SW Engineering CSC648-848 Spring 2023

GatorGrub

Team 3

Shauhin Pourshayegan: spourshayegan@mail.sfsu.edu	Team Lead, Github Master, Document Editor
Hieu Ma:	Front End Lead
Preetham Ramesh:	Back End Lead
Lin Tun:	Front End Developer
Derrick Liang:	Back End Developer

Milestone 2

Date Submitted:	3/31/2023
Date Revised:	

1. Executive Summary:

People at universities often find themselves stuck when searching for their next meal. Food options on campus are likely few and far between, and have the potential to taste like the ever-dreaded dining hall food. Students and instructors alike find issues managing their food habits when heavy workloads and packed schedules take precedent. As students of San Francisco State University, we find ourselves trekking back and forth to Stonestown Galleria just so we can get access to more options, wasting our time when it's most valuable to us. Other than the inconvenience of food locations and availability, the food available is not always affordable, which is a major issue for students and important for everyone. The inconvenience of our current options and the lack of affordable options are the primary driving forces of our team project.

As students, we wanted to provide a solution to these issues when we came to discuss what GatorGrub will be. We propose a service which will strengthen the connection between SFSU students, faculty and staff to their local food scene, as well as offer this primary user base with features which make their lives easier and solve the problems related to getting food on campus. GatorGrub considers the needs of our local restaurants by helping them reach their customers and persuade customers on the edge with exclusive deals on their next meal. GatorGrub aims to provide these users with a selection of local restaurants offering delivery.

Customers who don't have the time to leave their classroom or study area can get their food delivered to any room on campus, allowing their workshops and group study sessions to be uninterrupted by rumbling tummies. Customers can also decide whether they want their food delivered to one of several safe pickup spots throughout SFSU's campus. This feature benefits students who want to get food safely, without worrying about who they will encounter on the long walk to Stonestown or further out. GatorGrub also enables its users to get deals on different local restaurants. This will incentivise more sales, making restaurant owners and delivery drivers content and strengthening GatorGrub's infrastructure.

Our team is dedicated to this project, because of the convenience it could bring directly to us and our community. We are well structured, with a Team Lead at the top to allocate work among the section leads and overlook the entire project. The Back End Lead is responsible for developing the back end with the help of one Back End Developer. Similarly, the Front End Lead is responsible for developing the front end with the help of one Front End Developer. We're concerned with the issues related to student health and happiness, and we know that food is one of the factors which can strongly affect it. We think GatorGrub is the next step forward for the SFSU community.

2. Data Glossary:

List of main data items and entities:

- SFSUCustomer Entity:
    - SFSUCustomerID (PKey)
    - SFSUCustomerName
    - SFSUCustomerEmail
    - SFSUCustomerPhone
    - SFSUCustomerPassword
- Driver Entity:
    - DriverID (PK)
    - DriverName
    - DriverEmail
    - DriverPhone
    - DriverPassword
- RestaurantOwner Entity:
    - RestaurantOwnerID (PK)
    - RestaurantOwnerName
    - RestaurantOwnerEmail
    - RestaurantOwnerPhone
    - RestaurantOwnerPassword
- Restaurant Entity
    - RestaurantID (PK)
    - RestaurantOwnerID (FK)
    - RestaurantName
    - RestaurantEmail
    - RestaurantPhone
    - RestaurantPassword
    - RestaurantAddress
    - RestaurantCuisineType
    - RestaurantPriceTier (updated by trigger)
    - RestaurantHours (Hours of operation, specifically for pick-up/delivery)
    - DeliveryTime (time for order to get to door)
- RestaurantMenu Entity
    - RestaurantMenuID (PK)
    - RestaurantID (FK)
- MenuItem Entity
    - MenuItemID (PK)
    - MenuID (FK)
    - MenuItemName
    - MenuItemPicture
    - MenuItemPrice
    - MenuItemDescription
- Order Entity

    - OrderID (PK)
    - CustomerID (FK)
    - DriverID (FK)
    - RestaurantID (FK)
    - OrderTime
    - DeliveryTime (OrderTime + RestaurantPrepTime + Google Maps Journey Time)
    - DeliveryLocation
        - Shall be either a desired address, building and room, or a “safe pickup spot”
    - OrderSelections (CSV, menu item: ID, quantity, notes)
    - OrderNotes
    - OrderDiscounts
    - OrderPrice
- Review Entity
    - ReviewID (PK)
    - RestaurantID (FK)
    - UserID (FK)
    - Message
    - Rating

