# 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: | N/A |

# 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

Registered User: A general user that has registered shall be considered a registered user,. They must log in to access their account. A registered user shall opt into becoming a delivery driver or restaurant account, or remain as just a registered user. An admin shall be required to register an account. Faculty, staff and students must register using their sfsu.edu email or mail.sfsu.edu email.

Admin: An admin is a registered user that has the ability to approve restaurants and images uploaded that will be displayed.

Driver: A driver is a registered account that shall have access to a registered user's order. They are responsible for the delivery and the completion of an order. They are also able to view the restaurant's pickup point to pick up the food, as well as the registered user's drop off point to drop off the food.

Restaurant Account: A general user that shall create a restaurant account. A restaurant account can build a restaurant within their account, and manage their information. A restaurant account shall have its own menu, which has various menu items, as well as an orderlist to view all the current pending orders that are needed to be fulfilled.

Order History: Registered users shall be able to view all their past orders. Restaurants shall be able to view all orders made to their restaurant by the customers.

Order: An entity that contains a 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.

Ticket: A list that contains menuitems. The ticket is linked to an order.

Menu: A list that contains menuitems 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

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 a registered account. They shall submit a name while registering. They shall have at least one location and at least one phone number. They shall be able to submit relevant photos and menu of their restaurant

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 unregistered, registered, admin, 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:**

**Unregistered Accounts:**

1.1 An unregistered user shall be able to register for an account to browse the items on the website.

1.2 Upon successful registration, the user shall be granted access to browse the items on the website.

1.3 During the registration process, the user shall not be required to provide any information that is not necessary for the account creation, as per the website's privacy policy.

1.4 An unregistered user shall be able to search using food categories or cuisines since searching is a core function of the application, and users must be able to find the desired restaurants based on their food preferences.

**Registered Accounts:**

2.1 A registered user shall be able to log into their account, as logging in is essential for registered users to access their account and use all application features.

2.3 A 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.

2.5 A registered user shall be able to order food since ordering food is the application's primary purpose, and registered users must be able to use this feature to keep the business running.

2.6 A registered user shall inherit all functionality of an unregistered user, as all features available to an unregistered user must be available to a registered user to ensure a seamless user experience.
2.7 A registered user shall choose the location where the food will be dropped off since the delivery location is crucial for the restaurant to deliver the food to the correct address.
2.8 A registered user shall have a payment method since the payment method is essential for the user to pay for the food they order and for the restaurant to receive payment for the food.

**Restaurants Accounts:**

3.1. A restaurant account shall be able to register their account since registration is a crucial step for restaurants to create their account and start using the application's features.
3.2. A restaurant account shall be able to view orders since restaurants must view and manage their orders to fulfill customer requests and ensure timely delivery.
3.4. A restaurant account shall be able to set the prices for their dishes since the restaurant needs to set prices for their dishes to receive payment for the food and manage their business finances.
3.5. A restaurant account shall be able to accept food requests, as accepting food requests is the primary function of the application for restaurants, and it is essential to keep the business running.
3.6. A restaurant account shall be able to add or delete items from their menu since restaurants must be able to update their menu as needed to reflect changes in their offerings and keep their menu up to date.
3.8. A restaurant account shall be able to add one or more addresses since restaurants must be able to add addresses to receive delivery requests and manage their delivery logistics.

**Admin Accounts:**

4.1. An admin shall be the only user that shall approve restaurants since the admin's approval is crucial for maintaining the quality and reputation of the application and ensuring that all restaurants meet the necessary standards.

**Driver Accounts:**

5.1. A driver shall be able to register as a delivery driver since drivers are crucial for delivering the food and ensuring customer satisfaction.

5.2. A driver shall be able to receive a delivery order since this is the primary function of the driver and essential for the application's core functionality.

**Payment method user:**

6.1 A payment method shall be linked to the customer's gator pass since a reliable and secure payment method is crucial for the application's core functionality and for ensuring customer satisfaction. Linking the payment method to the customer's gator pass can provide a convenient and secure payment option for customers and simplify the payment process for both customers and the application.

**Orders:**

7.1. An order shall have the time the order was placed, the restaurant name, the estimated time of arrival, the price, the food that was ordered, and an SFSU map where the order shall be delivered since this information is crucial for the customer to track their order and ensure timely delivery. This feature is also essential from a business perspective as it can help manage orders and ensure timely delivery.

**Menu:**

9.1. A menu shall have one or more food options, descriptions, and prices since the menu is the application's core feature, and the customer should be able to view the available food options, descriptions, and prices. This feature is crucial for usability, marketing, and business aspects.

**Photo:**

10.2. Approval by an admin is a must-have requirement as it is essential for maintaining the quality and appropriateness of the photos displayed in the application.

10.3. Viewability by any general user after approval is a requirement, as the main purpose of displaying photos in the application is to make them accessible to the users. It is essential for the usability and success of the application.

**Category:**

11.1. Linking a category to zero to many restaurants is a must-have requirement as it is essential for organizing and categorizing the restaurants in the application. This feature will help users easily find and navigate to the restaurants they are interested in.

**Pickup Point:**

12.2. A pickup point must have coordinate points or a building name and a room number for accurate location identification. This is a must-have requirement for the successful delivery of goods.

12.3. A pickup point must be a point on the SFSU map that would be used for meetup/dropoff of delivery. This is a must-have requirement for the successful delivery of goods.

**Rooms:**

13.1. Defining a room as a pickup point with a building name is a must-have requirement as it provides a more specific location for pickup points, making it easier for delivery drivers to locate them accurately. This will improve the overall efficiency and success of the delivery process.

**Priority 2:**

**Unregistered Accounts:**

1.1 An unregistered user shall be able to browse the restaurants.

1.2 An unregistered user shall be able to sort by delivery time and price.

**Registered Accounts:**

2.2 A registered user shall be able to remain a customer as it is necessary for registered users to remain customers to keep the application's user base and promote repeat business.

2.3 A registered user shall be able to reset their password as it is essential to have a password reset function for users who forget their password and need to regain access to their account.

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

**Restaurants Accounts:**

3.3. A restaurant account shall upload many photos of their dishes, as having multiple photos of dishes can attract customers to the restaurant's menu and improve the application's user experience.

3.7. A restaurant account shall be able to add a phone number, as having a phone number can improve the customer experience by providing a way for customers to contact the restaurant directly.

3.9. A restaurant account shall link itself to up to three chosen categories, as linking to categories can improve the restaurant's visibility in search results and make it easier for customers to find them based on their cuisine type.

**Admin Account:**

4.2. An admin shall be able to approve uploaded photos by restaurant owners, as having approved photos can improve the customer experience by providing high-quality visuals of the restaurant's dishes and atmosphere.

**Driver Account:**

5.3. A driver shall be able to view the map of the school, as this feature can improve the driver's ability to navigate to the delivery location and ensure timely delivery.

**Order Account:**

8.1. An order list shall contain menu items and a map of where the order shall be delivered since it is a helpful feature for customers to view their order history and for restaurants to manage their orders. However, this feature is not as critical as the order placement feature.

**Menu Account:**

9.2. A menu shall have one or more relevant pictures since having pictures of the food can enhance the customer's experience and attract them to order from the restaurant. However, this feature is less critical than the menu's core functionality.

**Photo Account:**

10.1 A relevant name for a photo is desired as it improves the usability of the application and helps users to easily identify the photo. However, it is not a must-have requirement as the application can still function without a relevant name for each photo.

**Categories:**

11.2. Allowing users to filter their searches for a restaurant by category is a desired feature as it improves the application's usability and helps users find restaurants that match their preferences. However, it is not a must-have requirement, as users can still search for restaurants without filtering by category.

**Pickup Points:**

12.1. Giving a name to a pickup point is a desired feature as it improves the application’s usability and helps users quickly identify and remember the pickup points they frequently use. However, it is not a must-have requirement as the application can still function without a name for each pickup point.

**Priority 3:**

**Admin Accounts:**

4.3. An admin shall be able to delete users, as the ability to delete users is not essential for the application's core functionality but can be useful for managing user accounts and ensuring the application's security.

**Driver Accounts:**

5.4. A driver shall be able to set the time they are available to deliver, as this feature can be useful for managing the driver's availability and ensuring that there are enough drivers to fulfill orders, but it is not essential for the application's core functionality.

