SW Engineering CSC648/848 Spring 2021

Our Project Name

Team 05

Team lead: Bryan Caldera / bcaldera@mail.sfsu.edu

Frontend lead: John To

Backend lead: Denny Feng

Github lead: Marco Marino

Calvin Tan

Huan Nguyen

Milestone 1

3/5/2021

History Table

- Version 1 of Milestone 1 - 3/3/21
- Version 2 of Milestone 1 - 3/10/21

Table of Contents

1 - Executive Summary

The students of San Francisco State University maintain a busy lifestyle as they work hard to complete their degrees and stay safe while doing so. Our service aims to help alleviate the stress of daily dining by getting these students their favorite restaurant food delivered to them. By providing the swift, safe, and secure experience that only our service offers, we can help bring a little comfortable normalcy back to the tables of SFSU students,faculty and staff.

With our application, users will be able to easily order from a host of local restaurants. They will be able to tailor search results with a number of categories such as type of cuisine, price, and distance away. To best ensure safe and convenient delivery we allow the user to decide where and when to meet with the driver. If the user wishes to leave a review or browse reviews, then they will be able to post and read reviews of any particular restaurant. Finally, price savvy users will be able to take advantage of deals and offers only available through our website.

The team is made up of 6 Computer Science seniors at San Francisco State University. We all have familiarity with website building and various modern technologies that will facilitate us in making a responsive experience. We have team members with different strengths such as customer service, user interface design, and backend systems. This diverse team will be able to create an experience tailored to San Francisco State's needs.

2 - Persona and Main Use Cases

Student Persona

Nancy

- Full time Nursing student
- Studying on a regular basis
- Living with roommates under university housing
- Most of the time, would rather order out than cook

Goals

- Loves to experience different cultural food
- Would love to spend less time thinking about meal preparations
- The ability to enjoy a delicious meal while studying

Qualifications

- Cooking – 2/5 (basic cooking skills)
- Studying – 5/5
- Technology – 5/5

Limitations

- Have the ability to cook only a few dishes
- Unable to drive to pick up meals
- Have a weekly budget
- Due to student budget, does not want to spend money on transportation to pick up orders

Pain Points

Spend less time thinking about meal preparations. A way to pick up orders that don't require spending money on transportation, and staying on a student budget.

Admin persona

James

- From San Francisco
- Administrator of food ordering and delivery site
- Very strong Technical ability

Bio

James is a website admin. He lives in a San Francisco apartment. He works from home so he needs a way to do his administrative tasks remotely. He enjoys tech but wants to get his work done as fast as possible so he can get back to his true passion of cooking.

Goals

- Wants good functionality over aesthetics.
- Wants fast access to the information he needs to do his job.

Skills

- Very well versed with technology
- Has plenty of IT experience

Pain Point

Doesn't like unintuitive menus that make it hard to find what he’s looking for.

Restaurant Persona

Burger Queen Restaurant

- Full time Nursing student
- Fast food
- Very busy
- Clerk have to serve customer and prepare food
- Rush hours
- Possible miss online order
- Not all clerks know how to use the app
- Clerks do not know how to change order when they cannot fulfill customers' order
- Clerks will have conflict with deliverer, extra charge

Goals

- The restaurant clerks want an intuitive UI. The software should be easy to navigate and to learn.
- It can easily update the online menu when the restaurant is in shortage of some supply.
- Clerks will not miss new orders from online customers.
- Restaurant owner registers to the app.

Pain Points

The restaurant received a big order, and it needed time to prepare. However, the restaurant is in rush hour, and clerks are too busy serving customers in the restaurant. Nobody notices there is an order being placed. Since the restaurant is so busy, it needs effective

Deliverer Persona

Thomas

- From San Francisco
- Delivering foods

Profile

Thomas is a food deliverer, who lives in San Francisco. When he receives requests from customers, he then goes to a restaurant (sometimes more than 1 restaurant) to pick up the orders. Then, he delivers those foods to his customers.

Qualifications

- Driving: 5/5
- Punctuation: 5/5
- Technology: 4/5

Goals

- He would like to have a friendly app to use. So he can do his job better and make more money.
- Be easy to use
- Unfamiliar with SFSU area, would like map guide
- Have a tipping function.

