

# SW Engineering CSC648/848 Spring 2021

## HungryGators-19

Food Delivery and Restaurant Review Service

Available on Mobile and Desktop

### Team 01

Branden J. Nguyen – Team Lead – [bnguye11@mail.sfsu.edu](mailto:bnguye11@mail.sfsu.edu)

Jasmine Jahan – Front End Lead

Yongjian Pan – Back End Lead

Robert Freeman – GitHub Master

Gurjot Singh – Team Member Front End

Tin Thu Zar Aye – Team Member Back End

### Milestone 2

March 19, 2021

Date Submitted	Date Revised
March 19, 2021	

## Table of Contents

I.	Executive Summary.....	2
II.	List of main data items and entities.....	3
III.	Prioritized Functional Requirements.....	8
IV.	UI Mockups and Storyboards.....	13
V.	High-level Architecture, Database Organization Summary.....	27
VI.	Key Risks.....	28
VII.	Project Management.....	29

## **I. Executive Summary**

San Francisco State University is a place of cultural diversity. There are many people including students, faculty, and staff from different parts of the world whose food habits are also different from each other. Though there are a variety of cafeterias/canteens in SFSU, they are not available for online delivery in the campus area. Most of the time students have to go to the cafeteria to get the food. This can be a hassle especially if the students are having food in between classes, in which case, they may not have time to walk to the cafeteria, order the food, wait and be late for class. As well as the students have to select the food randomly without reading any review or rating which could be misleading. There are some other applications to get the food delivered from larger restaurants but available only outside the SFSU campus area. The process is clunky and hard to navigate.

We, Team 1, are building an application by which the students, faculty, and staff could see the review and order food within the campus, delivered straight to their classroom door or dorm. With this food delivery service students, faculty and staff will be able to order food by creating an account with their sfsu.edu email and name. If the person does not belong to the university they cannot order from the app. For the person who isn't familiar with using the app, they can get the phone number from the app and can pick it up. The minimum viable product of our internet application can be described as a restaurant review and delivery service solely catered to San Francisco State University students and staff. This app will not only solve the problem of food delivery for students of SFSU but also save time for the students to study.

The application would capitalize on the gold rush that is the student-delivery industry. This cornerstone of good execution of this business shall be the options to choose from, and the form delivery of the food. We would exceed our competitors by not only serving exclusively SFSU students but also monopolize students at the school, which are otherwise untapped resources. This emphasis on delivery addresses ease-of-use as well as delivery options and more restaurants, this service will have speedy delivery and a ton of options. HungryGators-19 is an application that will revolutionize food delivery services for students at San Francisco State University.

## II. List of main data items and entities

**KEY:** PK=Primary Key, FK=Foreign Key; the non-filled circle is a table, a black square is a column (attribute), and the dark circles are notes about the attributes.

### ◆ customer\_orders

- idcustomer\_orders
  - PK
- customer
  - FK: idsfsu\_customer from sfsu\_customer table
- receipt
  - FK: idcustomer\_receipts from customer\_receipt table

### ◆ customer\_receipts

- idcustomer\_receipts
  - PK
- customer
  - FK: idsfsu\_customer from sfsu\_customer table
- receipt
  - FK: idcustomer\_receipts from customer\_receipt table

◆ delivery\_driver\_record

- iddelivery\_driver\_record
  - PK
- name
- vehicle
  - VARCHAR(45)

◆ delivery\_employee

- iddelivery\_employee
  - PK
- target\_address
- delivery\_time
- pickup\_address

◆ person

- idperson
  - PK
- address
- age
- sfsu\_customer\_id
  - FK from sfsu\_customer table
- delivery\_employee
  - FK: iddelivery\_employee from delivery\_employee table

◆ receipt

- idreceipt
  - PK
- item\_price
- item\_name

◆ records

- idrecords
  - PK
- restaurant
  - FK from idrestaurant from restaurant table
- delivery\_driver\_record
  - FK iddelivery\_driver\_record from delivery\_driver\_record table
- user\_reg\_record
  - FK iduser\_reg\_record from user\_reg\_record table

