University of British Columbia, Department of Computer Science

CPSC 304

2013 Winter Term 1

Project Part 2

Group Name: Bakerzin

Group Members:

Name	Student Number	Unix ID	Email Address
Reinhard Simeon	18711119	l1e8	reinhardsimeon@gmail.com
Dallas Leclerc	23681117	v4c8	dallas_leclerc@live.com
Kai Hin (Henry) Tang	42319103	m7x7	ht92@live.ca
Diana Sutandie	29929106	z8p7	dianasutandie@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

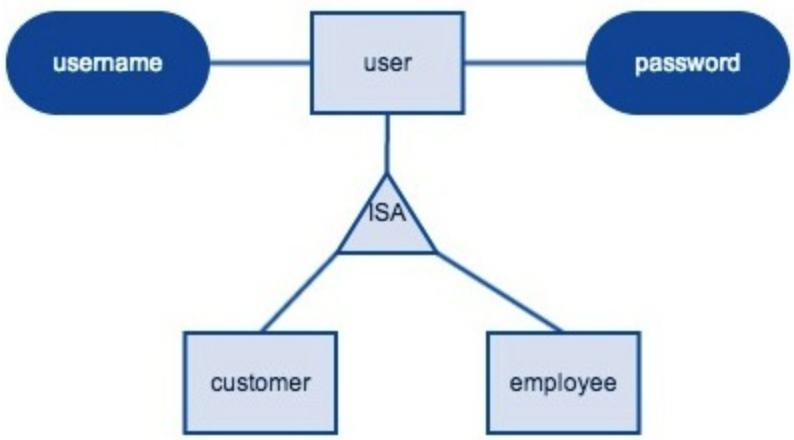
1. E/R Diagram

1.1 Updated E/R Diagram

For this submission, we have updated our E/R Diagram by adding more entities, attributes and participation constraints. The updated diagram can be found on the **last page**.

1.2 Update Details

- **User credentials are now stored in our database model.**



Since there will be two different classes of users, it is essential to store their credentials information. New entity called *user* is introduced, and it has an ISA relationship with *customer* and *employee*. User has *username* and *password* as its attributes. *username* has to be unique for every tuple of *user*.

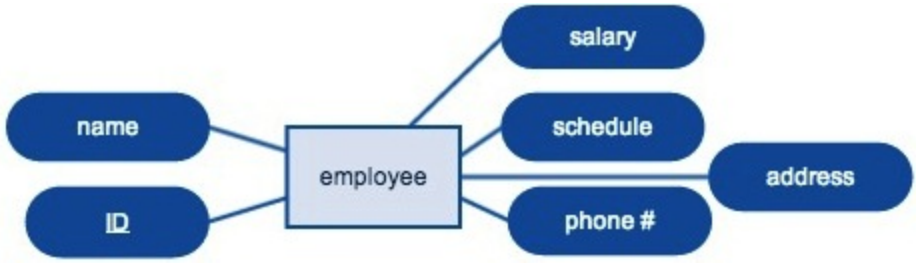
- **Total participation for *make* relationship among *order*, *customer* and *employee*.**

Each order must be made by one customer and be processed by one employee. However, customer can make more than one order, and the same rule applies for employee.

- **Total participation for *restock* relationship between *employee* and *inventory*.**

In this model, we assume that inventories have to be done manually, which requires employee. Without any employees, there are no inventory restock happening.

- **More attributes are added to *employee* entity.**



Since there will be particular interface for employees, we are storing *schedule, vacation* and *salary* information for *employee*. On the implementation, this information is only available in employee's interface, and only employees with certain positions can has the permission to access and modify this asdfa(with different view).

