**Final Project Proposal**

**Group: Wandi Zhang**

**Jianle Chen**

**Project Scenario**

Nowadays, with the improvement of people's living standard and the popularity of the networks, housekeeping service providers tend to be electronical, which has become a new trend. People buying housekeeping services online becomes a new trend.

We have been asked by a local housekeeping company in Tacoma, Washington to help them transition to a database driven book catalog, employee, and booking schedule system. We have been chosen as the DBA for this project and will oversee all aspects of the database components of the new system. The housekeeping company has only a single location, Tacoma and only provide services within the state. Up to this point, the manager has been manually schedule arrangements through excel.

We are required to build the backend and a programming interface for the front end that abstracts all of the database interaction. The front end should be able to instantiate the class and use it's methods to do all the interactions that needs without knowing a thing about the database.

This is just a first usable prototype of the system, so many features will not yet be implemented.

**Project Artifacts**

The database application implements the main functions that users need in ordering a housekeeping service online. The users of this application are able to register and login accounts, search for housekeepers based on demands and preferences, order and buy services, and check the current order services.

The company provided us with their records of employees and booking arrangements. There are four types of services provided, babysitting, hourly-paid housekeeper, housekeeper and personal tutor. Each employee hired by the company can be registered for only one kind of service. And each worker has their own company belongs to and price for their service. Hourly-paid and personal tutor are paid by hour, while babysitting and housekeeper are paid by day. More detailed settings are written below in the schema Design part.

**Deliverable Details**

**Database API**

The frontend application is a desktop-based window application programmed with C#. **未写完**

**Schema Design**

Final normalized schema design:

user(userID, username, password, address, phone, registerTime)

worker(workerID, name, companyID, SSN, nationality, gender)

workerDetails(workerID, introduction, availableTime, type, weight, price, fitness, language, image, price, hasMarried, hasCertificate, evaluation)

company(companyID, name, foundTime, introduction)

order(orderID, orderNumber, userID, workerID)

orderDetails(orderID, , orderState, serviceAddress, details, orderTime, payment, startDate, endDate)

Start with the following schema:

*user(userID, username, password, address, phone, registerTime)*

*worker(workerID, type, name, companyID, SSN, image, availableTime, price, nationality, introduction, fitness, gender, weight, hasMarried, language, hasCertificate, evaluation)*

*company(companyID, name, foundTime, introduction)*

*order(orderID, orderNumber, userID, workerID, orderState, serviceAddress, details, orderTime, payment, startDate, endDate)*

where,

userID is the primarykey for table user.

workerID is the primarykey for table worker, while companyID is the foreign key.

companyID is the primarykey for table company.

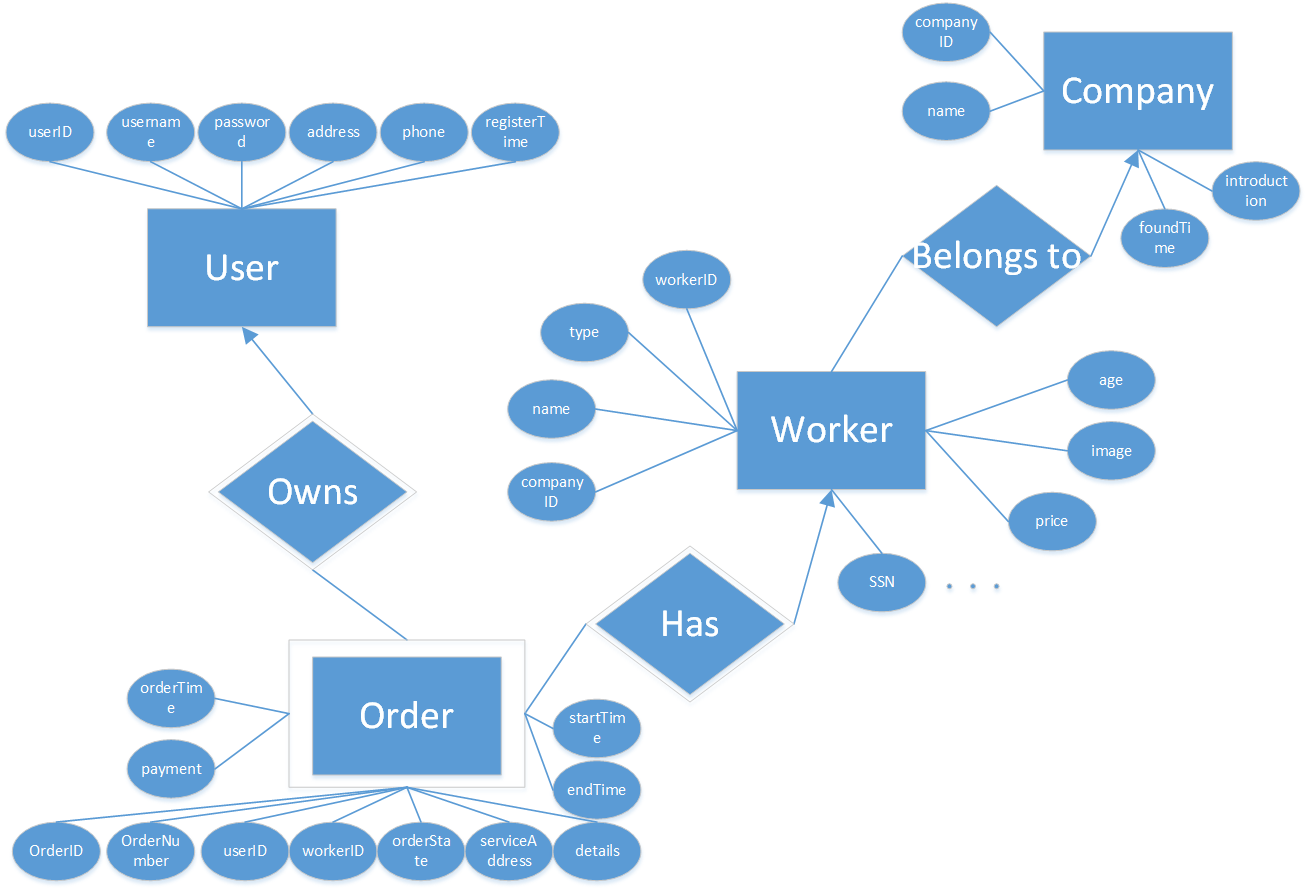
orderID is the primarykey for table order, while userID and workerID are the foreign keys.