3. Functional Requirements Prioritized:

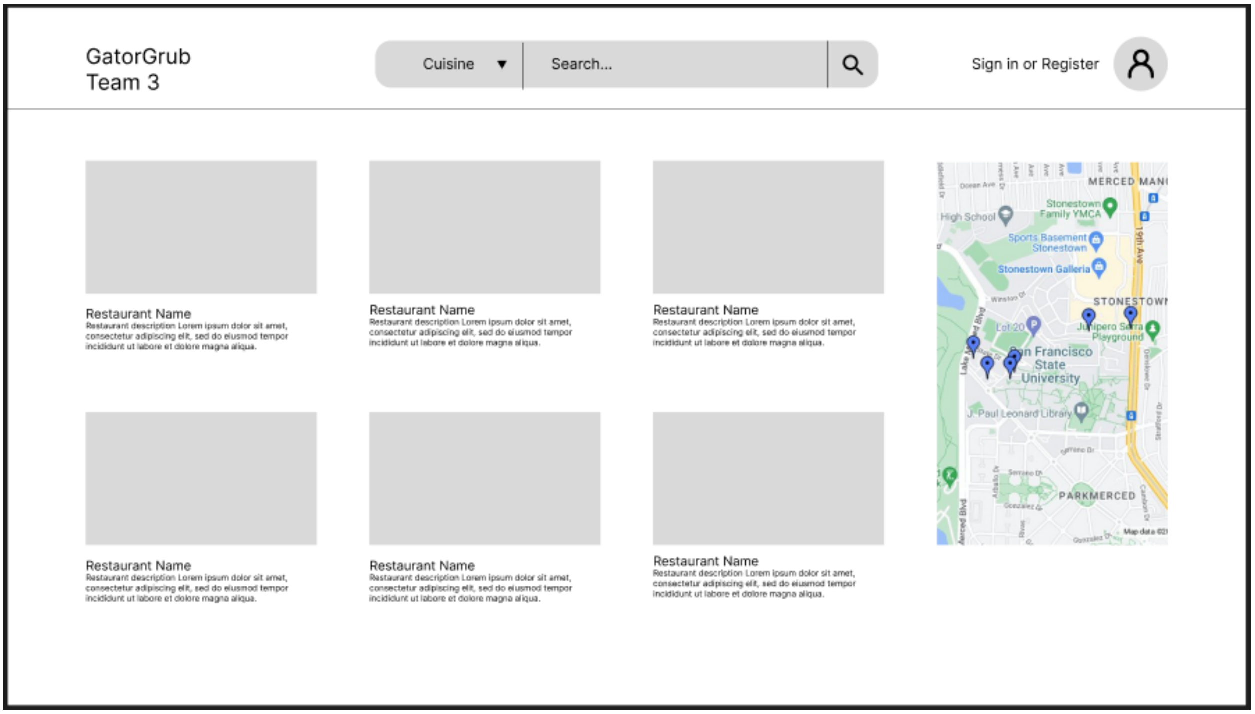
- **Priority 1**
    - Unregistered Users:
        - Unregistered users shall be able to search for food and restaurants.
        - Unregistered users shall be able to search the restaurant map from their location.
        - Unregistered users shall be able to select menu items and add menu items to their cart.
        - Unregistered users shall be able to choose an order’s delivery location.
        - Unregistered users shall be able to read restaurant menus.
    - SFSU Users:
        - SFSU Users shall inherit all unregistered user functional requirements.
        - SFSU Users shall register their account with an SFSU email.
        - SFSU Users shall be able to submit an order.
        - SFSU Users shall be able to see their order’s delivery status.
    - Admins:
        - Admins shall be required to approve all restaurant applications before they go live on the app.
        - Admins shall review restaurant applications and provide the appropriate response.

        - Admins shall be able to view the user account.
        - Admins shall be able to warn users and restrict orders from the user account, if a user violates GatorGrub policy with the account.
    - Restaurant Owners:
        - Restaurant owners shall be able to upload menu item pictures and prices.
        - Restaurant owners shall be able to submit restaurant details. (e.g. hours of operation, avg. order time.)
        - Restaurant owners shall be able to choose to accept or reject an order.
        - Restaurant owners shall confirm when orders are ready. This updates the order progress.
    - Driver Users:
        - Driver users shall be able to accept or reject an order.
        - Driver users shall confirm when food is picked up. This updates the order progress.
        - Driver users shall confirm when food is delivered. This updates the order progress.
        - Driver users shall be able to see a map of the pickup point.
        - Driver users shall be able to see a map of the delivery point.
- **Priority 2**
    - SFSU Users:
        - SFSU Users shall be able to write feedback on the order.
        - SFSU Users shall be able to upload pictures.
        - SFSU Users shall be able to send and receive messages to customer service.
    - Restaurant Owners:
        - Restaurant owners shall be able to respond to the feedback of their restaurant.
        - Restaurant owners shall be able to send and receive messages to customer service.
    - Driver Users:
        - Driver users shall be able to contact customers.
- **Priority 3**
    - Unregistered Users:
        - Unregistered users shall be able to read restaurant reviews.
    - SFSU Users:
        - SFSU Users shall be able to contact drivers.
    - Restaurant Owners:
        - Restaurant owners shall be able to contact their customers and drivers.
    - Driver Users:
        - Driver users shall be able to send and receive messages to customer service.

