Тз проекта  
  
Разработка ПО для сети кофеен “Вкусный кофе”. Разработка и оптимизация ПО, функциями которого является онлайн-касса и онлай-склад.   
  
**Структура ПО для онлайн-кассы ресторана:**

1. **Фронтенд:** предоставляет графический интерфейс для пользователей, включая кассиров, администраторов и менеджеров ресторана. Может быть реализован в виде веб-приложения или мобильного приложения.
2. **Бэкенд:** предоставляет серверную логику и функциональность для работы с данными и взаимодействия с другими сервисами. Может быть реализован с использованием микросервисной архитектуры.
3. **База данных:** предоставляет хранилище для данных о товарах, продажах, остатках на складе, отчетах и т.д. Может быть реализована с использованием реляционных или нереляционных СУБД.
4. **API:** предоставляет интерфейс для взаимодействия между фронтендом, бэкендом и другими сервисами. Может быть реализован с использованием REST, gRPC или других протоколов.
5. **Интеграция с внешними системами:** предоставляет функциональность интеграции с платежными системами, бухгалтерской системой, системами доставки и другими внешними сервисами. Может быть реализована с использованием API этих систем.
6. **Безопасность:** предоставляет функциональность защиты данных и доступа к ним, включая аутентификацию, авторизацию, шифрование и т.д. Может быть реализована с использованием стандартов безопасности, таких как OAuth 2.0, OpenID Connect, SSL/TLS и т.д.
7. **Мониторинг и логирование:** предоставляет функциональность мониторинга работы системы, сбора статистики и логирования событий. Может быть реализована с использованием систем мониторинга и логирования, таких как Prometheus, Grafana, ELK Stack и т.д.
8. **Развертывание и масштабирование:** предоставляет функциональность развертывания и масштабирования системы, включая развертывание в облачных средах, автоматическое масштабирование, контейнеризацию и т.д. Может быть реализована с использованием систем развертывания и масштабирования, таких как Kubernetes, Docker, AWS, Azure и т.д.

**Требования к структуре ПО:**

1. Фронтенд должен предоставлять интуитивно понятный и удобный интерфейс для пользователей.
2. Бэкенд должен предоставлять надежную и эффективную серверную логику и функциональность.
3. База данных должна обеспечивать надежное и эффективное хранение и обработку данных.
4. API должен предоставлять надежный и эффективный интерфейс для взаимодействия между фронтендом, бэкендом и другими сервисами.
5. Интеграция с внешними системами должна обеспечивать надежную и эффективную работу с внешними сервисами.
6. Безопасность должна обеспечивать надежную защиту данных и доступа к ним.
7. Мониторинг и логирование должны обеспечивать надежный и эффективный контроль работы системы.
8. Развертывание и масштабирование должны обеспечивать надежную и эффективную работу системы в различных средах и при различных нагрузках.

**Структура онлайн-склада для онлайн-кассы ресторана:**

1. **Каталог товаров:** предоставляет информацию о товарах, включая название, описание, цену, единицу измерения, категорию и др. Характеристики товаров. Может быть реализован в виде таблицы в базе данных.
2. **Складские остатки:** предоставляет информацию о количестве товаров на складе, включая текущее количество, минимальное количество для заказа и максимальное количество для хранения. Может быть реализован в виде таблицы в базе данных.
3. **Поступления товаров:** предоставляет информацию о поступлениях товаров на склад, включая дату поступления, количество, поставщика, цену и др. Характеристики поступления. Может быть реализован в виде таблицы в базе данных.
4. **Расходы товаров:** предоставляет информацию о расходах товаров со склада, включая дату расхода, количество, причину расхода (продажа, брак, перемещение и т.д.) и др. Характеристики расхода. Может быть реализован в виде таблицы в базе данных
5. **Перемещения товаров:** предоставляет информацию о перемещениях товаров между складами, включая дату перемещения, количество, причину перемещения и др. Характеристики перемещения. Может быть реализован в виде таблицы в базе данных.
6. **Заказы товаров:** предоставляет информацию о заказах товаров у поставщиков, включая дату заказа, количество, поставщика, цену и др. Характеристики заказа. Может быть реализован в виде таблицы в базе данных.
7. **Отчеты о товарах:** предоставляет отчеты о товарах, включая отчет о текущих остатках, отчет о поступлениях, отчет о расходах, отчет о перемещениях и отчет о заказах. Может быть реализован в виде отчетов, генерируемых из базы данных.
8. **Интеграция с онлайн-кассой:** предоставляет интеграцию между онлайн-складом и онлайн-кассой, включая обновление остатков товаров при продаже, обновление цен товаров в онлайн-кассе при изменении цены на складе и др. Функции. Может быть реализована с использованием API онлайн-кассы.
9. **Интеграция с внешними системами:** предоставляет интеграцию между онлайн-складом и внешними системами, включая интеграцию с системами управления закупками, системами управления поставками, системами бухгалтерского учета и др. Может быть реализована с использованием API этих систем.

