Champion’s Slice System

Software Requirements Specification

Phase 2 Report

Aliaa Abdelrahman
Ismail Akram
Inna Baryanova
Kevin Peter
Version <1.1>

Revision History

Date	Version	Description	Author
October 20th, 2020	1.0	Create a software project specification.	Aliaa Abdelrahman Ismail Akram Inna Baryanova Kevin Peter Murjan Urmey
November 17th, 2020	1.1	Create a design report	Aliaa Abdelrahman Ismail Akram Inna Baryanova Kevin Peter

Table of Contents

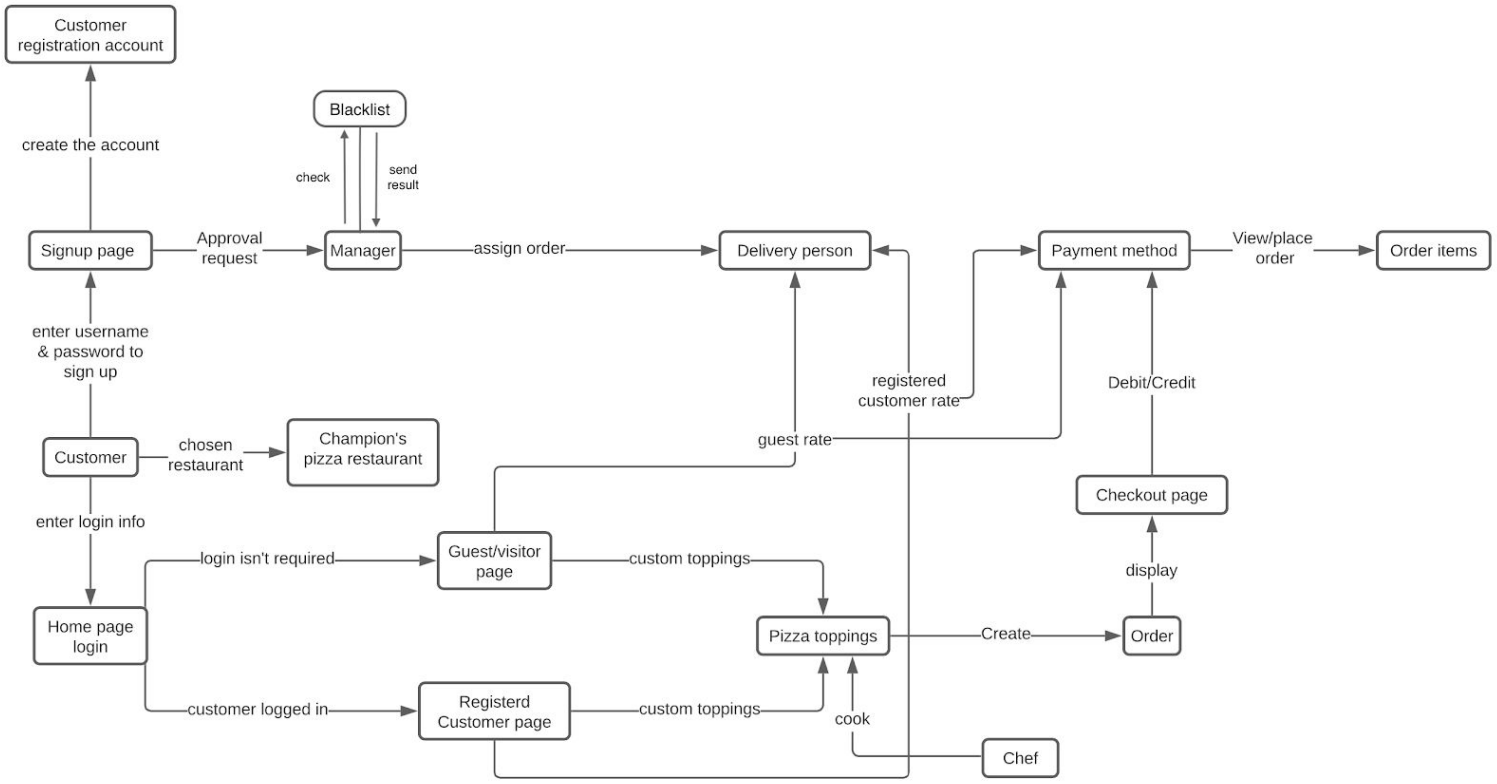
1. **Introduction**

This report is an overview of the pizza system design and functionality.

1.1 The **purpose** of this report is to explain the details and functionality of the system

1.2 **Collaboration class diagram:**

Collaboration diagram explains everything in class. It explains the behaviors of group objects which collaborate similarly. The diagram shows the functionalities and objects passed between classes. The diagram below shows how the overall system works.



