****МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

**Институт №3\_\_\_\_\_\_Кафедра \_\_\_\_\_304 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Направление подготовки \_09.03.04 «Программная инженерия» Группа\_3О-\_**

**Квалификация (степень) \_Бакалавр \_\_\_\_\_\_\_**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

На тему: «Разработка пилотного проекта веб-сервиса для взаимодействия пользователей портала государственных услуг г. Москвы».

Автор квалификационной работы \_ ( )

(Фамилия, Имя, Отчество)

Руководитель ( )

(Фамилия, Имя, Отчество)

Рецензент ( )

(Фамилия, Имя, Отчество)

**К з а щ и т е д о п у с т и т ь**

Зав. кафедрой 304 Брехов О.М. ( )

(Фамилия, Имя, Отчество)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

Москва 2023 г.

# ВВЕДЕНИЕ

Традиционно серверные приложения разрабатывались с использованием монолитного подхода к проектированию архитектуры приложения.

Монолитная архитектура - это подход к разработке приложения, при котором все компоненты приложения находятся в одном исполняемом файле или модуле, при этом приложение реалезуется на одном языке программирования

Это делает процесс разработки, тестирования и развертывания приложения относительно простым и удобным. Все компоненты приложения могут легко общаться между собой, что упрощает обработку запросов и увеличивает производительность.

Однако, монолитная архитектура имеет некоторые недостатки. При увеличении размера приложения становится все сложнее разрабатывать, тестировать и развертывать новые функции.

Другим недостатком монолитной архитектуры является тот факт, что она не является очень гибкой и масштабируемой, что может стать проблемой при увеличении нагрузки на приложение.

В целом, монолитная архитектура является удобным и простым подходом к разработке сервера, но при создании крупных и сложных систем может ограничивать возможности разработчиков и увеличивать затраты на поддержку и развитие приложения.

Альтернативным подходом к разработке приложения является микросервисная архитектура.

Она позволяет создавать сложные системы из небольших, независимых и легко масштабируемых сервисов. Такой подход имеет ряд преимуществ перед монолитной архитектурой, так как позволяет повысить гибкость, масштабируемость и отказоустойчивость системы.

Другим важным аспектом, который будет рассмотрен в данной работе является способ хранения состояния приложения. При традиционном подходе в базе данных сохраняется только последнее состояние приложения, что ведет к потере данных о предыдущих состояниях приложения и невозможности восстановить предыдущее состояние приложения. Для борьбы с этой проблемой существует подход Источники событий, который предполагает хранение истории изменений состояния приложения в виде событий. Каждое изменение состояния приложения записывается в журнал событий в хронологическом порядке. Используя этот журнал, можно воссоздать состояние системы на любой момент времени.

В данной дипломной работе будет рассмотрена микросервисная архитектура вместе с подходом Источники событий на примере разработки сервера для веб-приложения. Будут изучены основные принципы построения микросервисов, архитектурные шаблоны и инструменты, необходимые для создания и управления такой системой. Также будет проведен анализ преимуществ и недостатков данного подхода и рассмотрены сценарии его применения в реальных проектах.

Целью данной работы является изучение и анализ микросервисной архитектуры и хранения состояния приложения в виде событий, а также разработка и реализация сервера с использованием данного подхода.

# ЗАДАЧА ​РАЗРАБОТКИ ПРИЛОЖЕНИЯ-СЕРВЕРА С ИСПОЛЬЗОВАНИЕМ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ И СОБЫТИЙНОГО ПОДХОДА К ХРАНЕНИЮ ДАННЫХ.

* 1. Постановка и анализ задачи
     1. Назначение приложения и предметная область

Разрабатываемое программное обеспечение предназначено для управления рестораном общественного питания и предоставляет функционал необходимый для работы с заказами, их сборки и приготовления.

Предметной областью веб-сервера является сфера услуг общественного питания и отображает ее с точки зрения как работников ресторана, так и его посетителей.