◆ restaurant

- idrestaurant
  - PK
- address
- phone\_number
- zip\_code

◆ restaurant\_person

- idrestaurant\_person
  - PK
- idrestaurant
  - FK idrestaurant from restaurant table
- idperson
  - FK idperson from person table

◆ review

- idreview
  - PK
- review
- sfsu\_customer
  - FK idsfsu\_customer from sfsu\_customer table

◆ sfsu\_customer

- idsfsu\_customer
  - PK
- name
- address
- zip\_code

◆ takeout\_order

- idtakeout\_order
  - PK
- item
- price

◆ user\_reg\_record

- iduser\_reg\_record
  - PK
- username
- email



### **III. Prioritized Functional Requirements**

#### **Priority 1**

##### **Unregistered Users**

1. Shall be able to search for restaurants based on different parameters.

1.1 Location, restaurant name, menu items, etc.

2. Shall be able to browse existing restaurant reviews.
3. Shall be able to view, edit, or cancel their order via a cart or list.

##### **SFSU Customers**

4. Shall be able to create an account using their SFSU email.

4.1 SFSU email shall be verified.

4.2 Shall be registered as 'student/staff/faculty' profile.

4.3 Registration shall include name and SFSU email.

5. Shall be able to login and logout of their account.
6. Shall be able to search for restaurants based on different parameters.

6.1 Location, restaurant name, menu items, etc.

7. Shall be able to leave a rating or review of searchable restaurants.

7.1 Ratings shall include a star rating from 0-5 with an optional comment review section or photo attachment.

8. Shall be able to browse existing restaurant reviews.

9. Shall be able to place a food order at a restaurant with a delivery or pickup option.

9.1 Once the order is placed, the order is automatically paid for (not simulated).

9.2 Unique features shall include the ability to leave specific delivery instructions.

10. Shall be able to view, edit, or cancel their order via a cart or list.

## **Restaurant Owners**

11. Shall be able to create an account using their email.

11.1 Shall be registered as 'restaurant owner' profile.

12. Shall be able to login and logout of their account.

13. Shall be able to create a profile page to be published by the Admin.

13.1 Create and update their name, profile image, and advertisement.

14. Shall be able to manage incoming placed orders.

14.1 This includes marking the completion of preparation of an order, order cancellation, and assigning a delivery driver to the order.

## **Delivery Drivers**

15. Shall be able to create an account using their email.

15.1 Shall be registered as 'delivery driver' profile.

16. Shall be able to login and logout of their account.

17. Shall be able to download and view order details to facilitate restaurant food delivery to SFSU Customers.

## **Admin**

18. Shall be able to login and logout of their account.

19. Shall be required to review and approve restaurant profiles to be published that meet the terms of service.

20. Shall be required to delete inappropriate items from restaurant profiles that violate the terms of service.

21. Shall be able to delete SFSU Customers, Restaurant Owners, or Delivery Drivers that have violated the terms of service.

## **Priority 2**

### **SFSU Customers**

22. Shall be notified of the status of a placed order.

### **Restaurant Owners**

23. Shall be able to add or remove pictures from their menu or advertisement.

23.1 Unique feature shall include the attachment of photos to menu items or a photo carousel of popular menu items.

24. Shall be able to add or remove menu items.

25. Shall be able to add, remove, or update menu item descriptions.

### **Delivery Drivers**

26. Shall be able to access a SFSU campus map UI for delivery directions.

## **Priority 3**

### **SFSU Customers**

27. Shall be notified in real time about the exact status of their order through a notification system.

27.1 Notifications shall be sent when: an order is placed, a command has been entered to check the order status, a command has been entered to cancel the order, a restaurant is in the process of preparing the order, a restaurant has completed

preparing the order, a delivery driver has been assigned to deliver the order, the delivery driver is on their way to pick up the order, the delivery driver has picked up the order, the delivery driver is en route to the delivery address, the order has been successfully delivered, the restaurant is closed or no longer taking orders.

