

HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY FOR HIGH QUALITY TRAINING



SPECIALIZED ESSAY

ANALYSIS, DESIGN, CONSTRUCTION
OF RESTAURANT SUPPLY CHAIN
MANAGEMENT SYSTEM

Mentor: M.S. Nguyễn Minh Đạo

Student Name: Student ID

Ngô Công An 16110002

Đào Xuân Thủy 16110544

Ho Chi Minh, December 9th 2019

COMMENTARY OF MENTOR

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ho Chi Minh, December 2019

M.S. Nguyễn Minh Đạo

COMMENTARY OF REVIEWER

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ho Chi Minh, December 2019

D.Sc. Lê Văn Vinh

OUTLINE OF SPECIALIZED ESSAY

Name of student: Ngô Công An

Student ID: 16110002

Name of student: Đào Xuân Thủy

Student ID: 16110544

Major: Software Engineer.

Topic name: Analysis, design, construction of restaurant supply chain management system.

Mentor: M.S. Nguyễn Minh Đạo.

Content:

Theory:

- Research about microservices architecture.
- Research about authentication & authorization in microservices.
- Research technologies for an e-commerce system.

Practice: Apply to construction of restaurant supply chain management system.

Execution time from 09/09/2019 to 08/12/2019

Signature of student:

Signature of student:

Ho Chi Minh, December 9th 2019

HEAD OF INFORMATION TECHNOLOGY

MENTOR

(Name and signature)

(Name and signature)

ASSURANCE

We assure that this project is our own implementation. We do not copy, use any material or source code of others without specifying the source. We assume responsibility for violations.

Ho Chi Minh, December 9th 2019

Đào Xuân Thủy

Ngô Công An

MANY THANKS

In this fact, there are no successes that are not associated with support or assistance, whether more or less, directly or indirectly by others. Now I would like to send this sincere thanks to M.S. Nguyễn Minh Đạo, who has supported and transmitted motivation to us in the process of choosing topics, instructions and comments. Although we do not ask much, but when asking the teacher for help, it is the motivation for us to complete the course, without the instructions and practical experiences of the teacher, I think the article This report will be difficult to complete and completed on time. Once again, I would like to thank the teacher.

We would like to send our thanks to our fellow students who provided useful information and knowledge to help us improve our topic. We would also like to thank M.S. Nguyễn Minh Đạo, who directly guided us to complete this project. We will not be able to complete without the guidance of the master. Due to lack of experience in designing and building software that applies this relatively new process, it is possible that during construction process, there are many errors and problems, so please understand.

Due to limited time, with limited knowledge and many other worries, there is an inevitable problem, so I look forward to receiving your valuable suggestions for my knowledge. more complete later. We sincerely thank you.

Ho Chi Minh, December 9th 2019

Đào Xuân Thủy

Ngô Công An

TABLE OF CONTENT

| | |
|--|----|
| LIST OF TABLES | 4 |
| LIST OF FIGURES | 5 |
| CHAPTER 1. OBJECTIVE OF PROJECT | 7 |
| 1.1. Current Status Survey..... | 7 |
| 1.2. Current systems | 7 |
| 1.2.1. Ocha..... | 7 |
| 1.2.1.1. Introduction..... | 7 |
| 1.2.1.2. Ocha POS..... | 8 |
| 1.2.1.3. Ocha Boss | 9 |
| 1.2.1.4. Advantages..... | 10 |
| 1.2.1.5. Disadvantages | 10 |
| 1.2.2. Kiot Viet | 10 |
| 1.2.2.1. Introduction..... | 10 |
| 1.2.2.2. Website functionality | 11 |
| 1.2.2.3. Function of sales screen..... | 12 |
| 1.2.2.4. Functions of Sales App | 14 |
| 1.2.2.5. Functionality of App Manager..... | 14 |
| 1.2.3. Suno | 15 |
| 1.2.3.1. Introduction..... | 15 |
| 1.2.3.2. Function | 17 |
| 1.2.4. Compare Ocha, Kiotviet and Suno | 18 |
| 1.3. Proposing ideas for new solution | 18 |

| | |
|---|----|
| 1.3.1. Document Management System..... | 18 |
| 1.3.2. Store Management System | 19 |
| 1.3.3. Order System for each table | 20 |
| 1.3.4. Operation Management System | 20 |
| 1.3.5. Electronic Commerce Website | 21 |
| 1.3.6. Mobile Electronic Commerce..... | 22 |
| 1.3.7. Data Processing System | 23 |
| 1.3.8. Other systems | 24 |
| CHAPTER 2. ROAD MAP | 25 |
| CHAPTER 3. DESIGN..... | 26 |
| 3.1. Architechture | 26 |
| 3.2. Component | 26 |
| CHAPTER 4. AUTHENTICATION AND AUTHORIZATION | 27 |
| 4.1. Overview | 27 |
| 4.1.1. Monolithic architecture | 27 |
| 4.1.2. Microservices architecture..... | 28 |
| 4.1.2.1. JSON Web Token | 29 |
| 4.1.2.2. Opaque Token..... | 30 |
| 4.1.2.3. OAuth 2..... | 31 |
| 4.1.2.4. Authentication, authorization at gateway service | 32 |
| 4.1.2.5. Authentication, authorization at all services..... | 33 |
| 4.1.2.6. Access Control..... | 34 |
| 4.2. Solutions for project | 37 |
| 4.2.1. Json web token..... | 37 |

| | |
|--|----|
| 4.2.2. Authentication and Authorization achitechure..... | 38 |
| 4.2.3. Access control..... | 38 |
| CHAPTER 5. SYSTEM DETAILS..... | 39 |
| 5.1. Document Management System..... | 39 |
| 5.1.1. Overview | 39 |
| 5.1.2. Design..... | 39 |
| 5.1.2.1. Architechure | 39 |
| 5.1.2.2. Database..... | 40 |
| 5.1.2.3. Interface | 41 |
| 5.1.2.4. Use case diagram | 42 |
| 5.1.2.5. Sequence diagram..... | 42 |
| 5.1.3. Technologies..... | 45 |
| 5.2. Mobile Electronic Ecommerce..... | 45 |
| 5.3. Electronic Commerce Website..... | 46 |
| 5.4. Operation Management System | 46 |
| 5.5. Store Management System..... | 47 |
| CHAPTER 6. TECHNOLOGIES | 48 |
| CONCLUSIONS..... | 49 |
| 1. Results..... | 49 |
| 2. Advantages..... | 49 |
| 3. Disadvantages | 49 |
| 4. Development strategy | 49 |
| REFERENCES | 50 |

LIST OF TABLES

| | |
|--|----|
| Table 1.2.4.1. Table compare ocha, kiotviet, suno | 18 |
| Table 4.1.2.2.1. Compare Opaque Token vs Transparent Token | 31 |
| Table 4.1.2.6.1. Access Control List..... | 34 |
| Table 6.1. Technologies for project | 48 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1.2.1.2.1. Machine Ocha POS | 8 |
| Figure 1.2.1.2.2. Function Ocha Pos..... | 8 |
| Figure 1.2.1.3.1. Machine Ocha Boss..... | 9 |
| Figure 1.2.1.3.2. Function Ocha Boss..... | 9 |
| Figure 1.2.2.1.1. Machine KiotViet | 11 |
| Figure 1.2.2.2.1. Overview website functionality..... | 11 |
| Figure 1.2.2.2.2. Function KiotViet on website..... | 12 |
| Figure 1.2.2.3.1. Sales screen | 12 |
| Figure 1.2.2.3.2. Function KiotViet on sales screen..... | 13 |
| Figure 1.2.2.4.1. Function KiotViet on sell app | 14 |
| Figure 1.2.2.5.1. Function KiotViet on app management | 15 |
| Figure 1.2.3.1.1. Introduction suno..... | 16 |
| Figure 1.2.3.2.1. Function Suno application..... | 17 |
| Figure 2.1. Roadmap of project | 25 |
| Figure 3.1.1. Overview of architectural design for system..... | 26 |
| Figure 3.2.1. Overview of component design for system..... | 26 |
| Figure 4.1.1.1. Monolithic vs Microservices | 27 |
| Figure 4.1.2.1. Lifecycle of request in microservices..... | 29 |
| Figure 4.1.2.1.1. Json web token structure | 30 |
| Figure 4.1.2.4.1. Authentication & Authorization at Gateway API | 32 |
| Figure 4.1.2.4.2. Authentication & Authorization at Gateway API diagram | 33 |
| Figure 4.1.2.5.1. Authentication & Authorization at all services diagram..... | 34 |
| Figure 4.1.2.6.1. Role-Based Access Control..... | 35 |

| | |
|---|----|
| Figure 4.1.2.6.2. Policy-Based Access Control | 36 |
| Figure 5.1.2.1.1. Architechture of Document Management System | 39 |
| Figure 5.1.2.2.1. Entity relationship diagram for Document Management System | 40 |
| Figure 5.1.2.2.2. Database diagram for Document Management System | 40 |
| Figure 5.1.2.3.1. Main screen of Document management System | 41 |
| Figure 5.1.2.3.2. Login screen for Document Management System | 41 |
| Figure 5.1.2.3.3. Edit screen for Document Management System | 41 |
| Figure 5.1.2.4.1. Use case diagram for Document Management System..... | 42 |
| Figure 5.1.2.5.1. Sequence diagram for View Post Detail of Document Management System..... | 42 |
| Figure 5.1.2.5.2. Sequence diagram for Login of Document Management System | 43 |
| Figure 5.1.2.5.3. Sequence diagram for Create Post of Document Management System..... | 44 |
| Figure 5.2.1. Use case diagram for Mobile Electronic Commerce | 45 |
| Figure 5.3.1. Use case diagram for Electronic Commerce Website | 46 |
| Figure 5.4.1. Use case diagram for Operation Management System | 46 |
| Figure 5.5.1. Use case diagram for Store Management System..... | 47 |

CHAPTER 1. OBJECTIVE OF PROJECT

1.1. Current Status Survey

Milk tea, clothing, fast food, electronic items, ... All of these commodities can be easily found in shops and commercial websites. From single stores, to wide chain stores across Vietnam and around the world.

In that business, from the management of warehouse products, managing employees, managing the operation process of the stores, to the e-commerce of items, transporting goods, analyzing customer data, ... All of these jobs need dozens of papers, hundreds of professionals and thousands of services attached.

So how does an inexperienced business person easily manage all of that information without needing a lot of time, manpower, no need to bother with the services to be accompanied by when to manage and commercial product?

1.2. Current systems

1.2.1. Ocha

1.2.1.1. Introduction

Ocha is a member of Sea (formerly Garena), a leader in digital entertainment, e-commerce and digital financial services throughout Southeast Asia. Sea's mission is to improve the lives of consumers and small businesses with technology, Sea has been listed on NYSE under the SE code.^[2]

Ocha is the management platform for all small and medium businesses in the Food and Beverage industry, helping to lift and elevate businesses in the economy. Designed to help owners set up, manage and develop businesses easier and more effectively.^[2]

Ocha includes two applications: Ocha POS and Ocha Boss.

1.2.1.2. Ocha POS

Overview:

Ocha is the management system used in the shop to help create menu, order, charge and manage at the shop.



Figure 1.2.1.2.1. Machine Ocha POS

Function:

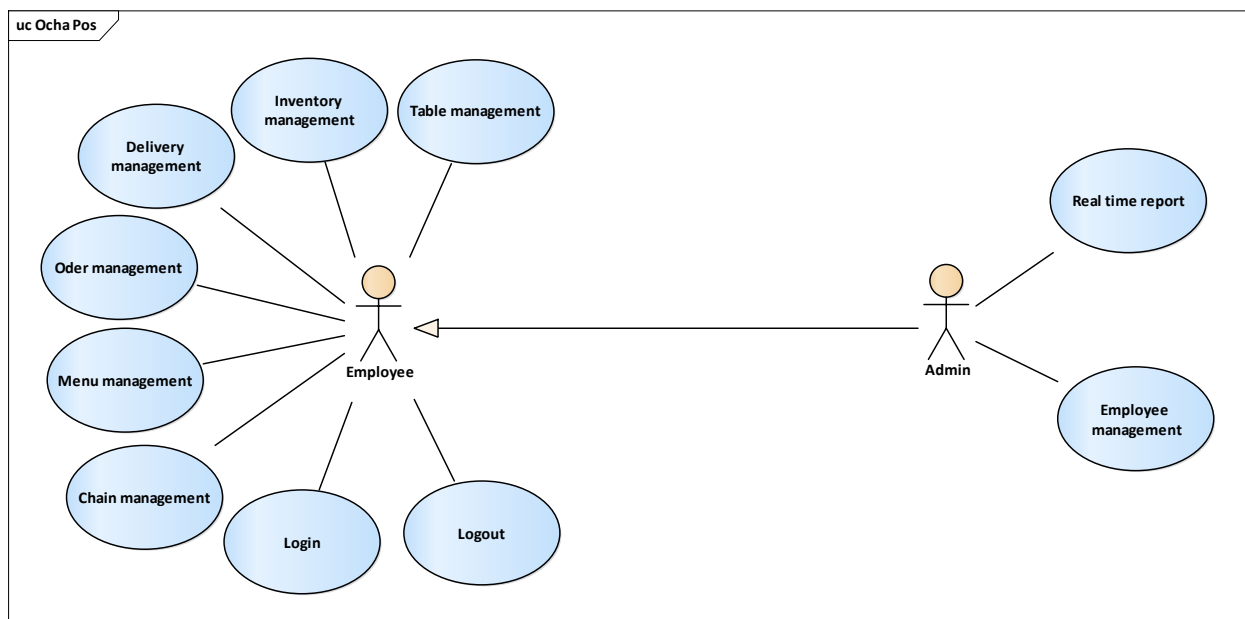


Figure 1.2.1.2.2. Function Ocha Pos

1.2.1.3. Ocha Boss

Overview:

Ocha Boss is a mobile application that helps owners manage remotely effectively and confidentially.



