Foodie Villa

Online Restaurant Order and Delivery System
Design Report
For Web and Mobile Application

Version <1.1 >

Foodie Villa	Version: <1.1>
Software Requirement Specification	Date: November 17th, 2020
Foodie Villa Phase 2 Report.docx	

Revision History

Date	Version	Description	Author
October 20, 2020	1.0	Create software requirement Specification	Sahrina Bhuiyan Fnu Palak Hope Dunner Nicholas Comer
November 10, 2020	1.1	Create design report	Hope Dunner Sahrina Bhuiyan Fnu Palak Nicholas Comer

Table of Contents

Software Requirements Specification

1. Introduction

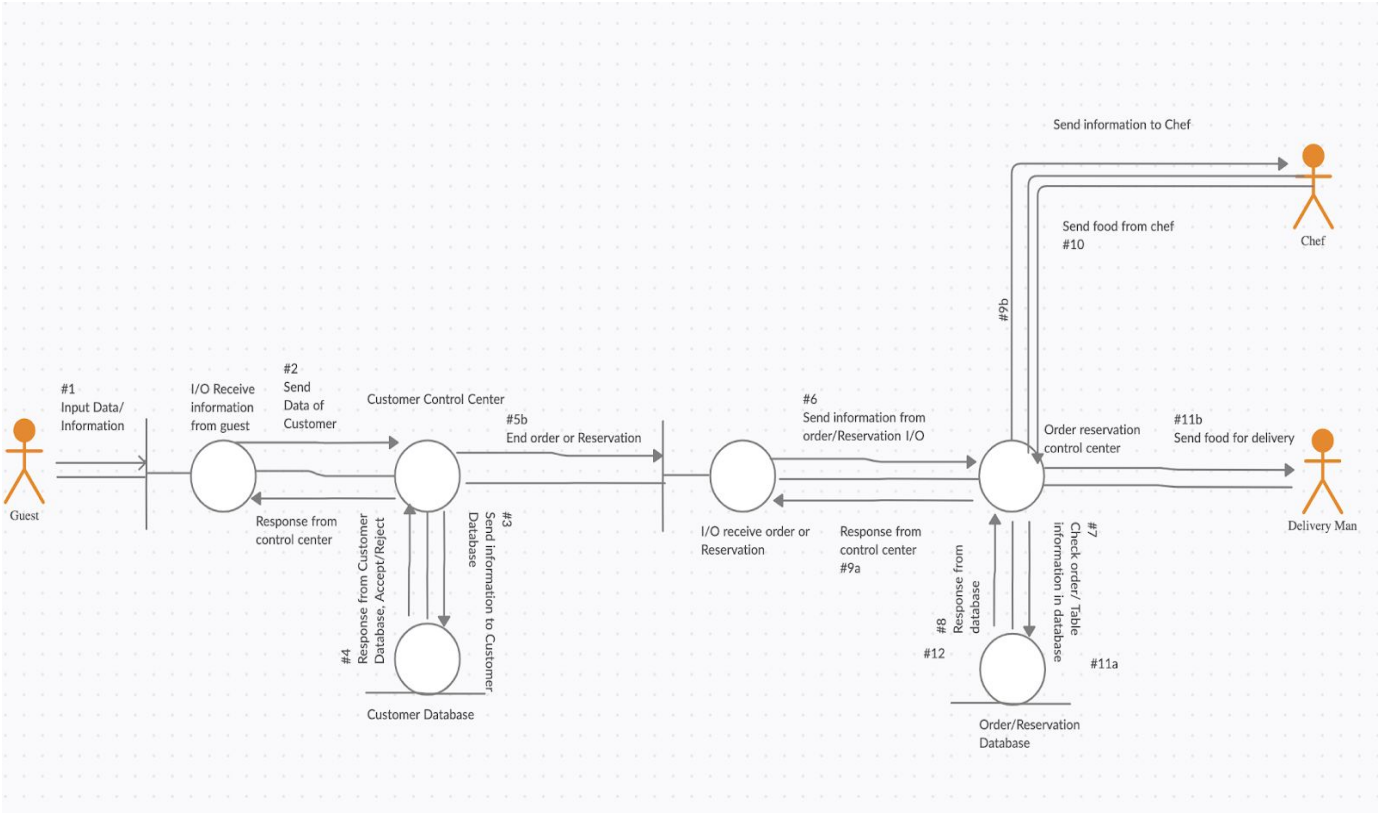
This design report will be an overview of the functionality and design of our online ordering system for our company. The purpose will include use of key definitions, and pictures of the system depicted through a collaboration class diagram.

