VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

SOFTWARE ENGINEERING

Technical Specification and Architecture

3rd course

Vadim Čeremisinov

Vilnius

2023

Table of Contents

1. High-Level Design 4
   1.1. User Story Mapping 4
   1.2. Business Processes 4
   1.3. Component Breakdown 4
      1.3.1 Transaction Management Component: 5
   1.3.2 Customer Management Component: 6
      1.3.3 Employee Management Component: 7
      1.3.4 Inventory Management Component: 8
   1.3.5 Loyalty Program Management Component 9
      1.3.6 Reporting and Analytics Component **(nėra būtinas, geriau nedėt):** 10
      1.3.7 Reservation and Appointment Management Component **(discuss again)** 11
      1.3.8 System Administration Component (logs – hostinger/azure, part of system environment, System settings better without configuring,): 12
2. API Endpoints 13
   2.1. Endpoint Descriptions 13
      2.1.1. Transaction Management Component 13
      2.1.2. Inventory Management Component 13
      2.1.3. Employee Management Component 13
      2.1.4. Reporting and Analytics Component 13
      2.1.5. System Administration Component 14
      2.1.6. Customer Management Component 14
      2.1.7. Security and Compliance Component ..... **Error! Bookmark not defined.**
   2.2. Implementation Definition 14
   2.3. Logical Grouping 14
3. Non-Functional Requirements 15
   3.1. Performance 15
   3.2. Legal Requirements 15
   3.3. Usage Context 15
   3.4. User Experience 16
   3.5. Infrastructure and Deployment 16
4. Scalability 17
   4.1. User Expectations 17
   4.2. Geographic Usage 17
   4.3. Internationalization (i18n) 17

4.4. System Availability 17
4.5. Legal Aspects 17
4.6. Usage Patterns 17
4.7. Read/Write Ratios 17
4.8. Platform Focus 17
4.9. External System Dependencies 18

1. High-Level Design

1.1. User Story Mapping

- **Administrators Role**
    - Story 1: "As an administrator, I want to manage discounts and loyalty programs efficiently to enhance customer satisfaction."
    - Story 2: "As an administrator, I need comprehensive control over employee information for effective system management."
- **Customers Role**
    - Story 1: "As a customer, I want to easily order food, book rooms, and request services online for convenience."
    - Story 2: "As a customer, I expect to participate in loyalty programs and receive discounts seamlessly during my transactions."
- **Waiters, Chef, Specialists, Receptionists Role**
    - Story 1: "As an employee (waiter, chef, specialist, receptionist), I need to manage orders, appointments, and reservations effectively to ensure customer satisfaction."

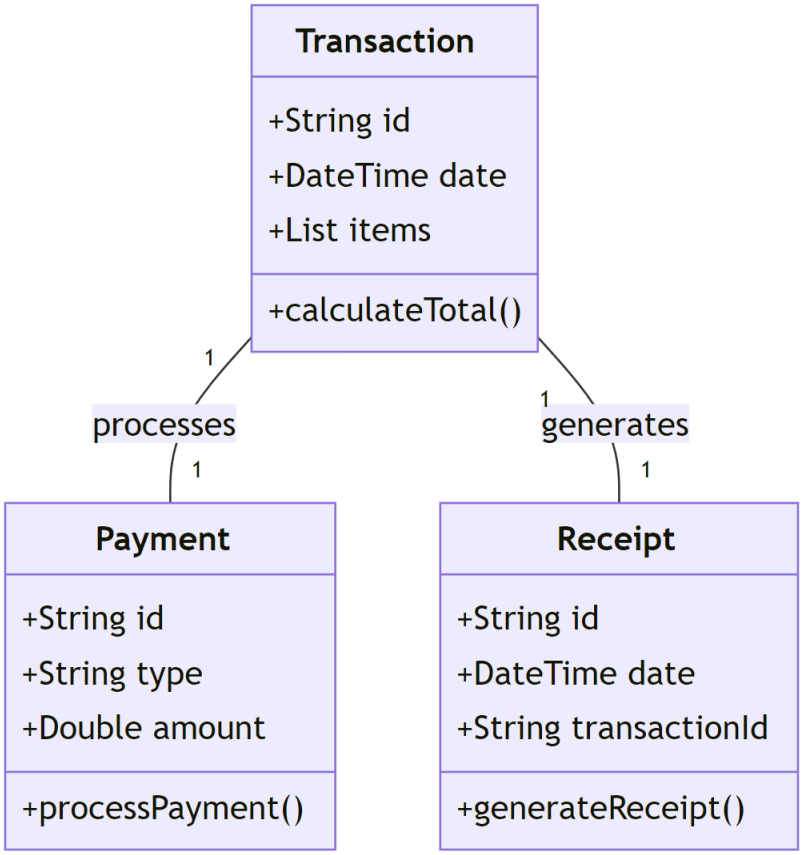
1.2. Business Processes

- **Order Management and Processing**
    - Includes order creation, modification, status updates, and completion in various settings (restaurant, beauty salon, etc.).
- **Payment and Transaction Handling**
    - Handling diverse payment methods (cash, card, gift voucher), applying discounts and loyalty points, generating receipts.
- **Scheduling and Appointment Management**
    - For beauty salons and hotels, managing appointments, and room reservations efficiently.