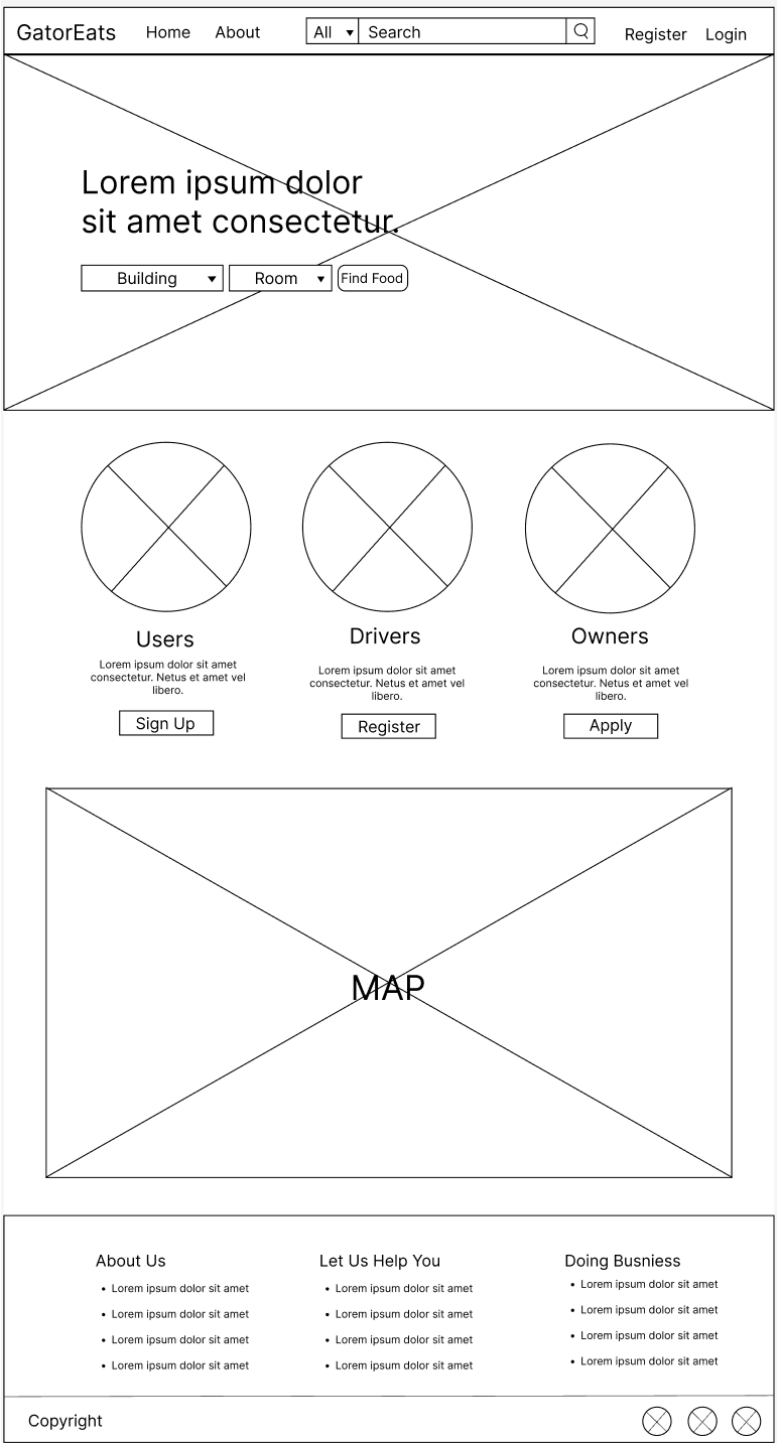
5.5. A driver shall turn their availability status on or off as this feature can be useful for managing driver availability and ensuring that there are enough drivers to fulfill orders, but it is not essential for the application's core functionality.

# UI Storyboards for each main use case

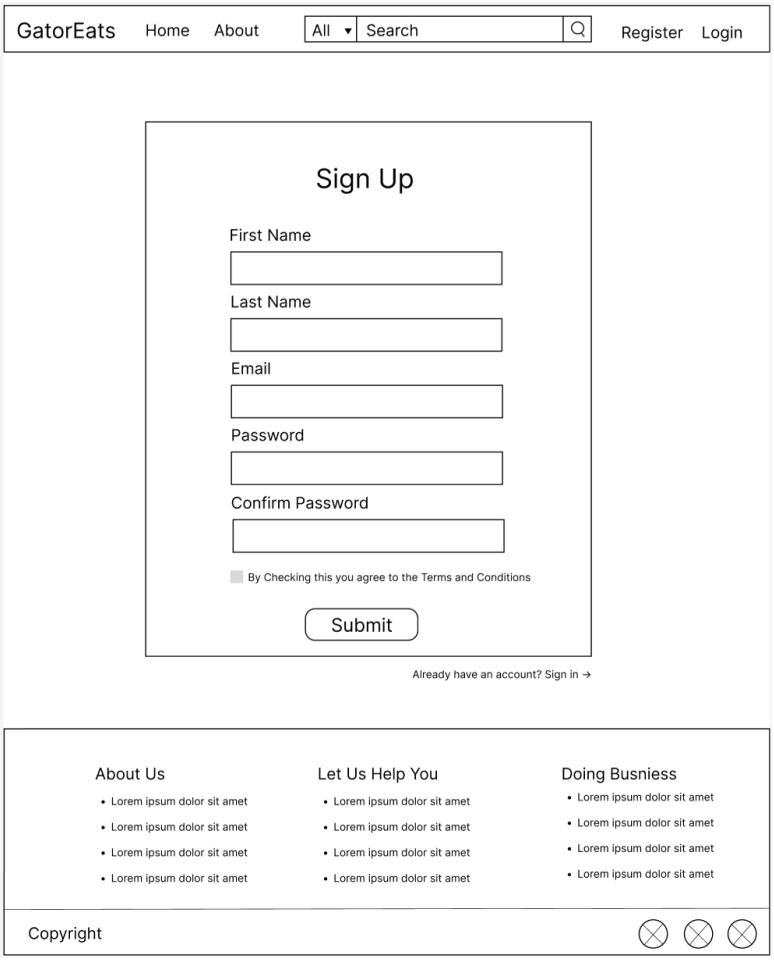
## Faculty

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.

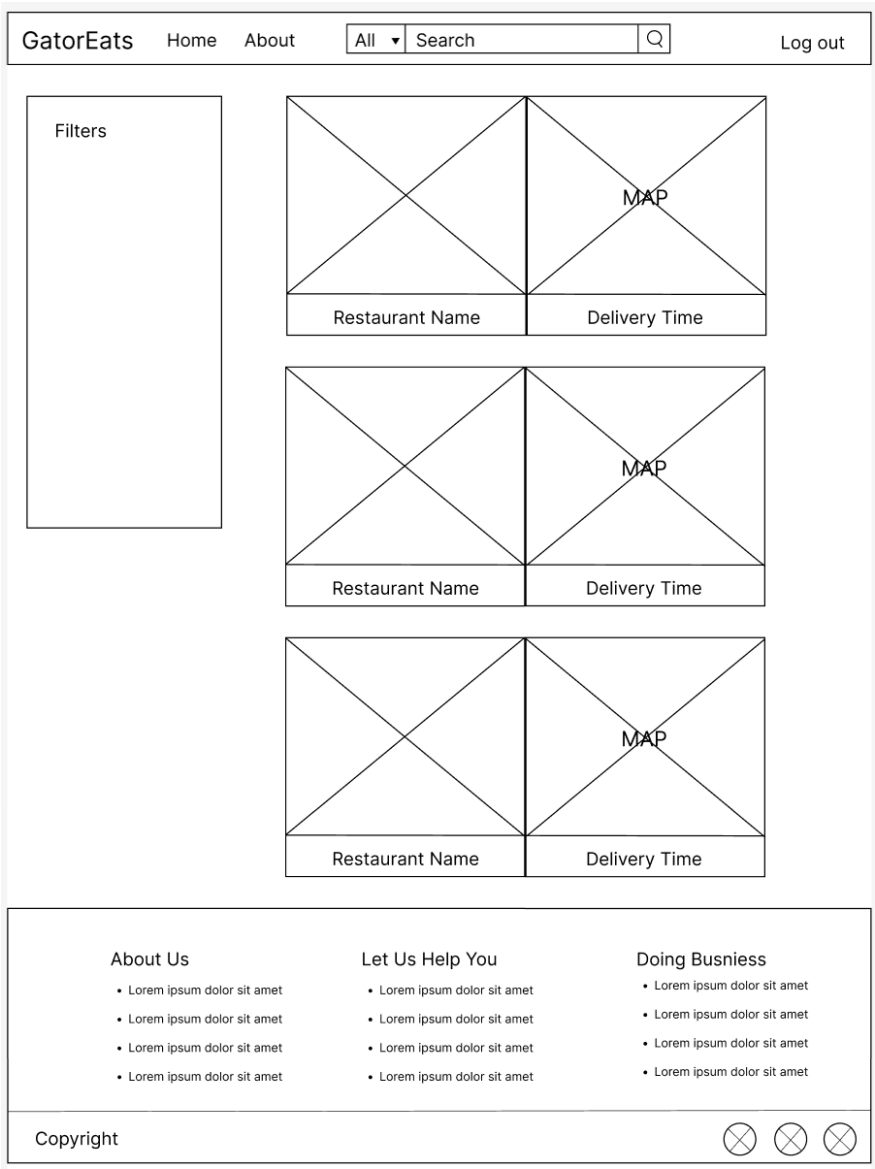
Home Page



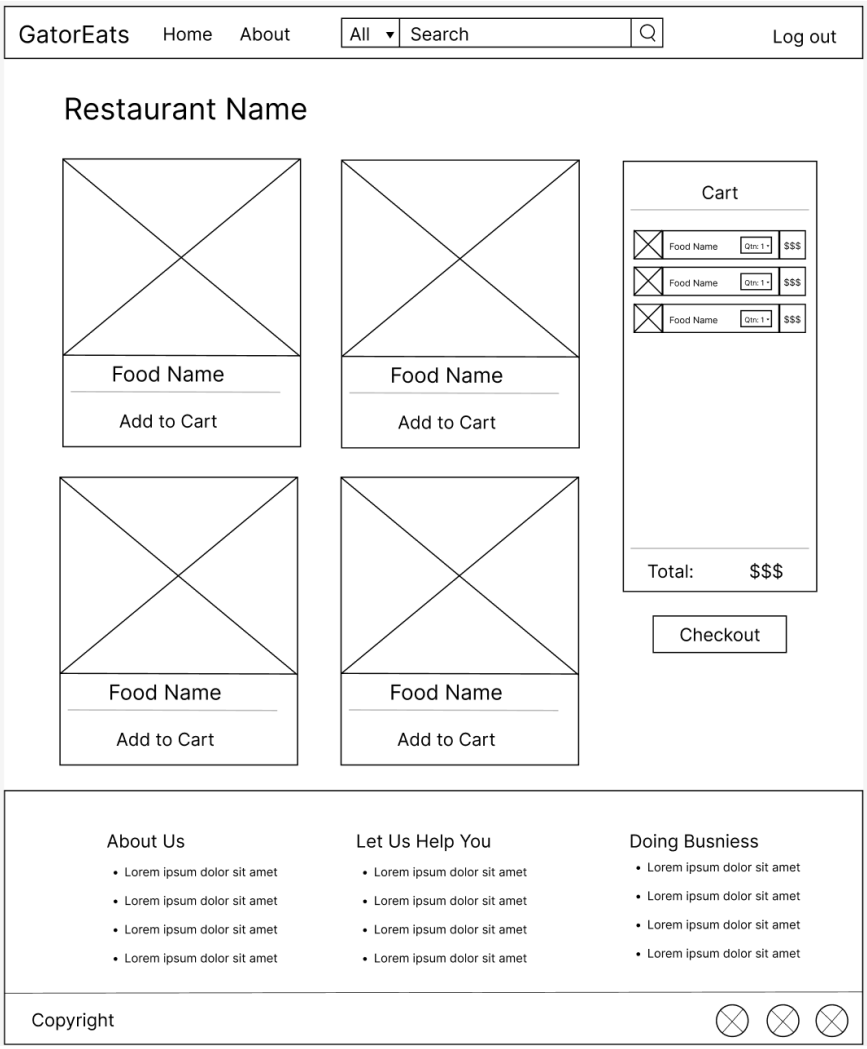
# Sign Up Page


GatorEats
Home
About
All
Search
Register
Login
Sign Up
First Name
Last Name
Email
Password
Confirm Password
By Checking this you agree to the Terms and Conditions
Submit
Already have an account? Sign in →
About Us
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Let Us Help You
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Doing Busniess
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Copyright