2. **Use Case Analysis**

For this section, here's a more detailed overview of each main use cases specified in our specification report. Below is a collaboration class diagram showcasing the interaction between classes/objects in the system and a petri-net detailing the various processes involved in each use-case.

1. Register account
2. Login
3. Design pizza
4. Price tiers
5. Order submission
6. Rating system

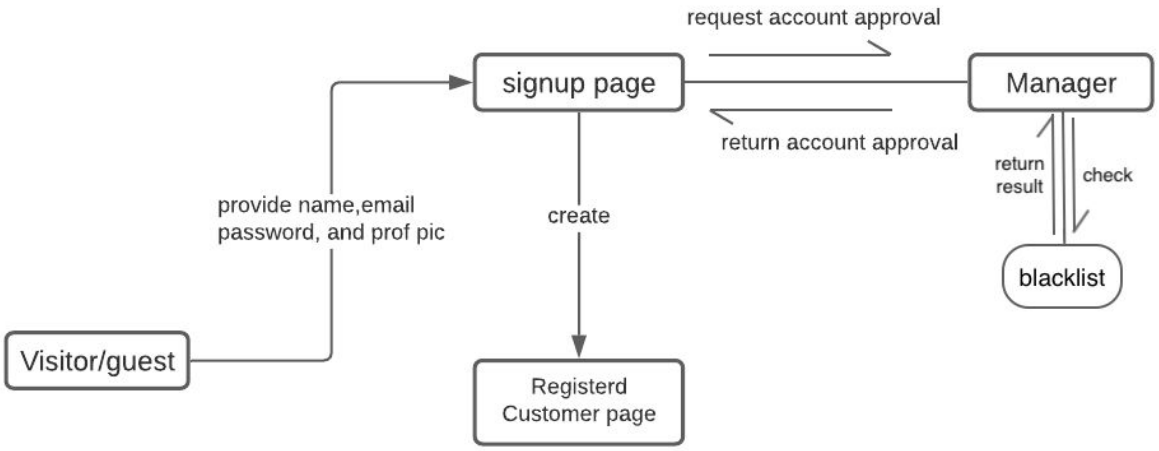
2.1 Register account

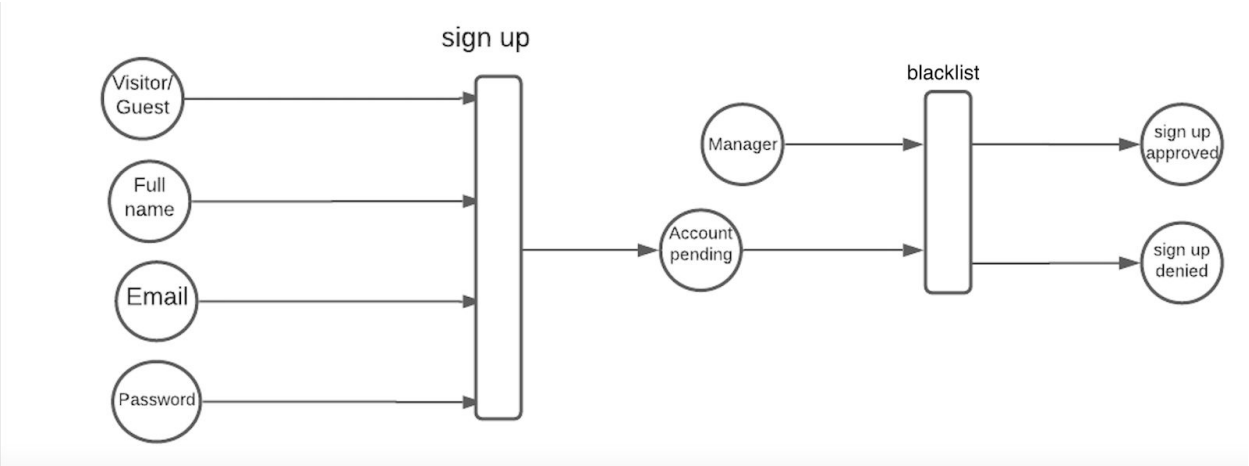
Normal scenario:

Upon entering the Pizzeria's website, a customer may register an account. Details that must be provided include: first name, last name, and email. After providing these details, customers will be prompted to create a username and password in the sign-up section.

Exceptional scenario:

If a customer opts out of registering their account, they are regarded as a guest.