1.3. Component Breakdown

- **Transaction Management Component**
- **Customer Management Component**
- **Employee Management Component**
- **Inventory Management Component**
- **Loyalty Program Management Component**
- **Reservation and Appointment Management Component**
- **Reporting and Analytics Component**

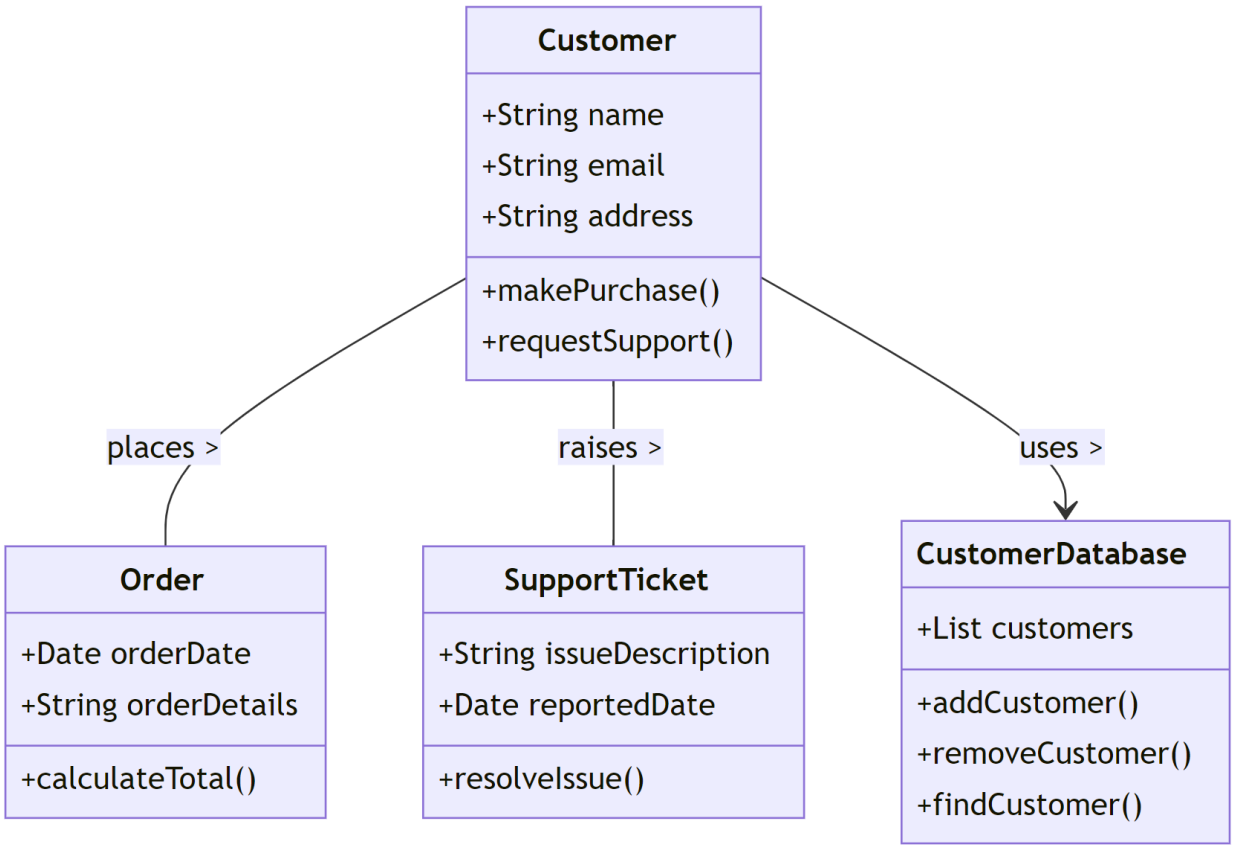
1.3.1 Transaction Management Component:



Classes: Transaction, Payment, Receipt.