2. Schema

2.1 Table definitions

Customer (ID integer
phoneNumber string,
name string,
address string,
userName string UNIQUE,
password char[20])

Employee (ID integer,
name string,
salary double,
schedule string,
address string,
phoneNumber string,
userName string UNIQUE,
password char[20])

Order (orderID integer,
customerID integer,,
employeeID integer,
orderDate date,
madeFor date)

- customerID is a foreign key NOT NULL
- employeeID is a foreign key NOT NULL

Restocks (employeeID integer,
inventoryType string)

- Foreign Key: (Employee.ID), (Inventory.Type)

Inventory (Type string,
expiryDate date,
quantity integer,
cost integer)

OrderStockFrom (employeeID integer, supplierPhoneNumber string)

- Foreign Key: (employeeID), (supplierPhoneNumber)

Supplier (ID integer,
phoneNumber string,
companyName string,
address string,
cost integer,
itemType string)

Member (memberSince date,
customerID integer,
type string,
rewardPoints integer)

- Foreign Key: (customerID)

Student (customerID integer,
totalTuitionFee double,
scholarship double,
prerequisite string)

- Foreign Key: (phoneNumber)

EnrollsIn (customerID integer,
classID char[7],
startDate date)

- Foreign Key: (customerID), (ClassID, StartDate)

Class_TaughtBy (classID char[7],
startDate date,
type string,
fee integer,
meetingTime string,
Duration integer
employeeID integer NOT NULL)

- Foreign Key: employeeID

Instructor (employeeID integer,
academicBackground string)

- Foreign Key: (Employee.ID)

Baker (employeeID integer,
specialization string)

- Foreign Key: (Employee.ID)

Bake (employeeID integer, itemID integer)

- Foreign Key: (Employee.ID), (Item.ID)

Item (ID integer, price double, type string, flavour string)

Contains (itemsID integer, orderID integer, trackingID integer)

- Foreign Key: (itemID), (orderID, trackingID)

WithDeals_Shipment (TrackingID integer,
ShippingTime string,
ShippingType string,
ShippingDate string,
fee integer,
ShippingAddress string
orderID integer)

- Foreign Key: (orderID)

2.2 SQL DDL to create the Tables

CREATE TABLE Customer (ID INTEGER PRIMARY KEY
phoneNumber CHAR[10],
name CHAR[20],
address CHAR[35],

```
                    userName CHAR[20] UNIQUE,
                    password CHAR[20])

CREATE TABLE Employee (  ID INTEGER PRIMARY KEY,
                    name CHAR[20],
                    salary FLOAT,
                    schedule CHAR[25],
                    address CHAR[35],
                    phoneNumber CHAR[10],
                    userName CHAR[20] UNIQUE,
                    password CHAR[20])

CREATE TABLE Order ( orderID  INTEGER PRIMARY KEY,
                    customerID INTEGER NOT NULL,
                    employeeID INTEGER NOT NULL,
                    orderDate DATE,
                    madeFor DATE,
                    FOREIGN KEY(customerID) REFERENCES Customer,
                    ON DELETE CASCADE
                    FOREIGN KEY(employeeID) REFERENCES Employee
                    ON UPDATE CASCADE)

CREATE TABLE  Restocks (employeeID INTEGER,
                     inventoryType CHAR[15],
                    PRIMARY KEY(employeeID, inventoryType),
                    FOREIGN KEY(employeeID) REFERENCES Employee,
                    ON UPDATE CASCADE
                    FOREIGN KEY(inventoryType) REFERENCES Inventory
                    ON UPDATE CASCADE
                    ON DELETE CASCADE)

CREATE TABLE Inventory (Type CHAR[15] PRIMARY KEY,
                    expiryDate DATE,
                    quantity INTEGER,
                     cost INTEGER)

CREATE TABLE OrderStockFrom (employeeID INTEGER,
                         supplierID INTEGER,
                         PRIMARY KEY(employeeID, supplierID),
```

```
                                    FOREIGN KEY(employeeID) REFERENCES Employee
                                   ON UPDATE CASCADE
                                    FOREIGN KEY(supplierID) References Supplier
                                    ON UPDATE CASCADE )

CREATE TABLE Supplier ( ID INTEGER PRIMARY KEY
                                    phoneNumber CHAR[10],
                                    companyName CHAR[15],
                                    address CHAR[35],
                                    cost INTEGER,
                                    itemType CHAR[15])

CREATE TABLE Member (memberSince date,
        customerID integer,
        type string,
        rewardPoints integer
        PRIMARY KEY customerID,
        FOREIGN KEY (customerID) REFERENCE Customer)

CREATE TABLE Student(customerID integer,
        prerequisite string,
        scholarship double,
        totalTuitionFee double,
        PRIMARY KEY customerID,
        FOREIGN KEY (customerID) REFERENCE Customer)

CREATE TABLE EnrollsIn (customerID integer,
                        classID char[7],
                        startDate date
                        PRIMARY KEY customerID, classID, startDate,
                        FOREIGN KEY (customerID) REFERENCES Student
                        ON UPDATE CASCADE,
                        FOREIGN KEY (classID, startDate) REFERENCES class
                        ON UPDATE CASCADE)


CREATE TABLE Class_TaughtBy (classID STRING,
                startDate DATE,
                type STRING,
```