**Требования к структуре онлайн-склада:**

1. Каталог товаров должен предоставлять полную и точную информацию о товарах.
2. Складские остатки должны предоставлять точную информацию о количестве товаров на складе.
3. Поступления товаров должны предоставлять полную и точную информацию о поступлениях товаров на склад.
4. Расходы товаров должны предоставлять полную и точную информацию о расходах товаров со склада.
5. Перемещения товаров должны предоставлять полную и точную информацию о перемещениях товаров между складами.
6. Заказы товаров должны предоставлять полную и точную информацию о заказах товаров у поставщиков.
7. Отчеты о товарах должны предоставлять полную и точную информацию о товарах в виде отчетов.
8. Интеграция с онлайн-кассой должна обеспечивать надежную и эффективную работу онлайн-склада с онлайн-кассой.
9. Интеграция с внешними системами должна обеспечивать надежную и эффективную работу онлайн-склада с внешними системами.

|  |  |
| --- | --- |
| Функциональные | Нефункциональные требования |
| * Онлайн касса: возможность регистрации продаж, печати чеков, возвратов, работы с безналичными платежами. | Надежность: система должна обеспечивать бесперебойную работу и сохранение данных при возможных сбоях. |
| * Онлайн склад: возможность учета товаров, списания продукции, внесения поставок, формирования отчетов о остатках. | * Безопасность: защита данных от несанкционированного доступа. |
| * Открытие и закрытие смены: возможность учета выручки, формирования отчета о продажах за смену. | * Масштабируемость: возможность расширения функциональности системы в будущем. |
|  | * Интуитивно понятный интерфейс: система должна быть легкодоступна для пользователей с разным уровнем подготовки. |

|  |  |  |
| --- | --- | --- |
| Бизнес требования | Пользовательские требования | Системные требования |
| 1. Автоматизация процессов продаж и учета товаров: для увеличения эффективности работы ресторана и уменьшения риска ошибок при ручном учете. 2. Учет остатков товаров на складе: для своевременного пополнения запасов и предотвращения ситуаций, когда товар отсутствует на складе, но присутствует в меню. 3. Открытие и закрытие смены: для контроля выручки и предотвращения мошенничества со стороны персонала. 4. Интеграция с платежными системами: для возможности оплаты заказа безналичными средствами. 5. Интеграция с бухгалтерской системой: для автоматизации учета и отчетности. | 1. Интуитивно понятный интерфейс: для удобства работы персонала ресторана. 2. Быстрая работа системы: для уменьшения времени ожидания клиентов. 3. Возможность вносить изменения в меню: для гибкости работы ресторана. 4. Возможность отслеживать статистику продаж: для анализа эффективности работы ресторана. 5. Возможность создавать счета и печатать чеки: для удобства клиентов и соответствия требованиям законодательства. | 1. Надежность и стабильность работы системы: для минимизации риска потери данных и сбоев в работе ресторана. 2. Безопасность данных: для защиты конфиденциальной информации клиентов и ресторана. 3. Масштабируемость: для возможности расширения функциональности системы в будущем. 4. Интеграция с внешними системами: для автоматизации процессов и увеличения эффективности работы ресторана. 5. Возможность работы в автономном режиме: для обеспечения работы ресторана в случае отсутствия соединения с интернетом. 6. Возможность работы с несколькими точками продаж: для расширения бизнеса и увеличения выручки. 7. Возможность работы с несколькими типами устройств: для гибкости работы персонала ресторана. 8. Возможность настройки доступа для разных типов пользователей: для контроля доступа к данным и функциям системы. 9. Возможность ведения журнала действий: для контроля и анализа работы персонала ресторана |

| **№ п/п** | **Задача** | **Описание** | **Ответственный** | **Срок выполнения** |
| --- | --- | --- | --- | --- |
| 1 | Анализ требований | Проведение анализа требований к системе, определение функциональных и нефункциональных требований, разработка технического задания | Аналитик | 1 неделя |
| 2 | Создание архитектуры | Разработка архитектуры системы, определение компонентов и взаимодействий между ними, выбор технологий и инструментов разработки | Архитектор | 2 недели |
| 3 | Создание базы данных | Разработка структуры базы данных, определение таблиц и полей, настройка связей между таблицами, создание скриптов для за��олнения тестовыми данными | Разработчик БД | 2 недели |
| 4 | Разработка API | Разработка API для взаимодействия между компонентами системы, определение методов и параметров, создание документации | Backend-разработчик | 3 недели |
| 5 | Разработка онлайн-кассы | Разработка веб-интерфейса онлайн-кассы, реализация функционала регистрации продаж, открытия и закрытия смены, печати чеков | Frontend-разработчик | 4 недели |
| 6 | Разработка онлайн-склада | Разработка веб-интерфейса онлайн-склада, реализация функционала списания продукции и внесения поставок, реализация отчетов о остатках и движении товаров | Frontend-разработчик | 4 недели |
| 7 | Интеграция с платежными системами | Интеграция системы с платежными системами, реализация функционала оплаты счетов онлайн | Backend-разработчик | 2 недели |
| 8 | Тестирование | Проведение функционального и нефункционального тестирования системы, исправление ошибок и недочетов | Тестировщик | 2 недели |
| 9 | Развертывание и настройка | Развертывание системы на сервере, настройка параметров безопасности, оптимизация производительности | DevOps-специалист | 1 неделя |
| 10 | Документирование | Создание пользовательской документации, руководств по эксплуатации и обучающих материалов | Технический писатель | 1 неделя |

**Общий срок выполнения проекта:** 5 месяцев

**Команда разработки:**

* Аналитик
* Архитектор
* Разработчик БД
* Backend-разработчик
* Frontend-разработчик
* Тестировщик
* DevOps-специалист
* Технический писатель