28. Shall be able to deregister their account.

29. Shall be able to update their account and contact information.

### **Restaurant Owners**

30. Shall be able to de-list their profile if they do not wish to continue service.

31. Shall be able to deregister their account.

32. Shall be able to update their account and contact information.

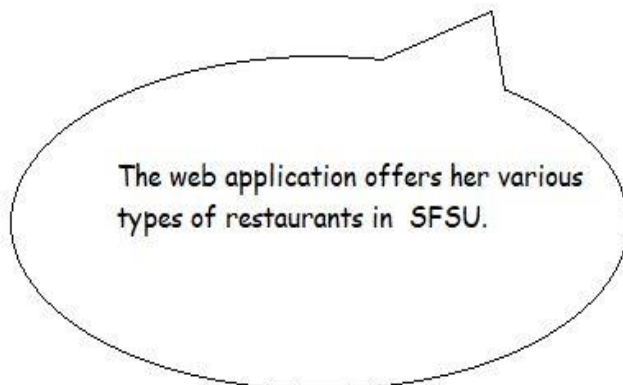
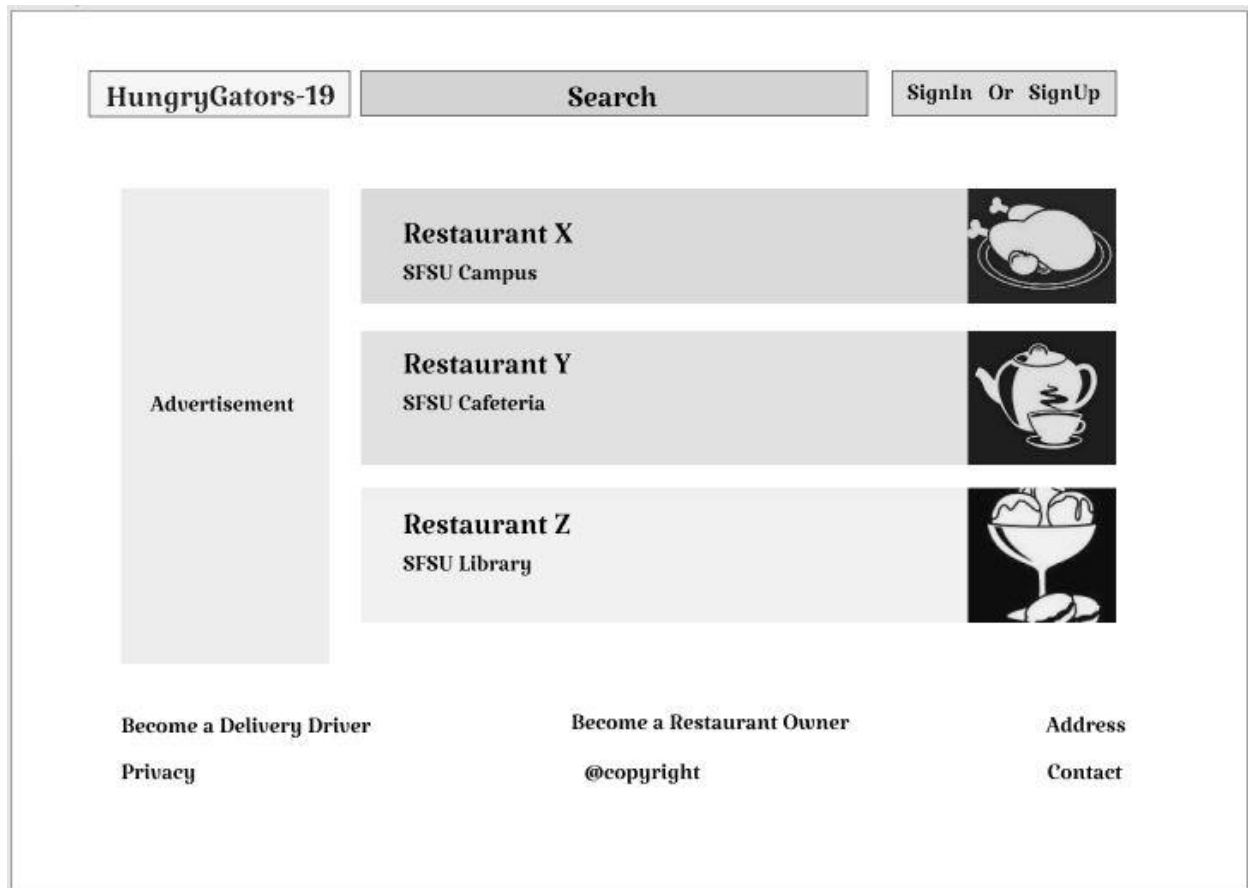
33. Shall be able to stop taking online orders.