fee INTEGER,
meetingTime STRING,
Duration INTEGER,
employeeID INTEGER NOT NULL,
PRIMARY KEY classID, employeeID,
FOREIGN KEY (employeeID) REFERENCES Employee
ON DELETE NO ACTION,
ON UPDATE CASCADE)

CREATE TABLE Instructor (employeeID INTEGER,
academicBackground STRING,
PRIMARY KEY employeeID,
FOREIGN KEY (employeeID) REFERENCES Employee)

CREATE TABLE Baker (employeeID INTEGER,
specialization STRING.
PRIMARY KEY employeeID,
FOREIGN KEY (employeeID) REFERENCES Employee)

CREATE TABLE Bake (employeeID INTEGER,
itemID INTEGER,
PRIMARY KEY employeeID, itemID,
FOREIGN KEY (employeeID) REFERENCES Employee,
FOREIGN KEY (itemID) REFERENCES Item)

CREATE TABLE Item (ID INTEGER,
price DOUBLE,
type STRING,
flavour STRING,
PRIMARY KEY ID)

CREATE TABLE Contains (itemsID INTEGER,
orderID INTEGER,
trackingID INTEGER
PRIMARY KEY (itemsID, orderID, trackingID),
FOREIGN KEY (itemsID) REFERENCES Items,
FOREIGN KEY (orderID, trackingID) REFERENCES Order)

```
CREATE TABLE WithDeals_Shipment (TrackingID INTEGER,
                    ShippingTime STRING,
                    ShippingType STRING,
                    ShippingDate STRING,
                    fee INTEGER,
                    ShippingAddress STRING
                    orderID INTEGER
                    PRIMARY KEY(orderID, TrackingID),
                    FOREIGN KEY(orderID) REFERENCES Order,
                    ON DELETE CASCADE)
```

2.3 Tables with sample data

CUSTOMER

```
INSERT
INTO      Customer (ID, phoneNumber, name, address, userName, password)
VALUES  (5000, '778-123-4567', 'Susan Smith', '12 Hazel Street, Burnaby', 'ssmith', 'smith123')

INSERT
INTO      Customer (ID, phoneNumber, name, address, userName, password)
VALUES  (5001, '778-859-9393', 'Carlos Tang', '33 Cornell Ave, Vancouver', 'ctang', 'carlosssh')

INSERT
INTO      Customer (ID, phoneNumber, name, address, userName, password)
VALUES  (5002, '604-159-8645', 'Salma Hassan', '1330 Fir Street, Vancouver', 'shassan',
'02091981')

INSERT
INTO      Customer (ID, phoneNumber, name, address, userName, password)
VALUES  (5003, '604-321-6214', 'Guy Lemiux', '200 Ash Street, Vancouver', 'glemiux', 'guyguy')

INSERT
INTO      Customer (ID, phoneNumber, name, address, userName, password)
VALUES  (5004, '778-545-6877', 'Paul Stonestreet', '3387 Oak Streeet, Vancouver',
'pstonestreet', 'paulstreet')
```

INSERT
INTO Customer (ID, phoneNumber, name, address, userName, password)
VALUES (5005, ‘604-989-5511’, ‘Tony Xu’, ‘14 Port Ave’, ‘txu’, ‘momo’)

