**Cuisine A La Carte**

**Restaurant Management System**

**Design Report**

**Online Restaurant System**

**By**

**Natalia Harrow**

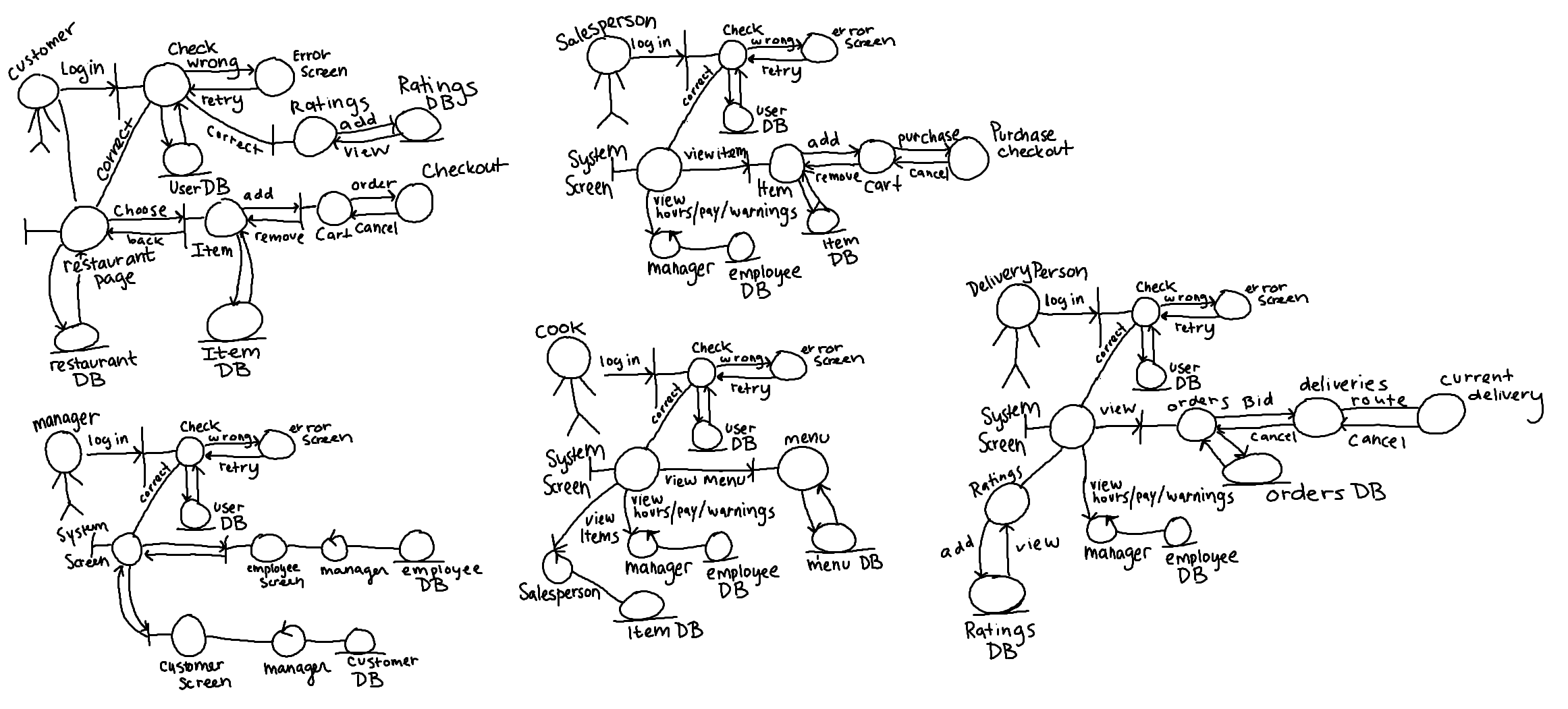
**Raul Meruelo**

**Terri Thampan**

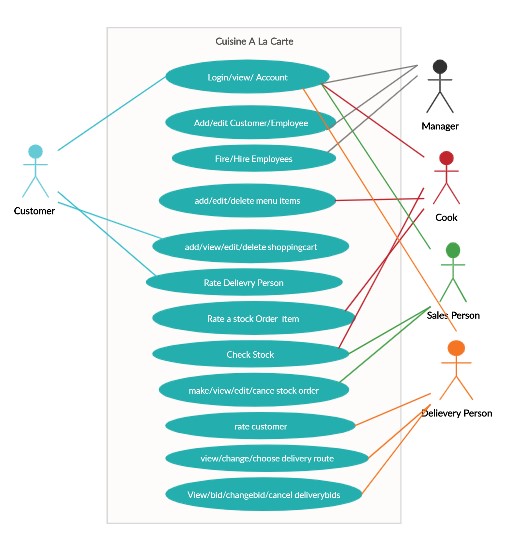
**Nick Weatherley**

1. **Introduction**

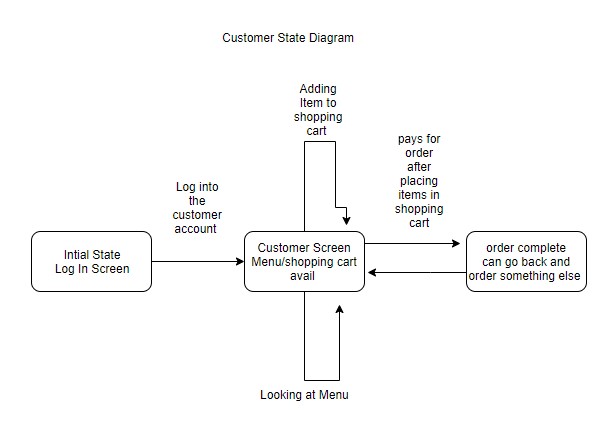
The collaboration class diagram for the whole system is shown below. Each individual class diagram is shown in part 2, below the use case diagram.