1.1- Purpose

The purpose of this document will entail the functionalities which will be carried out by our application. This section will introduce how the system functions.

1.2- Collaboration Class Diagram

Collaboration diagrams that represent the relationships among the objects working together. The diagrams reflect the number of objects and the messages that are passed between the systems. Below is our collaboration class diagram that outlines our Foodie Villa system. This will serve as an outline of our interactions between the various users with the system as well as the overall functionality of our system.



2. All use cases

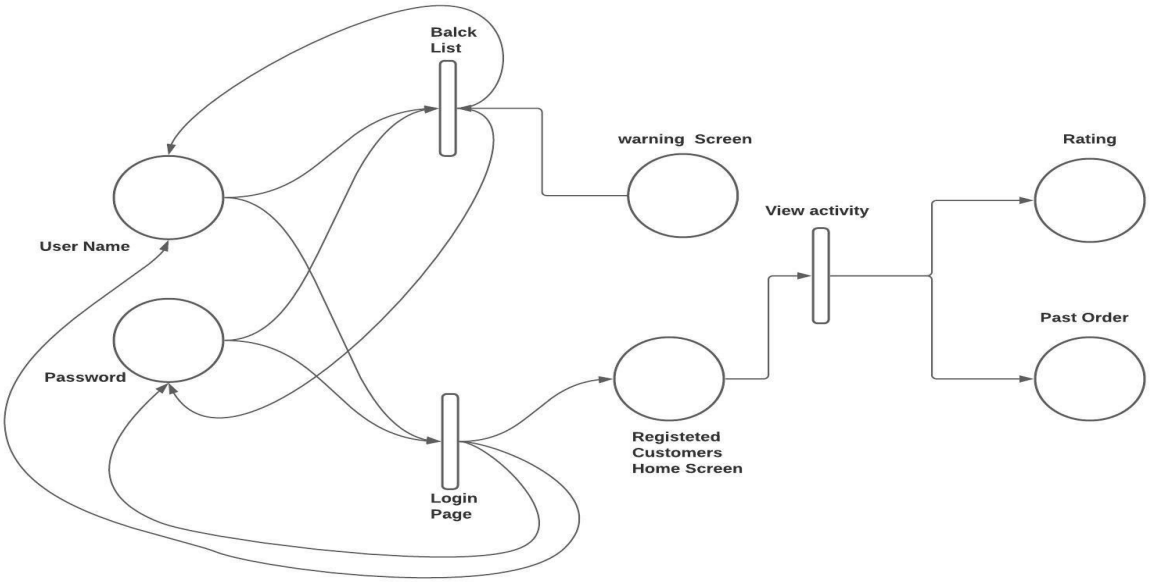
Login:

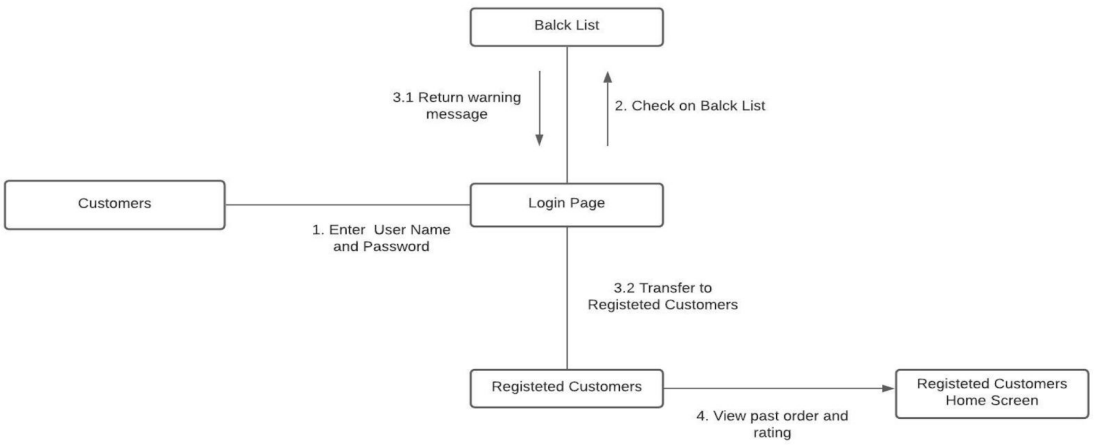
Normal Scenario:

The customer will be prompted to either sign up or input their User ID and their password if they already have an account. Once a customer logs in, he/she will have access to view their account. In the account, the customer is able to see his/her rating, give rating and also will be able to order food.

Exceptional Scenario:

The manager has the option of blacklisting customers that has a history of having a lot of bad ratings. When customers are blacklisted they will be warned and notified during login and denied from proceeding on the website.





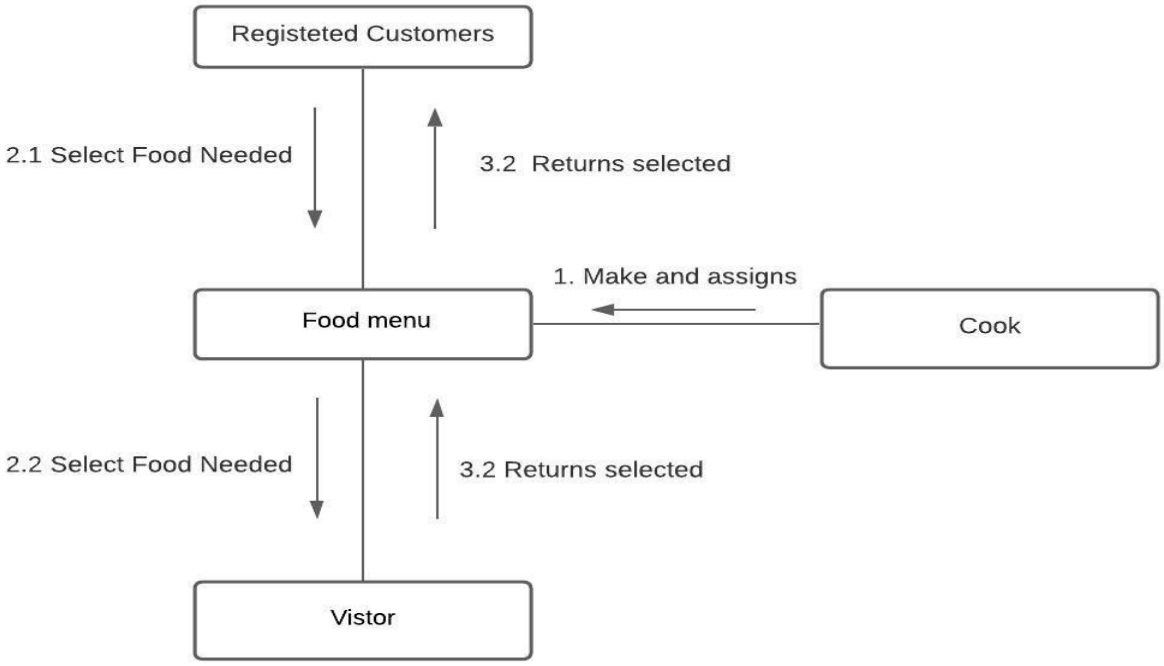
Food Choices:

Normal Scenario:

Registered customers are greeted with a screen that allows them to choose from the food menu. We have vegetarian and non vegetarian options they can choose from. They can also choose one cook that makes the food. Then also choose the way to take the food, either he eats in the restaurant or takes his plate in his hand or through the delivery.

Exceptional Scenario:

If registered customers or visitors are dissatisfied with the food quality, they can lodge a complaint or a good review with the chef.