INSERT
INTO Customer (ID, phoneNumber, name, address, userName, password)
VALUES (5006, ‘604-778-4212’, ‘Greg Thomas’, ‘19 Slow Road’, ‘gthomas’, ‘iamthomas’)

EMPLOYEE

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0001, ‘John Doe’, ‘5500.0’, ‘Mon-Tue-Wed-Thu-Fri’, ‘4808 McKay Ave, Burnaby’, ‘604-321-3322’, ‘jdoe’ , ‘546456’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0002, ‘Helena Chan’, ‘6800.0’, ‘Mon-Tue-Wed-Thu-Fri-Sat-Sun’, ‘1000 University Road, Vancouver’, ‘778-987-6464’, ‘hchan’, ‘america’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0003, ‘Chris Henkel’, ‘6800.0’, ‘Mon-Tue-Wed-Thu-Fri-Sat-Sun’, ‘88 Unity Way, Langley’, ‘778-320-9854’, ‘chenkel’, ‘unityway’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0004, ‘Yoana Gerson’, ‘6800.0’, ‘Mon-Tue-Wed-Thu-Fri-Sat-Sun’, ‘3 West 16th Avenue, Vancouver’, ‘604-878-9813’, ‘ygerson’, ‘yoanagerS0n’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0005, ‘Herbert Powter’, ‘3300.0’, ‘Mon, Wed, Fri’, ‘19 Yolk Street, Langley’, ‘604-159-6587’, ‘hpowter’, ‘123123’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0006, ‘Carla Jarre’, ‘8300.0’, ‘Mon, Wed, Fri’, ‘10 Gong Street, Maple Ridge’, ‘604-773-5513’, ‘cjarre’, ‘jarresweet’)

INSERT
INTO Employee (ID, name, salary, schedule, address, phoneNumber, userName, password)
VALUES (0007, ‘Smith Bourne’, ‘6300.0’, ‘Tue Thu’, ‘10 Cricket Ave, Maple Ridge’, ‘604-975-1245’, ‘sbourne’, ‘s1234’)

ORDER

INSERT
INTO Order(orderID, customerID, employeeID, orderDate, madeFor)
VALUES (9000, 5000, 0002, ‘2013-09-13’, ‘2013-09-23’)

INSERT
INTO Order(orderID, customerID, employeeID, orderDate, madeFor)
VALUES (9001, 5003, 0002, ‘2013-09-13’, ‘2013-09-15’)

INSERT
INTO Order(orderID, customerID, employeeID, orderDate, madeFor)
VALUES (9002, 5001, 0003, ‘2013-09-14’, ‘2013-09-15’)

INSERT
INTO Order(orderID, customerID, employeeID, orderDate, madeFor)
VALUES (9003, 5004, 0003, ‘2013-09-14’, ‘2013-09-20’)

INSERT
INTO Order(orderID, customerID, employeeID, orderDate, madeFor)
VALUES (9004, 5002, 0003, ‘2013-09-14’, ‘2013-09-22’)

RESTOCKS

INSERT
INTO Restocks(employeeID, inventoryType)
VALUES (0005, ‘baking soda’)

INSERT
INTO Restocks(employeeID, inventoryType)
VALUES (0005, ‘sugar’)

INSERT

```
INTO      Restocks(employeeID, inventoryType)
VALUES  (0005, ‘flour’)

INSERT
INTO      Restocks(employeeID, inventoryType)
VALUES  (0005, ‘icing sugar’)

INSERT
INTO      Restocks(employeeID, inventoryType)
VALUES  (0005, ‘chocolate syrup’)
```

INVENTORY

```
INSERT
INTO      Inventory(type, expiryDate, quantity, cost)
VALUES  (‘sugar’, ‘2017-09-28’, 200, 400)

INSERT
INTO      Inventory(type, expiryDate, quantity, cost)
VALUES  (‘flour’, ‘2018-01-01’, 150, 3000)

INSERT
INTO      Inventory(type, expiryDate, quantity, cost)
VALUES  (‘baking soda’, ‘2015-03-19’, 13, 39)

INSERT
INTO      Inventory(type, expiryDate, quantity, cost)
VALUES  (‘icing sugar’, ‘2015-08-14’, 30, 30)

INSERT
INTO      Inventory(type, expiryDate, quantity, cost)
VALUES  (‘chocolate syrup’, ‘2016-04-13’, 10, 150)
```