Figure 1.2.1.3.1. Machine Ocha Boss

Function:

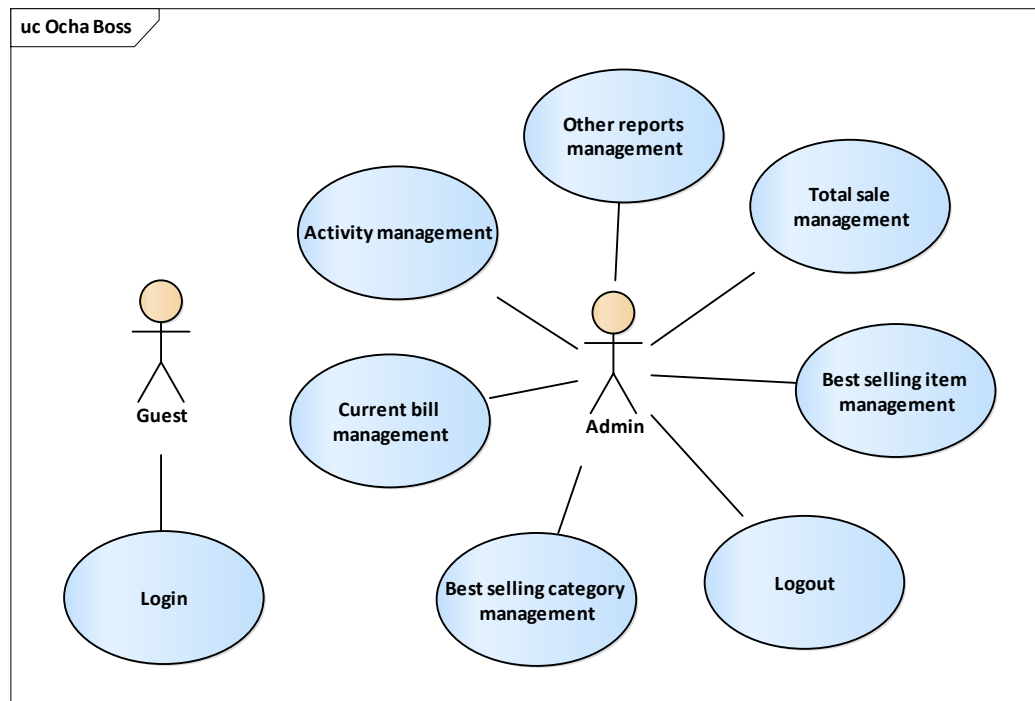


Figure 1.2.1.3.2. Function Ocha Boss

1.2.1.4. Advantages

- Simple interface, easy to use.
- The feature is applicable to many career such as: grocery, telephone, interior, diner, pub, ...
- Integrating many outstanding features such as revenue management, warehousing, reporting, ...

1.2.1.5. Disadvantages

- When damaged, need expert advice.
- Because the data is stored on the 3rd party server, the data loss, server crash or server communication line may still occur.

1.2.2. Kiot Viet

1.2.2.1. Introduction

KiotViet sales management software developed by Citigo Software Joint Stock Company. After many years of working in the field of software development for customers in Australia, France and the US market and working with many experts in the retail sector, we aspire to bring technology to the stores. Retail in Vietnam, helping you solve difficulties in the sales management process in a simple and easy way without spending too much.^[1]

KiotViet is a product of Vietnam Retail Association of AVR, recognized by the Vietnam Software and Information Technology Services Association (VINASA) in the field of software technology.^[1]

We have gone into the retail business market in Vietnam, understand the difficulties you face and strive to bring the best sales solution with KiotViet. We hope that KiotViet will become the most popular tool, supporting retail business activities easily and effectively.

The purpose is to provide technology solutions to help small, medium and micro enterprises do business more easily and effectively.



Figure 1.2.2.1.1. Machine KiotViet

1.2.2.2. Website functionality

Overview:



Figure 1.2.2.2.1. Overview website functionality

Help users manage information quickly and most generally about the operation of the stall from:

- The statistics of the business situation of the stall in the form of a chart.
- Introducing new features, connecting programs between Kiotviet and customers.

- Information about the activities of users on the booth.
- Display customers' birthday notifications on the store.

Function:

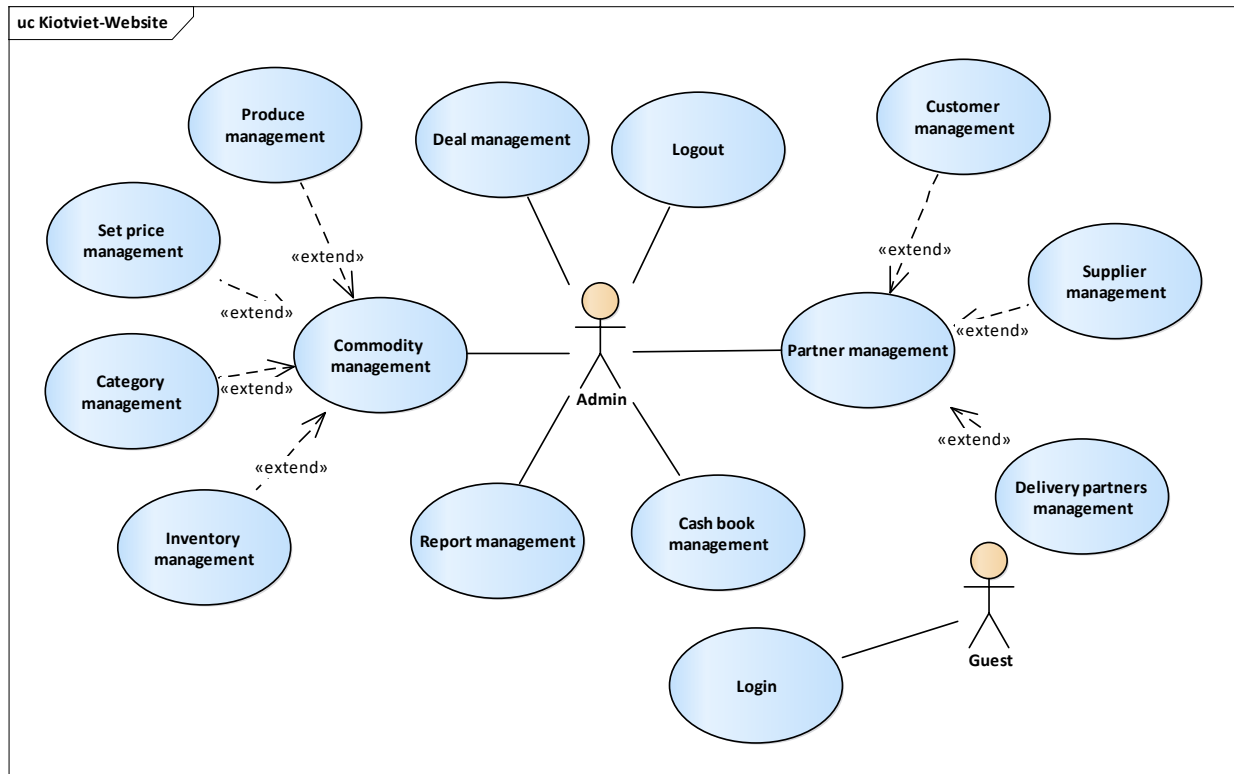


Figure 1.2.2.2. Function KiotViet on website

1.2.2.3. Function of sales screen

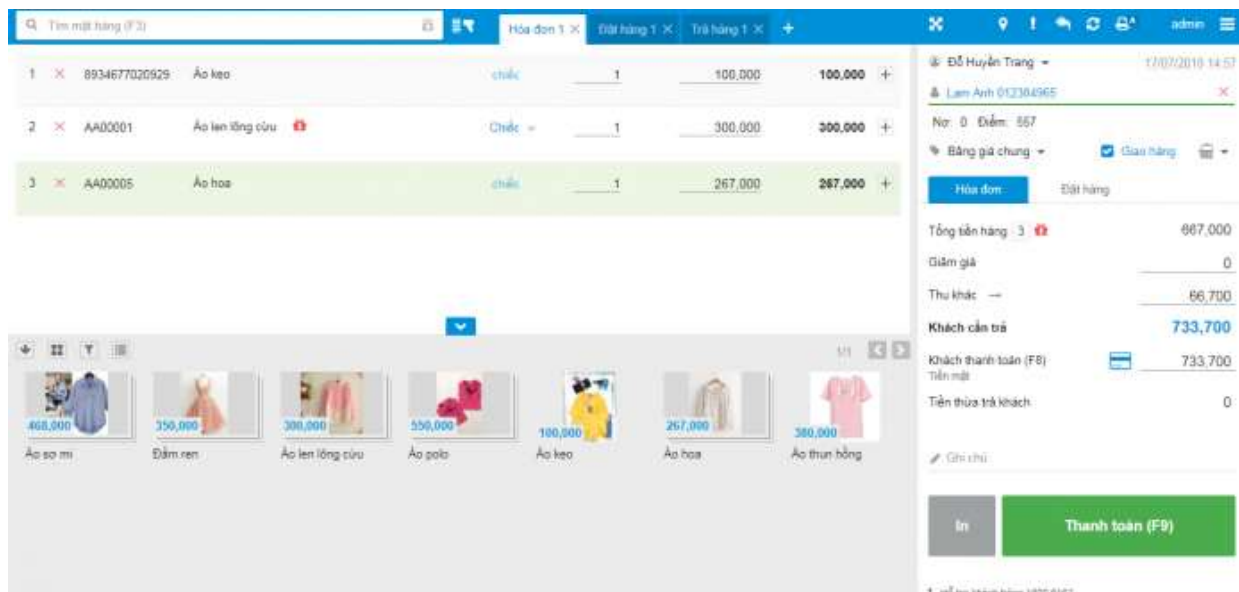


Figure 1.2.2.3.1. Sales screen

Sales screen is an interface to help users perform transactions with customers such as creating new invoices, placing orders and returning goods, setting up receipts and viewing end-of-day reports.

KiotViet's sales screen interface is designed according to the trend of flat, streamlined, coherent and professional design. Features are spread out over an interface to shorten the operations on the screen, giving users a convenient and fast experience.

Function:

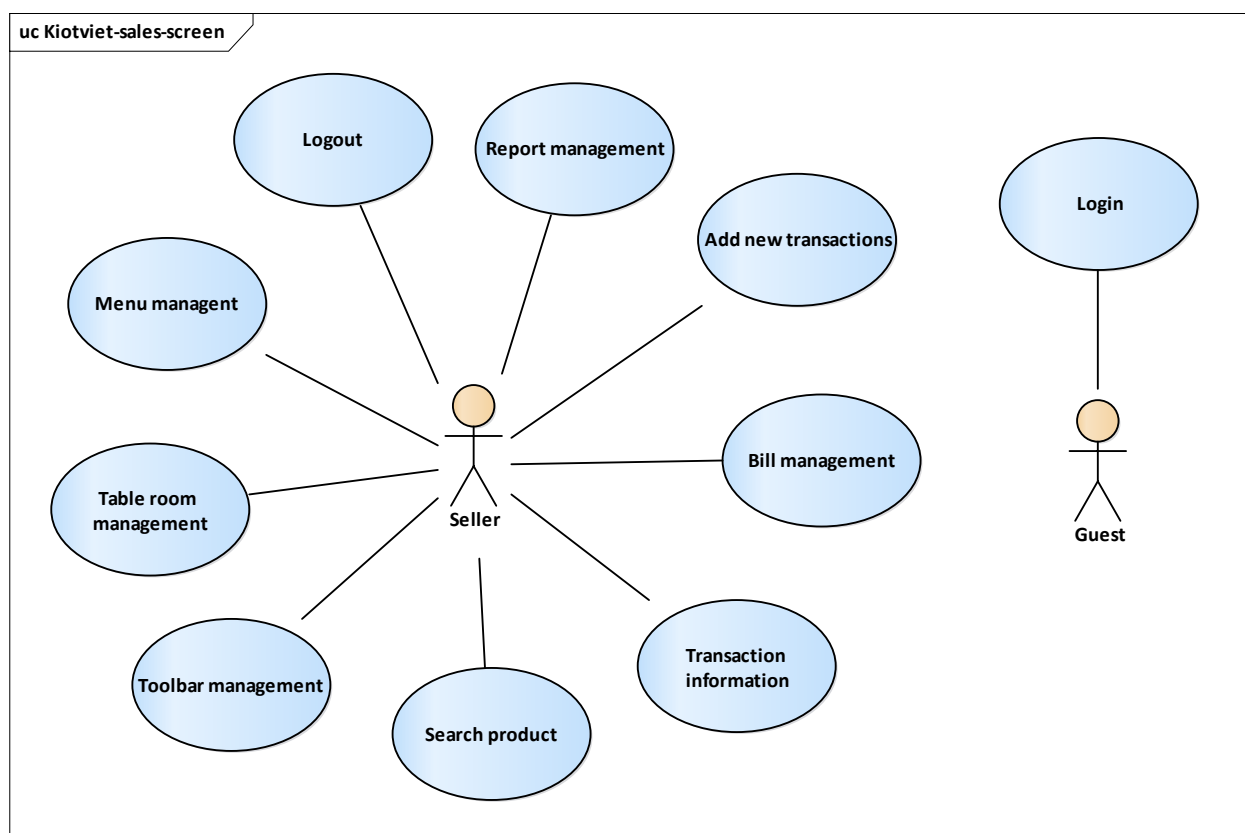


Figure 1.2.2.3.2. Function KiotViet on sales screen

1.2.2.4. Functions of Sales App

Function:

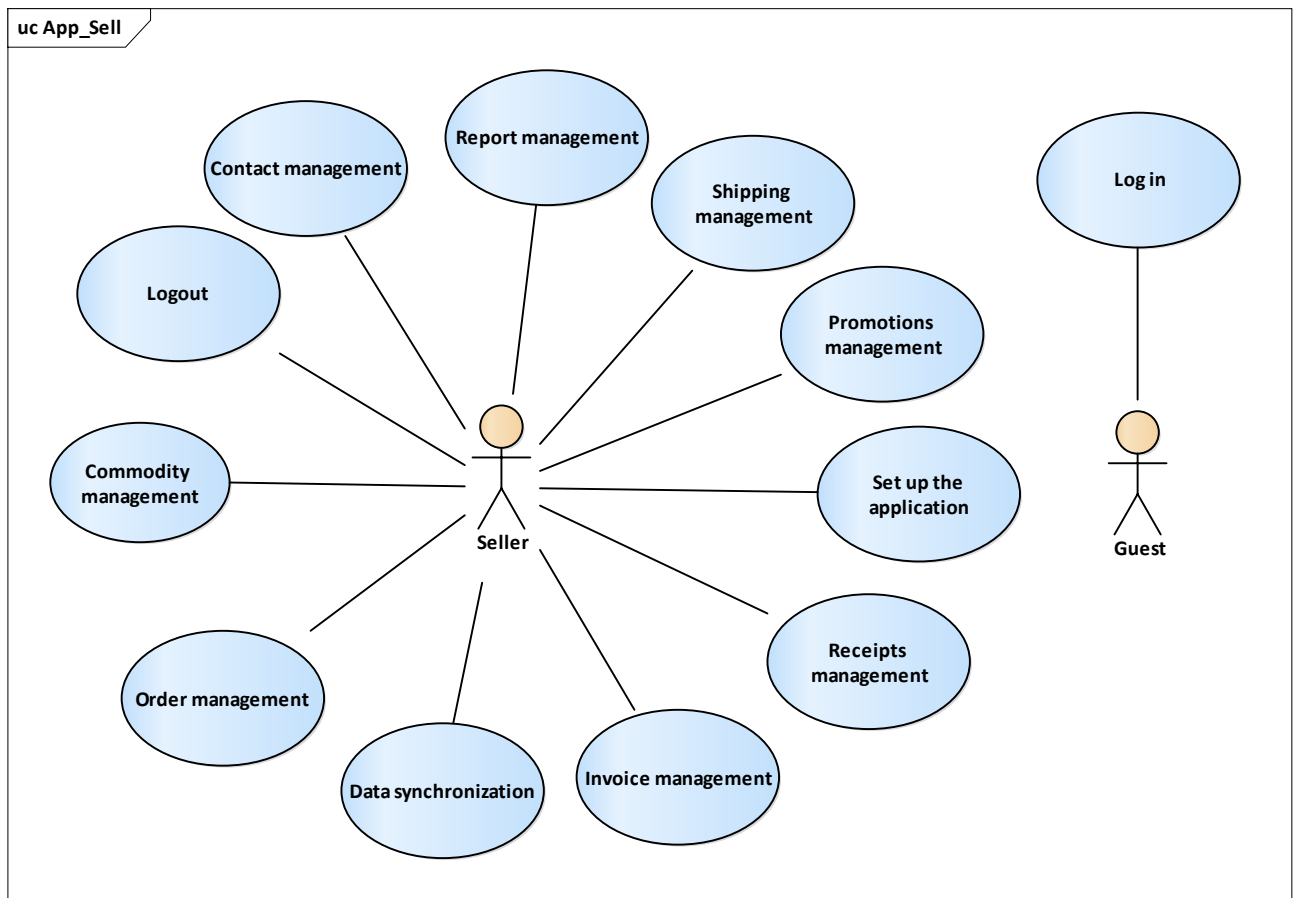


Figure 1.2.2.4.1. Function KiotViet on sell app

1.2.2.5. Functionality of App Manager

KiotViet Management application helps users perform booth management operations easily, quickly, accurately and instantly via phone or tablet.

Overview:

This is a feature on the Management application that helps users manage information quickly and most generally on the status of the operation of the stall from the statistics in the form of charts.

Function:

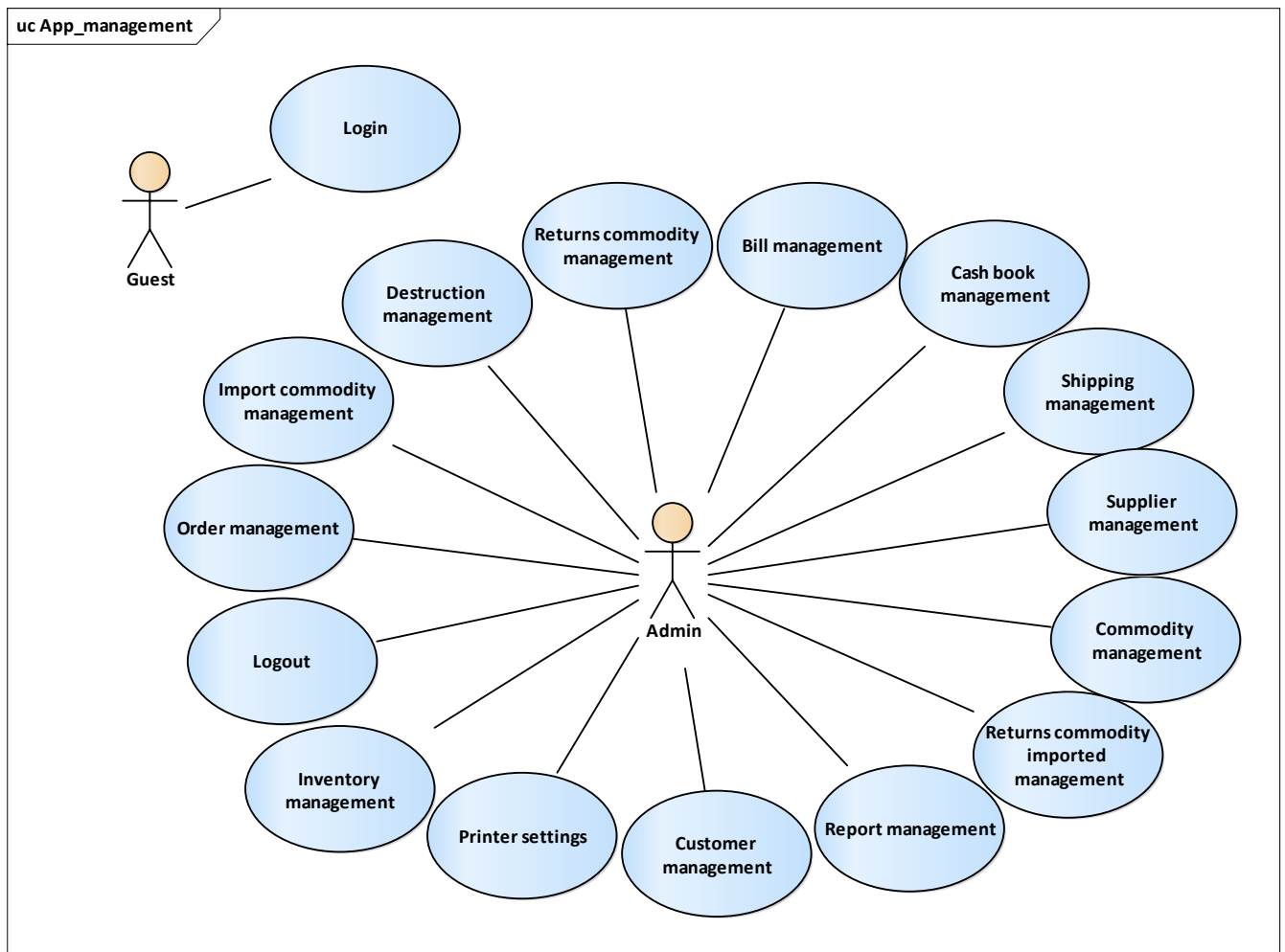


Figure 1.2.2.5.1. Function KiotViet on app management

1.2.3. Suno

1.2.3.1. Introduction

Suno is a software for managing sales, cash registers, and inventory activities online - helping you to manage stores remotely via phone, ipad.

Support all sales devices: receipt printers, barcode readers, barcode printers ... Help you sell more professionally and accurately.

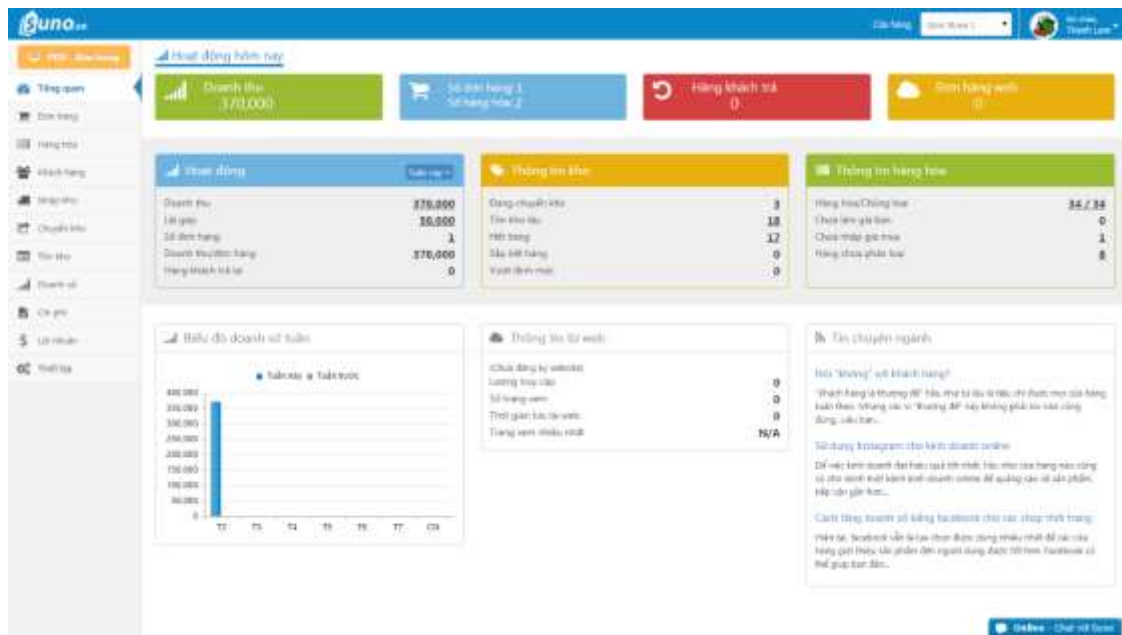


Figure 1.2.3.1.1. Introduction suno

You will like SUNO.vn because:

- Sales, billing, fast receipt printing extremely easy to use - just 5 minutes to get acquainted, even for people who do not know much about computers.
- Manage revenue, profits, accurate inventory.
- Connect easily with online sales channels: Bizweb, website, facebook,

1.2.3.2. Function

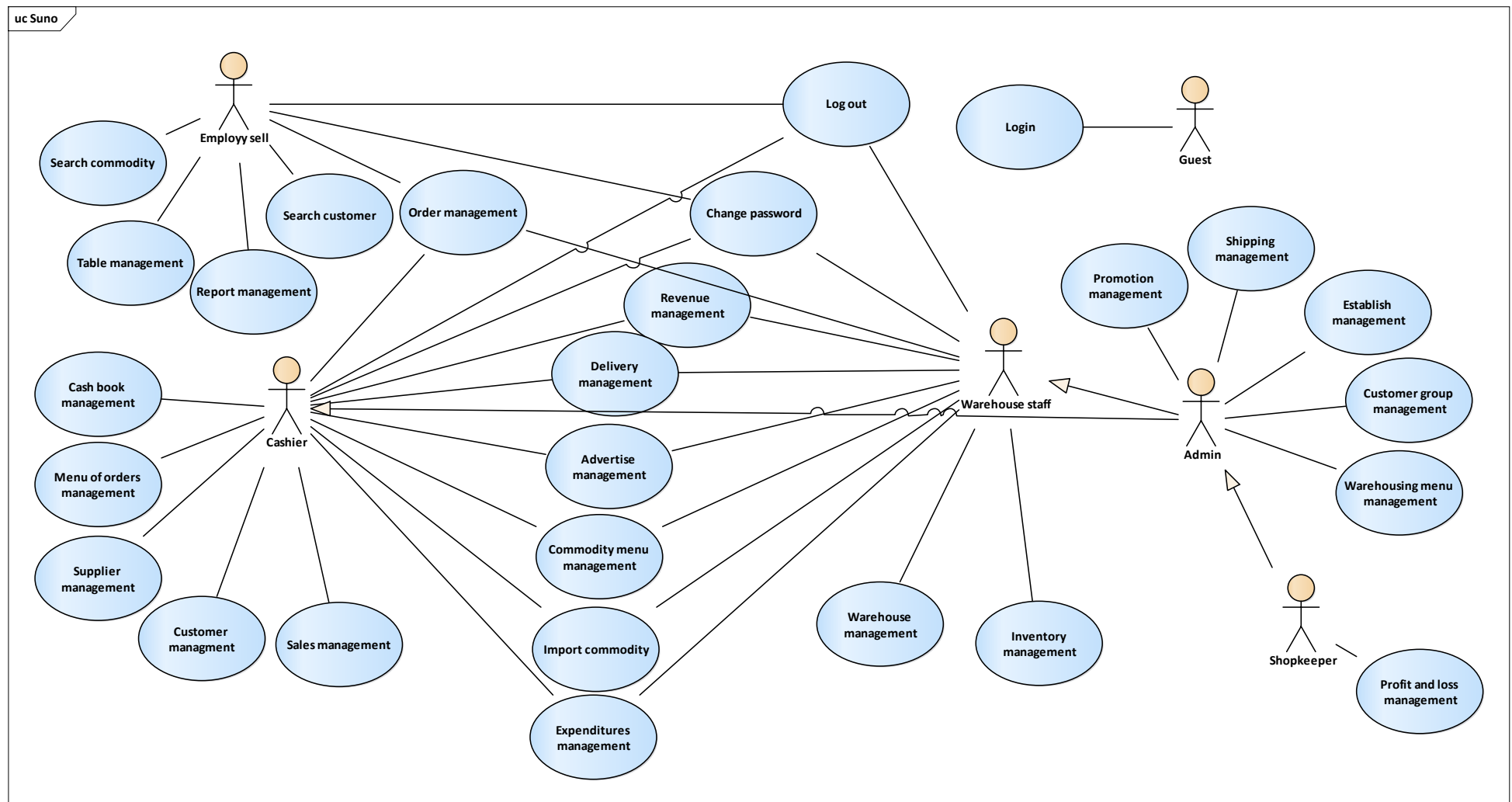


Figure 1.2.3.2.1. Function Suno application

1.2.4. Compare Ocha, Kiotviet and Suno

Table 1.2.4.1. Table compare ocha, kiotviet, suno

| | Ocha | Kiotviet | Suno |
|--------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Price | 125.000/month | 160.000/month | 180.000/month |
| Permissions | Minimal permissions | Maximum permissions | Minimal permissions |
| Function | Limited functional career | Fully functional career | Limited functional career |
| Report | Limited number of report features | Number of full reporting features | Limited number of report features |

1.3. Proposing ideas for new solution

With the results of collecting and examining the current systems, we propose a new solution to help bring together the subsystems to create a complete, unified and friendliest with users. The system will include the following sections.

1.3.1. Document Management System

This system will manage all documents, drawings, detailed description of the entire system, helping newcomers in construction and development can read and understand the system quickly and easily. In addition, this system will make maintenance and upgrades a breeze, or for transferring technology to a new team more easily and quickly.

