SW Engineering CSC648/848 Spring 2021

Project title: "Campus Cantina"

Milestone 1

Team 04

Member Name	Member Role
Rajdeep Singh	Team Lead (rsingh12@mail.sfsu.edu)
Rinay Kumar	Backend Lead
Bhavani Goruganthu	Frontend Lead / Document Editor
Frederick White	Github Master
German Perez	Frontend Team
Henzon Zambrano	Backend Team

Revisions History:

Draft Number	Date Submitted	Instructor Comments	Date Revised
Draft 1	March 04 2021		

1) Executive Summary:

Motivation:

"Life is what happens when you are busy making other plans" - John Lennon

As digital technology continues to progress, people are migrating towards online platforms for their various needs such as shopping, entertainment as well as ordering a meal for the day.

College campuses represent some of the most concentrated markets for certain types of delivery food and many restaurants find that delivering food items to local college campuses can exponentially increase their order volume when college is in session. In these current times, college students become busier and occupied, but they all still need to consume food and that takes up too much time as customers have to drive to restaurants, order and wait for the food.

The solution created is "Campus Cantina". The motivation of Campus Cantina is to be able to provide a convenient food delivery service to SFSU college students within campus.

Importance of the Application:

Through an online food ordering system, a customer can easily access a restaurant's menu in a hassle-free manner. As lives get busier, more customers want to order the taste of the restaurant at home and eat within its comfortable confines. New ordering and food delivery options serve students and faculty who increasingly want their meals when and where it's most convenient for them.

Project Description:

"Campus Cantina" is a unique and user-friendly online food ordering system for exclusive use by SFSU students, staff and faculty. This web-based service is easily accessible from your handy devices i.e. Laptop, Tablet & Mobile. All you need to do is to register and login using your sfsu email id and start hunting for your favorite food from the nearby restaurants.

Key Advantages:

"Campus Cantina" shall have many outstanding features, out of which the ability to order from multiple nearby restaurants is the dominant one. Additionally, Campus Cantina wants to provide a user-friendly UI experience for SFSU students & faculty to allow ease of ordering food online.

Functions & Services:

By creating a web-based service exclusive for SFSU students and faculties, we aim to provide a variety of food cuisines that our users will be able to choose from. Our service will be able to help busy college students that have no time to physically go to their desired restaurants that they want to order in. Our users will be able to see different restaurants on our website as well as weekly special deals. Our service will also open opportunities for restaurant owners/vendors to be able to advertise their menu and services.

Our Uniqueness:

Our team plans to have an astounding search feature where we'll be simplifying browsing of food menus using category filtering. In addition, we plan to have a more detailed map of the available restaurants and the delivery location.

About our Team:

Our team is not just a group of individuals but is one strong effort put by six innovative minds. We have different strengths in different areas, but we continuously help and motivate each other as a team. Our goal is not only to finish the project we have, but we also hope we can contribute and help other people succeed.

2) Personae and Main Use Cases:

Persona for User Category 1: Kim - Student

About:

- Full time student, active in extracurricular clubs on campus
- Very busy schedule
- Doesn't own a car and currently unemployed
- Tech savvy, uses multiple social media apps

Goals and scenario:

- On her lunch break, she checks CampusCantina for quick and affordable food delivery options.
- After making a selection, she waits for her food to get delivered to her dorm room.

Persona for User Category 2: Sam – Admin

About:

- Works for CampusCantina (regular business hours)
- Very detail oriented and efficient
- Review menu changes each day at a fast pace
- Tech savvy

Goals and scenario:

- Approves restaurant partners before going live.
- Reviews updates ordered by oldest to newest
- Verify that all updates abide by CampusCantina guidelines

Persona for User Category 3: John - Delivery Driver

About:

- Works a second job for CampusCantina on evenings and weekends
- Saving up for a car
- Physically active
- Basic computer skills

Goals and scenario:

- Checks into work by using the CampusCantina app.
- Receives his first pick up location and heads there by bike. Once picked up, he heads to the customer's location.

