Team 10

Shopaa
Software Architecture Document

Version <1.1>

Revision History

Date	Version	Description	Author
24/12/2022	1.1	fill 5. and 6.	Nguyễn Thanh Phong and Lê Anh Vinh
26/12/2022	1.2	edit 3 tier Architecture models picture	Nguyễn Thanh Phong

Table of Contents

Software Architecture Document

1. Introduction

- Purpose:
  This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.
- Scope:
  This Software Architecture Document provides an architectural overview of the Shopaa Website. The Shopaa provides a place to sell and buy typical local food where people can sell and enjoy local food. This document describes the current architectural structure of our website and is intended to be read by future developers who will join the project, and create and interface their own website of this project.
- Definitions, Acronyms and Abbreviations:
  + HTTP : Hypertext transfer protocol - a communicating method between web browsers and web servers
  + GUI: Graphical User Interface - interact with user by interface

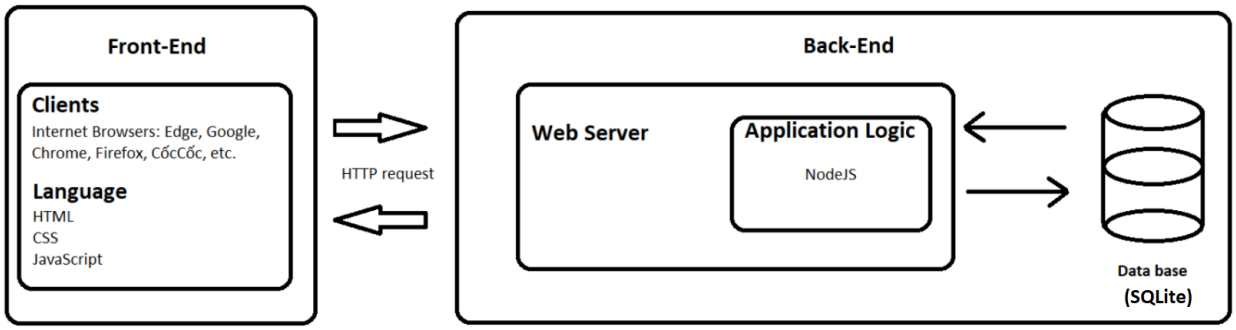
2. Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impacts on the architecture and also captures the special constraints that may apply

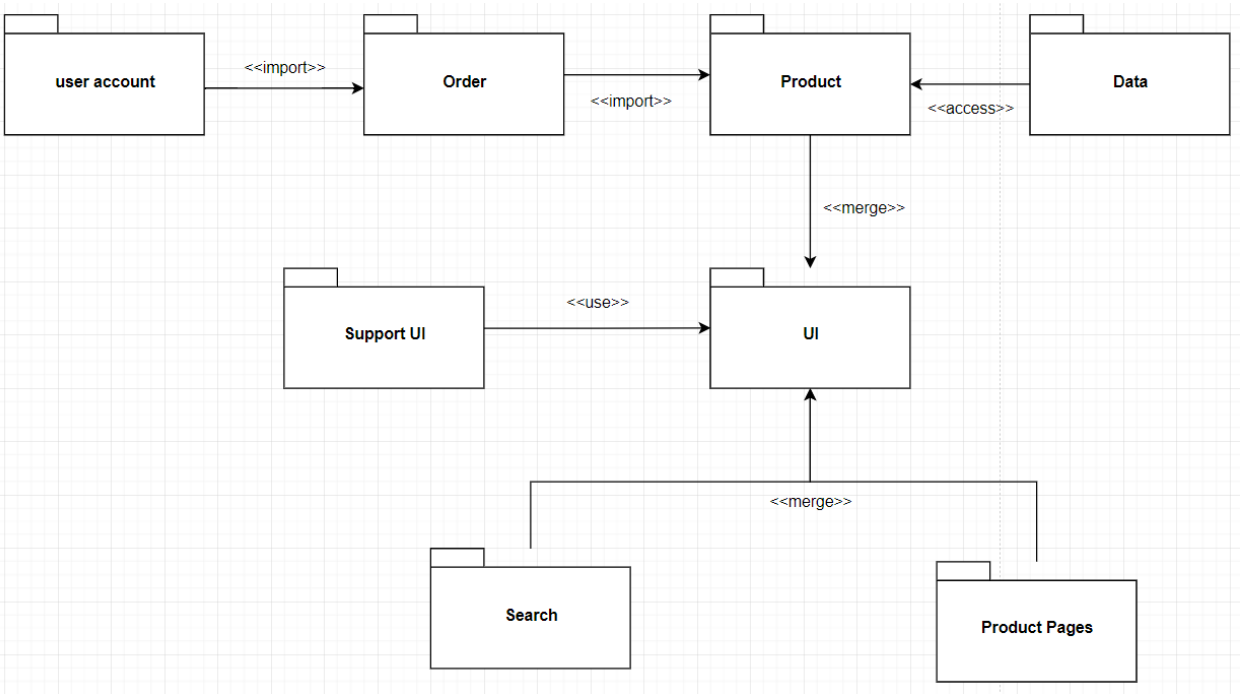
- **Performance**: the web should reply to the client's commands must be under 10 seconds
- **Availability:** the system will be available 24 hours a day, 7 days a week except when maintenance activities
- **Security:** authenticate the user with login using at least a user name and a password. For internet access,we need the following requirements:
  + **Confidentiality:** the data must be encrypted so that it can't be viewed by others
  + **Data integrity:** guarantee the data sent across the network can't be modified by a tier
- **Persistence:** data persistence will be addressed using a relational database
- **Capacity:** server must be large enough to store data.