Целью разработки данного приложения является анализ и реализация основных принципов микросервисной архитектуры приложения и так же анализ и реализация событийного подхода к хранению данных, его достоинств и недостатков.

Ставится задача реализации следующего функционала в административной панели управления:

* Создание/редактирование товаров
* Создание/редактирование меню
* Создание, сборка и выдача, отслеживание статусов заказов
* Отслеживание недостающих товаров для сборки заказов
* Отслеживание запасов товаров
* Создание и выполнение запросов на приготовление товаров

Так как пользователи не должны иметь доступ к жизненно важным функциям приложения, то для них нужно реализовать пользовательскую панель управления, к которой предъявляются следующие функциональные требования:

* Просмотр меню
* Создание и отслеживание статусов заказов

Так же для более наглядного отображения событийного подхода к хранению состояния приложения необходимо предусмотреть следующий функционал в административной панели управления:

1. Отображение журнала событий для
   1. Заказов
   2. Товаров
   3. Меню
   4. Запасов товаров
   5. Запросов на приготовление товаров
2. Отображение списка событий, связанных с запрашиваемыми объектами для
   1. Заказов
   2. Товаров
   3. Меню
   4. Запасов товаров
   5. Запросов на приготовление товаров
      1. Выбор средств разработки

Выбор средств разработки основывается на удобстве реализации выбранных подходов и приспособленности средств к поставленным целям и функционалу.

* + - 1. Выбор целевой операционной системы

Операционная система (ОС) - это программное обеспечение, которое управляет аппаратными и программными ресурсами компьютера и обеспечивает работу других приложений на этом компьютере.

ОС обеспечивает интерфейс между пользователем и компьютером, позволяет запускать программы, управляет доступом к ресурсам компьютера, обеспечивает безопасность и защиту данных, управляет памятью, обрабатывает ввод и вывод данных и выполняет множество других функций.

Для разработки сервера решено использовать ОС Windows 10, среди ее преимуществ:

1. Широкое распространение: Windows является одной из самых популярных операционных систем в мире, что делает ее доступной для использования и поддержки.
2. Удобство использования: Windows имеет простой и интуитивно понятный интерфейс, что делает ее легкой в освоении даже для новых пользователей.
3. Большой выбор программного обеспечения: Windows поддерживает большое количество программного обеспечения, включая популярные приложения для офисной работы, развлечений и игр.
4. Поддержка различных устройств: Windows поддерживает большой выбор аппаратного обеспечения, включая настольные компьютеры, ноутбуки, планшеты и смартфоны.
   * + 1. Выбор языка программирования

При разработке сервера стоит обратить внимание на языки программирования, которые наиболее распространены в сфере веб-разработки и обладают большой системой библиотек для уменьшения времени разработки приложения.

Одним из лидеров в сфере вэб-разработки является язык Javascript. Хотя изначально Javascript предназначался для выполнения в браузере, сейчас он может быть выполнен и вне браузера при помощи Node.js.

Node.js - это среда выполнения JavaScript, основанная на движке V8, который используется в браузере Google Chrome. Она позволяет запускать JavaScript-код на сервере, что делает ее популярной для создания масштабируемых и высокопроизводительных веб-приложений. Среди преимуществ Node.js:

1. Масштабируемость: Node.js использует модель событийного цикла, что делает ее масштабируемой и позволяет обрабатывать большое количество запросов одновременно.
2. Удобство разработки: Node.js позволяет использовать JavaScript как на стороне сервера, так и на стороне клиента, что упрощает разработку веб-приложений и уменьшает время разработки.
3. Большое количество модулей: Node.js имеет огромное количество модулей, которые позволяют быстро и удобно добавлять функциональность в приложение.
4. Большое сообщество: Node.js имеет большое и активное сообщество разработчиков, которое предоставляет поддержку и помощь в разработке приложений.