

Albert Le, Maximilian Mok,

Hoai An Nguyen, Lam Nguyen,

Thu Theresa Nguyen

Panjiyar Krishna, Kevin Pham

Group VI

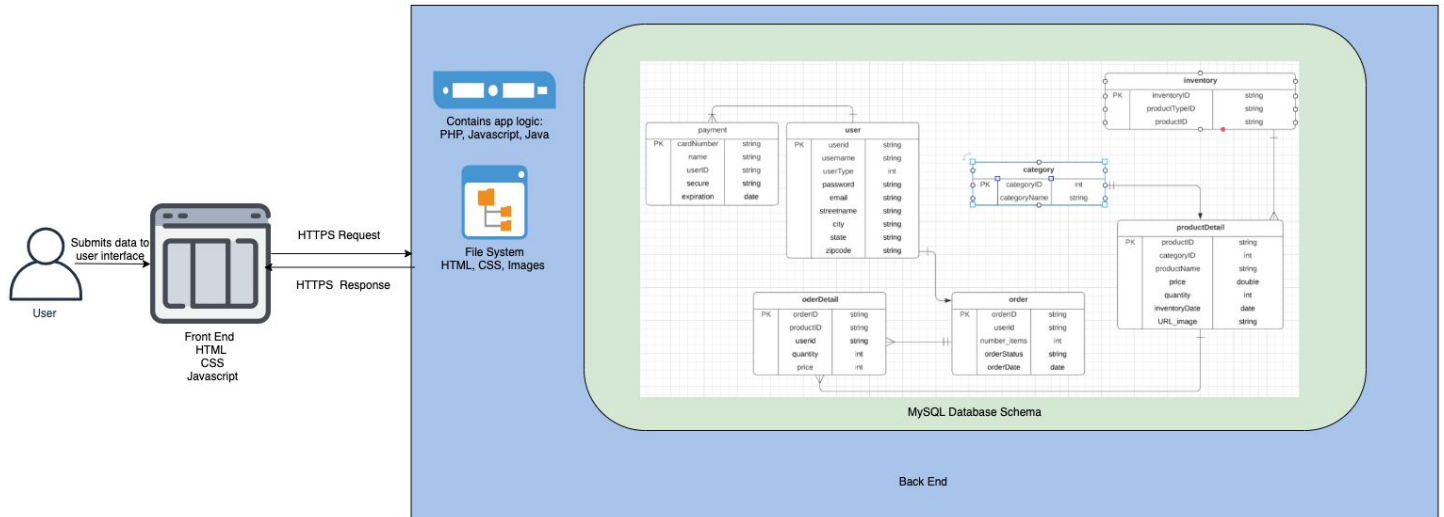
CMPE 131, Section 3

3 December 2020

Part 2 (Updated)

Everything highlighted in yellow are the updated changes

Low-level Design with Details



Hosting service: Google Cloud Platform (GCP)

<http://35.222.200.82/homepage.php>

Logic Represented as Pseudocode

Object	Method Name	Function (what it does)	pseudo
Cart	add	Adds object to cart	Check if object quantity in database. If above zero, add to cart, otherwise add 1 if itemQuantity > 0 Add item to cart Else Print "Item out of stock"
	checkWeight	Calculates the total weight of the shopping cart	cartWeight = 0; for (all items in cart) cartWeight += item.weight return cartWeight
	removeItem	Remove object from cart	Removes item from cart (places item back into inventory database?) Quantity = Quantity - 1 If quantity = 0 Remove from the cart
	showItems	Displays all items in the cart	for(all items in cart) print(item + item.quantity + (item.price*item.quantity)) print (total price)
	calculateTotal	Calculates the total price	Compute the sum of all item prices in the cart. totalCartPrice = 0; for (all items in cart) totalCartPrice += Item.price * item.quantity return totalCartPrice