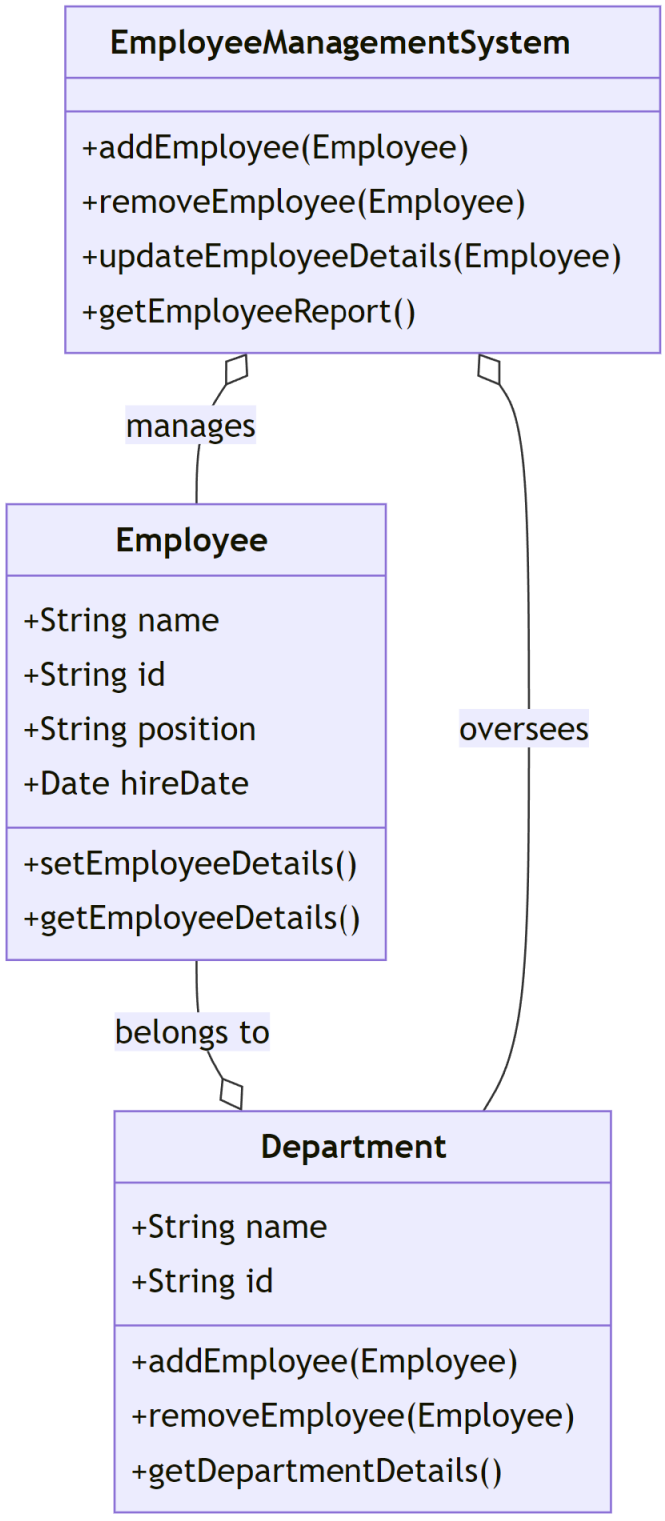
Relationships: Transaction should be associated with both Payment and Receipt.

1.3.2 Customer Management Component:



- **Customer**: Represents the customer with attributes like name, email, and address, and methods for making purchases and requesting support.
- **Order**: Details about customer orders, including order date and details, with a method to calculate the total.
- **SupportTicket**: Manages support tickets raised by customers, with attributes for issue description and reported date, and a method to resolve issues.
- **CustomerDatabase**: Handles the database of customers, with methods to add, remove, and find customers.

