Milestone 1

SW Engineering CSC 648 - 848 Spring 2023

Food Delivery Application

GatorBites

Team 02

Team Lead: Dylan Burns (dburns6@sfsu.edu)
Github Master/Back End: Christian Jackson
Front End Lead: Tim Polich
Back End Lead: Kevin Aziz
Front End: Wallace Man
Back End: Jeremiah Ruvalcaba

Date Submitted	Date Revised
	03/09/2023
	03/11/2023
	03/12/2023

Table of Contents

1 Executive Summary

Introducing GatorBites - the ultimate food delivery app designed exclusively for the hungry and busy SFSU community. Our goal is simple - to provide a hassle-free experience for students and staff by delivering delicious food right to their classrooms.

We understand that time is of the essence and that's why we've leveraged our specialized knowledge of the SFSU campus, leading to a whopping 25% reduction in delivery times. Say goodbye to cold food and frustration, as our in-app navigation tool provides our drivers with real-time routing directly to your classroom of choice.

We take pride in our team of active students who work part-time to pay for their college education. These students are familiar with the campus, making food delivery an absolute breeze. However, we understand that not all our drivers may be familiar with the campus, which is why we have a stringent sign-on screening process to ensure they know their way around before joining our team.

At GatorBites, we make food delivery easy. With just four simple steps - create an account, add a payment method, enter your delivery location, and order your food - you can have your favorite meals delivered to you in no time.

Whether you're having a busy day and forgot to bring lunch to campus, need a coffee to power through your late-night classes, or just craving a cheesy pizza for a group study session, GatorBites has got you covered. We know that students and faculty are always on the move, constantly working and planning, and often forget to eat at regular intervals. With GatorBites, you can focus on what really matters, while we take care of your food delivery needs.

Join us today and experience the ultimate food delivery service designed exclusively for SFSU students and staff - GatorBites!

2 Personae and Use Cases

Max

Busy Student

- Age: 20
- Major: Business Administration
- Family Status: Single

Bio

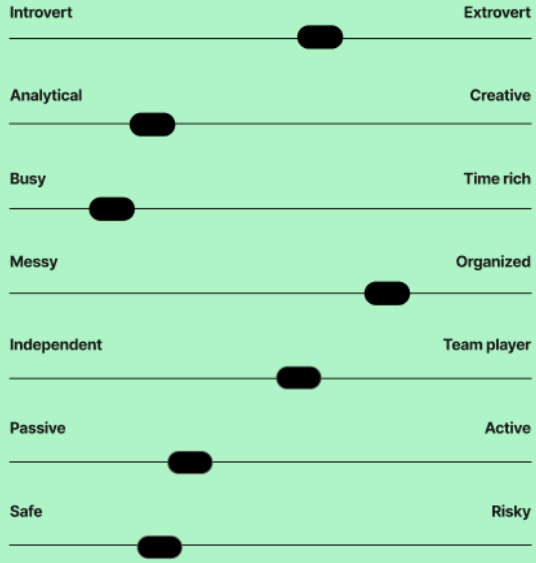
General Characteristics: A busy student who has a packed schedule with deadlines, classes, and extracurricular activities. He values convenience and affordability.

Goals:
1. To save money as a college student
2. To be able to order meals quickly
3. To have meals delivered to his dorm

Skills:
1. Time management
2. Well versed in technology
3. Budgeting and financial planning

Pain Points:
1. Limited funds for food
2. Difficulty finding time in his schedule to prepare meals
3. Requires a lot of energy for his daily routine

Personality



Michelle

Staff Member

- Age: 32
- Occupation: Logistic Coordinator
- Family Status: In Relationship

Bio

General Characteristics: A busy staff member who has to oversee and manage the storage and transportation of San Francisco State University's inventory. She values convenience.

Goals:
1. Wants to be able to receive healthy and nutritious meals quickly
2. To be able to order meals quickly
3. To have the meals delivered to her office

Skills:
1. Time management
2. Well versed in technology
3. Multi-tasking

Pain Points:
1. Limited time to drive and pick up food on breaks
2. Difficulty finding time in his schedule to prepare meals