Main functions:

- Create, edit, delete posts.
- View posts in a directory tree.
- Log in and delegate to admin.
- Share posts (twitter, linkedIn, facebook, email).
- Adjust the interface background color (light, dark, high constraint).
- Download the document as a pdf.

Advance functions:

- Comment with each post.

- Like and unlike with each post.
- Search posts.
- A new user can create an account with a contributor rule.
- An admin can manage and apply a new version for posts.

1.3.2. Store Management System

Store owners need to capture the business of their store.

There are many ways to manage your store: By sales invoices, using monthly reports, with CCTV. So that, we want to manage information visually, without many types of paper, no need to hire staff to write reports, no need to wait for employees to send monthly reports, ... We need to find a new solution, which saves more time, more money for customers and store owners.

From which information technology began to be applied to the trading business.

For each store, we will have a management application that will manage invoices and products at that store.

With current retail stores, an order management application is indispensable.

This application helps store owners manage sales invoices, existing products in the store, customer information, working hours and shifts of employees at the store.

From there, the store owner can keep track of real-time sales.

Main functions:

- Add, edit information of products for that store.
- See all promotions.
- Order products and manager invoices.
- Login, logout for employees.
- Manager a request song from customers in store.
- Look up customer information.
- View revenue statistics with each employee.
- Manage orders placed from mobile phones and websites.
- Manage invoice requests from customers in that store.

- Create/edit new account for customer.

1.3.3. Order System for each table

For large stores, the system at the store can integrate the order application at the tables for customers, they don't need to order at the counter or wait for employees to come to their orders or vice versa, employees must wait customers choose food, drinks, ... time-consuming for both parties and customers do not get the most comfortable.

With each table, we will have an order application for customer in that store.

Main functions:

- Login using QR code from Mobile app if they are a member.
- Logout if they have not sent order request yet.
- Choose products and send order request to sellers.
- View all coins which customer have from last order and use it when request invoice.
- Send a request to seller to issue an invoice when finished and logged out.
- Send a request to call a waiter/waitress.
- View all promotion combos and order according to the chosen combo.
- View an information of customer.

Advance functions:

- Send a request to seller play a music song.
- What to eat/drink/buy today, get suggestions from promotions from admin and order history.

1.3.4. Operation Management System

In order to manage its chain of stores, in addition to managing the activities at the shop, the store owner wants to manage and track the activities, sales, customers, etc. of the chain remotely and constantly updated. From there, it is necessary to build a system for the shop owner with special features.

The store chain management system can be built on many platforms such as applications for Windows operating systems, applications for MacOS or mobile applications (Android, IOS, Window Phone). So that, store owners want to use on any device, anytime and anywhere, this system is usually built on the web platform, allowing users to access anytime, anywhere and any device, just need a device with an internet connection, allowing users to update data quickly and continuously from their stores.

Main functions:

- Manage employees, products, and stores.
- Reports on sales, taxes, product inventory, discounts, depreciation of products.
- Automatically send reports on emails according to the set schedule.
- Payroll for employees, job applications, employee termination or general personnel management.
- Reports on customer requests, feedback, and opinions for our product chain.

Advance function:

- Customize the functions used throughout the system (enable/disable functions).

1.3.5. Electronic Commerce Website

To promote products to the market and bring convenience to customers, purchase at home or market for products without premises, ... an e-commerce website is essential.

Advantages:

- Marketing for the store right on social networks without ground rent.
- Buying and selling online at home, brings convenience to customers.
- Easy to hire third party advertising services, help many people know.
- There is no cost to display new products, avoiding costs for advertise new products.

Disadvantages:

- The system must ensure the safety of the transaction process, which are vulnerable to hackers.
- Must ensure confidentiality and information security for customers.
- Cannot use without internet connection.
- Exchange cost if the product is defective or defective by the shipper.

Main function:

- View, search, review products.
- Cart management.
- Manage payment methods.
- Connect to shipping units for delivery.
- Manage personal information, promotions, special offers for individuals.
- Manage orders delivered, shipping, processing, ...
- View and participate in the store's promotions.
- Chat with the buyer for direct support.
- Payment using ATM cards, credit cards, electronic wallets, ...
- Connect with shipping services such as Giaohangnhanh, Viettel Post, ... or fast carriers such as Grab Express, Go Viet, Bee, Loship, ...

1.3.6. Mobile Electronic Commerce

In addition to the quickly development of mobile devices today, each person virtually owns at least one smartphone, the demand for customer convenience is increasing. Since then, the response to the needs of customers is increasingly advanced. In order for customers to have the best experience for online sales system, mobile e-commerce applications have been born more and more, helping users to have an experience and most convenient.

Main function:

- View, search, review products.
- Cart management.

- Manage payment methods.
- Connect to shipping units for delivery.
- Manage personal information, promotions, special offers for individuals.
- Manage orders delivered, shipping, processing, ...
- View and participate in the store's promotions.
- Chat with the buyer for direct support.
- Payment using ATM cards, credit cards, electronic wallets, ...
- Connect with shipping services such as Giaohangnhanh, Viettel Post, ... or fast carriers such as Grab Express, Go Viet, Bee, Loship, ...

Advance functions:

- From the application, generate the QR code to log in to the order application at the shop.
- What to eat/drink today, get suggestions from promotions from your admin + orders history.
- When customer logged into the app at the shop, the user can send a request to play the song in store to the server app in the shop using the app on the phone.
- The promotions of each store.

1.3.7. Data Processing System

With large systems, handling of customer information is essential. To improve customer experience, to improve sales, marketing effectively for each customer, ... Since then, this system was born.

Main functions:

- Scanning virtual accounts, accounts that have been inactive for a long time, accounts in violation of regulations.
- Get information about products that customers have viewed to recommend for customers and offer promotions, ads, products displayed on the homepage to each customer.
- Learn and analyst application usage habits from customers.

- Create promotions and advertising with different customer groups, calculate revenue and expenditure to create the most beneficial promotions for the store.

Advance functions:

- Get information from the requests from guest and user about the songs to synthesize favorite songs for the whole system.
- Get information from user requests to suggest to the user when users open the song request screen.
- Get the information of drinks/foods that have been ordered to suggest the next order at the store and take a bonus for users for the next order at store as well as order on the phone.
- Create promotions for each customer: create suggestions from the customers' order history.

1.3.8. Other systems

- Operation Management System on Mobile (Admin, Stores owner).
- Mobile app for employees (includes security, seller, shipper, warehouse manager, ...)
- Web app for employees (includes security, seller, shipper, warehouse manager, ...)
- Marketing using public wifi:
 - + In stores, we have free wifi for customers to use, the store owner can install ads for the marketing for chain store when customers connect to wifi.
 - + This requires some configuration for your wifi and uses some third part devices.
 - + Show ads when customers have connected to wifi, then customers need to click connect to wifi to use the wifi. The ad will be displayed when customers access the screen and press the wifi connection.

CHAPTER 2. ROAD MAP

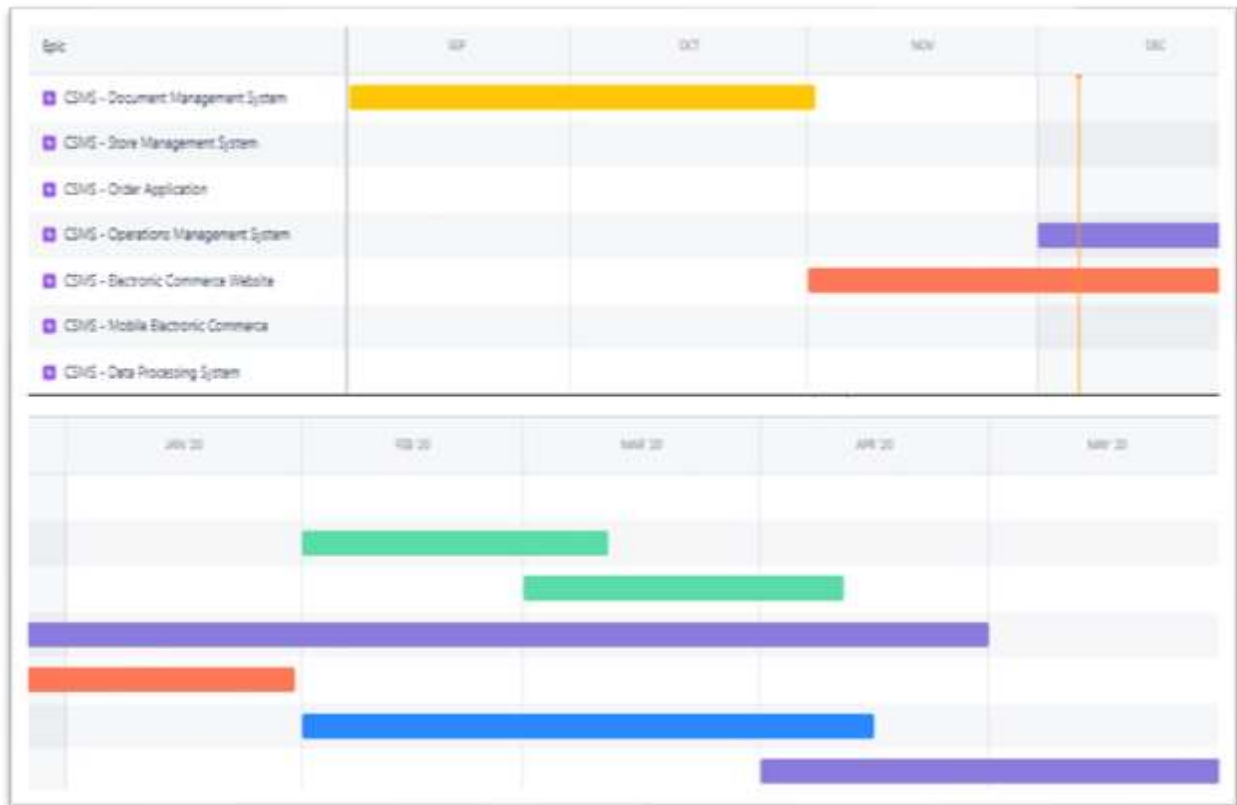


Figure 2.1. Roadmap of project

Follow about the roadmap, the plan of project is:

- Document management system: 2019/09/02 – 2019/11/01.
- Store management system: 2020/02/01 – 2020/03/11.
- Order application: 2020/03/01 – 2020/04/11.
- Operations Management System: 2019/12/01 – 2020/04/30.
- Electronic Commerce Website: 2019/11/01 – 2020/01/30.
- Mobile Electronic Commerce: 2020/02/01 – 2020/04/15.
- Data Processing System: 2020/04/01 – 2020/05/31.

CHAPTER 3. DESIGN

3.1. Architecture

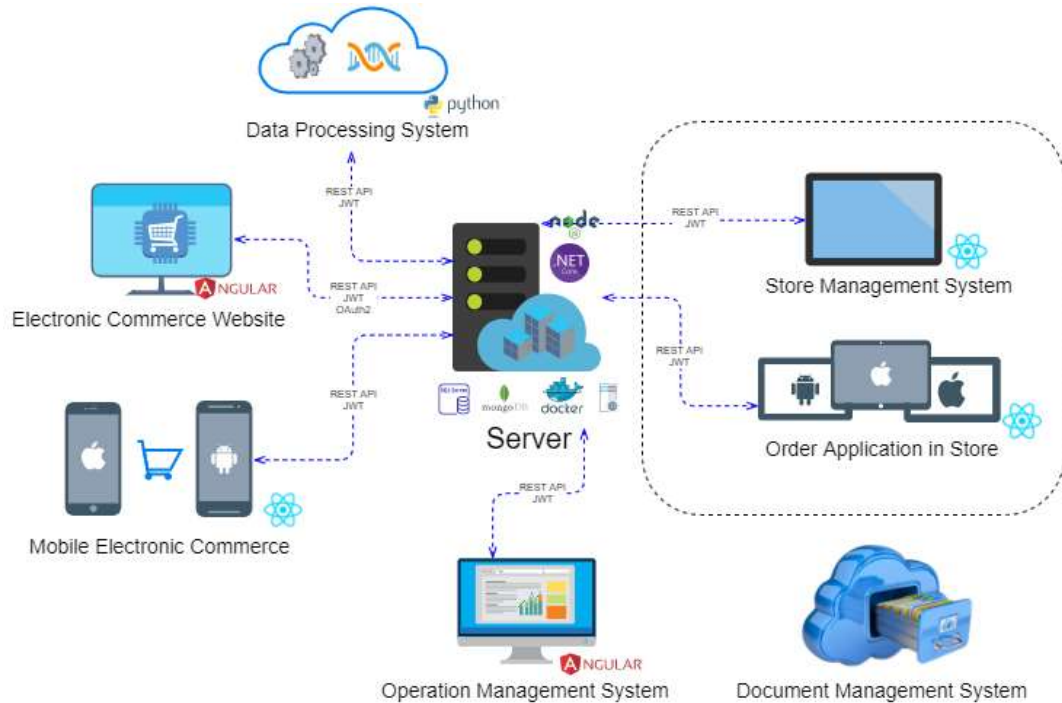


Figure 3.1.1. Overview of architectural design for system

3.2. Component

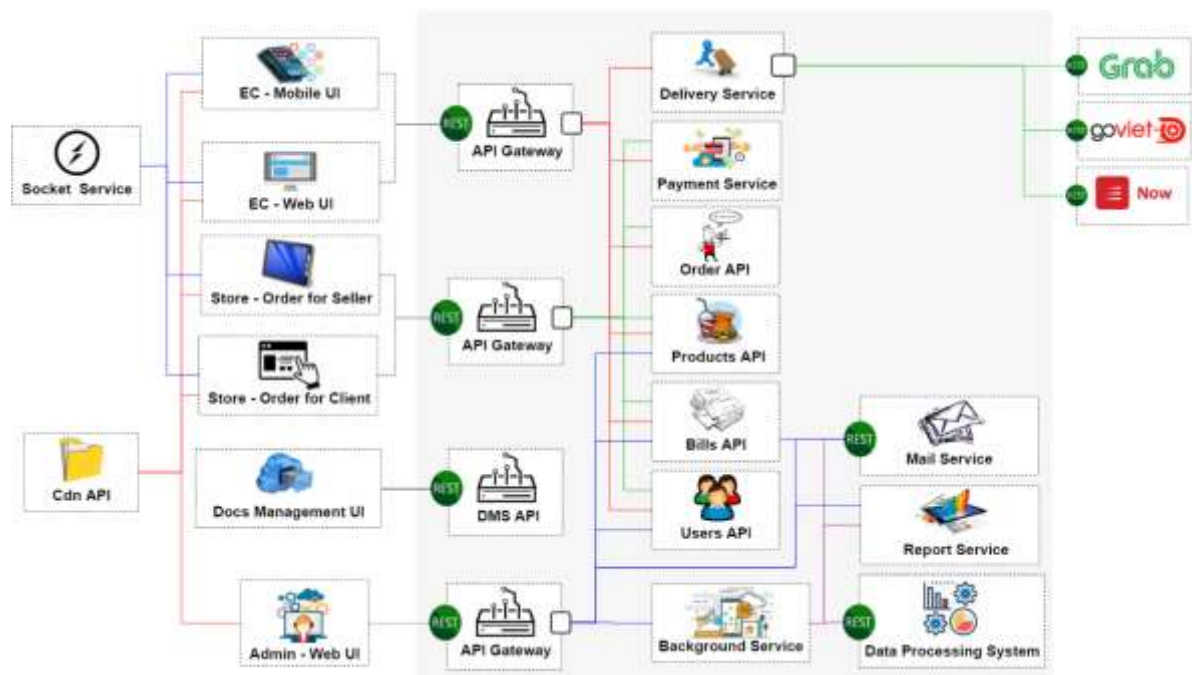


Figure 3.2.1. Overview of component design for system

CHAPTER 4. AUTHENTICATION AND AUTHORIZATION

4.1. Overview

Authentication (who you are) and authorization (what you can do). Microservices are always an indispensable component of every system.