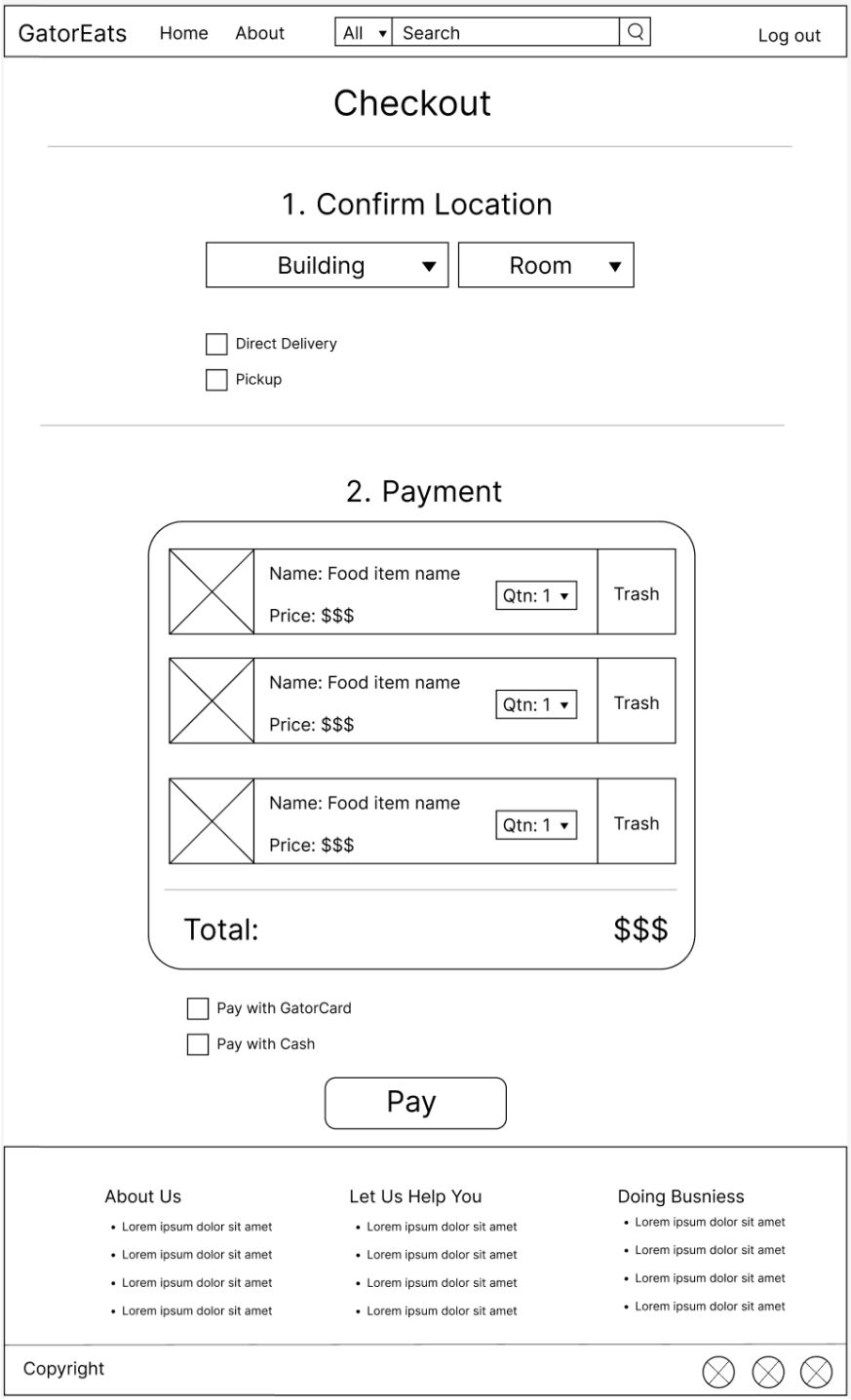
Restaurant List Page



# Menu and Cart



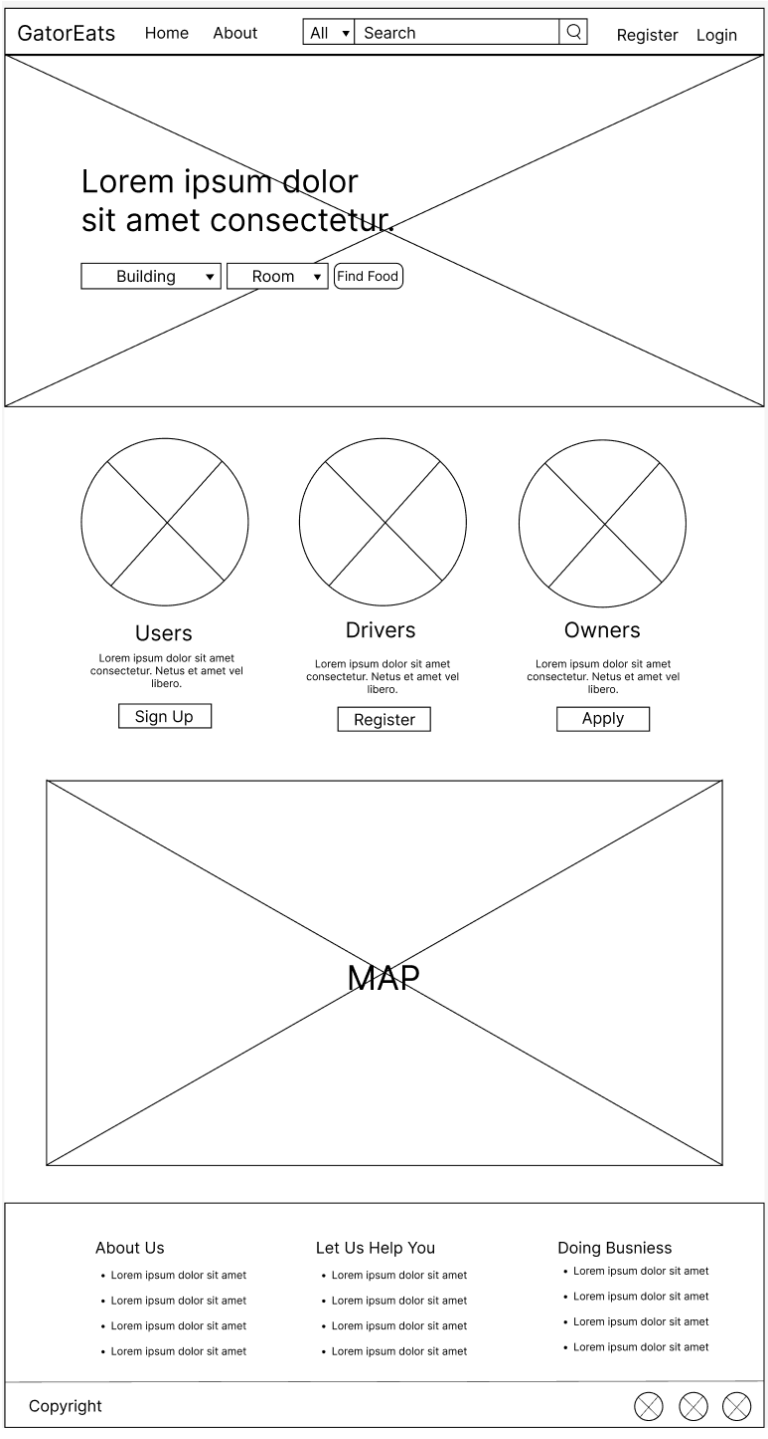
# Checkout Page



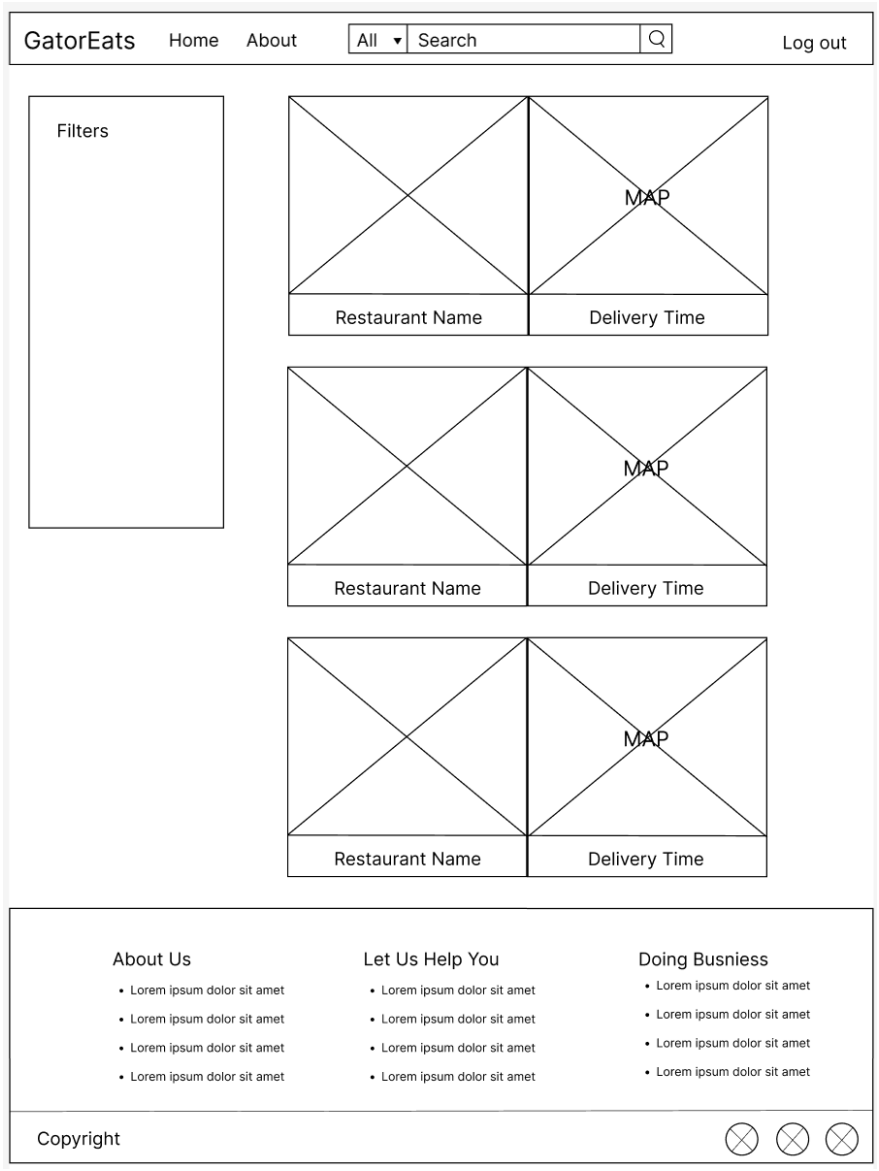
# Student

Zaki is a SFSU student who decides to try out GatorEats because he's often too busy studying, but wants to try new foods. He heads to the website and enters his location. He is then directed to a list of restaurants. He decides to look at pizza places and decides on one with a 15 min delivery time. He looks at the menu and decides on Hawaiian pizza and a mango smoothie. He adds them to his cart and checks out. He is then asked to sign  up. He signs up and checks out. He reconfirms his location and chooses to pay with cash.

