VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Software Engineering (CO3001)

Assignment

"Restaurant POS 2.0"

Lecturer: Assoc. Prof. Quan Thanh Tho.

TA: Dr. Duc Anh.
Students: Nguyen Vo Hoang Thi - 1952996.
Tran Hung Cuong - 1952606.
Nguyen Thanh Loc - 1952330.
Nguyen Truc Phuong - 1952402.
Le Tu Ngoc Minh - 1952844.

HO CHI MINH CITY, 2021

Contents

1 Requirement elicitation

1.1 Identify the context of this project.

1.1.1 Who are relevant stakeholders?

Stakeholder	Role
Customers	whose information and food ordering are recorded in the system.
Restaurant Owner(s)	responsible for the growth and development of the restaurant.
Clerks	control the bills carefully, check if the food is similar to what customers order.
Delivery	deliver food to the customers in time.
Managers	is to streamline operations across concepts, customize the management, inventory tracking, smart scheduling and manage the customer relationship.
IT Staff	takes the responsibility to install and maintain the system.
Chefs	prepare the food ordered from the system by customers.
Customer Care Staffs	record feedback and provide support to the customers.

1.1.2 What are expected to be done?

POS stands for Point Of sales, which is a very useful and important system to restaurants and retailed industry. Therefore, there are many advantageous features involved in this system:

- Online payment and order processing.
- Customer relationship and experience.
- Providing sales metrics and reporting.
- Inventory stock and management.
- Employee management.

1.1.3 What are the scope of the project?

- The system provides many information about the restaurant online such as menu, pictures, operating time, etc.
- Customers can order food online, which means they just need a phone or a laptop to order food without going to the restaurant.
- Online Payment reduces the interaction between people, especially for this COVID-19 period.
- Recording all feedback and provide more information for the customers.
- The manager can control the bills, payment and work flow through this system.

- Chefs can easily know what food ordered.
- Delivery staffs are provided locations and time to deliver food in time.
- Managing inventory across all locations, both online and offline.
- Managing customer data effectively.
- Easily receive feedback from customers and improve in store sales.
- Adapting to business needs with customization.

1.2 Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system.

1.2.1 Functional Requirement

- **Ordering System**

  1. Create account.
  2. Customers can manage their accounts (name, avatar, ...).
  3. Log in to the system.
  4. Verify account by sending code every six months.
  5. Skim the digital restaurant's menu.
  6. Select item from the menu, also the amount of item.
  7. Customize the option for the item.
  8. Note bar for customers to note something about their ordered items.
  9. Review the order list.
  10. Remove unwanted items.
  11. Place an order.
  12. Calculate and display the payment.
  13. Delivery options.
  14. Send confirmation message of successful order via phone message.
  15. Specific QR code for each payment.
  16. Send the order number of the customer's order.

  If customers choose delivery option:

  1. Send notification when their order is in processing and finished.
  2. Send notification when the shipper starts to go and has arrived.
  3. Include shipping fee into the payment.

- **Reservation Management:**

  1. Store the reservation information to the reservation database.
  2. Send notification whether the reservation is successful or not.

- **Menu Management:**

1. Add a new/update/delete vendor to/from the menu.
2. Provide a filter option for the food category.
3. Update the price of an item (if any).
4. Provide recommendations based on the current order.
5. Update additional information on the item (photo, description, ...).

- **Restore order:**

  1. Restore the order to the database.
  2. Record the transaction.
  3. Add processing/finished condition of an order.
  4. Restore the clerk and chef who is responsible for the order.
  5. Display the order in a suitable way.

- **Customer Service:**

  1. Restore the data of the customer.
  2. Send a thank you letter and ask for feedback after customers using the service.
  3. Provide Help Center to improve the service from the ratings and reviews from the customer, also provide solutions to the customer's problem.

1.2.2 Non-Functional Requirement

1.2.2.a OPERATION Requirement

Describe how well the system performs the tasks.

1. **Access Security**
   The extent to which the system is safeguarded against deliberate and intrusive faults from internal and external sources.