Store	checkInventory	Displays the different products in the store	for(all items in database) print (item)
	sortByWeight	Sorts the products by weight	sort(items.weight) checkInventory();
	sortByDate	Sorts the products by expiration date	sort(items.date) checkInventory();
	sortByPrice	Sorts the products by price	sort(items.price) checkInventory();
	sortByType	Sorts the products by type	If user_input == item.type; print(database[item])
	search	Look up product in store: User inputs string, search database for product containing that string	If user_input == str.database; print(database[user_input])
Payment	checkOut	Completes payment of products : Ask user to confirm total price, cart inventory, and shipping address. If user does not agree, send back to cart to allow changes. If user accepts: process payment, generate tracking code, update inventory stock.	If (total_price, inventory and shipping address) == yes: print("Enter payment details and click submit") print("receipt number & details") Else: Return to home page
Account	checkAccountExists	Checks if the account exists in the database	checkAccountExists(user) Boolean exists = false; If database contains user Exists = true Return exists;
	changePassword	Updates the existing password, User selects to change personal details as they change passwords.	changePassword(user, newPassword) Account = user; Account.password = newPassword;
	login	Sets the global variables to designated user (logs into the users account) : User enters the specified characters and credentials including email and password to log in.	Input = Enter(email and password) If Input == database[]: print("Login Successful") Else: Return to login page

	register	Registers a new account, User fills up the credentials as name, email, password, address and phone number as details to register for a new account.	registerNewUser(email, password, address, phoneNumber) Account newUser = new Account(); This.email = email; This.password = password; This.address = address; this.phoneNumber = phoneNumber;
	trackOrder	Track a completed order, If user is signed in, show completed orders. Display items purchased, total price paid, last 4 digits of card paid with, shipping address, and shipping code.	trackOrder(order number) Search order number in records User = order number owner Order information = items purchased, total price, last 4 digits of card, shipping address, shipping number Return order information

Project Management Backlog:

Tasks	Description	Date Created	Priority(1=H, 2=M, 3=L)	Estimates	Team Member	Date Started	Date Finished	Comments/ Notes
Documentation	Product description and specifications. Project plan and report	09/24	2	8 weeks	Everyone	09/29	12/03	3 parts Due on 12/3
Logo	Create logo for the store	09/25	3	1 week	Hoai An	09/30	12/01	
Database design	Creating schema and populating database locally	10/08	1	3 weeks	Albert Theresa Max Lam Krishna	10/08	10/29	
User parameters	Create data table containing the user parameters	10/08	1	1 week	Max Albert	10/08	10/15	
Research GCP Hosting	Figure out how to host on GCP	10/08	1	1 week	Lam Theresa Albert Kevin	10/08	10/15	
Cart Functions	Data table containing cart parameters	10/08	2	1.5 weeks	Albert Theresa Max	11/21	11/27	
Product parameters	Create data table containing the product parameters	10/08	3	2 weeks	Albert Theresa Max Lam Krishna	11/07	11/24	
User Interface design	Begin designing layout of web page	10/08	2	3 weeks	Hoai An Kevin	10/08	11/ 20	

Payment Functions	Develop functions for payment	10/08	2	2 weeks	Max Lam	10/15	12/01	
OFS query document	List useful queries for admin	10/29	3	1 week	Krishna	11/01	11/26	
Push all working components to one repository branch	Push to master branch on main repo	11/08	1	2 days	Everyone	11/27	11/29	Everyone should have a working local copy
Admin Panel	Show all necessary tables from in admin view	11/12	2	3 weeks	Lam	11/12	11/30	
Dynamic item page	Create a dynamic version of item page	11/12	2	1 week	Hoai An	11/13	11/18	Display item page according to productID
Centralize php files	Merge php files	11/12	2	1 week	Max	11/13	11/19	Double check with everyone with directory paths
Inventory search	Implement product search	11/12	3	1 week	Albert	11/13	11/19	
Fix CSS on dynamic pages	Check CSS on all pages	11/12	3	2 weeks	Hoai An	11/14	12/01	
Record Video	Record functioning web application	11/19	1	3 days	Theresa	12/01	12/03	12/3 Will be recorded when product is finished
Displaying order tab in account	Connect front end and back end for order tab	11/19	2	1 week	Krishna Lam	11/20	11/28	
Display weight on each product page	Modify dynamic page to show the weight of the product	11/19	2	1 week	Hoai An	11/20	11/27	
Hide the Tabs in account if user is not logged in	Work with sessions and navigation	11/19	1	1 week	Lam	11/20	11/24	