Personality

Charlie

Faculty Member

- Age: 58
- Occupation: Faculty Member
- Family Status: Married

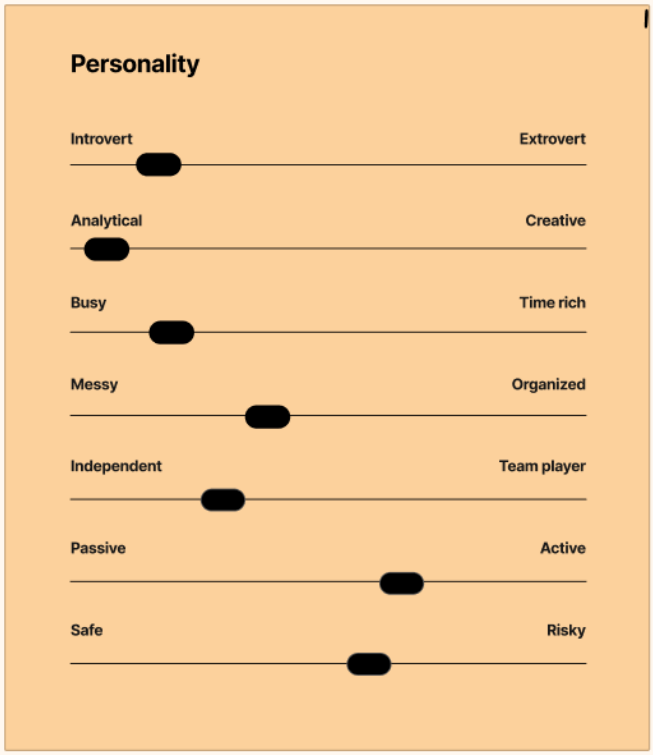
Bio

General Characteristics: A busy professor that has to give lectures to many different classes. Often times, he doesn't have time in between lectures to grab food. Values efficiency and speed.

Goals:
1. To be able to get meals quick and easy
2. To not have his work flow disturbed
3. To be able to give lectures without having to leaving to grab food

Skills:
1. Unfamiliar with using food delivery services
2. Great at multi-tasking

Pain Points:
1. Is unable to eat heavy meals
2. Has bad eyesight
3. Walking to get food takes too long



Aaron

Delivery Driver

- Age: 22
- Major: Computer Science
- Family Status: In Relationship

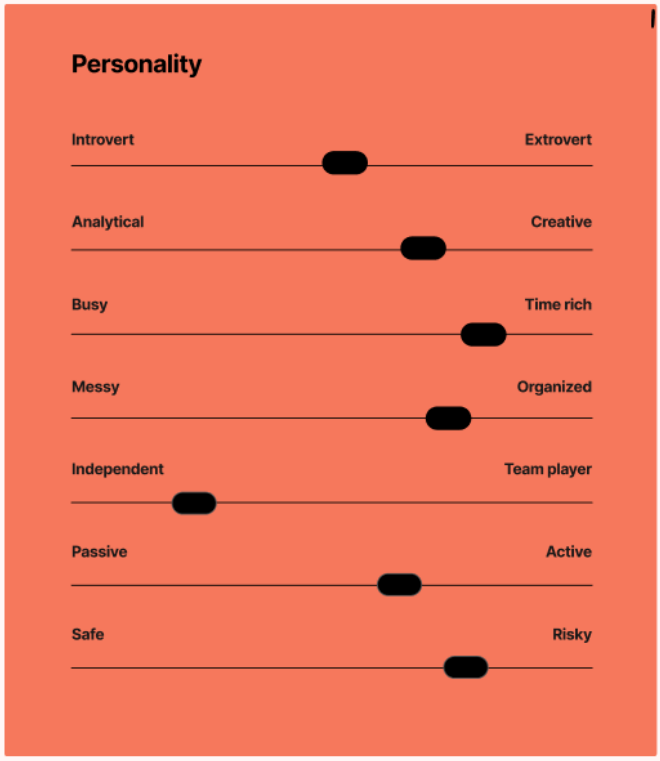
Bio

General Characteristics: A student who only has 2 class this semester and wants good way to make money as he takes the rest of his classes that he needs to graduate. He values time and ease of use.