2.2 Login

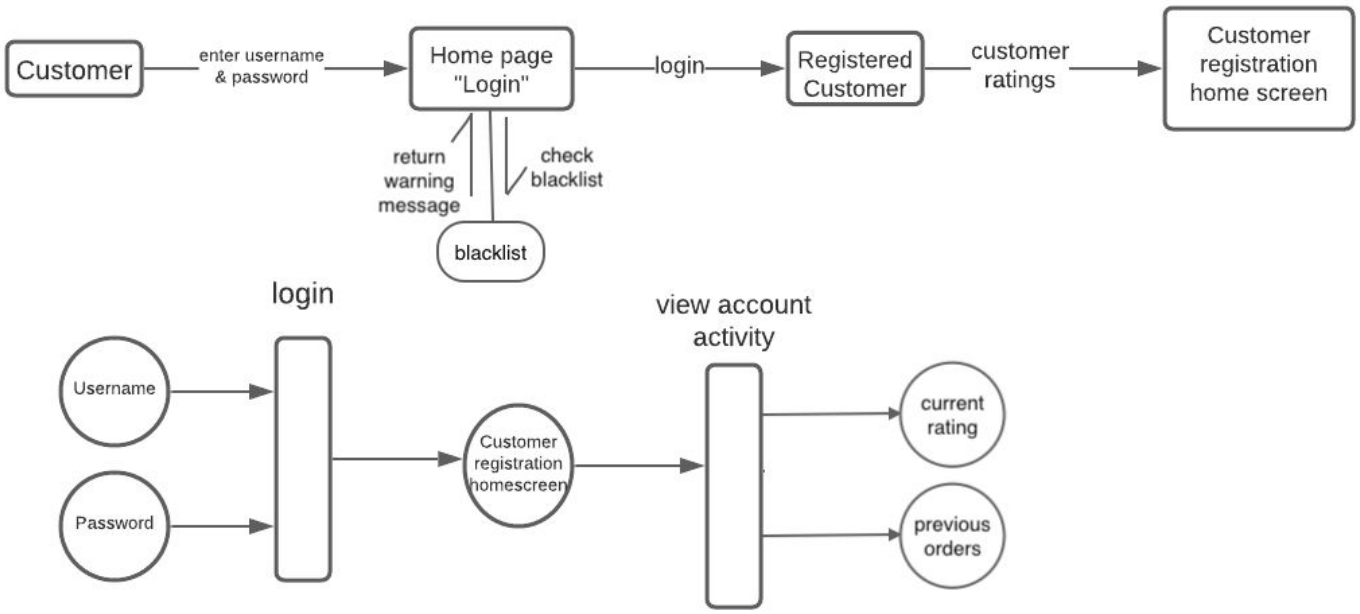
Normal scenario:

The customer will be prompted to input in their username and password upon the login screen. After logging in, access to the main account is granted where their account standing (personal rating) is displayed and they can order pizza!

Exceptional scenario:

If the customer is in negative standing (bad rating history); the manager may blacklist them as a precaution. If the customer is blacklisted, they will receive a message notifying them of said blacklist.

If incorrect username and/or password is entered, the customer will be redirected to the main login screen.