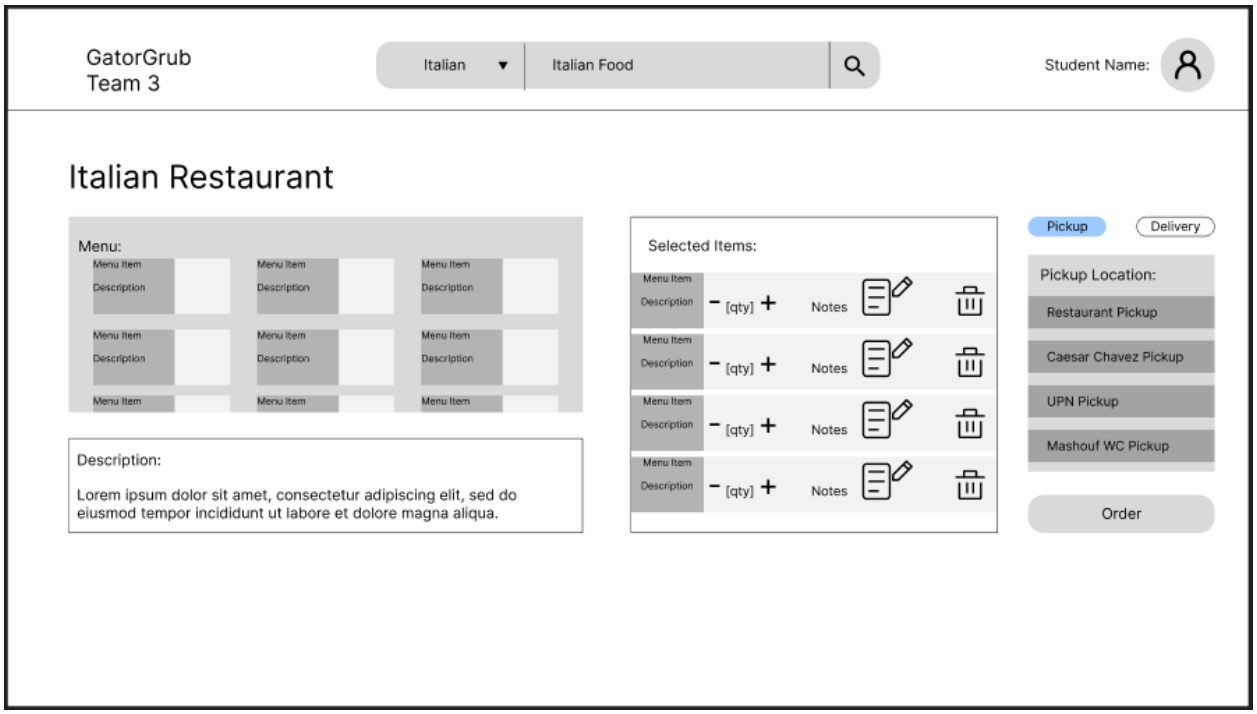
4. UI Storyboards Per Use Case:

Case 1: Kate (**SFSU student**) wanted to grab a quick bite to eat without interrupting her study time. She turned to GatorGrub, a food ordering, delivery, and pickup service, to help her find a convenient solution.

