

OSD Product Management System

Low-Level Design

Team 5

Abdurrahman Mohammad

Phuc Phan

Ashley Ma

Camille Pina

Component Design Breakdown

Usage Scenarios

Visitors

View company information: business hours, locations, etc.

- Open the homepage, on the top of the webpage displays the name of the company.
- Scroll down to the bottom of the webpage, the locations of two warehouses and business hours of the customer support and warehouse are displayed.

Browse products with/without an account

- Open the homepage, various recommended items are displayed.
- Click the product commercial image on the home page, a new tab is opened and display details about the product.

Search products with/without an account

- Type the product name in the search bar on the home page and click search button, the page will be redirected to the searching result of the product in both two warehouses.
- On the result page, the preview image, the name and the price of all related products are displayed.
- Choose the filter option on the right/top of the items to limit results
- Click the image or the name of the product, a new tab is opened and display details about the product.

Create account

- Click “login/register” button on the top of the website
- Click “Create account” button under the two input bars of login information
- The login page will be redirected to the register page. Fill out the required information (ex. Username, email and password) requested by the register form on this page, click the “submit” to finish registration.

Sign into account: depending on account type, visitor becomes administrator or customer

- Click “login/register” button on the top of the website
- Enter the email and passwords in the two credential input bars and click “login” button to submit

Administrators

Log into system

1. Click “login/register” button on the top of the website
2. Enters their username and password in the login page on the home page
3. If credentials were correct, user will be led to the administrator view
4. If credentials were not correct, a warning displays under the input text bars

Logout

- Click the “logout” button on the right side of top navigation bar

Add items

- In the administrator portal, user clicks “add item” button, an “add item” form pops up.
- Enters item information required by the form and clicks “add item” button at the bottom of the form.

Search for items

User enters name of item in the search bar, click the “search” button on the right

- User can filter items based on category

Update items

- In the administrator portal, user browses for and clicks add an existing item
- User clicks modify item
- Updates item information and fields

Remove items

- In the administrator portal, user browses for and clicks add an existing item
- User clicks modify item
- User clicks delete item

Create item categories

- In the administrator view, click create new category and enter category name

Remove item categories

- In the administrator view, click edit categories
- Select the delete button next to the category name

Manage administrator accounts

- Click manage accounts
- Click administrators
- Select account
- You can modify the account (username and password) and delete the account

Manage customer accounts

- Click manage accounts
- You can modify the account (username and password) and delete the account

Customers

Log into system

- Click “login/register” button on the top of the website
- Enter the username and password in credential input text bars and click “login” button
- If credentials were correct, user will be redirect to the customer view homepage
- If credentials were not correct, a warning displays under the input text bars

Logout

- Click the “logout” button on the right side of top navigation bar

Browse Products

- Users will be able to browse products in the same way as visitors (described above)

Search for product by name

- When browsing products, enter the product name in the search bar and click enter

Filter products based on category

- When browsing products, click filter and select category to filter

View product information and description

- When browsing, click a displayed item
- You should see item description and information

Save items to cart

- When browsing, select add to cart next to the item

View saved items in cart

- Select view cart from the navbar

Modify the multiplicity of an item in cart: x2, x3, etc.

- Click view cart
- Next to the item, modify the number from 1 to the desired multiplicity of the product

Remove items from cart

- Click view cart
- Select remove next to item in cart

View cart total

- Select view cart
- Locate the field total to view cart total

Select valid shipping options

- Locate shipping
- Select the desired shipping option from the option menu

Checkout items

- Select view cart
- Select checkout
- Select delivery option

View order total

- See the total field with the order total including shipping

View delivery weight

- See the weight field for order weight

Enter payment information

- Select proceed
- Enter credit card information and delivery address and click submit

Account Management

- Click account in navbar
- Users can modify account information and select reset password
- Users can select delete to delete account

View previous order and receipt

- After submitting an order and entering payment information, users will be able to view order receipt immediately after transaction
- Users can also log into their account and select orders in navbar to view previous order receipts

Track order

- Select orders in navbar
- Select order to track
- View order status

View delivery window

- Select orders in navbar
- User can select order to view
- User can see the delivery window

Update delivery address

- Select orders in navbar
- User can select order to update
- Select update delivery address
- If not delivered, enter new delivery address and click submit

Cancel order

- Select orders in navbar
- User can select order to delete
- Click cancel order

Frontend

Presentation Layer

The frontend is the portion that users will be able to view and interact with. All activities will be conducted on the frontend and passed on to the backend for processing. The frontend for this application will be a multi-page website that allows users to perform specific actions on specific pages. Access control will be implemented based on account type as administrators will have pages specific to them and customers will have pages specific to them.