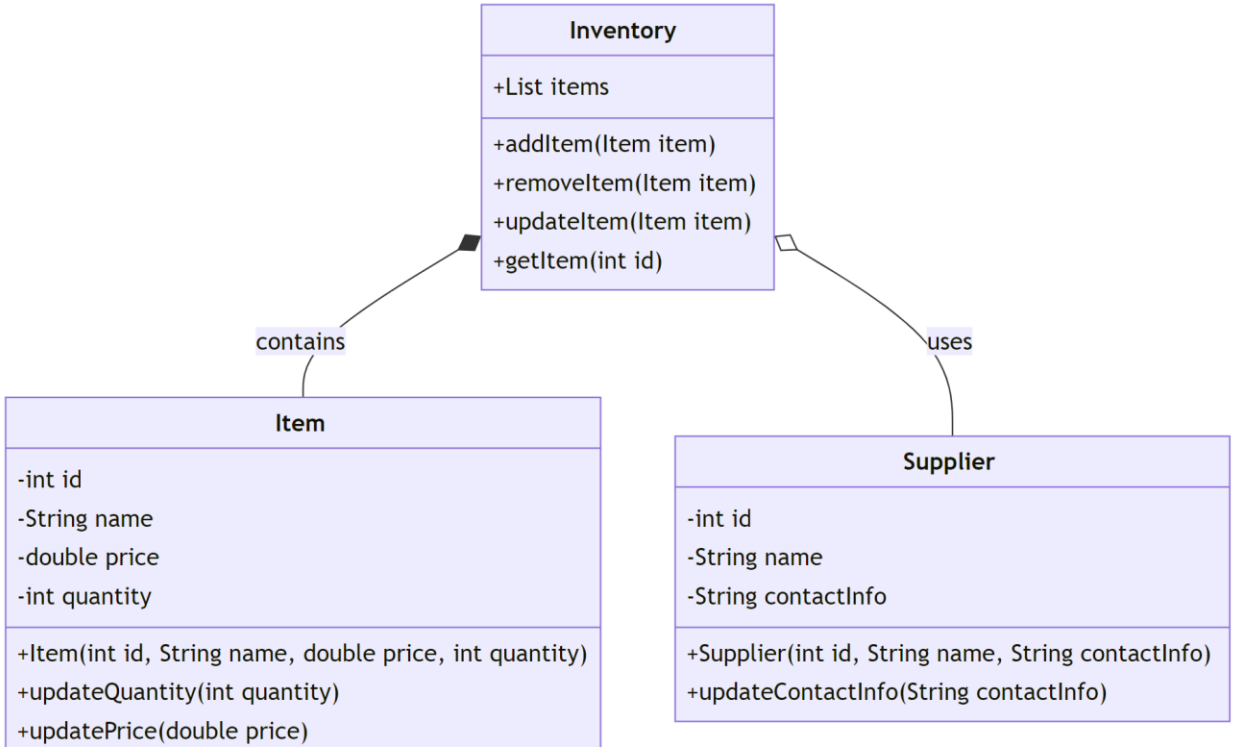
1.3.3 Employee Management Component:



Classes: Employee, Schedule, Role.

Relationships: Employee has a Role and is linked to a Schedule.

1.3.4 Inventory Management Component:



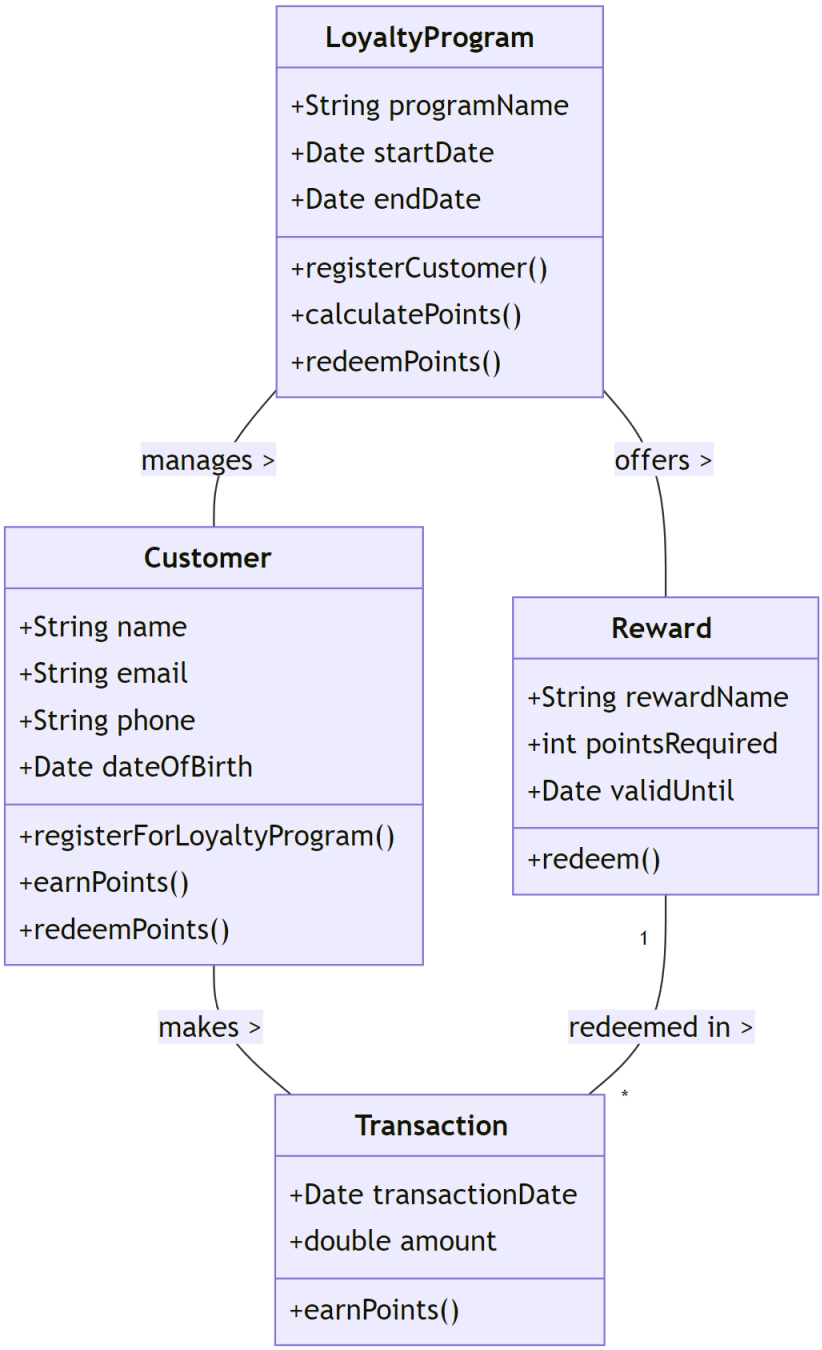
Classes: Product, Inventory, Supplier.

