VILNIUS UNIVERSITY

FACULTY OF INFORMATICS AND MATHEMATICS

SOFTWARE ENGINEERING

**LABORATORY WORK 2**

**“Point of Sale System’s Technical Specification and Architecture”**

Work done by:

Meinardas Klinkovas

Jorūnas Tocionis

Paulius Minajevas

Neringa Majauskaitė

Vilnius, 2023

# Table of Contents

[1. Table of Contents 2](#_Toc152375386)

[2. System components 3](#_Toc152375387)

[2.1. Major business processes in components 3](#_Toc152375388)

[2.1.1. Employee Management 3](#_Toc152375389)

[2.1.2. Product and Service Management 3](#_Toc152375390)

[2.1.3. Tax Rate Group Management 4](#_Toc152375391)

[2.1.4. Promotion and Loyalty Program Management 4](#_Toc152375392)

[2.1.5. Order Management 4](#_Toc152375393)

[2.1.6. Customer Management 4](#_Toc152375394)

[2.2. System functionalities 4](#_Toc152375395)

[3. API Endpoints 5](#_Toc152375396)

[3.1. Employee Management 5](#_Toc152375397)

[3.2. Product and Service Management 6](#_Toc152375398)

[3.3. Tax Rate Group Management 6](#_Toc152375399)

[3.4. Promotions and Loyalty Management 6](#_Toc152375400)

[3.5. Customer Management 7](#_Toc152375401)

[3.6. Order Management 7](#_Toc152375402)

[4. Non-functional requirements 8](#_Toc152375403)

[4.1.1. Performance 8](#_Toc152375404)

[4.1.2. Security 8](#_Toc152375405)

[4.1.3. Scalability 8](#_Toc152375406)

[4.1.4. Usability 8](#_Toc152375407)

[4.1.5. Availability 8](#_Toc152375408)

[4.1.6. Maintainability and Support 9](#_Toc152375409)

[4.1.7. Legality 9](#_Toc152375410)

[4.1.8. Details on scalability 9](#_Toc152375411)

# System components

Figure . API system component diagram

The diagram shows 7 components that represent the microservices of the system. Each component is responsible for unique tasks.

## Major business processes in components

### Employee Management

The Employee Management process involves the creation, update, and deletion of employee accounts. Business managers can create accounts, providing necessary information. Editing functionalities allow modification of roles and privileges. Business employees log in using credentials and log out when needed.

### Product and Service Management

This process encompasses the creation, retrieval, modification, and removal of both products and services. Business managers create and categorise products, assign tax rates, and manage categories. Similar actions apply to services. Editing categories ensures efficient organisation, and removal removes outdated items.

### Tax Rate Group Management

The Tax Management process, led by the Accounting and Finance team, includes adding, editing, and removing tax rate groups. This ensures accurate tax calculations during transactions, meeting compliance requirements.

### Promotion and Loyalty Program Management

Marketing managers handle the creation and management of promotions, discounts, and special offers. Loyalty programs are created, specifying earning rates and redemption options to encourage customer loyalty.

### Order Management

Frontline staff can create, modify, cancel orders and apply discounts and loyalty programs to customer bills. In addition, frontline staff prints bills, processes payments (payments are processed in “Payment” component and authenticated by third party service), and manages refunds. This ensures smooth transaction completion and customer satisfaction. Kitchen and frontline staff receive updates about order status, ensuring effective coordination between the front and back-end operations.

### Customer Management

Customers can have accounts which are used to track their purchases and provide loyalty benefits.

## System functionalities

1. Business managers can create, update, and delete employee accounts.
2. Employees can log in and log out of their accounts.
3. Managers can create, find, edit, and remove products. Products can be categorised, assigned tax rates, and managed efficiently.
4. Managers can create, find, edit, and remove services. Services can be categorised, assigned tax rates, and have available time slots.
5. The accounting and finance team can add, edit, and remove tax rate groups.
6. Marketing managers can create and manage promotions, discounts, and special offers. They can set dates, times, and specify applicable products or categories.
7. Marketing managers can create and manage customer loyalty programs. They manage loyalty points, earning rates and redemption options.
8. Frontline staff can create, modify, and cancel orders.
9. Kitchen staff can access order information related to kitchen tasks.
10. Frontline staff can print bills, process payments, and handle refunds.
11. Frontline staff can assign and manage tables. They can provide product information and check stock levels.
12. Business managers can log arrival and departure for frontline staff attendance tracking.
13. Frontline staff can apply discounts and loyalty programs to customer bills.
14. Frontline staff can receive and input customer feedback for service improvements.
15. Frontline staff can view order status.
16. Kitchen staff can set order status.
17. Support of multiple payment methods, including credit/debit cards, cash, and gift vouchers.