Some pages will be static, and some will be dynamic. Static pages are hardcoded and do not display data that may change. Dynamic pages will have data that will vary alongside the MySQL tables. Dynamic pages will have more function calls compared to static pages and therefore may be slower at times. Whenever input is entered and submitted, it will be sent to an intermediate layer of PHP for processing and sanitization before being passed to PHP methods that contain MySQL statements.

The frontend will be implemented using HTML, CSS, and JavaScript. Bootstrap and jQuery will be used to beautify the pages, make them more interactive, responsive, and user-friendly. This will also help separate the program logic from the frontend logic.

Backend

Intermediate/Business Layer

To limit code clutter and extensively long webpages, the frontend will contain only a few methods which will comprise the intermediate tier. The intermediate layer is responsible for processing input and calling multiple PHP methods with MySQL statements to store data. This is the heart of the program and will combine the frontend with the database backend. All non-database related logic will be done here. This layer will send and receive data between the frontend and MySQL backend.

Data Layer

The MySQL statements will be implemented in this layer. They will contain additional data sanitization and will contain methods for search, insert, update, and delete. This layer will connect with the database and will implement logic and security mechanisms to ease and speed up queries.

Database

The MySQL layer is responsible for conducting transactions with the database. It will perform the queries and interact directly with the database.

Additional API

Google Maps API will be used to calculate the shortest or fastest distance between two locations. Every time an order is placed, the system will detect its delivery type (pickup, truck, drone), date, and location. Using this information, the system will use the Google Maps API to plot an optimized delivery route and update the routes of the affected orders and packages. Doing so will speed up delivery time.

Logic

Below is a breakdown of some of the general logic of this program. Methods will be implemented in detail and will contain similar or same signatures and structures. For brevity, only major functions have been included.

Homepage

```
// Web application starts: print page
printHomepage()

// Company information: business hours, locations, etc.
displayCompanyInfo() {
    printInfo() // Print company information
    printAddress() // Implement Google map API to display address
}

// Browse featured products with/without an account
listFeaturedProducts(){
    // Request information from Table featured product
    printShopping() // Display on home page
}

// Search products with/without an account
Search(keyword,) {
    If(keyword != empty)
        submitSearch(keyword) // Fetch user keyword input    printSearchResults() //
        Redirect to search result page, with keywords
}

//On the search result page, customer filter results
FilterSearch(keyword, filter) {
    If(keyword != empty)
        submitSearch(keyword, filter) // Fetch user keyword input
        printSearchResults() // Redirect to search result page, with keywords
}
```

Sign in page

```
// Print sign in page: Display username and password form
printSignIn()

// Sign into account
username = getUsername() // Fetch user input in the form when submit
password = getPassword() // Fetch user input in the form when submit
signIn(username, password) {
```

```

        if(Authenticate(username, password))
            if(getType(user) == admin)
                adminRedirect()
            else
                customerRedirect()
        else
            printIncorrectUser()
    }
Authenticate(username, password) {
    If(user entered username && user entered password){
        If(username exists in the user information database){
            // Fetch salt and salted password hash.
            // Hash user input password with salt
            If(user input hash == fetched hash){
                Return true;
            }
        }
    }
    Return false;
}

CreateAccount()
    printCreateAccount() // Display registration form
    getInput() // User Enter information
    If(submit) // User click submit
        if (!user fill all required information)
            If(password match format)
                createUser (username, password, email, phone, address)
            else
                PrintWarning()
        else
            PrintWarning()
    }

createUser (username, password, email, phone, address){
    generateSalt()
    hashPassword(password, salt)
    insertUser (username, password, email, phone, address)
    print(registration successful!)
}

Logout() // Loses the user's cookie which authenticates the user on every site

```

Administrator

```

// Print admin portal: Display administrative function menu
printAdminPage()