   - [Forgotten password] Clients may request to change password. An Identity verification link will be sent to their primary email address or phone message which they used to sign up.
   - [Initial Login] Users must change the initially assigned login authentication information (password) immediately after the first successful login. The initial password may never be reused.
   - Clients shall not be allowed to access other clients' information and order, and any such attempt shall be reported to the security administrator.
   - Each unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail. After 5 unsuccessful attempts to access, there shall be a notification to that customer's email.
   - [Notification] Customers shall receive notification of every successful transaction and modification in profile information via preferred communication method of record.

2. **Accessibility**
   The extent to which the software system can be used by people with the widest range of capabilities to achieve a specified goal in a specified context of use.

- The customers' receiving address must be at most 10km far from the restaurant's location.
- The system shall be accessible by people with specific vision needs, to the extent that a user shall be able to:
    - Display the whole user interface in a large font without truncating displayed text or other values.
    - Use a screen magnifier to magnify a selected part of the screen.

3. **Availability**
   The degree to which users can depend on the system to be up (able to function) during "normal operating times".
    - The system shall be available for use between the hours of 6:00 a.m. and 11:00 p.m.
    - Unless the system is non-operational, the system shall present a user with notification informing them that the system is unavailable.

4. **Confidentiality**
   The degree to which the software system protects sensitive data and allows only authorized access to the data.
    - [No sensitive cardholder retention] The RQ Website will not retain customer credit or debit card information entered during the Checkout payment processing.

5. **Efficiency**
   The extent to which the software system handles capacity, throughput, and response time.
    - Complete report summaries of the current business day's trading shall be available one minute after the end-of-day close of trading.
    - System shall be able to process every operation in 3 second or less.
    - The system must accommodate 300 simultaneous users or less within the peak load period.
    - Maximum simultaneous user capacity loading at non-peak periods will be 300.
    - Any interface between a user and the automated system shall have a maximum response time of 2 seconds.

6. **Integrity**
   The degree to which the data maintained by the software system are accurate, authentic, and without corruption.
    - All monetary amounts must be accurate to integers.
    - Order history will be saved when the customer leaves the system.

7. **Reliability**
   The extent to which the software system consistently performs the specified functions without failure.
    - The probability of the website crashing when a customer logs in is 1/10000.

8. **Safety**
   The degree to which a software system prevents harm to people or damage to the environment in the intended context.

9. **Survivability**
The extent to which the software system continues to function and recovers in the presence of a system failure.

10. **Usability**
The ease with which the user is able to learn, operate, prepare inputs, and interpret outputs through interaction with a system.

    - The new product shall be easy to use by adult members (age 18 to 80) of the public who may only have one hand free without training.
    - The product shall be self-explanatory and intuitive.
    - Customers can easily send queries to clerks via the online chat application available at the website, or giving feedback on usage, adjusting confirmed orders, and solving arising problems.

1.2.2.b REVISION Requirement

- How efficiently the software system can be corrected or fixed when errors occur.
- How easily new features can be added?

1. **Flexibility**
The ease with which the software can be modified to adapt to different environments, configurations, and user expectations.

    - Provisions shall be made for the future usage of multiple languages:
        - The structure of the data store shall be such that multi-lingual support shall not necessitate additional components or the need to replace current components.
        - A user shall be able to nominate their preferred language when entering their personal information.
    - The billing system shall be able to process invoices and payments in multiple different currencies.
    - The system shall be suitable for use by any restaurants in any country.

2. **Maintainability**
The ease with which faults in a software system can be found and fixed.

3. **Modifiability**
The degree to which changes to a software system can be developed and deployed efficiently and cost effectively.

    - No piece of text that might be displayed to a user shall reside in program source code. Every piece of text that a user might see must be modifiable without changing source code.

4. **Scalability**
The degree to which the system is able to expand its processing capabilities upward and outward to support business growth.

    - The system shall be scalable to support unlimited growth in the number of orders due to the different size of restaurants.

- The average order per day is 300.