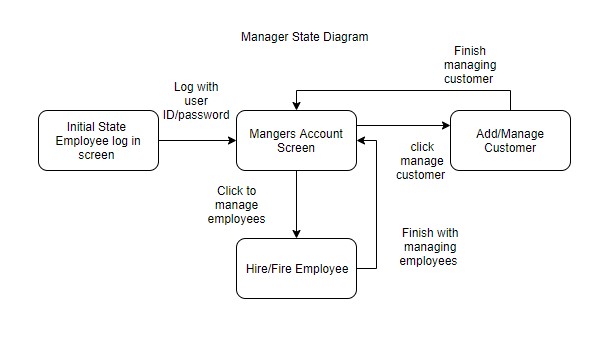


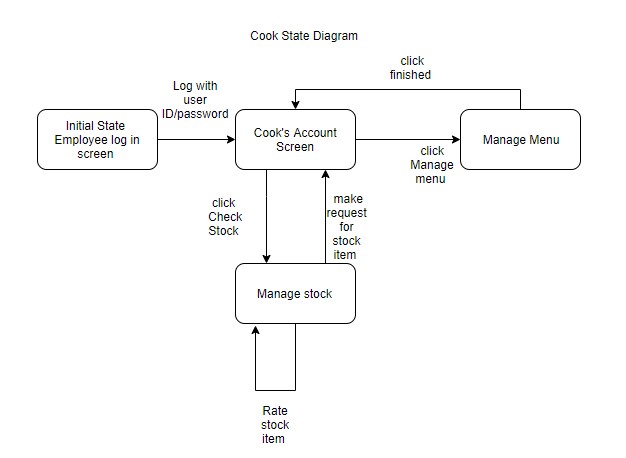
1. **All Use Cases**

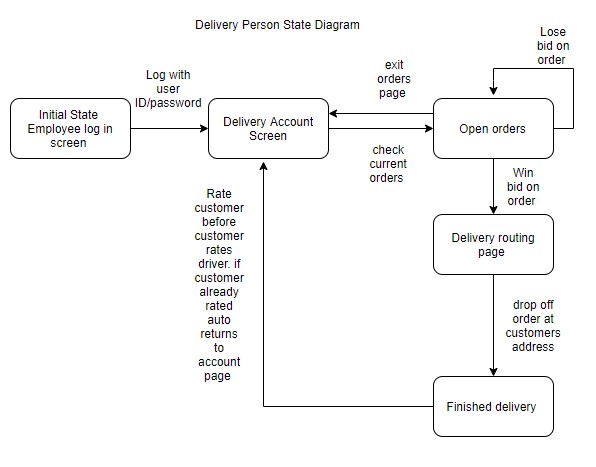
****

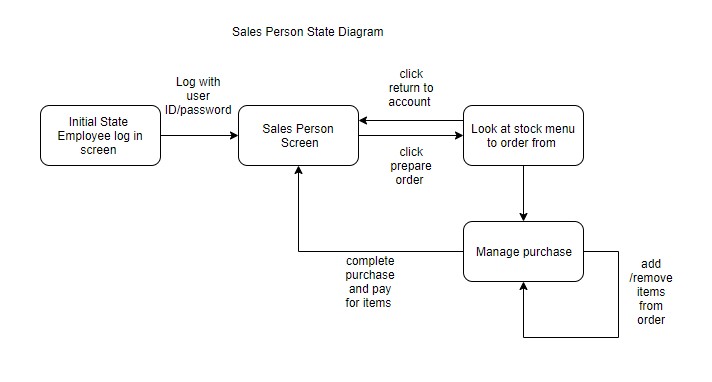
The use case diagram is shown above. The collaboration class diagrams for each user are as below in the order of: Customer, Manager, Salesperson, Cook, and finally Delivery Person.