4.1.1. Monolithic architecture

With a monolithic system, normally in such a system there will be a common module that manages authentication and authorization, and each user after login will be given a unique Session ID to identify.

The client can save the Session ID as a cookie and enclose it in every request. The system will then use the Session ID sent to determine the identity of the access user, so that the user does not need to re-enter the authenticate information in the next time.

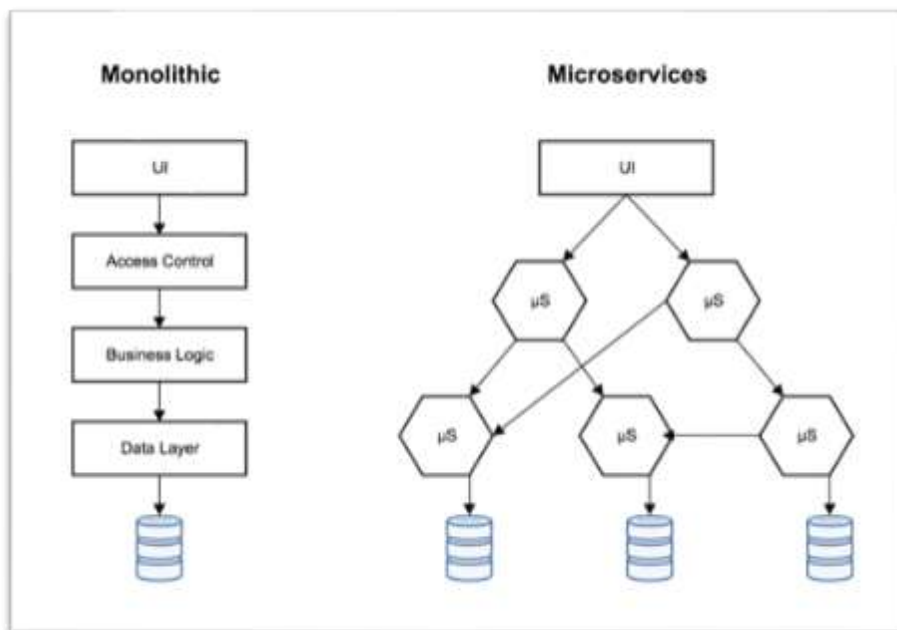


Figure 4.1.1.1. Monolithic vs Microservices

When the Session ID is sent, the server will identify the user associated with that Session ID, and will check the user's information to access the service or not.

Session and cookie solutions can still be used, but today we have more requirements, such as Hybrid or SPA applications (Single Page Application) that may need access to many different backend systems, so sessions and cookies from one server may not be available to another.

4.1.2. Microservices architecture

In microservices architecture, the system is broken down into many subsystems, taking care of different businesses and functions. Each of these subsystems also needs to be authenticated and decentralized, if handled in the way of the above Monolithic architecture, we will have the following problems: ^[3]

Each service needs to perform its own authentication and authorization in its own service. Although we can use the same libraries at each service to do it, the cost of maintaining that shared library with many different language platforms is too great.

Each service should focus on building its business, building on the logic of decentralization that decreases the speed of development and increases the complexity of the services.

Regular services will provide interfaces in the form of a RESTful API, using the HTTP protocol. HTTP requests will be passed through many components of the system. The traditional way of using server-side sessions (stateful) makes it difficult to scale the system horizontally.

The service will be accessed from different applications and users, be it a user, a hardware device, a 3rd-party, crontab or another service. Identifying such identity and authorization in many different contexts is extremely complicated.

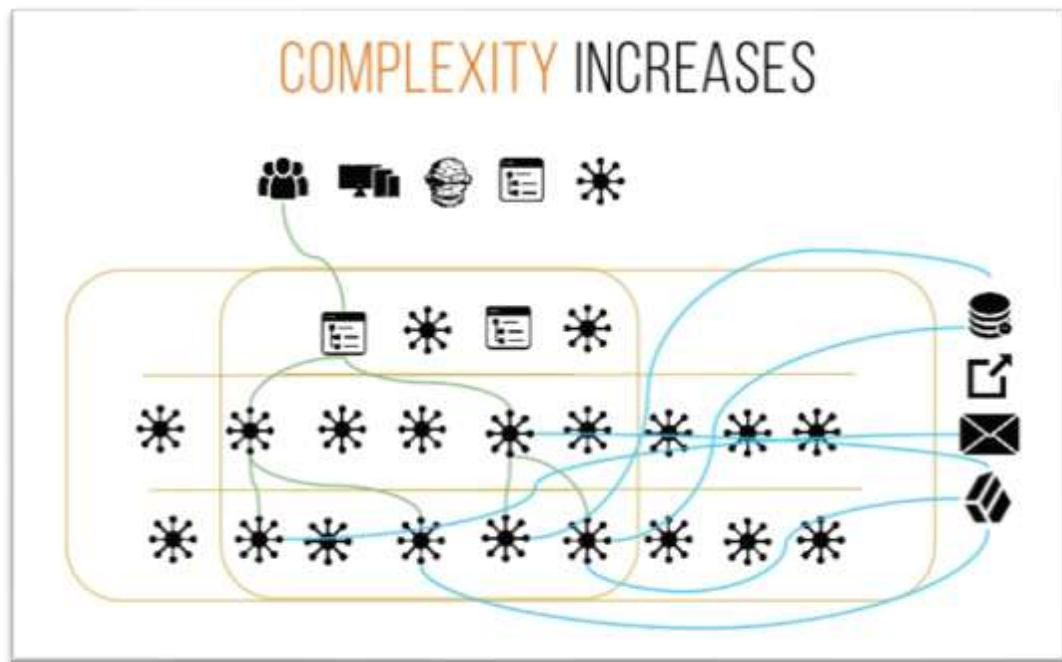


Figure 4.1.2.1. Lifecycle of request in microservices

4.1.2.1. JSON Web Token

JWT (Json Web Token) is an open standard token used to exchange information with HTTP requests. This information is verified and reliably marked based on the signature. JWT has many advantages over sessions.

- Stateless, information is not stored on the server.
- Easy to develop, expand.
- Better performance because the server reads the information in the request (if use session, the system must query information in storage or database).



Figure 4.1.2.1.1. Json web token structure

The signature part will be encrypted using HMAC or RSA.

HMAC: The JWT creator (token issuer) and the JWT (token verifier) receiver use the same secret key to encrypt and check.

RSA using 1 key pair, JWT initialization object uses private key to encrypt, JWT receiver uses public key to check.

So that, with HMAC both sides must share a secret key with each other, and the receiver JWT can completely generate another valid JWT code based on that secret key. As for RSA, the receiver uses the public key to check but cannot initiate a new JWT based on that key. Therefore, encryption using RSA makes the signature security better when sharing JWT with many different objects.

4.1.2.2. Opaque Token

Use Opaque Token when desired to control the session is better.

Opaque Token (also known as stateful token) is a type of token that does not contain information in it, usually a random string and requires an intermediary service to check and get information.

Transparent Token (also known as stateless token) is the JWT form, this token itself contains information and does not need an intermediary service to check.

Compare these two types of tokens:

Table 4.1.2.2.1. Compare Opaque Token vs Transparent Token

| Criteria | Opaque Token | Transparent Token |
|--------------------|--|---|
| <i>Performance</i> | Not good, because it must check token in Authentication Server | Good, because it just checked itself using secret key |
| <i>Encryption</i> | Do not expose sensitive information in the token content | Can view the token content |
| <i>Validity</i> | All tokens must be verified by the Authentication Server | The token will be valid until expiry |

Thus, Transparent Token have a better performance, simplicity and ease of use on both sides, don't need verified by the Authentication Server. Opaque Token has better control the sessions, for example when the system want to log out all devices that are logged in.

4.1.2.3. OAuth 2

The tokens will be generated via OAuth 2, which is the most popular functional method now, through which a service or a 3rd party application, can represent a delegation to give users access to an account. After that, user can access to some services in the system.

OAuth 2 is an open standard, full of documents, libraries in all different languages making integration and development based on it easy and fast.

4.1.2.4. Authentication, authorization at gateway service

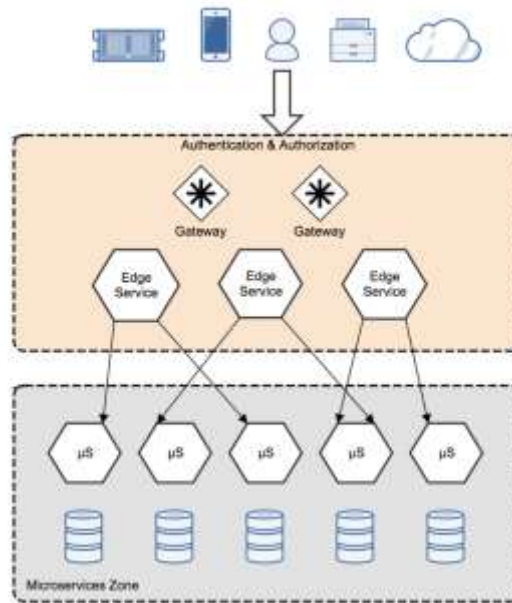


Figure 4.1.2.4.1. Authentication & Authorization at Gateway API

According to the model, all requests will be validated when passing through API Gateway. Authentication and authorization at this layer.

The API Gateway will force all requests to include a token to identify them

If this token is JWT, Gateway can check the validity of the token through signature, information (claim) or issuer.

After this token is Opaque Token, Gateway can analyze (introspect) tokens, exchange for JWT and pass it on to services.

API Gateway checks whether the policies are valid through the Authorization Server.

The microservices do not implement any authentication or authorization, it can freely access inside the internal network.

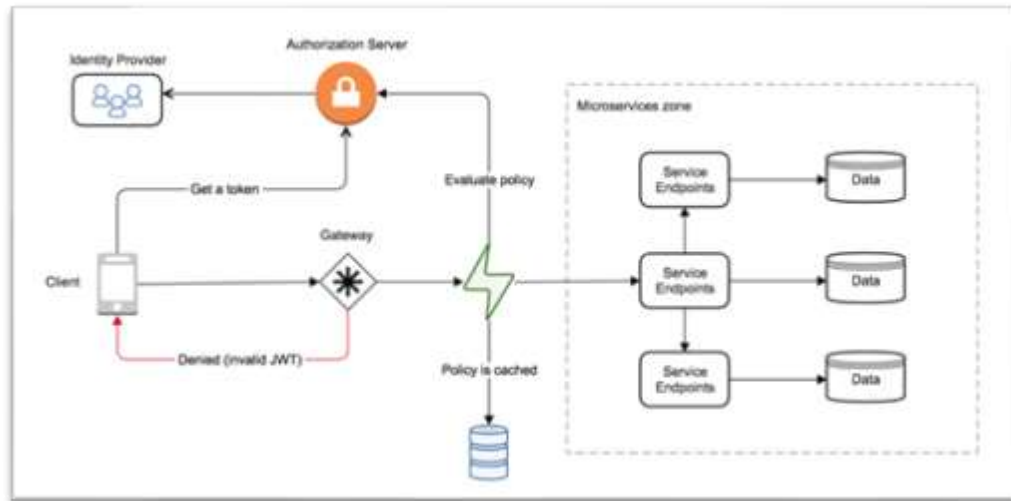


Figure 4.1.2.4.2. Authentication & Authorization at Gateway API diagram

This architecture has similarities with the Monolithic architecture when setting authentication at some services, construction and maintenance will pay a lower cost, but will reveal a huge security gap in the inner layer because the services can freely access each other.

4.1.2.5. Authentication, authorization at all services

In this architecture, each service when designed and built APIs (API Interface) can be expandable and access from the third party system. A service is built for business insiders, but in the future it may be ready to publish to external partners and developers.

This will help the service/development team be completely proactive about the existing resources, which resources are allocated to which objects, partially or fully accessed ...

To do this, a huge role will lie in the IAM (Identity Access Management) service, IAM holds the identifiers of all objects (user, service, command ...) along with detailed authorization rules for each resource type.