OrderStockFrom

```
INSERT
INTO      OrderStockFrom(employeeID, supplierID)
VALUES  (0005, 55000)
```

INSERT
INTO OrderStockFrom(employeeID, supplierID)
VALUES (0005, 55005)

INSERT
INTO OrderStockFrom(employeeID, supplierID)
VALUES (0005, 55001)

INSERT
INTO OrderStockFrom(employeeID, supplierID)
VALUES (0005, 55002)

INSERT
INTO OrderStockFrom(employeeID, supplierID)
VALUES (0005, 55003)

INSERT
INTO OrderStockFrom(employeeID, supplierID)
VALUES (0005, 55004)

Supplier

INSERT
INTO Supplier(ID, phoneNumber, companyName, address, cost, itemType)
VALUES (55000, '604-315-7598', 'Sweet Sugar Ltd', '123 Alberni Street, Vancouver', 2, 'sugar')

INSERT
INTO Supplier(ID, phoneNumber, companyName, address, cost, itemType)
VALUES (55001, '604-147-8523', 'Baking Needs', '1 Hampton Place, Vancouver', 3, 'baking soda')

INSERT
INTO Supplier(ID, phoneNumber, companyName, address, cost, itemType)
VALUES (55002, '647-323-8426', 'Hai Tong Company', '1100 Busy Road, Berrie', 20, 'flour')

INSERT
INTO Supplier(ID, phoneNumber, companyName, address, cost, itemType)
VALUES (55003, '604-315-7598', 'Organic Producers Company', '1333 McKay Avenue, Vancouver', 15, 'chocolate syrup')

INSERT
INTO Supplier(ID, phoneNumber, companyName, address, cost, itemType)
VALUES (55004, '604-451-2984', 'Cheap Baking Needs', '1414 West Hastings, Vancouver', 1, 'icing sugar')

MEMBER

INSERT
INTO Member(memberSince, customerID, type, rewardPoints)
VALUES ('2009-10-10', 5006, 'Gold', 13200)

INSERT
INTO Member(memberSince, customerID, type, rewardPoints)
VALUES ('2010-12-5', 5004, 'Green', 200)

INSERT
INTO Member(memberSince, customerID, type, rewardPoints)
VALUES ('2010-12-27', 5005, 'Bronze', 675)

INSERT
INTO Member(memberSince, customerID, type, rewardPoints)
VALUES ('2012-01-05', 5001, 'Gold', 15200)

INSERT
INTO Member(memberSince, customerID, type, rewardPoints)
VALUES ('2013-10-20', 5002, 'Green', 50)

Student

INSERT
INTO Student(customerID, totalTuitionFee, scholarship, prerequisite)
VALUES (5001, 4521.13, 0.0, 'Baking 101')

INSERT
INTO Student(customerID, totalTuitionFee, scholarship, prerequisite)
VALUES (5003, 5530.00, 0.0, 'Baking 101, Decoration 101')

INSERT

INTO Student(customerID, totalTuitionFee, scholarship, prerequisite)
VALUES (5004, 4521.13, 0.0, ‘Baking 101’)

INSERT
INTO Student(customerID, totalTuitionFee, scholarship, prerequisite)
VALUES (5005, 3200.45, 0.0, null)

INSERT
INTO Student(customerID, totalTuitionFee, scholarship, prerequisite)
VALUES (5006, 10530.00, 1000.0, ‘Baking 101, Decoration 101’)

EnrollsIn

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5001, ‘BAKE102’ , ‘2013-11-01’)

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5004, ‘BAKE102’ , ‘2013-11-01’)

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5003, ‘DECR102’ , ‘2013-11-01’)

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5006, ‘BAKE102’ , ‘2013-11-01’)

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5006, ‘DECR102’ , ‘2013-10-25’)