Minimal set of FD/MVDs that cover all FD/MVDs in as few FD/MVDs as possible listed below:

SSN → name, companyID, SSN, nationality, gender

orderID → orderState, serviceAddress, details, orderTime, payment, startDate, endDate

**ER Design**



**SQL Script**

Based upon our ER Design, SQL script is generated for proposed database in MySQL. Our data types and constraints vary based on the MySQL DBMS. Include our SQL script the following:

* SQL commands for CREATE TABLE

CREATE SCHEMA `545pjcthskp` ;

CREATE TABLE `545pjcthskp`.`user` (

`userID` INT NOT NULL AUTO\_INCREMENT COMMENT '',

`username` VARCHAR(45) NOT NULL COMMENT '',

`password` VARCHAR(45) NOT NULL COMMENT '',

`address` VARCHAR(100) NULL COMMENT '',

`phone` VARCHAR(45) NULL COMMENT '',

`registerTime` DATETIME NOT NULL COMMENT '',

PRIMARY KEY (`userID`) COMMENT '');

CREATE TABLE `545pjcthskp`.`woker` (

`workerID` INT NOT NULL COMMENT '',

`type` VARCHAR(45) NOT NULL COMMENT '',

`name` VARCHAR(45) NOT NULL COMMENT '',

`companyID` INT NOT NULL COMMENT '',

`SSN` INT NOT NULL COMMENT '',

`image` VARCHAR(300) NULL COMMENT '',

`availableTime` VARCHAR(45) NULL COMMENT '',

`price` FLOAT NOT NULL COMMENT '',

`nationality` VARCHAR(45) NULL COMMENT '',

`introduction` VARCHAR(45) NULL COMMENT '',

`fitness` VARCHAR(45) NULL COMMENT '',

`gender` VARCHAR(45) NULL COMMENT '',

`age` INT NULL COMMENT '',

`hasMarried` INT NULL COMMENT '',

`language` VARCHAR(45) NULL COMMENT '',

`hasCertificate` INT NULL COMMENT '',

`evaluation` VARCHAR(45) NULL COMMENT '',

PRIMARY KEY (`workerID`) COMMENT '');

CREATE TABLE `545pjcthskp`.`company` (

`companyID` INT NOT NULL COMMENT '',

`name` VARCHAR(45) NOT NULL COMMENT '',

`foundTime` DATETIME NULL COMMENT '',

`introduction` VARCHAR(300) NULL COMMENT '',

PRIMARY KEY (`companyID`) COMMENT '');

CREATE TABLE `545pjcthskp`.`order` (

`orderID` INT NOT NULL AUTO\_INCREMENT COMMENT '',

`orderNumber` INT NOT NULL COMMENT '',

`userID` INT NOT NULL COMMENT '',

`workerID` INT NOT NULL COMMENT '',

`orderState` VARCHAR(45) NOT NULL COMMENT '',

`serviceAddress` VARCHAR(45) NOT NULL COMMENT '',

`details` VARCHAR(45) NULL COMMENT '',

`orderTime` DATETIME NOT NULL COMMENT '',

`payment` FLOAT NOT NULL COMMENT '',

`starDate` DATETIME NOT NULL COMMENT '',

`endDate` DATETIME NOT NULL COMMENT '',

PRIMARY KEY (`orderID`) COMMENT '');

* Foreign keys and referential integrity actions

ALTER TABLE `545pjcthskp`.`order`

ADD INDEX `userID\_idx` (`userID` ASC) COMMENT '',

ADD INDEX `workerID\_idx` (`workerID` ASC) COMMENT '';

ALTER TABLE `545pjcthskp`.`order`

ADD CONSTRAINT `userID`

FOREIGN KEY (`userID`)

REFERENCES `545pjcthskp`.`user` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

ADD CONSTRAINT `workerID`

FOREIGN KEY (`workerID`)

REFERENCES `545pjcthskp`.`woker` (`workerID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT;

ALTER TABLE `545pjcthskp`.`woker`

ADD INDEX `companyID\_idx` (`companyID` ASC) COMMENT '';

ALTER TABLE `545pjcthskp`.`woker`

ADD CONSTRAINT `companyID`

FOREIGN KEY (`companyID`)

REFERENCES `545pjcthskp`.`company` (`companyID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT;

* SQL commands to enter data into the table.