33.1 This feature is available if somehow the restaurant is closed during the specific hours or is no longer in service.

## IV. UI Mockups and Storyboards

**Use Case 1:** Diane, a full-time student at SFSU. She orders her meals through the app regularly.

Diane opens the web application to order her breakfast.



She selected Restaurant X primarily. Then the page of Restaurant X appears.

Restaurant X

Photo of Restaurant X

Read Reviews

Write a Review

Checkout

Breakfast

Lunch

Dinner

Desserts

Drinks

Breakfast Item 1 with Photo

☒

☐

Breakfast Item 2 with Photo

☒

☐

Breakfast Item 3 with Photo

☒

☐

Privacy

@copyright

Contact

After reading some good reviews, she added 2 breakfast items and went to checkout.

The login page appears before checking out.

The screenshot shows a web page with a login interface. At the top, there are two buttons: "SignIn" and "SignUp". Below these are two input fields: "Email Sign In" and "SFSU Student/Faculty/Staff ID". To the right of the "Email Sign In" field is the text "OR". Below the "SFSU Student/Faculty/Staff ID" field is a "Login" button. At the bottom of the page, there are three links: "Privacy", "@copyright", and "Contact".

**SignIn** **SignUp**

Email Sign In **OR**

SFSU Student/Faculty/Staff ID

**Login**

**New to HungryGators-19? Please SignUp !**

**Privacy** **@copyright** **Contact**

Diane signed in with her SFSU ID which took her to the Checkout page. She really liked that easy Sign in the system, it saves her time always.



Lastly, Diane provided the delivery information and phone number then placed the order. She got an order confirmation immediately.

CheckOut			
Delivery Information	Coupons	Payment on Delivery	Place Order
Delivery Address :	<input type="text"/>		
Phone Number :	<input type="text"/>		
Order Summary :			
Privacy	@copyright	Contact	

**Use Case 2:** Jose is a full-time student at SFSU but now he also wants to work as a driver at HungryGator-19. He opened the app and found “Become a delivery driver ” on the home page which takes him to the SignUp page.

SignIn    SignUp

Email    OR

SFSU Student/Faculty/Staff ID

SignUp

Privacy    @copyright    Contact

He signed with his SFSU ID but this time the app gave him a different profile as a delivery driver.

Jose finds the location via the SFSU campus map and the app also shows him the address and time. The pick-up location was close to him so he accepted the order to deliver.

Desktop - 9

<b>SFSU Campus Map</b>	<b>PickUp Info</b>	<b>Delivery Info</b>
------------------------	--------------------	----------------------

**Address :**

**Estimated Time:**

**Accept**

**Privacy**                      **@copyright**                      **Contact**

After picking up the order from the restaurant, he again opens the app to find the exact delivery location with the help of SFSU Campus Map. Finally, Jose headed towards the delivery address.