INSERT
INTO EnrollsIn(customerID, classID, startDate)
VALUES (5005, ‘BAKE101’ , ‘2013-11-20’)

Instructor

INSERT
INTO Instructor(employeeID, academicBackground)
VALUES (006, ‘Diploma in Culinary and Baking’)

INSERT
INTO Instructor(employeeID, academicBackground)
VALUES (007, ‘Master in Culinary, Decorating and Icing’)

INSERT
INTO Instructor(employeeID, academicBackground)
VALUES (001, ‘Diploma in Culinary and Baking’)

INSERT
INTO Instructor(employeeID, academicBackground)
VALUES (003, ‘Certification on Pastry’)

INSERT
INTO Instructor(employeeID, academicBackground)
VALUES (004, ‘Master in Baking with Specialization on Wedding Cake’)

Item

INSERT
INTO Item (ID, price, type, flavour)
VALUE (60000, 3, ‘Cupcake’, ‘Red Velvet’)

INSERT
INTO Item (ID, price, type, flavour)
VALUE (60001, 5, ‘Ice Cream, ‘Chocolate)

INSERT
INTO Item (ID, price, type, flavour)
VALUE (60002, 3, ‘Cheese Cake, ‘Strawberry)

INSERT
INTO Item (ID, price, type, flavour)
VALUE (60003, 4, ‘Cheese Cake’, ‘Red Velvet’)

INSERT
INTO Item (ID, price, type, flavour)
VALUE (60004, 4, ‘Milkshake’, ‘Original’)

Class_TaughtBy

INSERT
INTO Class_TaughtBy(classID, startDate, type, fee, meetingTime, duration)
VALUE (‘BAKE102’, ‘2013-11-01’. ‘Baking’, ‘Mon, Wed’, 2)

INSERT
INTO Class_TaughtBy(classID, startDate, type, fee, meetingTime, duration)
VALUE (‘BAKE101’, ‘2013-11-20’. ‘Baking’, “Mon, Wed, Friday”, 1)

INSERT
INTO Class_TaughtBy(classID, startDate, type, fee, meetingTime, duration)
VALUE (‘DECR101’, ‘2013-09-06’. ‘Decoration’, “Mon, Wed”, 2)

INSERT
INTO Class_TaughtBy(classID, startDate, type, fee, meetingTime, duration)
VALUE (‘DECR102’, ‘2013-11-01’. ‘Decoration’, ‘Tue, Wed’, 2)

INSERT
INTO Class_TaughtBy(classID, startDate, type, fee, meetingTime, duration)
VALUE (‘BAKE202’, ‘2013-09-04’. ‘Baking’, “Tue, Thurs”, 2)

Bake

INSERT
INTO Bake (employeeID, itemID)
VALUE(0001, 60002)

INSERT
INTO Bake (employeeID, itemID)
VALUE(0002, 60003)

INSERT
INTO Bake (employeeID, itemID)
VALUE(0003, 60004)

```
INSERT
INTO Bake (employeeID, itemID)
VALUE( 0004, 60000)

INSERT
INTO Bake (employeeID, itemID)
VALUE( 0005, 60001)
```

Baker

```
INSERT
INTO Baker (employeeID, specialization)
VALUE (0006, 'Pastries')

INSERT
INTO Baker (employeeID, specialization)
VALUE (0005, 'Bread')

INSERT
INTO Baker (employeeID, specialization)
VALUE (0004, 'Cakes')

INSERT
INTO Baker (employeeID, specialization)
VALUE (0003, 'Doughnuts')

INSERT
INTO Baker (employeeID, specialization)
VALUE (0002, 'Cookies')
```

WithDeals_Shipment

```
INSERT
INTO WithDeals_Shipment (TrackingID, ShippingTime, ShippingType,
                         ShippingDate, fee,ShippingAddress, orderID)
VALUE (75000, '15:30', 'UPS', '2013-12-01', 5, '33 Cornell Ave, Vancouver', 9004)

INSERT
```