Goals:
1. To earn money by maximizing the number of delivery he can take
2. To be able to deliver meals quickly and easily using the GatorBites

Skills:
1. Well versed in food delivery applications
2. Very efficient and good at multi-tasking
3. Well versed in using their phone

Pain Points:
1. Uses a bike because he can't afford a car
2. Wants to be able to do delivery for multiple orders at the same time



Ceci

Restaurant Manager

- Age: 38
- Occupation: Restaurant Manager
- Family Status: Married

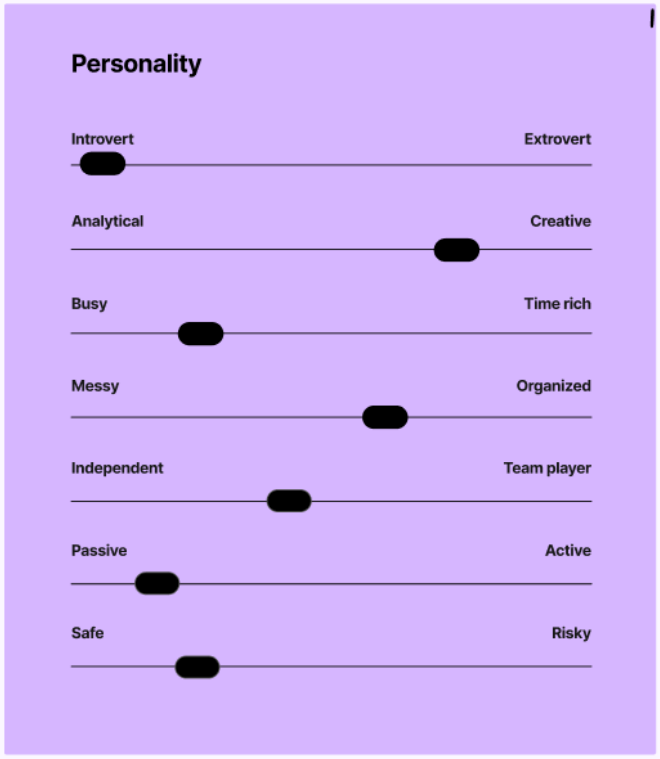
Bio

General Characteristics: A restaurant manager who wants to see her business grow and expand. She has many contacts in the restaurant industry. She values time and convenience.

Goals:
1. To get more customers ordering from her restaurant
2. To be able to set up business on the application easily
3. Ease to set up deliveries

Skills:
1. Time management
2. Not well versed in technology
3. Budgeting and financial planning

Pain Points:
1. Has never used food delivery services before
2. Isn't the best with computers



William

Admin

- Age: 26
- Occupation: Employee of GatorBites
- Family Status: Single

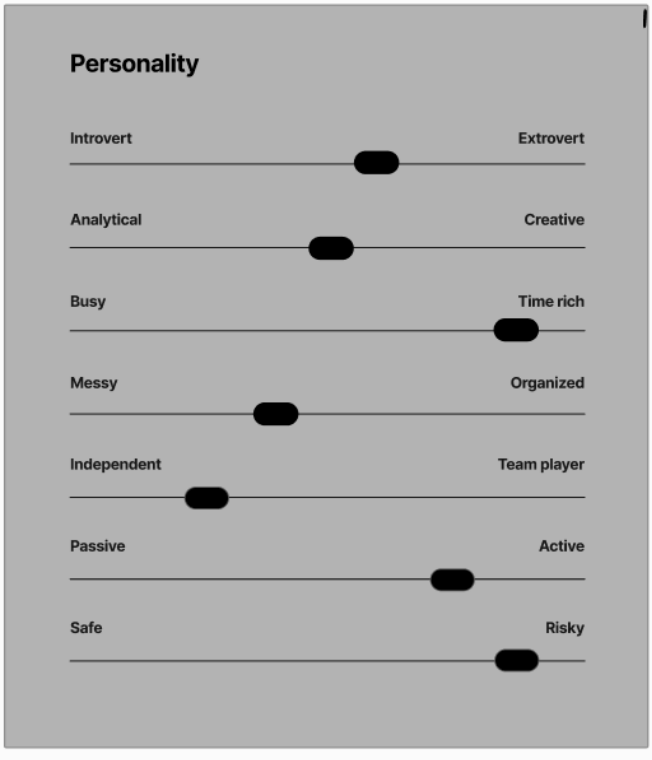
Bio

General Characteristics: An admin working for GatorBites. He has several programming projects that he does on the side. His job is to verify to pending restaurants and check reports submitted by users.

Goals:
1. To have his own startup
2. Wants to buy a house
3. Wants to be able to seamlessly approve and check restaurants

Skills:
1. Well versed in using MySQL
2. Well versed in navigating and using GatorBites

Pain Points:
1. Wants to be able to work from home



Use Cases:

1) Max has to finish his project by midnight and the only way he can do that is if he orders his dinner to his door. He heard about GatorBites from his roommates so he decided to give it a try. He opens the GatorBites website and **searches** what he wants into the search bar. He is then greeted with a list of restaurants that specializes in the cuisine that he is craving. He clicks on a restaurant that catches his eye and adds his favorite menu items to the cart. Before he can checkout, he is greeted with the login page. Since he is an unregistered user, he has to **register** using the sign up page. He also gets student-only discounts because he registered with a SFSU email. He is then able to enter his information and his GatorPass to pay for the food. He also has to enter the building that he is currently residing in and the floor he is on. 20 to 30 minutes later, the delivery driver shows up at his door.