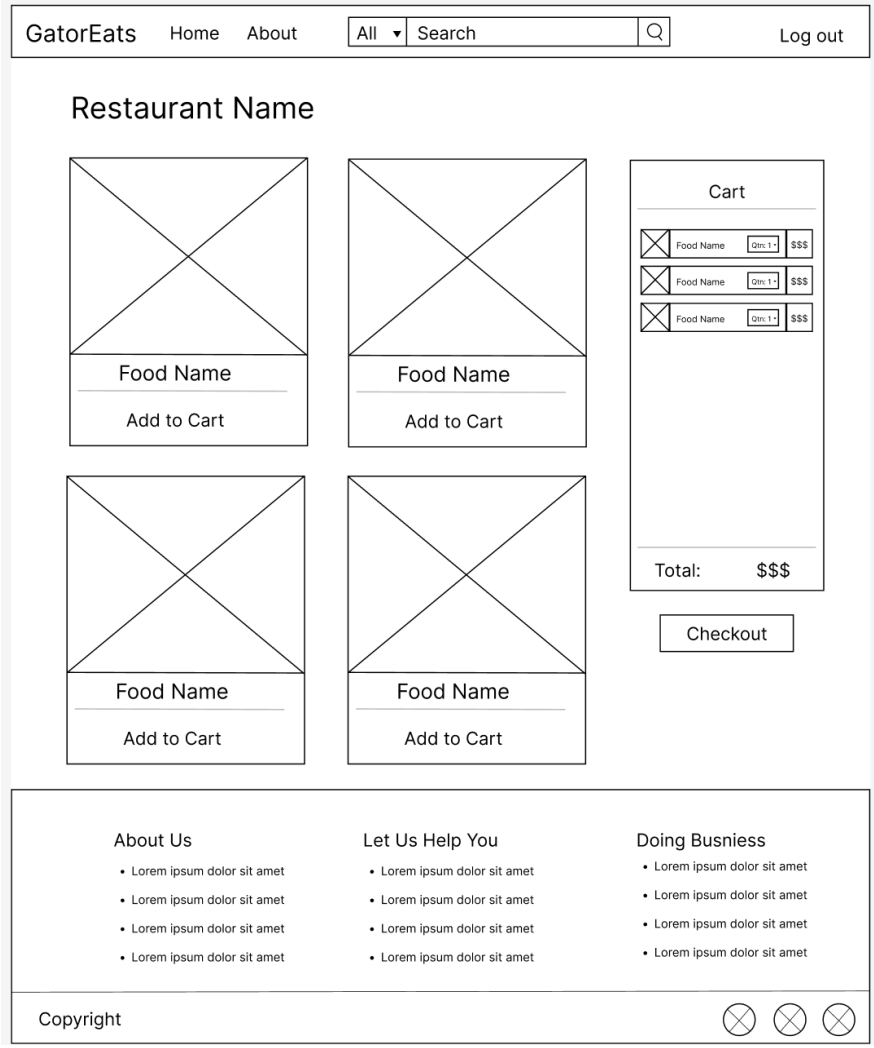
Home Page



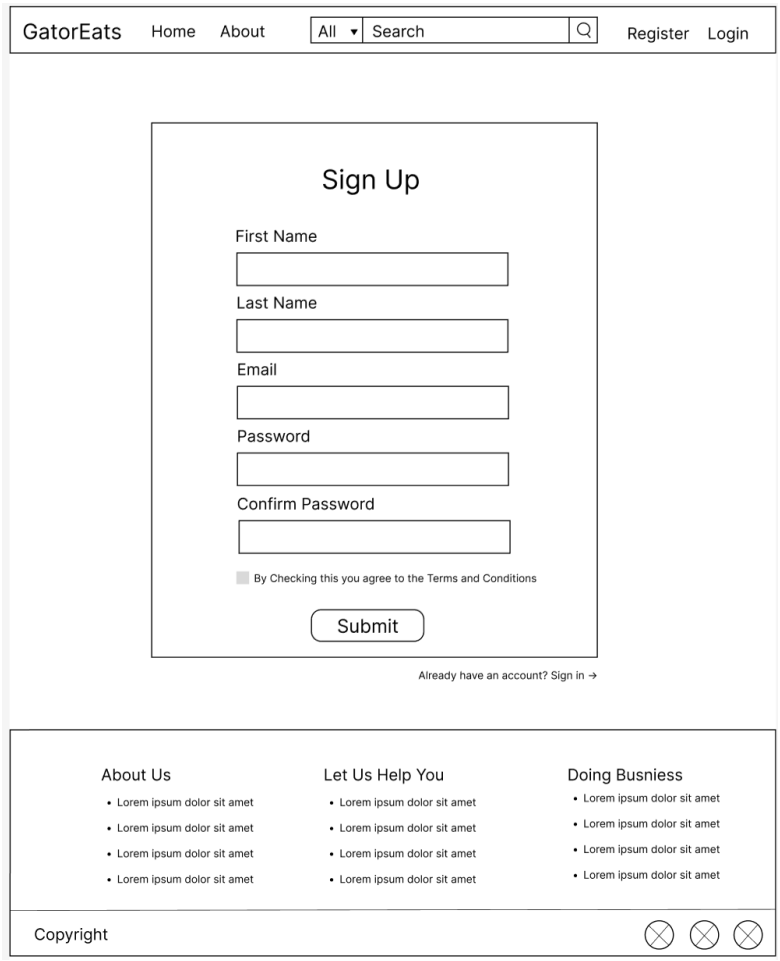
# Restaurant List Page



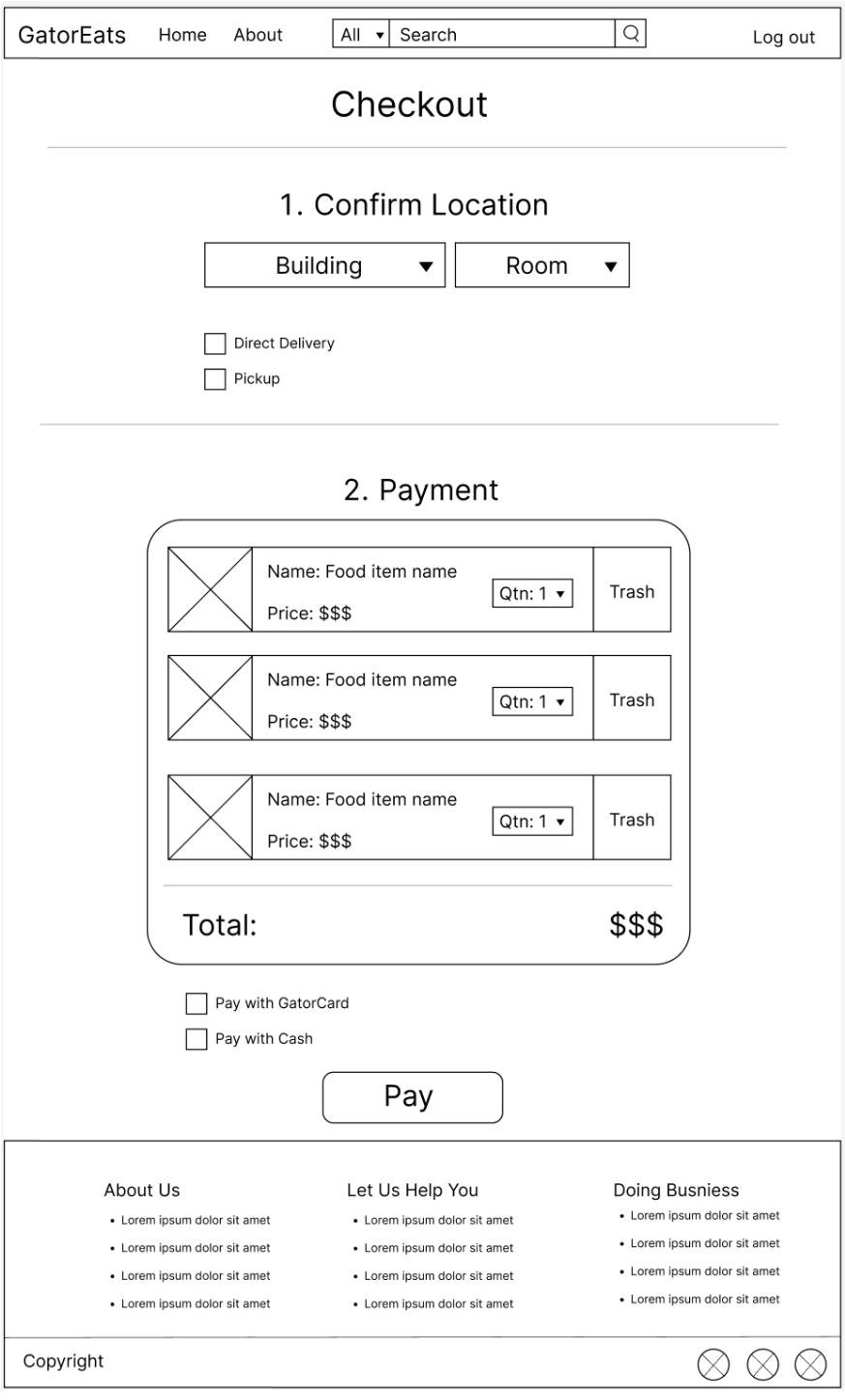
# Menu and Cart



# Sign Up


GatorEats
Home
About
All
Search
Register
Login
Sign Up
First Name
Last Name
Email
Password
Confirm Password
By Checking this you agree to the Terms and Conditions
Submit
Already have an account? Sign in →
About Us
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Let Us Help You
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Doing Busniess
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Copyright

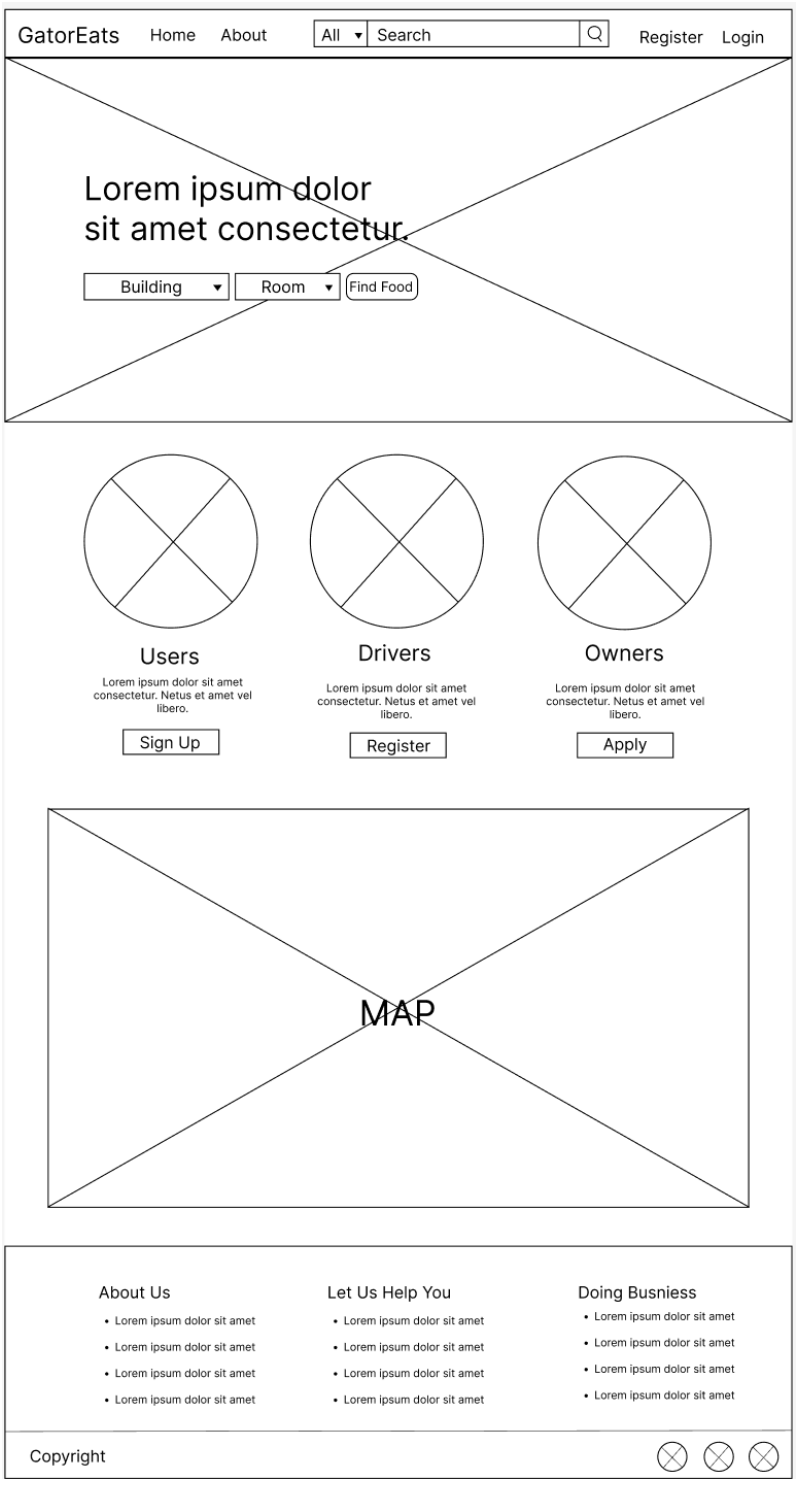
Checkout


GatorEats
Home
About
All
Search
Log out
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: $$$
Qtn: 1
Trash
Total:
$$$
Pay with GatorCard
Pay with Cash
Pay
About Us
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Let Us Help You
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Doing Busniess
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Copyright

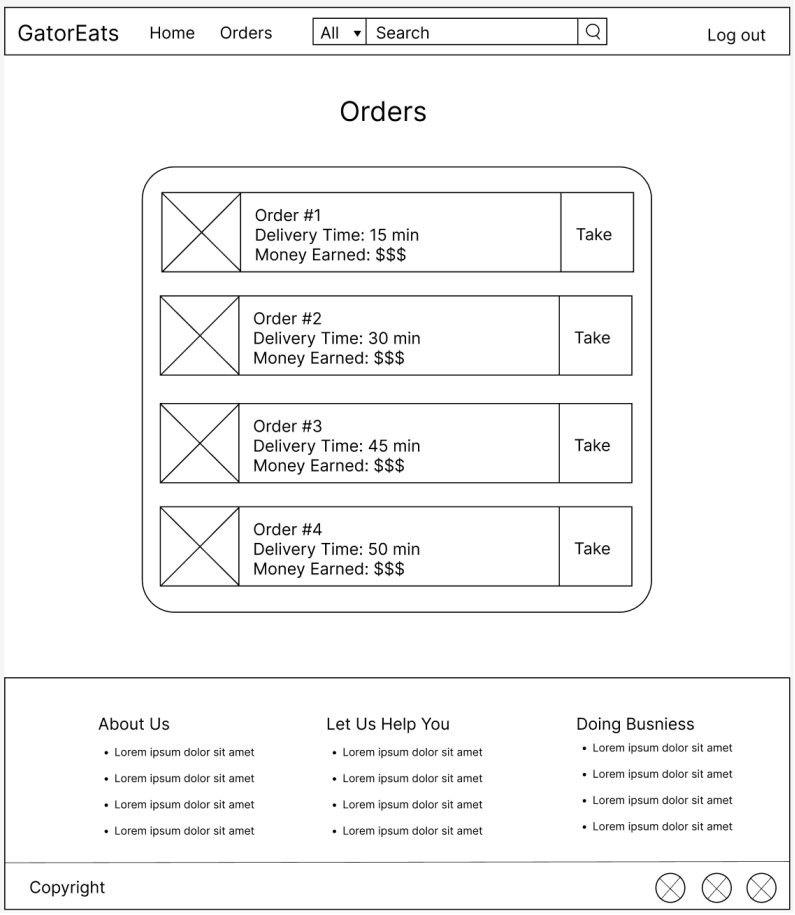
# Driver

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.