Cart Checkout	Insert items in database	11/19	1	1 week	Max	11/19	11/26	
User Guide	Create a user guide with screenshots	11/19	1	2 weeks	Theresa	11/19	12/02	
Backlog history	Update the backlog management table	11/19	1	2 weeks	Hoai An	11/19	12/03	
Upload to GCP	Upload everything on GCP	11/19	2	2 week	Lam	11/20	11/30	Share the link to updated website
Testing	Test all the links and functions	11/20	1	1 week	Everyone	11/25	12/02	Check the functionality

High-level Test Plan

Test strategy

1. Scope

This document will be reviewed by all team members, and then approved by Professor Frank Butt. Testing will focus on all the functions and external interface of the website. Website Security will not be tested. The Software Testing activities will include:

- Testing connection between frontend and backend
- Testing all the buttons and links
- Testing available fields of different structures
- Testing cart functionality
- Testing checkout functionality
- Testing email and password validation
- Testing card validation
- Testing address (local)
- Testing search feature

2. Test Approach

Testing types:

- Unit Testing: Check every single unit performance
 - Entry Criteria:

- all functional requirements have been defined
 - units are ready for testing
 - a unit testing environment has been set up
- Exit Criteria
 - all test cases have been covered
 - performance of key modules has been tested
 - all the bugs have been found and fixed
- System Testing: Conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements
 - Performance Testing
 - Usability Testing
 - Functional Testing

3. Test Environment

Define the number of requirement and setup required for each environment

Test on different web browsers with compatible versions (Google Chrome, Microsoft Edge, Apple Safari). The website should not have pop up blocked but adobe plugin enabled.

It also depends on the type of operating system and the network whether the OS is Windows, Mac or Linux and the network is either public or private.

Define backup of test data and restore strategy

4. Testing Tools

Automation and Test management tools needed for test execution

Figure out a number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly

Generate script to test core features on a website, such as search, adding to cart, checkout, etc. This script will be run before each branch is merged to master. By using a script, the results will be consistent. If script output fails what is to be expected, debug code and try again.

Track known bugs on GitHub. Assign bugs as a task on Asana so project members will be aware of its existence.

5. Release Control

We will use Github as our version control. A changelog will be created to track what has been changed in the code, as well as mark areas that need improvement. Asana will be used to track our work by breaking down each step into four parts: To Do, In Progress, Testing, and Deployment

6. Risk Analysis

Perform risk analysis inside the web application logic by analyzing code violations that presents a threat to the stability, security, or performance of the code. We can do this by making sure that the database has integrity and constraints while ensuring that the code logic won't compromise its security. The web application risk will be measured during testing to assess the code for both risks within the code itself and between units that must interact inside the application.

The contingency plan to mitigate the risks will be to remove any unstable code that presents vulnerabilities in the web application and to redesign the object functionalities.

7. Review and Approvals

All these activities are reviewed and signed off by the team at the end of the document.

Test Case Descriptions

No.	Objective	Steps	Test Data	Expected Data
Registration & Login				
	Create a new account	Go to “Register a new account Enter user information (username, password, email, phone number, etc) Click on “Register”	username:testingUse (unique) Password: testing123 user: usertest@ofs.com Phone: 696 696 6969	The website directs users to a new page to confirm registration successful. If the credentials are not valid, the failed to create account pops up.
1	Check for valid username and password	Navigate to login page Insert account user and password Press Login	Default data in database: user: usertest@ofs.com Password: testing123	User should be redirected to store with confirmation that login was successful. If login not successful then returns an error message stating that there was an incorrect username or password.
4	Check account exists	Navigate to login page Input username and password Click Login	Default data in database: user: usertest@ofs.com Password: testing123	Should log in if the account exists, or give an error message if the account does not exist. Displays an error message stating that the username or password is incorrect.
Shopping				
3	Test searching item	Input a valid and an invalid string(item name) into the search bar	Default data in database: user: usertest@ofs.com	Should return the item that is searched if the item exists If the item does not

