**Assignment1**

Name: Jiacheng Niu NUID: 001810427

# Hiking

1.1. Rent a ReachNow to drive to mountain rainier on weekends with friends

1.2. On the way, fill the gas

1.3. On the way, eat at restaurant for lunch

1.4. Buy the ticket to get into the park

1.5. Book a hotel or check in the hotel.

**Things:**

* **Hiker:**

Data: Name, Address, Phone, Money

Behaviors: Rent, Drive, Fill in the Gas, Order, Eat, Buy, Play, Book

* **Car\_Rental\_Company:**

Data: Name, Collection of Cars

Behaviors: Loan, Charge

* **Car:**

Data: Type, Price, HasGas

Behaviors: Run

* **Gas\_Station:**

Data: Address, Collection of Gases

Behaviors: Sell

* **Gas:**

Data: Price, Type

Behaviors: Combust

* **Restaurant:**

Data: Name, Address, Collection of Foods, Seat

Behaviors: Sell, Cook, Offer seats

* **Food:**

Data: Name, Price, Volume, Taste

Behavior:

* **Park:**

Data: Address, Price

Behavior: Sell Ticket, Check Ticket

* **Hotel:**

Data: Collection of Rooms, Address, Name, Phone, HasRoom

Behavior: Reserve Rooms, Change, Clean Rooms

* **Room:**

Data: Area, Price, Decoration

Behavior:

Hiker Jack;

Car\_Rental \_Company ReachNow\_Company;

Car ReachNow;

Gas\_Station Station;

Gas Gas92;

Restaurant WholeFood;

Park Park;

Hotel Hotel;

Bool BookCondition = true;

Jack. RentACar -> ReachNow, ReachNow\_Company: Car;

ReachNow = Car;

Jack. DriveToMountain -> ReachNow: ReachNow run;

If (ReachNow. HasGas != true)

Jack. FillInGas -> Gas92, Station, money: ReachNow. HasGas = true;

End

Jack. OrderFood –>WholeFood, money: Foods;

Jack.eat -> Foods;

Jack.BuyTicket –> money, Park: ticket;

Jack.PlayInThePark;

Jack.BookRoom -> Hotel, money: BookCondition

If Hotel.HasRoom == false

BookCondition = false;

Else

BookCondition = true;

End

# Organize a career fair (Suppose you are the organizer)

**Things:**

* **Organizer:**

Data: Money, Phone, Website

Behaviors: Rent, Confirm, Check, Contact, Advertise

* **Place:**

Data: Location, Area, SeatNumber

Behaviors:

* **Company:**

Data: Name, Address, Collection of Jobs, Collection of employers

Behaviors: interview, hire, SendEmployer, IsAttend

* **Employer:**

Data: Name, Position, Company, IsApprove

Behavior: Talk, Review

* **Work:**

Data: Position, Salary, Type, Company

Behavior:

* **JobSeeker:**

Data: RegisterStatus, Name, Phone, Skill, Major

Behavior: Register, Attend, SendResume, Talk

Organizer Organizer;

Place CareerFairPlace;

Company[] Companies = {Company1, Company2, …};

Employer Employer;

Work Programmer;

JobSeeker Seeker;

Organizer. RentPlace -> Money, Place: Place;

CareerFairPlace = Place;

For(int i = 0; i < Companies. Length; i ++)

Organizer. Contact -> Companies[i];

If Companies[i]. IsAttend == true

Companies[i]. SendEmployer -> Employers, Organizer, CareerFairPlace;

End

Organizer. Advertise -> Oranizer.Website, CareerFairPlace: Website

Seeker. Register -> Organizer.Website, Organizer: Seeker. RegisterStatus

If Seeker. RegisterStatus == true:

Seeker. AttendCareerFair -> CareerFairPlace, Organizer;

Seeker.SendResume -> Employer;

Employer. ReviewResume;

Seeker. Talk -> Employer;

If Employer. IsApprove == True:

Employer. OfferAJob ->Seeker;