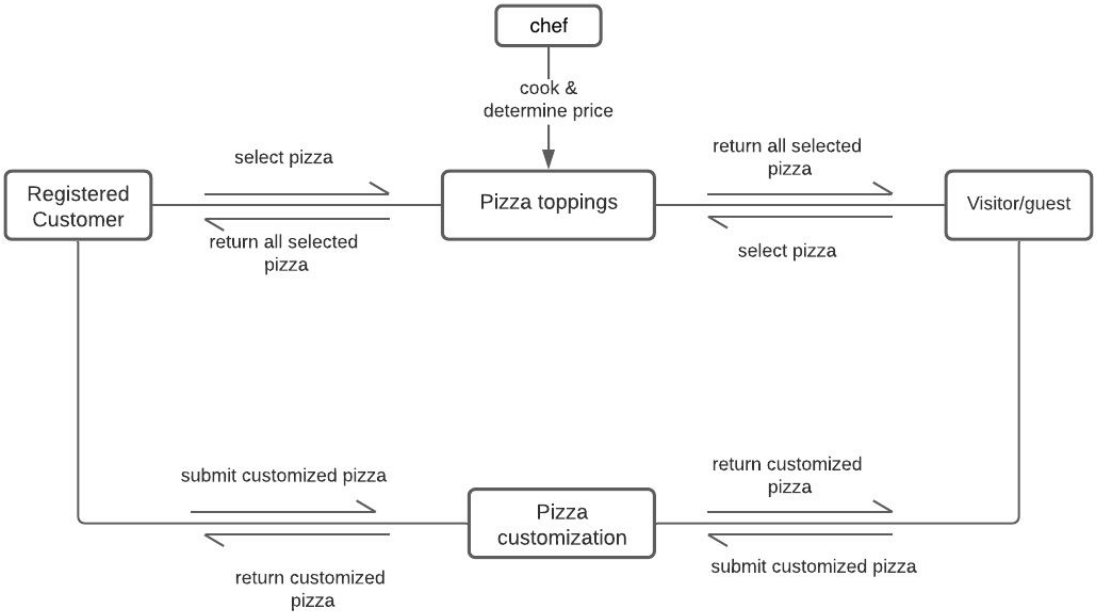
2.3 Design pizza

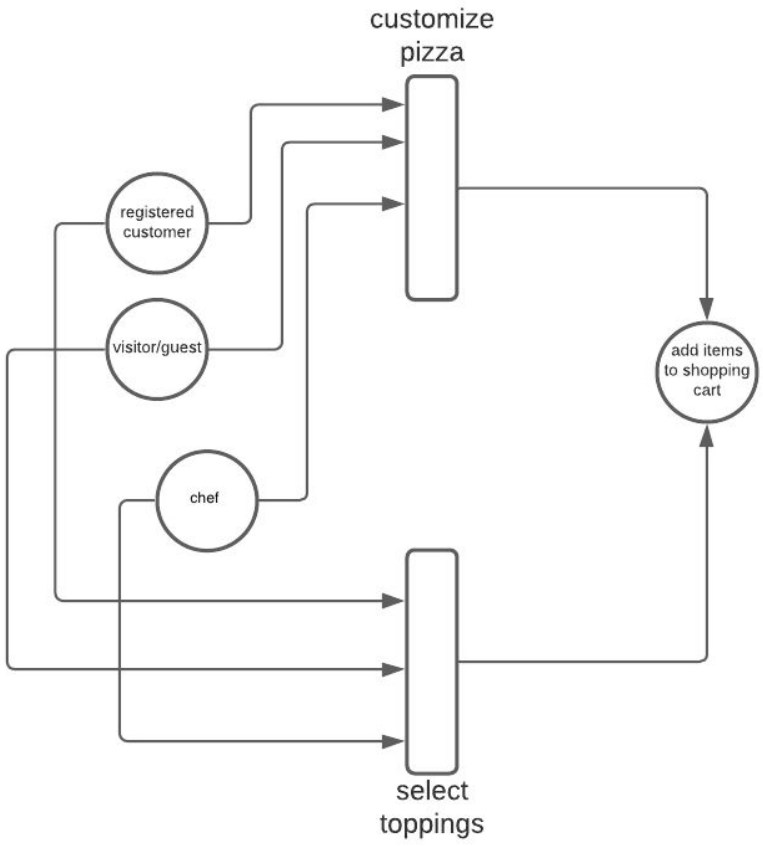
Normal scenario:

Both registered customers and guests can select from a wide variety of toppings for their pizza(s) (and choose the quantity). They can also pick one cook that will make the specified pizza(s).

Exceptional scenario:

If the listed pre-selected toppings are not to the customer's liking; they may create a customized topping arrangement of their choice. Since it's customized, it will be more expensive.


chef
cook &
determine price
Registered
Customer
select pizza
return all selected
pizza
Pizza toppings
return all selected
pizza
select pizza
Visitor/guest
submit customized pizza
return customized
pizza
Pizza
customization
return customized
pizza
submit customized pizza


customize
pizza
registered
customer
visitor/guest
chef
add items
to shopping
cart
select
toppings