1. Kate opens the site and sees home page:

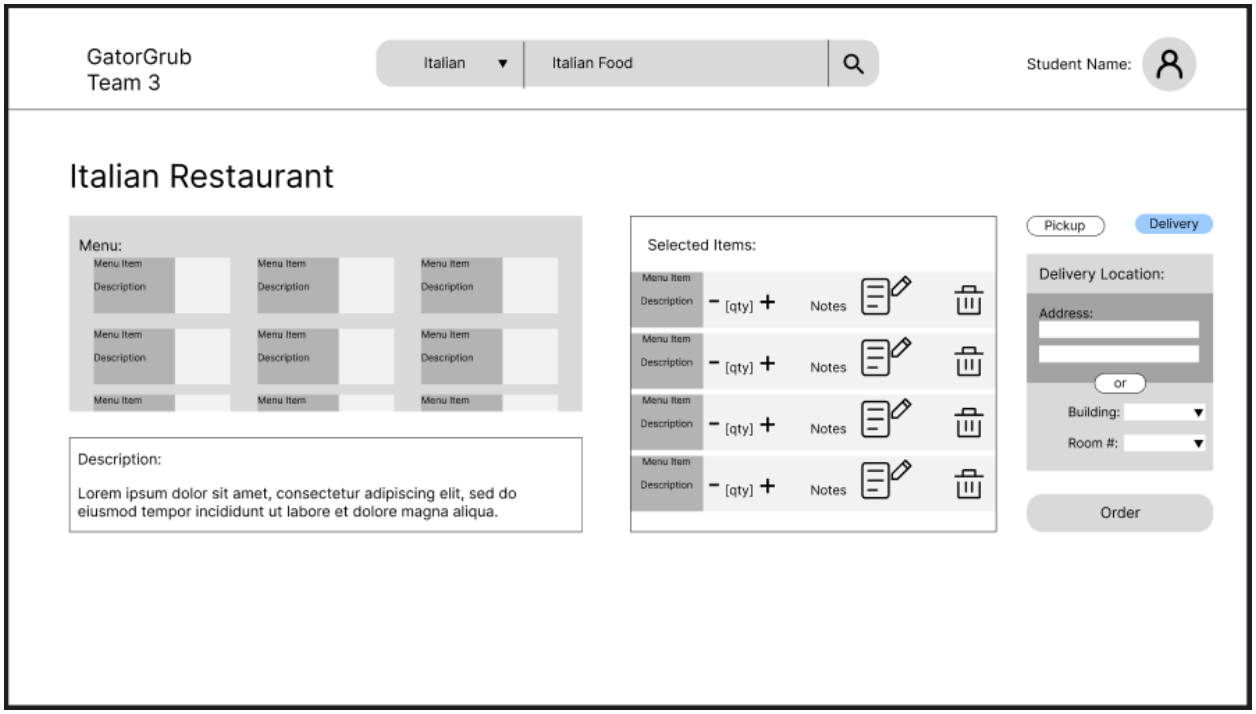


2. Kate then clicks on a restaurant, fills in her order, selects a pickup location, then clicks order to finish.



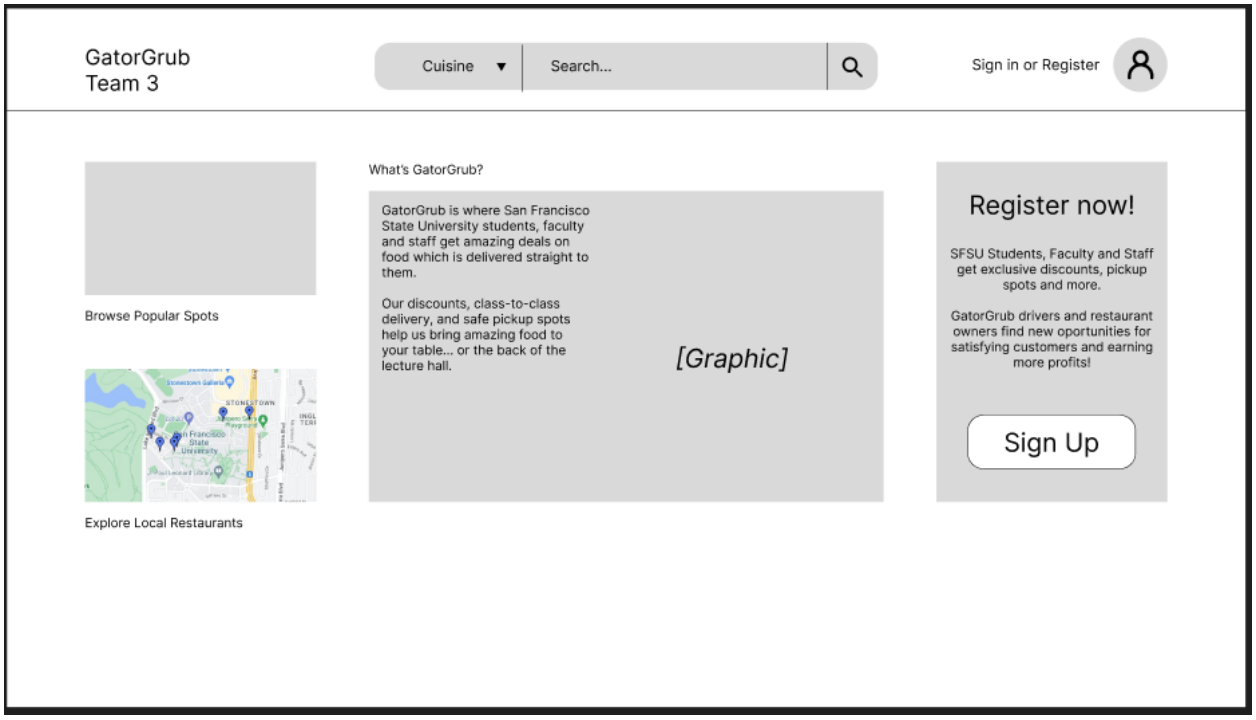
Case 2: Long days answering student questions can make managing Jane's (**SFSU faculty/staff**) lunch difficult. It makes it easy when her food can be delivered straight to her. When she wants something for lunch that's easy to manage, Jane opens up GatorGrub.

1. After Jane selects a restaurant, she fills in her order, selects a location for delivery, and then clicks order to finish.

