Github:

1) Allow users to create an account or sign in

2) Allow for the creation of Repositories

3) Allow users to import files into the repositories

4) Allow collaborators to be added to the repositories

5) Allow creating branches from the main branch to work on new changes or features

6) Allow commits

7) Allow pull requests to initiate discussion on the commits

8) Allow conversation to discuss feedback and review on the commits

9) Allow deployment of code once the branch passes the testing

10) Allow for merging of the code with master branch once deployed.

Objects:

        1) Github

                    Data: code, files, repositories, collaborators, users, branches, commits

        2) User

                    Data: username, password

                    Behaviour: create account, sign in, create repositories, add collaborators, make commits, initiate pull requests, create a branch,  deploy code, merge branch

        3) Repository:

                    Data: files, commits

        4) Files

                    Data: Content

        5) Mainbranch

                    Data: code

                    Behaviour: Get deployed into production

        6) Branch

                    Data: Code

                    Behaviour: Merge with main branch, pulled back, get modified, pushed back

DesignGitHub:

    Github github

    User user

    Repository repo

    Files file

    Mainbranch mBranch

    Branch branch

    github.AuthorizeUser(username, password)

    If Authorize.True Then

        If user.InitiatesCreateRepositoryAction

                github.CreateRepository(repositoryName)

        End If

        If user.InitiatesImportFileAction Then

                    repo.ImportFile(file,source,repo,repositoryName)

        End If

         If user.InitiatesAddColaboratorsAction Then

                   colCount = collaborator.count

                    For(i=1,i<colCount,i++)

                            repo.AddCollaborators(collaborator(i),repositoryName)

                    Next

          End If

        If user.initiatesCreateBranchAction Then

            github.CreateBranch

        End If

        If user.initiatesCommitAction Then

            branch.Commit(file)

        End If

        If user.initiatesPullRequest Then

            branch.RequestPullForBranch

            If branch.Feedback.Exists Then

                user.ReceiveFeedbackAndReviews

            End If

        End If

        If user.initiatesTestAction Then

            result = branch.Test

        End If

        If result ==true Then

            branch.deployCode

        Else

            branch.ModifyCodeAndPushBack

        End If

        If user.initiatesMergeAction

            branch.Merge(mBranch)

        End If

Else

    user.CannotLogin

End If

1. Allow user to select products : Coke, Pepsi, Fanta
2. Find out amount of the product being selected
3. Display the status for each product (in stock, out of stock)
4. Display the total amount to be paid for the product selected
5. Accept coins of 1,5,10,25 cents
6. Find out how much was inserted
7. Calculate change to be returned, if any
8. Allow user to cancel the request
9. Refund the inserted coin
10. Return selected product and remaining change if any

Objects:

    1) VendingMachine

                                    Data: Soft drinks, display screen showing the products available, buttons to choose, outlet to collect products, stock of coins(to return as 'change' to customer)

                                    Behaviour: Return selected products, Return change

    2) VendingProduct

                                    Data: Soft Drink

    3) DisplayScreen

                                    Data: Product available, price for each product, status of each product

**Pseudo Code**

Customer cust

VendingMachine machine

VendingProduct product

DisplayScreen screen

drink = cust.selectProduct -> fanta, coke, pepsi : user chooses a drink

If status.drink != "out of stock" Then

        screen.displayStatus

        priceOfDrink = drink.Price

        screen.display (amount)

        cust.InsertCoin

        inputAmount = FindAmountFromCustomer

        screen.display("fetching drink")

        machine.RetrieveDrinkFromStorage(drink)

        machine.Return(drink)

        screen.display("collect drink")

        If inputAmount ! = priceOfDrink Then

            If inputAmount < priceOfDrink Then

                change  = machine.CalculateChange->(priceOfDrink - inputAmount)

                machine.ReturnBalanceAmount(change)

                screen.Display ("Collect change")

            ElseIf inputAmount>priceOfDrink Then

                machine.ReturnInputAmountFRomCustomer

                screen.Display ("Insert correct amount")

            End If

            End If

        End If

Else

     System.Out.Print ("Product out of stock. Try later or choose a different product")