Persona for User Category 4: Nicole - Restaurant Owner

About:

- Manages her own restaurant in West Portal
- Busy schedule
- Restaurant has low sales during the day
- Basic computer skills

Goals and scenario:

- She registers with CampusCantina and awaits approval from the admin to provide restaurant services.
- She promotes weekly lunch specials in order to increase sales.
- She logs onto CampusCantina and sends an update request with new specials

Use Case 1: Ordering Food

Kim has a 2-hour break from her college classes and feels hungry. She opens up her CampusCantina app to look for restaurants. She's craving tacos, so she uses the search bar to find Mexican restaurants. After making her selection, she browses the menu. The items are divided into categories. In the specials section, she found a Taco Tuesday deal. She chooses how many tacos she wants and then proceeds to the next window. Kim is prompted to login or register with the app before proceeding. She enters her information and creates an account. Before sending her order, she realizes she forgot to order a drink. After adding a coke, she sends her order to the restaurant. The app reminds her to have a cash or card payment on hand for the delivery driver. While she waits for her order to be dropped off. After the driver arrives, she meets them, pays for her food, and closes the door feeling happy.

Use Case 2: Updating Menu

Nicole wants to update her restaurant's weekly lunch specials on CampusCantina. She logs into her account and finds the menu section. She adds new dishes into the specials category and deletes the previous specials. She clicks save and waits for a confirmation message from the admin. Nicole appreciates how quick and easy this task is because she has a demanding work schedule.

Use Case 3: Approving Menu Updates

Sam arrives to work at CampusCantina and logs onto his admin account. He can see a list of updates from restaurants (ordered chronologically). He spends the majority of his day approving or disapproving the updates. He opens his first update request. Once everything is in order, he approves the request. Sam understands that he needs to work at a fast pace in his job while still paying attention to detail in all his requests.

Use Case 4: Pick-Up/Drop Off Orders

John logs into CampusCantina to start working. He receives his first order with the following information: customer's first name, restaurant's address, and the customer's address. He gets on his bike and starts pedaling. He arrives and enters the restaurant. Tells them he's picking up

Kim's order. The restaurant staff hands him the food. He heads to the customer's location. Once reaching Kim's dorm, John hands her her food in exchange for a cash payment. In this whole process, John never contacts the customer directly unless he is experiencing a longer commute time or has issues with the order.

3) List of Main Data Items and Entities - Data Glossary/Description:

Entities:

1. Guest
    - Shall view and search restaurants and menus, see menu item prices and details
    - Shall register an account (only with valid SFSU email)
2. Registered User
    - Shall do everything a guest can do
    - Shall login and logout securely
    - Shall message admin/help
    - Shall add menu items to a cart
    - Shall select delivery location or pick up
    - Shall place orders
3. Partner (Restaurant Owner)
    - Shall do everything a user can do
    - Shall register their restaurant as a CampusCantina partner (per admin approval)
    - Shall upload address, photos, menus
    - Shall view and accept or decline user orders
4. Driver
    - Shall do everything user can do
    - Shall view and accept or decline user orders
5. Admin
    - Shall have full access including everything other entities can do
    - Shall accept or decline restaurants as partners

- Shall accept messages and relay to users, partners, drivers
- Shall delete or remove users, drivers, partners, menus, menu items
- Shall edit delivery and service fees

Data Items (subject to change):

1. User profile
    - Shall contain user full name, id, address, type (student/faculty/staff), phone, email, password.
2. Partner profile
    - Shall contain restaurant name, id, address, phone, category, tags, contact name, contact phone, contact email, logo pics, banner pics
3. Driver profile
    - Shall contain driver full name, id, address, phone, email, delivery vehicle.
4. Admin profile
    - Shall contain username, id, password.
5. Menu
    - Shall contain id, restaurant id, menu items
6. Menu Item
    - Shall contain id, menu id, name, description, price, pictures
7. Cart
    - Shall contain id, items, quantity, subtotal
8. Order
    - Shall contain id, restaurant id, restaurant name, restaurant address, user id, user name, delivery location, cart id, cart contents, tip amount, delivery fee, service fee, total cost, delivery ETA.
9. Message
    - Shall be a message sent from any registered user, driver, or partner to admin, and from admin to anyone.
10. Search Result
    - Shall contain id, restaurant id, restaurant name, category, cost.