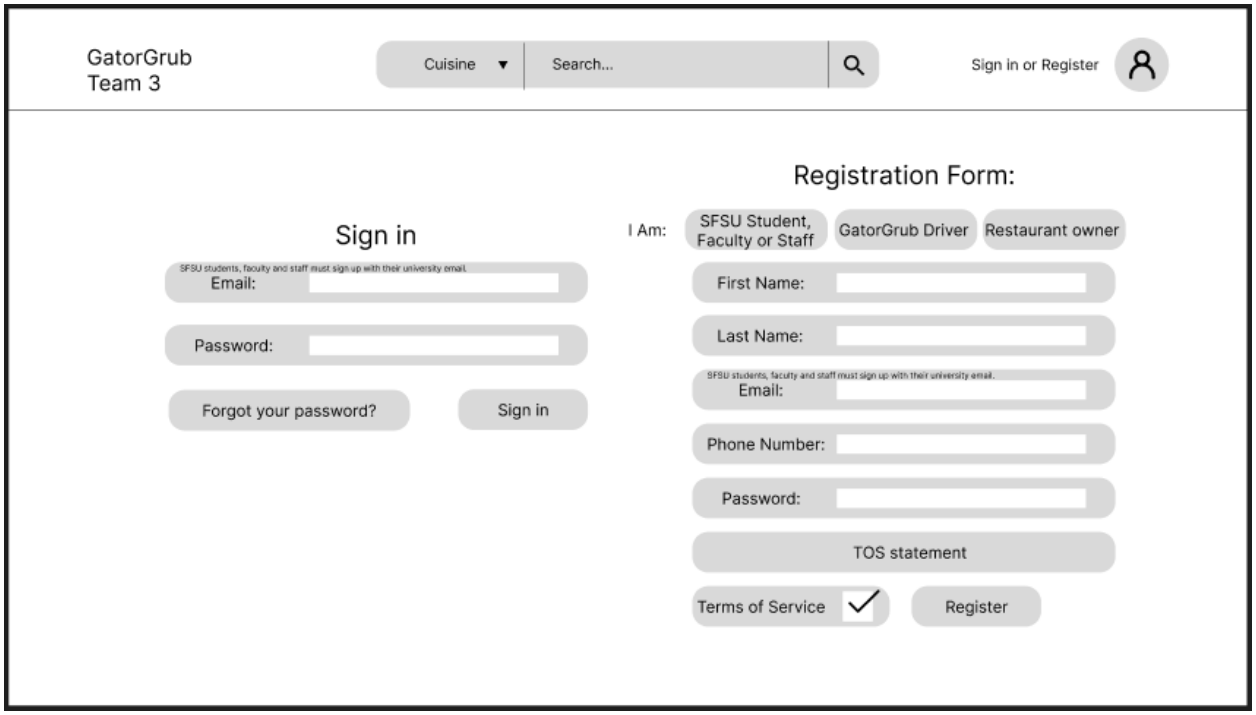


Case: 3 Jared (**GatorGrub driver**) is looking for additional ways to make money while still being able to manage his classes. He realizes that he can leverage his access to different modes of transportation, depending on his class schedule, to earn extra cash. To do so, he decides to register as a driver for GatorGrub, a food delivery service on campus. One of the benefits of being a GatorGrub driver is that Jared has complete control over the orders he accepts.

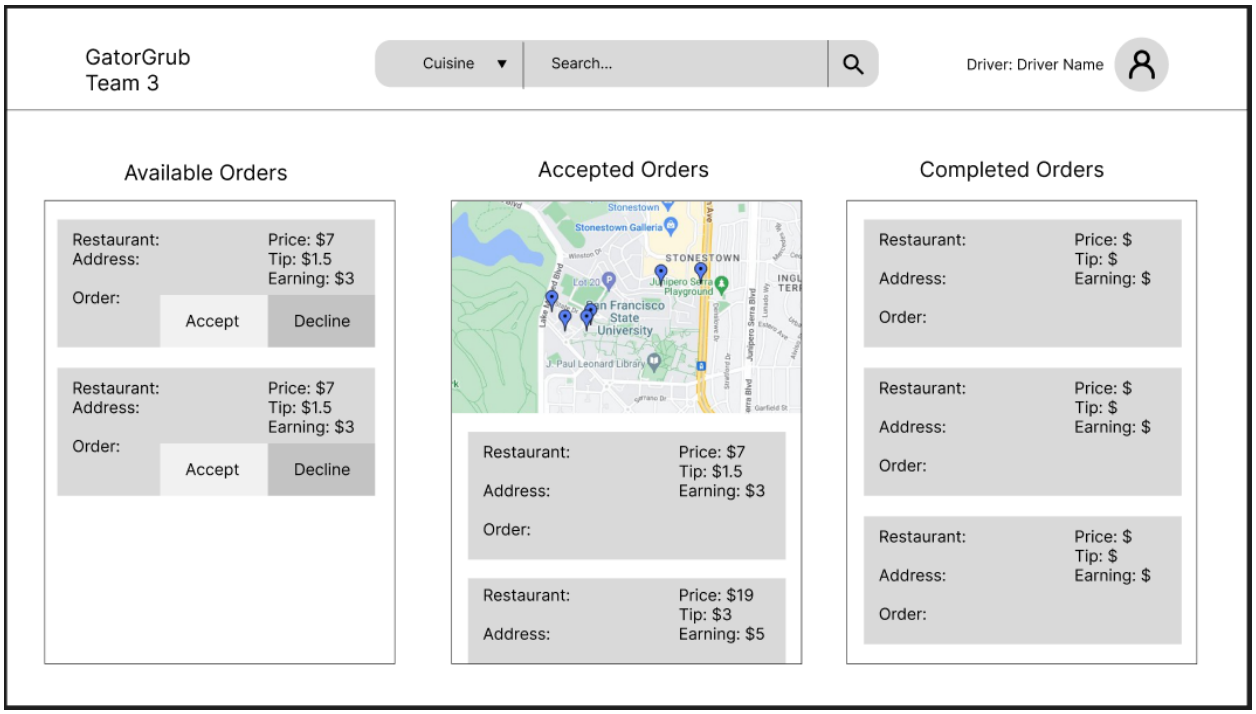
1. Jared visits the homepage to start registering as a driver. Jared clicks the sign up option.



2. The registration form appears and now after going through the process of registering, Jared now can take delivery requests.