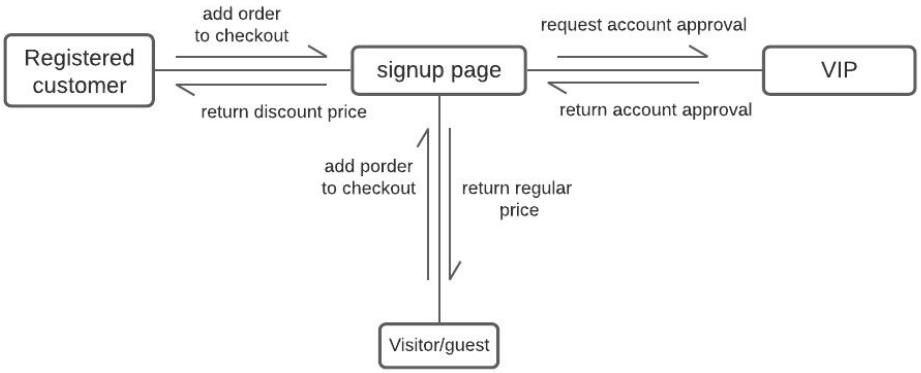
2.4 Price tiers

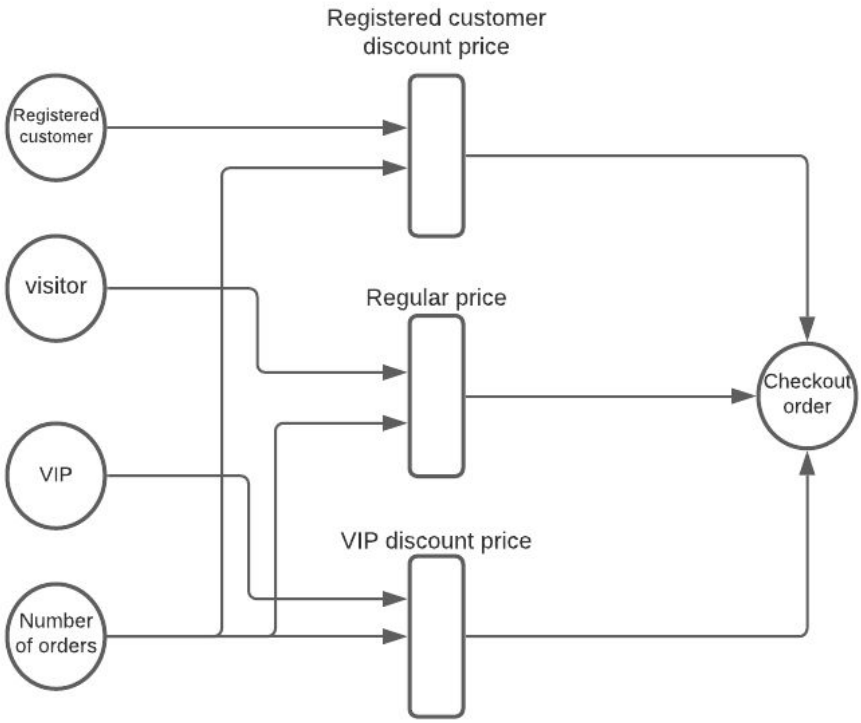
Normal scenario:

After the customer is satisfied with the design of their pizza, they can check out their order. The respective pricings is determined by the chosen cook.

Exceptional scenario:

VIP customers will get a VIP discount upon their order. Registered customers will get a respective discount (not as high as VIP) and guests will play the regular price.





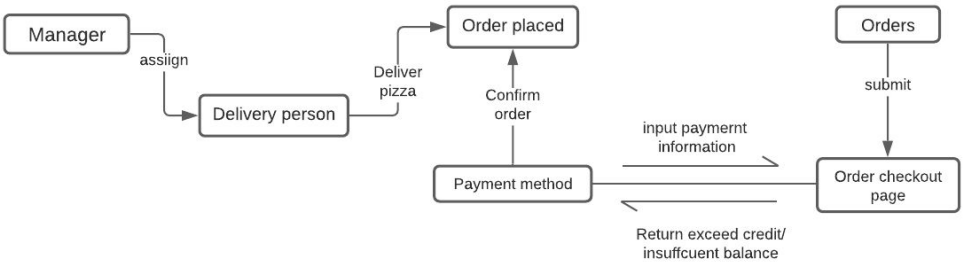
2.5 Order submission

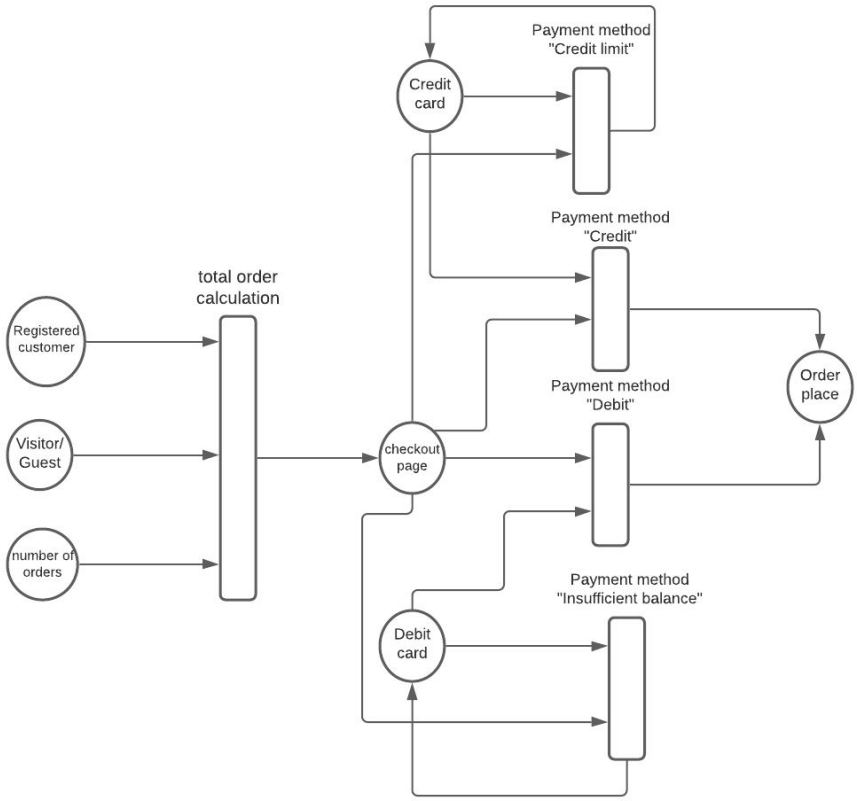
Normal scenario:

After customers submit their order, they are directed to the checkout page where they can pay by credit or debit card. After payment information is accepted, they can submit the order. The delivery personnel will choose the optimal route to deliver the pizza as quickly as possible.

