We use mysql and Java in this project,so all the following SQL's are in mysql syntax. All heighted variables represent the dynamically given values in the application.

The followings are the sql's included in the specs:

Insert new reservation by customer:

INSERT INTO Restaurant (address,rname,resscore,resphone,ownerid) VALUES (?, ?, ?, ?, ?)

Owner can update the food menu:

UPDATE Food SET price = p, fname = f WHERE foodid = id

Owner can find all the customer names who have written review for the restaurant (join 3 tables)

SELECT cname FROM Customer c, Review r, Restaurant res

WHERE c.customerphone = r.customerphone AND r.resphone = res.resphone AND res.ownerid = oid

Owner should be able to get the total price of all the reserved food from a reservation (join 2 tables)

SELECT * FROM Food f, Contain c

WHERE f.foodid = c.foodid AND c.reservationid = reserveid

Owner should be able to read all the reviews given owner ID (join 2 tables)

SELECT * FROM Review rev, Restaurant res

WHERE rev.resphone = res.resphone AND res.ownerid = oid

Owner wants to find the most popular food in his/her restaurant (use group by FoodID)

SELECT f.foodid, f.fname, COUNT(c.reservationid) AS Popularity FROM Contain c, Food f

WHERE c.reservationid IN (SELECT r.reservationid FROM Reservation r where resphone = resp)

AND c.foodid = f.foodid

GROUP BY f.foodid, f.fname

ORDER BY popularity DESC

Member can write review for the restaurant:

INSERT INTO Review (reviewid,resphone,customerphone,content,reviewscore) VALUES (?, ?, ?, ?, ?)

Owner can add new food:

INSERT INTO Food (foodid,price,fname,resphone) VALUES (?, ?, ?, ?)

Customer should be able to see name, address, phone number and score of all restaurants

CREATE VIEW V_Customer AS

SELECT rname, address, resphone, resscore

FROM Restaurant

The remaining are queries in the source that are not in the specs:

Member can delete review:

DELETE FROM Review WHERE reviewid = ?

Owner can see all his/her restaurants:

SELECT * FROM Restaurant where ownerid = oid

Owner can see one specific his/her restaurants (given restaurant phone number):

SELECT * FROM Restaurant where resphone = resp

Owner can see all reservations of one of his/her restaurant (given restaurant phone number):

SELECT * FROM Reservation where resphone = resp

Owner can see all reviews of one of his/her restaurant (given restaurant phone number):

SELECT * FROM Review where resphone = resp

Owner can see all food (menu) of one of his/her restaurant (given restaurant phone number):

SELECT * FROM Food where resphone = resp

Insert a new customer when his/her is making a reservation:

INSERT INTO Customer (customerphone,cname) VALUES (?, ?)

Insert a new reservation:

INSERT INTO Reservation (reservationid,reservationdate,numberofcustomer,customerphone,resphone,cname,ownerid) VALUES (?,?,?,?,?,?,?)

Insert an foodid and reservationid pair when the food is ordered in the reservation:

INSERT INTO Contain (foodid,reservationid) VALUES (?, ?)

Insert a new member if his/her registers:

INSERT INTO Member (customerphone,points,email,cname) VALUES (?, ?, ?, ?)

Get all information of a member:

SELECT * FROM Member WHERE customerphone = cphone

Update the member's point when a new reservation is made:

UPDATE Member SET points = ? WHERE customerphone = ?

Member can update his/her email:

UPDATE Member SET email = ? WHERE customerphone = ?

Customers and Members can see all restaurant once login:

SELECT * FROM Restaurant