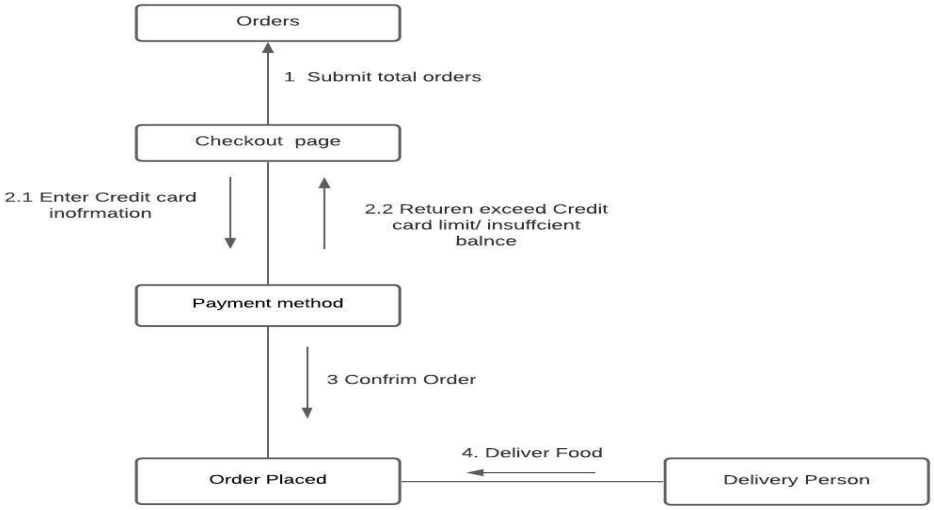
Submit Order:

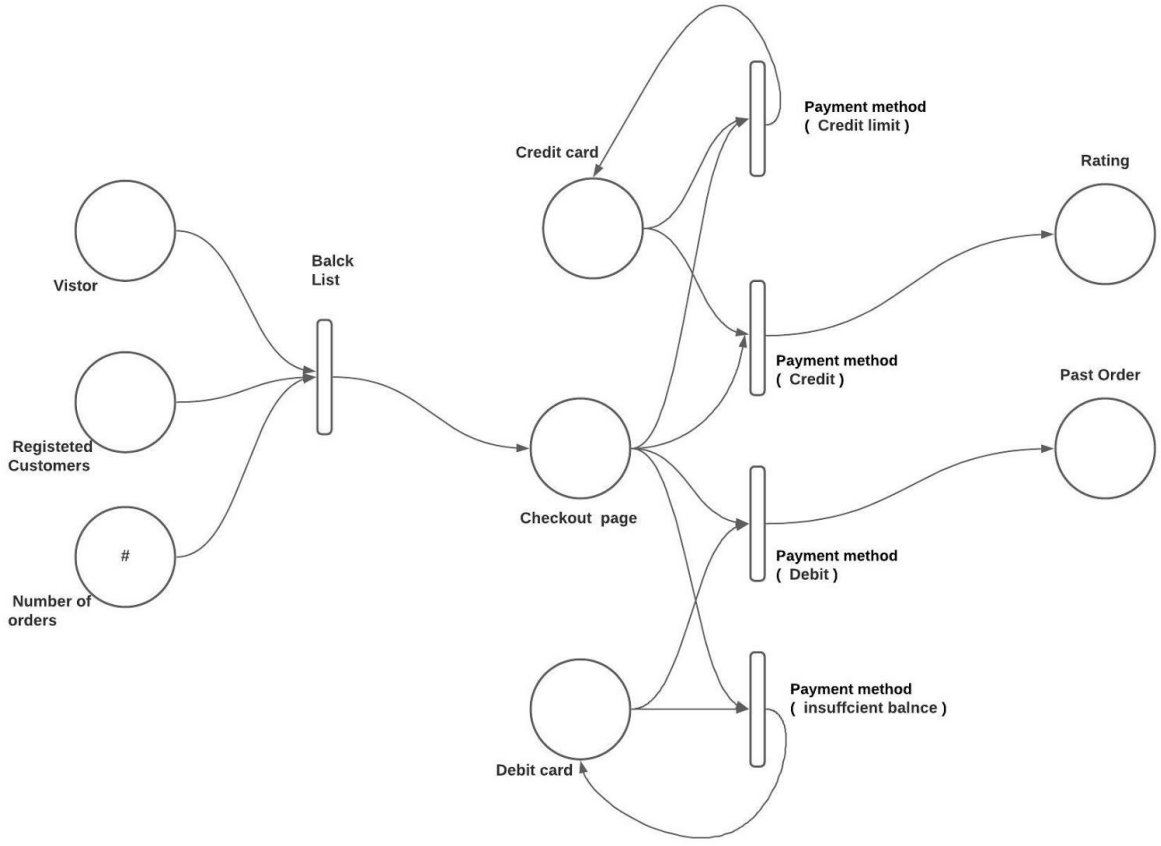
Normal Scenario:

After selecting and adding food to the cart, customers can proceed to the checkout page where they are greeted with an option to pay with either credit or debit card. After the payment information is entered, they can submit their order. This will notify the delivery personnel to compete for who will deliver that food.

