SW Engineering CSC648/848 Spring 2023

Project Name: GatorEats - Online Meal-Delivery Service

By: Team 7

Milestone 2 Part 1

Github: https://github.com/CSC-648-SFSU/csc648-03-sp23-team07.git

Team Lead: Andy Almeida - aalmeida1@sfsu.edu,

Team Members: Emily Huang, Melisa Sever, Eunice Borres, and Juan David Liang Liao

Date: 3/31/2023

Document History:

Date Submitted:	3/31/2023
Date Revised:	4/09/2023

Table of Contents

Executive Summary

GatorEats is an online platform that connects food-selling establishments with San Francisco State University supporters. By offering culinary services customized for the campus and student life, this service is intended to improve the SFSU experience. Everyone with an SFSU email account, including students, employees, and faculty, can access GatorEats. GatorEats can assist small and large businesses in San Francisco in reaching more clients by offering this service.

GatorEats is a revolutionary food delivery app that provides a convenient way for users to order food from various local restaurants. Clients can have their food delivered to one of the numerous drop-off locations or dorm rooms on campus. To simplify things for users with limited availability during the day, GatorEats also gives users the choice of their favorite delivery time. Customers can effortlessly use the app and place orders for their preferred meals thanks to GatorEats' user-friendly interface. Customers have the freedom to select from a variety of cuisines thanks to the app's extensive range of options from numerous eateries. Also, GatorEats collaborates with neighborhood eateries to provide consumers unique specials and discounts, allowing them to taste new foods while saving money.

With its unique delivery service to particular areas, GatorEats distinguishes itself further and gives clients a hassle-free delivery experience. By offering everyone a quick, dependable, and practical service, we hope to change the food delivery sector. We are confident that our app will establish itself as a go-to resource for SFSU students, employees, and teachers who yearn for good cuisine without the bother.

Our team is a startup of five San Francisco Computer Science students. Together, we are a combination of talented front-end, back-end developer, quality assurance, and GitHub master spearheaded by a team lead.We are proficient in JavaScript and BootStrap. Our team is passionate about creating solutions that make a positive impact on people's lives, and we believe that GatorEats has the potential to do just that. Being students, we understand the unique challenges and needs of the college community, and we're excited to provide a service that caters specifically to those needs.

Main Data Items and Entities List

Guest User: A general user that can access the website, browse restaurants, add food to cart, and register to continue the process
SFSU Registered User: A general user that has registered shall be considered a SFSU registered user. They must log in to access their account. Faculty, staff and students must register using their SFSU email.
Admin User: An admin has the ability to approve restaurants and images uploaded that will be displayed via MySQL Workbench.
Driver Registered User: A driver is a registered user that shall have access to a list of SFSU registered user's orders. They are responsible for the delivery and the completion of an order. They are also able to view the restaurant's location to pick up the food, as well as the SFSU registered user's drop off point to drop off the food.
Restaurant Registered User: A restaurant owner shall create a restaurant account. A restaurant registered user can build a restaurant within their account, and manage their information. A restaurant registered user can access their restaurant which shall have its own menu.
Order History: SFSU registered users shall be able to view all their past orders.
Order: An entity that contains a SFSU registered user's order information including the restaurant that the order was made from, the cost, the delivery driver delivering the order, the date the order was placed, the date/time the order was completed, and a Ticket.
Menu: A list that contains menu items for a restaurant. The menu is linked to a restaurant
Menu Item: A menu item would be linked to a menu, which is in turn linked to a restaurant. It would contain food options, descriptions, prices, and has a link to a photo
Ticket: A list that contains menuitems. The ticket is linked to an order.
Photo: Photos that are uploaded by a Business account. Photos shall have a relevant name and can be viewed by any user. Shall be approved by the admin before being viewed by other general users
Restaurant: A restaurant shall be linked to a restaurant registered account. They shall submit a name while registering. They shall have at least one location. They shall be able to submit relevant photos and the menu of their restaurant. They shall include the delivery time it takes for their food to be delivered to SFSU registered users.
Category: Restaurants owners will be able to link their restaurant to categories. This would allow for users to search through the list of restaurants based on categories.
Pickup Point: A pickup point will consist of a longitude, latitude, and a name. The pickup points can be a room inside of a building or a coordinate on the map within the SFSU campus.
Building: Buildings located on SFSU Campus. These building shall contain rooms
Room: Rooms located within SFSU buildings. These rooms shall serve as a point for delivery for drivers.

Functional Requirements - Prioritized

The initial functional requirements in Milestone 1 are grouped by user type, including guest user, SFSU registered user, and any other users. On the ground of Milestone 1, the functional requirements are explained further in Milestone 2, grouped by development priority. Herein, this version of requirements expands the initial list, in which more details have been added as sublists, and some inappropriate parts have been modified or removed. For each the condition that exists in Milestone 1, the reference numbers for each requirement are kept the same to avoid confusion.Every requirement is prioritized in three levels, namely **p1, p2,** and **p3.** Three priorities are **p1 - necessary, p2 - desired,** and **p3 - opportunistic**. Below are the requirements grouped by development priorities. The design of such groups is made after considering both the needs of users and the resources of our team, including technical support and delivery schedules.

Priority 1:

Guest Users:

1.1 Guest users shall be able to browse the site and search using food categories or cuisines.

1.2 Guest users shall be able to register for an account if they choose to.

1.3 Guest users shall be able to add items to a cart.

1.4 Guest users shall be required to sign up for an account in order to check out an order.

SFSU Registered Users:

2.1 A SFSU registered user shall be able to log into their account.

2.2 A SFSU registered user shall have a valid email address (sfsu.edu or mail.sfsu.edu) and a password since a valid email address, and password are required to create a user account and ensure the security of the user's personal information.

2.3 A SFSU registered user shall be able to order food.

2.4 A SFSU registered user must choose a location for the food during the order check out.

2.5 A SFSU registered user must complete the check out by using a Gator Card as a payment option.

Restaurants Registered Users(Restaurant Owner):

3.1. A restaurant user shall be able to register their account.

3.4. A restaurant user shall be able create a menu with dishes and prices for those dishes.

3.7 Once an order is created, the restaurant where the order is placed shall receive the orders details like customer name, time that the order was placed, and order items.

Admin User:

4.1 An admin user shall approve the restaurant via WB(Work Bench).

Driver Registered Users:

5.1. A driver user shall be able to register as a delivery driver.

5.2. A driver user shall be able to receive a delivery order.

5.3 A driver shall have access to a list of created orders to choose from and deliver.

Menu:

9.1. A menu shall have one or more food options, descriptions, and prices.

Category:

11.1 Restaurants shall be linked to a category.

Pickup Point:

12.1 A pickup point shall have a building name and a room number.

<u>Priority 2</u>:

SFSU Registered User:

2.1 A registered user shall be able to rate a restaurant that they have ordered from

2.2 A registered user shall be able to view previous completed orders

2.3 Once an order is created, the registered user shall receive an order confirmation and order details like the estimated time of arrival, time of order creation, and items ordered.

