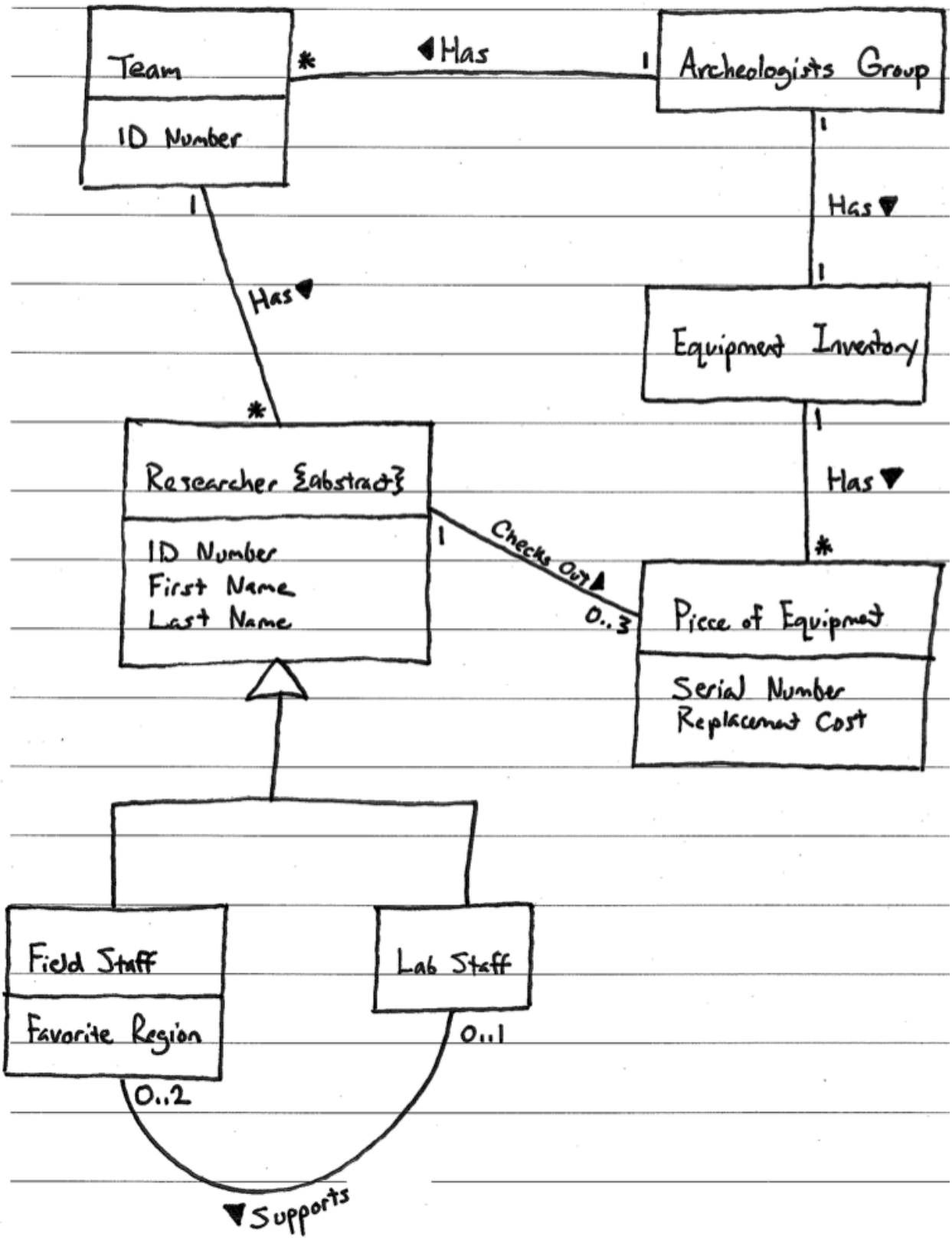
Problem: Create a domain model (using class diagram notation) based on the following description.

You have been asked to build a management system for a group of archeologists. The group is comprised of multiple teams of researchers. Each team has a letter ID (e.g., team A, team B). Each researcher belongs to one of the teams, and has an ID number, a first name, and a last name. There are two types of researchers: field and lab staff. Each field staff member has a favorite region (string). Each lab researcher supports up to 2 field researchers. Some researchers may not be supported by a lab researcher. The company also manages an inventory of equipment. Researchers of any type may check out up to 3 pieces of equipment. Each piece of equipment has a serial number and replacement cost.