End If

1) Assemble a planning team

2) Determine target audience and companies to invite

3) Select venue, date and time

4) Establish a budget

5) Communicating about the event to companies and audience

6) Marketing and promoting the event through fliers and posters and ads

7) Organize food and beverages

8) Make arrangements for setting up booths

9) Send Thank you notes for all the participating companies after the career fair

Objects:

            1) Organizer

                                Data: Information and details required for planning the event

                                Behaviour: Execute the planned activities

            2) Audience

                                Behaviour: Attend the career fair

            3) Companies

                                Data: Open positions available

                                Behaviour: Interview the attendees, recruit suitable participants

            4) Venue

                               Data: Name, area

            5) MarketingItems:

                                Data: Details about the career fair

            6) Caterers:

                                Data: Food, Beverages

            7) OrganizationHostingTheCareerFair

                                Data: Name, address

                                Behaviour : Plan and host the career fair

**Pseudo Code**

OrganizeCareerFair

                    Organizer Arch

                    Audience audience

                    Companies companies

                    Venue venueForEvent

                    MarketingItems fliers

                    Caterers caterer

                    Organization NEU

                    Arch.DetermineTargetAudienceAndParticipatingCompanies -> List of companies and audience decided

                    companies = list of companies

                    audience = list of audience

                    Arch.EstablishBudget -> approximate cost of (venue rent, marketing(fliers, posters etc), printing, food & beverages, parking fee) : Budget amount decided

                    Arch.SearchForVenue -> areas close to the organization hosting the fair : list of options

                    venueOptions = list of options

                    count = venueOptions.Count

                    For(i=0, i<=count, i++)

                           location = getAddress.venueOptions(i)

                           If location = NEU.area                    // Trying to add a condition to check if the location in current iteration is atleast close to the venue for career fair; not sure how to show that in

                                                                                      pseudo code. Hence used "="  //

                                If location.isAvailable(date, time)

                                        venueForEvent = location

                                        break

                                End If

                           End If

                    Next

                    Arch.CommunicateWithParticipantsAndAttendees -> companies, audience,venueForEvent, date, time emails : details about career fair communicated

                    Arch.MarketAndPromoteTheEvent -> Fliers, posters, newspaper ads, facebook posts : Career fair advertised through various channels

                    Arch.SearchForCaterer-> eateries, coffee shops, restaurants,venueForEvent: list of places

                    catererOptions = list of places

                    count = catererOptions.Count

                    For(i=0, i<=count, i++)

                           place = getAddress.catererOptions(i)

                           If place = venueForEvent.area           // Trying to add a condition to check if the location in current iteration is atleast close to the venue for career fair; not sure how to show that i

                                                                                        pseudo code. Hence used "="  //

                                If place.isAvailable(date, time)

                                    caterer = place

                                    break

                                 End If

                            End If

                    Next

                    Arch.ProvideMenuToCaterer -> food, beverages, caterer : menu decided

                    caterer.prepareMenuForCareerFair

                    Arch.SetUpBoothsForCareerFair -> labour, material, chairs, tables : booths set up and ready

                    Arch.SendOutThankYouNotes -> companies, mail Ids, ThankYouEmails : Thank you notes sent out to participated companies after the fair.

------------------------------------------------------------

   1) Browse for the Pizza Company's website

2) Choose order online option

3) Select the nearest location

4) Select the delivery date and time

5) Choose from the menu options

6) Pay for the order

Things:

            1) Internet

                            Data: Collection of websites

                            Behaviour: Search for electronic websites

            2) Customer

                            Data: Name, phone/laptop/tab, credit card, address, city

                            Behaviour: Login,Select, search, enter information

            3) PizzaDeliveryWebsite

                            Data: URL, Menu, Bank Account

                            Behaviour: Select, search, take in user input, place the order,authorizer login details

            4) Pizza

                            Data: Price, Veg/Non-Veg, toppings, crust, sauce, cheese

            5) CreditCard

                            Data: Name, card number, expiry date, security code

            6) CreditCardCompany:

                            Behaviour: Authorize the transaction

**PseudoCode**

OrderPizzaOnline:

        Internet internet

        PizzaDeliveryWebsite pizzaHut

        Customer cust

        Pizza myPizza

        Creditcard creditcard

        CreditCardCompany visa

        If internet.isAvailable

                    cust.SearchForWebsite -> internet, pizza hut : link for pizza hut

                    cust.SelectPizzaHutLink -> link for pizza hut: pizza hut page opened

                    cust.LoginToPizzaHutWebsite -> username, password : logged in

                    If (authorize is true)

                            cust.SelectOnlineOrderOption

                            cust.SelectLocation -> State, city : opens up nearby stores

                            i = 1

                            location = abcd

                            While (location is NotEmpty)

                                    location = pizzaHut(i).FetchStoreLocation

                                    If location = city

                                            cust.SelectTheNearestStore -> location : page opens

                                            break

                                    Else

                                        i = i+1

                                    End If

                            Loop

                            cust.EnterDeliveryDetails -> date, time, address : input received

                            cust.SelectPizzaBasedOnCustomer'sChoice -> price, veg, thin crust, cheese : pizza

                            myPizza = pizza

                            pizzaHut.OrderThePizza ->myPizza, creditcard,visa : pizza ordered

                    Else

                            cust.CannotMakeCalls

                    End If

        Else

                  cust.BrowseLater

        End If

-----------------------------------------------------------------

Rent a car from reachnow

Fill gas at a gas station

Stop at Starbucks for breakfast

Use maps for navigation

Buy ticket to go to the park

Park car in the parking lot for Rainier visitors

Things:

1) Internet:

                             Data: ReachNow, Google maps

                             Behaviour: Find and reserve a car using ReachNow, Navigate to Mt.Rainier using Google maps

2) ReachnowApp:

                            Data: List of cars, details about cars, addresses for car pickup

                            Behaviour: Search for cars, book a car

3) GoogleMaps:

                            Data: Directions to reach Mt.Rainier

                            Behaviour: Navigate to Mt.Rainier

4) Hiker:

                            Data: Name, phone, credit card

                            Behaviour: Search for cars, book a car, drive, fill gas, buy breakfast, buy tickets

5) GasStation:

                            Data: Gas, Gas pipes used for filling tanks

                            Behaviour: Fill car tank with required quantity of gas

6) Starbucks:

                            Data: Food, Beverages, card swiping machine

                            Behaviour: Prepare meal/drink, collect money for the purchase

7) CreditCard:

                            Data: Number, name, company, expiry, security code

8) CreditCardCompany:

                            Behaviour: Authorize transaction

9) RainierTicketCounter:

                            Data: Tickets/passes

                            Behaviour: Issue tickets for visitors entering the Mt.Rainier National Park

**Pseudo Code**

HikingToRainier:

            Internet internet

            Hiker hiker

            ReachnowApp reachnow

            GoogleMaps maps

            GasStation chevron

            Starbucks starbucks

            CreditCard creditcard

            CreditCardCompany visa

            RainierTicketCounter counter

            If internet.isAvailable(

                        hiker.FindCarInReachNow -> Open app, type of car, price : List of cars

                        reachnow.ReserveCar -> List of cars: myCar

                        If myCar is not Empty

                                  hiker.PickUpmyCarAndDrive

                        hiker.FillGasAtGasStation -> Quantity of Gas, creditcard : Tank filled with required quantity of gas

                        hiker.GetBreakfast -> food, drink, creditcard : ordered breakfast

                        starbucks.ProcessTheOrder -> prepare the ordered items : Breakfast purchased

                        hiker.FindRouteInGoogleMaps -> Open App, enter destination : List of routes

                        hiker.SelectBestRoute -> List of routes : route\_1

                        myRoute = route\_1

                        maps.NavigateToDestination -> myRoute : Navigation started

                        If counter isOpen

                            hiker.BuyTicketsToEnterRainier -> Number of persons, creditcard : Tickets purchased

                            hiker.ContinueDrivingToParkingArea -> myRoute: Reaches parking area

                            hiker.ParkCarAndStartHiking

                        Else

                            hiker.CannotProceedToRainier

                        End If

                        Else

                                hiker.NoCarToDrive

                        End If

             Else

                        BrowseInternetLater

             End If