3. Use-Case Model

4. Logical View



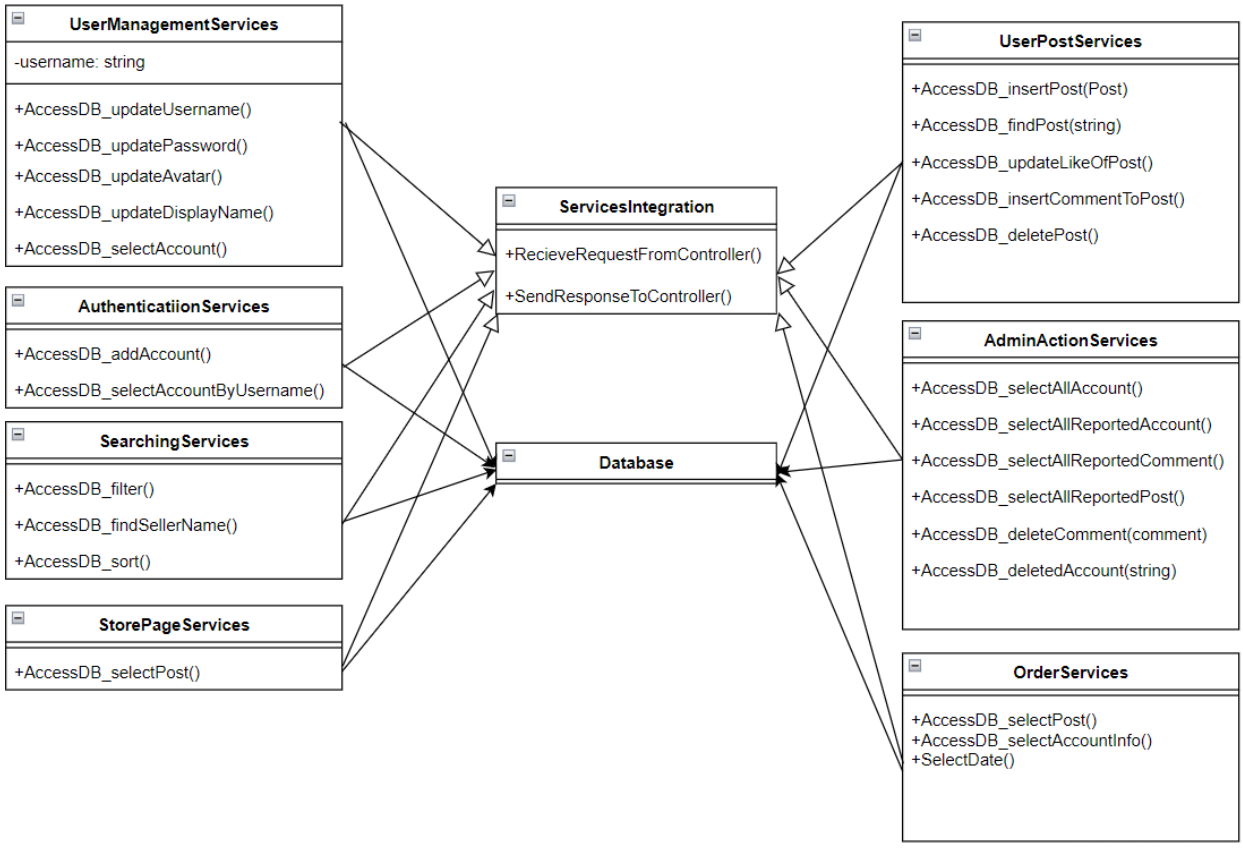
(3-tiers Architecture Models)



(Packages diagram for whole system)

4.1. Component: Service Integration

The Service Integration is constructed to communicate between client and server side. From the client, the Integration will receive the commands from the Controllers, send them to the server (by using GET and POST methods) to request the server to access the database. Then, the accessed data will be sent back to the client side to display on the GUIs.



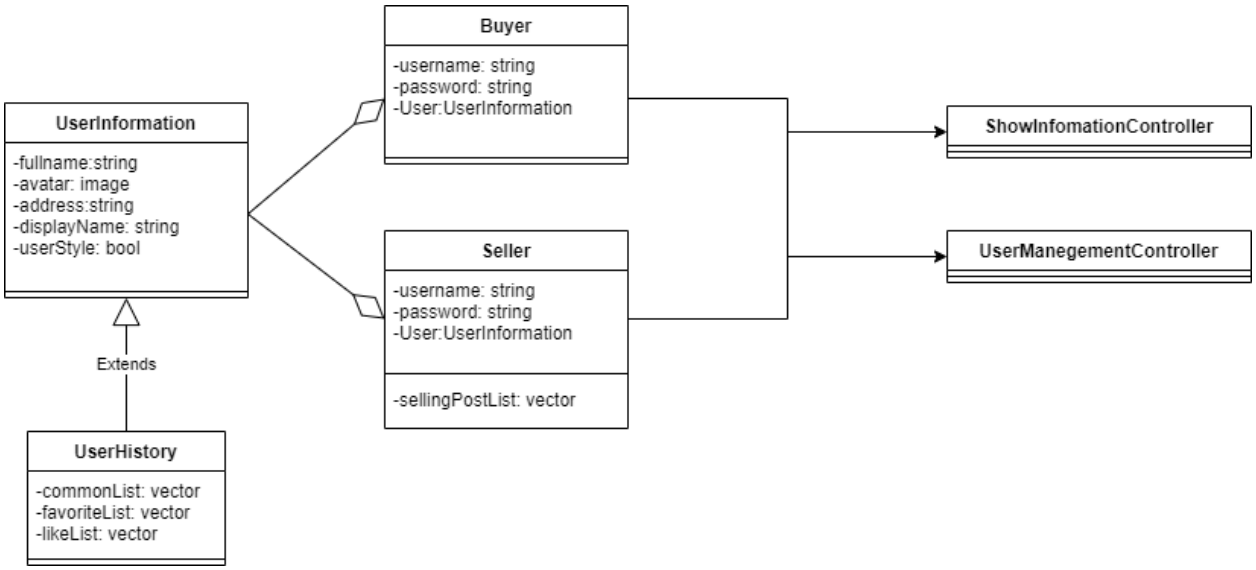
4.2. Component: User Management

User history will save the list of favorites created by the user, recommend new foods, the list that the user has liked