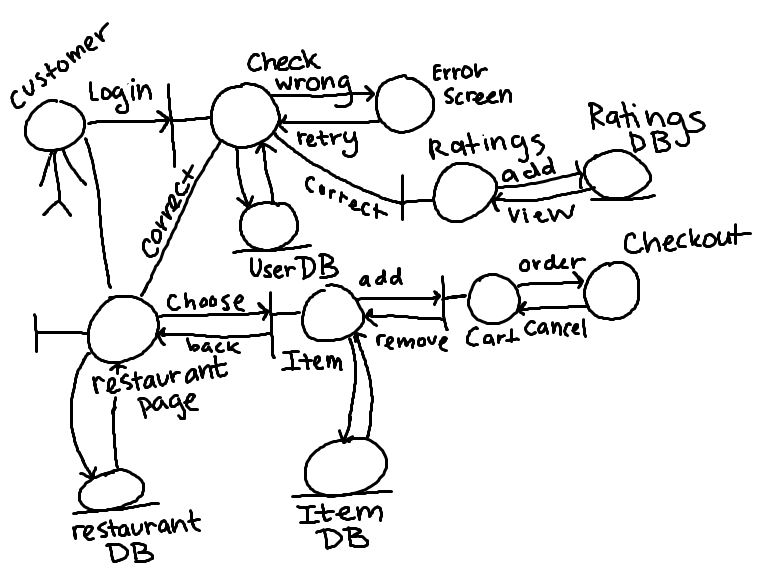


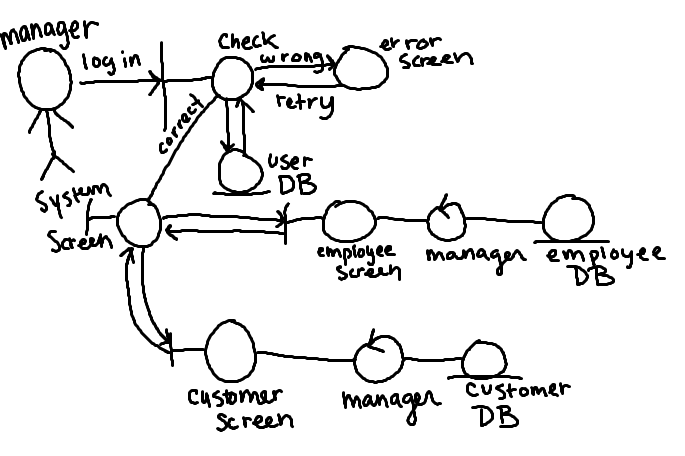