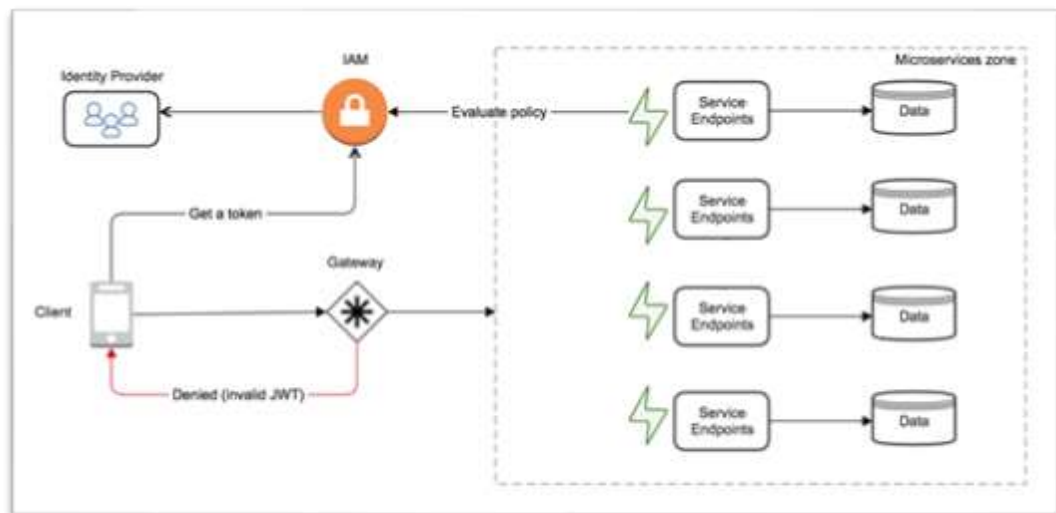


Figure 4.1.2.5.1. Authentication & Authorization at all services diagram

The fact that each service has to perform its own authentication and authorization will increase the cost when building the services.

However, services will have full autonomy, take the initiative to provide resources for objects, and accelerate development because many cases the client can access directly to the service without development add authentication layer in the middle.

4.1.2.6. Access Control

Access Control List (ACL):

Table 4.1.2.6.1. Access Control List

| Title | Owner Control | Promote Version | Modify Content | View Content | View Props | Publish |
|-------------------|---------------|-----------------|----------------|--------------|------------|---------|
| Administrator | x | x | x | x | x | x |
| John | | x | x | x | x | |
| Finance Admins | x | x | x | x | x | x |
| Finance Manager | | x | | x | x | |
| Finance Reviewers | | | | x | x | |

In the example above, a matrix of objects and permissions, it suitable for applications with few objects. As the system grows, this model will not be managed because the matrix created is too large and complicated. Therefore, this model is no longer popular at the moment.

Role-Based Access Control (RBAC):

RBAC links objects to roles, and from roles to permissions. For example, the Administrator role can inherit all the permissions that the Manager role has, which helps reduce the complexity of the permissions matrix, instead of assigning the entire rights to the Administrator, only need the Administrator to inherit the permissions of the Manager.

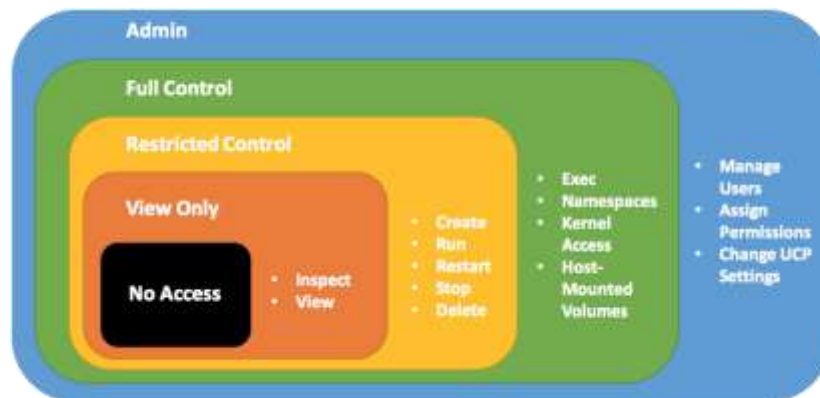


Figure 4.1.2.6.1. Role-Based Access Control

RBAC is very popular and it everywhere, compared to ACL, RBAC helps reduce the complexity when the number of objects and permissions increases. However, RBAC does not satisfy some cases, for example, when a product only by the creator, the user is in a specific department or the permissions to distinguish users from multiple different systems.

Policy-Based Access Control (PBAC):

PBAC is based on the Attribute Based Access Control (ABAC), which defines the permission to express a request to be allowed or denied. ABAC uses attributes to describe the object to be tested, each attribute is a key-value pair, for example Department = Marketing. Thus, ABAC can help decentralize more smoothly, suitable for many contexts and different business rules.

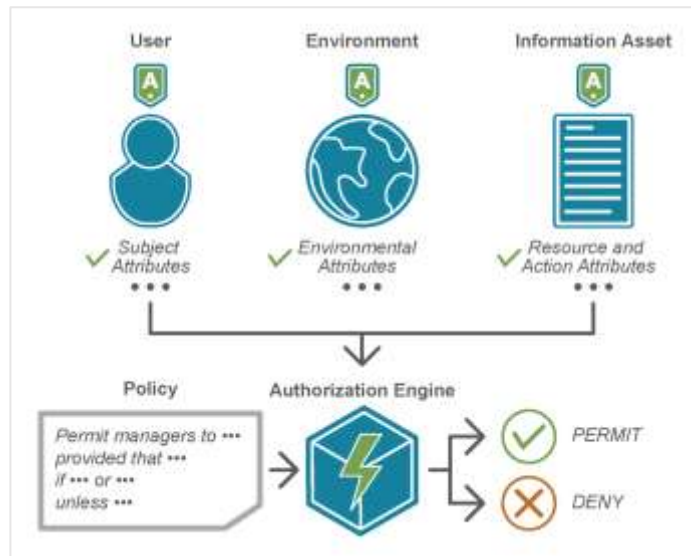


Figure 4.1.2.6.2. Policy-Based Access Control

PABC is defined through policies written in a common XACML language (eXtensible Access Control Markup Language). A policy defines 4 objects: subjects, effect, action and resource. For example, john (subject) is allowed (effect) to delete (action) product with john-leman (resource) ID. At first glance, it is similar to the definition of an ACL.

```
{
  "subjects": ["user: john"],
  "effect": "allow",
  "actions": ["catalog: delete"]
  "resources": ["product: john-leman"],
}
```

We can add subject, action as well as resource to policy.

```
{
  "subjects": ["user: john", "user: kathy", "user: perry"],
  "effect": "allow",
  "actions": ["catalog: delete", "catalog: update", "catalog: publish"]
  "resources": ["product: john-leman", "product: john-doe"]
}
```

What is the difference between PBAC and ACL:

Rule of Priority:

- By default, if there is no suitable policy, the request will be denied.
- If there is no deny policy, at least one policy is allowed.
- If there is a policy that is deny, then the request is always denied.

Regular Expression:

Policies allow declarations using a regular expression, as in this example that allows all users to see product information.

```
{
  "subjects": ["user: <.*>"],
  "effect": "allow",
  "actions": ["catalog: read"],
  "resources": ["product: <.*>"]
}
```

Condition:

Policies can add conditions to narrow the scope of rights, for example, apply only to a certain IP range, or only allow the creator of the product to update that product.

```
{
  "subjects": ["user: ken"],
  "actions": ["catalog: delete", "catalog: create", "catalog: update"],
  "effect": "allow",
  "resources": ["products: <.*>"],
  "conditions": {
    "IpAddress": {
      "addresses": [
        "192.168.0.0/16"
      ]
    }
  }
}
```

4.2. Solutions for project

4.2.1. Json web token

The project use microservices architechture, so that it must authentication and authorization in many services, so that Json Web Token is a good choice.

Advantages:

- Information is not stored on the server.
- Easy to develop and expand.
- Better performance because the server reads the information in the request.

4.2.2. Authentication and Authorization achitecture

In this architecture, each service when designed and built APIs (API Interface) can be expandable and access from the third part system. Continuous expansion of the business and the system requires the service to authenticate itself, thereby not distinguishing whether it is internal or external, making it easy for teams to expand integration with each other.

So that Authentication, authorization at all services is a good choice for a big system and easy to extend. For each service, it will use token in the request to authentication and authorization. It will be more secure and help them protect themselves.

4.2.3. Access control

Role-Based Access Control (RBAC)

In this project, the decentralization between roles is quite clear and there is no complexity between the permissions for many different target roles.

Role list:

- Admin: Admin_System, Admin_Store.
- Employees: Security, Seller, Manager, Shipper, Waiter.
- Customers: Guest, Customer.

CHAPTER 5. SYSTEM DETAILS

5.1. Document Management System

5.1.1. Overview

This system will manage all documents, drawings, detailed description of the entire system, helping newcomers in construction and development can read and understand the system quickly and easily.

In addition, this system will make maintenance and upgrades a breeze, or for transferring technology to a new team more easily and quickly.