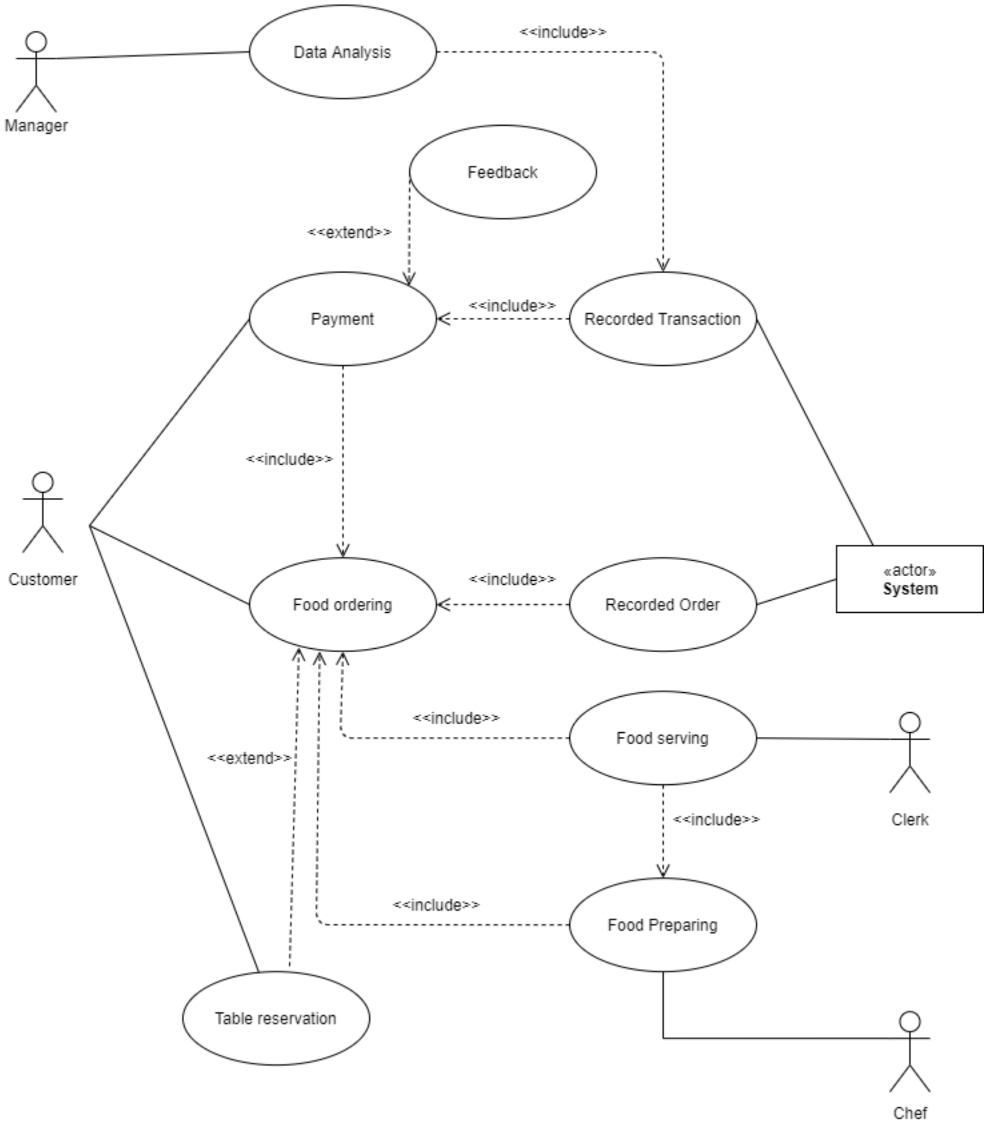
5. **Verifiability**
The extent to which tests, analysis, and demonstrations are needed to prove that the system will function as intended.

1.2.2.c TRANSITION Requirement

Transition requirements describe the ability of the software system to adapt to its surrounding environment.