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('1', 'salianu', 'bty8life', '1717 market street, seattle, ', '2062347892', '2010/10/20');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('2', 'bowenrock', '911217', '\"7432 Ivy Lane \nGlen Ellyn, seatac 60137\"', '20612345678', '2011/8/23');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('3', 'bjqiang8', '913269', '\"3177 8th Street West \nPoint Pleasant Beach, lynnwood 08742\"', '2534567890', '2006/7/4');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('4', 'szh', '68297730zsz', '\"8922 Creek Road \nFort Worth, lakewood 76110\"', '2536127788', '2014/12/1');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('5', 'tb45592', 'zwd19911217', '1717 commerse street, aubern', '2534043371', '2001/1/3');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('6', 'abouterwave', 'guapo', '\"483 Lafayette Street \nGermantown, federal way 20874\"', '2025550129', '1998/12/20');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('7', 'danella', 'guapa', '\"957 13th Street \nPhillipsburg, burien 08865\"', '2023450156', '1997/1/2');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('8', 'nycfree', 'alto', '\"73 Ashley Court \nOxford, kirkland 3865\"', '2790218543', '2008/9/30');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('9', 'jocapo', 'alta', '\"725 Smith Street \nCircle Pines, tacoma 55014\"', '3985782345', '2004/11/21');

INSERT INTO `545pjcthskp`.`user` (`userID`, `username`, `password`, `address`, `phone`, `registerTime`) VALUES ('10', 'pounboxp', 'latinaughs', '\"792 Brandywine Drive \nVienna, university place 22180\"', '5550129824', '2001/2/3');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('1', 'West Seattle Housekeeping', '1991-10-25',

'Small, local, eco-friendly, woman-owned housekeeping service based in

and serving West Seattle.');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('2', 'Molly Maid Seattle Eastside', '1998-05-

06', 'All services are 100% guaranteed. Your satisfaction is our number

one goal; if for any reason you aren\'t happy with our professional

maid services, we will come back and clean the specific areas that

didn\'t meet your expectations.');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('3', 'Fivestar-Housekeeping', '2001-02-07',

'Five Star Housekeeping specialize on have the costumers happy, that is

our first goal, and how we can have them happy? ');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('4', 'Lovely Housekeeping', '2010-09-01',

'Lovely House did a very thorough cleaning of our apartment and several

other apartments that we managed.');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('5', 'The Maids', '1999-01-01', 'You\'ll be

surprised at the free time you\'ll gain allowing you to explore the

city with home cleaning services from The Maids. ');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('6', 'THA House Cleaning and Housekeeping',

'2013-02-02', 'The THA crew has all right tools to make your home

clean,sparkling and refreshed. ');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('7', 'Green Wagon Cleaning, LLC', '2010-10-02',

'Green Wagon Cleaning has acquired Green Glove Cleaning! We are adding

3 new employees and 24 fabulous clients to our roster! We are very

excited for this growth opportunity!');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('8', 'Adrianne\'s Housekeeping', '2013-09-06',

'You can say \"so long\" to dirt and grime when you hire the cleaning

service professionals from Adrianne\'s Housekeeping. ');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('9', 'Simply Clean', '1997-10-05', 'Simply

Clean has been a premier Seattle cleaning company since 2009, servicing

all of King County for over 5 years. And while we’ve only been in

Seattle since ’09, our experience in house cleaning extends much

farther back. ');

INSERT INTO `545pjcthskp`.`company` (`companyID`, `name`, `foundTime`,

`introduction`) VALUES ('10', 'Dana\'s Housekeeping', '2003-03-03',

'Let us help you find the perfect professional housekeeper.');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('1', ' Housekeeper', 'Adelaide', '3', '876374812', '\image\head1.jpg', '9:00-13:00', '15', 'China', 'Hi, everyone', 'Good', 'Female', '27', '1', 'Chinese, English', '1', '5');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('2', ' Housekeeper', 'Carmen', '2', '183747284', '\image\head2.jpg', '13:00-22:00', '35', 'USA', 'Hi, everyone', 'Good', 'Female', '37', '1', 'English', '1', '5');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('3', ' Tutor', 'Carole', '4', '123435465', '\image\head3.jpg', '16:00-22:00', '22', 'USA', 'Hi, everyone', 'Good', 'Female', '42', '1', 'English', '1', '5');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('4', ' Babysiter', 'Daphne', '7', '674536543', '\image\head4.jpg', '5:00-10:00', '40', 'Philippines', 'Hi, everyone', 'Good', 'Female', '41', '1', 'English, Philipino', '1', '4');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('5', ' Tutor', 'Ellen', '7', '783920341', '\image\head5.jpg', '7:00-14:00', '60', 'Philippines ', 'Hi, everyone', 'Good', 'Female', '35', '0', 'English,Philipino', '1', '3');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('6', ' Housekeeper', 'Felicia', '9', '234637543', '\image\head6.jpg', '8:00-17:00', '38', 'China', 'Hi, everyone', 'Good', 'Female', '39', '1', 'English,Chinese', '0', '4');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('7', ' Tutor', 'Hilda', '8', '123332778', '\image\head7.jpg', '8:00-17:00', '18', 'USA', 'Hi, everyone', 'Good', 'Female', '36', '1', 'English', '1', '5');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('8', ' Housekeeper', 'Katherine', '3', '888955676', '\image\head8.jpg', '9:00-17:00', '26', 'India', 'Hi, everyone', 'Good', 'Female', '29', '0', 'Indian,English', '0', '5');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('9', ' Babysiter ', 'Leona', '1', '334587692', '\image\head9.jpg', '9:00-17:00', '32', 'USA', 'Hi, everyone', 'Good', 'Female', '37', '1', 'English', '0', '4');