5.1.2. Design

5.1.2.1. Architecture



Figure 5.1.2.1.1. Architecture of Document Management System

5.1.2.2. Database

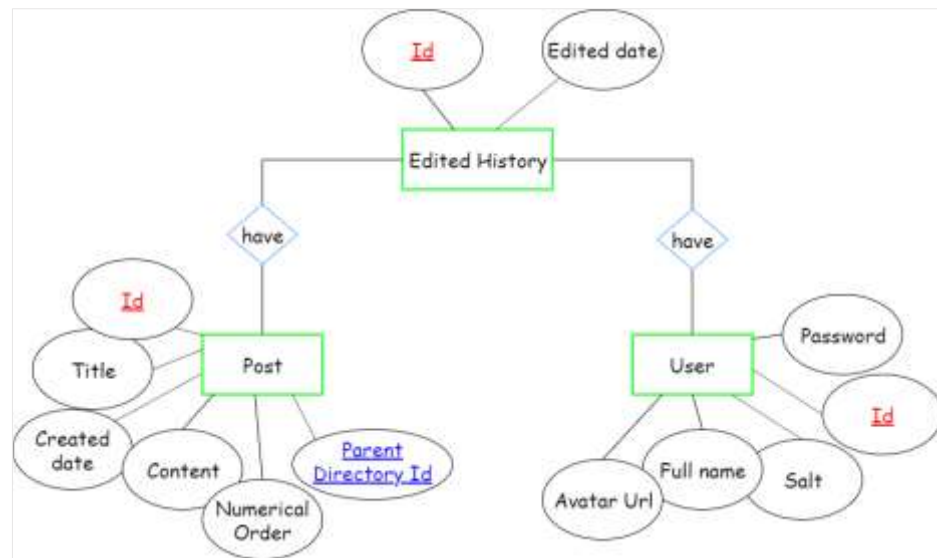


Figure 5.1.2.2.1. Entity relationship diagram for Document Management System



Figure 5.1.2.2.2. Database diagram for Document Management System

5.1.2.3. Interface

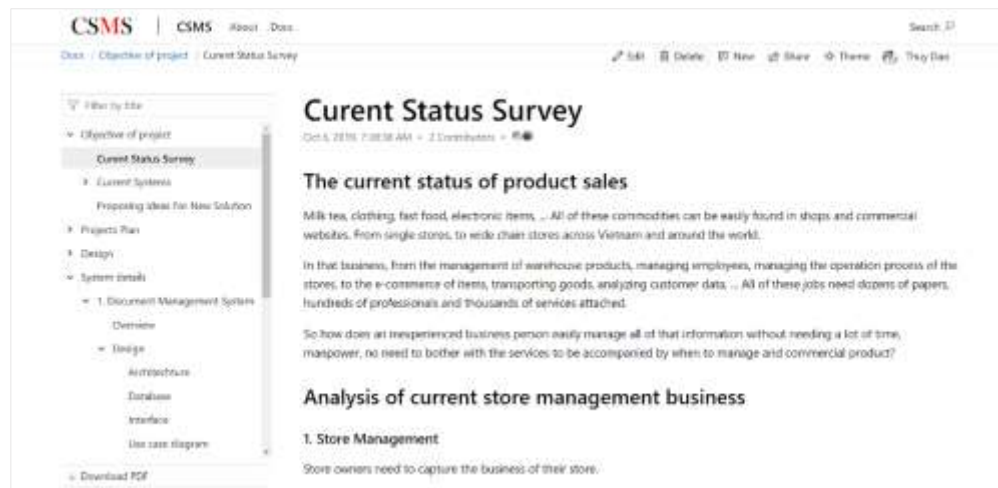


Figure 5.1.2.3.1. Main screen of Document management System

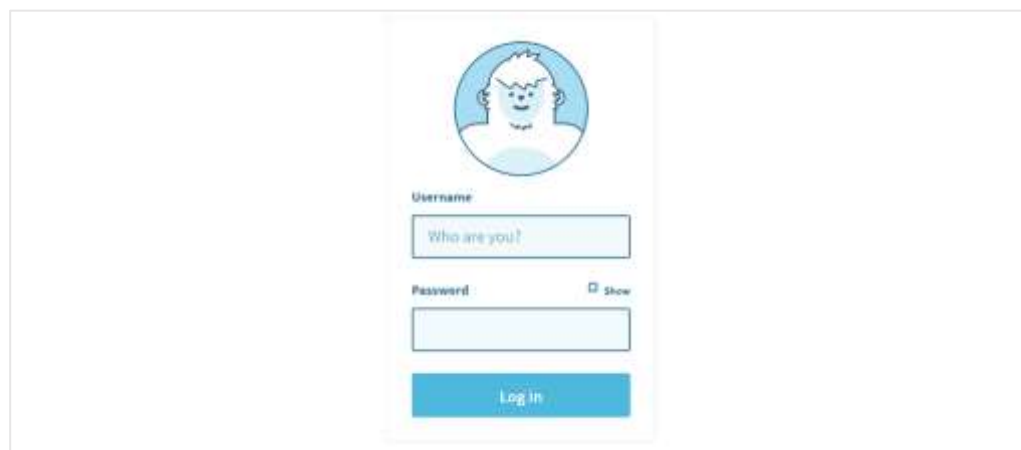


Figure 5.1.2.3.2. Login screen for Document Management System

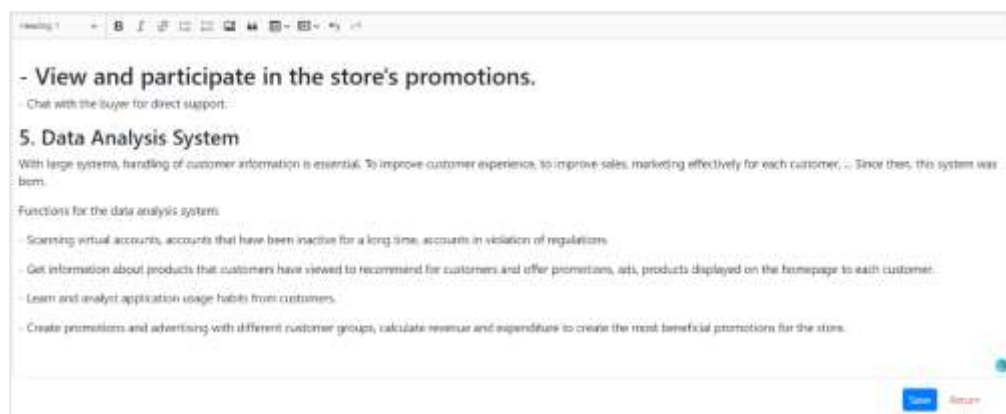


Figure 5.1.2.3.3. Edit screen for Document Management System

5.1.2.4. Use case diagram

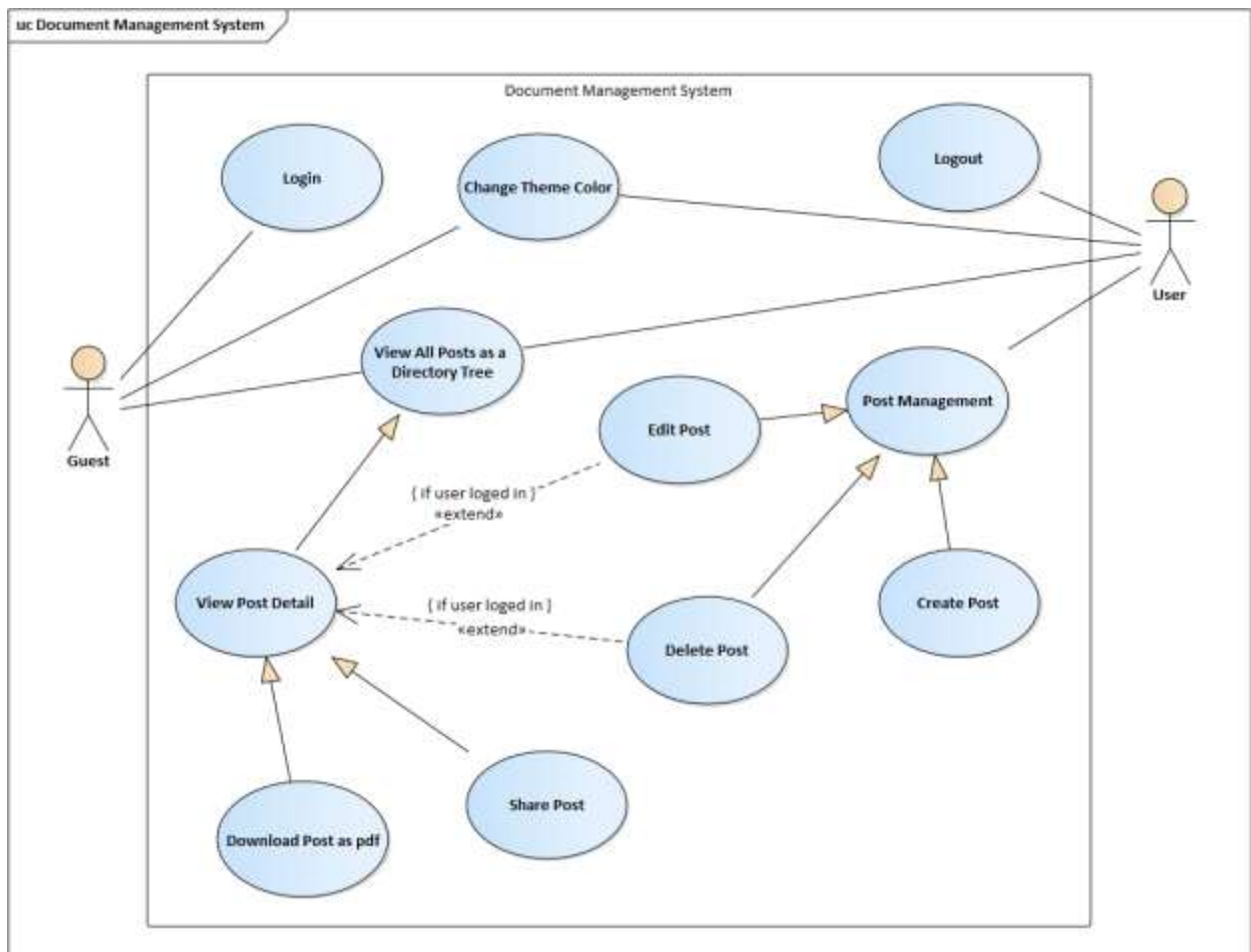


Figure 5.1.2.4.1. Use case diagram for Document Management System

5.1.2.5. Sequence diagram

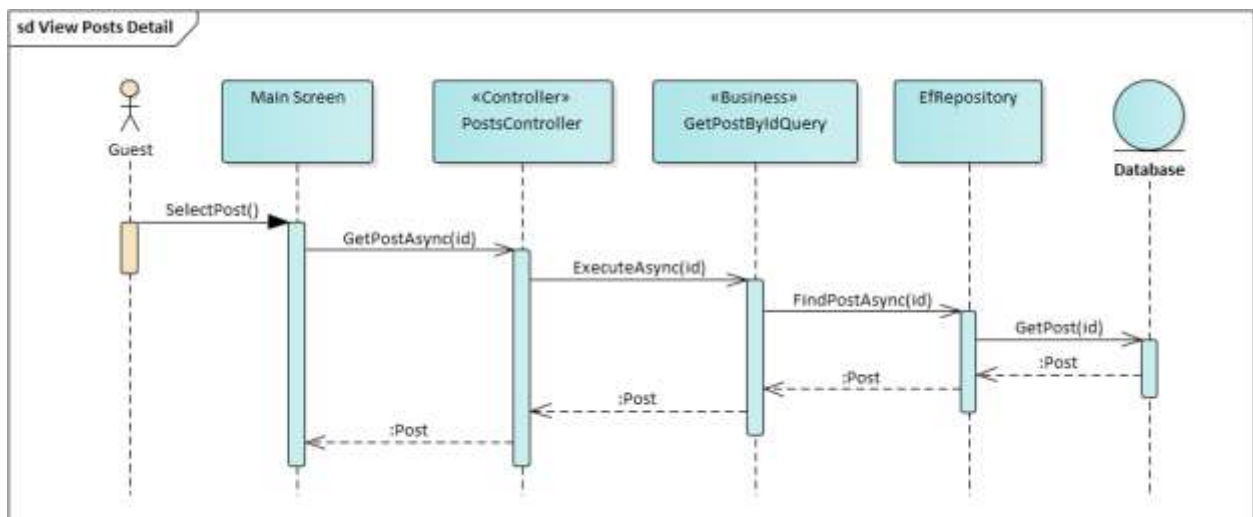


Figure 5.1.2.5.1. Sequence diagram for View Post Detail of Document Management System

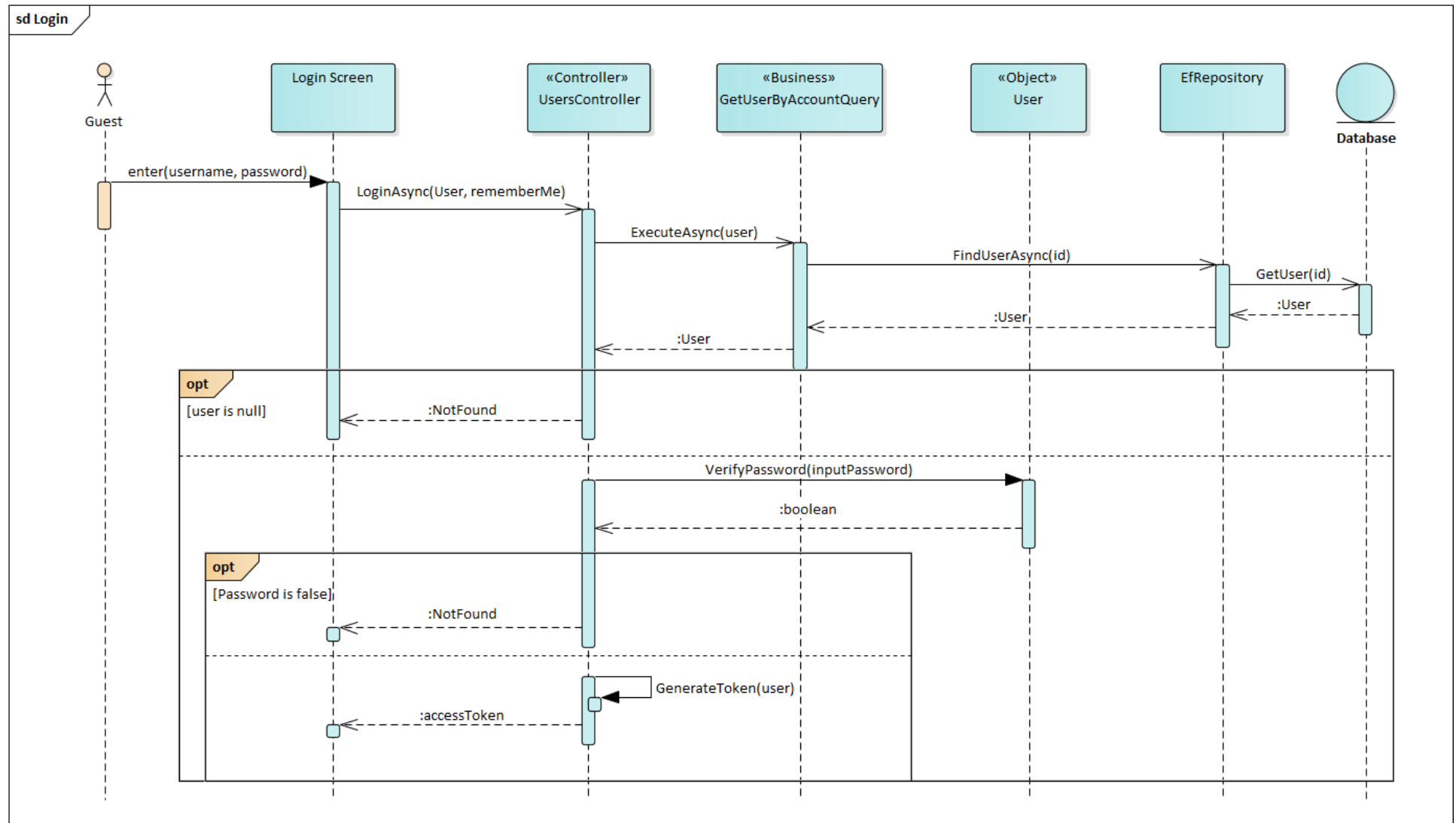


Figure 5.1.2.5.2. Sequence diagram for Login of Document Management System

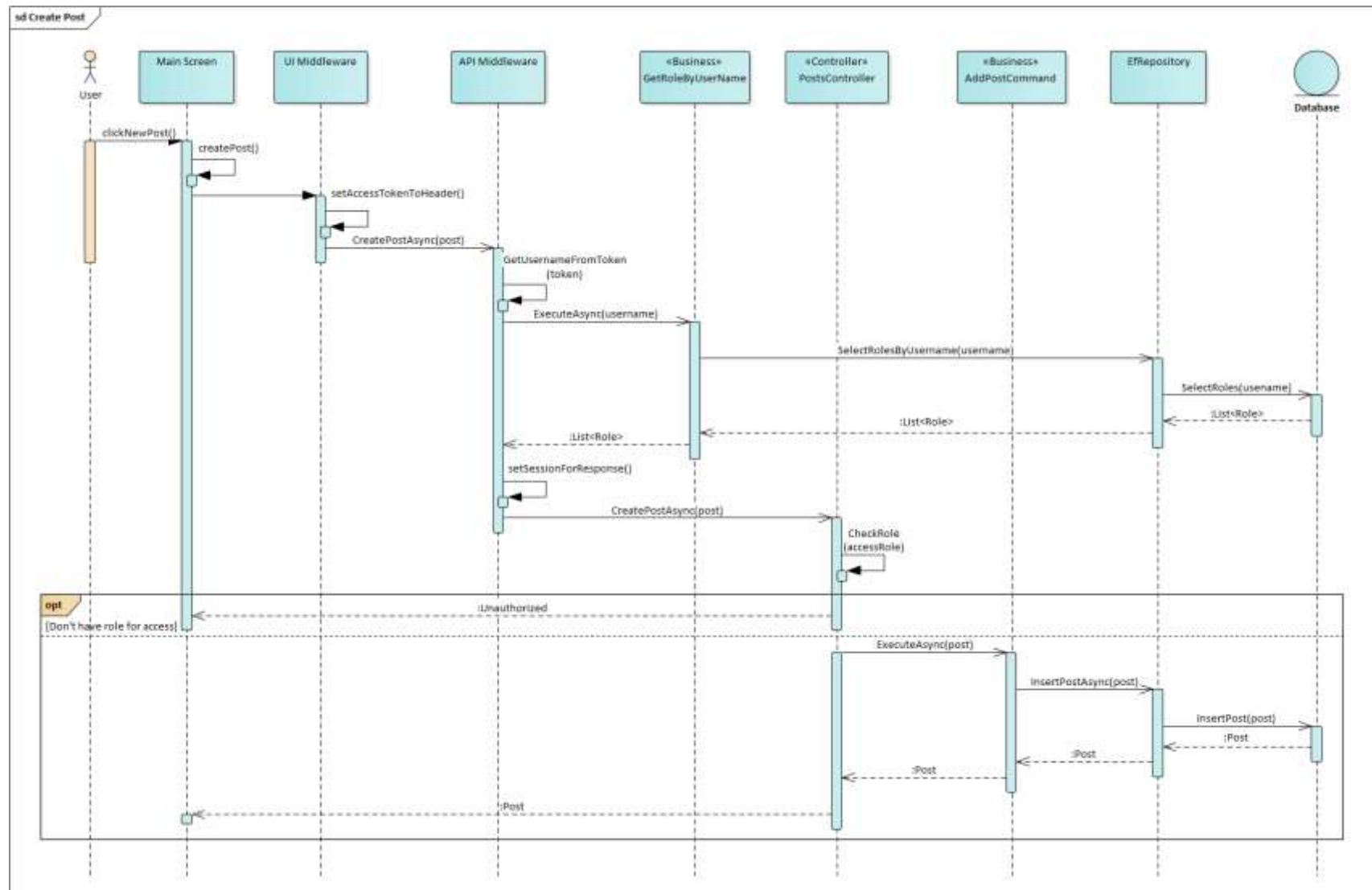


Figure 5.1.2.5.3. Sequence diagram for Create Post of Document Management System

5.1.3. Technologies

In this system, the technologies include:

- User interface using Angular 8.3.2.
- Backend using ASP.NET Core 3.0.
- Database using Microsoft SQL Server 2017.