Desktop - 10

**SFSU Campus Map**

**PickUp Info**

**Delivery Info**

**Address:**

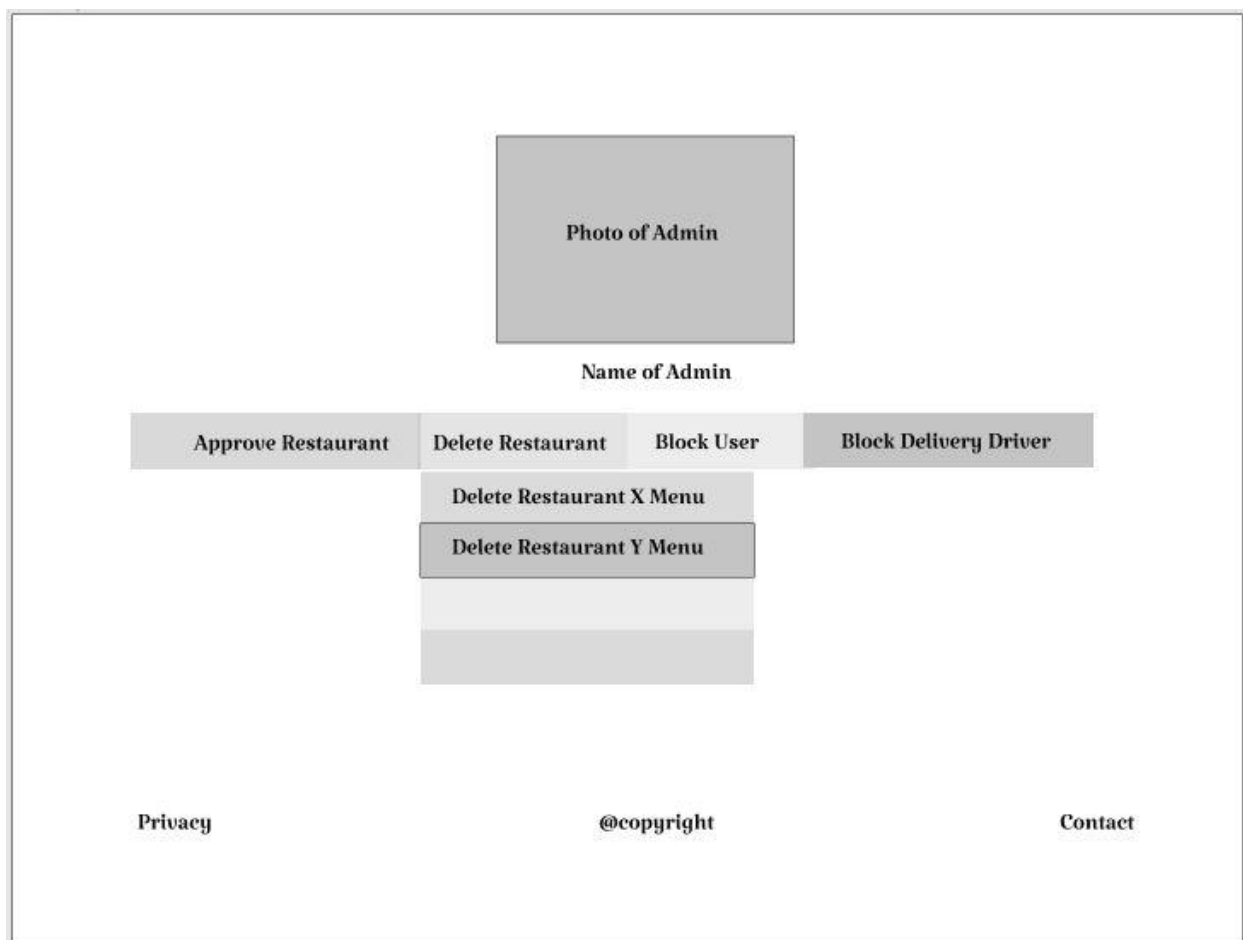
**Estimated Time:**

**Privacy**

**@copyright**

**Contact**

**Use Case 3:** Leo, A SFSU staff member who is also an admin of HungryGators-19. Recently, he read some bad reviews about some food items from Restaurant Y.



**Use Case 4:** John is a busy full-time staff at SFSU. He is more concerned about the rating of the restaurant than what kind of food item he wants.

First, he selected Restaurant X, but the reviews were not good enough then he switched to the menu of Restaurant Y.

Frame 8

Restaurant Y

Photo of Restaurant Y

Read Reviews

Write a Review

Checkout

Breakfast

Lunch

Dinner

Desserts

Drinks

Lunch Item 1 with Photo

☒☐

Lunch Item 2 with Photo

☒☐

Lunch Item 3 with Photo

☒☐

Privacy

@copyright

Contact

He opened the HungryGators-19  
web app to read some reviews.

Read Reviews

Review 1  
\*\*\*\*\*  
Text Field with Photo

Review 2  
\*\*\*\*\*  
Text Field with Photo

Privacy @copyright Contact

After having a delicious lunch John  
opened the web app to write a review  
to let others know about the food  
items.