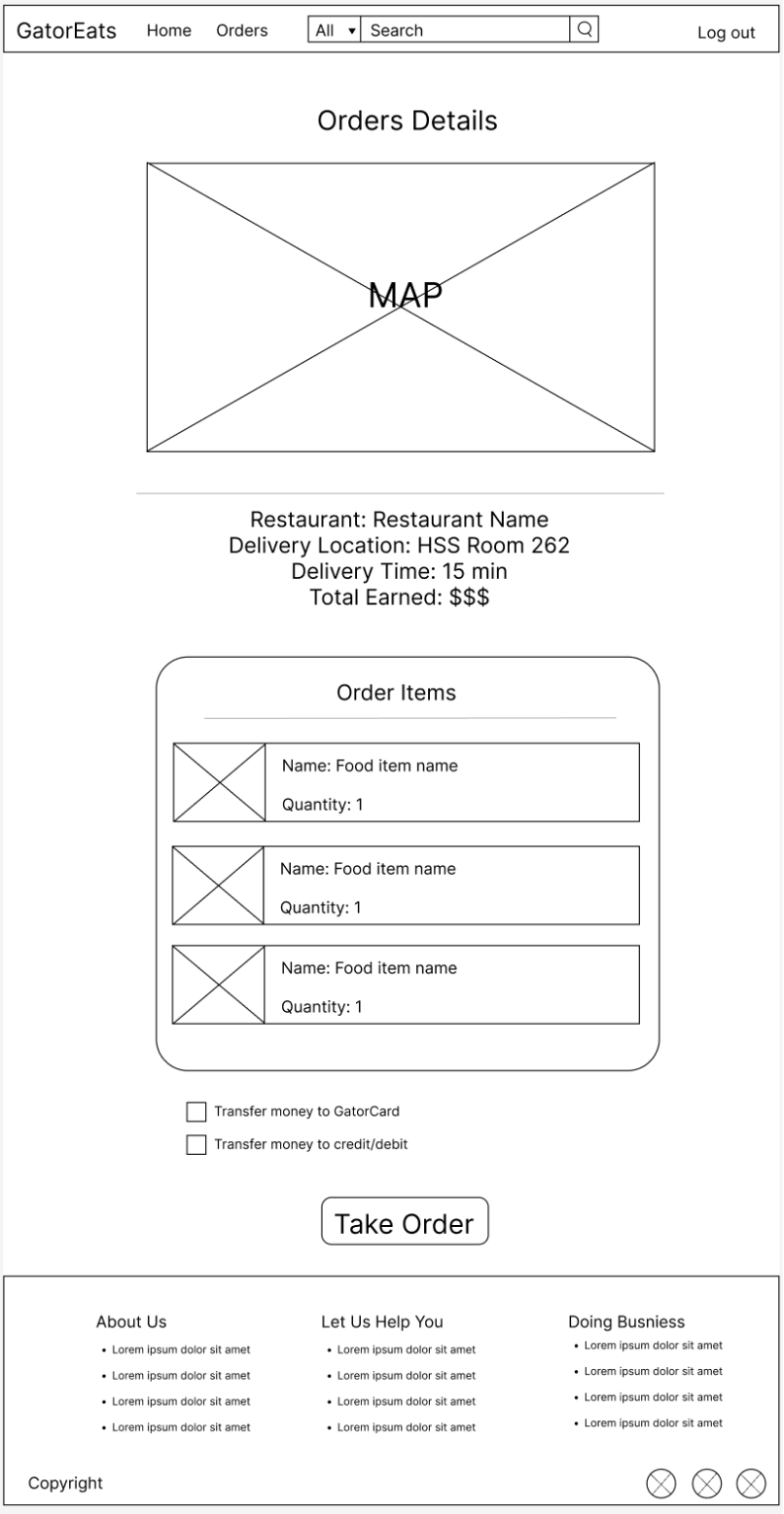
Home Page



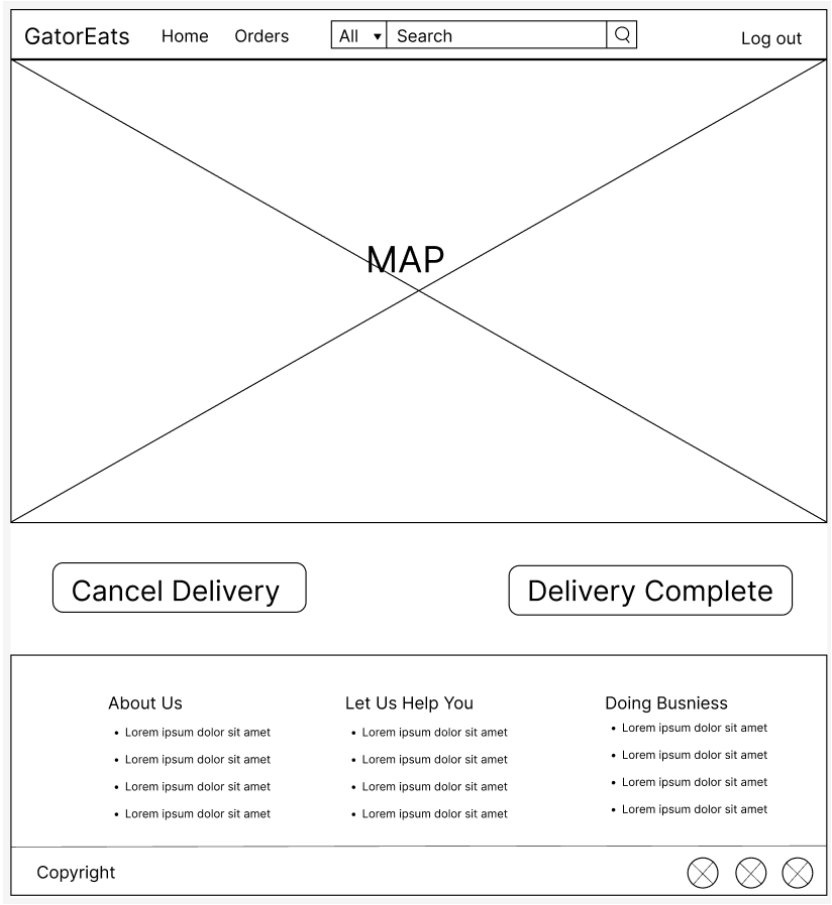
# Orders Page


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
About Us
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Let Us Help You
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Doing Busniess
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Copyright

