1. **Design a Course registration Platform:**

Objects and Behaviour

Computer:

State: Processorname, ModelNo

Behaviour: isWorking(), restart()

Internet:

State: IPAdress, InternetType

Behaviour: Connect(), isAvailable() , retry()

Website:

State: URL

Behaviour: login(), logout()

Student:

State: name, age, studentId, password

Behaviour: list browseCourses(), registerCourse(), resheduleCourses(), dropCourse()

Course:

State: courseName, CourseDuration, StartDate, EndDate,

Behavior: register(),drop()

Account:

State: tutionFee, balance

Behaviour: pay() , confirm(), int getBalance()

***Pseudo code to register and drop a course:***

1. Student register for course:

Course Java= new Course; //object creation

Student Amisha= new Student()

Amisha.registerCourse(Course Java) //Amisha register for java course

if-->Computer:Windows10.connect()

else retry()

if(Internet.isAvailable(URL))

else retry() // check for internet availibility

Website.login(StudentID, password)

list browseCourse()

course.register(java); //register for java course

duebalance = Account.getBalance()

If duebalance>0 then tutionfee=duebalance+tutionfee; // check for due tutionfee

Account.pay(tutionfee)--->confirm()

website.logout() // paid and logged out

else:

retry()

b) Pseudo code to Student drop a course:

Student John;

Course Database;

John.dropCourse(Database) //John drops course

Computer.connect()

if(Internet.isAvailable())

Website.login(ID, Pwd)

list browseCourse()

course.drop(CourseNAme); // browse course and drop

else:

retry();

**2.) Order Food in a food Delivery App (Like Uber Eats)**

Objects and Behaviour:

Mobile:

State: MobileModel, MobileNo

Behavior: boolean isAndroid(name)

App:

State: ID, Password

Behaviour: signIn(), signUp()

Customer:

State: Custname, CustID, CustMobileNo, CustAdress

Behavior: menu checkMenu(), addToCart(), pay(), reviewFood()

Restaurant:

State: Rname, RCode, RContactNumber, RAdress

Behaviour: listFood(), acceptOrder(), confirmOrder()

Payment:

State: CardNo, BankName

Behaviour: transaction(), Confirm()

Delivery:

State: DeliveryTime

Behavior: assignDeliveryPerson(), checkForDelays()

* ***Pseudo code to order food from food app :***

Customer Amisha, Restaurant BurgerKing

if-->Mobile M1.isAndroid(pixel3) else Print--App needs Android phone

Amisha-->App.signin(ID,Password)

If true:menu Burgerking.checkMenu()

BurgerKing.addtoCart(vegBurger)--BurgerKing.acceptOrder() // Restaurant accepts order

Amisha-->calls Payment DebitCard(CardNo, BankName)

transaction(Amount)-->confirm Payment

Delivery.assignDeliveryPerson(XYZ,CustAddres) //XYZ assigned to deliver veg burger and

address for delivery

if Deliver.checkforDelays(DeliveryTime)

inform Customer(CustPhoneNumber)

else order Complete

**3.) Design a Platform for buying tickets of Local events:**

Objects and Behaviour:

Computer:

State: Processorname, ModelNo

Behaviour: isWorking(), restart()

Internet:

State: IPAdress, InternetType

Behaviour: Connect(), isAvailable() , retry()

Event:

State: Name, Time, Address,Ticket,Venue, EventFee

Behavior: perform()

Website:

State: URL

Behavior: events showEvents()

Buyer:

State: Name, PhoneNp, ID, Password

Behavior: bookEvent(), login()

Payment:

State: CardNo, BankName

Behavior: pay(), confirm()

EventReview:

State: Reviewcomment

Behavior: showReview(event)

***- Pseudo code:***

Event Concert: Buyer Nobita : Website events.com

if Computer Windows10.isWorking()

else restart()

Internet( connect to URL)

else retry()

Nobita.login(id,password) // logged in successfully

events.com.showEvents() // List of available events

If event of interest found:

Nobita.bookEvent(Concert)

if Eventreview showReview(concert) is good

Payment.pay(EventFee)

venue=Event.venue(concert) // venue for Event sent to Buyer

confirm payment

logout()

**4.Buy a computer from Amazon:**

Objects and Behaviour:

Internet:

State: IPAdress, InternetType

Behaviour: Connect(), isAvailable() , retry()

AmazonApp:

Behavior:boolean login(), logout()

Computer:

State: Processorname, ModelNo

Buyer:

State: Name, PhoneNp, ID, Password

Behavior: buy(ModelNo), list checkavailableComputers()

Payment:

State: CardNo, BankName

Behavior: pay(), confirm()

ComputerReview:

Behavior: reveiw showReview(ModelNo)

***- Pseudo Code to buy a Dell computer from AmazonApp :***

Buyer John -->Internet.connect() //connecttion to internet

If connect succesfully

If true: AmazonApp.login(Id, password) //login to App if ID and password are correct

list John checkAvailableComputers() //list of available computers

reveiw ComputerReview showReveiw(ModelNo) // check reveiws for specific ModelNo

If reveiw=good

John.buy(Modelno Dell) : payment pay(CardNo)

confirm() //payment done

AmazonApp.logout()

**5. Design a app for booking hotels**

Objects and Behaviors:

Mobile:

State: MobileModel, MobileNo

Behavior: boolean isAndroid(name)

App:

State: ID, Password

Behaviour: signIn(), signUp()

Hotel:

State: Name, Address, HotelID,

Behavior: listPrices(), listVacantRooms()

Customer:

State: Name, Age, ID

Behavior: bookHotel(), checkRatings(), cancelHotel(), checkUtilities

Room:

State: RoomNo, RoomCapacity, NoOfBeds, Price

Behavior: isAvailable(),

Transaction:

State: CardNo, BankName

Behavior: pay(), confirm(), retry()

Utility:

State: UtilityName, UtilityCategory

Behavior: reserve(), release()

RateHotel:

State: comments

Behavior: rateRoom(HotelName)

***-Pseudo code to book a hotel room:***

Customer Amisha, Hotel Marriott

if-->Mobile M1.isAndroid(pixel3) else Print--App needs Android phone

Amisha-->bookHotel()-->App.signin(ID,Password)

list Hotel.listprices() // check prices of hotel list

If true:Room.isAvailable()-->Rate( Marriott) //check for available rooms and

its ratings

if ratings=good

Amisha.bookHotel(Marriot) //book hotel

Transaction.pay(Amisha cardNo,Room Price)

confirm()

Amisha-->Utilities.reserve(UtilityName gym) //Book hotel utility while booking

App.signout()