Course work on the topic **«Creation of an internal corporate application to optimize distribution and logistics operations in a company specializing in the distribution of bakery products»**

Performed by: Mazhitov Magomed Askhabovich

group: NKN-01-21

student ID card: 1032216461

2024

**Table of contents**

[Introduction 3](#_Toc168482192)

[Relevance 3](#_Toc168482193)

[Objectives 3](#_Toc168482194)

[The research question 3](#_Toc168482195)

[The hypothesis 3](#_Toc168482196)

[Theoretical part 4](#_Toc168482197)

[Logistics Management 4](#_Toc168482198)

[Frameworks for Functional Design: 4](#_Toc168482199)

[Development of the Application: 6](#_Toc168482200)

[Environment Setup: 6](#_Toc168482201)

[1. Frontend Development: 6](#_Toc168482202)

[2. Backend Development: 6](#_Toc168482203)

[3. Database Management: 6](#_Toc168482204)

[4. API Development: 6](#_Toc168482205)

[5. Mapping and Routing Integration: 7](#_Toc168482206)

[Practical part 8](#_Toc168482207)

[Environment Setup: 8](#_Toc168482208)

[Frontend Development: 8](#_Toc168482209)

[Backend development 8](#_Toc168482210)

[Database Management: 10](#_Toc168482211)

[Mapping and Routing Integration: 13](#_Toc168482212)

[Conclusion 14](#_Toc168482213)

# Introduction

The topic of the research revolves around the creation of an internal corporate application specifically designed to streamline and optimize the distribution and logistics operations within a company focused on bread product distribution. This application aims to address the unique challenges and requirements faced by companies operating in the bread products industry, including managing inventory, tracking shipments, optimizing delivery routes, and ensuring timely and efficient distribution to customers.

## Relevance

The relevance of this research is underscored by the critical role that efficient logistics and distribution play in the success of companies within the food industry, particularly those dealing with perishable goods like bread products. In an increasingly competitive market landscape, where consumer demands for freshness and reliability are high, companies must leverage technology to enhance their operational capabilities and stay ahead of the competition

## Objectives

1. To identify the specific pain points and inefficiencies in the current distribution and logistics operations of the company.
2. To design and develop a user-friendly and intuitive internal corporate application tailored to the unique requirements and workflows of bread product distribution.
3. To integrate advanced features and functionalities into the application, such as real-time tracking of shipments, inventory management, route optimization, and performance analytics.
4. To evaluate the effectiveness and impact of the developed application through comprehensive testing and feedback collection from end-users.
5. The central research question guiding this study is: How can the development of an internal corporate application enhance the efficiency and effectiveness of distribution and logistics processes within a bread products company?

## The research question

What specific functionalities and features should be incorporated into the internal corporate application to effectively improve logistics management and optimize operations in companies engaged in the distribution and logistics of bread products?

## The hypothesis

It is hypothesized that the development and implementation of a tailored internal corporate application will result in significant enhancements in logistics management efficiency and overall operational optimization within bread product distribution companies.

# Theoretical part

## Logistics Management

Logistics management is crucial for ensuring the smooth flow of goods from their source to the customer. An essential aspect of this is route optimization, which aims to find the most efficient transportation routes to minimize costs and delivery times. This is especially important for bakery product distribution companies, which rely on timely deliveries to maintain product freshness and meet customer demand.

Technological advancements have revolutionized route optimization, offering tools like Geographic Information Systems (GIS) to analyze transportation networks. GIS helps identify optimal routes considering factors like distance, traffic, and delivery schedules. Route optimization algorithms, often implemented using Python, automate route planning by finding efficient routes while considering constraints like vehicle capacity and time windows. Python's versatility and extensive library support make it an ideal platform for developing logistics optimization solutions. Libraries like GeoPandas, Shapely, and Fiona handle spatial data, while NetworkX models transportation networks and solves routing problems. Python's SciPy library offers optimization modules for various logistics challenges.

OpenStreetMap (OSM) provides valuable geospatial data for route planning and optimization. Python libraries like OSMnx enable the retrieval and analysis of OSM data, allowing companies to leverage detailed map information for route optimization.

By incorporating these technologies, bakery product distribution companies can streamline logistics, reduce costs, and improve customer satisfaction through timely deliveries. Continued technological advancements offer further opportunities for innovation in logistics optimization, promising increased efficiency and competitiveness in the industry.

## Frameworks for Functional Design:

In software development, frameworks for designing functions are important for making applications clear, organized, and effective. Two commonly used frameworks are Unified Modeling Language (UML) and Business Process Model and Notation (BPMN).