Restaurants Registered User (Restaurant Owner):

3.1. A restaurant user shall be able to add or delete items from their menu.

<u>Priority 3</u>:

Guest User:

1.1 Unregistered users shall be able to filter restaurants by price and rating.

SFSU Registered User:

2.1 A registered user shall be able to add extra instructions for the driver as a note during the checkout.

Admin User:

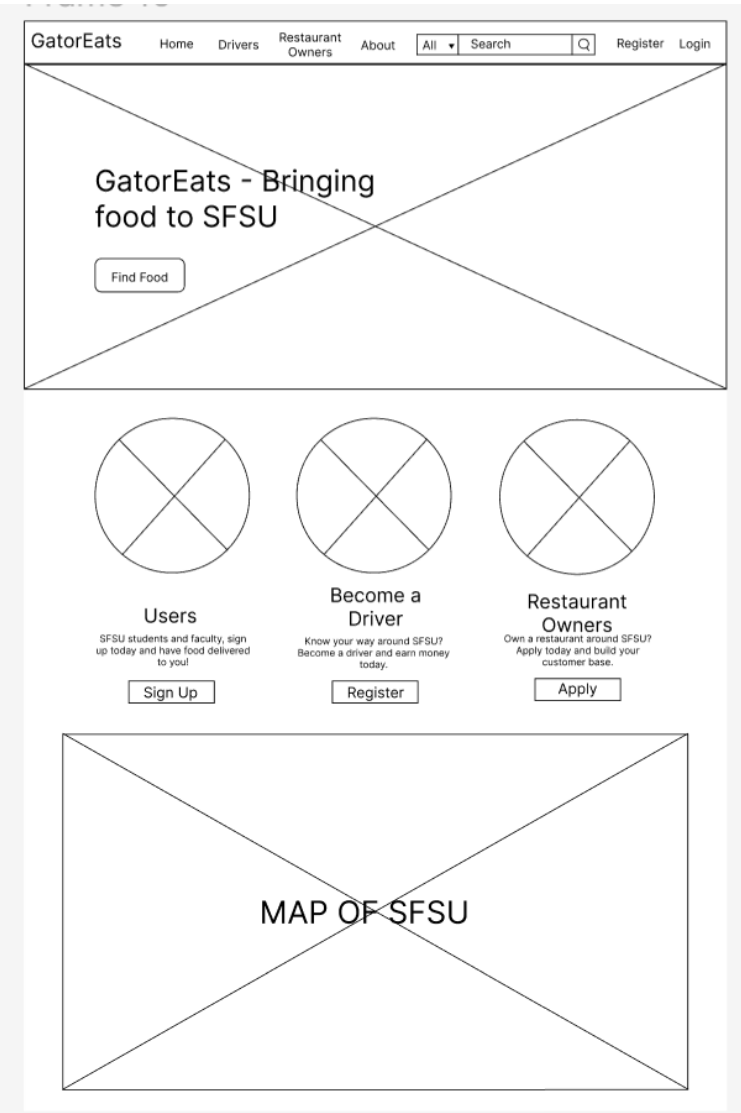
4.1. An admin shall be able to delete users and restaurants. This shall be done on WB.

UI Storyboards for each main use case

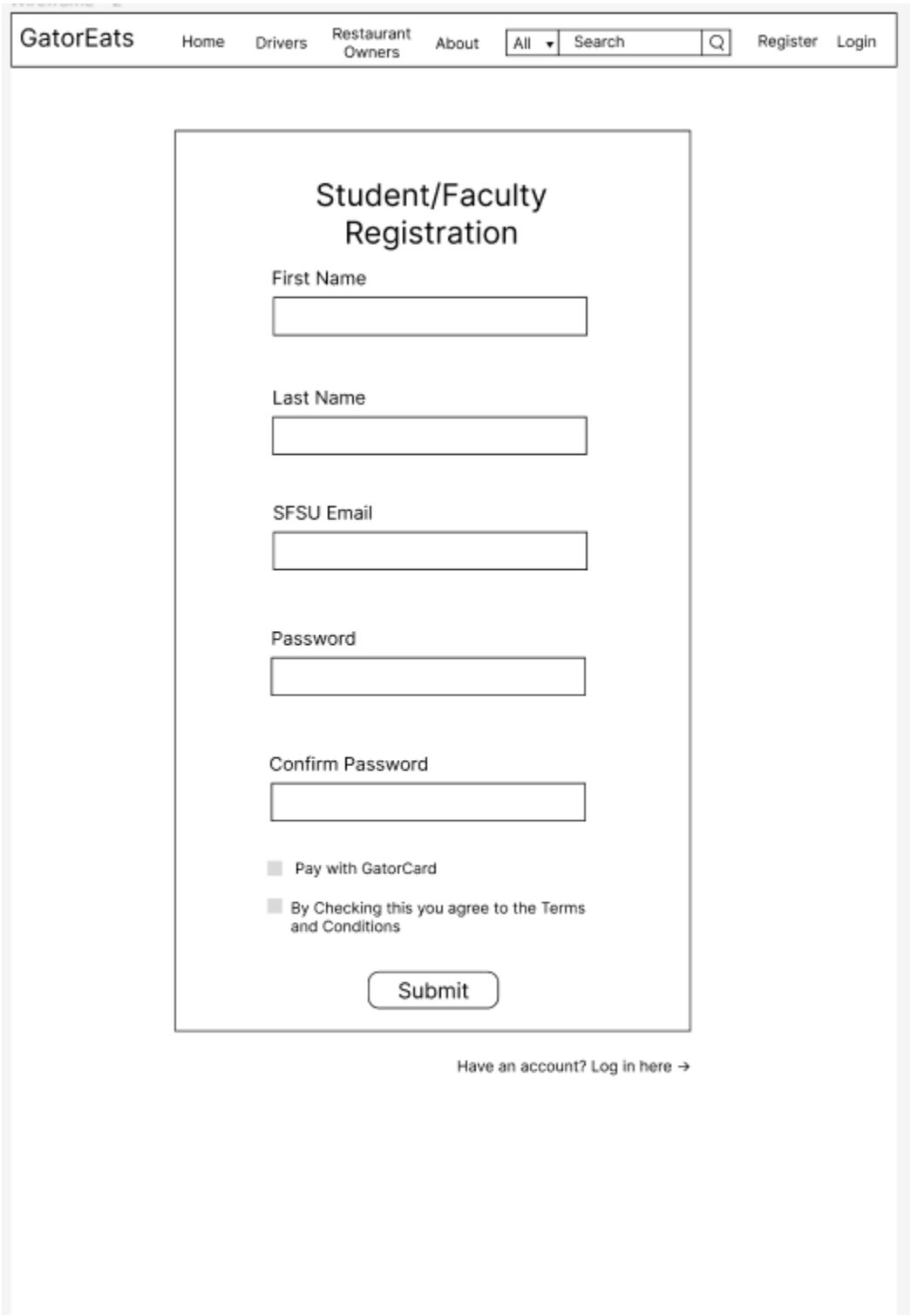
Faculty - Normal Registration

Maria is an English Professor who decides to try out GatorEats to cut down on time spent going out to find food between classes. She goes to the website and decides to sign up. Once signed in, she sees a list of all the restaurants. She decides on a restaurant and looks at the menu. She adds what she likes to the cart and checks out. She puts in her location and chooses to pay with her GatorCard.