**Write Review**

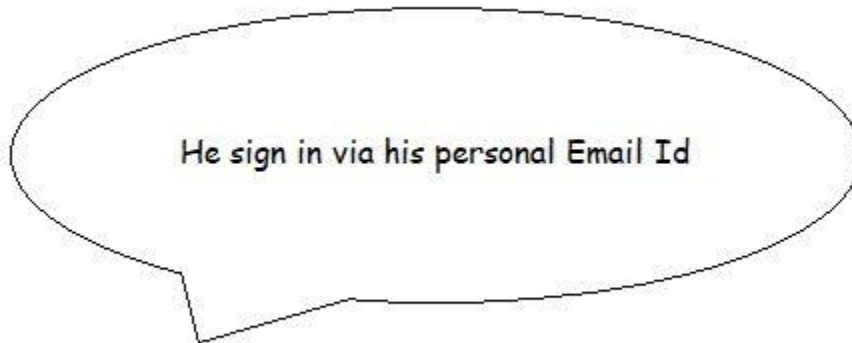
**Choose File** No file choosen

**Text Field**

**Privacy** **@copyright** **Contact**



**Use Case 5:** Calvin is one of the restaurant owners at SFSU neither an SFSU student nor an SFSU staff.

A screenshot of a web login page. At the top, there are two buttons: "SignIn" and "SignUp". Below these are two input fields: "Email Sign In" and "SFSU Student/Faculty/Staff ID". To the right of the "Email Sign In" field is the text "OR". Below the input fields is a "Login" button. At the bottom of the page, there is a message: "New to HungryGators-19? Please SignUp!". At the very bottom, there are three links: "Privacy", "@copyright", and "Contact".