# API Endpoints

## Employee Management

1. Create Employee Account: `POST https://api.pos.com/employees`
2. Update Employee Account: `PUT https://api.pos.com/employees/{employee\_id}`
3. Delete Employee Account: `DELETE https://api.pos.com/employees/{employee\_id}`
4. Get Employee Account: `GET https://api.pos.com/employees/{employee\_id}`
5. Employee Login: `POST https://api.pos.com/employees/login`
6. Employee Logout: `POST https://api.pos.com/employees/logout`
7. Log arrival: `POST https://api.pos.com/shifts/{employee\_id}/arrival`
8. Log departure: `POST https://api.pos.com/shifts/{employee\_id}/departure`

## Product and Service Management

1. Manage Products: ` GET/PUT/DELETE https://api.pos.com/products/{product\_id}`
2. Add product `POST https://api.pos.com/products`
3. Categorize Products: `PUT https://api.pos.com/products/{product\_id}/category`
4. List Products: `PUT https://api.pos.com/products/{category} `
5. Assign Tax Rates: `PUT https://api.pos.com/products/{product\_id}/tax-group`
6. Manage Services: ` GET/PUT/DELETE https://api.pos.com/services`
7. Add Service: `POST https://api.pos.com/services`
8. List Services: `PUT https://api.pos.com/services/{category} `
9. Categorize Services: `PUT https://api.pos.com/services/{service\_id}/category`
10. Assign Tax Rates: `PUT https://api.pos.com/services/{service\_id}/tax-group`
11. Assign Time Slots `POST https://api.pos.com/services/{service\_id}/time-slots`
12. Create Category `POST https://api.pos.com/categories `
13. List Categories `POST https://api.pos.com/categories`
14. Manage Category `GET/PUT/DELETE https://api.pos.com/category/{category\_id}`

## Tax Rate Group Management

1. Manage Tax Rate Groups: `GET/PUT/DELETE https://api.pos.com/tax-rates/{tax\_rate\_id}`
2. Create Tax Rate Group: `POST https://api.pos.com/tax-rates`
3. List Tax Rate Groups: `GET https://api.pos.com/tax-rates`

## Promotions and Loyalty Management

1. Manage Discounts: `GET/PUT/DELETE https://api.pos.com/discounts/{discount\_id}`
2. Add Discount: `POST https://api.pos.com/discounts`
3. List Discounts: `GET https://api.pos.com/discounts`
4. Manage Loyalty Programs: `GET/PUT/DELETE https://api.pos.com/loyalty-programs/{loyalty\_program\_id}`
5. Create Loyalty Program: `POST https://api.pos.com/loyalty-programs`
6. List Loyalty Programs: `GET https://api.pos.com/loyalty-programs`
7. Set Points Earning Rates: ` PUT https://api.pos.com/loyalty-programs/{program\_id}/points`

## Customer Management

1. Manage Customer: `GET/PUT/DELETE https://api.pos.com/customer/{customert\_id}`
2. Add Customer: `POST https://api.pos.com/customer`

## Order Management

1. Manage Orders: `GET/PUT/DELETE https://api.pos.com/orders/{order\_id}`
2. Create Order: `POST https://api.pos.com/orders`
3. Print Bills: `POST https://api.pos.com/orders/{order\_id}/print`
4. Process Payments: `POST https://api.pos.com/orders/{order\_id}/payments`
5. Get Payment: `GET https://api.pos.com/payments/{payment\_id}`
6. List Payments: `GET https://api.pos.com/payments`
7. Handle Refunds: `POST https://api.pos.com/orders/{order\_id}/refund`
8. Add Item: `POST https://api.pos.com/orders/{order\_id}/item`
9. Manage Reservations: `GET/PUT/DELETE https://api.pos.com/reservations/{reservation\_id}`
10. Create Reservation: `POST https://api.pos.com/reservations`
11. List Reservations: `GET https://api.pos.com/reservations`
12. Apply Discounts/Loyalty: `PUT https://api.pos.com/orders/{order\_id}/apply-discounts`
13. Manage Order Status: `GET/PUT https://api.pos.com/orders/{order\_id}/order-status`

# Non-functional requirements

### Performance

* The system should process tasks with minimal delay, ensuring that most of the transactions and operations complete within 2 seconds.
* It should handle 150 operations an hour during peak times and 80 operations per hour on average without any visible degradation of the system (At peak hours system’s response time should only be up to 20% slower, than on average).