There are 2 types of users: Buyer and Seller, Seller will be able to post food items for sale

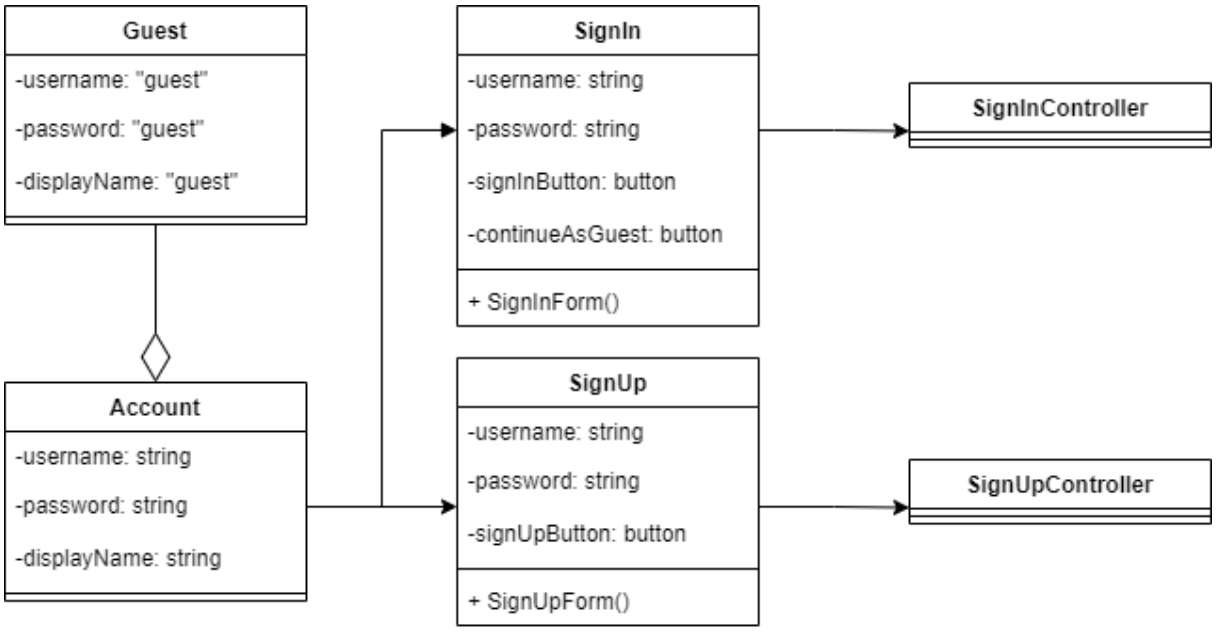
The data will be processed by Show Information Controller to present Buyer Information, Seller Information, a list user History

When the user change the information, the User Management Controller will assist in handling



4.3. Component: Authentication GUI

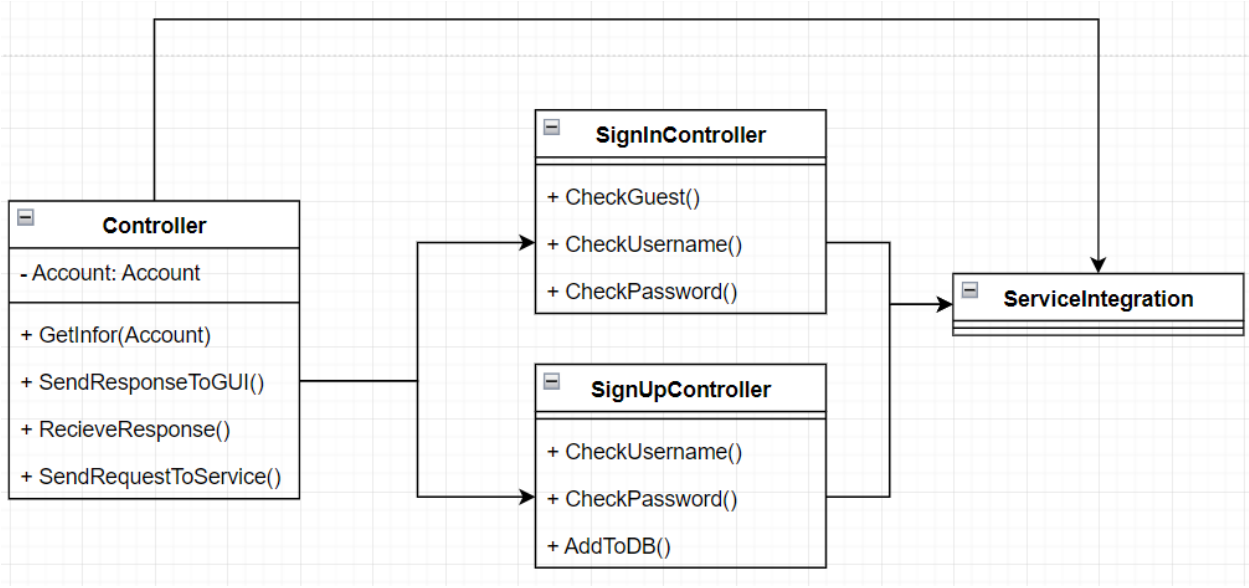
At the beginning, the visitor to the website will be a guest. The guest will see 2 forms: Sign In and Sign Up.
If the guest chooses to sign in, they will be asked to enter a username and password, and there is have a submit button, click it to sign in.
If the guest chooses to sign up, they will be asked for information to sign up. After that, they will sign in.
After signing in successfully, they will be directed to the homepage.
All “Sign In” data that users enter into SignInController and SignUpController will be sent to their appropriate address, controller, respectively. The controller will process and process the input data



4.4. Component: Authentication Controller

In the SignInController, the controller will receive a response from the GUI, it will get a list account from Database and check the account and password. ServiceIntegration will send the response whether sign in is successful or not, then the Controller will send that response to the GUI