**SignIn** **SignUp**

Email Sign In **OR**

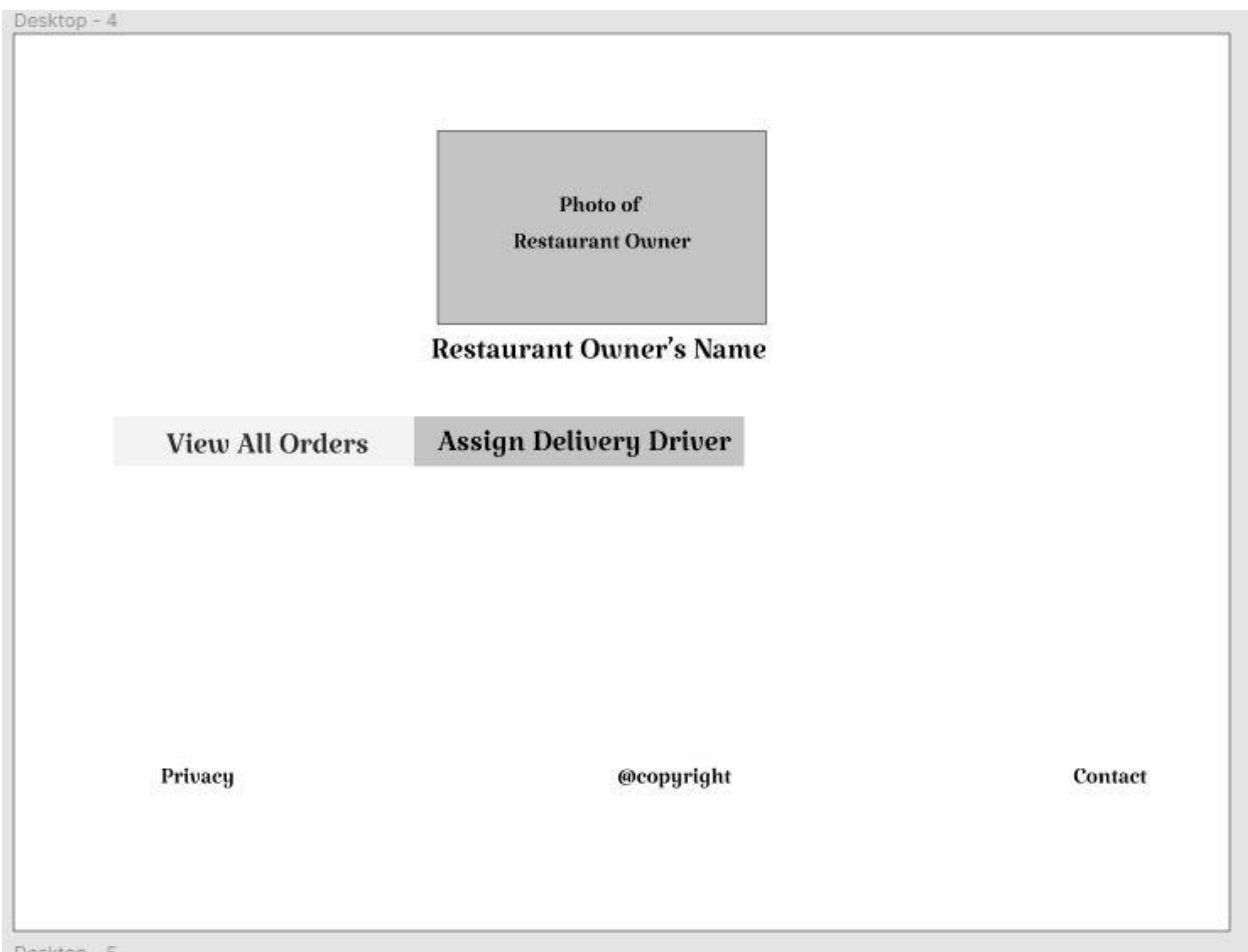
SFSU Student/Faculty/Staff ID

**Login**

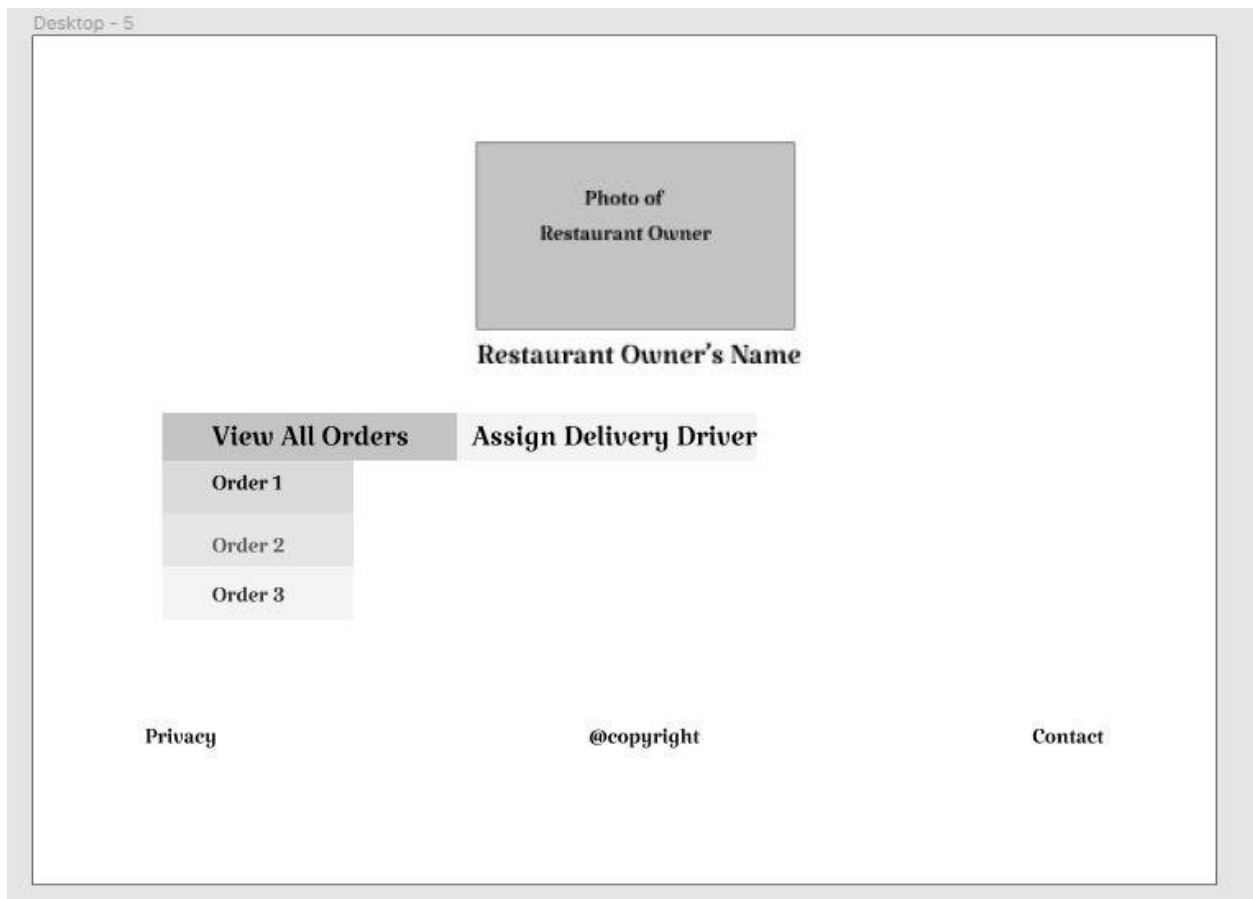
**New to HungryGators-19? Please SignUp !**