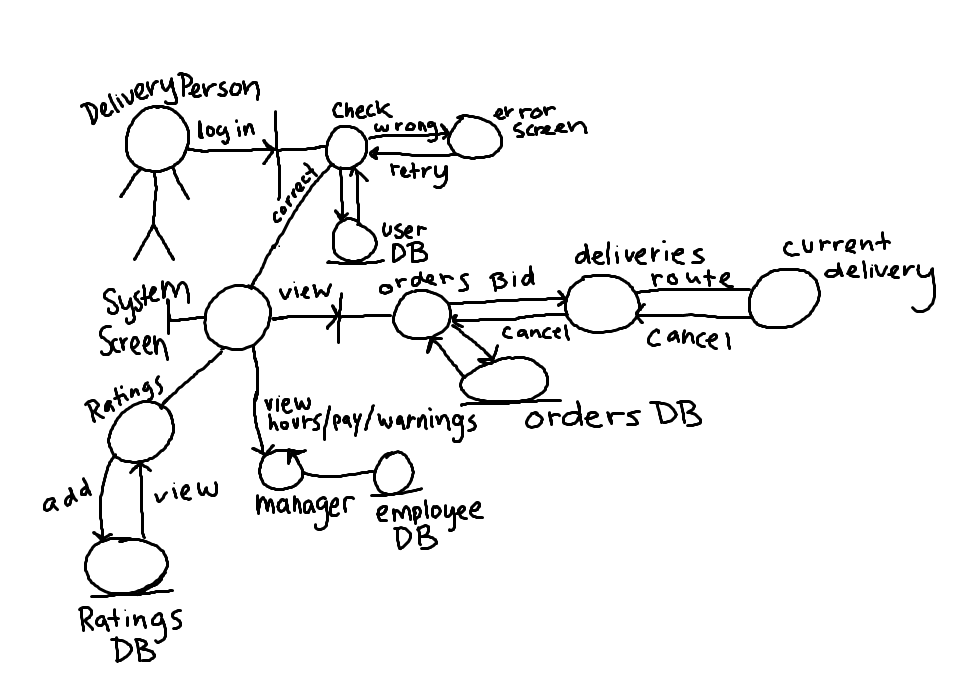
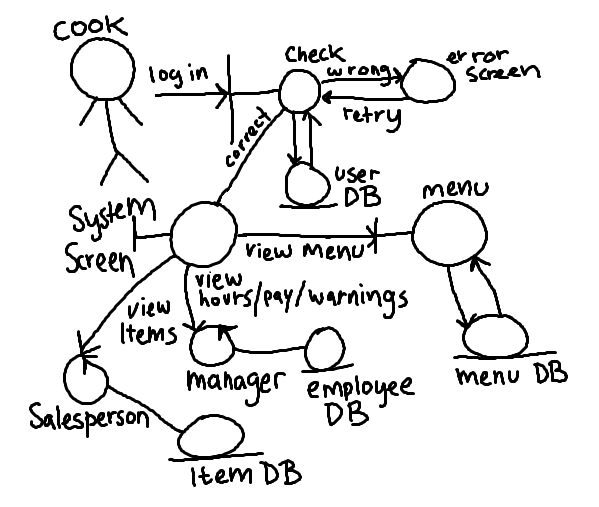
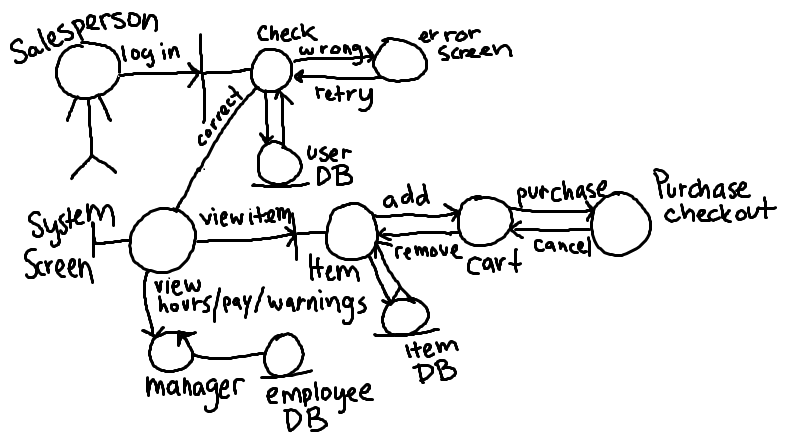