INSERT INTO `545pjcthskp`.`woker` (`workerID`, `type`, `name`, `companyID`, `SSN`, `image`, `availableTime`, `price`, `nationality`, `introduction`, `fitness`, `gender`, `age`, `hasMarried`, `language`, `hasCertificate`, `evaluation`) VALUES ('10', ' Babysiter', 'Madeline', '5', '190009835', '\image\head10.jpg', '12:00-17:00', '29', 'USA', 'Hi, everyone', 'Good', 'Female', '29', '1', 'English', '1', '5');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('1', '10000', '6', '5', 'Paid', '\"483 Lafayette Street ', '2015-11-20', '50', '2015-11-20', '2015-11-21');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('2', '10001', '6', '7', 'Paid', '\"483 Lafayette Street ', '2015-11-20', '38', '2015-11-20', '2015-11-20');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('3', '10002', '4', '5', 'Paid', '\"8922 Creek Road ', '2015-11-20', '20', '2015-11-20', '2015-11-20');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('4', '10003', '3', '2', 'Paid', ' \"3177 8th Street West ', '2015-11-23', '21', '2015-11-23', '2015-11-23');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('5', '10004', '8', '5', 'Paid', '\"73 Ashley Court ', '2015-11-23', '29', '2015-11-23', '2015-11-24');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('6', '10005', '8', '4', 'Paid', '\"73 Ashley Court ', '2015-11-18', '30', '2015-11-19', '2015-11-10');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('7', '10006', '7', '3', 'NotPaid', '\"957 13th Street ', '2015-11-18', '18', '2015-11-19', '2015-11-10');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('8', '10007', '1', '8', 'NotPaid', '1717 market street, seattle, ', '2015-11-18', '18', '2015-11-18', '2015-11-18');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('9', '10008', '2', '9', 'Finished', '\"7432 Ivy Lane ', '2015-11-26', '18', '2015-11-26', '2015-11-26');

INSERT INTO `545pjcthskp`.`order` (`orderID`, `orderNumber`, `userID`, `workerID`, `orderState`, `serviceAddress`, `ordeTime`, `payment`, `starDate`, `endDate`) VALUES ('10', '10009', '2', '1', 'Finished', '\"7432 Ivy Lane ', '2015-11-26', '21', '2015-11-26', '2015-11-26');

**Implemented all these fundamental functions in front end, 9 in total:**

* Adding a new user to the system

Boolean addUser(User \_newUser);

* Check the username and password

User verifyUser(String username, String password);

* Searching workers based on user preferences, including age, price, gender

List<Worker> getWorkersByConditions(String workType, String age, String gender, String price, String workTime);

* Showing all workers of particular type

List<Worker> getWorkerByType(String workerType);

* Adding a new order

Boolean addNewOrder(Order newOrder);

* Searching orders by a particular user

List<Order> getOrderByUserID(int userID);

* Searching the worker of a particular order

Worker getWorkerByOrderID(int orderID);

* Searching company of a worker

Company getCompanyByWorkerID(int workerID);

* Update the state of order(e.g. when the order is paid, change the status of order into “During Service”)

Boolean updateOrderState(int orderID, String newStatus);

**Testing**

Test code extensively with 5 test cases to make sure it works fine.

**test case 1**: assume a new user who haven’t registered before would like to complete the whole process from registration to order completed.

Registration name: “Wandi22”

Registration password: “123”

Registration address: “Tacoma”

Registration phone: “123456789”

Service type looking for: “Tutor”

Preference: price below 20, female, age 20-29

Date and time preferred: From 2015/12/15 To 2015/12/16 (2days)

Pay immediately: Yes

Expecting result:

From frontend end, successfully login with `username`(“Wandi”) and `pwd`(“123”), can search based on given reference and place order successfully, could see its own order from its account with order status ‘Paid’.

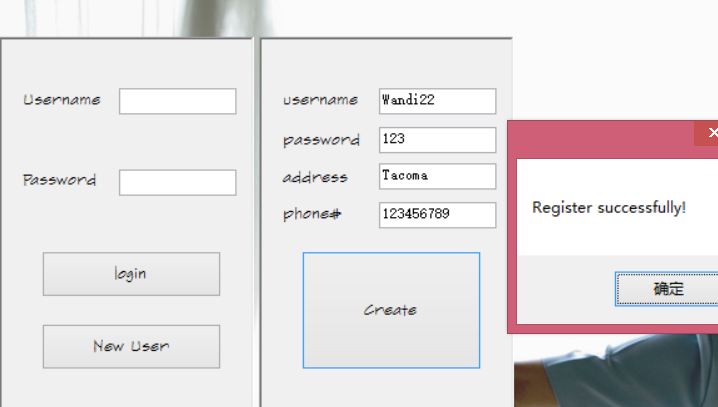
From back end, two new records inserted to table *User* and *Order* successfully. With user-details userID(*Auto Increment*), `username` (“Wandi”), `password`(“123”) `address` (“Tacoma”) `phone` (“123456789”) `registerTime` (“2015/12/15”), and `orderID`( *Auto Increment*)

`orderNumber` (*Randomly Generated*) `userID`(*Current UserID*) `workerID`(*Selected WorkerID*) `orderState` (“Paid”) `serviceAddress` (“Tacoma”) `details` (null) `orderTime` (2015/12/15) `payment` () `starDate` () `endDate` ().