Exceptional Scenario:

If the credit card exceeds the credit limit or the debit card balance is insufficient to pay for the order, the order will not process and the customer will be prompted to re-enter another credit or debit card or cancel order.





Rating System

Normal Scenario:

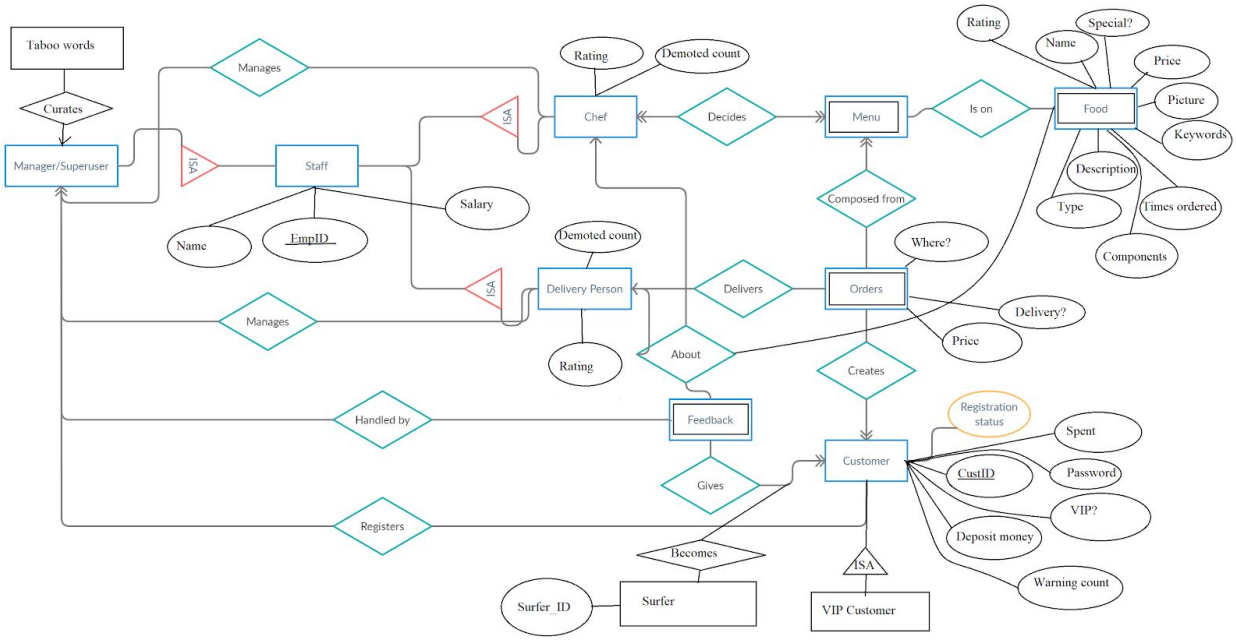
Customers and deliverers will be asked to rate each other based on service and manner, after receiving and delivering the food. Then, the customer will be given the option to and is also asked to rate the food which is a rating that will go towards the performance of the cook. These ratings are based on a 1-5(worst-best).

Exceptional Scenario:

Ratings of 1 and 2 will be viewed as a complaint and must provide a detailed explanation for the reason of their rating. Registered customers whose rating is below an average of 2/5, but greater than 1/5 will be demoted to a “visitor”. Customers rating with an average of 1/5 will be blacklisted. A customer with an average rating of above 4/5 will be considered a VIP member. A deliverer whose average rating is below 2/5 in the last three orders will be given a warning. If given more than three warnings, the deliverer will have to be laid off. Food items with an average rating less than two in the last three orders will be removed. A cook who made two dropped foods will be warned and after more than three warnings, the cook will be laid off.