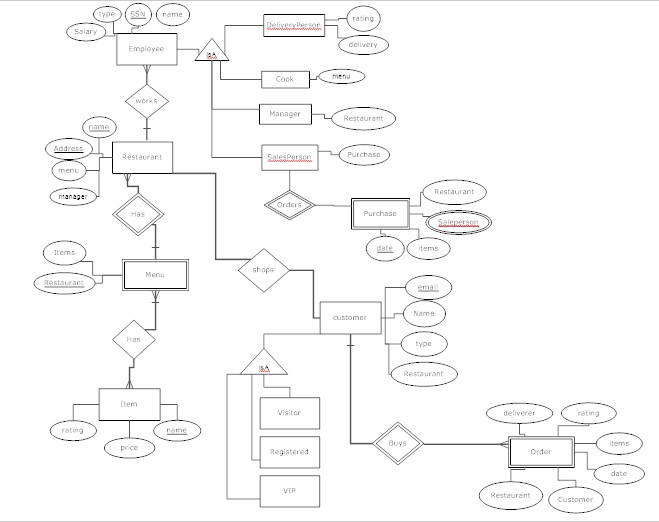




****

****

1. **E-R Diagrams for Entire System**

****

1. **Detailed Design**

The management system is being created using Django and Python. The pseudo code for each of the apps will be shown below.

One of the apps is the customer app which lets customers to view and modify their shopping carts, rate food and delivery people, and login or create accounts.

function viewShoppingCart(Customer customer)

if customer is not null

get the shopping cart that is assigned to customer

return the list of items in the shopping cart or empty list if nothing is in the cart

function addItemToShoppingCart(Customer customer, Item item)

get the shopping cart assigned to customer

get the list of items in the shopping cart

add item to the list

return true if added and false otherwise

function editShoppingCart(Customer customer, Action action, Item item)

get the shopping cart that is assigned to customer

get the list of items in the shopping cart

get the specified item

if action == addItem

increase the count of the item

else if action == removeItem

decrease the count of the item

return modified list of items in the shopping cart

function removeItemFromShoppingCart(Customer customer, Item item)

get the shopping cart that is assigned to customer

get the list of items in the shopping cart

get the specified item

if item is in the shopping cart

remove the item from the list

else

return false

return modified list of items in the shopping cart

function rateFood(Customer customer, Item item)

get customer's order history to check if the customer ordered the item

if customer ordered and received the item

get item's rating (list)

add customer's rating to the list

update the rating

else

return false - customer cannot rate item their didn't order

return new rating

function login(Customer customer, Password password)

get the list of registered users

if customer found in the registered users list

if password == to the password in the database

give user permission to access the site

else

ask to retype the password

else

return false - customer cannot login because they're not registered

function viewAccountInfo(Customer customer)

if customer is logged in

return their profile

else

return false - customer is not logged in

Another app is the Cook app, which allows cooks to check on stock, request more stock if there is an insufficient amount, and edit the items on the menu for the restaurants they are employed by.