Review and Confirm Order Details Page



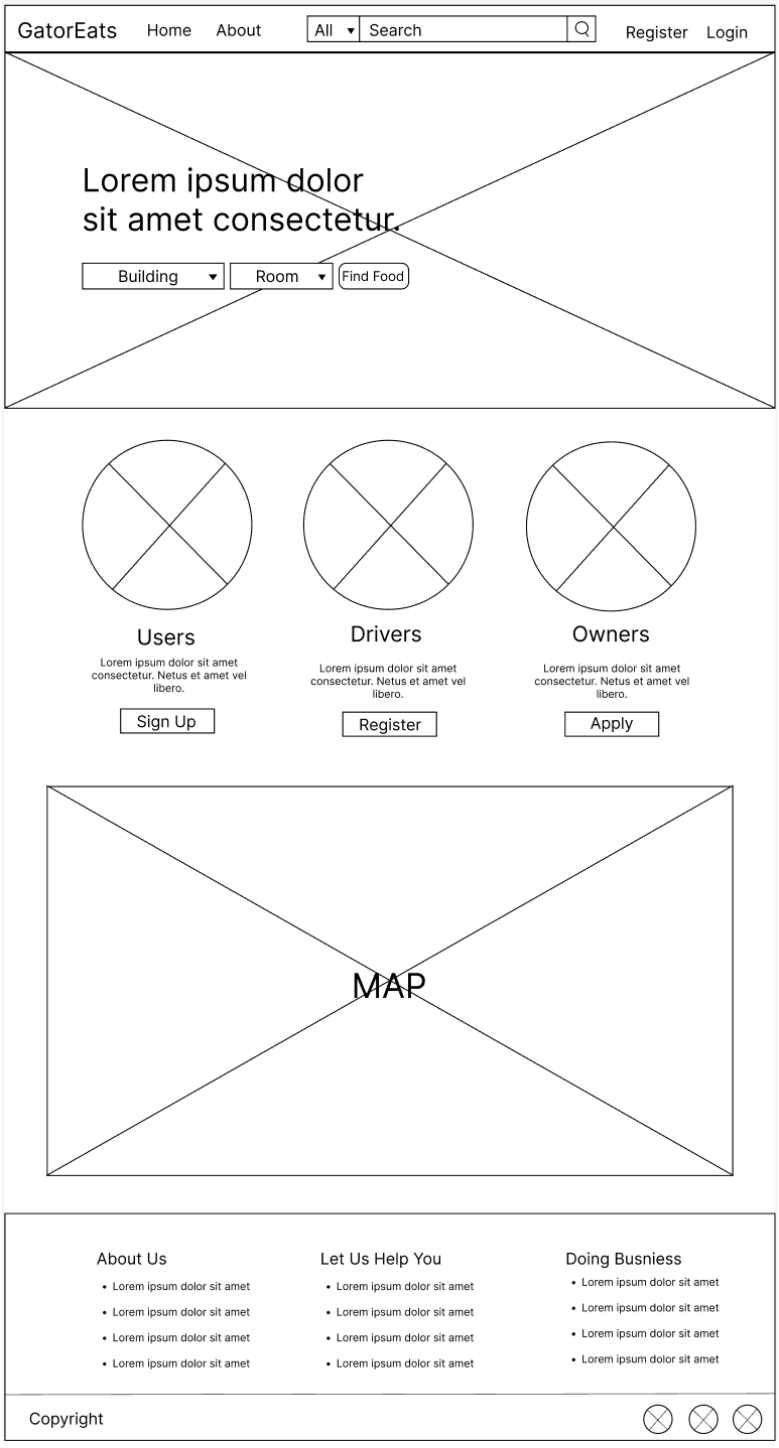
Map and Delivery Confirmation Page



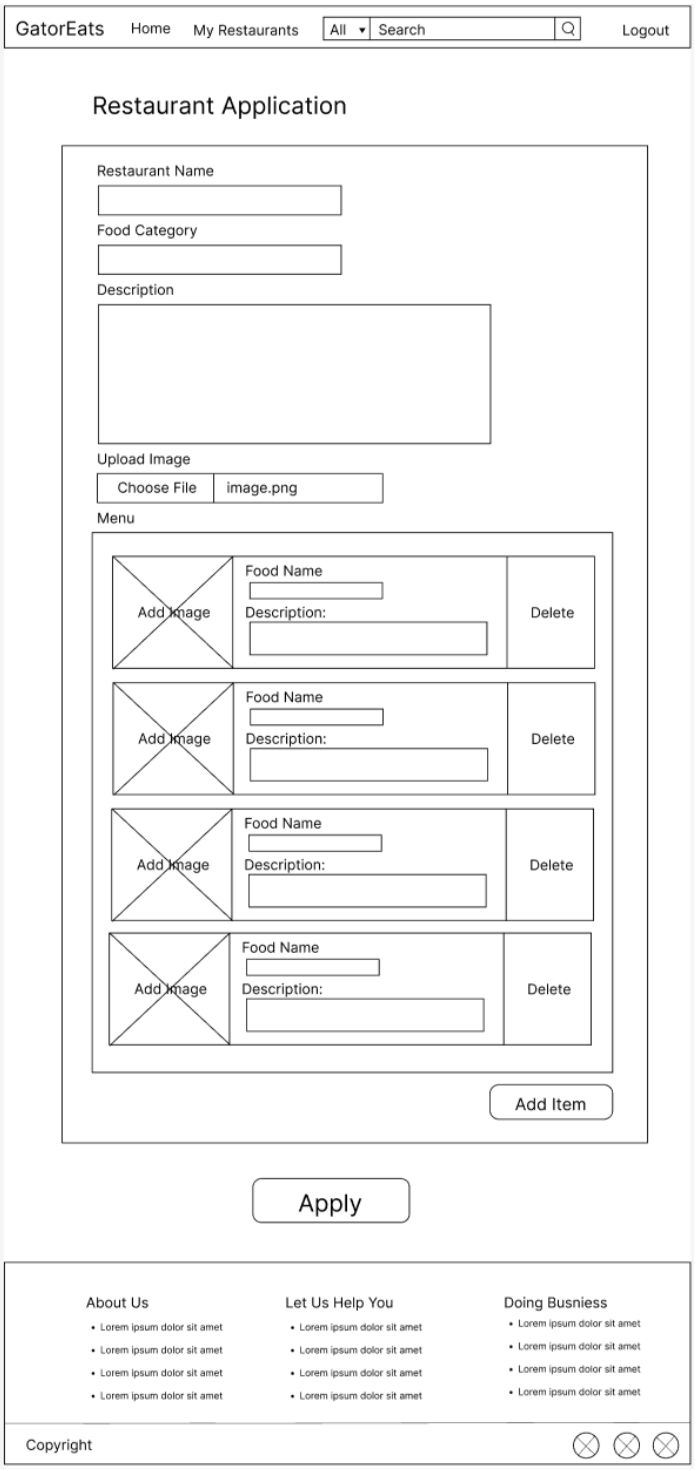
# Restaurant Owner

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 an 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.