3. E-R Diagram for the whole system

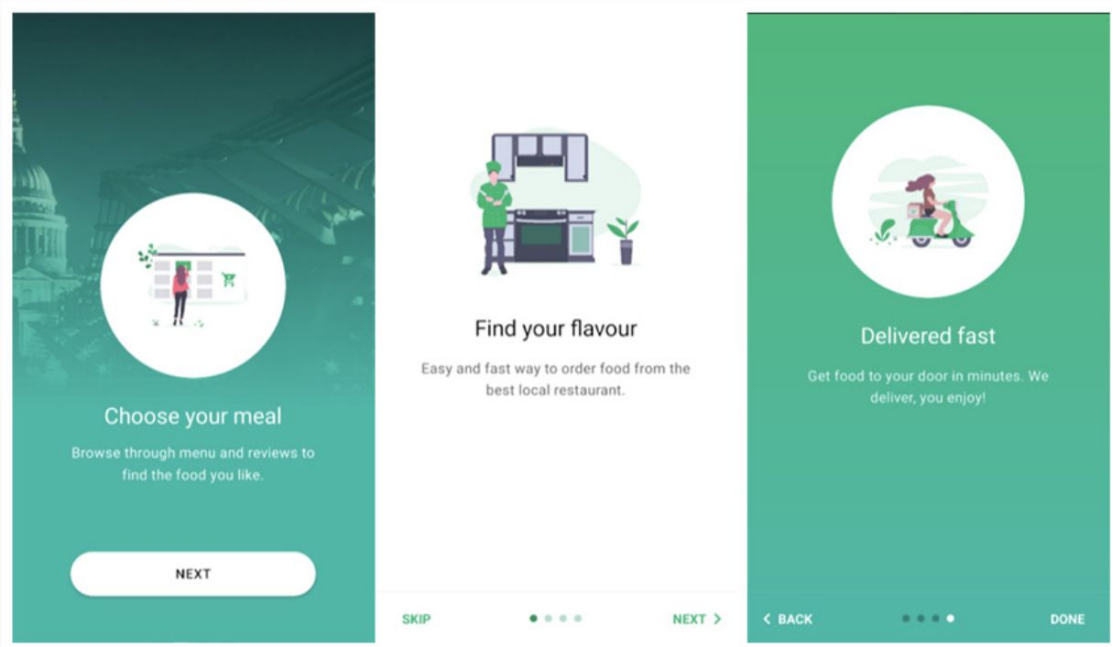
Entity relationship diagram is a graphical representation of an information system that represents relationships among people, objects, and concepts within our system. Below is an Entity Relationship Diagram describing the overall Foodie Villa and outlining our database.



4. Detailed Design System Screens

```
Food = pd.DataFrame({ #Food table initialized with pasta data
    'Food_ID':[0,1,2],
    'Name':['Pasta','Pizza', 'Cheeseburger'],
    'Description':["It's pasta.",'Actually just pizza.', "Almost as cheesy as my jokes."],
    'Price':[19.99, 19.99, 14.99], # I have no idea how food is priced normally. Someone can fix this but it doesn't really matter.
    'Chef_ID':[0,0,1], #Foreign key that tells which chef this food is from.
    'Picture':['images/pasta.jpg','images/pizza.jpg','images/cheeseburger.jpg'], #These are file names of pictures saved on the git.
    'Times_Ordered':[10,10,10],
    'Rating':[7,7,7], # This is a sum of customer feedback. +1 per compliment, -1 per complaint.
    'Tag':['Italian', 'Italian', 'Burger']}) #Tag to implement different listings shown for customers. Display based on similar tags of past ordered food.

Customer = pd.DataFrame({
    'Username':['A','B'],
    'Password':['password','alsoPassword']})

Order = pd.DataFrame({
    'Customer_ID':[0],
    'Food_ID':[0]})

Chef = pd.DataFrame({
    'Chef_ID':[0,1]})

Food #Prints the customer table. You can delete or comment this out if you want, it's just so you can see how the examples work.
```

5. System Screens



6. Minutes of group meetings/concerns

Meeting #	Discussion
1	Discussed all requirements and started report 1
2	Discussed program functionalities and database
3	Started implementation

7. Address of git repo

https://github.com/NComer98/CSc322