Limitations

- He can’t cook by himself.
- He has to depend on traffic, which can be frustrating from time to time.
- If customers order from different restaurants, he has to go to multiple places.

<u>Use Cases</u>

<u>**Student Use Case**</u>

1. <u>**Searching for a restaurant**</u>

   Nancy is a Nursing **Student** SFSU, and is proficient with technology. She loves to experience different cultural foods on a different daily basis. However, today, she would love to eat, Italian. She noticed on the front page of the website, she sees a section of our website that displays **discounted meals** of the day. Luckily for her, some Italian restaurant had **posted** some of their discounted meals of the day. She then goes toward that section and **searches** for the "Italian" section.

2. <u>**Creates and order for pickup or delivery**</u>

   After the search, she then reviews the search result and sees what's the **nearest** Italian restaurant. Shortly after finding the nearest restaurant, she then checks to see if it requires **pickup** or has the **option** to be **delivered**. If the requirement comes out as to only pickup, she continues her Italian restaurant search until she finds a deliverable option. When Nancy finds deliverable options for her meal, she then sees if the delivery is **free.** If not, she will check out how much the **delivery fee** is and spend according to her budget.

3. <u>**Signs up for account to place the order**</u>

Once everything has fit her budget, she proceeds to **checkout** ,which **prompts** her to **login** or **register**. After logging in Nancy can now **choose** a **delivery time** and **location**. Once she chose her delivery time and location, she will now wait for her order at the specified time and location. When everything is completed, as a **new user**, she is prompted to the **join page,** and has the **option** to **create an account**, so she can **save** her searches and have a faster checkout process next time around.

Admin Use Cases

1. **Approving a new restaurant to service**

   James' job involves approving new restaurants to be added to his platform. He has to log into his admin account and then he looks at a page that shows him which restaurants have applied to be added to the service. He can then view the restaurant and choose to add it to the list of restaurants available.

2. **Removing restaurant from service**

   James is also in charge of removing restaurants from the service. If a restaurant is violating the terms of service such as repeatedly failing to fulfill orders, he removes them from the service.

3. **Viewing database contents**

James also is able to view database contents. He can check for data that has been corrupted or is missing to keep the service running properly.

<u>Restaurant Use Cases</u>

1. **<u>Contacting the customer</u>**

   While completing an order the clerk is notified that they are out of tomatoes. The clerk checks the queue of orders and sees there are several orders that cannot be completed because of the tomato shortage. The clerk contacts all the affected customers to notify them, the clerk offers a refund for the items or the customer can reorder

2. **<u>Checking deliverer position</u>**

   The clerk has just finished several orders and no deliverers are currently available to deliver them. The clerk would like to know where all the deliverers are so they can more efficiently plan their next move. The clerk pulls up the map and checks the map, they notice that a deliverer has almost arrived back at the restaurant. The clerk then hurries to finish another order so the returning deliverer can take it out with them.

3. **<u>Checking order feedback</u>**

   The restaurant manager would like to see which areas in which he and his staff could improve on to better serve customers. The manager opens up the app to check on recent order feedback from customers. Customer reception is generally good however there have been many complaints about orders not having enough sauce or napkins. The manager then instructs the staff to hand out more sauce packets and napkins with every order.

4. **Checking item popularity**

   The restaurant manager is out buying groceries for their restaurant but would like to prioritize items that the restaurant uses the most. The manager then checks the menu and sees which meals are the most popular. This makes it easier for the manager to see which ingredients to prioritize.

Deliverer Use Cases

1. **Using GPS**

   Nowadays, It's difficult to go anywhere without gps. He doesn't want to check his map every 5 minutes. So, he uses the app, which has a GPS system to guide him to ride.