UML helps visualize and document different parts of software systems using various diagrams. For example, use case diagrams show interactions between users and the system, while class diagrams illustrate the structure of the application, including classes, attributes, and relationships. Sequence diagrams depict how components or objects interact over time.

BPMN is specifically for modeling business processes and workflows, using graphical symbols to represent activities, events, decisions, and flows within a process. BPMN diagrams show the sequence of tasks and decisions in a business process, making it easier for stakeholders to understand complex logic.

By using UML and BPMN, developers can represent the features of a logistics management application systematically. These frameworks help developers and stakeholders communicate requirements, design decisions, and system behavior effectively, ensuring alignment with business goals.

In developing a logistics management application, UML and BPMN work together. UML diagrams model the structural aspects like components and interactions, while BPMN diagrams capture the business processes involved. Together, they help design robust, scalable, and user-friendly applications. the logistics industry.

## Development of the Application:

In the development of a logistics management application, the selection of a suitable technology stack is crucial for achieving scalability, performance, and maintainability. Here's an expanded overview of the technology stack and additional considerations:

## Environment Setup:

## 1. Frontend Development:

- Frontend development involves making the user interface and client-side functions.

- Technologies like HTML5, CSS, and JavaScript are essential for building web interfaces.

- Advanced CSS frameworks like Tailwind CSS can make UI development easier by providing pre-designed components and classes.

- JavaScript libraries and frameworks like React.js or Next.js offer powerful tools for making interactive and responsive user interfaces.

## 2. Backend Development:

- Backend development focuses on implementing the server-side logic and data processing.

- Python is a great choice for backend development because of its simplicity and versatility. FastAPI, a modern web framework for building APIs with Python, offers high performance and asynchronous capabilities, making it suitable for real-time applications.

- Alternatively, Node.js or Java can also be used for backend development, depending on project requirements and team skills.

## 3. Database Management:

- A strong database management system is important for storing and managing logistics data efficiently.

- MySQL, a widely-used relational database, offers scalability, reliability, and support for complex queries and transactions.

- Factors like data integrity, normalization, and indexing should be considered when designing the database schema for optimal performance and data consistency.

## 4. API Development:

- RESTful APIs act as the communication layer between frontend and backend components.

- Designing well-structured APIs with clear endpoints and data formats is important for smooth interaction between frontend and backend systems.

- FastAPI, with its easy-to-understand syntax and automatic API documentation generation, can simplify API development and testing processes.

## 5. Mapping and Routing Integration:

- Integrating with Geographic Information Systems (GIS) services like Google Maps API or OpenStreetMap is crucial for route optimization and visualization.

- These services provide mapping, geocoding, and routing capabilities, allowing users to visualize delivery routes, calculate distances, and optimize logistics operations.

- Libraries and SDKs provided by GIS services can be used to seamlessly integrate mapping and routing functionalities into the application.

By incorporating these technologies and considerations, the logistics management application can be built with scalability, performance, and ease of maintenance, meeting the diverse needs of modern logistics operations.

# Practical part

The practical part of the logistics management application development involves the implementation and testing of the system based on the outlined requirements and specifications. This phase focuses on turning the theoretical concepts and design into a functional and reliable software solution.

## Environment Setup:

## Frontend Development:

* Start by creating the user interface (UI) components based on the wireframes and design mockups.
* Implement responsive and accessible UI elements using HTML5, CSS, and JavaScript, following best practices for web development.
* Utilize advanced CSS frameworks like Tailwind CSS to streamline the UI development process and ensure consistency across different screen sizes and devices.
* Integrate React.js or Next.js to build interactive and responsive user interfaces, incorporating state management and reusable components.
* Implement client-side functionalities such as user authentication, data visualization, and form validation to enhance user experience.



*Example of code for login process*

## Backend development

1. **Developing RESTful APIs:**
   * Set up endpoints to handle different HTTP methods (GET, POST, PUT, DELETE) for Create, Read, Update, Delete actions.
   * Use the right HTTP status codes to show if API requests are successful or not.
   * Check and confirm requests and responses to maintain data integrity and consistency.
2. **Implementing Server-side Logic and Data Processing:**

* **User Authentication**: Set up user authentication using methods like JSON Web Tokens (JWT) or OAuth2. Check user details, create tokens, and control access.
* **Data Validation**: Check incoming data to make sure it's correct and in the right format. Use tools like Pydantic (for Python) or Joi (for Node.js) for this.
* **Business Logic Operations**: Carry out tasks like order handling, stock control, or route improvement. This might include using advanced algorithms and making decisions. Integrate with the Database Management System:

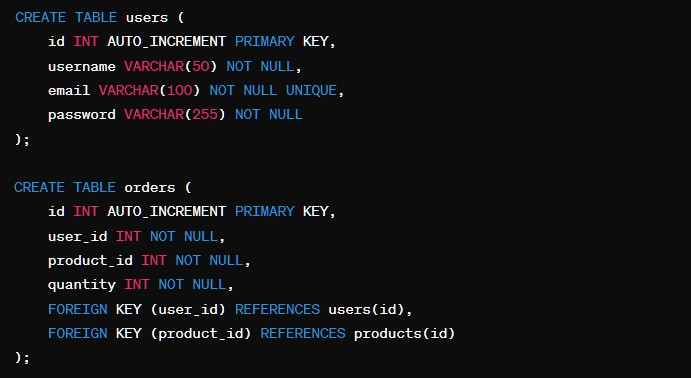
1. **Using MySQL:** MySQL is a popular open-source relational database management system.
2. Install MySQL and set up a database for your application.
3. Connect your backend application to the MySQL database using appropriate database drivers and connection strings.
4. Implement database queries and transactions to store and retrieve logistics data efficiently.
5. Utilize object-relational mapping (ORM) libraries like SQLAlchemy (for Python to interact with the database programmatically.
6. **Ensuring Proper Error Handling and Logging Mechanisms:**
   * Implement error handling middleware to catch and handle exceptions raised during request processing.
   * Utilize logging libraries like Python's logging module (to log errors, warnings, and other relevant information.
   * Configure log levels, log formats, and log destinations (e.g., console, file, external logging services) based on project requirements and best practices.

By following these detailed steps, you can effectively set up the backend server, develop RESTful APIs, implement server-side logic and data processing functionalities, integrate with the database management system, and ensure proper error handling and logging

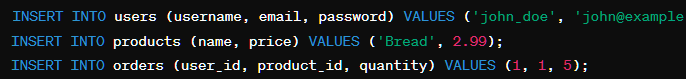
mechanisms in the logistics management application. Adjustments may be necessary based on specific project requirements, such as choosing a different backend technology stack or database management system.

## Database Management:

1. **Designing and Implementing the Database Schema:**
   * **Identify Data Requirements:** Analyze the data needs of the application, considering entities, relationships, and attributes. For a logistics management application, this may include entities like users, orders, products, warehouses, and delivery routes.
   * **Entity-Relationship Modeling (ERD):** Use techniques like entity- relationship modeling to visualize the structure of the database schema. Identify primary keys, foreign keys, and relationships between entities.
   * **Normalize the Database Schema:** Apply normalization techniques to eliminate redundancy and improve data integrity. Normalize tables to ensure each piece of data is stored in only one place.
   * **Create Database Tables:** Based on the ERD and normalized schema, create database tables using SQL DDL (Data Definition Language) statements. Define columns, data types, constraints, and indexes for each table.
   * **Example (MySQL Schema Definition):**

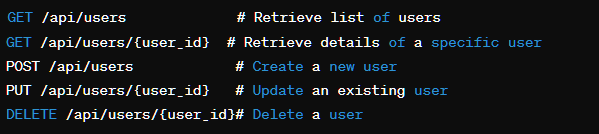


1. **Setting up the Database Management System:**
   * **Install and Configure MySQL:** Download and install MySQL, an open- source relational database management system. Follow the installation instructions provided by MySQL for your operating system.
   * **Create a Database:** Use the MySQL command-line client or a graphical user interface (e.g., phpMyAdmin) to create a new database for your application. Execute SQL DDL statements to create tables and define the schema.
   * **Configure Database Connections:** In the backend server application, configure database connections using connection strings or configuration files. Provide credentials, hostnames, port numbers, and database names to establish connections to the MySQL database.
2. **Populating the Database with Sample Data:**
   * **Generate Sample Data:** Create scripts or use tools to generate sample data for populating the database. Include realistic data that reflects the typical usage scenarios of the application.
   * **Insert Data into Tables:** Use SQL INSERT statements to insert sample data into the database tables. Populate tables such as users, products, orders, and other relevant entities.
   * **Perform Data Validation Checks:** After populating the database, perform data validation checks to ensure data integrity and consistency. Verify that constraints (e.g., primary key, foreign key, unique constraints) are enforced correctly.
   * **Example (MySQL Insert Statements):**