Exceptional scenario:

If the credit card limit is exceeded, or if there are insufficient funds in the debit card; the order will not be processed. The customer will then have to enter another payment method (credit or debit).





2.6 Rating System

Normal scenario:

After the delivery person delivers the pizza to the customer, they will both rate each other based on service and mannerism. Then the customer will be prompted to rate the pizza itself, which by implication, means the cook. Lastly, the customer also rates the store itself (Champion's Slice). All ratings are on a 1-5 scale with 1 being the worst, and 5 being the highest rating.

Exceptional scenario:

Generally, if any rating is below 3, then regarded as a "complaint" and must accompany a justification through a comment.

The following customer ratings corresponds to their respective "status":

- If a registered customer rating is above 4 the promote the custom to VIP
- If a registered customer rating is greater than 2 but less than 1 then demote the customer to visitor/guest

The following delivery person's ratings corresponds to their respective "status":

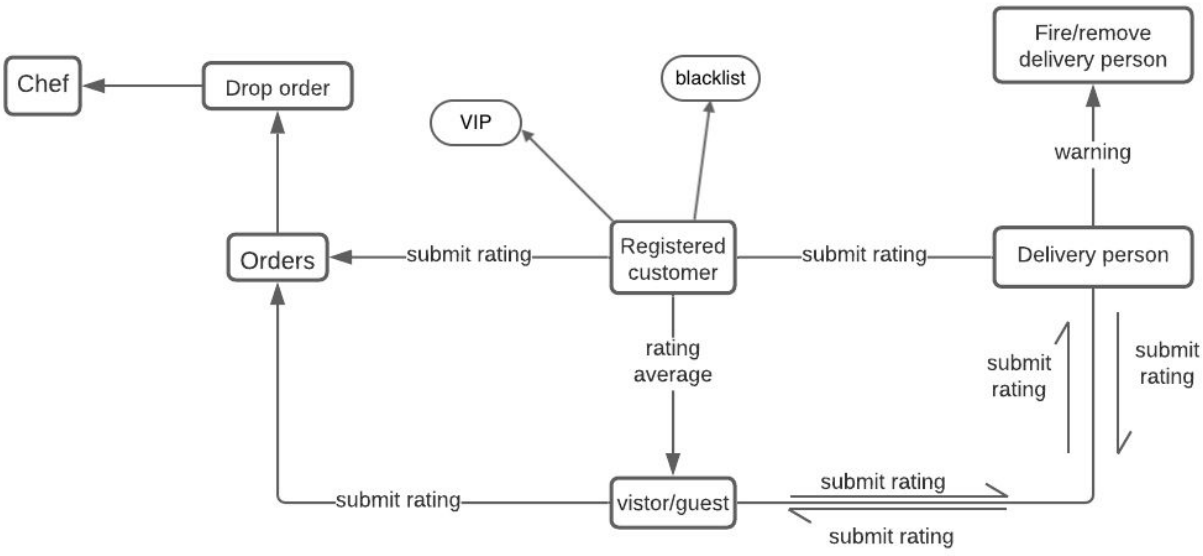
- If the rating of the delivery person is below 2 in the last three orders: formal warning (that can be erased).
- If the rating of the delivery person is below 2 in the last four warnings then termination.

