

Тема: «Система заказа лекарств на дом»

Группа: ИСЭбд-41

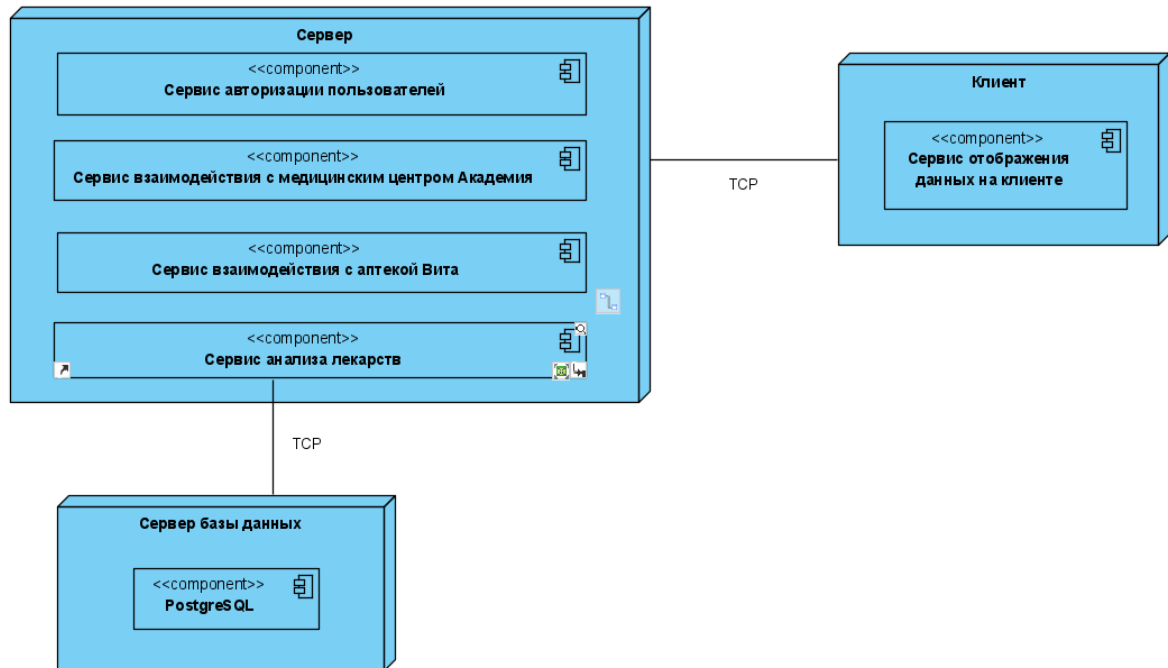
Плетнева М.А.

Реализация задумок СРО

Архитектура приложения – микросервисы. Используя такой подход вместо одного большого приложения, мы создаем набор небольших слабосвязанных и легко заменяемых модулей, которые взаимодействуют друг с другом.

Преимущества микросервисов:

1. *Написание и поддержка небольших сервисов гораздо проще*, чем написание и поддержка монолитного проекта. Каждый элемент системы несет одну функцию и для внесения изменений в этот элемент не нужно перестраивать весь алгоритм.
2. *Распределение работы* по созданию отдельных модулей можно распределить между разными командами.
3. *Существование возможности выбора* для каждого сервиса языка и библиотеки, подходящих для конкретной задачи.
4. *Наиболее легкая и простая возможность проведения обновлений* при микросервисной архитектуре – постепенный процесс не затормозит работу всей системы.
5. *Плавность и бесперебойность работы*: сервисы, которые оказались не нужны можно просто выключать, и это не отразится на работе системы в целом.
6. *Учет очередности*: сервисы можно разрабатывать и внедрять постепенно, в зависимости от очереди потребностей бизнес-подразделений.
7. *Многоканальность*: микросервисы могут работать на любом устройстве, а также в облачных средах.