3 - Main Data Items and Entities

General users: Does not need to login/ register

- Id

Approved users: Must be logged in /registered

- Name
- Email
- Password
- Address
- Phone number

Deliverers: Must be logged in /registered

- Name
- Email
- Password
- Number of orders completed

<u>**Restaurants: Must be logged in /registered**</u>

- Restaurant name
- Address
- Restaurant Logo
- Cuisine type

<u>**Admin: Must be logged in /registered**</u>

- Name
- Email
- Password

4 - Functional Requirements

1. Admin Services:
    a. Admins shall be able to view a list of restaurants that have applied to join the service.
    b. Admins shall have an easy to access list of the restaurants that are applying to join the service.
    c. Admins shall have the ability to add and remove restaurants from the service.
    d. Admins shall have a secure page for admins that gives them more control.
    e. Admins shall have the ability to remove users and deliverers who violate the terms set by the service. This could be included in the admin page as well.

   f. Admin shall have direct access to database contents. They can view all tables and edit them.
   g. Admins shall be required to approve all new restaurant registrations before they go live.
2. Registered User Services:
   a. Registered users shall be able to search for their food categories and preferred restaurants.
   b. Registered users shall see discounted or undiscounted meals.
   c. Registered users shall be able to see the distance of their search results of their preferred restaurant
   d. Registered users shall be able to see their spendings to not exceed their daily budgets.
   e. Registered users shall have the option to choose delivery or pick up orders.
   f. Registered users shall be able to see the displayed fees required for the delivery services from different restaurants .
   g. Registered users shall have the ability to choose delivery times and delivery locations.
   h. Registered users shall have the ability to save their searches, their delivery time, delivery locations, to pick up or deliver orders.
3. General User Services:
   a. General users shall have an option to create an account.
   b. General users shall be able to use the search feature
4. Deliverer Services:
   a. Deliverers shall be able to pick orders to deliver
   b. Deliverers shall be able to use GPS feature
5. Restaurant Services:
   a. Restaurants shall be able to apply.
   b. Restaurants shall be able to update their menu.
   c. Restaurants shall be able to access their orders.
   d. Restaurants shall be able to check their store’s performance.

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. Application delivery shall be from chosen cloud server
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 must render well on mobile devices (specifics to be developed in consultation with users e.g. Petkovic)
4. Ordering and delivery of food shall be allowed only for SFSU students, staff and faculty
5. Data shall be stored in the database on the team's deployment cloud server.
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application should follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used
13. No email clients shall be allowed.
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor
simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data
items
16. Application shall be media rich (images, maps etc.). Media formats shall be standard as used
in the market today

17. Modern SE processes and practices shall be used as specified in the class, including
collaborative and continuous SW development
18. The application UI (WWW and mobile) shall prominently display the following exact text on
all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2021 For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

6 - Competitive Analysis

Feature	DoorDash	UberEats	Postmates	Grubhub	Our App
Specific restaurant search	++	++	+	++	++
Order Customization	++	+	+	+	++
Schedule a delivery	+	+	+	+	++
Reviews	+	+	-	+	++
Save favorite restaurants	-	+	++	+	++
Shows nearby dining locations	+	-	+	+	++

- Feature Does Not Exist	+ Feature Exists	++ High Quality

Specific location restaurant search is better because it shows restaurants near sf state only. Schedule a delivery is better because you can have food delivered to a specific classroom on campus whereas competitors would struggle doing that. We will also allow users to customize their orders and leave comments for the restaurant with any requests. Reviews will be allowed only to verified SF State members which cuts down on fake reviews. The nearby dining options will be better because they will show restaurants close to the school and can be sorted by walking distance for students to easily get to.

7 - High-Level System Architecture and Technologies Used

Server Host	AWS t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)
Operating System	Ubuntu 20.04 LTS
Database	MySQL 8.0.23
Web Server	Express 4.17.1
Additional Technologies	Node 14.15.5 LTS React 17.0.1 Redux 4.0.5

8 - Team and Roles

Team lead	Bryan Caldera
Frontend Lead and Document Master	John To
Backend Lead	Denny Feng
Github Master	Marco Marino
Team Members	Calvin Tan Huan Nguyen

9 - Checklist

Done	So far all team members are engaged and attending ZOOM sessions when required
Done	Team found a time slot to meet outside of the class
Done	Back end, Front end leads and Github master chosen
Done	Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
ON TRACK	Team lead ensured that all team members read the final M1 and agree/understand it before submission
Done	Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)