**

Customer. Sign\_Up\_For\_Dominos(Name, Phone\_Number, Email)

App.Display\_Pizzas()

If Customized\_Pizza()

Customer.Select\_Pizzabase()

Customer.Select\_Toppings()

Customer.Select\_Pizza\_Size()

Else:

Customer.Select\_pizza()

Customer.Select\_pizza\_size()

App.Display\_Shopping\_Options()

Shopping\_Method = Customer.Select\_Shopping\_Option()

If Shopping\_Method == Delivery

App.Get\_Customer\_Details(Address, Phone\_Number)

App.Display\_Store\_Nearby()

Customer.Select\_Store ()

Customer.Select\_Delivery\_Time()

Customer.Select\_Delivery\_location()

End

If Shopping\_Method == Carryout

App.Get\_Customer\_Details(Address, Phone\_Number)

App.Display\_Store\_Nearby()

Customer.Select\_Store ()

End



Customer.Checkout()  
Card.Get\_Card\_details(Card\_number, CVV, Billing\_Address, Expirydate)  
Card.Review\_card\_details()  
Card.Send\_Verification\_Code()  
Customer.Pay\_the\_bill()  
App.Payment\_confirmationMail()