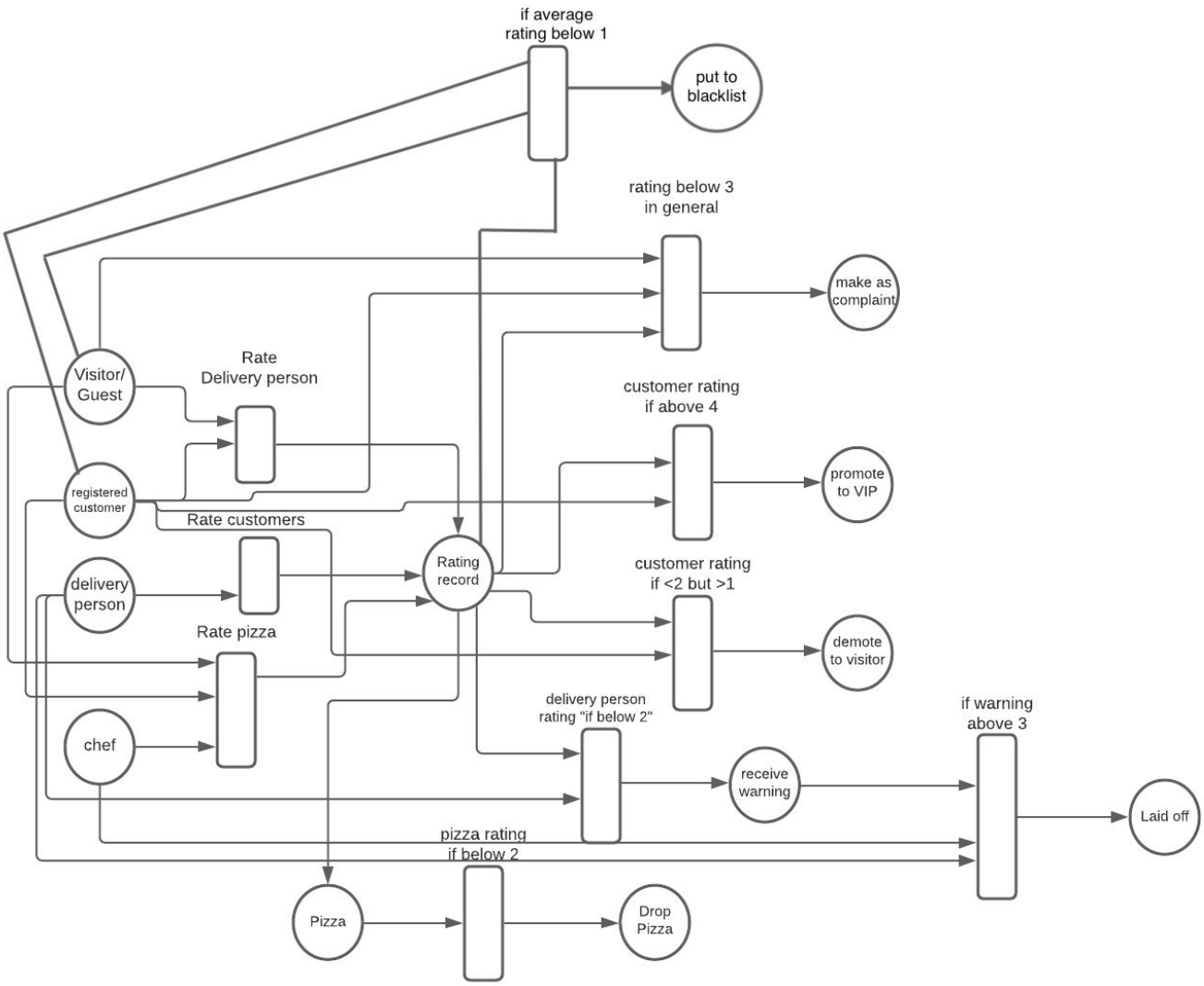
The following pizza rating corresponds to their respective "status":

- If pizza rating is below 2 in the last two orders then dropped.

The following cook rating corresponds to their respective "status":

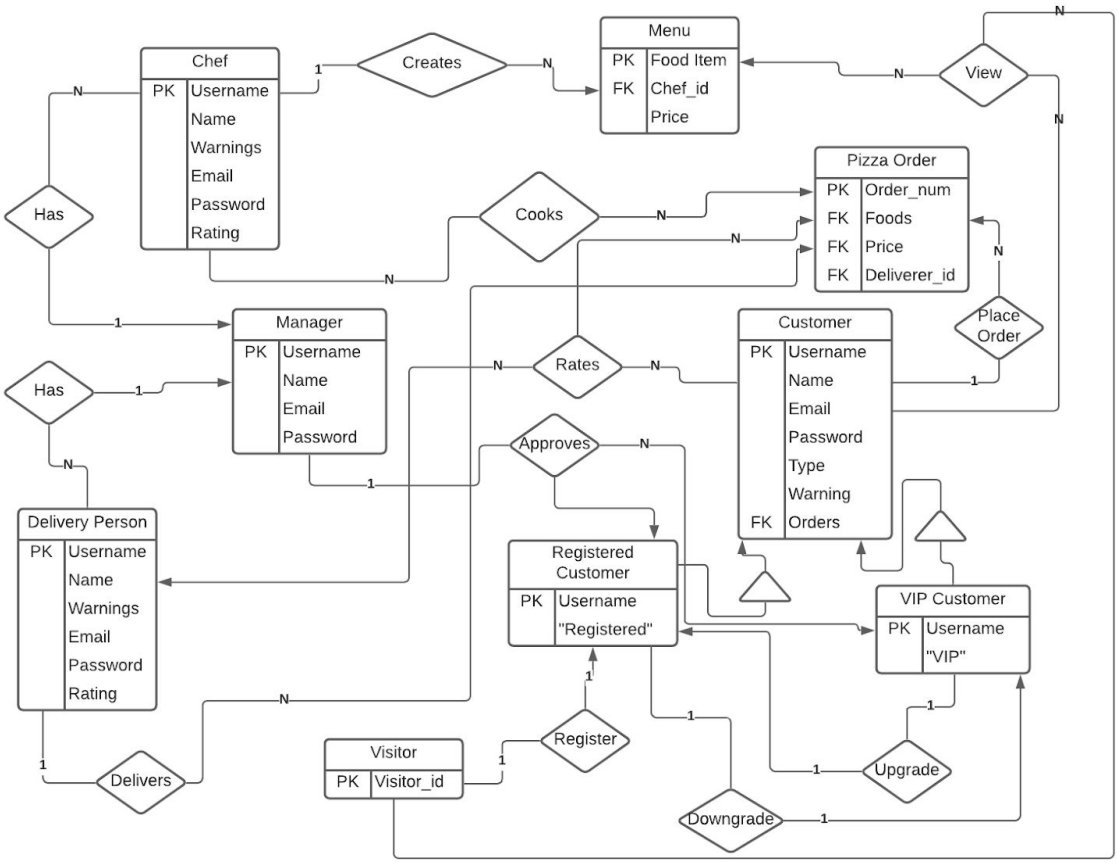
- If 2 dropped pizzas then warning
- If greater than 3 warnings then termination