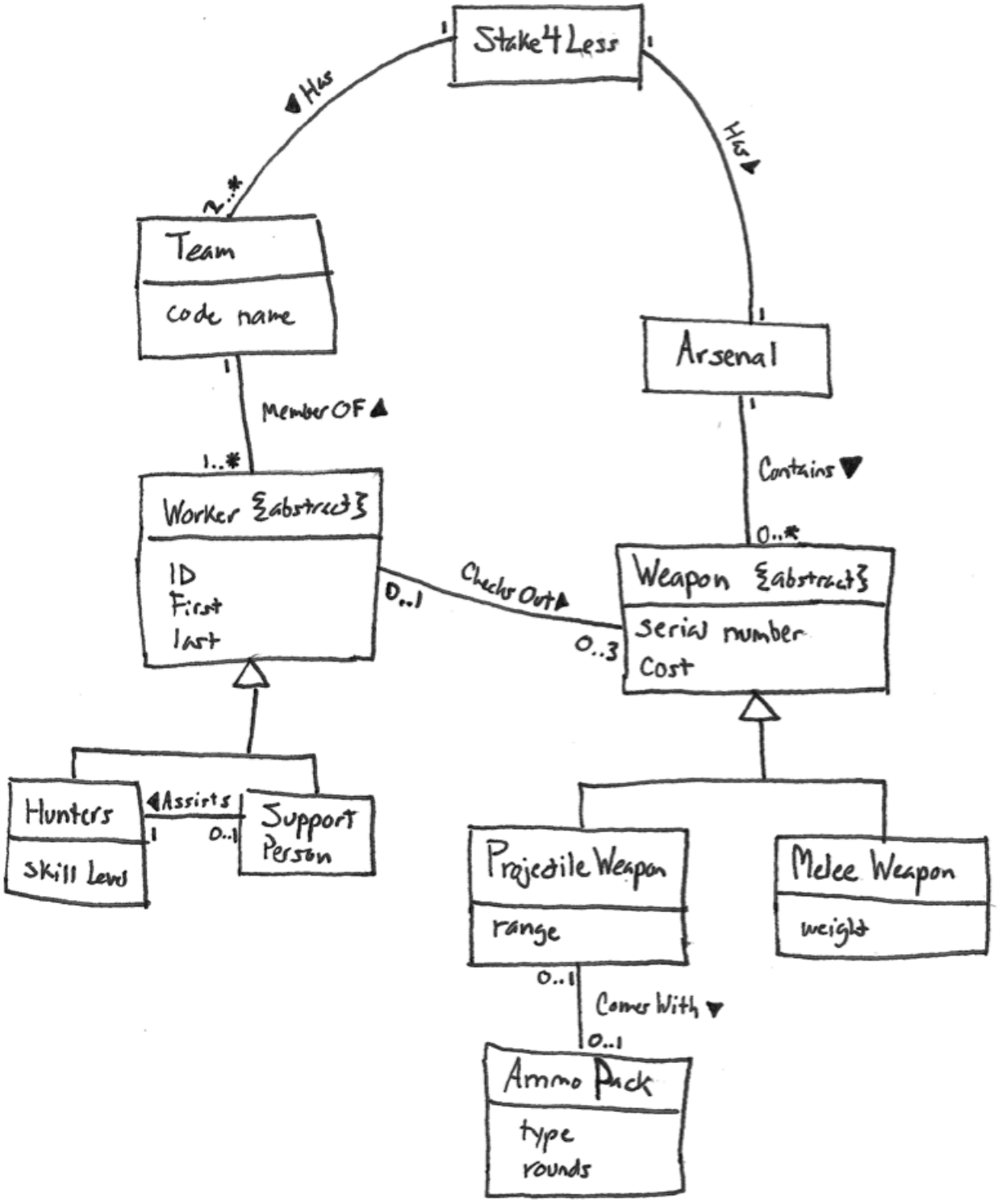
Solution:


Team
ID Number
Has
Archeologists Group
Has
Has
Equipment Inventory
Has
Researcher {abstract}
ID Number
First Name
Last Name
Checks Out
0..3
Piece of Equipment
Serial Number
Replacement Cost
Field Staff
Favorite Region
Lab Staff
0..2
0..1
Supports

Problem: Create a domain model (using class diagram notation) based on the following description.