function addMenuItem(list of items, menu item name, restaurant)

check list of items != 0

check name is != null and name is not empty

check restaurant != null

create new menu item w/ list of items

add menu item to restaurant menu

function changeMenuItems(list of items, menu item name, restaurant)

check list of items != 0

check name is != null and name is not empty

check restaurant != null

set items on menu item to items

function deleteMenuItem(menu item name, restaurant)

check name is != null and name is not empty

check restaurant != null

if menuItem is in restaurant.menu:

delete menu item

return True

else

return False

function checkStock(item, restaurant)

check item != null

check restaurant != null

if item is in restaurant.items

return remaining item count

else

return -1

function requestStock(item, restaurant)

check item != null

check restaurant != null

if item is in restaurant.items

send notification to salespeople who work at restaurant to purchase more

else

log error to log

function login(cook, Password password, restaurant)

get the list of registered cooks for restaurant

if cook found in the registered cook list

if password == to the password in the database

give user permission to access the site

else

ask to retype the password

else

return false

There is also the Delivery Person app, which gives delivery people the ability to view open orders, bid on those orders, choose the route for their deliveries, cancel their deliveries, and also rate customers.

function viewDelivery(delivery)

if delivery is in self.employedRestaurant

if delivery.byWho == self

return delivery

print not found error

function bidOnDelivery(double bid, delivery)

if delivery is in list of self.employedRestaurant.deliveries

if delivery.isOpenToBids

add bid to list of delivery.isOpenToBids

return True -> Bid added

return False -> Error when adding bidOnDelivery

function cancelDelivery(delivery)

if delivery is in self.employedRestaurant

if delivery.byWho == self and delivery.status == inprogress

set delivery status to 'CANCELED'

return True

return False

function viewDeliveryRoute(delivery)

if delivery is in self.employedRestaurant

if delivery.byWho == self

return delivery.route

print not found error

function changeDeliveryRoute(new route, delivery)

if delivery is in self.employedRestaurant

if delivery.byWho == self and delivery.status == inprogress

if route.destination == delivery.route.destination

delivery.route == route

return True

else

print Incorrect route destination

return False

function chooseRoute(delivery)

if delivery is in self.employedRestaurant

if delivery.byWho == self and delivery.status == inprogress

return list of routes w/ destination == delivery.customer.address

print delivery not found error

function rateCustomer(delivery, int rating)

if delivery is in self.employedRestaurant

if delivery.byWho == self and delivery.status == completed

add new rating to delivery.customer.ratings

return True

return False

function login(deliveryperson, Password password, restaurant)

get the list of registered deliverypersons for restaurant

if deliveryperson found in the registered deliverypersons list

if password == to the password in the database

give user permission to access the site

else

ask to retype the password

else

return false

function viewAccountInfo(self)

if self.deliveryperson is logged in

return their profile

else

return false - deliveryperson is not logged in

The Salesperson app allows for salespeople to view stock and to also create, view, and delete orders.

function createNewOrder(item, restaurant, quantity of item)

check item, restaurant != null

check quantity >= 1

add purchase to salesperson's list of purchases

if item not in restaurant.item list

add item and quantity to restaurant.item list

else

increase quantity of item

function viewOrder(order number)

if order number is in restaurant.purchases

if purchase was by self.salesperson

return purchase

return None

function changeOrder(order number, item, quantity, restaurant)

if order number is in restaurant.purchases

if purchase was by self.salesperson

purchase = new purchase(item, quantity, restaurant, self)

return True

return False

function cancelOrder(order number)

if order number is in restaurant.purchases

if purchase was by self.salesperson

set purchase status to 'CANCELLED'

return True

return False

function login(salesperson, Password password, restaurant)

get the list of registered salespersons for restaurant

if salesperson found in the registered salespersons list

if password == to the password in the database

give user permission to access the site

else

ask to retype the password

else

return false

function viewAccountInfo(self)

if self.salesperson is logged in

return their profile

else

return false - salesperson is not logged in

function checkStock(item, restaurant)

check item != null

check restaurant != null

if item is in restaurant.items

return remaining item count

else

return -1

The Manager app allows the Manager/Superusers to view all the employees working, and lets the manager fire or hire employees. They can also view customers and edit customers, as well as view ratings.