End

End

# Order Pizza from Pizza Hut

**Things:**

* **PizzaHut:**

Data: Name, Address, Phone, Collection of pizzas

Behaviors: Sell, Cook, Package, deliver

* **Pizza:**

Data: Size, Taste, Price

Behaviors:

* **Consumer:**

Data: Name, Address, Phone, Money

Behaviors: Call, Order, Pay

* **Phone:**

Data: PhoneNumber, IsAvailable

Behaviors: Connect

PizzaHut PizzaHut;

Pizza LargePizza;

Consumer Jack;

If PizzaHut. Phone. IsAvailable == true AND Jack.Phone.IsAvailable == true:

Jack. CallPizzaHut -> PizzaHut.phone: Connected;

Jack. OrderPizza -> Jack.address, PizzaHut, LargePizza: Pizza Ordered

End

PizzaHut. Cook -> Materials: Pizza

LargePizza = Pizza

PizzaHut.deliver -> LargePizza, Jack. Address: PizzaDelivered

Jack.Pay -> Money, PizzaHut: JackGetPizza

# Design a code sharing platform (eg: Github).

**Things:**

* **Database:**

Data: Table, Collection of Codes, Collection of Codes

Behaviors: Insert, Delete, Update, Query

* **Website:**

Data: URL, Collection of Users, Collection of Projects, IsAvailable

Behaviors: Confirm, Display

* **Code:**

Data: Size, Language, Function

Behaviors:

* **Project:**

Data: Collection of Codes

Behaviors:

* **Manager:**

Data: Name

Behaviors: Build, Manage

* **User:**

Data: Name, Account, Collection of Projects

Behaviors: Register, Login, CreateProject, Upload, Delete, Share, Clone, Fork, Visit

* **Internet:**

Data: Group (Collection) of Websites, IsAvailable

Behaviors: search for electronic websites

Database Database;

Website GitHub;

Project Project;

Code Code;

Manager Manager;

User Jack, Peter;

Manager. BuildTheWebsite –> Internet: WebDatabase, Website

Database = WebDatabase, GitHub = Website;

Jack. Register -> GitHub: Jack Registered;

If Internet. IsAvailable == true AND GitHub. IsAvailable == true:

Jack. Login -> GitHub;

Jack. CreateProject -> GitHub: Project;

Jack. Upload -> GitHub, Project, Code: CodeUploaded;

Database. InsertData -> Code: CodeSaved;

Jack. Share -> Project, GitHub: ProjectShared;

Peter. Login -> GitHub;

Peter. Visit -> Jack. Project, Database;

Peter. Fork -> Jack. Project, Database: ProjectForked;

Peter. Clone -> Jack. Project, Database: ProjectCloned;

End

# Design a soft-drink/snacks vending machine

* **Machine:**

Data: Size, IsFull, IsEnough, Collection of goods

Behaviors: GetMoney, GiveChange, Calculate, Show, Output

* **Goods:**

Data: Name, Price, Number

Behaviors:

* **Consumer:**

Data: Money

Behaviors: Choose, Buy

* **Manager:**

Data: Name

Behaviors: AddGoods, AddMoney, CollectMoney

Machine Machine;

Goods Drink, Snack;

Consumer Jack;

If Machine. IsFull == false

Manager. AddGoods -> Machine: GoodsAdded;

Machine. IsFull = true

Machine. IsEnough = true

End

If Machine. IsEnough == true

Jack. ChooseGoods -> Machine: GoodsChoosed;

Machine. CalculateMoney: ShowMoney;

ShouldMoney = ShowMoney;

Jack. Pay -> Money, Machine;

PaidMoney = Money;

Machine. GetMoney -> Money, Machine;

If PaidMoney >= GoodsMoney

Machine. Output -> Goods: ConsumerGetGoods;

If PaidMoney > GoodsMoney:

Machine. GiveChange -> Money: ConsumerGetMoney;

End

End

Else:

Machine. ShowNotEnough: ConsumerKnow;

End