4) Initial List of Functional Requirements:

User Actions:

1. SFSU Faculty, Students and Staff shall register for seeking the delivery service of 'Campus Cantina' and shall login with their credentials.
2. Users shall search for local restaurants and their services.
3. Users shall be able to view details of a particular restaurant and browse through their menu before logging in/registering following the concept of "Lazy Registration"
4. Users shall be able to add menu items from multiple restaurants to the cart.
5. Users shall be able to manage the items added to the cart.
6. Users shall either request for delivery in the campus or pick up from the restaurant directly.
7. Users shall be able to update their profile and delivery location details. (Delivery in SFSU campus)
8. Users shall have access to a map of area surrounding campus with restaurants
9. Users shall be able to enter a chat to request help.
10. Users shall be able to filter by favorites.
11. Users shall have access to order history.

Restaurants:

12. Restaurant owners/partners shall be able to register and advertise their services.
13. Site Administrator shall approve the restaurants before being searchable by the user.
14. Partners shall be able to view and manage the meal orders and update the status accordingly.

Drivers:

15. Driver shall be able to login and check the orders assigned to him/her.
16. Driver shall be able to see customer order items.

17. Driver shall be able to see the customer drop off location.
18. Driver shall be able to chat with customers.
19. Drivers shall be able to share their location after signing in so that an order can be assigned to him/her.

Backend Functionality:

20. System shall filter out restaurants by using the search keyword.
21. System shall add a restaurant to the database after approval by the administrator.
22. System shall add orders & user info to the database in real time.
23. System shall assign a nearby driver (close to the restaurant) automatically when an order is placed.
24. System shall verify login credential information and pass status messages to the user.
25. System shall calculate the cost of the orders.
26. System shall update the status of the orders in real time.
27. System shall update the cart based on user changes.

5) List of 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:

Features	Campus Cantina	DoorDash	GrubHub	Postmates	Uber Eats
Category Filter	+	+	+	-	+
Cart	+	+	+	+	+
Restaurant Browsing	+	+	-	+	+
Restaurant Map	++	+	+	+	+
Text Searching	+	+	+	+	+
Menu Binary Searching	+	+	-	+	+

There are a couple key features that we intend to include that some of our competitors do not have, and some features we plan to do better than our competition. The UI experience for restaurant browsing is done in a similar way across much of the competition and we intend to stick to what would be expected while providing our own experience.

GrubHub is one competitor that tried to go too far outside the expected and the result is a poor UI experience. Category filtering is another feature offered by all of the competition except Postmates and as such we intend to include it as well.

Finally, we intend to have a more detailed map of the available restaurants because all users are congregated in a specific area. This means we can show the restaurants on the map itself.

7) High-Level System Architecture and Technologies Used:

Technology Type	Technology	Description
Server Host	AWS Elastic Cloud Computing (EC2)	EC2 - t2.micro - 1 vCPU - 1 GiB RAM
Operating System	Ubuntu Server	Version 18.04
Database	MySQL	Version 8.0.x
Web Server	NGINX	Version 1.18.0
Server Side Language	JavaScript	NodeJS
Web Framework	ExpressJS	Version 4.17.x
Frontend UI Components	ReactJS	Version 17.0.x
Frontend UI Designing	Bootstrap	Version 4
IDE	Visual Studio Code	Version 1.53.2
Supported Browsers	Google Chrome Mozilla Firefox	Most Stable Version

8) Team and Roles:

Member Name	Member Role
Rajdeep Singh	Team Lead (rsingh12@mail.sfsu.edu)
Rinay Kumar	Backend Lead
Bhavani Goruganthu	Frontend Lead / Document Editor
Frederick White	Github Master
German Perez	Frontend Member
Henzon Zambrano	Backend Member

9) Checklist:

- 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 **- DONE**
- 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.) **- DONE**