### Security

* Data encryption standards like TLS/SSL should be used to protect sensitive information during operations.
* The system must also comply with Payment Card Industry Data Security Standard to ensure secure usage of credit cards with the system

### Scalability

* The system should be optimised to allow for an increase of operations and users at any point in time.
* It should also allow for an easy way of implementing new features without disrupting the flow of the system.
* At minimum the system is expected to host 25 users at any time, with the average being around 50-60.

### Usability

* The user interface should be intuitive and easy to learn, allowing employees and other users of the system to operate it without needing to spend from 2 hours to a day depending on position.
* It should support different devices (phones, tablets, laptops, specialised devices) and screen sizes for flexibility.

### Availability

* The system should have an uptime of 99% and be running throughout the work hours.
* It should also recover within 60 minutes after a disruption without any loss of data.

### Maintainability and Support

* System should have consistent updates and bug fixes.
* Instructions on how to use the system should be easily accessible to anyone using it.

### Legality

* The system should comply with any tax regulations for accurate taxation during sales operations.
* Industry standards and any legal requirements should be considered for data storage and privacy handling.

### Details on scalability

* To increase the scale of data storage, additional memory within the used database should be added through hardware or, if using cloud-based databases, increased through the means specific to the cloud service.
* In case of performance, additional hardware would be the first option for increased processing power and improved response time. Upgrades to various data handling algorithms would allow for faster and larger amounts of data to be processed.
* Any updates to the system would be done with a detached, similar database to the live version by generating the same amount of data for testing, creating an environment that closely resembles the current conditions the PoS system is operating in.
* The system should allow for different types of payment, meaning that it should be easy to add new methods of payment with cash, different credit/debit cards, cryptocurrency or any other method that may be created in the future.
* There should be multiple supported languages at the launch of the system: English, German, Chinese, French, Portuguese and Spanish. The system should support more languages in the future.

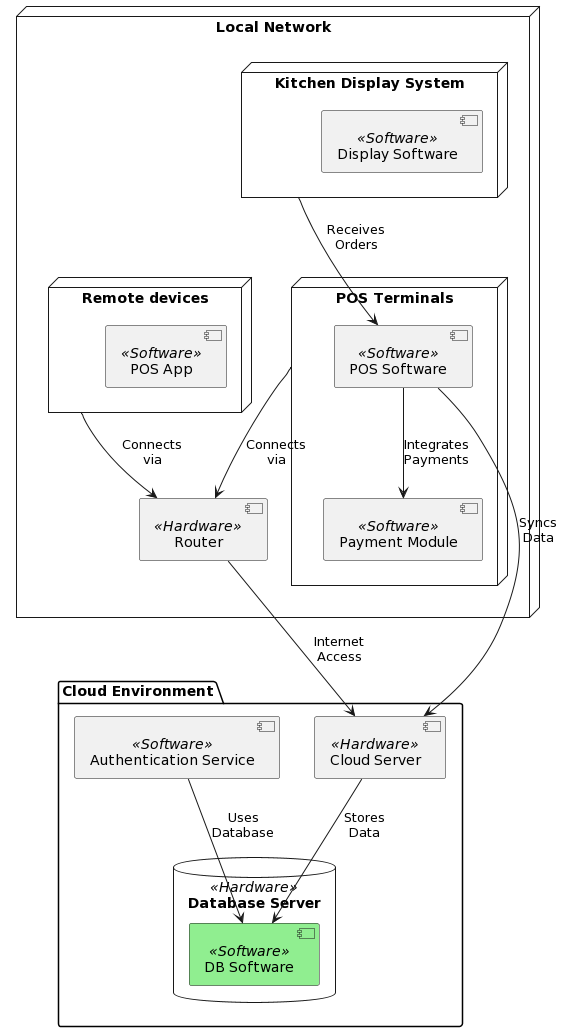


Figure . PoS System Deployment Diagram