Project Requirements
StoreFront

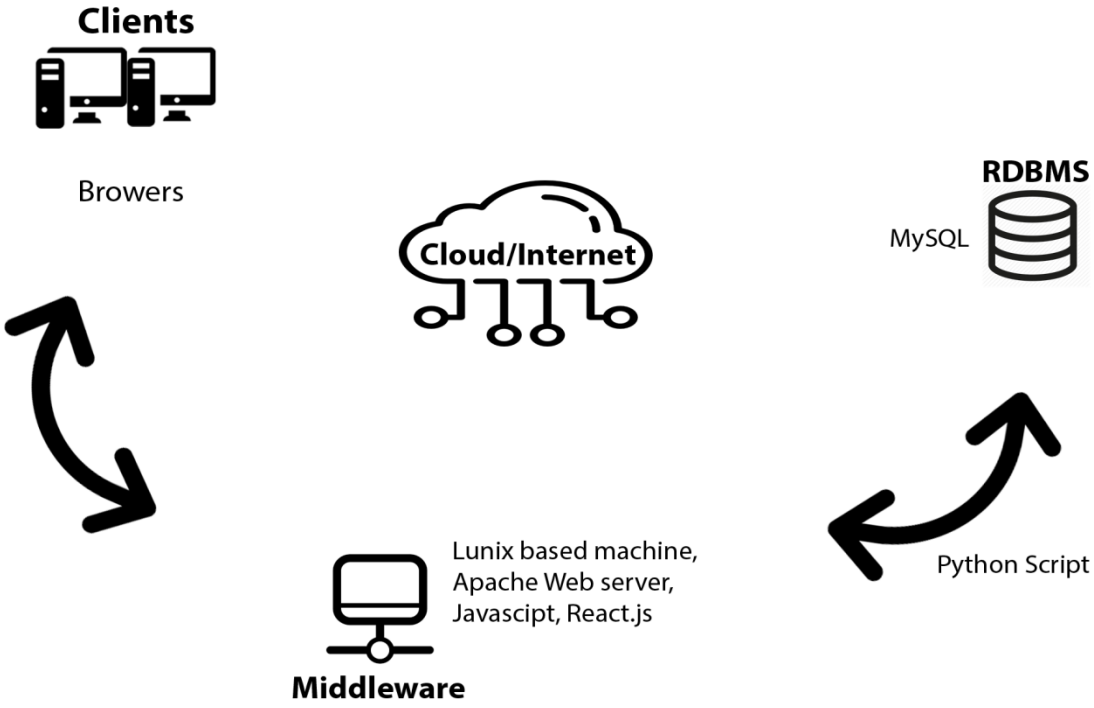
CMPE 157A-01 Team 30
Aaron Warren
Phu Tran
Evan Ugarte

Project Overview

XXXXXXXXXXXX

System Environment

XXXXXXXXXXXX



Functional Requirements

Our system is design for customers of all ages and gender. The system is design and target everyone ranging from those who wishes to purchase food from a store to those who just want to browse the items on the webpage. Users can access the website anytime and anywhere as long as they have access to the internet and any device that can connect to the browser and internet such as desktop, mobile phone, etc. There are two types of general users for this StoreFront application: registered and unregister users. As for the registered user, they can enjoy all of the features of the StoreFront application including browsing, select multiple items to purchase, add to shopping cart for later purchase, save favorate items, modify payment information/method, and update their personal account information.

Functional requirements features will include:

- Create account/sign-up
    - Users will be able to create their account by entering their email which is the username, password, and other additional information. As for those who already have an account, but still try to sign-up for one, the website will display the message that letting the user know if their email has already been use to account an account before.
    - The system will check if the email and username already exist in the database. If email/username did not exist, the system will record their information inside the database in a secure way especially for their username and password and display a message to the webpage to let users know that they have create an account successfully. If email/username did exist, the system will display an error message to let users know that they have unsuccessfully create an account.
- Login/sign in
    - Users can login to the webpage by entering their username and password.
    - The system will check to see if the given username and its associated password are the same with the information that are stored in the database. If it is, the system will allow the users to access their account. If it is not, the system will display an error message through the webpage to let the user know what is wrong.
- Browse items
    - Users can check out all the items that are displaying on the webpage including the image of the item, its name, and price.
    - The system will access the information in the database to display the popular items in a specific format on the webpage for the user to see when they first login to the page.
- Search items
    - Users can use the system's search engine to search for specific items such as by category, name, etc.
    - The system will access the database, sort out specific items, and display it as the result of the user search. If the items user looking for does not exist or out of stock, display a notify message to let users know.
- Select items/add items in the cart
    - Users can select the items that they are interested and add to the cart to purchase later while continue to browse the shop.

    - System will keep track of current items in the cart inside the database for later use and only remove when the user delete the items or they have ordered the item(s).
- Delete items in the cart
    - Users can remove any item that they no longer want from the their shopping cart.
    - The System will access a particular relation in the database and delete the entity of the specify item(s) that the user wishes to remove.
- Add new payment method
    - Users will be able to add a new credit card as a method payment by entering card number, card type, holder name, CVV (card verification value), and card expiration date.
    - System will check with the database to see if the card number already exist or not. If it is, let the user know that this payment method already exist through the website. If it not, store all this information in a secure way in the database
- Delete payment method
    - User will be able to delete exist payment method of their choice.
    - System will access the database and delete the entity that match the value that the user selected.
- Select payment method
    - Users can select already exists method that they have created to check out their order faster.
    - System will use the data in the user chosen payment method to access the database, retrieve the card information, and charge the user according to the cost of items that are currently in the cart.
- Edit/update account information
    - Users can change their password, username, and other personal information such as date of birth.
    - After ask for additional information to verify the user, system will access the database and modify the entity inside a particular relation according to the changing information.

Non-functional Requirements

- Security
    - Users's login information, username and password, as well as their payment method information will not be store directly inside the database. Instead, we will generate a key know as salt (the term use in cybersecurity) and add it to these information and then hash it. We will then use the result of the hashes to store inside the database. By doing this, we can minimize the users' information from leaking out in case the database get hacked or hijacked.

Non-functional requirements include many things such as performance, scalability, security, endurance, usability, and stability. Our goal for these non-function requirements would be to hit most of these for our final product. For performance, we plan on having our website able to support a large amount of users. For

scalability, we are going to try to efficiently manage how we access our database to allow for easy expansion as well as code our front-end in a way where we could easily modify one section without effecting the others. For security, we are planning on working making sure our authentication system works effectively for individual accounts. For usability, users will be able to easily and effectively navigate our website regardless of their personal circumstances. And finally for stability, our website should be able to handle a large amount of users without effecting another users experience. Overall, the website should be easy to use for both users and programmers alike as well as being able to efficiently handle large numbers of users.