Result get:

* Registeration:

In frontend:



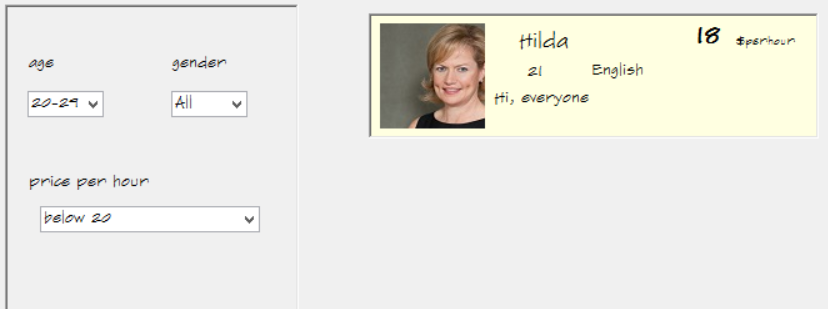
In backend:



* Choose service Type



* Search for workers based on preference, and choose the worker

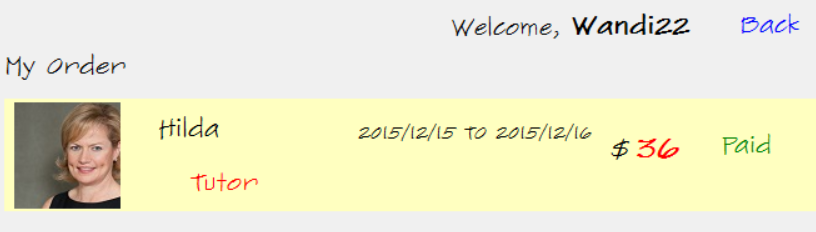


* Confirm the order information and schedule the service time



* Click “Pay Now” button

In frontend:



In backend:

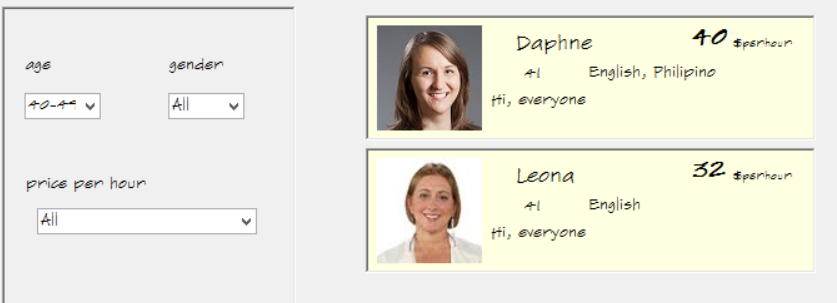


**test case 2**: Assume a registered user would like to search workers based on his/her preference. He/she can have multiple preferred conditions: age, gender and price. When the user has chosen one of the conditions and continues to choose other conditions, the user must assume the chosen condition remain to be chosen. So the final searching result must be based on multiple conditions chosen in different time.

Input operation1: choose the age from 40-49

Expecting result: show all the workers age from 40-49

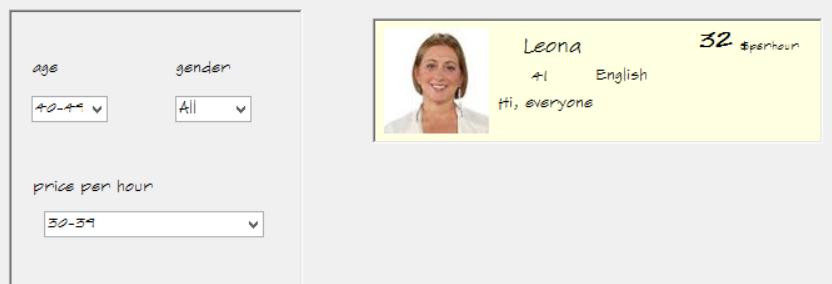
Result get:



Input operation2: choose the price from 30-39

Expecting result: show the workers age from 40-49 and have the price from 30-39

Result get:



**test case 3**: Assume a user want to manage his/her orders in the “myOrder” page, he can see all the orders he/she has. And he/she can pay for the order that has not been paid. The order state should be changed both in frontend and backend after the order is paid.