Home Page



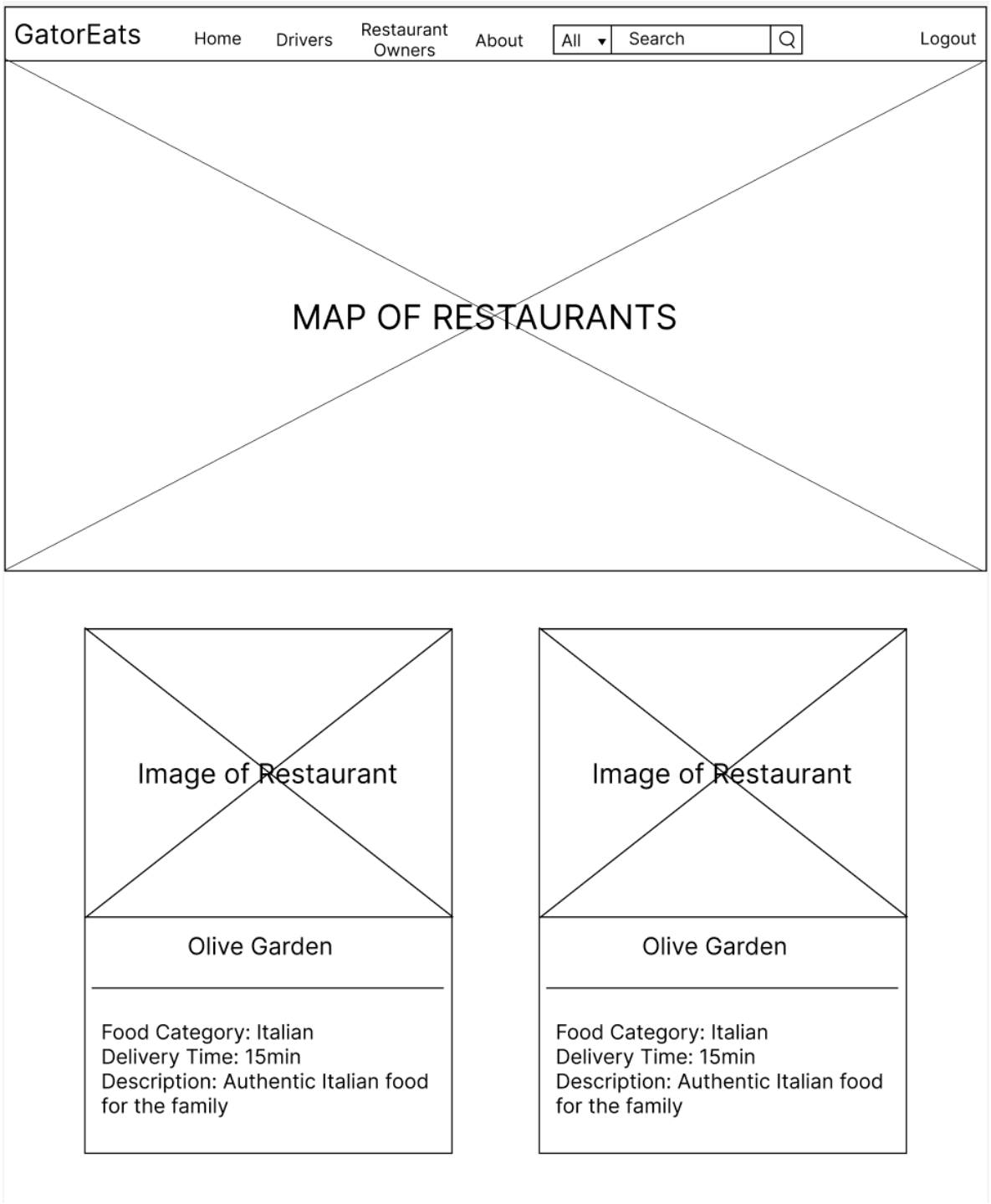
Sign Up Page


GatorEats
Home
Drivers
Restaurant Owners
About
All
Search
Register
Login
Student/Faculty Registration
First Name
Last Name
SFSU Email
Password
Confirm Password
Pay with GatorCard
By Checking this you agree to the Terms and Conditions
Submit
Have an account? Log in here →

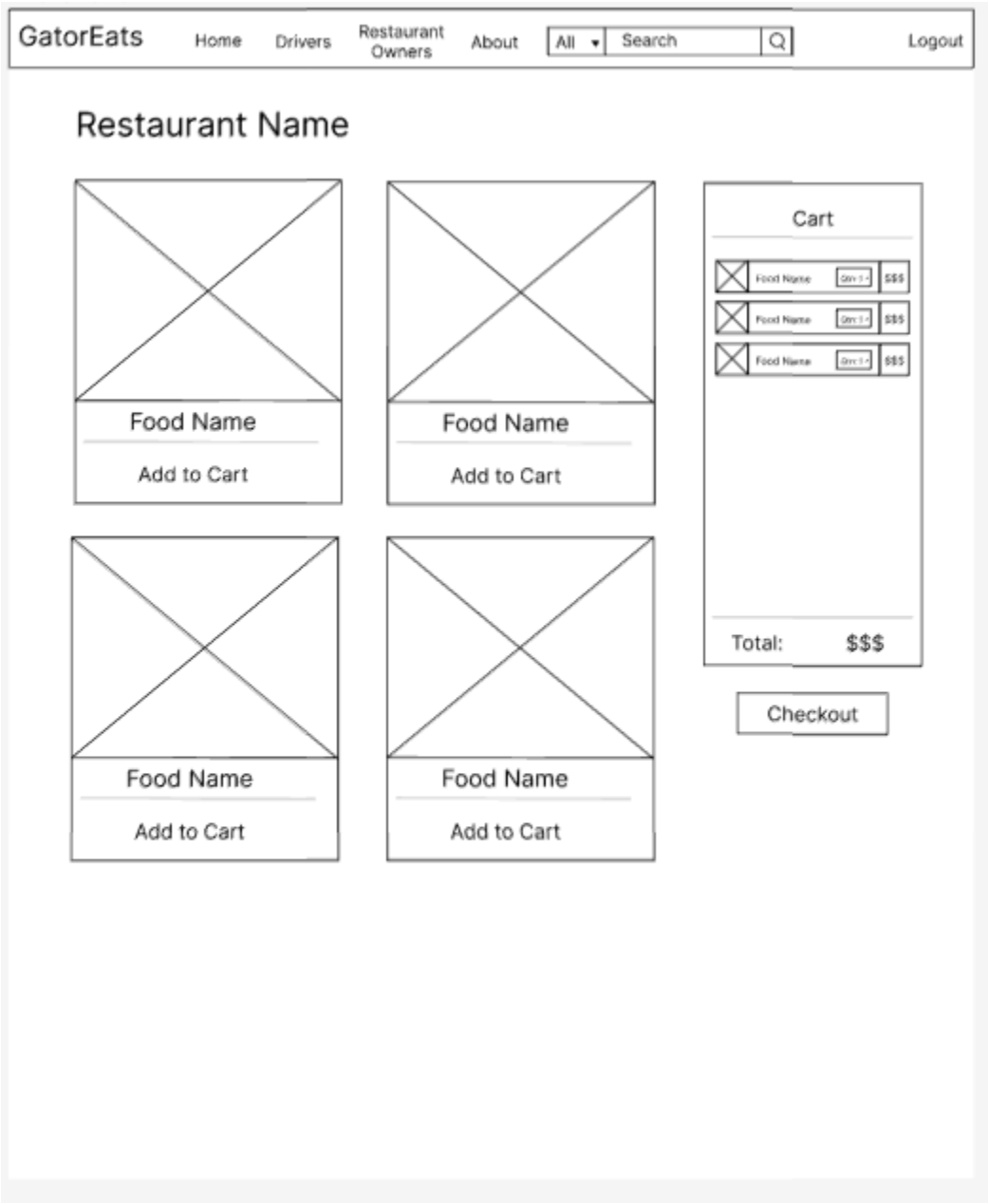
Login Page


GatorEats
Home
Drivers
Restaurant Owners
About
All
Search
Register
Login
Login
Email
Password
Submit
Don't have an account? Register here →