function hireCandidate(Manager manager, Candidate candidate)

if manager is in self.manager list

if candidate.isApproved

if candidate.isCookCandidate

get the list of cooks

register the candidate as cook

add the candidate to the list

return the updated list of cooks

if candidate.isCookSalespersonCandidate

get the list of salespeople

register the candidate as salesperson

add the candidate to the list

return the updated list of salespeople

return False -> Error when hiring

function fireEmployee(Manager manager, Employee employee)

if manager is in self.manager list

if employee is not null

if employee.isCook

get the list of cooks

remove the cook

return updated list

if employee.isSalesperson

get the list of salespeople

remove the salesperson

return the updated list

return False -> Employee not found

function viewEmployees(Manager manager, Employee employee)

if manager is in self.manager list

if employee is not null

if employee.isCook

get the list of cooks

if employee.isSalesperson

get the list of salespeople

return False -> Employee not found

function editEmployee(Manager manager, Employee employee)

if manager is in self.manager list

if employee is not null

return edit page with editing options

return false -> cannot edit employee

function viewRatings(Manager manager, Ratings rating)

if manager is in self.manager list

if ratings is not null

get the rating lists

return False -> cannot access the ratings

function login(Manager manager, Password password)

get the list of registered managers

if manager found in the registered mananagers list

if password == to the password in the database

give manager permission to access the site

else

ask to retype the password

else

return manager cannot login because they're not registered

function viewAccountInfo(Manager manager)

if manager is logged in

return their profile

else

return false - manager is not logged in

function addCustomer(Manager manager, Customer customer)

if manager is in self.manager list

if customer is not null

get the list of customers

add customer to the list

return updated list

return false -> couldn't add customer

function editCustomer(Manager manager, Customer customer)

if manager is in self.manager list

if customer is not null

return edit page with editing options

return false -> cannot edit customer

function viewCustomer(Manager manager, Customer customer)