**Privacy** **@copyright** **Contact**

This is Calvin's profile where he maintains all orders coming to his Restaurant.



He opened the web app to view coming orders and assign the delivery drivers to the orders.



## V. High-level Architecture, Database Organization Summary

- **Media storage:** Images and video/audio will be kept in the file system. There are no other special data format requirements for video/audio/GPS etc.
- **Search/filter architecture and implementation:** Using the LIKE % method. (Tutorial we will follow found here: <https://medium.com/@joseortizcosta/search-utility-with-flask-and-mysql-60bb8ee83dad>)
- **Your own APIs:**
  - Search Utility with Flask and MySQL
    - Flask
    - flask-MySQL
  - Bootstrap
- **Any significant non-trivial algorithms:**
  - A significant and non-trivial algorithm will be queuing takeout orders and assigning them to drivers dynamically. This shall be done by traversing a list of takeout\_order objects which have not yet been assigned a driver. Then, any available delivery drivers (and of course the restaurant) should be alerted about this new order first in, first out. The delivery driver shall be alerted of the new order and the restaurant shall start making the order. No time estimate will be given.
- No changes have been made to any SW tools or frameworks. None were added. We are continuing as planned.

## VI. Key Risks

- **Skill Risks:** Being able to learn and apply the right skills to build the prototype by the due date (This key risk is due to the level of expertise and time needed to build the prototype by the due date and solution may be to self-learn and search for online resources and ask around for help)
- **Schedule Risks:** Being able to launch the prototype remotely by the due date (This key risk is due to the amount of time needed to launch the prototype remotely and the solution may be to begin working on individual parts without waiting for teammates' part to complete)
- **Technical Risks:** Being able to deploy the prototype locally and remotely by due date (This key risk is due to the level of expertise needed in building, running, debugging, and deploying the prototype remotely by the due date and solution may be to ask for help if necessary)
- **Teamwork Risks:** Being able to designate and coordinate work in a timely manner (This key risk is mainly due to communication and skills needed to coordinate the efforts effectively and solution may be to communicate in a timely manner if problem arises)
- **Legal/Content Risks:** Being able to obtain legal content by the due date (This risk is due to obtaining online content but not giving citation needed and solution may be to ensure giving citation when obtaining online contents unless they are stated for free)

## **VII. Project Management**

To Plan M2, we meet on Friday and Sunday to go over the document requirements and assign the work to everyone. We also set the due date for everyone, so we have the document ready before the deadline, and everyone goes over the document the following Friday to review. Once we finish reviewing the document, we fix the mistakes, if there are any. We are using Zoom for meetings and Trello for organizing our work tasks. We are currently focusing on the search menu, restaurants, food menu, Driver sign-in, Admin sign-in, restaurant owner sign-in, customer sign-in, cart, and food review. In the future, we will work on text notification, campus map, working Sign-in/out tab, and adding removing order.