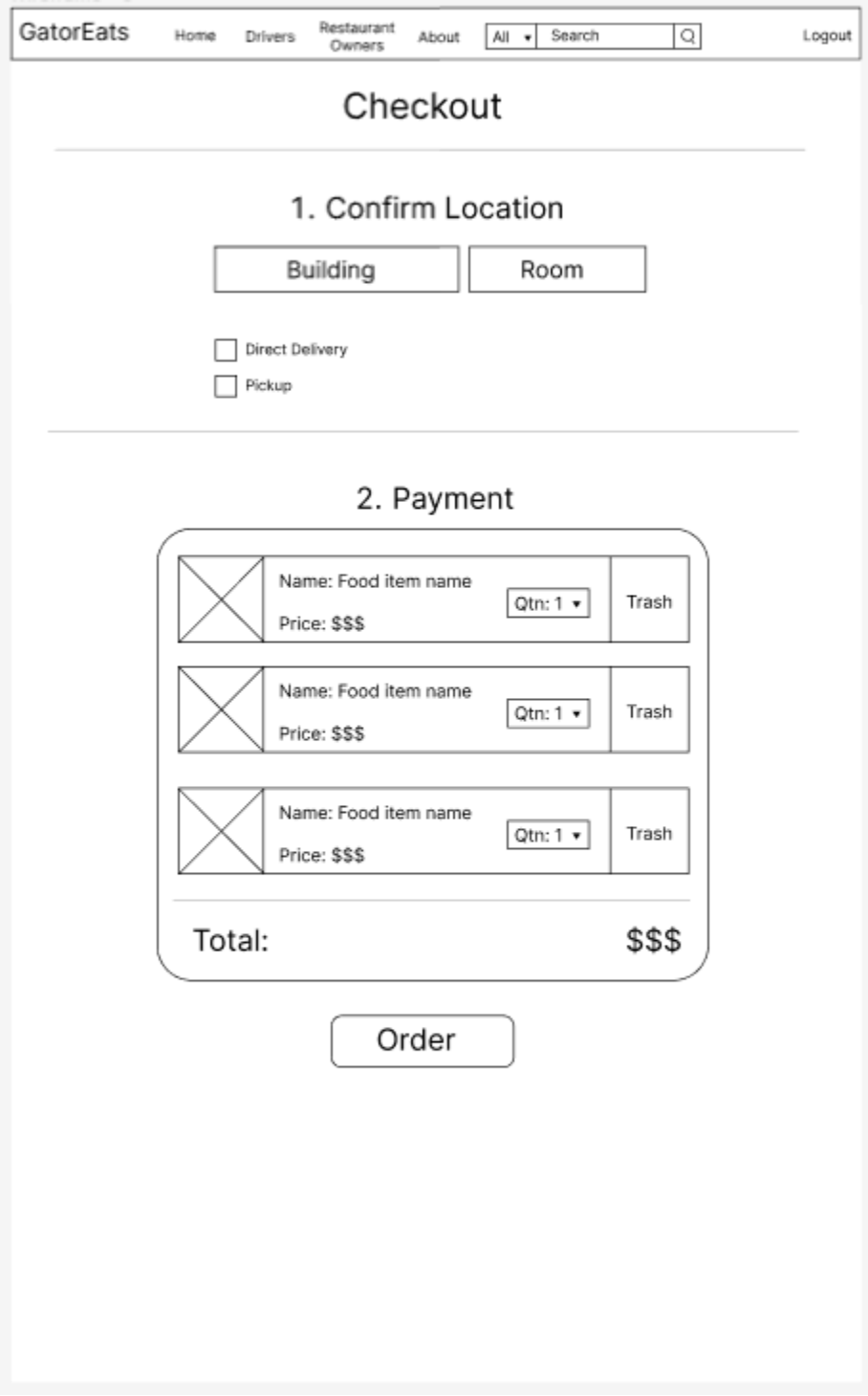
Restaurant List Page



Menu and Cart

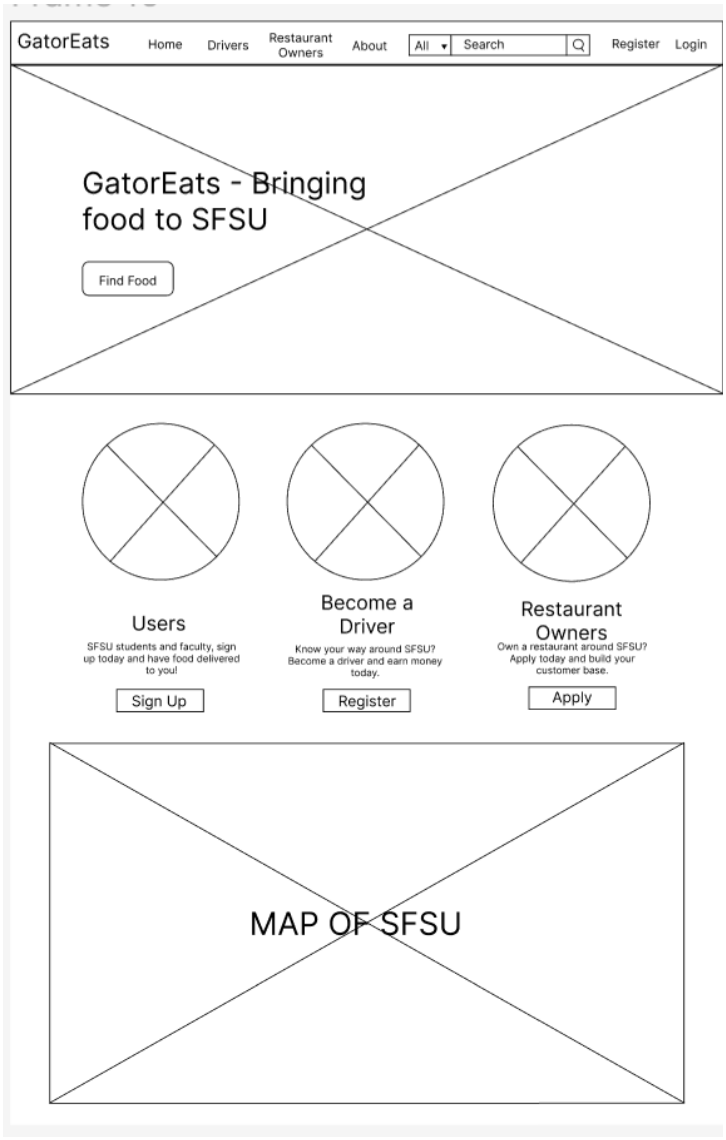


Checkout Page

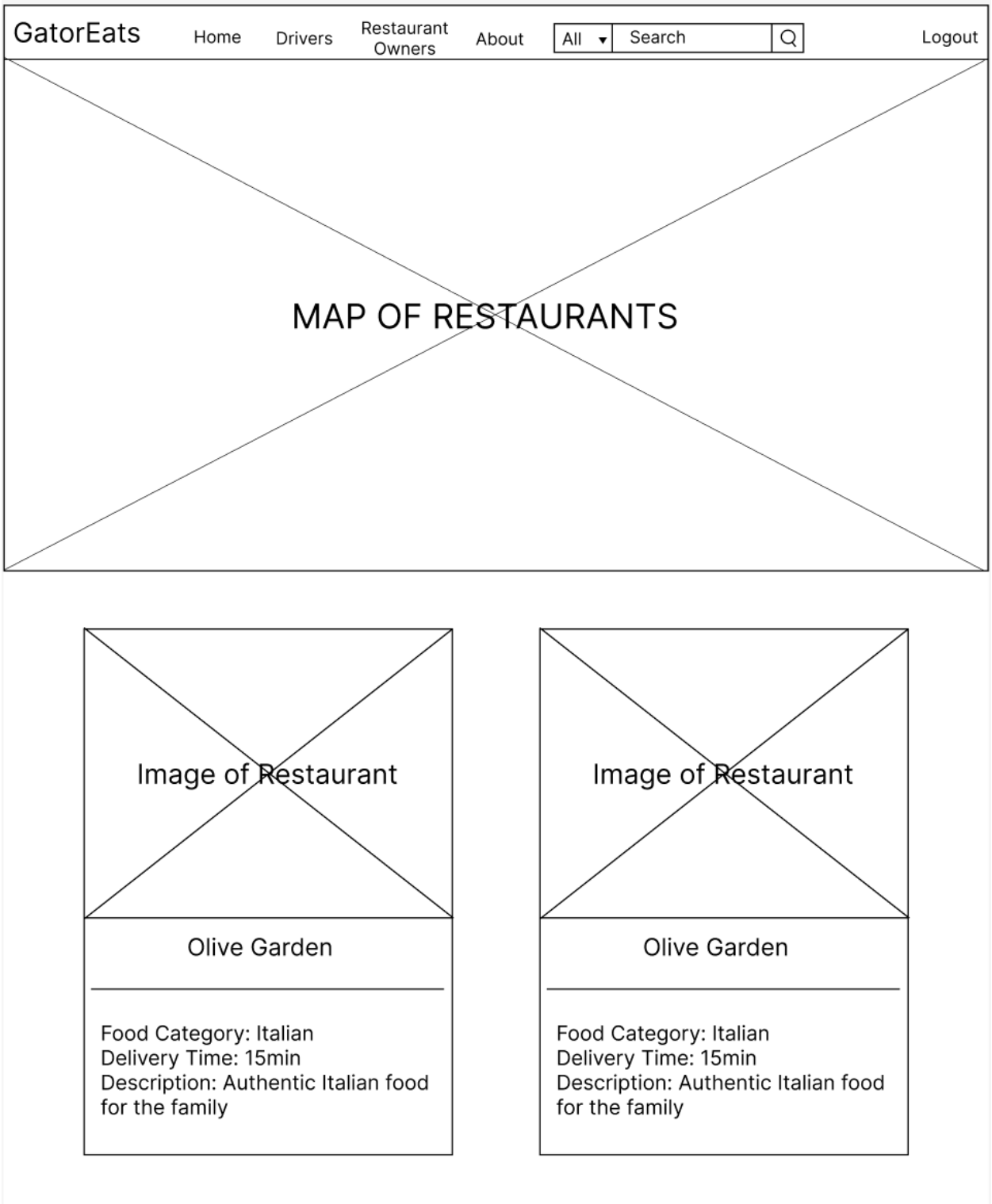


Student - Lazy Registration

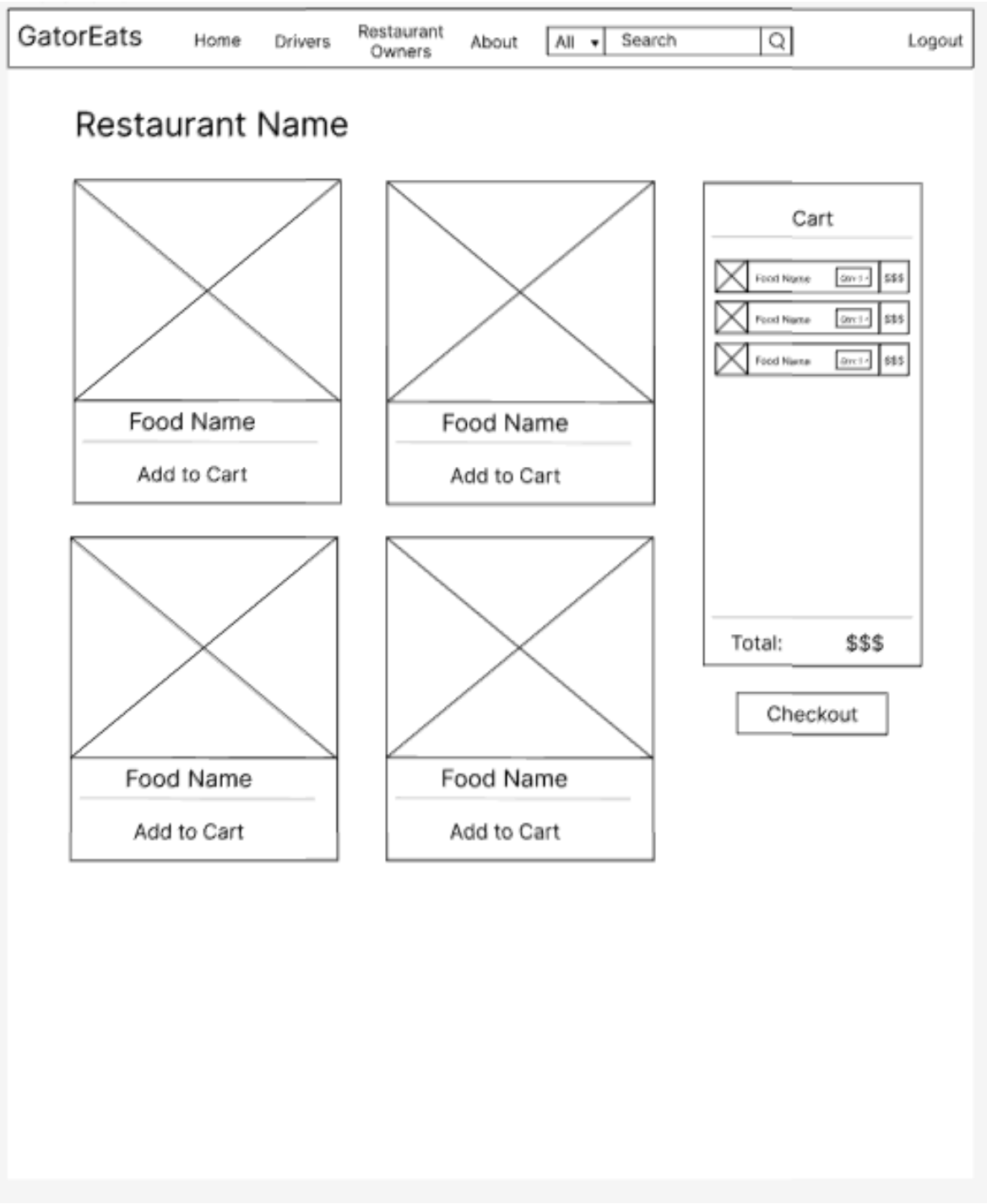
Home Page



Restaurant List Page



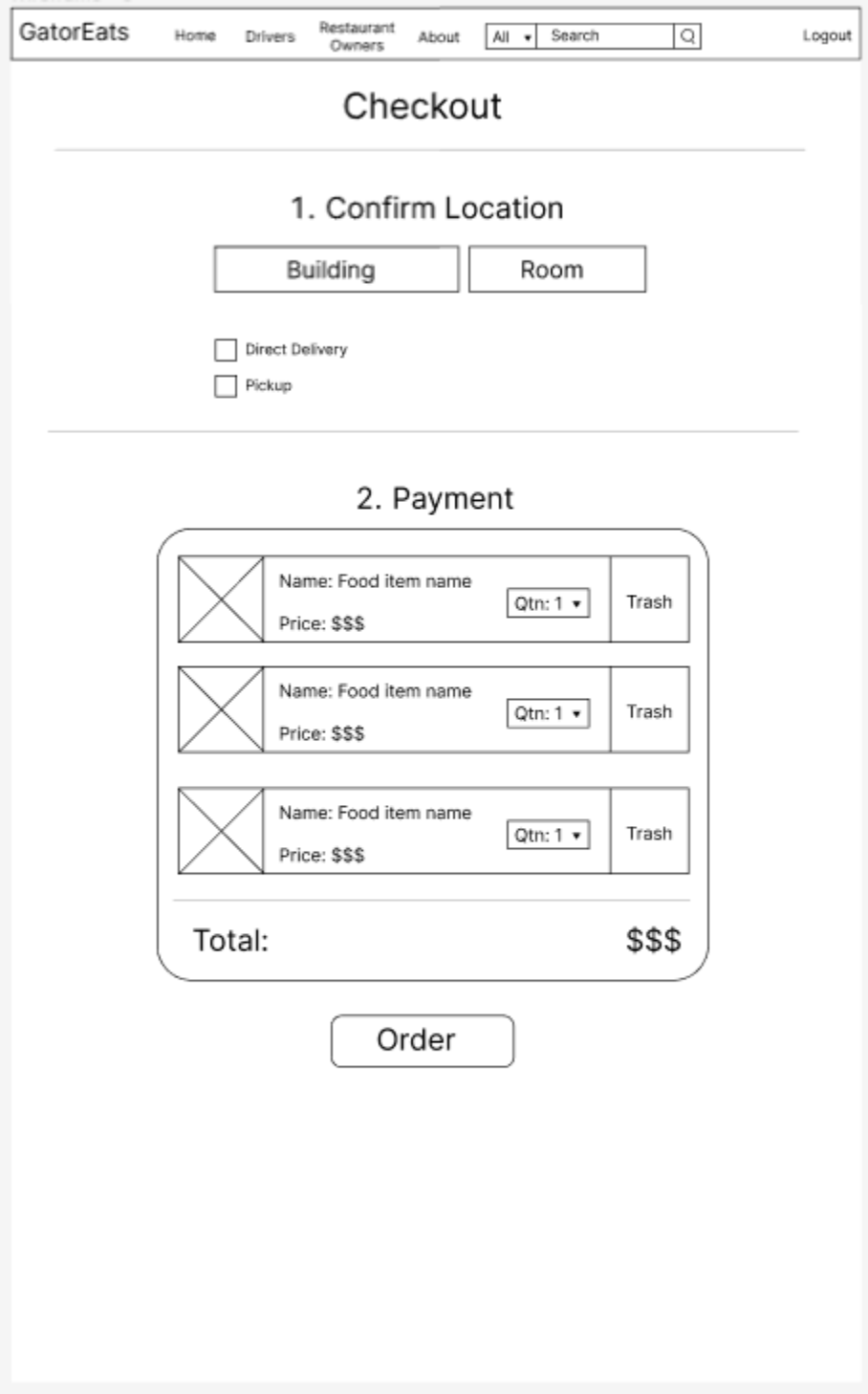
Menu and Cart



Sign Up


GatorEats
Home
Drivers
Restaurant Owners
About
All
Search
Register
Login
SFSU User Registration
First Name
Last Name
SFSU Email
Password
Confirm Password
Pay with GatorCard
By Checking this you agree to the Terms and Conditions
Submit
Have an account? Log in here →