2) Michelle is currently working with a student group and doesn't have time to grab food. She wants to be able to order from her office because she doesn't want to leave the student group alone. She goes on her phone to order from GatorBites. She **logs** into her account and starts to browse restaurants that will deliver under 30 mins. After deciding on one, she adds menu items to her cart. Not being able to decide what she wants, she **adds** and **removes** several menu items before checking out. After she enters her building's information, she is able to go back to helping the student group. 25 minutes later, she is able to pick up her food from the delivery driver that is standing at the door of her office.

3) Charlie is busy grading the midterms that most of his students will probably fail. He wants to be able to grab a healthier food option because that is what his doctor recommends but is not able to pick up in between classes due to his old age. He doesn't have good vision and struggles with reading text on monitors so when he arrives on the GatorBites website, he clicks on whatever looks good. He doesn't have much time before his next class starts so he has to **order** quickly. After placing his order, he gets an estimated time of arrival so that he knows that the person entering his classroom late is not one of his students. He is able to get it delivered straight to his classroom.

4) Aaron is a delivery driver for GatorBites and he is waiting on campus near his bike for an order to come in. He will be **notified** via text message on his phone containing a URL link. This page contains the customer's name and location, as well as the restaurant's address and location. It will also have the food items that the customer ordered and the customer's delivery instructions. Once Aaron gets the notification, he is able to bike to the restaurant to pick up the order and to bike back to the

campus and deliver to his customer. He will also have a map of the campus so that he knows exactly where to go to deliver the order.

5) Ceci is a restaurant manager and wants to get more customers ordering from her restaurant. She hears about a new delivery service called GatorBites from her friends who work in the restaurant industry. Curious, she wants to sign up her restaurant and wants to offer discounts for students to get more of them as customers. In order to sign up for her restaurant, she needs to be a registered user. After creating her account, she is able to create her own page. She is able to **fill** out her restaurant's information like address, hours opened, how expensive her restaurant will be which will be represented by the amount of dollar signs. She is also able to add as many menu items as she wants, giving each menu item a name, price, description, and photo. Descriptions and photos are optional fields. After she has finished **entering** all the information, she will get a confirmation email that her restaurant will be in the process of being approved and to wait 24 hours to be approved.

6) William is an employee of GatorBites and is one of the admin in charge of pending restaurants and handling complaints. He will have **access** to information that only admin users can see. William will be able to **view** pending restaurants for approval and click a button to approve them. He will also be able to view complaints and **reports** from users. These reports can range from inappropriate language or images to a restaurant no longer taking orders.

3 Data Items & Entities

1. User

    1. A user shall create only one account
    2. A user shall have at least one role

2. Admin

    1. A admin shall create users
    2. A admin shall be able to access order reports
    3. A admin shall be at least one user
    4. A admin shall have one role

3. Role

    1. A role shall be linked to many users
    2. A role will have at least one user
    3. A role shall be a restaurant owner
    4. A role shall be a driver
    5. A role shall be a customer

4. Restaurant

    1. A restaurant will have at least one owner
    2. A restaurant can have many orders
    3. A restaurant can have a review
    4. A restaurant will have a menu

5. Menu

    1. A Menu shall be at one restaurant
    2. A Menu shall

6. Menu Item

    1. A menu shall have at least one category
    2. A menu shall have at least one restaurant

7. Review

    1. A review shall be given by a user
    2. A review shall be given to a restaurant
    3. A review shall be given to a driver

8. Driver

    1. A driver shall take on at least one delivery
    2. A driver shall receive a review
    3. A driver shall receive one to many orders

9. Order

    1. A Order shall have at least one user
    2. A Order Shall be placed from at least one restaurant
    3. A Order shall have one driver
    4. A Order shall be made at many restaurants

10. Payment

    1. A Payment shall be made by at least one user
    2. A Payment should have be made by at least one order
    3. A Payment shall have a transaction

11. Delivery

    1. A delivery shall have at least one driver
    2. A delivery shall have at least one order
    3. A delivery shall be made at one to many restaurants