```

```

adminOption() { // Click button, redirect to different function page

```

LOW-LEVEL DESIGN

```
    InsertItem(itemID, name, price, weight, description, categoryID)
    UpdateItem(itemID, name, price, weight, description, categoryID)
    DeleteItem(itemID, name, price, weight, description, categoryID)
    createCategory(ID, name)
    deleteCategory(ID, name)
}

// Insert/Create new item
InsertItem(itemID, name, price, weight, description, categoryID) // Leave unused fields blank

// Update item in inventory
UpdateItem(itemID, name, price, weight, description, categoryID) // Leave unused fields blanks

// Delete item in inventory
DeleteItem(itemID, name, price, weight, description, categoryID) // Leave unused fields blanks

// Search item in search bar
SearchItem(itemID, name)

// Filter item based on selections
FilterCategory(category)

// Create category
createCategory(ID, name)

// Delete category
deleteCategory(ID, name)

// Manage accounts
manageAccount() {
    updateUser(userID, first name, last name, email, phone, address)
    updateAccount(userID, username, password, accountType)
}
```

Customer

```
// Add item to cart
addToCart(itemID) {
    insertCart(userID, itemID, 1)
}

// Remove item from cart
removeToCart(itemID) {
    deleteCart(userID, itemID)
}

// Update multiplicity
updateToCart(itemID) {
    updateCart(userID, itemID, 1)
}
```

```
// Search item in search bar
SearchItem(itemID, name)

// Filter item based on selections
FilterCategory(category)

// View item info
viewItem(itemID)

// View cart
viewCart() {
    printCart()
    printTotal()
    printWeight()
    removeItem(userID, itemID)
}

// Checkout items
Checkout() {
    selectShipping()
    enterPayment()
    printReceipt()
}

// Manage account
manageAccount() {
    updateUser(userID, first name, last name, email, phone, address)
    updateAccount(userID, username, password, accountType)
    printHistory(userID)
    trackOrder(orderID)
}

// View purchase and transaction history
printHistory(userID) {
    printPurchases(userID)
    printTransactions()
}

// Track order
trackOrder(orderID) {
    printOrder(orderID) // View order
    printDelivery(orderID) // View delivery window
    cancelOrder(orderID)
    updateAddress(address)
}

// Cancel order
cancelOrder(orderID) {
```

```
        deleteOrder(orderID)
        deleteDelivery(orderID)
        deletePurchases(orderID)
        deletetransactions(orderID)
    }

    // Update address
    updateAddress(address) {
        updateDelivery(orderID, address)
    }
```

Database Schemas

Item(itemID, name, price, weight, description, categoryID)

Inventory(warehouse_number, itemID, number_in_stock)

Categories(categoryID, name)

User(userID, first name, last name, email, phone, address)

Account(userID, username, password, accountType)

Cart(userID, itemID, multiplicity)

Transactions(transactionID, userID, credit card, timestamp)

Order(orderID, userID, transactionID, shipping_option, total, weight, address)

Purchases(orderID, itemID, multiplicity)

Delivery(orderID, route, delivery_date)

Backlog

Frontend

Create interactive frontend pages only. Include unimplemented methods which may be implemented simultaneously or later to make features work.

Task Name	Task Description	Date Created	Priority (1=H,2=M,3=L)	Estimates (days)	Team Member	Date Started	Date Finished	Comments/Notes
Homepage	Implement homepage	9/28/20	1	1				
Sign in page	Implement sign-in page	9/28/20	1	2				
Administrator portal homepage	Implement Administration page	9/28/20	1	3				
Create item and its pages	Implement view for sign	9/28/20	2	3				
Manage categories (frontend)	Implement categories creation page	9/28/20	3	1				
Search feature (frontend)	Implement search page	9/28/20	2	2				
Update items page	Implement an update item page	9/28/20	1	1				
Manage administrator accounts	Implement administrator management view	9/28/20	1	2				
Manage customer accounts	Implement customer management view	9/28/20	1	1				
Logout	Implement logout view	9/28/20	1	1				
Customer portal homepage	Implement customer login view	9/28/20	2	3				
Item view	Implement item view	9/28/20	1	2				

LOW-LEVEL DESIGN

Cart view	Implement cart view	9/28/20	1	3				
Payment view	Implement payment view	9/28/20	1	1				
Delivery view	Implement delivery view	9/28/20	1	2				
Account management	Implement personal account management	9/28/20	3	2				
Track order	Implement order tracking view	9/28/20	3	1				
Order management	Implement order managing view	9/28/20	3	3				

Backend

Intermediate/Business Layer

Implement methods that make features work. This is the heart of the program. Include unimplemented data layer methods which may be implemented simultaneously or later.