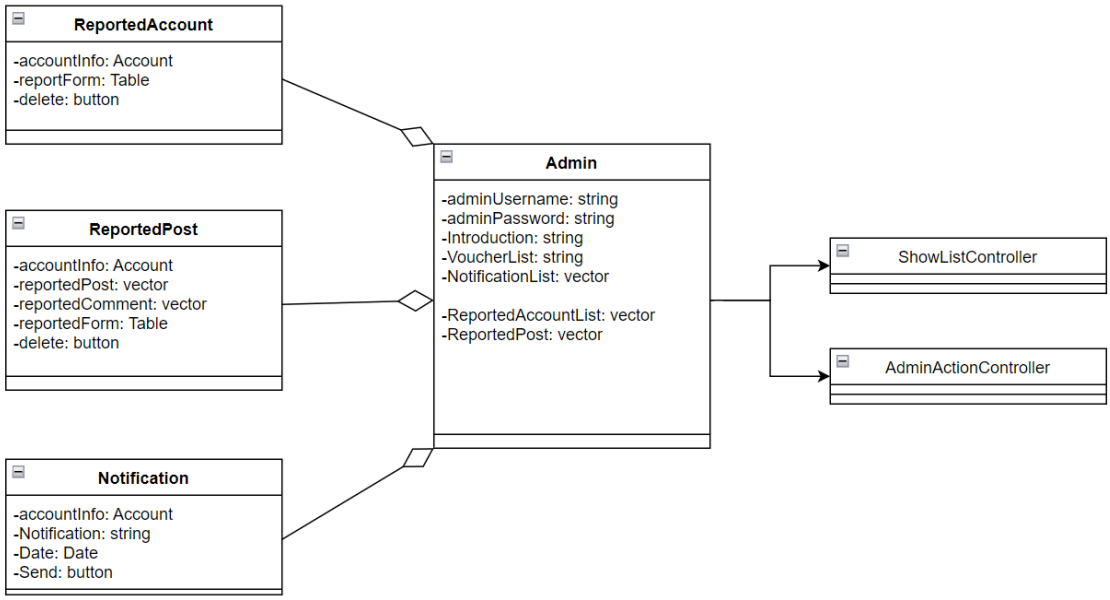
In the SignUpController, The controller will receive a response from the GUI, it will get a list account from Database and check if the username exists or not, if the username doesn't exist, add that account to Database through AddToDB(). Finally the Controller will send that response to the GUI



4.5. Component: Admin GUI

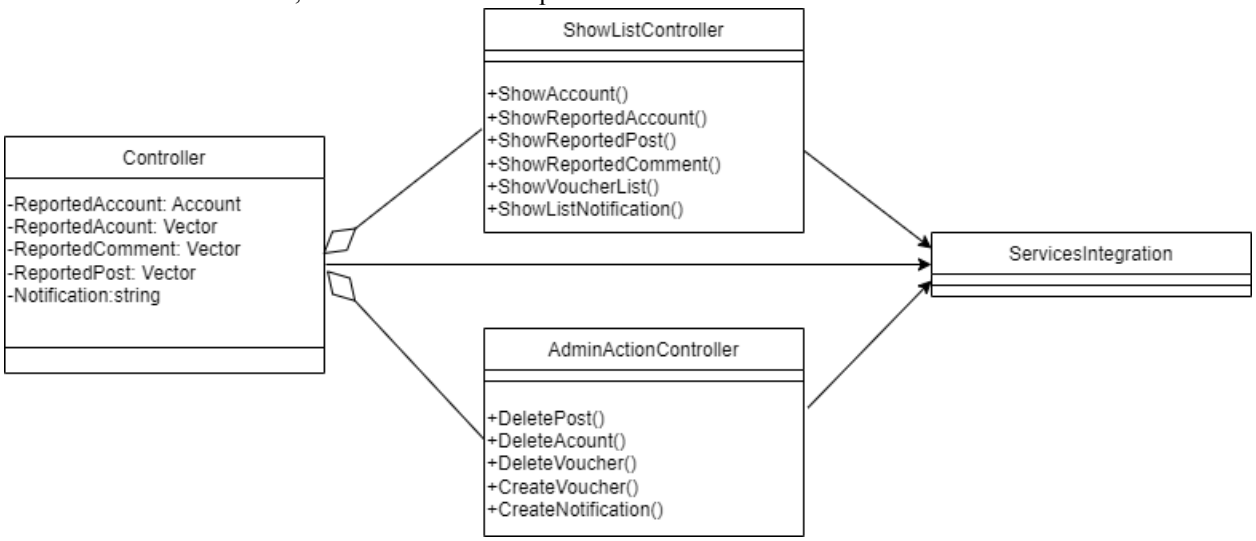
An administrator can view a list of accounts and reports, furthermore create vouchers and notifications for all users. The data will be processed by ShowListController to present a list of accounts, lists of reported posts, accounts, and comments on the admin screen.

The admin can “Delete” a post, comment or “Ban” an account, “Create” notification, voucher for users, the AdminActionController will handle it.