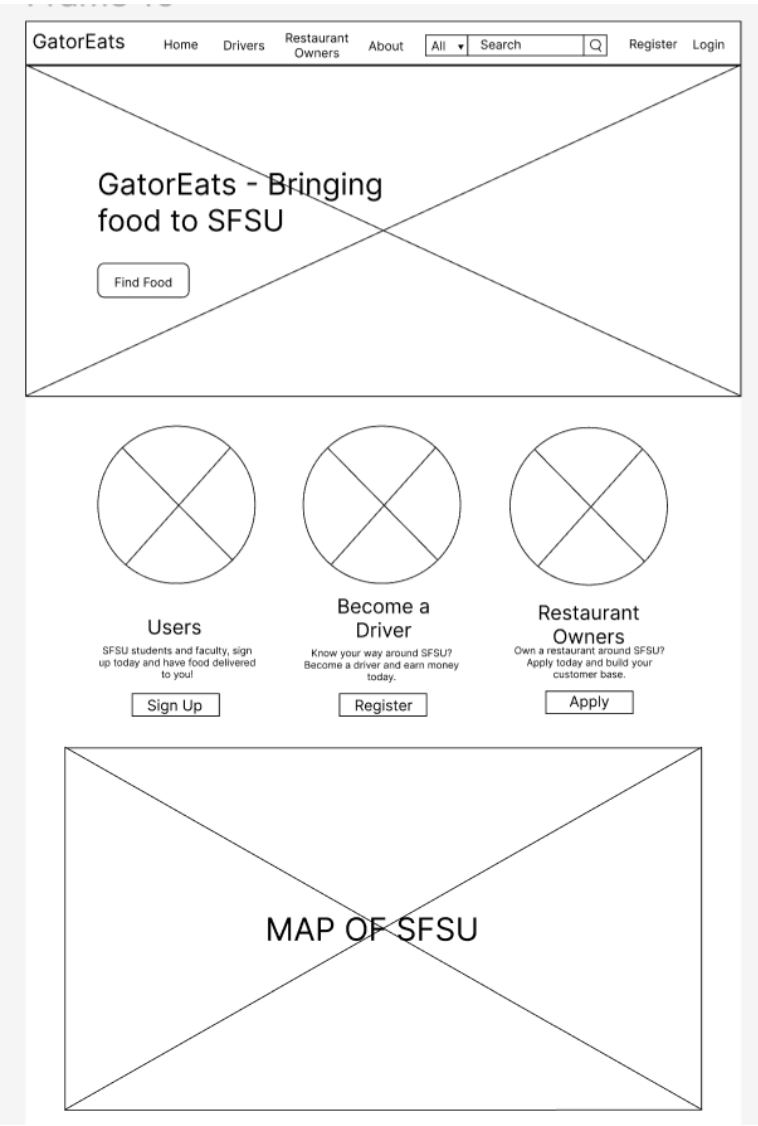
Checkout


GatorEats
Home
Drivers
Restaurant Owners
About
All
Search
Logout
Checkout
1. Confirm Location
Building
Room
Direct Delivery
Pickup
2. Payment
Name: Food item name
Price: $$$
Qtn: 1
Trash
Name: Food item name
Price: $$$
Qtn: 1
Trash
Name: Food item name
Price: $$$
Qtn: 1
Trash
Total:
$$$
Order

Driver - Applying as a Driver and Taking an Order

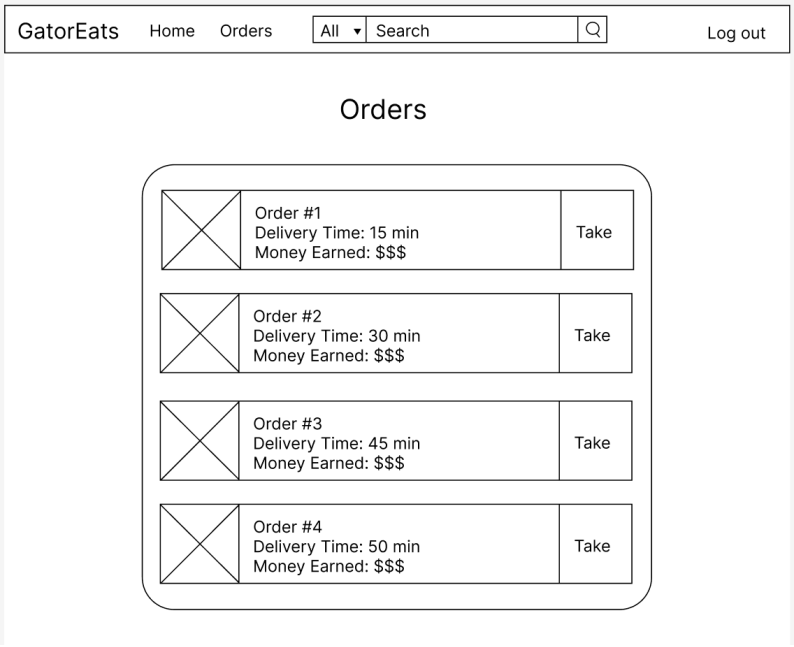
Joe is a SFSU student who wants to earn money on the side, but doesn't want to work far from campus so he decides to become a driver on Gatoreats. He goes to the website and signs up. After signing up he is directed to a list of orders he can take. He takes the first order and reviews the order details. He confirms that he is taking the order and chooses to transfer money earned via GatorCard. He is shown a map with directions and when he is done with the order he confirms that it is completed.