if manager is in self.manager list

if customer is not null

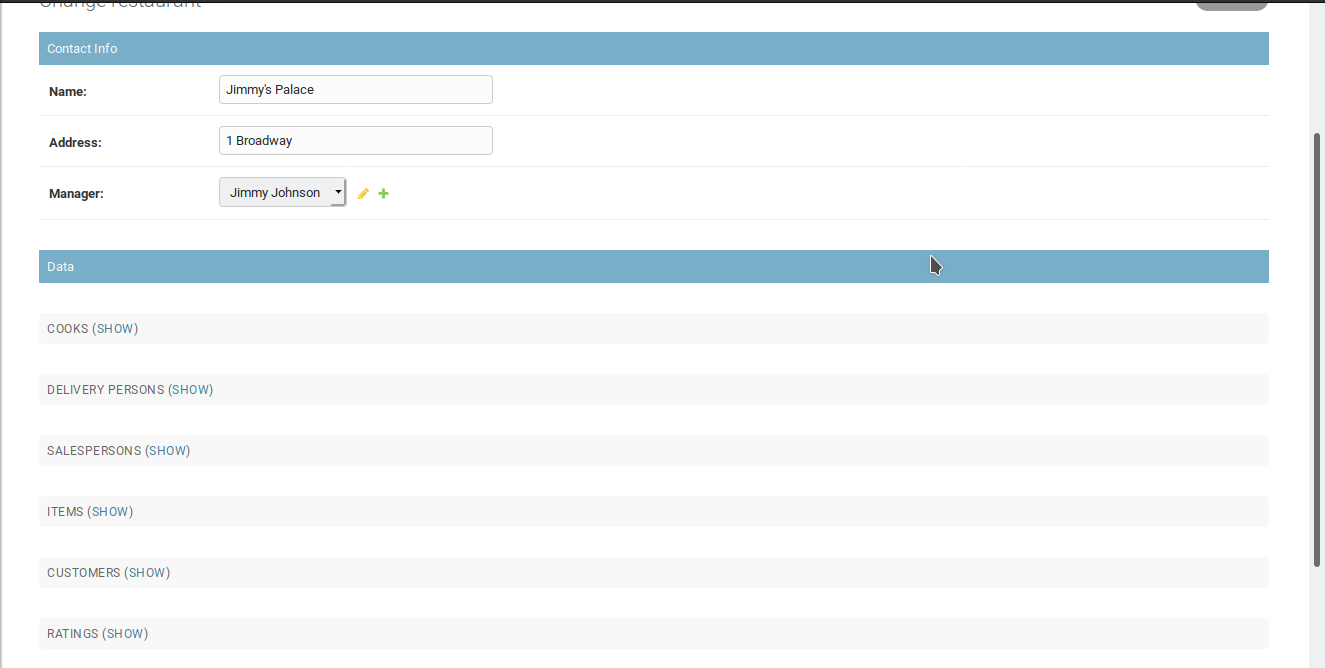
get the list of customers

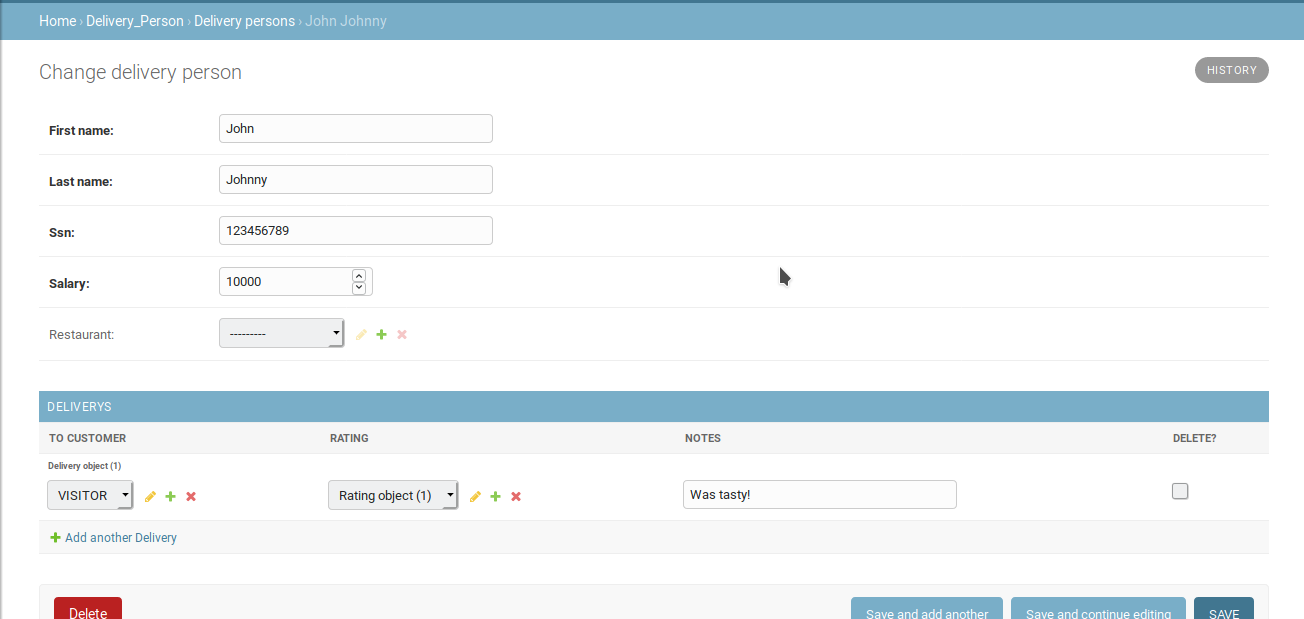
if customer is in the list

return customer's info

return False -> Customer not found

1. **System Screens**

****

****

1. **Group Minutes**

The team met for around a bit more than 25 hours total, or 1508 minutes. Team meetings were scheduled for multiple days in a row and lasted around an average of 4 hours, with the minimum amount of time met up being around an hour. Each member of the team contribute towards the project and there are no current concerns over teamwork.