4.6. Component: Admin Action Controller

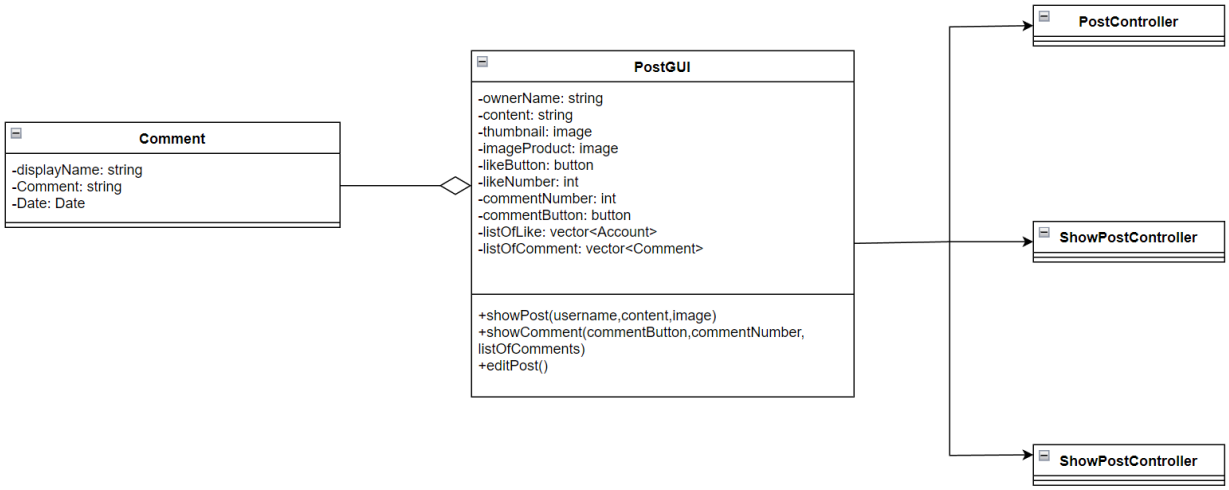
Functions for the admin consists of: display list Account, Reported Account, Reported comment, Voucher, List notification. Admin can delete posts, accounts, vouchers, notifications and they can create vouchers for the website. All the functions will be serviced by Services Integration and give the response to the GUI.
In Admin Action Controller, the administrator can perform functions such as delete and create at the GUI.



4.7. Component: Post GUI

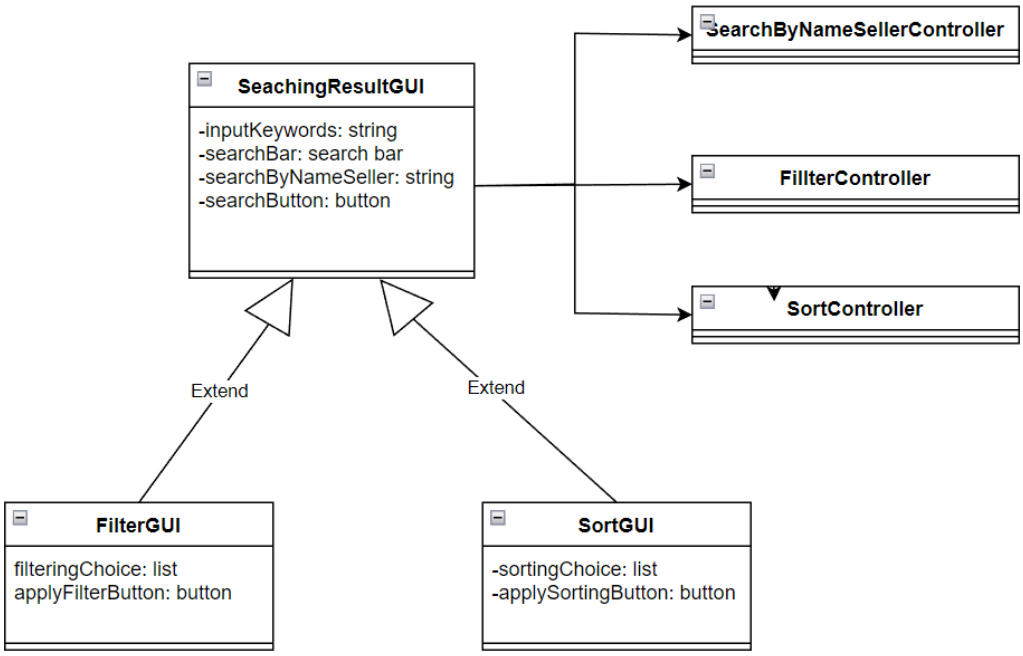
The user can see the post and the post GUI will contain content, images, like number, and the posting account. The user can interact with the post by like the post or open the textbox to comment on the post. All of the interactions and comments would be put below the post content.

The seller can choose to post a new post, they will be shown the form to enter the information about the product, and can add an image on their post. Then the input data is sent to controller to handle the posting process, and three controllers PostController, ShowPostController, ShowPostController will process the data of the post



4.8. Component: Searching Result GUI

The user enters the keywords into the search bar. The client receives the data from the server, in their page, there will be the list of results which can be applied to be filtered or sorted. All of the the presentation and process will be handled by the SearchingController

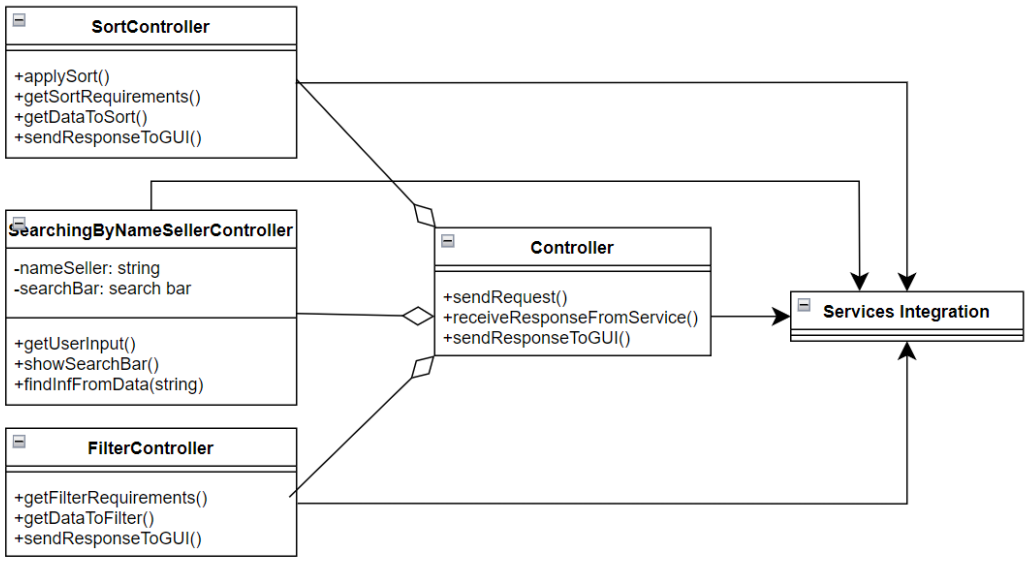


4.9. Component: Searching Controller

When the controller receives request from the GUI, it will analyze what user wants to search by name seller(class SearchingByNameSellerController), to sort(SortController), or to filter(FilterController) by the input data from the search bar and filter/sort tools.