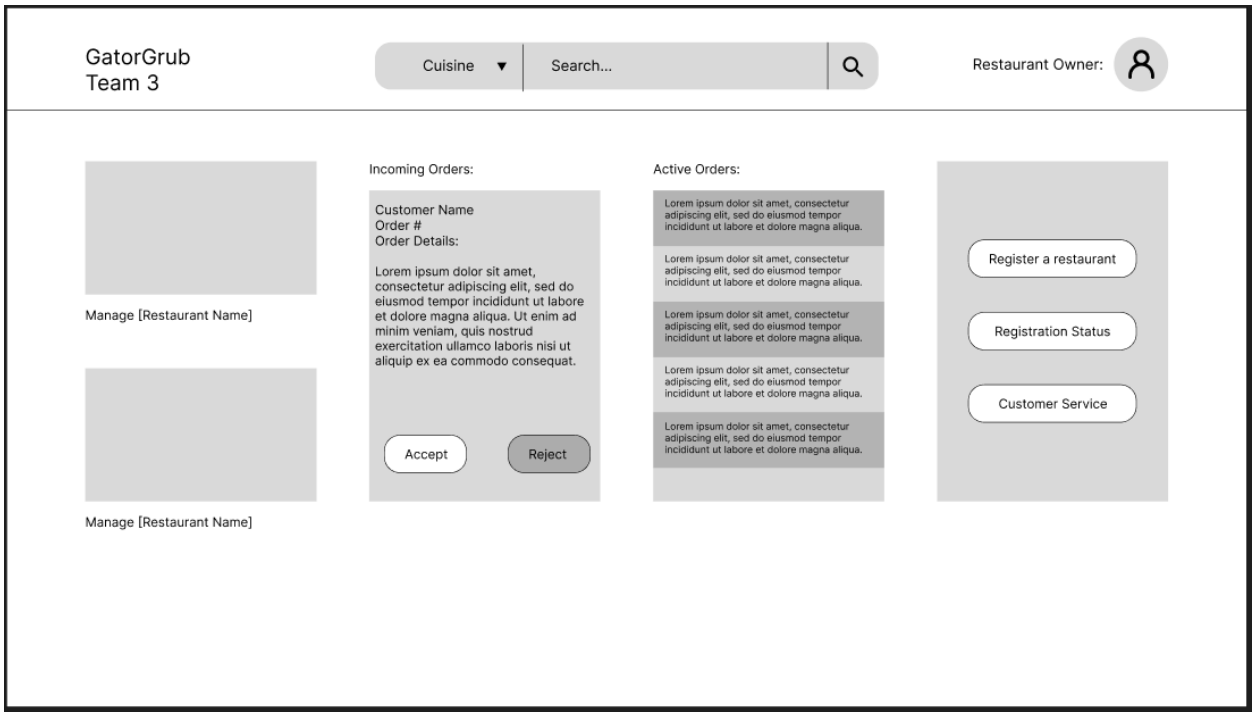
3. After clicking on the driver profile on the top right, it leads to this page for the driver where they can accept/decline orders:



Case 4: Marcus (**Restaurant owner**) is looking for ways to increase the exposure of his restaurant and attract more customers. He realizes that by partnering with GatorGrub, he can cater to a specific group of customers: the students and staff at San Francisco State University. He will need to provide details about his restaurant, such as the menu pdf, selecting operating hours, and selecting location. To register his restaurant with GatorGrub, Marcus needs to create a restaurant owner account.

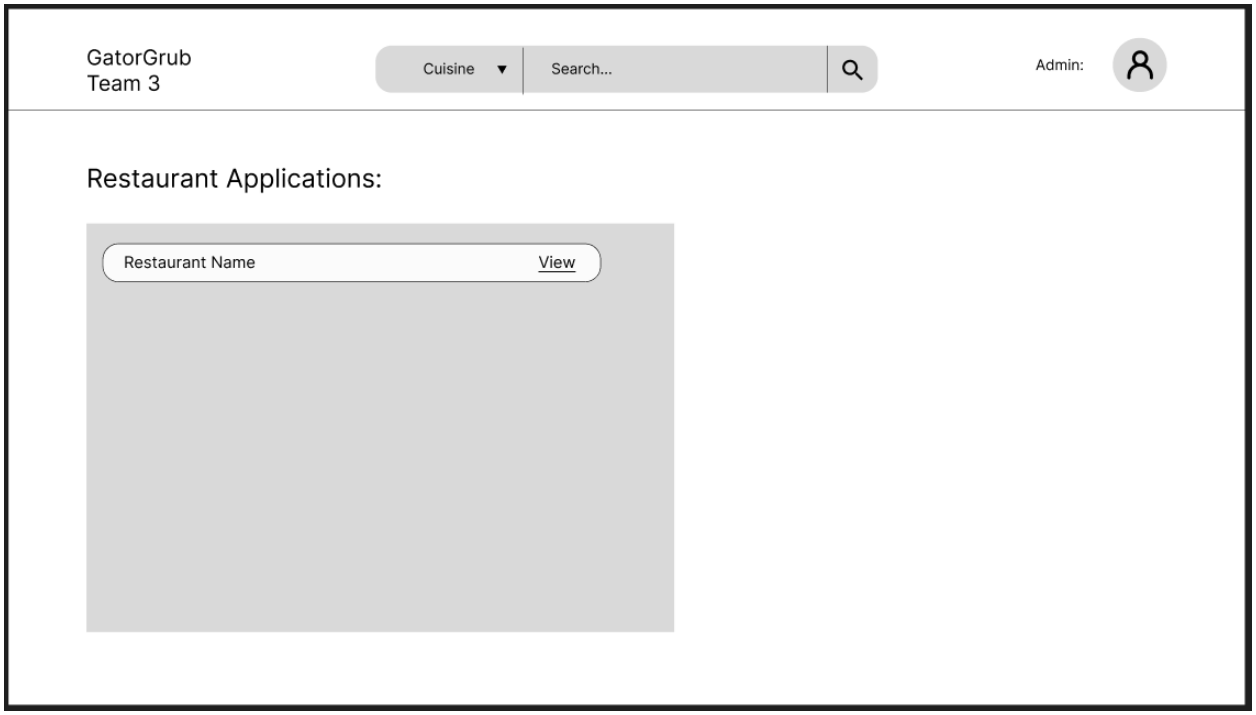
1. After the restaurant owner signs up their account through the home page, they then can click their icon and then this restaurant registration form will appear.