1. **Installability**
The ease with which a software system can be installed, uninstalled, or reinstalled into a target environment.
    - Using Web-based technology and QR Code instead of in-app so it's easy to get access.
2. **Interoperability**
The extent to which the software system is able to couple or facilitate the interface with other systems.
    - The common language used in the incoming mail department shall be English to increase communication effectiveness and reduce processing errors.
3. **Portability**
The ease with which a software system can be transferred from its current hardware or software environment to another.
    - Usable in various types of end systems.
4. **Reusability**
The extent to which a portion of the software system can be converted for use in another system.
    - Customers are allowed to reorder the previous menu.

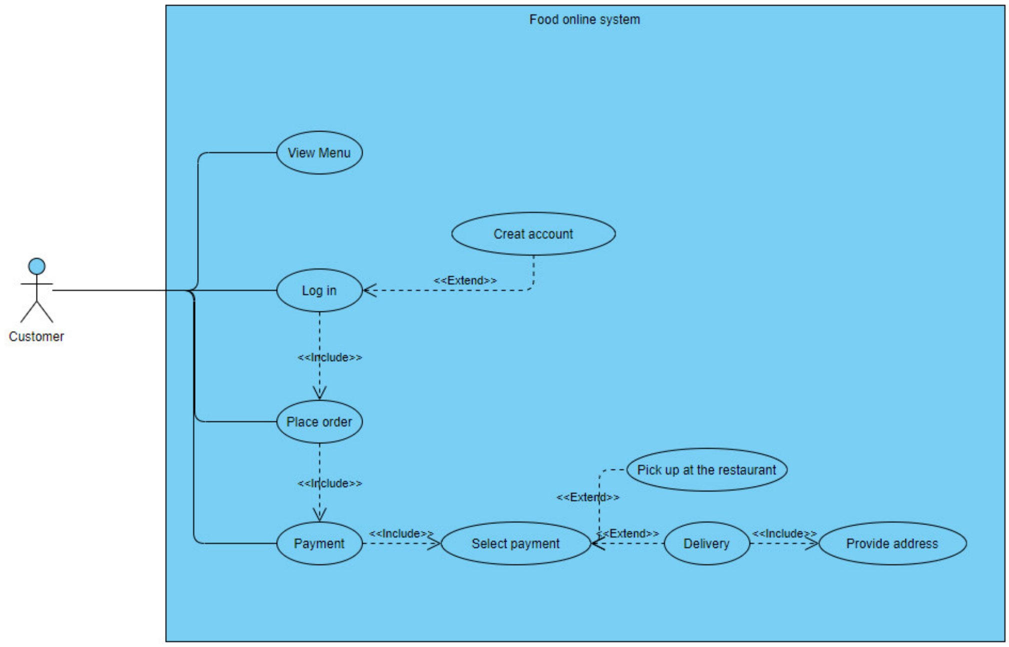
1.2.3 Use-case Diagram



1.3 Choose one specific feature, i.e. food ordering, table reservation, customer management. Draw its use-case diagram and describe the use-case using a table format