In the SearchingByNameSellerController class, it will get the name seller string in the search bar and send it to the private variable of this class. After finishing accessing the database to get the data, it will call the function to display the results to the user.

In the FilterController and SortController classes, it will access the input data in the search bar and chosen criteria in the filter/sorts tools. Then, the data will be sent to Service Integrations to handle the process. After finishing accessing the database to get the data, the results will be updated and the newly-found results will be displayed to the user

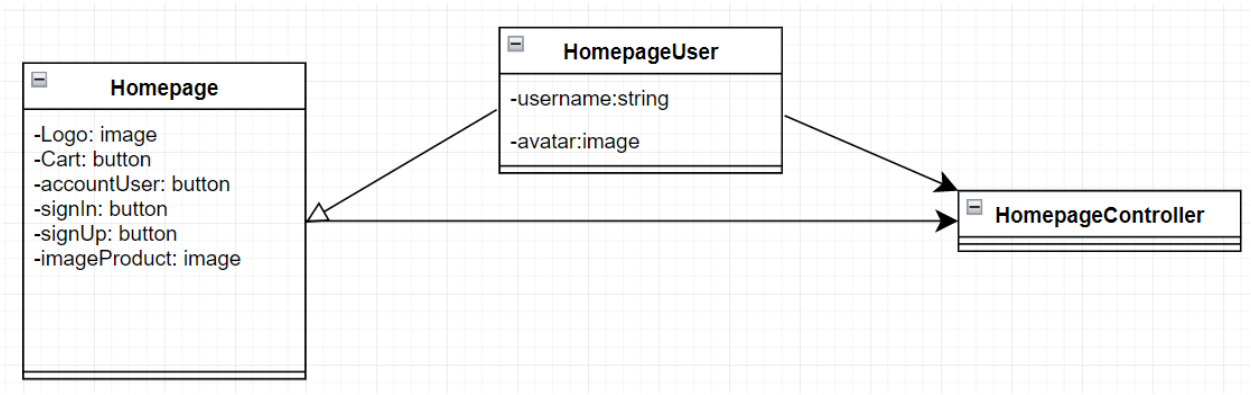


4.10. Component: Homepage GUI

After successful sign in, the user will be taken to the home page. The home page interface includes: Logo website, Cart, My Account, button sign in,sign up, Image Product,...

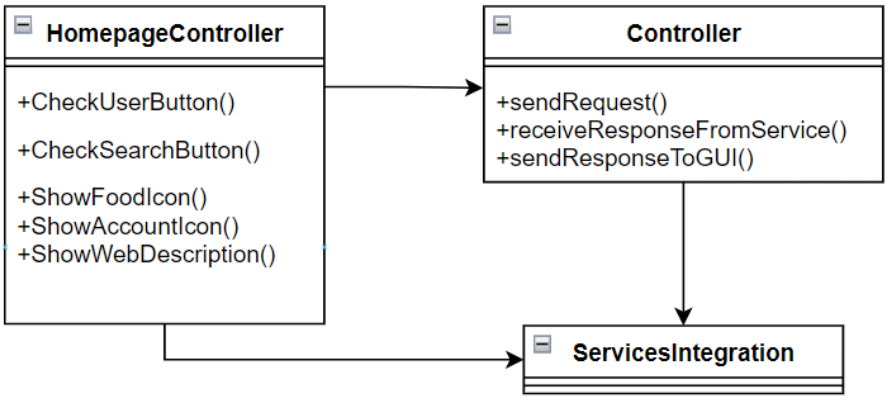
When you click on my account, the user's homepage will appear, there will be a username, avatar…

Homepage will show Food Icon for user to choose the one want to order



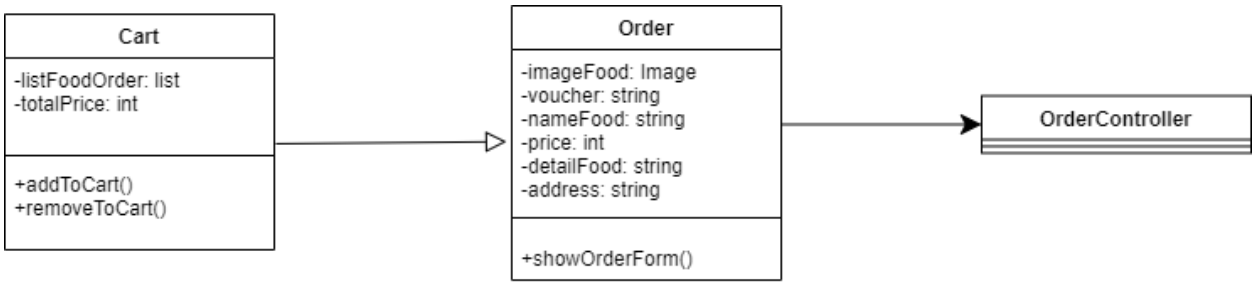
4.11. Component: Homepage Controller

Responsible for sending and receiving user requests. At the Homepage, users can request to see their user information, food type,web description.