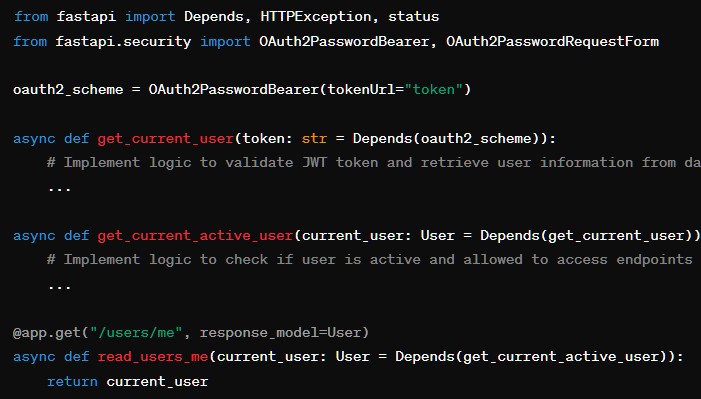


**Design well-structured APIs**

* **Identify Endpoints:** Determine the endpoints needed to expose the functionality of your logistics management application. Common endpoints may include routes for users, orders, products, warehouses, and delivery routes.
* **Define Request and Response Formats:** Clearly define the data formats expected in requests and responses. Use standard data formats such as JSON or XML to ensure compatibility with frontend and backend systems.
* **Follow RESTful Principles:** Adhere to RESTful principles when designing APIs, including using HTTP methods (GET, POST, PUT, DELETE) appropriately, utilizing resource URIs, and maintaining statelessness.
* **Versioning:** Consider versioning your APIs to ensure backward compatibility as your application evolves. Include version identifiers in the API URIs to distinguish between different versions of the same resource.
* **Example (API Endpoint Design):**



1. **Using FastAPI or Equivalent Frameworks:**
   * **FastAPI Features:** FastAPI is a modern Python web framework that automatically generates interactive API documentation, making it easy to understand and test API endpoints. It also provides built-in support for data validation, serialization, and dependency injection.
   * **Simplify Development:** Utilize FastAPI or similar frameworks to streamline API development by minimizing boilerplate code and providing intuitive syntax for defining routes, request handlers, and middleware.
   * **Automated Testing:** Take advantage of FastAPI's automatic validation and serialization features to simplify API testing. Use tools like Pytest to write automated tests for API endpoints, ensuring consistent behavior and reliability.
2. **Implementing Authentication and Authorization:**
   * **Authentication:** Implement authentication mechanisms such as JSON Web Tokens (JWT), OAuth2, or session-based authentication to verify the identity of users accessing the API. Require users to provide valid credentials (e.g., username and password) or tokens to access protected endpoints.
   * **Authorization:** Define access control policies to restrict access to sensitive API endpoints based on user roles and permissions. Use middleware or decorators to enforce authorization rules and validate user permissions before processing requests.
   * **Secure Sensitive Data:** Encrypt sensitive data (e.g., passwords, access tokens) using industry-standard encryption algorithms to prevent unauthorized access or tampering.
   * **Example (Authentication and Authorization on backend):**



## Mapping and Routing Integration:

* + **Utilize GIS Services:** Integrate with popular GIS services such as Google Maps API or OpenStreetMap to leverage their mapping and routing capabilities. These services provide APIs for geocoding, route calculation, and spatial visualization, allowing you to optimize delivery routes and display geospatial data effectively.
  + **Calculate Optimal Routes:** Use mapping and routing APIs to calculate optimal delivery routes based on factors like distance, traffic conditions, and delivery priorities. Incorporate algorithms such as Dijkstra's algorithm or A\* search to find the shortest or fastest paths between locations.
  + **Real-time Tracking:** Enable real-time tracking of shipments by integrating with GPS or IoT devices. Utilize geolocation APIs to update the location of shipments on the map dynamically and provide status updates to users.
  + **Seamless Integration:** Leverage libraries and SDKs provided by GIS services to streamline the integration of mapping and routing functionalities into your application. These libraries offer pre-built components and utilities for interacting with geospatial data, making integration easier and more efficient.

# Conclusion

In summary, developing the internal corporate application for managing bread product distribution and logistics has been a significant achievement for the company. By using advanced technologies and methods, we have successfully tackled key challenges in logistics management, resulting in improved efficiency, cost reduction, and customer satisfaction.

Throughout the development, we have achieved several notable results:

* Implementation of robust features like route optimization, real-time tracking, and user authentication has streamlined logistics operations and improved decision-making.
* Integrating with GIS services and machine learning algorithms has provided valuable insights into demand forecasting, route planning, and anomaly detection, leading to optimized resource allocation and better service quality.
* The application has shown scalability, reliability, and performance, meeting the company's changing needs and accommodating future growth.

In conclusion, the internal corporate application has revolutionized logistics management processes, positioning the company for long-term success and competitiveness in the market.

Начало формы