Feature: FOOD ORDERING

1.3.1 Use-case Diagram



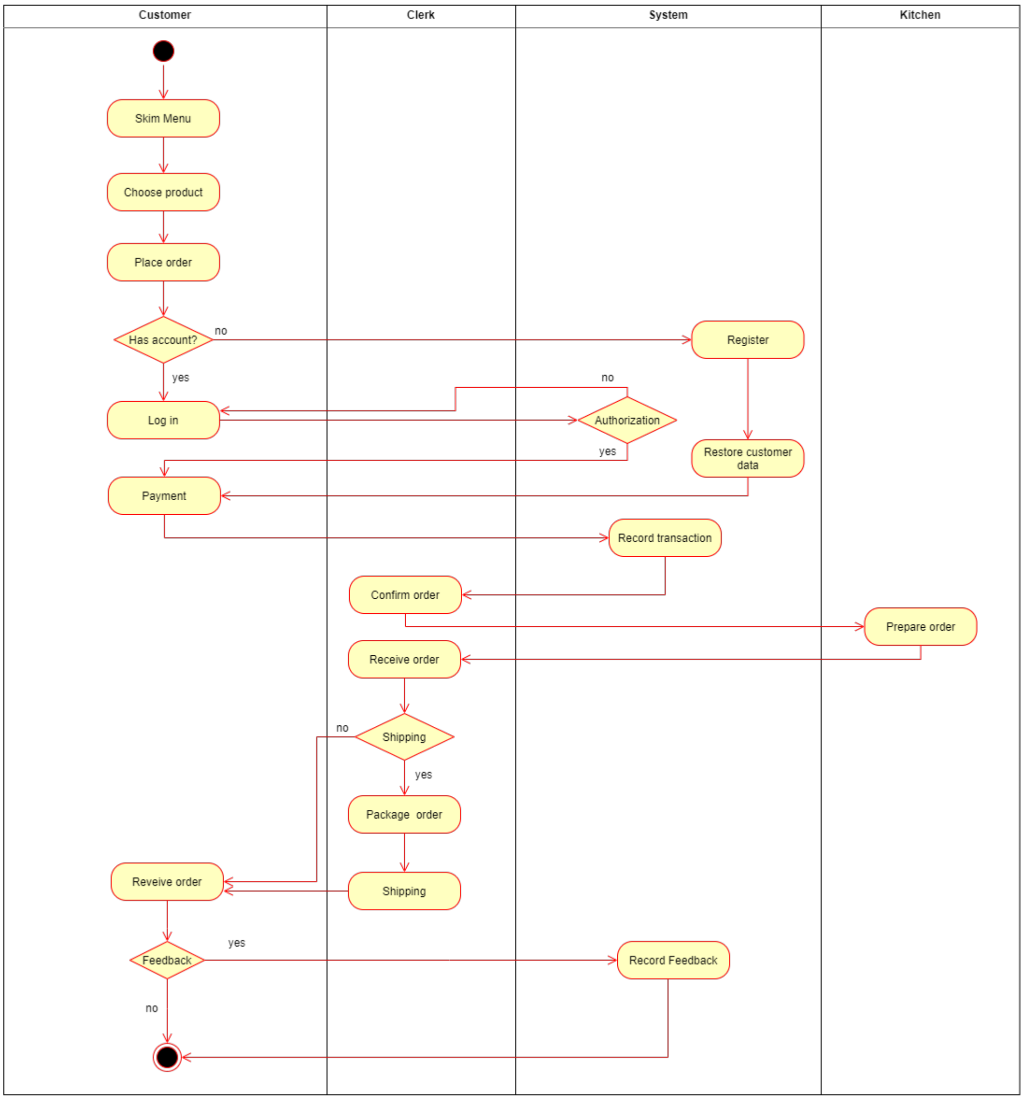
1.3.2 Table Format

Use-case name	Food ordering
Actor	food ordering system, customer
Description	Order food and pay online through website
Pre-condition	- Have an account on system - The device must have an internet connection
Post-condition	Order successfully and customers get the bills online after payment through online payment service
Basic path	- The system receives data about user information. - Customers order food, select payment and delivery. - Customers provide address - Customer service track and confirm the orders. - Chef cook and pack. - The system receives information about the food which the user has selected and sends it to the calculation system.

	- The system receives money from online payment services and prints invoices for users.
Alternative path	At the pre-condition if customers don't have account they can register through system.
Exceptional path	If there is no Internet connection, the system will notify users know that transferring data to the cloud has failed