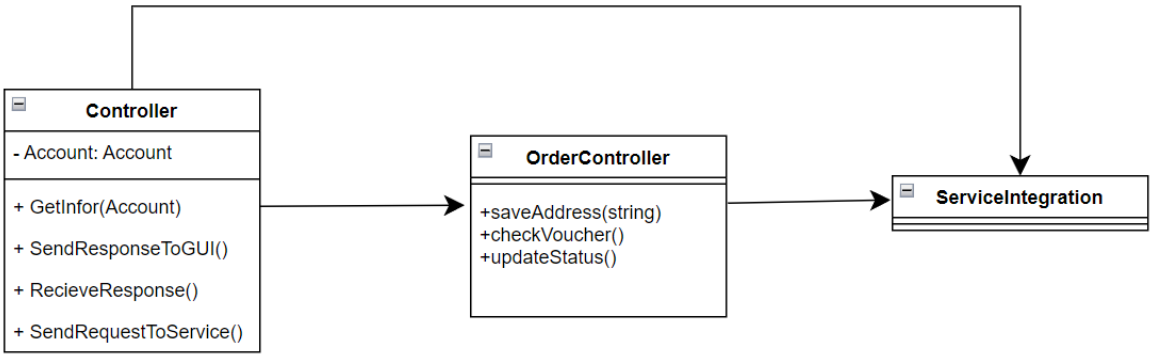
4.12. Component: Order GUI

The User can order by adding items to cart, the shopping cart will store the food items added by the user, user can also delete items in cart.Once the user has selected the order, the user will now be redirected to the order form including: Selected food, Voucher, price, address of buyer…All of the the presentation and process will be handled by the OrderController



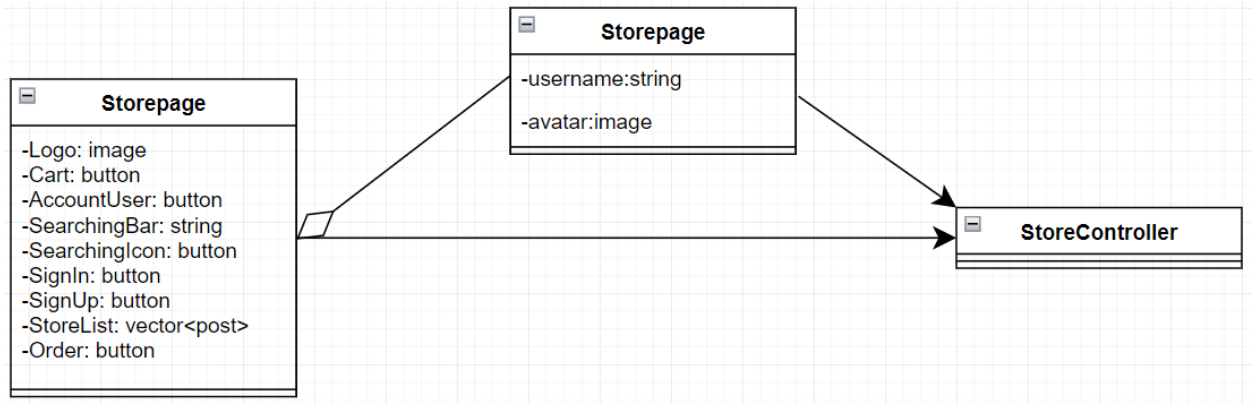
4.13. Component: Order Controller

In the Order Controller, the controller will receive a response from the GUI, it will get a list product, voucher from Database, and check stock or not, then update address, order status to Database . Service Integration will send the response for the order is successful or not, then the Controller will send that response to the GUI.



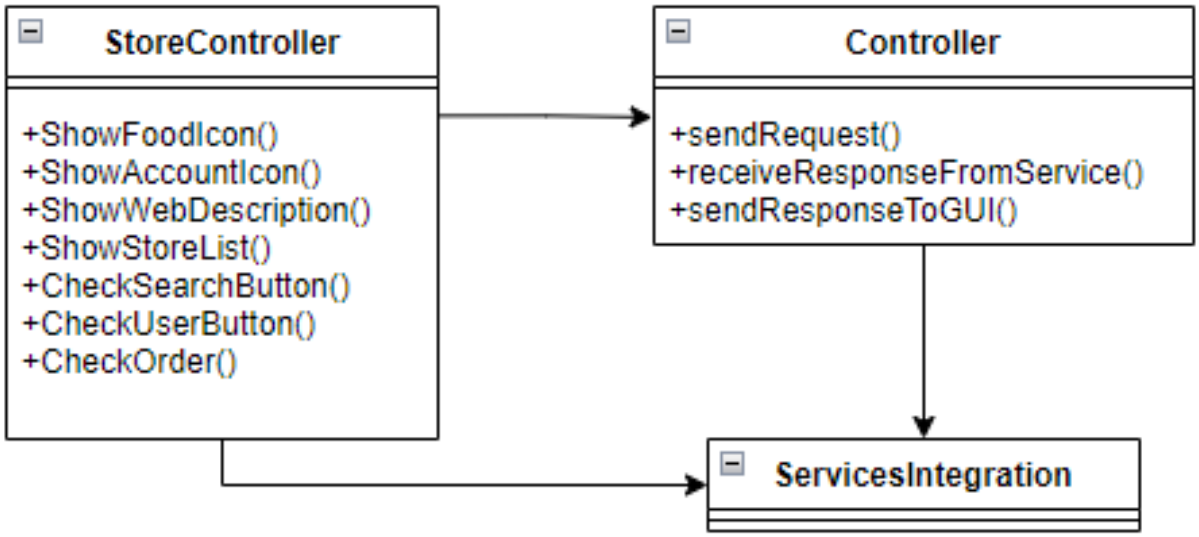
4.14. Component: Store page GUI

After user select the food want to order in homepage page, user will be taken to Store page
In the store page, users can search food, search buttons, add to cart buttons.
All user posts will also be displayed in the Storepage. All of the presentation and process will be handled by the Store page Controller.