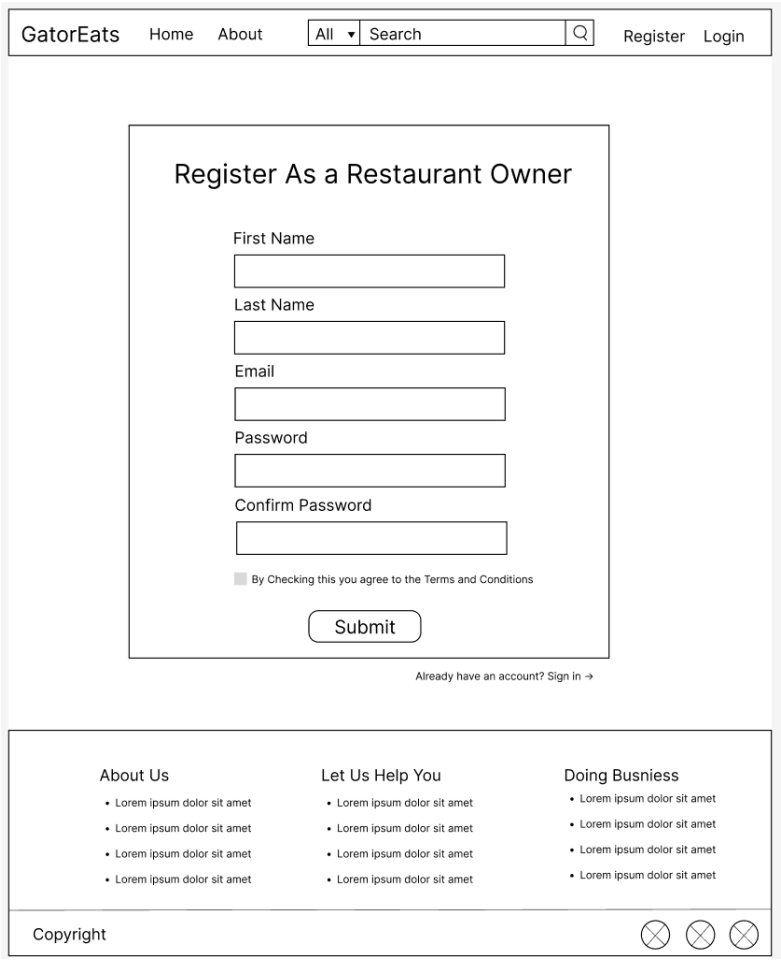
Home Page



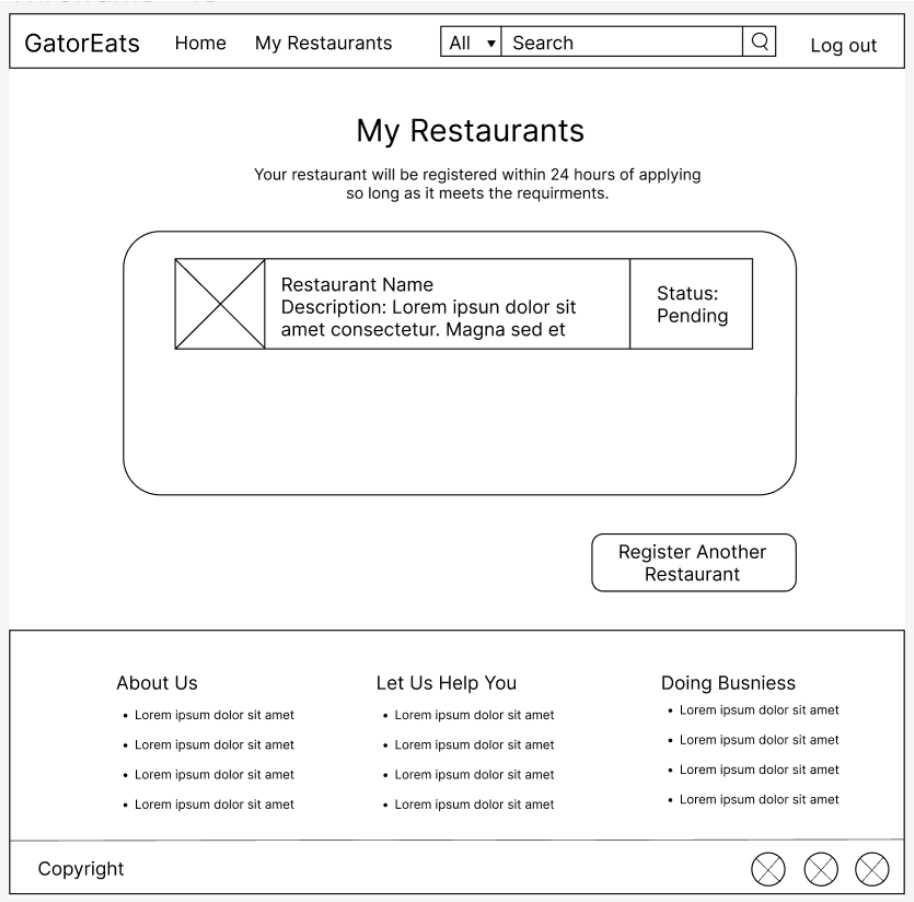
# Restaurant Application Page



# Restaurant Owner Sign in Page



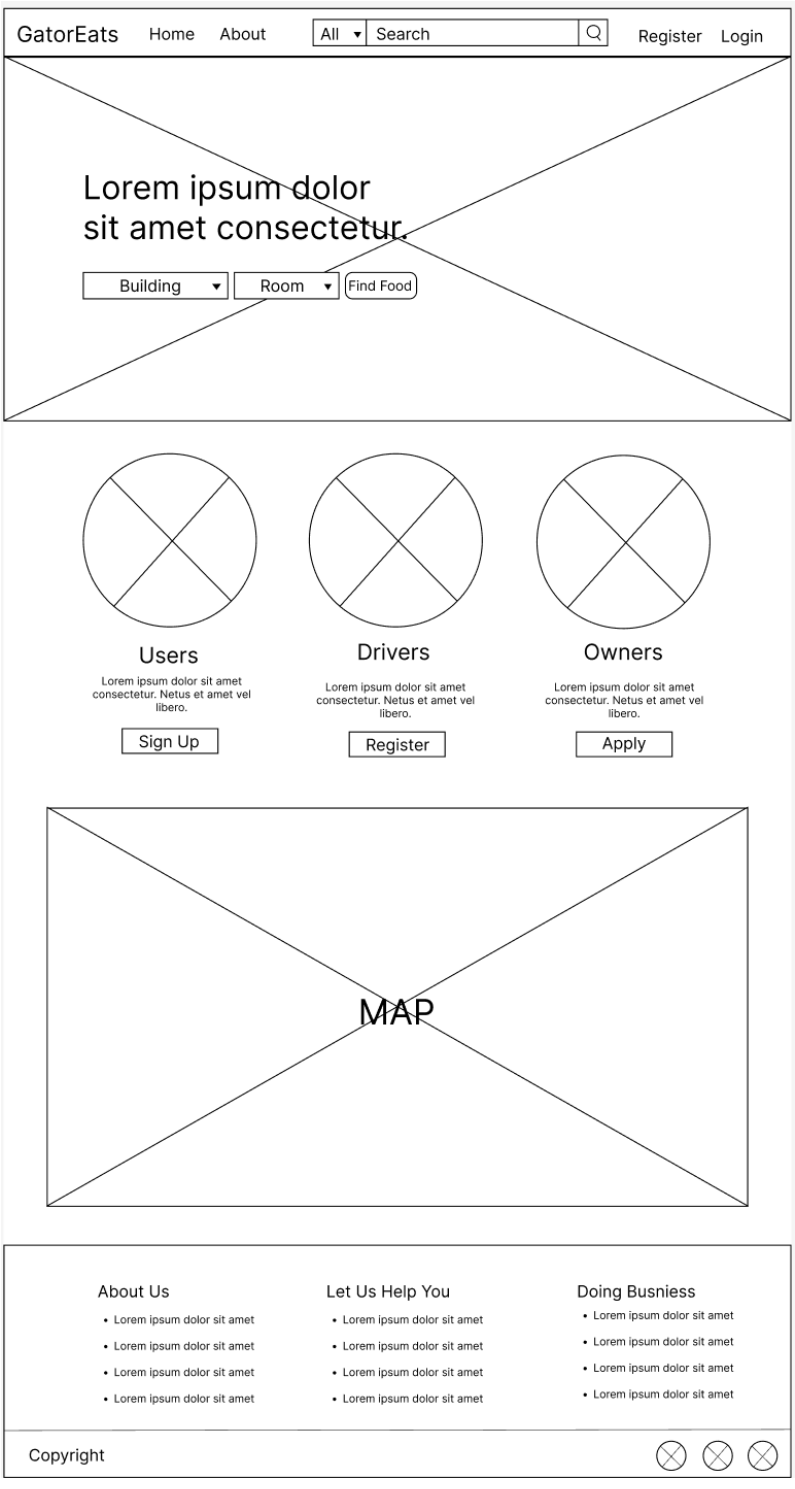
Pending and Registered Restaurants Page



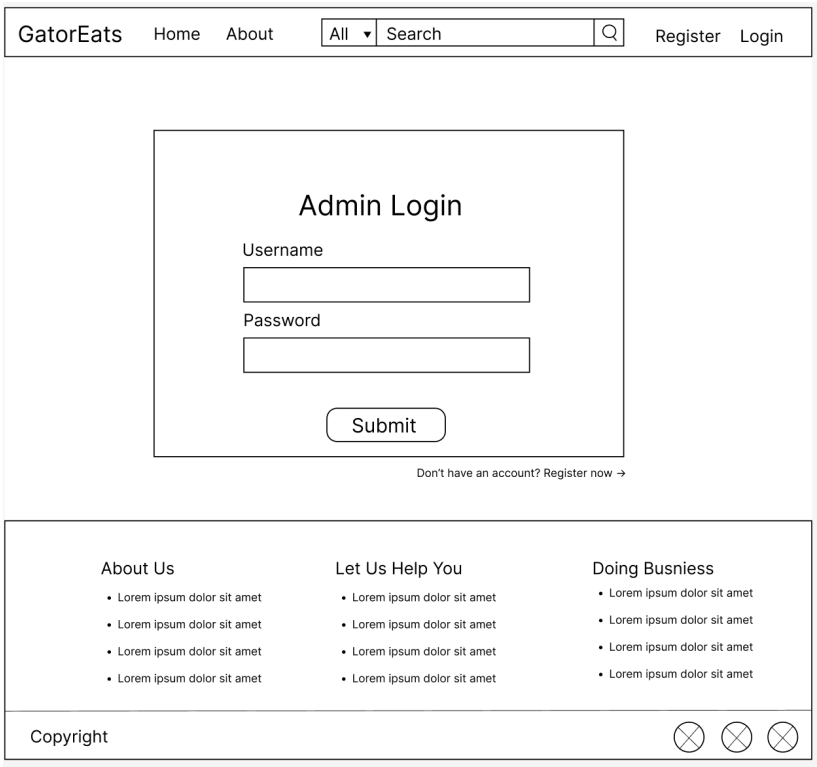
# Admin

Bruce is an admin working for GatorEats. His job is to approve restaurants that apply to GatorEats. He is given an admin username and password to login. He heads to the website and goes to login. He logs in with the provided admin username and password. He sees a few restaurants already needing to be approved. He reviews their details and approves/denies them.

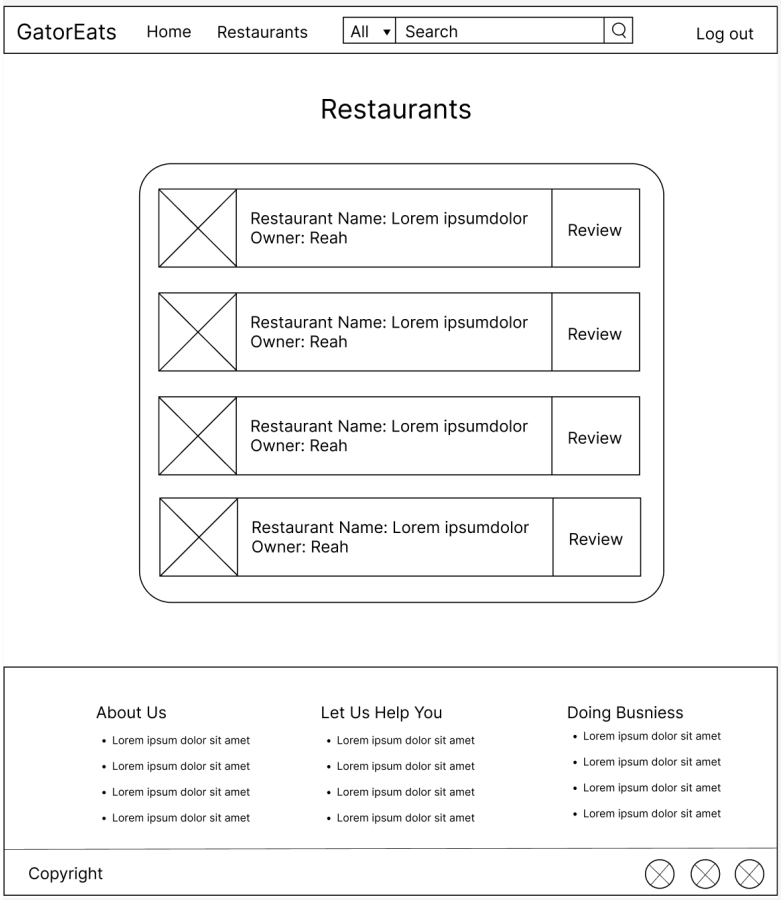
Home Page



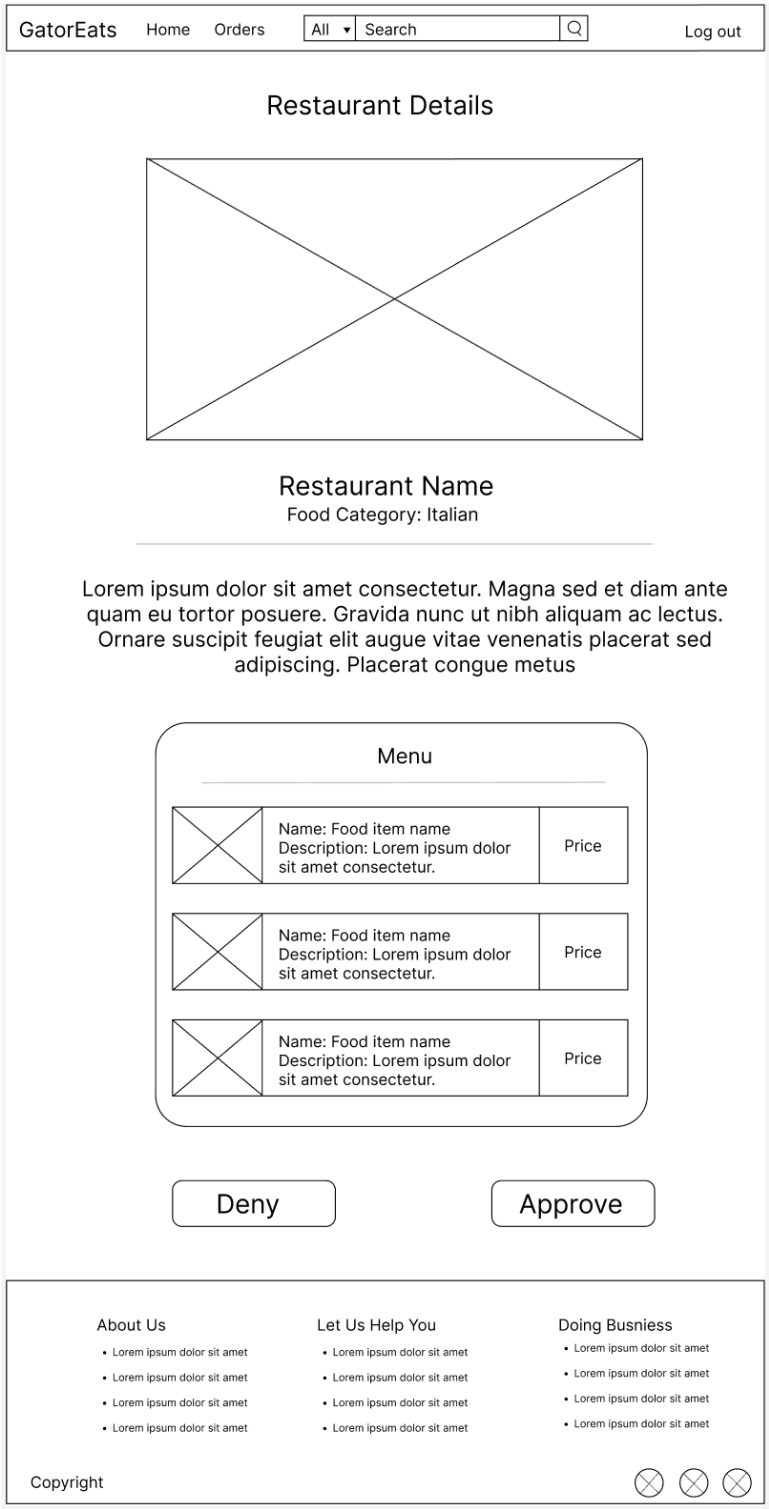
# Login Page


GatorEats
Home
About
All
Search
Register
Login
Admin Login
Username
Password
Submit
Don't have an account? Register now →
About Us
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Let Us Help You
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Doing Busniess
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Copyright

# Restaurant Approval Page



Restaurant Approve/Deny 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 varchar
    - Distance
    - Description
    - Latitude
    - Longitude
10. DropOffPoints
    - PointID PK
    - Name
    - Latitude
    - Longitude
11. Building
    - BuildingID PK int
    - Name varchar
    - Latitude
    - Longitude
    -

12. 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.

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 are confused about how to upload the menus for each restaurant. One option is that restaurant owners could upload a pdf file and we have a parse function that creates the menu on the website. Another option is to have a person manually create the menu for each restaurant. To solve this confusion, we will ask the professor for his opinion and see what's the best approach.

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 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.