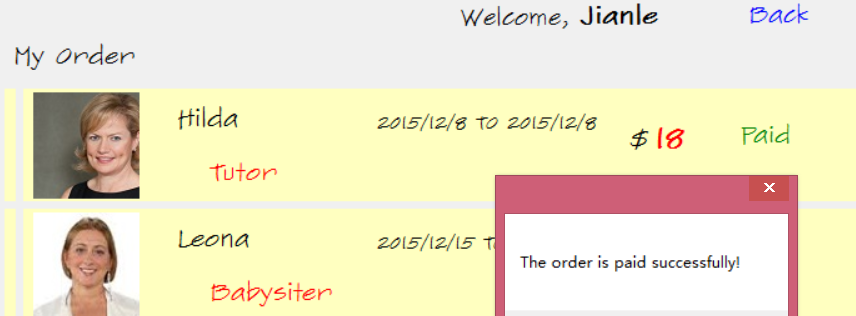
Input operation: click the “pay it now!” button for the tutor service which is served by “Hilda”.

Expecting result: In the frontend, show the order has been paid. In the backend, show the order status is changed to “Paid” in the database.



Result get:

In the frontend,



In the backend,



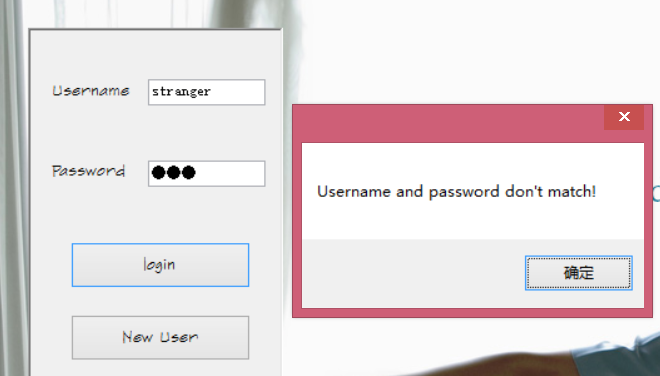
**test case 4**: Assume a user that has never registered before tries to login our system with unmatched username and password, we prevent this operation.

Input username: “stranger”

Input password: “123”

Expecting result: show warning message and prevent this login operation.

Result get:



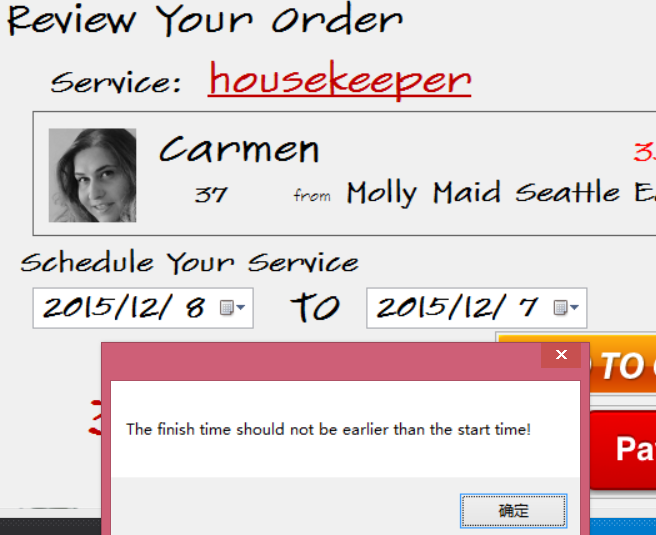
**test case 5**: When the user schedule his/her service, he/she might

* choose an end date is earlier than the start date, we avoid this situation.

Choose end date: 2015/12/7(earlier than start date 2015/12/8).

Expecting result: show warning message.

Result get:

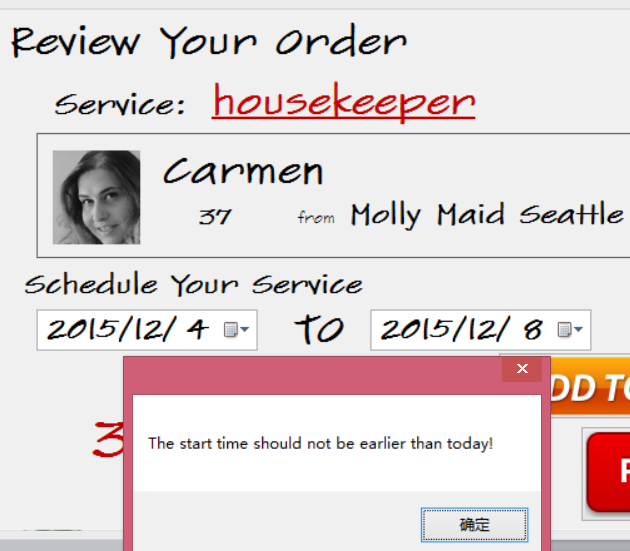


* choose an start date earlier than today, we avoid this situation.

Choose start date: 2015/12/4(earlier than today 2015/12/8).

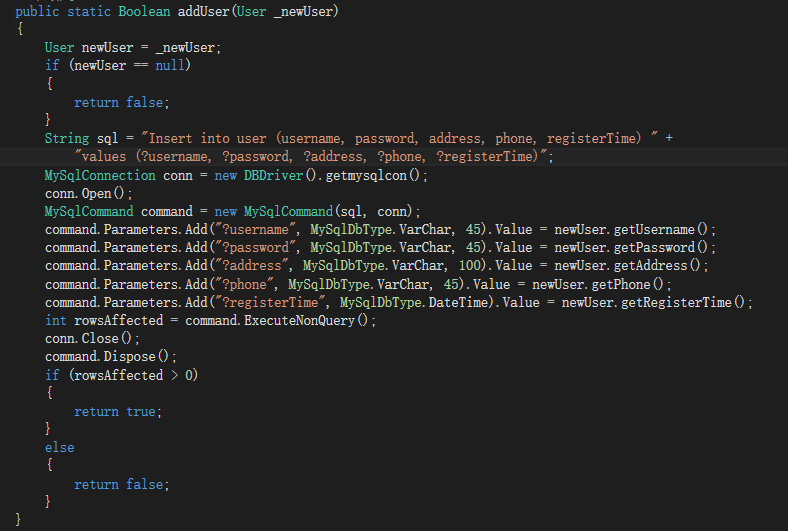
Expecting result: show warning message.