2. After their restaurant is approved by the website admin, they can see incoming orders and online orders through clicking their profile icon.

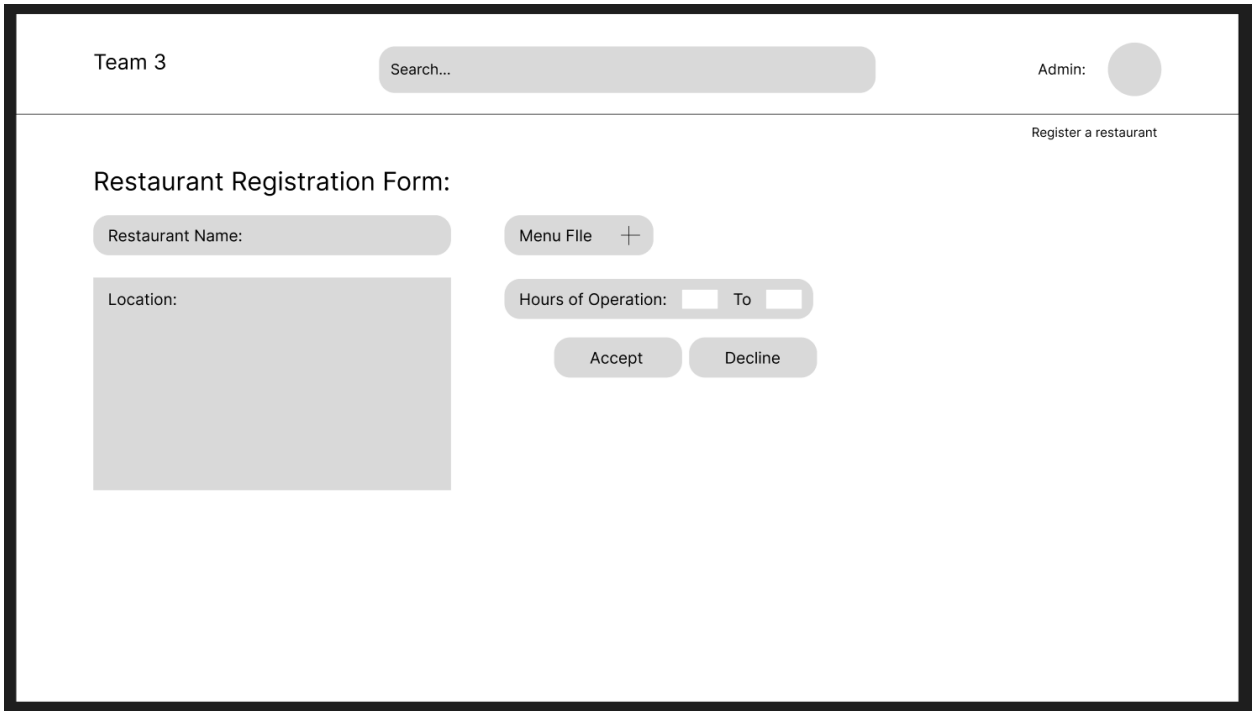


Case 5: Emma works as a GatorGrub admin, so she knows the daily ins and outs of running the restaurant verification process. When she wakes up she can check the newly submitted applications on her laptop. One by one, each submission is verified to see if the restaurant is real and if the food on the submitted menu is similar enough to the food served by the restaurant in question.