You have been contracted to build an asset-management system for a vampire-hunting company, *Stake4Less*. The organization consists of multiple teams of workers. Each team has a code name. Each worker belongs to one of the teams, and has an ID number, a first name, and a last name. There are two types of workers: hunters and support personnel. Hunters have a skill level. Each support person assists exactly one hunter on the team. Some hunters on the team may not be assigned a support person. The company also manages an arsenal of weapons. Workers of any type may check out up to 3 weapons from the arsenal. Each weapon has a serial number and replacement cost. There are two types of weapons in the arsenal: projectile and melee. Projectile weapons have a range, and may come with an ammo pack. An ammo pack has an ammo type, and includes some number of rounds. Melee weapons have a weight.

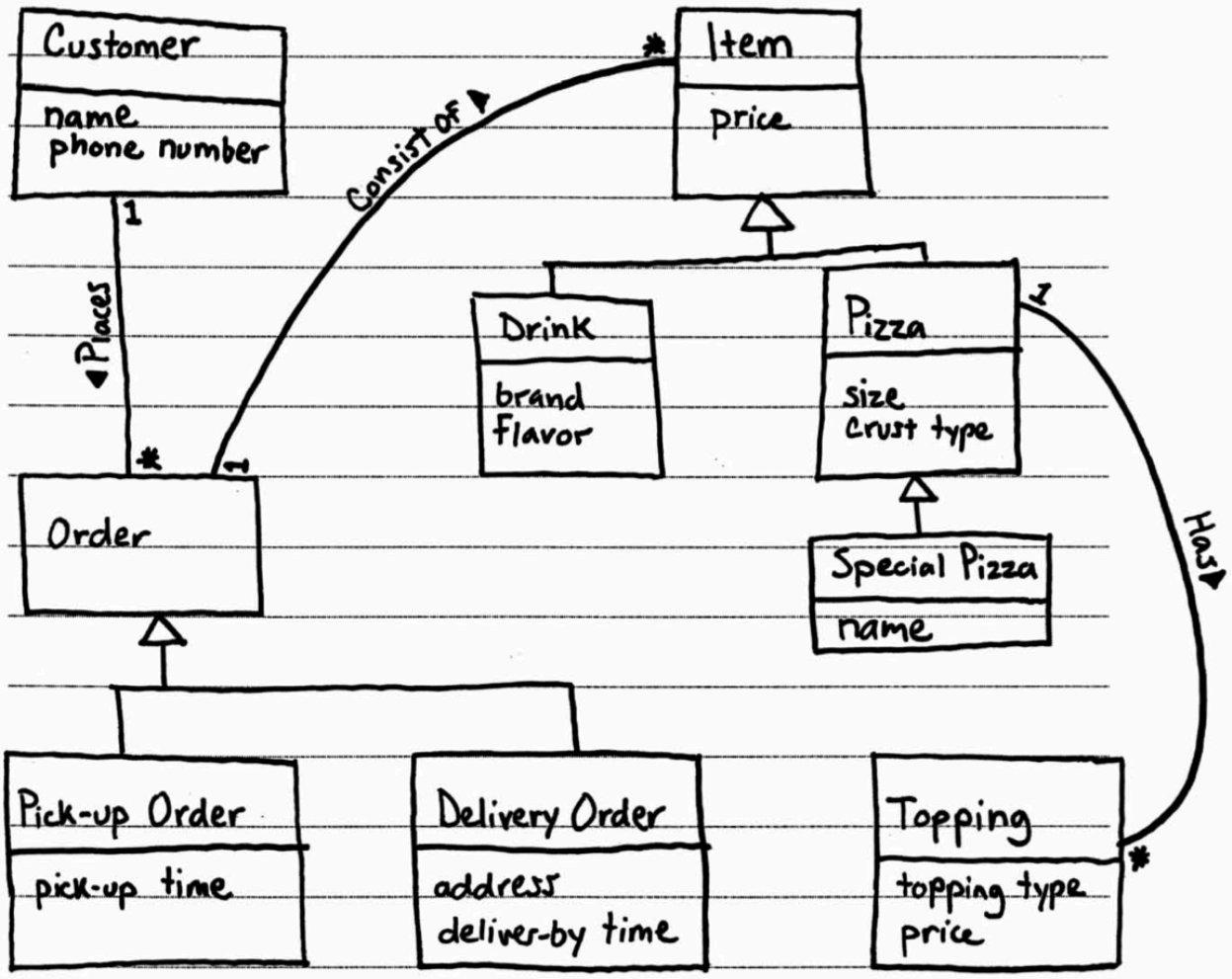
Solution:


Stake4Less
Has
Has
Team
code name
Arsenal
MemberOF
Contains
Worker {abstract}
ID
First
last
Checks Out
Weapon {abstract}
serial number
Cost
Hunters
skill Level
Assists
Support Person
Projectile Weapon
range
Melee Weapon
weight
Comes With
Ammo Pack
type
rounds

Problem: Imagine that you are tasked with developing a system for a pizza shop. Given the following description, create a Domain Model (in the form of a UML class diagram). Include all conceptual classes, attributes, associations, and generalization relationships mentioned in the descriptions. Label all associations and include all multiplicities.