Operation:

- Database: Amazon Relational Database Service.
- Frontend and backend using Docker container and deploy to Amazon Elastic Compute Cloud (EC2) with Ubuntu 18.04.
- Docker hub address: <https://hub.docker.com/r/thuydx9598/csms.dms>
- Source code: [https://github.com/thuydx98/csms](https://github.com/thuydx9598/csms)

5.2. Mobile Electronic Ecommerce

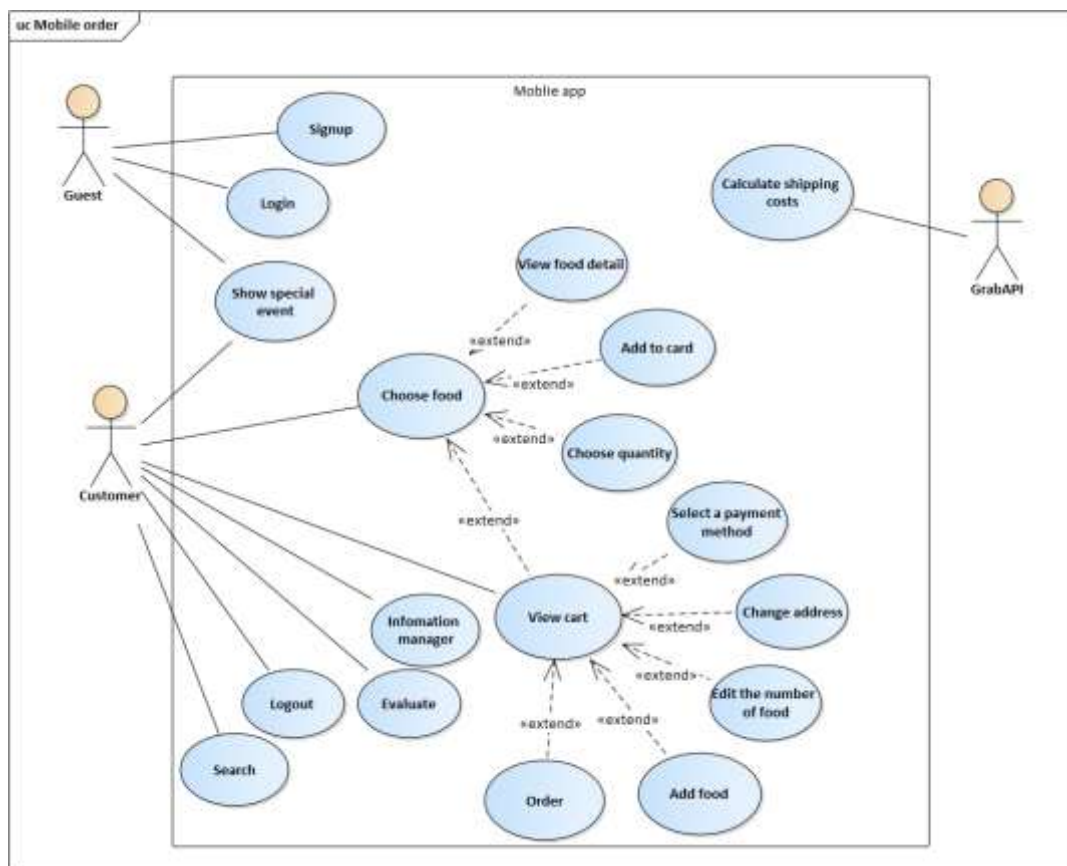


Figure 5.2.1. Use case diagram for Mobile Electronic Commerce

5.3. Electronic Commerce Website

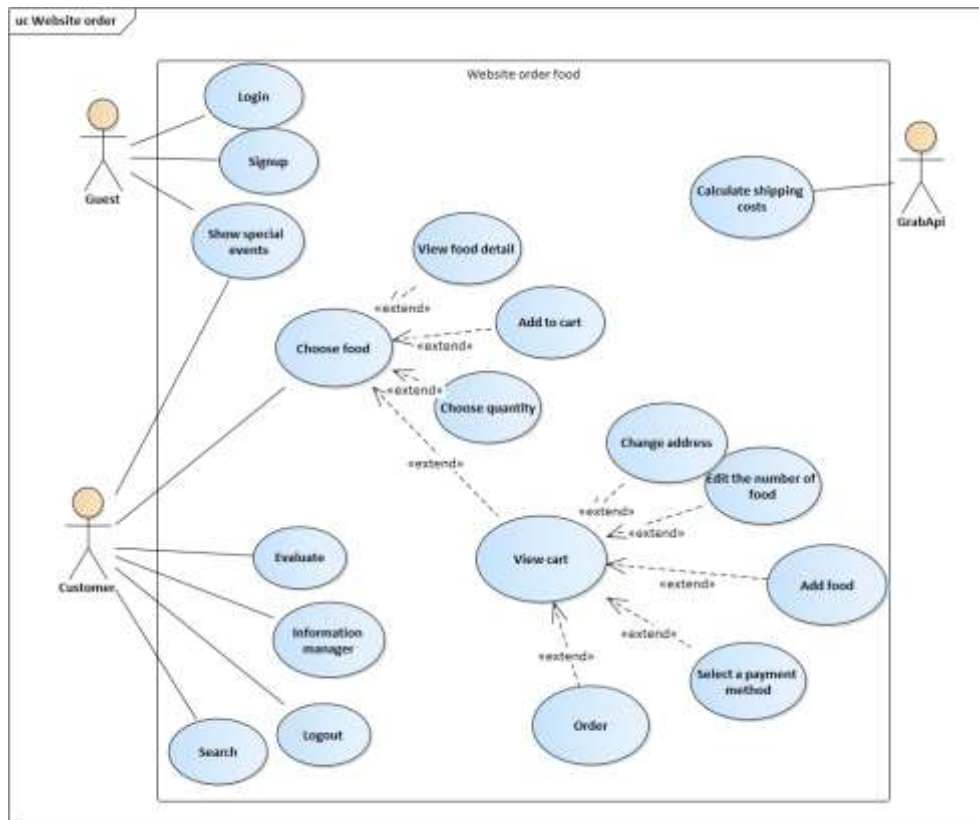


Figure 5.3.1. Use case diagram for Electronic Commerce Website

5.4. Operation Management System

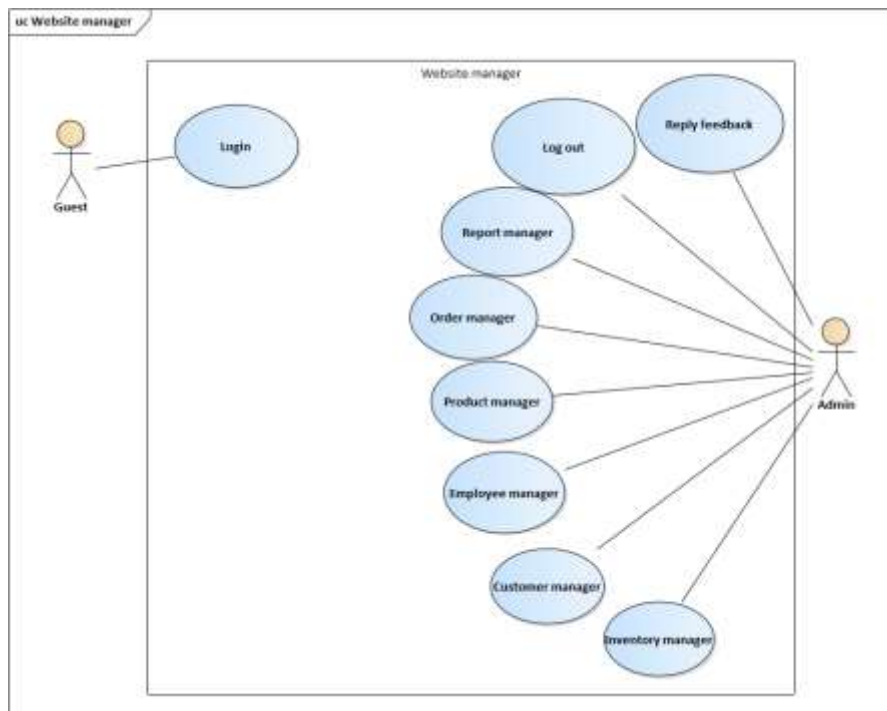


Figure 5.4.1. Use case diagram for Operation Management System

5.5. Store Management System

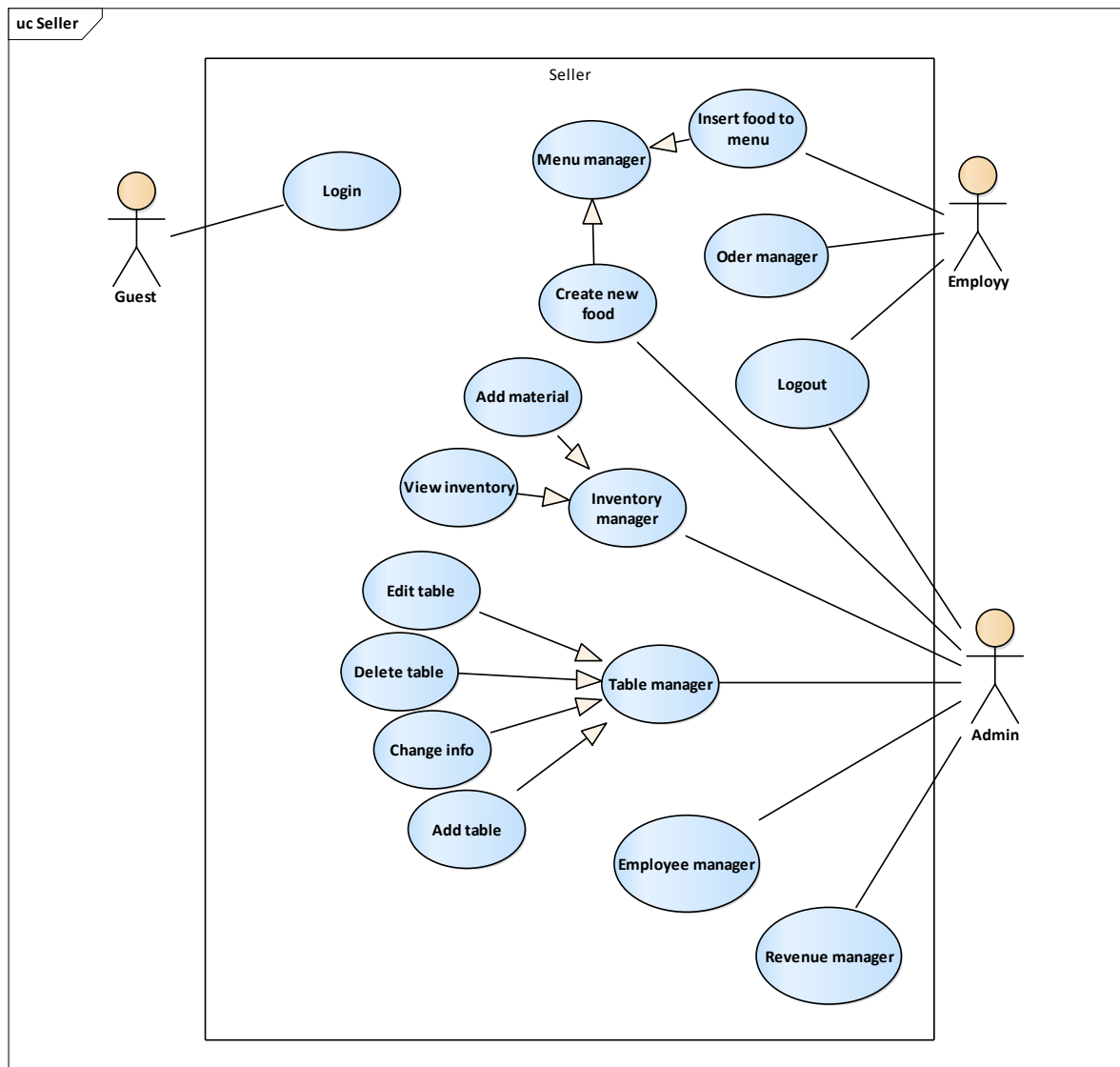


Figure 5.5.1. Use case diagram for Store Management System

CHAPTER 6. TECHNOLOGIES

In the project, the technologies will use which is listed in the table below:

Table 6.1. Technologies for project

| | |
|-----------------------|-----------------------|
| System Architecture | Microservices |
| | Docker |
| Transmission Protocol | REST API |
| | JSON Web Token |
| | OAuth2 |
| User Interface | Angular |
| | React Native |
| System Technologies | ASP.NET Core API |
| | Entity Framework Core |
| | Hangfire |
| | NodeJS |
| | Socket.io |
| Databases | Microsoft SQL Server |
| | MongoDB |

CONCLUSIONS

1. Results

The project is 20% complete, providing an overview of the architecture and system design. From there, we have an overview for the operation of the store chain management system. Thereby we have a knowledge about the scale and complexity of a large system as well as the security, development and expansion of a commercial project.

2. Advantages

- Solve the disadvantages of the current system.
- Building a comprehensive platform architecture on many platforms.
- Solve security issues for a commercial system using microservices architecture.
- Applying research knowledge to build a document management system for the project, thereby applying the theoretical knowledge studied and the basis for building services for the system.

3. Disadvantages

- Not surveyed the functionality with actual users yet.
- Performance has not been assessed with the current security solutions.
- Lack of practical experience in building system architecture.
- Performance has not been compared to a monolithic architecture.

4. Development strategy

- Actual survey of functions given to real users.
- analyze, evaluate and compare performance using microservices architecture compared to monolithic architecture.
- Analyze, evaluate and compare the authorization and authentication selected.
- Continue to complete the project and apply it to stores to keep updating system and fixing bugs.

REFERENCES

- [1] Kiot Việt, *Phần mềm quản lý bán hàng dành cho người Việt*, <https://www.kiotviet.vn/v-e-ki-ot-viet>, 28/10/2019.
- [2] Ocha, *Giới thiệu Ocha*, <https://ocha.vn/about>, 25/10/2019.
- [3] TopDev, *Xác thực và phân quyền trong Microservices*, <https://topdev.vn/blog/xac-thuc-va-phan-quyen-trong-microservices>, 30/10/2019.