Relationships: Product is managed within Inventory, and Inventory interacts with Supplier.

- **Inventory**: Manages the list of items, with methods to add, remove, update, and retrieve items.
- **Item**: Represents an individual item in the inventory, with attributes like id, name, price, and quantity, and methods to update quantity and price.
- **Supplier**: Represents suppliers with attributes like id, name, and contact information, and a method to update contact information.

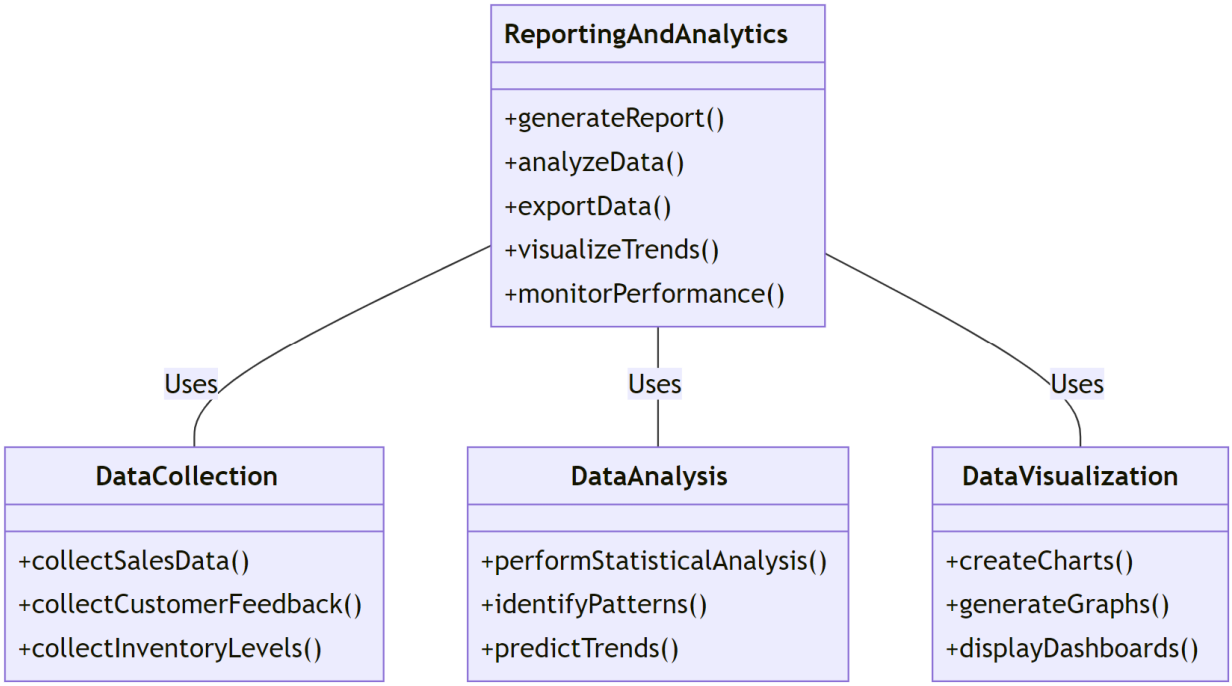
The diagram shows a composition relationship between Inventory and Item, indicating that Inventory contains Items. There's also an aggregation relationship between Inventory and Supplier, indicating that Inventory uses Suppliers but doesn't own them.

1.3.5 Loyalty Program Management Component



- **LoyaltyProgram**: Manages customers and offers rewards.
- **Customer**: Can register for the loyalty program, earn and redeem points.
- **Reward**: Details of rewards that can be redeemed with points.
- **Transaction**: Represents customer transactions that earn points.