3. Entity-Relationship Diagram

An E-R diagram is a graphical depiction of "Entity Relationship". This represents an information system that details the relationships between objects, concepts, events, places and of course, people. Here is our E-R diagram of our pizza ordering system for Champion's Slice.



4. Detailed Design

4.1 addRegUser

HTTPS-POST

Received: User information – first name, last name, email, username, password.

Returned: success/failure

---

data = {

first name: request -> body -> first name,

last name: request -> body -> last name,

email: request -> body -> email,

// check if email is in blacklist, if not - prompt for username and password

// if blacklisted – show the error message

username: request -> body -> username,

password: request -> body -> password

};

Add {data} in /RegUser/{request -> body -> uid}/

4.2 getUserCredData

HTTPS-POST

Received: user ID and user password

Returned: array of the account's details (personal rating, past orders, discount)

Promt {

get username: request -> body -> username,

get password: request -> body -> password

};

Request -> uid is in blacklist?

// if not – show rating/past orders/discount

// if blacklisted – show the error message

4.3 getOrderData

HTTPS_POST

Received: array of ordered items and the id of requested chef by choice

Returned: array of the items requested by customer for the future processing

---

{For each requested item from the menu:

oData = array of customized toppings per regular prices (+ additional if extra toppings),

For each requested customized pizza:

oData = array of customized toppings per increased prices,

Prompt the name of chef of the choice:

chef_id = chef's id

};

Add chef_id to oData and send all it back for customer's approval.

4.4 getPricingData

HTTPS-POST

Received: array of ordered items and customer's type (guest/registered/VIP)

Returned: array of the respectively priced items for the future submission

---

{get all ordered items -> data,

get the uid -> uid -> check the uid status -> apply discount,

send back discounted order

};

4.5 getPayment

HTTPS-POST

Received: debit/credit card information

Returned: success/failure

pData = {

name on the card: request -> body -> name_card

card number: request -> body -> card number

expiration date: request -> body -> exp_date

security code: request -> body -> security_code

};

Add {pData} to order processing

// success – return estimated delivery time

// failure – prompt for another payment method (expired card / insufficient funds)

4.6 Rating System

HTTPS-POST

Received: array of the ratings referred to delivery/pizza/store OR customer // from delivery personnel

Returned: array of the rating details

For delivery person:

{ getCustomerRate:

Data = get [1,5] for customer rating

// If below 3 -> send as complaint to manager and Prompt for the comment

};

For VIP & registered Customer:

{getDeliveryRate:

Data = get [1,5] for delivery person rating

// If below 3 -> send as complaint to manager and Prompt for the comment

};

{getPizzaRate

Data = get [1,5] for each ordered pizza -> assign to pizza & chef

// If below 3 -> send as complaint to manager and Prompt for the comment

};

{getStoreRate

Data = // get [1,5] for service overall

// If below 3 -> send as complaint to manager and Prompt for the comment

};

// Manager will further approach every “complaint” in the appropriate manner:

// (promote/demote/warning/termination)

5. System Screens

Below is a prototype of the main home page screen that showcases login, registration (addRegUser) and order (getOrderData) as a guest. Currently it is only for showcase purposes (non functional).

This is a very basic non-functioning prototype that allows for inputs.
However, no responses shall be yielded as of this phase. The finalized version will have a functioning model with redirects.

Login

you@example.com

Password

Sign in

Don't have an account?

Sign up

6. Minute of Group Meeting

Meeting	Content
1	Report Phase 1 and Diagrams
2	Standup: layout basic functions and brainstorm how our database will look like.
3	Report Phase 2 and Diagram. HTML prototype of our website home screen.

7. Supporting Information

Champion's Slice GitHub
Assignment link