Visitors

Task Name	Task Description	Date Created	Priority (1=H, 2=M, 3=L)	Estimates (days)	Team Member	Date Started	Date Finished	Comments/Notes
View company information: business hours, locations, etc.	Implement methods	9/28/20	1	1				
Browse products with/without an account	Implement methods	9/28/20	1	3				
Search products with/without an account	Implement methods	9/28/20	2	2				

LOW-LEVEL DESIGN

Create account	Implement methods	9/28/20	1	1				
Sign into account	Implement methods	9/28/20	1	1				
Order management	Implement methods	9/28/20	3	2				

Administrators

Task Name	Task Description	Date Created	Priority (1=H,2=M,3=L)	Estimates (days)	Team Member	Date Started	Date Finished	Comments / Notes
Admin log into system	Implement methods	9/28/20	1	1				
Logout	Implement methods	9/28/20	1	1				
Add items	Implement methods	9/28/20	1	2				
Search for items	Implement methods	9/28/20	2	3				
Update items	Implement methods	9/28/20	1	2				
Remove items	Implement methods	9/28/20	2	2				
Create item categories	Implement methods	9/28/20	3	1				
Remove item categories	Implement methods	9/28/20	3	1				
Manage administrator accounts	Implement methods	9/28/20	2	2				
Manage customer accounts	Implement methods	9/28/20	2	2				

Customers

Task Name	Task Description	Date Created	Priority (1=H,2=M,3=L)	Estimates (days)	Team Member	Date Started	Date Finished	Comments / Notes
Customer log into system	Implement methods	9/28/20	1	1				
Logout	Implement methods	9/28/20	1	1				
Browse Products	Implement methods	9/28/20	1	2				

LOW-LEVEL DESIGN

Search for product by name	Implement methods	9/28/20	2	3				
Filter products based on category	Implement methods	9/28/20	3	3				
View product information and description	Implement methods	9/28/20	1	2				
Save items to cart	Implement methods	9/28/20	2	2				
View saved items in cart	Implement methods	9/28/20	1	2				
Modify the multiplicity of an item in cart: x2, x3, etc.	Implement methods	9/28/20	3	1				
Remove items from cart	Implement methods	9/28/20	2	1				
View cart total	Implement methods	9/28/20	1	1				
Select valid shipping options	Implement methods	9/28/20	1	2				
Checkout items	Implement methods	9/28/20	1	2				
View order total	Implement methods	9/28/20	1	1				
View delivery weight	Implement methods	9/28/20	3	1				
Enter payment information	Implement methods	9/28/20	2	2				
Account Management	Implement methods	9/28/20	2	3				
View previous order and receipt	Implement methods	9/28/20	3	2				

LOW-LEVEL DESIGN

Track order	Implement methods	9/28/20	3	2				
View delivery window	Implement methods	9/28/20	3	1				
Update delivery address	Implement methods	9/28/20	3	1				
Cancel Order	Implement methods	9/28/20	3	1				

Data Layer

Create MySQL tables and include methods for insert, delete, search, and update for all tables with input sanitization and security features.

Task Name	Task Description	Date Created	Priority (1=H,2=M,3=L)	Estimates (days)	Team Member	Date Started	Date Finished	Comments/ Notes
Item(itemID, name, price, weight, description, categoryID)	Implement db methods	9/28/20	1	2				
Inventory(warehouse_number, itemID, number_in_stock)	Implement db methods	9/28/20	1	2				
Categories(categoryID, name)	Implement db methods	9/28/20	2	2				
User(userID, first name, last name, email, phone, address)	Implement db methods	9/28/20	1	1				
Account(userID, username, password, accountType)	Implement db methods	9/28/20	1	2				
Cart(userID, itemID, multiplicity)	Implement db methods	9/28/20	1	1				
Transactions(transactionID, userID, credit card, timestamp)	Implement db methods	9/28/20	2	1				
Order(orderID, userID, transactionID, shipping_option, total, weight, address)	Implement db methods	9/28/20	1	1				
Purchases(orderID, itemID, multiplicity)	Implement db methods	9/28/20	1	1				
Delivery(orderID, route, delivery_date)	Implement db methods	9/28/20	2	1				