1.3.6 Reporting and Analytics Component (nėra būtinas, geriau nedėt):



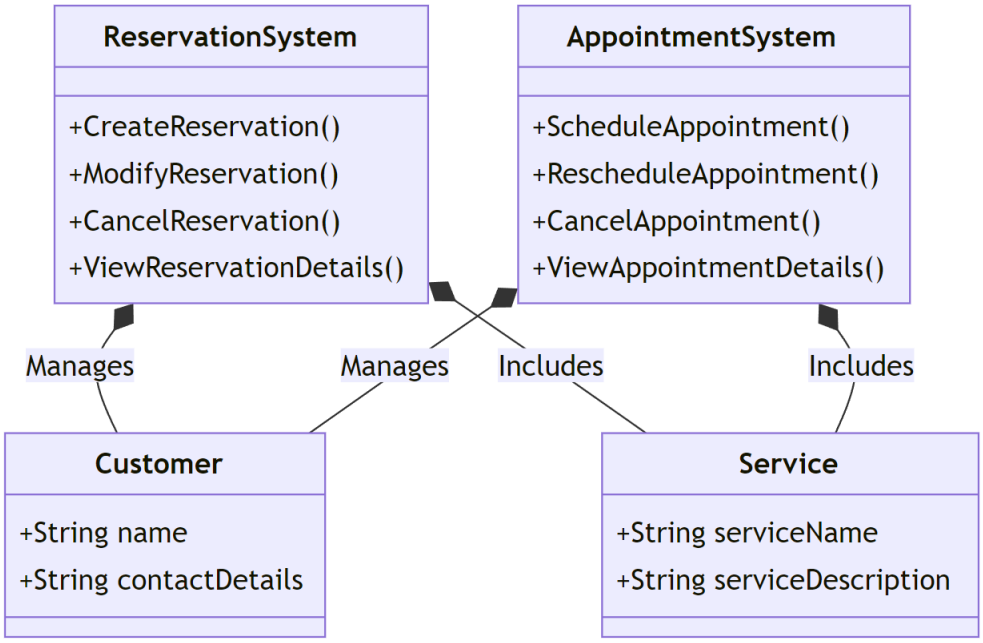
Classes: Report, Analytics, DataAnalysis.

Relationships: Report and Analytics should both contribute to DataAnalysis.

- **ReportingAndAnalytics**: This class includes methods like generating reports, analyzing data, exporting data, visualizing trends, and monitoring performance.
- **DataCollection**: This class is responsible for collecting various types of data such as sales data, customer feedback, and inventory levels.
- **DataAnalysis**: This class performs statistical analysis, identifies patterns, and predicts trends.
- **DataVisualization**: This class creates charts, generates graphs, and displays dashboards.

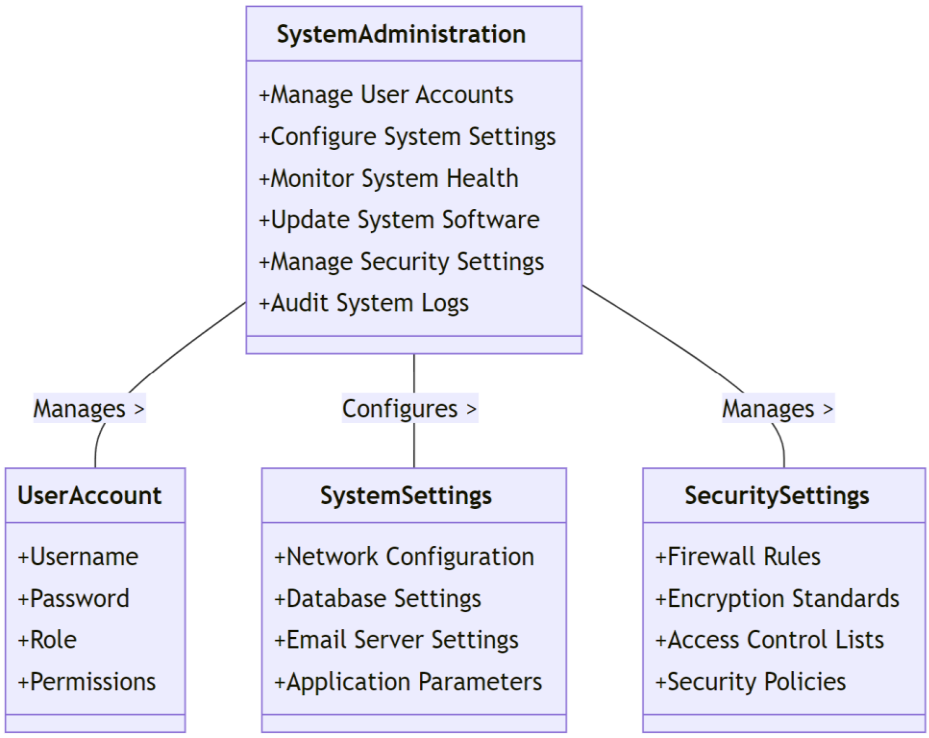
The ReportingAndAnalytics class uses the functionalities of DataCollection, DataAnalysis, and DataVisualization.