Result get:

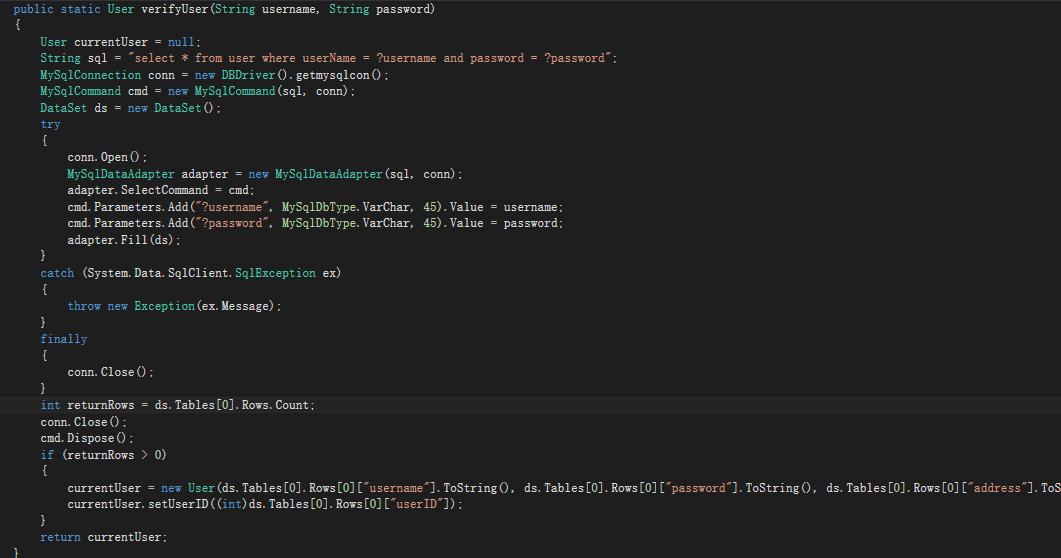


**Code Snippet**

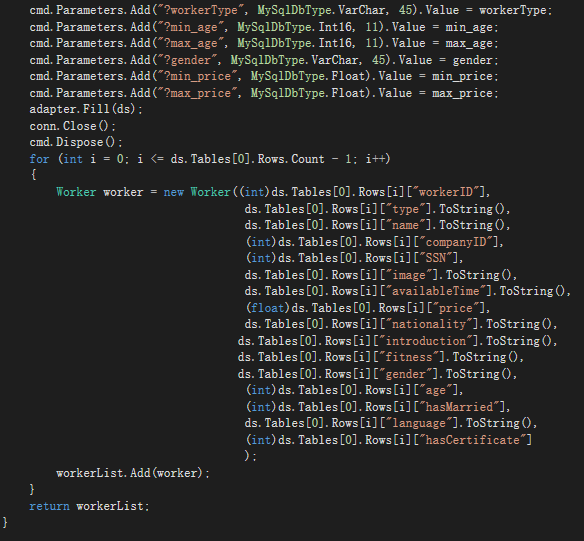
* Adding a new user to the system



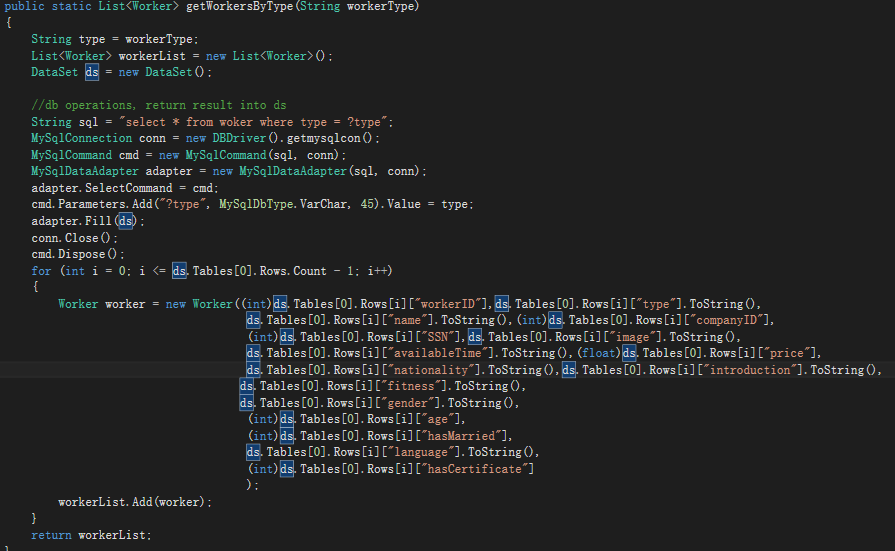
* Check the username and password



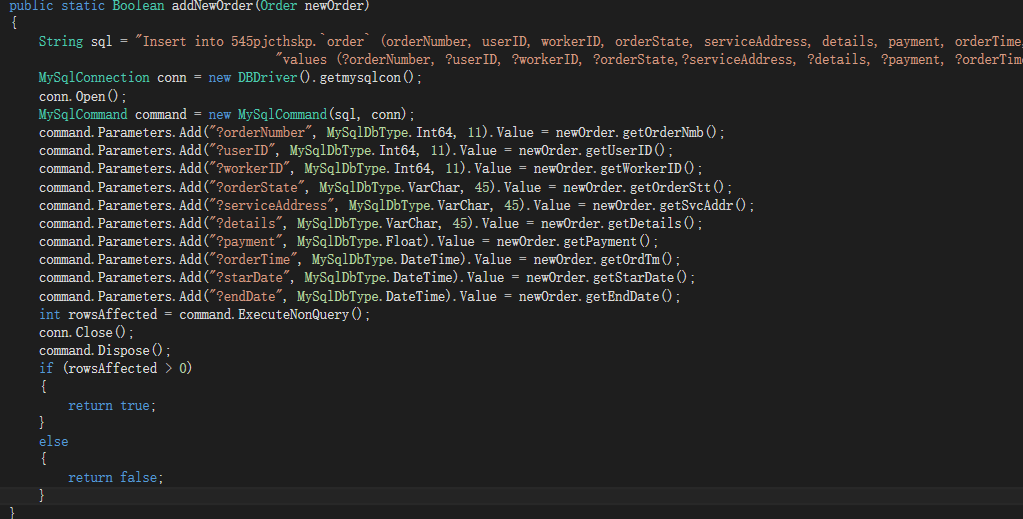
* Searching workers based on user preferences, including age, price, gender