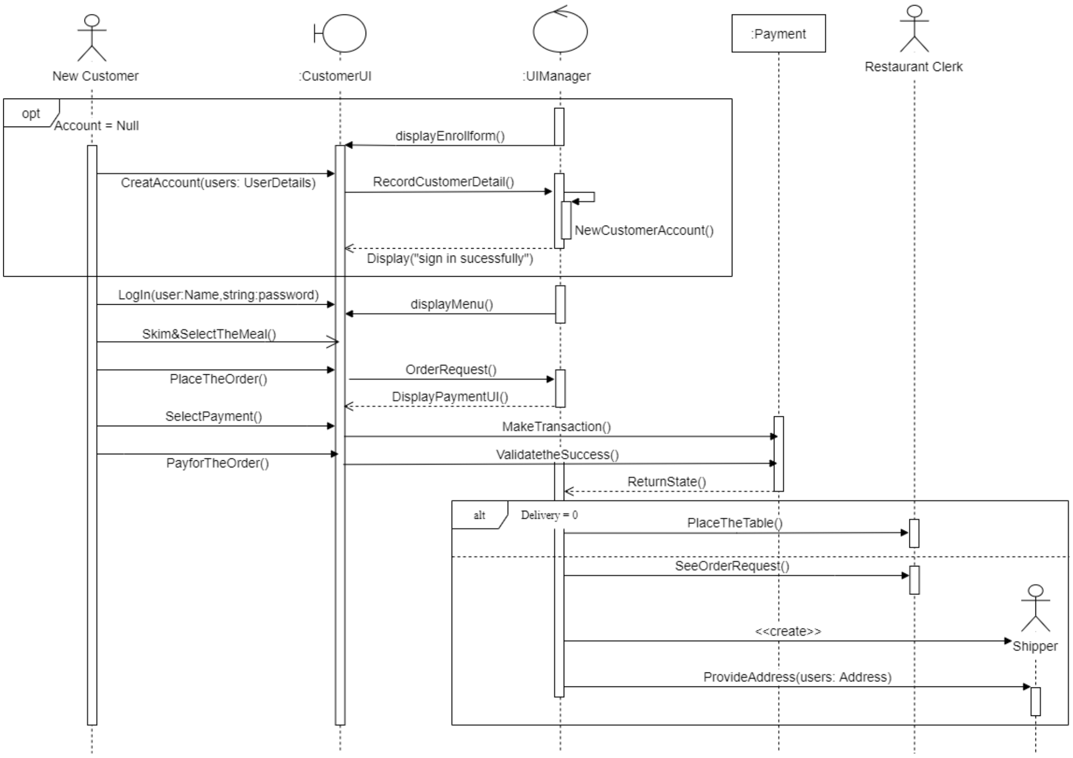
2 System Modelling

2.1 Draw a activity diagram

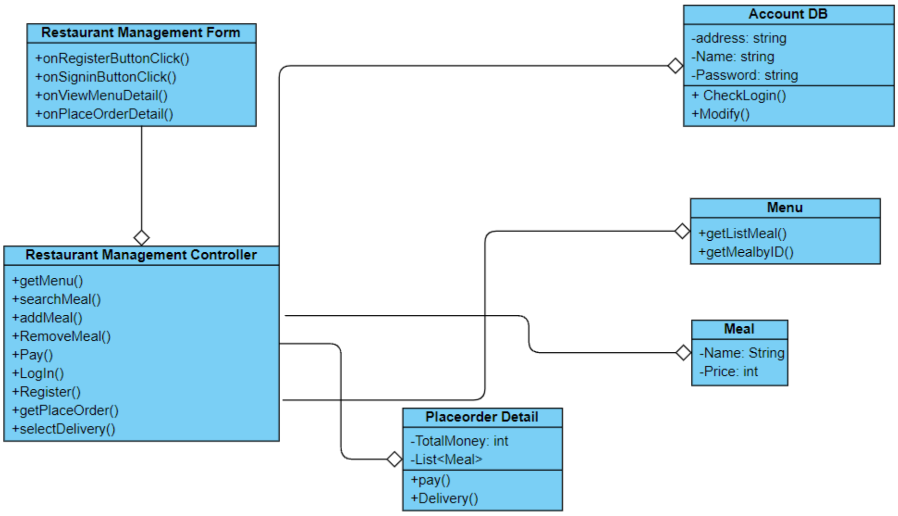


2.2 Draw a sequence diagram

Sequence Diagram of a Customer Management System



2.3 Draw a class diagram



3 Architecture design

3.1 Describe an architectural approach you will use to implement the desired system

In general, what is a client server architecture?