1.3.7 Reservation and Appointment Management Component (discuss again)



- **Reservation**: Manages booking details for customers, including scheduling and modifying reservations.
- **Appointment**: Handles specific appointment scheduling, including time, service, and customer preferences.
- **Customer**: Represents individuals making reservations or appointments, storing their details and history.
- **Calendar**: Integrates with reservations and appointments to manage and track scheduling, ensuring no conflicts.

1.3.8 System Administration Component (logs – hostinger/azure, part of system environment, System settings better without configuring,):



Classes: SystemHealth, Maintenance, Security.

Relationships: SystemHealth is monitored through Maintenance activities and is protected by Security measure

1. **SystemHealth Class**: This class is likely responsible for monitoring the overall health of the PoS system. It could include attributes and methods related to system performance metrics, error logs, and status indicators.
2. **Maintenance Class**: This class probably deals with the maintenance activities of the system. It might include functions for scheduling maintenance tasks, logging maintenance activities, and tracking the status of ongoing maintenance work.
3. **Security Class**: The Security class is crucial for managing the security aspects of the PoS system. It could encompass features like access control, encryption methods, security protocols, and breach detection mechanisms.
4. **Relationships**:
    - **SystemHealth and Maintenance**: There is likely a direct relationship between system health and maintenance activities. Maintenance actions could influence the system's health, and the system's health status might trigger specific maintenance tasks.
    - **SystemHealth and Security**: The relationship between system health and security indicates that the overall health of the system includes its security status. Security

breaches or vulnerabilities could impact the system's health, and the system's health monitoring might include security checks.

2. API Endpoints

2.1. Endpoint Descriptions

This section outlines the specific API endpoints for each system component, detailing their functions and purposes.

2.1.1. Transaction Management Component

- POST /transactions Initiates a new transaction.
- PATCH /transactions/{id}/ Modifies items within a transaction.
- --POST /transactions/{id}/ Finalizes a transaction.
- DELETE /transactions/{id} Deletes transaction.
- GET /transactions/{id}/ Retrieves transaction details, including receipt generation.

2.1.2. Inventory Management Component

- GET /inventory/ Lists all inventory items.
- DELETE /inventory/{id} Deletes item from inventory.
- POST /inventory/ Registers a new inventory item.
- PATCH /inventory /{id}: Updates existing inventory item details.
- GET /inventory/{id} retrieves transaction details.
- GET /suppliers Displays a list of all suppliers.
- GET /suppliers/{id} Gets information about a single supplier.
- DELETE /suppliers/{id} Deletes supplier.
- POST / suppliers Registers a new supplier.

2.1.3. Employee Management Component

- GET /employees: Displays all employee records.
- POST /employees: Registers a new employee.
- PATCH /employees/{id}: Updates specific employee details.
- GET /employee/{id}: Retrieves an employee's details.
- DELETE /employee/{id} Deletes employee.

2.1.4. Reporting and Analytics Component

- GET /analytics/reports: Accesses various reports.
- GET /analytics/data: Conducts data analysis for insights.

2.1.5. System Administration Component

- GET /system/overview: Provides an overview of system status.
- POST /system/maintenance/schedule: Schedules system maintenance activities.

2.1.6. Customer Management Component

- GET /customers Accesses customer profiles.
- POST /customers Creates a new customer profile.
- PATCH /customers/{id}: Updates an existing customer profile.
- DELETE /customers/{id}: Deletes customer.
- POST /customers/{id}/enroll: Enrolls a customer in the loyalty program.
- POST /customers/{id}/remove: Removes customer from the loyalty program.

2.2. Implementation Definition

The API endpoints are designed to effectively implement the functionalities identified in the user stories and business processes. Each component has specific endpoints that cater to its operational needs.

2.3. Logical Grouping

For better clarity and organization, the endpoints are grouped logically according to their respective components:

- Transaction Operations Group: Includes all endpoints related to transaction management.
- Inventory Control Group: Encompasses endpoints for managing inventory and supplier relations.
- Employee Administration Group: Contains endpoints for handling employee data and work schedules.
- Data Analysis and Reporting Group: Groups endpoints for analytics and report generation.
- System Oversight Group: Comprises endpoints for system monitoring and maintenance.
- Customer Relations Group: Involves endpoints for managing customer interactions and loyalty programs.
- Security and Regulatory Group: Includes endpoints focused on security and compliance monitoring.