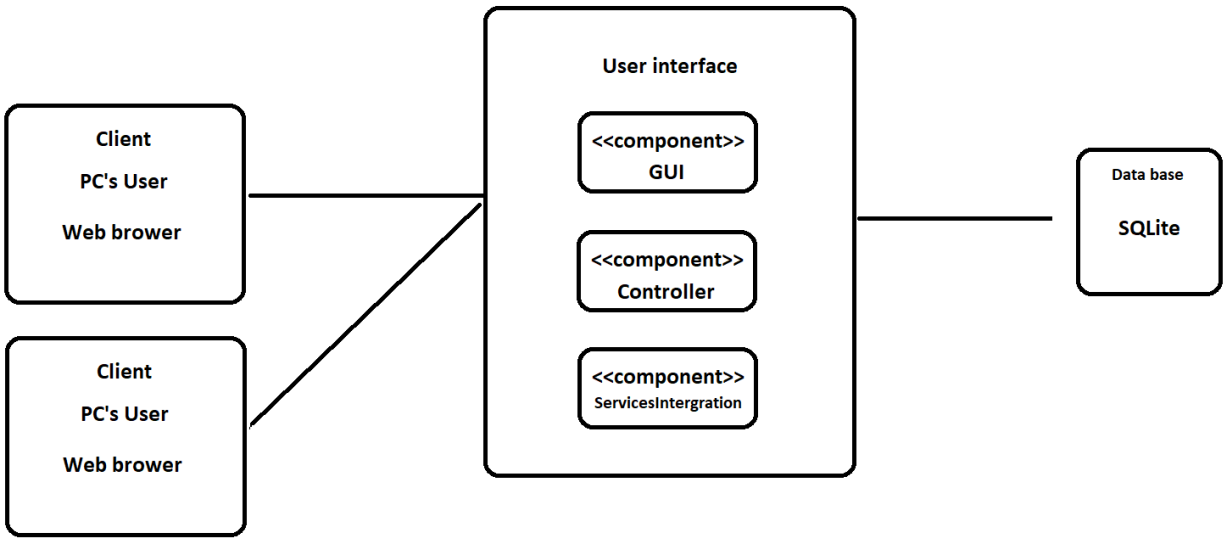


4.15. Component: Store page Controller

Responsible for sending and receiving user requests. At the Store page, users can request to see their user information, store ,web description, order food. All of the presentation and process will be handled by the Services Integration



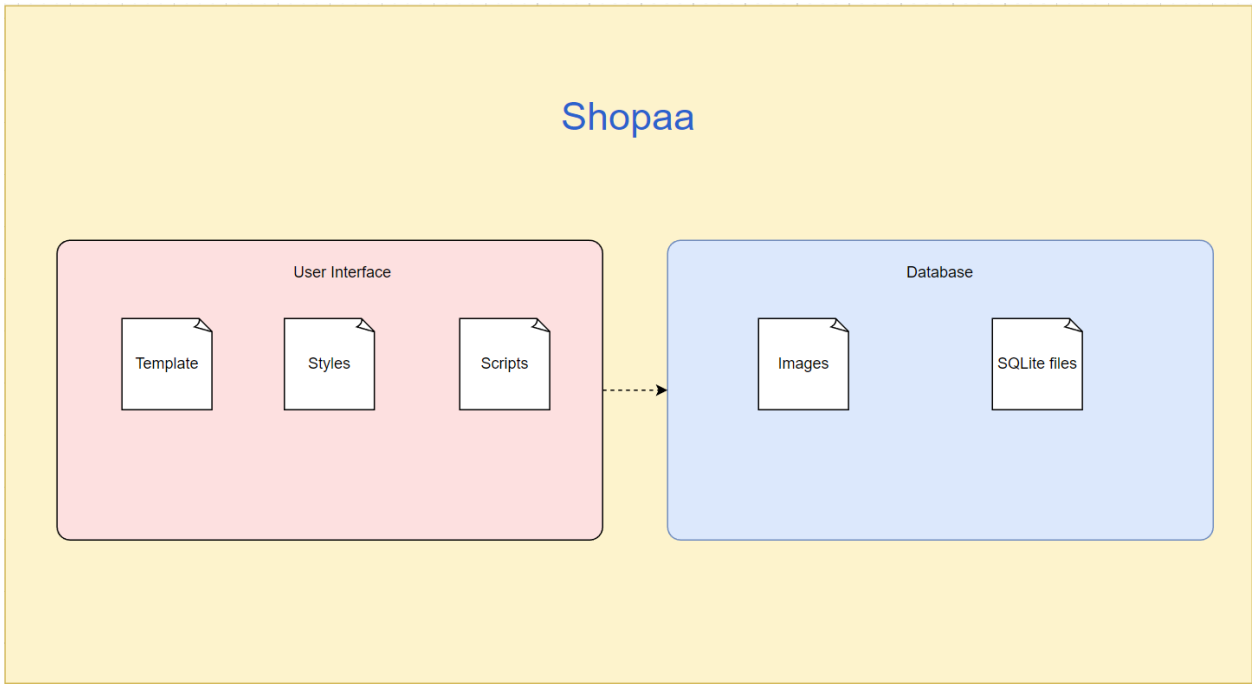
5. Deployment



- Users access our website using a computer or laptop and a web browser.
- Users can interact with our website through its user interface, which is only available on local hosts.
- The website will load data and display it on the screen

6. Implementation View

Using the implementation view, we will create a folder structure that will be utilized in this project, similar to the package diagram shown below.



User interface:

- Template: Save HTML files
- Scripts: Save external Javascript files
- Style: Save external CSS file

Database:

- Image: Save image
- Contain Sqlites files of data

There are folders called Templates, Styles, and Scripts on the client side. There are numerous HTML files in the Templates folder that describe the webpages; in addition, the Styles and Scripts folders contain external CSS and JavaScript files that enhance and add sophisticated functionality to these pages.

The database sqlite files will store and load the data to the browser (The device which runs the database must have Node.js, sqlite3 Node.js module, SQlite database file).