12. Promotion

    1. A Promotion shall be at one to many restaurants
    2. A Promotion shall be given to many customers

13. Notification

    1. A notification shall be sent many users
    2. A notification shall be sent to one to many restaurants

14. Transaction

    1. A Transaction shall be made by at least one to many orders
    2. A Transaction shall be made by at least one to many customers

15. Complaint

    1. A complaint shall be created by a customer

1. User

    a. PK - userid: INT(11)
    b. email : VARCHAR(45)
    c. Password: VARCHAR(255)
    d. Name: VARCHAR(255)
    e. Phone: INT(10)
    f. Role: VARCHAR(25)
    g. Address: VARCHAR(255)
    h. CreatedDate: DATETIME

2. Admin

    a. PK - adminid: INT(6)
    b. Username: VARCHAR(128)
    c. Password: VARCHAR(255)

3. Customer - (customer, driver, restaurant owner)

    a. PK - Customer_id: INT(11)
    b. Role_desc : TEXT(200)
    c. Password: VARCHAR(255)

4. Restaurant Owner

    a. PK- Resaurant_Ownerid: INT(11)
    b. Name: VARCHAR(255)
    c. Phone: INT(10)
    d. Password:VARCHAR(255)

5. Driver

    a. PK - driverid: INT(11)
    b. Driver_name: text
    c. Password:VARCHAR(255)
    d. Make:VARCHAR(25)
    e. Model:VARCHAR(25)
    f. License_plate: VARCHAR(25)
    g. phone:INT(10)
    h. rating :INT(5)
    i. Status: BOOL

6. Restaurant

    a. PK - re_id : INT(11)
    b. re_name: VARCHAR(255)
    c. Website: VARCHAR(255)
    d. phone: INT(10)
    e. address:VARCHAR(255)
    f. City: VARCHAR(255)
    g. State: VARCHAR(255)
    h. Zipcode: INT(5)
    i. Country:VARCHAR(255)
    j. Open: DATETIME
    k. Closed: DATETIME

7. Menu

    a. PK - menuid:INT(11)
    b. FK - Restaurant_id:INT(11)
    c. diet_type:VARCHAR(100)
    d. Description: VARCHAR(100)
    e. Price: Decimal
    f. Qty: INT(999)

8. Menu Item

    a. PK - menu_Item_id: INT(11)
    b. FK - Restaurant_id: INT(11)

c. price - DECIMAL

9. Order
    a. PK - orderid: INT(11)
    b. FK - customerid: INT(11)
    c. FK - menuid INT(11)
    d. Total_amount: Decimal
    e. payment_method :
    f. Payment_status: bool
    g. Delivery_address: Varchar(255)
    h. Status: BOOL

10. Order Report

    a. PK - reportid: INT(11)
    b. FK - driverid: INT(11)
    c. FK - orderid: INT(11)
    d. FK - customerid: INT(11)
    e. FK - transactionid: INT(12)
    f. Date: DATETIME

11. Payment

    a. PK - paymentid:INT(11)
    b. FK - orderid: INT(11)
    c. Payment method: TEXT(25)d
    d. transaction_id:INT(11)

12. Notification

    a. PK - notificationid:INT(11)
    b. FK - user_id : iNT(11)
    c. Message: VARCHAR(255)
    d. Notification_datetime: DATETIME

13. Delivery

    a. PK - deliveryid: INT(11)
    b. FK - orderid: INT(11)

c. FK - driverid: INT(11)
d. Delivery_datetime: DATETIME
e. Delivery_status: DATETIME

14. Promotion

a. PK - promotion _id: INT(11)
b. FK - restaurant_id: INT(11)
c. Promo_code: TEXT(25)
d. Promo_desc: VARCHAR(255)
e. discount: Decimal
f. Start_date: DATETIME
g. End_date: DATETIME

15. Complaint

a. PK - complaintid: INT(11)
b. FK - userid:INT(11)
c. complaint: TEXT

16. Transaction

a. PK - Transactionid: INT(11)
b. FK - Orderid: INT(11)
c. Transaction_type: INT(5)
d. Discount_amount: Decimal
e. Total_amount: Decimal
f. deliver_fee : Decimal

4 Functional Requirements

1. Unregistered User

1.1 Unregistered user shall be able to access and browse the web page.
1.2 Unregistered user shall be able to add, remove, and modify any selected items that are placed inside their checkout cart.
1.3 Unregistered user shall be able to freely search for a particular item or restaurant and be provided with such information.