Home Page



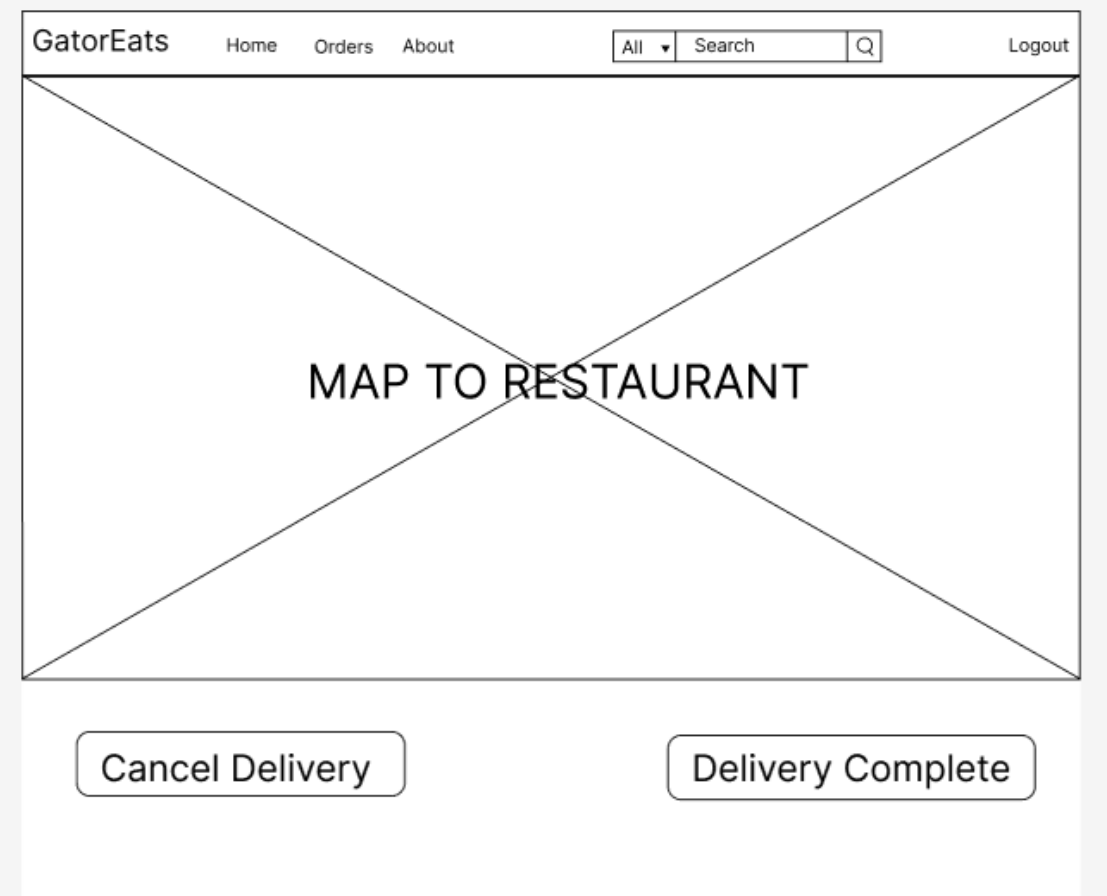
Drivers Registration Page

Orders Page


GatorEats
Home
Orders
All
Search
Log out
Orders
Order #1
Delivery Time: 15 min
Money Earned: $$$
Take
Order #2
Delivery Time: 30 min
Money Earned: $$$
Take
Order #3
Delivery Time: 45 min
Money Earned: $$$
Take
Order #4
Delivery Time: 50 min
Money Earned: $$$
Take

Review and Confirm Order Details Page

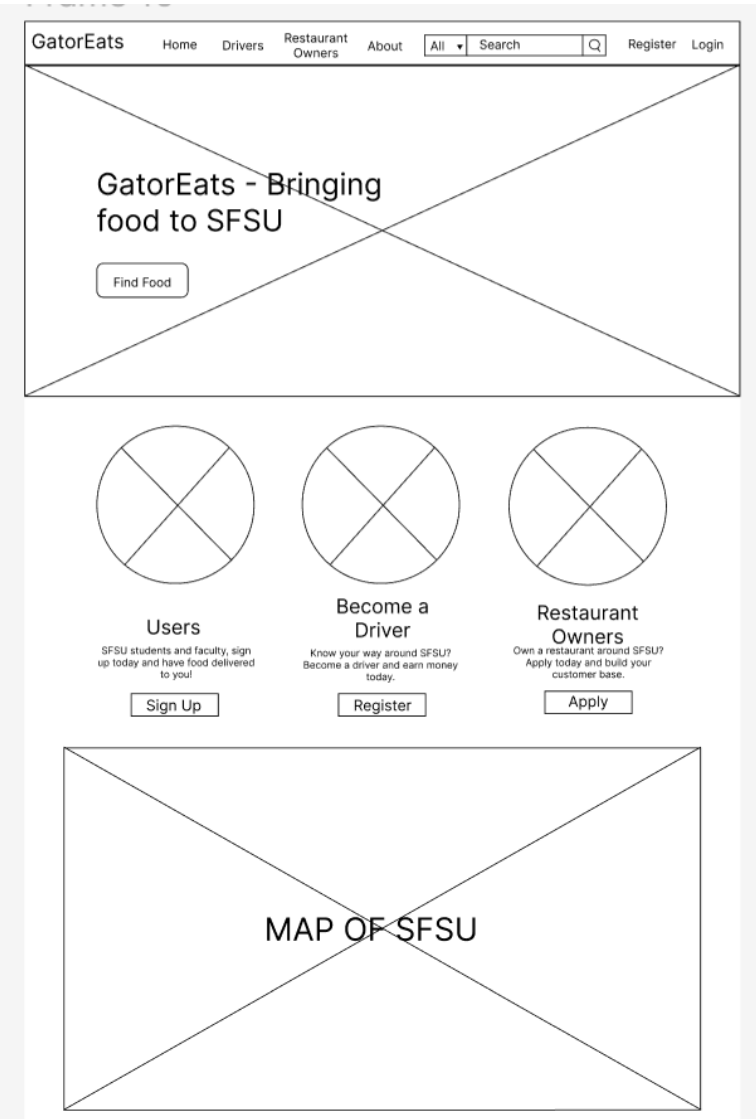
Map and Delivery Confirmation Page



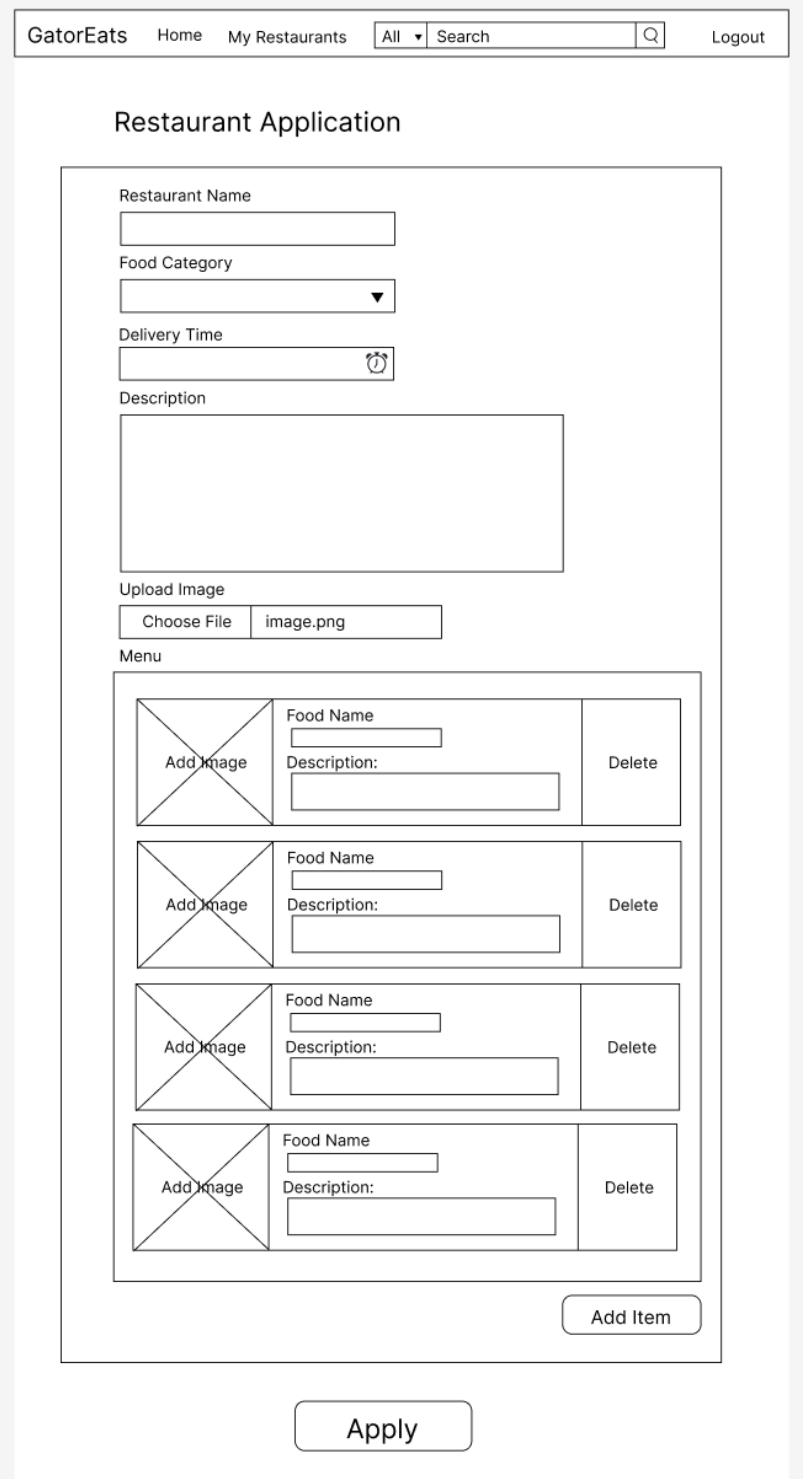
Restaurant Owner - Registering a Restaurant and Becoming a Restaurant User

Reah owns a restaurant near SFSU and wants to expand her customer base to include its students so she registers for GatorEats. She heads to the website and goes to register. She fills in the restaurant application and applies. She is then asked to make an account as a restaurant owner. After registering, she sees a list of all her registered restaurants. The one that just applied is pending to be approved. Checking in the next day shows her restaurant was approved.