- A centralized network architecture that classifies computer into two sections, client and server.
- A client is the requester, which can be a program that we use to make requests through the network with parameters included.
- A server is the response provider, which is a program that listens for the client's requests and responds to them.
- The server component provides a function or service to one or many clients, which initiate requests for such services.
- Server itself might be a client. For example, the server could request something from a database server, which in this case, would make the server a client of the database server.
- Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

Impose specific topological constraints? (diagram)

- Two levels, typically many clients with one server.
- From client to server and server to client.
- Constraints:
  - Clients cannot communicate directly with each other. If needed, the server acts as a message relay for the clients to communicate.
  - Only clients can initiates communication.
- All workloads are done at the server side.

Applicable to specific problems of this project

- Data can be centralized and easy to do for collaboration.
- All clients are requesting the same type of data.
- Clients can give specific information to request different data dynamically.
- Less computational burden on the client side, which make client more lightweight.
- Clients are unable to do the heavy computation and the computation are done on the server side.
- Provide better data integrity and backup system, thus higher reliability.
- In general, people can access the data at anytime as long as they have network and authorization.

- When mobility is needed, application and data can be easily moved and replicated.

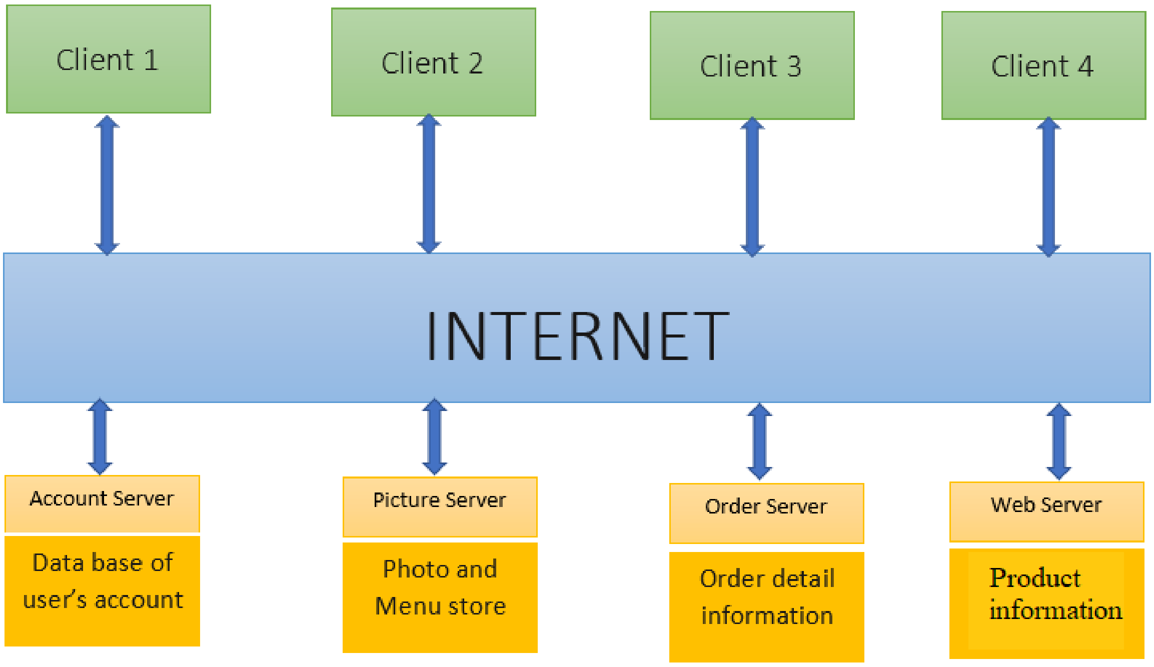
Advantage of this architecture when applying to this system

- Centralization of control:
  - A dedicated server controls the access of resources and integrity of the data so that a program or unauthorized client cannot damage the system easily.
  - Changes only need to be done on the server and the clients will be able to receive.
  - Network processing is done centrally, not at individual computers, which reduce the burden of the OS.
- Scalability:
  - You can increase the capacity of clients and servers separately. Any element can be increased or enhanced at any time, or you can add new nodes to the network.
  - You can add resources in the form of network segments, computers and servers to a client server network without major interruptions to the network.
  - Update task for data or other resources more efficient and easier to managed.
- Easy maintenance:
  - Since backup, security and antivirus are centralized, it is easier to setup and troubleshoot, where everything takes place at one physical server.
  - Fewer support staff are needed to manage centralized security accounts than would be needed if security and resource access had to be configured on each individual computer on the network.

