

SW Engineering CSC 648-848 Spring 2023

Hungry Gator Application

April 27, 2023

Milestone 3

Team 5

Student Name	Email
Maliah Chin	mchin4@sfsu.edu
Hajime Miyazaki	hmiyazaki@sfsu.edu
Mario Leyva Moreno	mleyvamoreno@mail.sfsu.edu
Scott Nguyen	snguyen23@mail.sfsu.edu
Aneida Guadalupe Blanco Palacio	ablancopalacio@mail.sfsu.edu

History Table

Date Submitted	Date Revised
4/27/2023	5/21/23

Milestone 3: Summary

Team Number: 05

Meeting Date: 4/26/30 1800

Our Priority 1 tasks have been reduced to meet the minimal viable product for release on the final day of class. This document contains a summary of feedback as well as what our current priorities are. Below is a list of what we need to do in order to meet our goals.

UI PAGE FEEDBACK:

Restaurant Details page:

Make the text field smaller for descriptions.

One column for all forms and text entries

Enter time to deliver, should be selectable

Add a category pull down to chose instead of typing in the category (to avoid typos)

*** = mandatory field (include this message on the top portion of the forms pages)

Drivers:

Show my orders -> goes to order page where they can pick which order to deliver

Checkout pages:

Create and include new pages for checkout functions

Requirements:

Order, total price, check out

*Use stacks for responsive design!

*Use flags instead of a cart functions to get the user's order requests

CODE REVIEW FEEDBACK:

Include headers in each file

Create a persistent search bar/ typing search

Create the order/checkout function for registered users (Use flags for ordering — avoid cart)

GITHUB FEEDBACK:

Create more detailed comments , a little more depth, have the comments be more specific

DATABASE FEEDBACK:

Create a way for Admin to approve/deny restaurant display requests: Admin on workbench, create a new column that sets the display as false automatically and when the admin changes it to true, we can display the restaurant

Have unregistered users create accounts: Allow for users to be entered and saved into the database

Pull down for restaurant categories on restaurant registration: On the registration form, be sure to have the restaurant chose their category from a **drop down menu** so we can make it more consistent with database

Delivery time selection: Add time to deliver for each restaurant on restaurant registration form
password should be encrypted

Restaurant key is foreign key to category : keep up with the consistency of the database, all of the keys should stem from one place

Fix path names in database for the images in Restaurant table (architecture slides on path names relative paths instead of absolute/ protect repo as one of the assets)

FRONT END FEEDBACK:

Restaurant owners to post food items/descriptions/photos: update menus upload form

Fix the format of the nav bar on the home screen + about us pages! (picture includes below of mockup)

Make the logo bigger on home screen : Allows for the user to see it better

Pull down for restaurant categories on restaurant registration so we can make it more consistent with database → for restaurant registration (instead of typing the category)

Include time for delivery on the time cards : Our most sought after feature is the delivery time so if possible we can sort the time cards based on delivery time

Repeat name of restaurant on menu page : When we display the menu, be sure to include which restaurant the menu is coming from

If there is no results for the users search include options: if there are 0 results for user search, ask them “Try other options” and display other menus!

** during display change 2 results found to 2 results found out of (total restaurants)

FEEDBACK ON TEAMWORK/RISK MANAGEMENT:

Teamwork is improving. The team is more cohesive and mostly responsive.

Risk Management:

Technical skills: Database skills are improving, but still minor issues prevail and may impact projects admin requirement functionality. To solve this issue, we are reaching out to CTO and other sources.

We have done an architecture review to check that developers adhere to MVC pattern, coding style, and minimal agreed documentation.

P1 List of Features:

1. Unregistered Users

- 1.1 Unregistered users shall be able to make an account.
- 1.2 Unregistered users shall be able to search/browse for a restaurant and its menu items.
- 1.3 Unregistered users shall be able to view search results.
- 1.4 Unregistered users shall be able to filter results by category.

2. SFSU Registered Users

- 2.1 Registered users shall inherit all priority 1 unregistered user functionality.
- 2.2 A registered user shall be associated with one account.
- 2.3 A registered user shall be able to purchase food.
- 2.4 A registered user shall be able to select drop off location on campus.
- 2.5 A registered user shall be able to sort restaurants based on delivery time.
- 2.6 A registered user shall be able to log in.

3. Admin

- 3.1 Admin shall be required to approve all restaurant posts before they go live on the application.
- 3.2 Admin shall be able to remove users.
- 3.3 Admin shall be able to remove restaurants

4. Restaurant Owner

- 4.1 A restaurant owner shall be able to register their restaurant on the app and wait up to 24 hours for admin approval.
- 4.2 A restaurant owner shall be able to post a menu of their dishes on the app for users to see.
- 4.3 A restaurant owner shall be able to register as a restaurant owner.
- 4.4 A restaurant owner shall be able to log in.

5. Delivery Driver

5.1 A delivery driver shall be required to register to apply for the job.

5.2 A delivery driver shall be able to access their delivery details.