		Press search	Password: testing123 search: banana	exists should return no item and display message that no item exists Returns error message stating that the inputted search is not valid.
5	Check the weight	Login using test account 1. Add item <20lbs Click Checkout 2. Add item >20lbs Click Checkout	Default data in database: user: usertest@ofs.com Password: Testing123	Should have free shipping for orders that are less than 20 lbs, should charge user additional \$5 for shipping for orders that are greater than 20 lbs
2	Test free shipping method	Log in using test account Randomly add items to cart until weight subtotal < 20lbs Navigate to checkout page Press checkout	Default data in database: user: usertest@ofs.com Password: testing123 Add items to cart Checkout	Shipping price of order should be free. Returns fail to do free shipping if cart weight is less than 20 lbs.
6	Check payment information	Login using test account Navigate to account page Click 'Change Payment Information' Update payment info Save changes	Default data in database: user: usertest@ofs.com Password: Testing123 Card: #- 1111222233334444 Date: 01/01 Security Code: 111	Should give error if parameters aren't fulfilled, or update card information. Displays error message stating that one of the card parameters was not correctly imputed.
7	Test final payment	Login using test account Add items to cart Click checkout	Default data in database: user: usertest@ofs.com Password: Testing123	Should complete payment, and give a receipt screen of what was purchased as well as the total price for the items that were checked out. Should give the error when card details don't match to the authorized card account.
8	Out of Stock	Click on add option	Default data in	Should show unable to

			database: 4 apples Instruction: Add 5 apples	add 5 apples, only 4 apples left in stock.
9	Check the internet connection	Clicks on any link in the website	Connection cannot establish	If the internet connection goes off while using the website for showing, it shows error for not establishing internet connection to the server.
10	Session timeout	Leaves the web screen for more than 25 minutes	User session will be out	If the logged in user leaves the web screen for more than 25 minutes then the webpage will show user session out.
11	Cancel the placed order	Stop order if the order placed within 6 hours of placed order before 3pm	Clicks on cancel the order	If the user tries to cancel the placed order 6 hours after 3pm from the time when the order is placed then it won't cancel the order and shows cannot cancel the order.
12	Enlarge the food images	Optimize the food image when pointer is placed on the certain position of image	Moves the cursor on the food image	When the user opens the food image and moves the cursor to the food then the image is supposed to enlarge and show an optimized version but in some food images, it might not enlarge due to high resolution issues.

Test Metrics

Execution and Development

Project timeline: 8 weeks

Meeting plan: 8 Scrum meetings and ~~3 standup meeting per week~~ We ended up with 1-2 standup meetings per week and meetings between different groups established amongst themselves outside of standups.

Test Efficiency and Tracking

$$\text{Passed Test Cases Percentage} = \frac{\text{Number of Passed Tests}}{\text{Total Number of Executed Tests}} \times 100\%$$

$$\text{Failed Test Cases Percentage} = \frac{\text{Number of Passed Tests}}{\text{Total Number of Executed Tests}} \times 100\%$$

Reviewed by

Name	Position/Role	Date
Albert Le	Backend - Database	12/02/2020
Maximilian Mok	Backend - Database	12/02/2020
Hoai An Nguyen	Frontend - User Interface	12/02/2020
Lam Nguyen	Backend - Database/Hosting	12/02/2020
Thu Theresa Nguyen	Backend - Hosting	12/02/2020
Panjiyar Krishna	Backend - Database	12/02/2020
Kevin Pham	Backend - Hosting Frontend - User Interface	12/02/2020