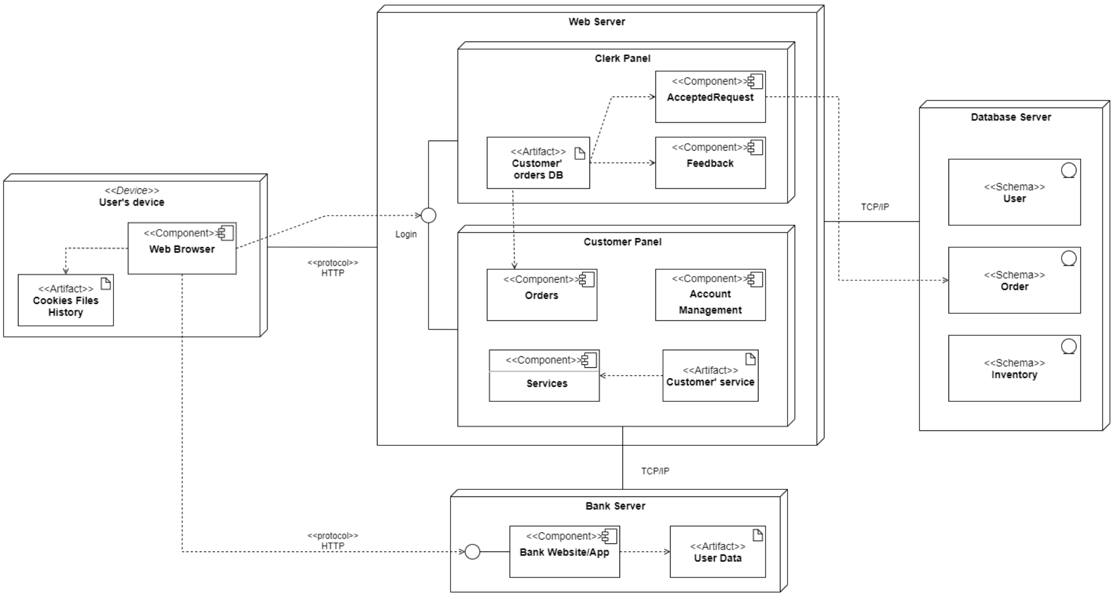
Disadvantage

- Single point of failure: Since there's a reliance on the central server, if it fails, client requests cannot be done.
- Traffic congestion: Happens when a large number of simultaneous clients send requests to the same server. This might cause the server to slow down or even shut down.
- Cost: The cost of server hardware and software is much greater than the cost of buying desktop hardware and software licences. Thus it is expensive to scale or even hard to scale.

Practical case in software project: There are 2 customers ordering food and the waiter takes their orders separately and brings them back the food they ordered. This example demonstrates how the server can be connected to more than one client. Furthermore, it also shows that for a communication to happen between a client and a server, the client have to be the one to first initiates the request after which the server will reply with a response. Here is out architecture for our project:



3.2 Draw an implementation diagram for Major (not all) functional requirements


Web Server
Clerk Panel
<<Component>>
AcceptedRequest
<<Artifact>>
Customer' orders DB
<<Component>>
Feedback
<<Device>>
User's device
<<Component>>
Web Browser
<<Artifact>>
Cookies Files History
Login
<<protocol>>
HTTP
Customer Panel
<<Component>>
Orders
<<Component>>
Account Management
<<Component>>
Services
<<Artifact>>
Customer' service
TCP/IP
Database Server
<<Schema>>
User
<<Schema>>
Order
<<Schema>>
Inventory
TCP/IP
Bank Server
<<protocol>>
HTTP
<<Component>>
Bank Website/App
<<Artifact>>
User Data