Home Page

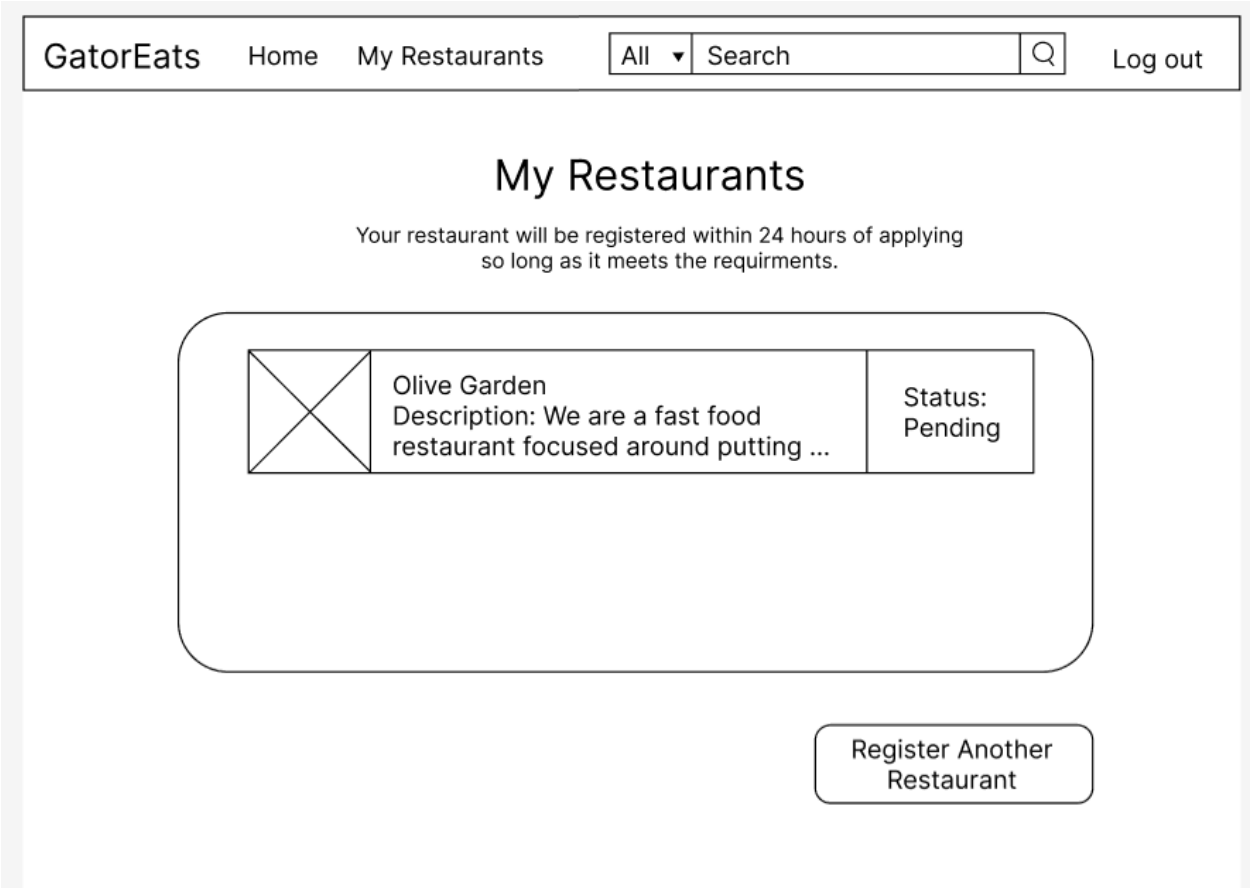


Restaurant Application Page



Restaurant Owner Sign in Page

Pending and Registered Restaurants Page



High Level Architecture, Database Organization Summary

DB Organization:

1. RegisteredUsers
    - UserID PK int
    - Name varchar
    - Password varchar
    - VerifiedEmail varchar
    - SFSUID varchar
    - Order FK
    - GatorPass varchar
    - Ticket FK
2. RestaurantAccount
    - RestaurantID PK int
    - Name varchar
    - Email varchar
    - Password varchar
    - Menu FK
    - Photo
    - Ticket FK
    - Category
3. Admin
    - AdminID PK int
    - Name varchar
    - Password varchar
4. Driver
    - DriverID PK int
    - Name varchar
    - Email varchar
    - Password varchar
    - OrderID
5. Order
    - OrderID PK int
    - CustomerID int
    - Dropoff
    - Total float
    - OrderPlaced

    - OrderDate
    - RestaurantName varchar
    - Driver FK
6. Ticket
    - OrderList ID
    - Number auto increment
    - OrderID FK
7. Menu
    - MenuID PK int
    - MenuList FK
8. MenuList
    - FoodName
    - FoodDescription
    - Photo
    - Price
9. RestaurantList
    - RestaurantID PK int
    - Name varchar
    - Category FK
    - Distance
    - Description
    - Latitude
    - Longitude
10. Category
    - CategoryID PK int
    - CategoryName varchar
11. DropOffPoints
    - PointID PK
    - Name
    - Latitude
    - Longitude
12. Building
    - BuildingID PK int
    - Name varchar
    - Latitude
    - Longitude
13. Room
    - RoomID PK int
    - RoomNumber int
    - Building FK

Media Storage:
We plan on keeping images in a file system and storing the absolute path to these files in the database.

Search/Filter Architecture:
For both the search functionality, our team will be using the SQL's function "SELECT * from the database table, WHERE “user input ”LIKE %" to determine whether or not the user input is similar to the restaurants that we have in our database. For the filter category functionality, we would use the “SELECT * from the database table, WHERE = “category”” to retrieve all the restaurants with the proper category.
If it matches, the resulting restaurants will then be shown on a separate page. The restaurants names and categories are all columns inside of a table called restaurants.

Key Risks

Schedule Risks: We all have very different schedules, which makes it hard for all of us to meet at once outside of class. Certain times work for some people but not for others. This can be addressed by having a good communication on Discord where we inform each other of our current progress. Finally we can limit the scope of the project further if time is of the essence.

Legal/content risks: When creating restaurants, we are worried that we might use copyrighted images for the restaurants. We can solve this by only using images from website likes, https://pixabay.com/, that provide non copyrighted images.

Technical Risks: We need to learn how to use multer and sharp so that users can upload images and we can make thumbnails out of their images. We also need to learn how to use Google’s geocoding API to translate restaurant addresses to coordinates to show on the Google maps API.

Skills Risks: We don’t have much experience building a functional website from scratch, most of us haven’t used a Stack before. Due to this, there is a really big learning curve in this project. We will try to make sure to take each step slowly, work in a small scope, and assure that each stage is fully working before moving to the next one.

Project Management

To manage the current list of tasks that are involved with Milestone 2 Part 1 and Part 2 as well as Future tasks and Milestone, the main source of organization will come from the use of constant communication and organized checklists. To make sure we are all on the same page, our team will be having weekly meetings over Discord as well as the in person sessions of class to ensure that we are able to speak together on topics regarding progress, ideas, worries, and project understanding. Our team will also be using the project organization app, Trello, to manage and handle our large list of tasks that are required to be completed throughout the course of the semester. Our team lead will build checklists and organize the required milestones and tasks within the milestones into easy to complete checklists. These tasks within these checklists will be delegated in order to ensure that there is a level of accountability and responsibility. Through the symbiotic use of both meetings and Trello, We are confident we will be able to complete our required tasks in a timely and professional manner.