INTO WithDeals_Shipment (TrackingID, ShippingTime, ShippingType,
ShippingDate, fee,ShippingAddress, orderID)
VALUE (75001, ‘15:30’, ‘Purolator’, ‘2013-12-01’, 5, ‘3387 Oak Streeet, Vancouver’, 9003)

INSERT
INTO WithDeals_Shipment (TrackingID, ShippingTime, ShippingType,
ShippingDate, fee,ShippingAddress, orderID)
VALUE (75002, ‘15:30’, ‘Registered Mail’, ‘2013-12-01’, 5, ‘12 Hazel Street, Burnaby’, 9002)

INSERT
INTO WithDeals_Shipment (TrackingID, ShippingTime, ShippingType,
ShippingDate, fee,ShippingAddress, orderID)
VALUE (75003, ‘15:30’, ‘FedEx’, ‘2013-12-01’, 5, ‘1330 Fir Street, Vancouver’, 9001)

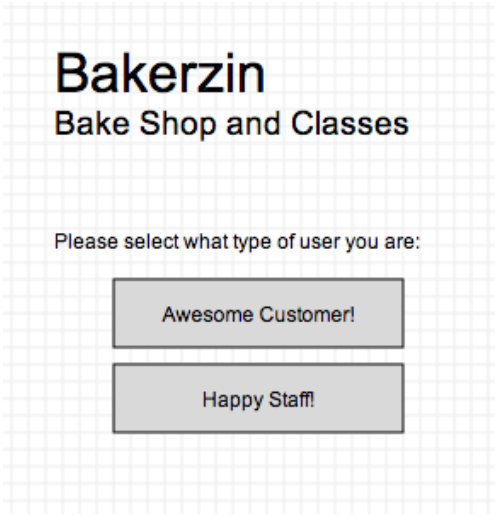
INSERT
INTO WithDeals_Shipment (TrackingID, ShippingTime, ShippingType,
ShippingDate, fee,ShippingAddress, orderID)
VALUE (75004, ‘15:30’, ‘Regular Mail’, ‘2013-12-01’, 5, ‘200 Ash Street, Vancouver’, 9000)

2.4 Platform and Technologies

We will be using the the CS Ugrad Oracle installation to implement our database, and we will use PHP to interface with the database.

2.5 Functionalities

As mentioned in earlier section, there will be two different classes of users, which are *customer* and *employee*. In order to accommodate each class properly, the program will have two different interfaces with distinct functionalities and access levels. Once the program starts, user will be prompted to click a category which describe their user type (refer to figure below). Depending on the choice, user will be brought to the associated interface. Further details for each interface will be outlined in subsequent sections.



Customer

List of functionalities for Customer's interface:

- Make a request to be Bakerzin's member
- Sign-in to member's account and see reward details
- Access list of items
- Access list of offered courses and instructors associated with each class.

Employees (Staff)

List of functionalities for Employees' interface:

- Create, modify and track status of order
- Contact suppliers and restock inventories
- Access working schedules (work hours is fixed, only working days is not fixed)
- Access salary and accept proposed schedule from subordinate employees (for manager and owner only)
- Organize, create and modify offered classes for the season
- Assign instructors for specific class
- Access each student's account and its details, like scholarship, pre-requisites and status of tuition fees.