* Showing all workers of particular type



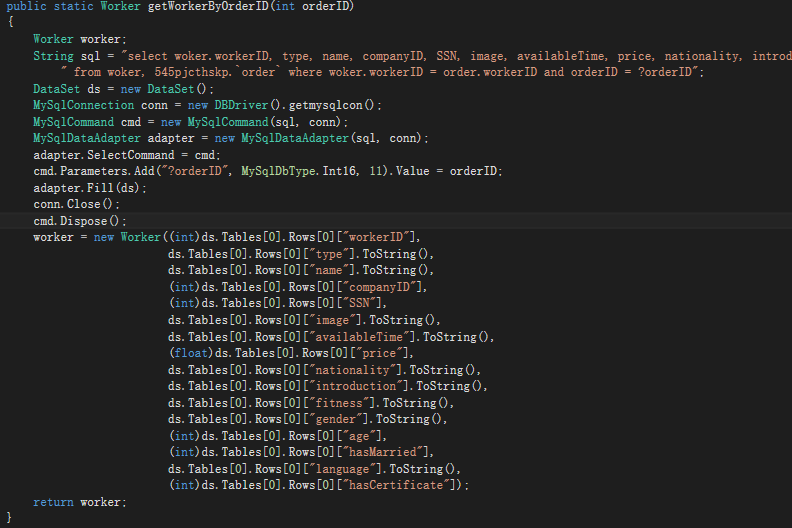
* Adding a new order



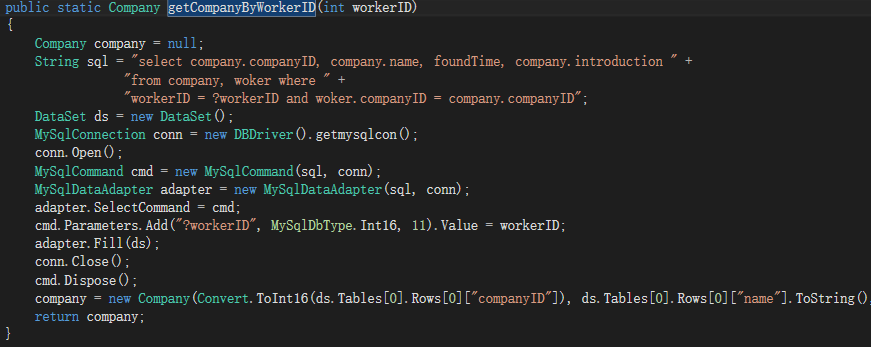
* Searching orders by a particular user

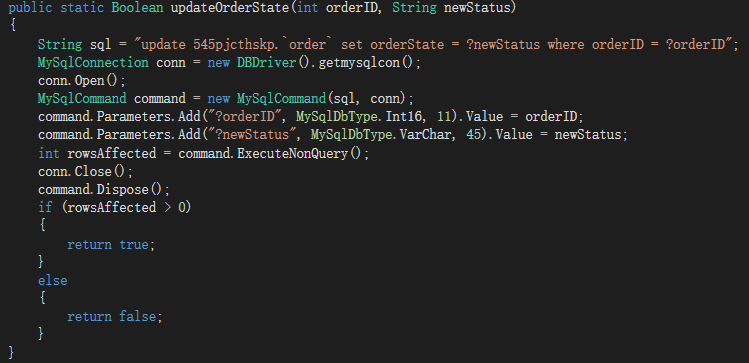


* Searching the worker of a particular order



* Searching company of a worker



* Update the state of order****