1. Emma can click on her user profile and see restaurants that have applied to be on the website



2. When the admin clicks view, they can approve or decline the restaurant application.



5. High-level architecture and DB Organization Summary:

DB Organization:

List of main data items and entities:

- SFSUCustomer Entity:
    - SFSUCustomerID (PKey)
    - SFSUCustomerName
    - SFSUCustomerEmail
    - SFSUCustomerPhone
    - SFSUCustomerPassword
- Driver Entity:
    - DriverID (PK)
    - DriverName
    - DriverEmail
    - DriverPhone
    - DriverPassword
- RestaurantOwner Entity:
    - RestaurantOwnerID (PK)
    - RestaurantOwnerName

    - RestaurantOwnerEmail
    - RestaurantOwnerPhone
    - RestaurantOwnerPassword
- Restaurant Entity
    - RestaurantID (PK)
    - RestaurantOwnerID (FK)
    - RestaurantName
    - RestaurantEmail
    - RestaurantPhone
    - RestaurantPassword
    - RestaurantAddress
    - RestaurantCuisineType
    - RestaurantPriceTier (updated by trigger)
    - RestaurantHours (Hours of operation, specifically for pick-up/delivery)
    - DeliveryTime (time for order to get to door)
- RestaurantMenu Entity
    - RestaurantMenuID (PK)
    - RestaurantID (FK)
- MenuItem Entity
    - MenuItemID (PK)
    - MenuID (FK)
    - MenuItemName
    - MenuItemPicture
    - MenuItemPrice
    - MenuItemDescription
- Order Entity
    - OrderID (PK)
    - CustomerID (FK)
    - DriverID (FK)
    - RestaurantID (FK)
    - OrderTime
    - DeliveryTime (OrderTime + RestaurantPrepTime + Google Maps Journey Time)
    - DeliveryLocation
        - Shall be either a desired address, building and room, or a “safe pickup spot”
    - OrderSelections (CSV, menu item: ID, quantity, notes)
    - OrderNotes
    - OrderDiscounts
    - OrderPrice
- Review Entity
    - ReviewID (PK)
    - RestaurantID (FK)
    - UserID (FK)
    - Message

- Rating

Media Storage:

Our media will be stored in the file system of the server. It is not advisable to store media in a database as it can put an unneeded strain on it. Our database will instead either store a path to a resource, or a link to that resource stored on a CDN.

Search functionality implementation:

Our search functionality will be fairly simple. Our database will store each restaurant along with metadata such as location, cuisine, price information, restaurant name, and price tier. This way we can organize restaurants by categories for users to browse. When a user wants to search using an input field, we will take the input, sanitize it and plug it into SQL queries for retrieval. We could use the LIKE function to perform a fuzzy search. We can implement indexing and implement pagination to optimize search functions.

SW tools changes:

We have switched our primary operating system from AlmaLinux to Ubuntu 20.04 LTS. AlmaLinux was a miscommunication and Ubuntu is a more standard OS.

6. Identifying Key Risks To Our Project:

Schedule Risks:

In-person meetings are difficult for our team to complete due to conflicting schedules. We have had times when some team members have not been present at a meeting.

We have addressed this in the past and plan to keep an eye on it by encouraging in-person meetings when possible, but also opening up to online meetings, which has the added benefit of screen sharing and clear discussion. We also host more casual meetings to check in with each other. Team Lead makes sure to post helpful meeting notes so that members have something to look back on and review.

Technical Risks:

Keeping track of GCE has been difficult and has left us with a little less time available than we initially thought we would have at this stage in the project. If the issue is not dealt with it can cause our team to spend money (against the requirements).

We will continue to keep an eye on the GCE and make sure we have as much time as possible, but we have to come to an agreement on how to move forward. We intend not to spend money on this project, so we have to consider migrating our server to another account and stopping our service with the previous account so the individual group member doesn’t get charged.

7. Project Management:

Milestone 2 was pretty straightforward to manage for our team. After the class where we discussed milestone 2, our team lead was able to set up a Trello workspace for our team. After the milestone sections had been added to it, everyone was invited to see what we had on our plate. A meeting took place when we had set up the Trello workspace, in order to make sure everyone was on the same page. Each section of the milestone 2 document was briefly outlined and then assigned to the different sections of the team (back end & front end). The team lead also had some sections to fill out independently. Our M2 part 1 document was held on google drive and everyone has access to edit whatever they'd like. For anyone who may have had to miss the team meeting, the meeting notes were posted on a dedicated Discord text channel on the team's Discord server. (Discord is a team based messaging platform built for small professional teams and gaming enthusiasts, very similar to Slack.) The meeting notes specified instructions for the team, including goals, requirements and things to get done. These are typically specified for the two subteams, front end and back end. If we get close to a deadline, individual members are asked to complete specific parts of the assignment so that we can stay on track. Our meeting after the M2 reveal consisted of reviewing CEO notes and editing the M1 document. We then discussed any thoughts and concerns we had for each section of the M2 document. Team lead is responsible for the whole document being presentable and concise, whereas front end leads and back end leads should discuss directly with their partner how much work, assigned by the team lead, each is responsible for completing. When the rough draft is done, all group members reconvene to discuss the state of the submission document, as well as any other concerns they have with upcoming requirements for M2 part 2. After everyone is able to make their edits and share their thoughts, everyone makes the appropriate changes and the team lead is responsible for finalizing the document for submission. After the finalized edited document is prepared, the team comes to review it one last time before submission to make sure they're happy with their work. These reviews usually take place over a Discord voice call so we can point out specific parts of the submission while sharing our screens for sharing notes.