A customer places orders. A customer has a name and phone number. There are two types of orders: pick-up and delivery. A pick-up order has a pick-up time. A delivery order has an address and deliver-by time. All orders consist of a set of items. There are two types of items: pizzas and drinks. All items have a price. A pizza has a size and a crust type. A pizza also has a number of toppings. A topping has a topping type and a price. Some pizzas are special pizzas that have a name (e.g., “Hawaiian” or “Meat Lovers”). A drink has a brand and a flavor.

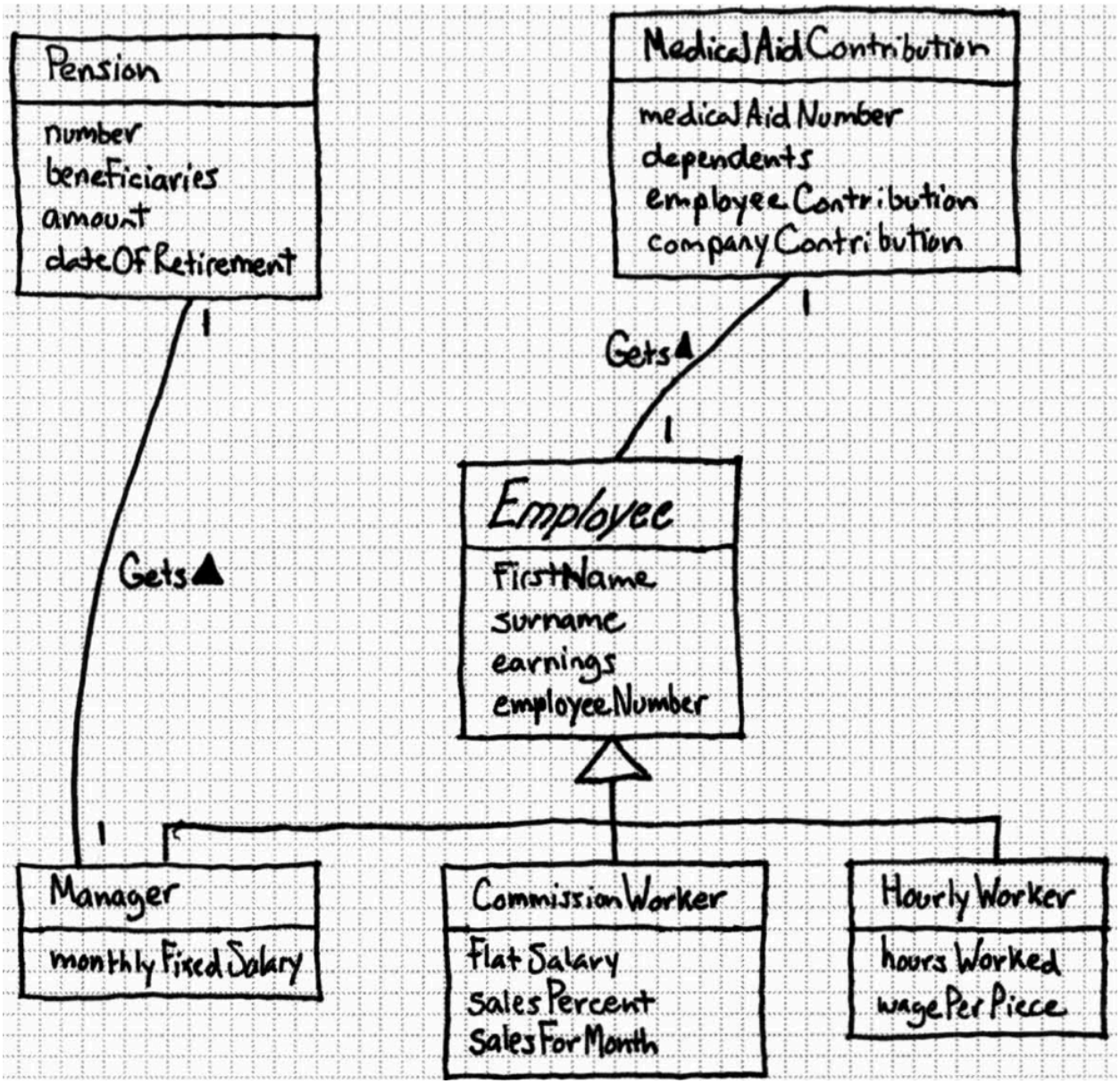
Solution:


Customer
name
phone number
Item
price
Consist of
1
Places
*
*
1
Drink
brand
Flavor
Pizza
size
Crust type
1
Has
Order
Special Pizza
name
Pick-up Order
pick-up time
Delivery Order
address
deliver-by time
Topping
topping type
price
*

Problem: Create a domain model in the form of a class diagram based on the following description. Model only things that are specifically described. Do not model "the system."

A company requires a payroll system. The company has three types of employees, namely, a manager, a commission worker, and an hourly worker. Information that needs to be stored for each employee is the employee's first name, surname, earnings and employee number. Additional information that must be stored for a manager is his/her monthly fixed salary; for a commission worker his/her flat salary, sales percent and the sales made for the month; for an hourly worker the number of hours worked and the wage per piece. All employees get a medical aid contribution. In addition to this, managers get a pension as a fringe benefit. Each medical aid instance specifies the medical aid number, dependents, employee contribution and company contribution, while each pension instance specifies the pension number, beneficiaries, amount and date of retirement.

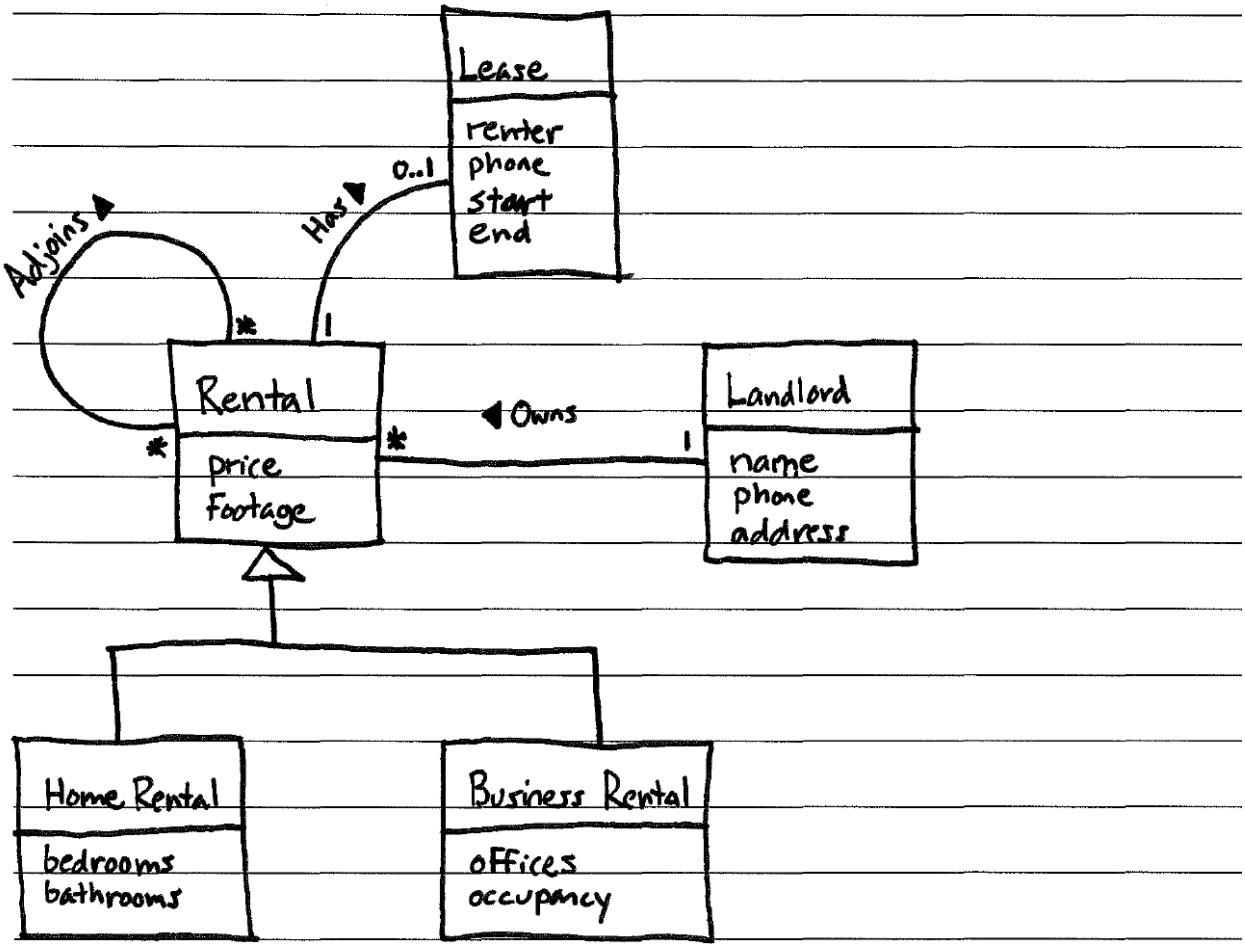
Solution:


Pension
number
beneficiaries
amount
dateOfRetirement
MedicalAidContribution
medicalAidNumber
dependents
employeeContribution
companyContribution
Gets
Employee
FirstName
surname
earnings
employeeNumber
Gets
Manager
monthlyFixedSalary
CommissionWorker
FlatSalary
salesPercent
salesForMonth
HourlyWorker
hoursWorked
wagePerPiece

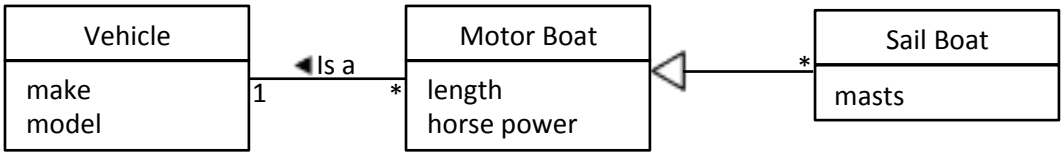
Problem: Draw a domain model in the form of a class diagram based on the following description. Model only things that are specifically described. Include all conceptual classes, attributes, associations, and generalization relationships mentioned. Label all associations and include all multiplicities. Do not model "the system."

> A system is needed for managing rental properties. Each rental property has a monthly rent price and square footage. Each rental is owned by a landlord with a name, a phone number, and an address. There are two types of rentals: home and business. Home rentals have a number of bedrooms and a number of bathrooms. Business rentals have a number of offices and a maximum occupancy. A rental may adjoin a number of other rental properties (like apartments in the same complex). If a rental property is currently rented, it will have an associated lease with the name of the renter, the renter's phone number, a start date, and an end date.

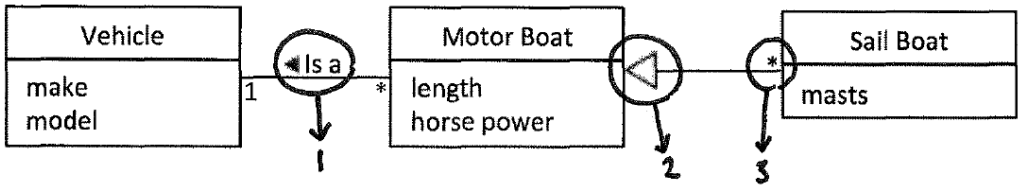
Solution:


Lease
renter
phone
start
end
0..1
Has
Adjoins
*
1
Rental
price
Footage
*
*
Owns
1
Landlord
name
phone
address
Home Rental
bedrooms
bathrooms
Business Rental
offices
occupancy

Problem: Name three problems with this domain model (esp. with respect to class diagram notation).



Solution:

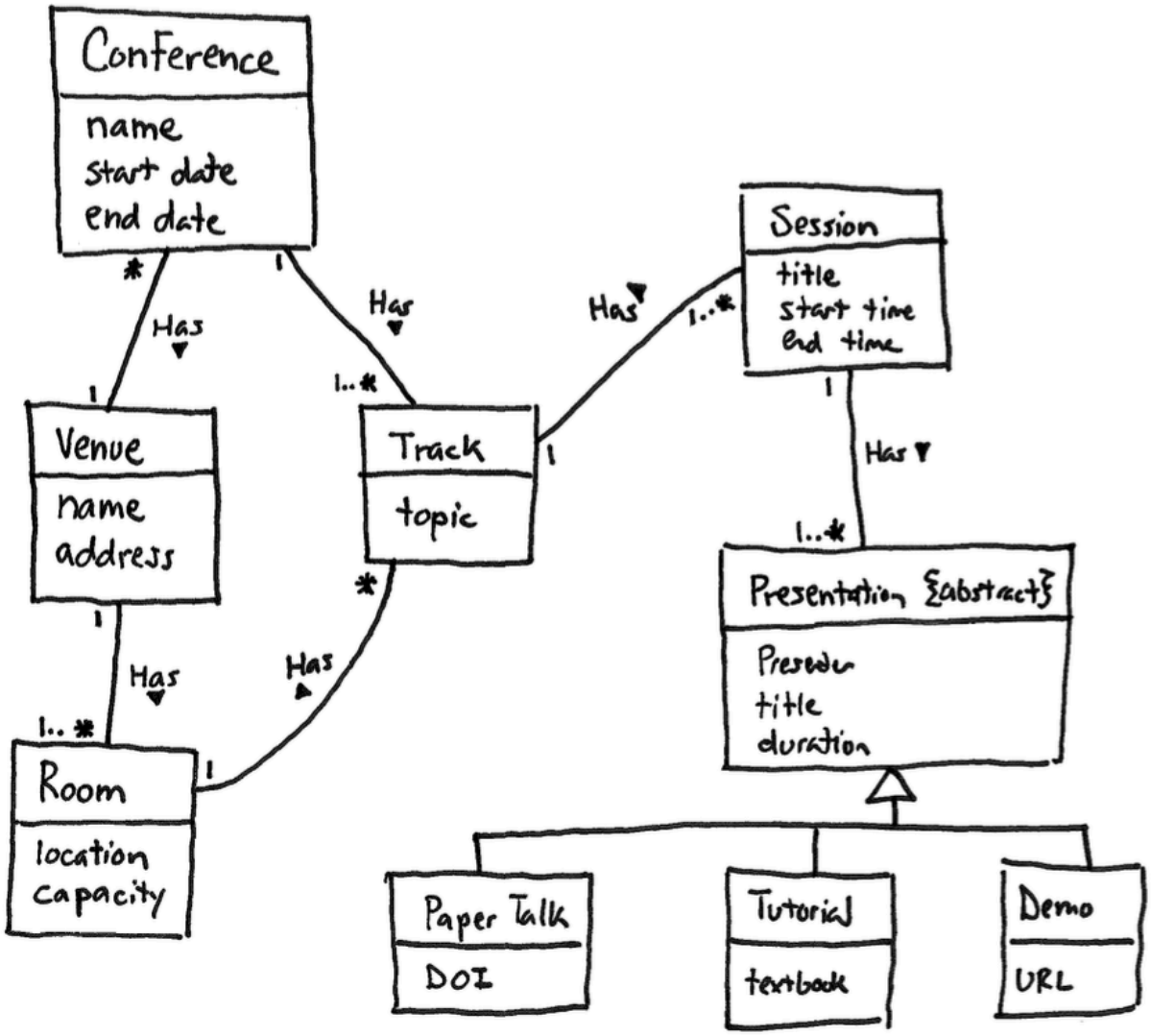


1. An "is-a" relationship should be modeled with a generalization (i.e., an arrow like this: ◁—).
2. This generalization violates both the Is-a and 100% rules: Not all sail boats have motors (with horse power),
3. Generalization relationships should not have multiplicities.

Problem: Create a domain model (using class diagram notation) based on the following description. Model only things that are specifically described. Include all conceptual classes, attributes, associations, and generalization relationships mentioned. Label all associations and include all multiplicities. Do not model "the system."

> You have been asked to build a conference-management system. Each conference has a name, start and end dates, and a venue. Conference venue has a name and an address. The venue also has rooms, each with a location description (e.g., "room 511") and a seating capacity. A conference can have several tracks, each with a topic (e.g., "research", "tutorial", "industry") and a designated room. Each track has one or more sessions. A session has title, a start time and an end time, and one or more presentations. There are three types of presentation: paper talk, tutorial, and demo. All presentations have a presenter name, title, and duration. A paper talk has a DOI code. A tutorial has a textbook description. A demo has a download URL.