3. Non-Functional Requirements

3.1. Performance

- Scalability:
    - NFR 1: The system should efficiently handle a 50% increase in transaction volume without performance degradation.
    - NFR 2: It should be able to scale horizontally to accommodate growing data and user base.
- Capacity Planning:
    - NFR 3: Conduct bi-annual capacity reviews to anticipate and address peak periods, ensuring the system can handle at least twice the average daily transaction volume.
- Availability:
    - NFR 4: Aim for 99.99% uptime, with maintenance windows scheduled outside of peak business hours.

3.2. Legal Requirements

- Security:
    - NFR 5: Implement AES 256-bit encryption for data storage and TLS 1.2 (or higher) for data transmission.
    - NFR 6: Regularly update to comply with the latest security standards.
- Data Integrity:
    - NFR 7: Implement transaction logging and regular database integrity checks.
    - NFR 8: Ensure data consistency with ACID (Atomicity, Consistency, Isolation, Durability) compliance in transactions.
- Regulatory Compliance:
    - NFR 9: Comply with GDPR for data privacy and PCI DSS for payment processing.
    - NFR 10: Conduct annual compliance audits.

3.3. Usage Context

- Environmental:
    - NFR 11: The software should be operable in temperatures ranging from 0°C to 50°C.
    - NFR 12: Hardware components should be rated for high durability in retail environments.
- Interoperability:
    - NFR 13: Ensure API compatibility with major accounting software (e.g., QuickBooks, Xero) and support standard POS hardware interfaces.

3.4. User Experience

- Utility:
    - NFR 14: Include key functionalities such as multi-method payment processing, inventory tracking, and customer management.
- Usability:
    - NFR 15: Design the interface to ensure a maximum three-step process for common transactions.
    - NFR 16: Provide customizable UI themes for accessibility.
- Manageability:
    - NFR 17: Include remote management capabilities for system updates and troubleshooting.
- Accessibility:
    - NFR 18: Ensure the system is compliant with WCAG 2.1 accessibility standards, offering features like screen reader compatibility and alternative text for images.

3.5. Infrastructure and Deployment

- Reliability:
    - NFR 19: Target a mean time between failures (MTBF) of 10,000 hours for system hardware.
    - **System observability dashboard, etc**
- Serviceability:
    - NFR 20: Implement a logging system capturing detailed information for errors and provide automated diagnostic tools for common issues.
- Recoverability:
    - NFR 21: Daily data backups with a recovery point objective (RPO) of no more than 15 minutes and a recovery time objective (RTO) of no more than 1 hour.
- Deployment Strategy:
    - NFR 22: Support both on-premises and cloud-based deployment options. Include containerization support for easy deployment and scaling.

4. Scalability

4.1. User Expectations

- **Estimated User Numbers**: The system is designed to support up to 10,000 active users concurrently, covering a range of users including customers, administrators, waiters, chefs, specialists, and receptionists.

4.2. Geographic Usage

- **Scope**: Initially targeting the domestic market, the system's architecture is adaptable for future expansion to international markets. It's built to support multiple time zones and regional settings, accommodating different geographic locations.

4.3. Internationalization (i18n)

- **Language Support**: Initially available in English, the system's architecture is flexible to include additional languages in the future.
- **Localization**: Designed to be adaptable to local tax laws, currency formats, and date/time formats, which can be customized as per regional requirements.

4.4. System Availability

- **Operation Hours**: The system is available 24/7, ensuring real-time functionality during peak business hours and accommodating the varied operational hours of different businesses (restaurants, beauty salons, etc.).
- **Maintenance Windows**: Scheduled during low-traffic periods, with prior notification to users, to minimize business impact.

4.5. Legal Aspects

- **Data Handling**: Complies with local data protection laws, ensuring customer consent for data collection and usage.
- **Identification Numbers**: Adheres to privacy regulations, especially in handling personal identification numbers.

4.6. Usage Patterns

- **Peak Times Analysis**: The system is designed to handle peak transaction volumes (up to a 300% increase) during weekends, holidays, and special promotions in restaurants, beauty salons, and other businesses.

4.7. Read/Write Ratios

- **Transaction-Heavy Operations**: Anticipates a higher ratio of write operations due to frequent transaction processing.
- **Reporting and Analytics**: Focuses on read operations, particularly during off-peak hours for data analysis, report generation, and business intelligence insights.

4.8. Platform Focus

- **Multi-Platform Orientation**: While prioritizing a robust desktop interface for comprehensive management, the system architecture also supports a mobile-first approach for specific

functionalities like inventory management, order processing, and reporting, catering to the on-the-go needs of staff and managers.

4.9. External System Dependencies

- **Payment Gateways**: Integrates with major payment processing services to support various payment methods (cash, card, gift vouchers, etc.).
- **Third-Party APIs**: Relies on external APIs for enhanced functionalities, such as advanced analytics, supplementary inventory data, customer relationship management tools, and integration with accounting software like QuickBooks or Xero.