1.4 Unregistered user shall be able to filter out information with regards to their search such as whether alcohol is being served or if the star rating is above or below a certain threshold.
1.5 Unregistered user shall be able to freely sign up for an account at any point while accessing the page.
1.6 Unregistered users shall be able to access and read reviews on a restaurant based on the star customer ratings.
1.7 Unregistered user shall be able to report bugs.

2. Registered User

2.1 Registered user shall be able to read their own account information.
2.2 Registered user shall be able to update their own account information
2.3 Registered user shall be able to delete their own account information.
2.4 Registered user shall be able to observe their transaction history of previously ordered items.
2.5 Registered user shall be able to access their payment method and account settings at any point while on the web page.
2.6 Registered user shall be able to write a review or a star rating based on food quality.
2.7 Registered user shall receive bonus or star points that add up every order that can be used to provide food at a discounted rate.
2.8 Registered users shall be able to provide a note in the system for the restaurant to read before they prepare the order.
2.9 Registered user shall be able to customize the color theme of their web page.
2.10 Registered user shall be able to report bugs.

3. System

3.1 System shall provide a confirmation number to the registered user after purchase to confirm the transaction.
3.2 System shall provide registered user with estimated wait time as well as updates when the food has been delivered.
3.3 System shall update the transaction log of the registered users order history with the newest order on top of the stack.
3.4 System shall display a sign up page when an unregistered user is checking out with their order as a way to complete the transaction.
3.5 System shall display store hours and availability to users on the web page.
3.6 System shall not display stores that are not open.
3.7 System shall remove stores that consistently receive less than or equal to a 1 star rating.

4. Admin

4.1 Admin shall be able to create their own account and information.

4.2 Admin shall be able to read their own account information.
4.3 Admin shall be able to update their own account information
4.4 Admin shall input the available items for purchase into the system.
4.5 Admin shall accept or decline incoming orders from the system.
4.6 Admin shall update store hours in the system if the restaurant closes beyond any normal period of time.
4.7 Admin shall be able to access and review user feedback.
4.8 Admin shall be able to access the restaurant's transaction history.

5 Non-Functional Requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. All or selected application functions shall render well on mobile devices
4. Data shall be stored in the database on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application shall follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
12. No email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2023. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

6 Competitive Analysis

Feature	GrubHub	DoorDash	UberEats/Postmates	GatorDash
Search	++	++	++	++
Food Categories	+	+	-	++
Map	-	-	++	++
SFSU Discounts	-	-	-	++
On-Campus Direct Delivery	-	-	-	++

+ feature exists; ++ superior; - does not exist

GatorDash shall include features that exist in the market competition, such as a search function, food categories, and a map. It seems that most, if not all, of the competition includes the ability to search from a database of restaurants. Some of the competition includes food categories as well to narrow down the search, however, GatorDash shall have a category list built into the search bar for a superior food category feature. Much of the competition did not include a map on the homepage to help with finding food nearby. GatorDash shall have a map feature on par with the best the competition has to offer. In addition to these features in the market, GatorDash shall feature discounts for members of San Francisco State University, including students, staff, and faculty. All that is required to receive discounts is an active account with a SFSU ID number. GatorDash shall also feature on-campus delivery directly to dorm rooms, classrooms, and other user-defined locations.

7 System Architecture & Technologies

Use	Software
Front End	Pug, CSS

Back End	Node.js, Express,js
Testing	Jest
Version Control	Git
Cloud Service	Google Compute Engine (Cloud Run)
Database	MySQL, CloudSQL
Supported Web Browsers	Safari 16 Google Chrome (version 111.0.5563.64)
API's	Google Places API, Google Maps API, MySQLClient API

8 Team roles

Team Member	**Roles**
Dylan Burns	Team Lead, Document Master
Christian Jackson	GitHub Master
Timothy Polich	Front End Lead
Kevin Aziz	Back End Lead
Wallace Man	Team Member Front End
Jeremiah Ruvalcaba	Team Member Back End

9 Checklist

Task	**Status**

So far all team members are engaged and attending team sessions when required	OK
Team found a time slot to meet outside of class	DONE
Backend Lead, Fontend Lead, and Github Master chosen	DONE
Team ready and able to use the chosen backend/frontend frameworks and those who need to learn are working on learning/practicing	ON TRACK
Team Lead ensured that all team members read the final M1 and agree/understand it before submission	ON TRACK
Github organized as discussed in class (master branch, development branch, folder for milestone documents etc)	DONE
Cite sources such as ChatGPT if used	DONE