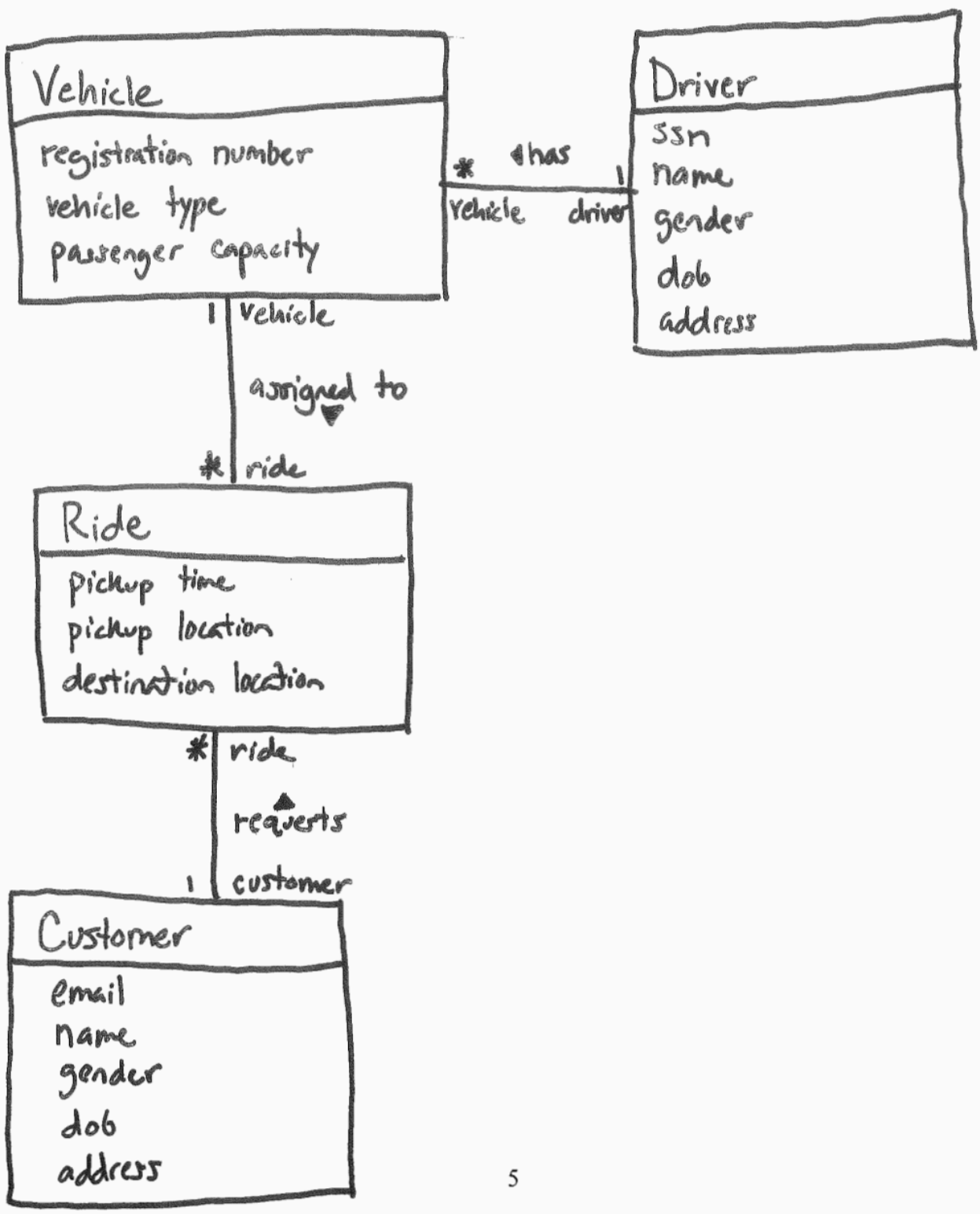
Solution:


Conference
name
start date
end date
Session
title
start time
end time
Has
Has
Has
Has
Has
Has
Venue
name
address
Track
topic
Presentation {abstract}
Presenter
title
duration
Room
location
capacity
Paper Talk
DOI
Tutorial
textbook
Demo
URL

Problem: Create a domain model (using class diagram notation) based on the following description. Model only things that are specifically described. Include all conceptual classes, attributes, associations, and generalization relationships mentioned. Label all associations and association ends and include all multiplicities. Do not explicitly model the "system".

You have been asked to build a taxicab system like Uber. A driver can register one or more of their vehicles with the system. Vehicles will have a registration number, a type (sedan, SUV, van) and a passenger capacity. A driver has some personal information including their SSN, name, gender, date of birth, and address. Potential customers will be able to sign up with the system by submitting their email and some personal information, including name, gender, date of birth, and address. Once registered, customers will be able to request a ride with a specific vehicle. The customer will need to specify the time and location (as address) of pickup and the destination location address.

Solution:


Vehicle
registration number
vehicle type
passenger capacity
Driver
ssn
name
gender
dob
address
* has 1
vehicle driver
1 vehicle
assigned to
* ride
Ride
pickup time
pickup location
destination location
* ride
requests
1 customer
Customer
email
name
gender
dob
address