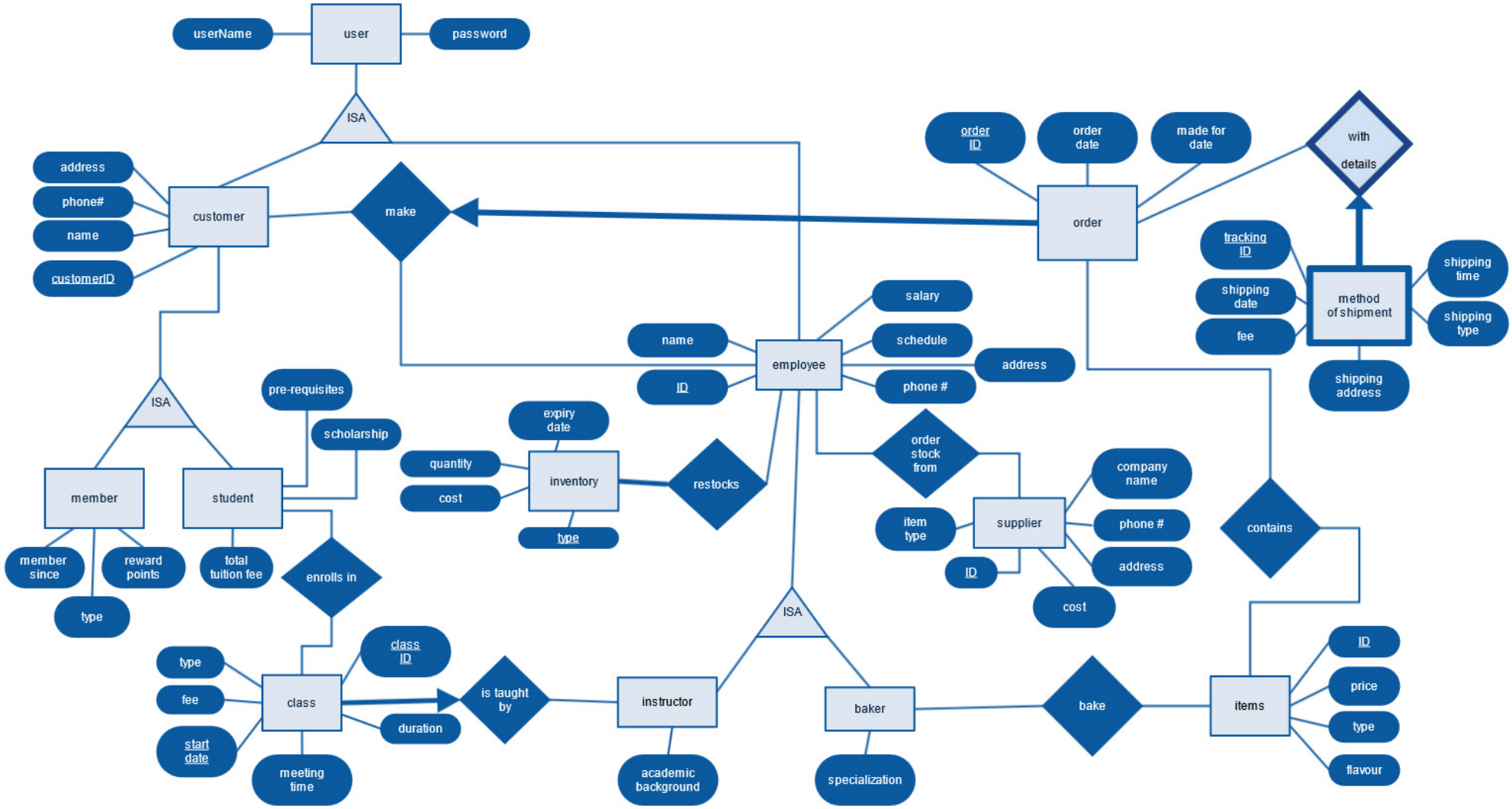
2.6 Division of Labor

At this point, the task is divided roughly to be like the following:
Dallas Leclerc: database

Reinhard Simeon: GUI - (for customer)
Diana Sutandie: GUI - (for employees)
Henry Tang: database

2.7 Particular feedback

- We want to know if there is redundancy on our ISA relationships, especially the *employee, instructor* and *baker*.


userName
user
password
ISA
address
phone#
name
customerID
customer
make
order ID
order date
made for date
order
with details
tracking ID
shipping date
fee
method of shipment
shipping time
shipping type
shipping address
salary
schedule
address
phone #
name
ID
employee
ISA
pre-requisites
scholarship
member
student
member since
reward points
type
total tuition fee
enrolls in
expiry date
quantity
cost
inventory
type
restocks
order stock from
item type
supplier
company name
phone #
address
ID
cost
contains
ISA
type
fee
start date
meeting time
class
class ID
duration
is taught by
instructor
academic background
baker
specialization
bake
items
ID
price
type
flavour