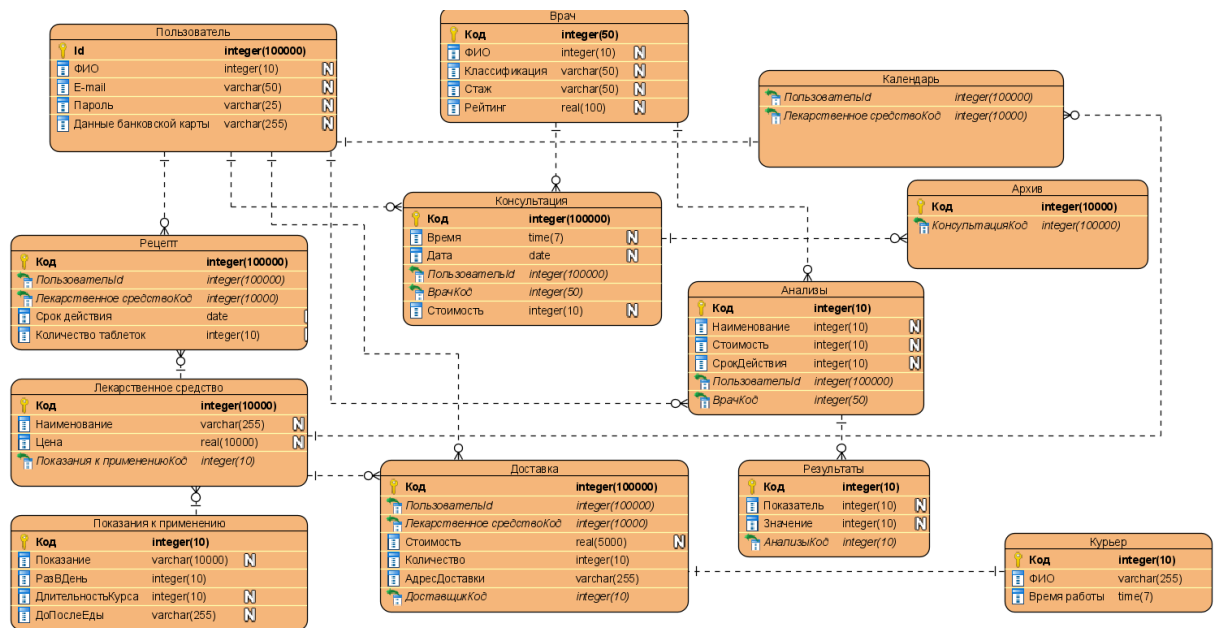
Сервис авторизации пользователей – микросервис, отвечающий за авторизацию и регистрацию пользователей в системе.

Сервис взаимодействия с медицинским центром Академия – микросервис, отвечающий за консультации с терапевтом и узкими специалистами. Также хранит выписку рецептов врачей и заказ анализов на дом, хранение всей истории лечения и анализов в личном кабинете

Сервис взаимодействия с аптекой Вита – микросервис, отвечающий за заказ и доставку лекарств на дом

Сервис взаимодействия лекарств – микросервис, отвечающий за подбор витаминов для текущего времени года и категории людей, напоминание и уведомление о приеме лекарств и консультации у специалистов.

Сущности:



Технологии разработки

Frontend - vue, один из самых популярных в настоящее время фронтенд фреймворков. Он прост и понятен, в сравнении с Angular. Кроме малых размеров, его основными преимуществами являются визуальный DOM, компонентный подход и двустороннее связывание данных.

Плюсы:

- обширная и подробная документация;
- простой синтаксис. Программисты, знающие Javascript, легко разберутся с Vue.js;
- гибкость при разработке структуры приложения;
- поддержка Typescript.

Когда использовать:

Vue.js рекомендуется для гибких конструкций, позволяющих создавать все с нуля.

Конкуренты: React, JQuery, Angular

<https://firecode.ru/blog/luchshie-frontend-freymvorki-2020/>

Bootstrap – инструмент для создания сайтов, включает в себя HTML- и CSS-шаблоны оформления разных компонентов для веб-сайтов: типографику, веб-формы, кнопки, блоки навигации и другое для типографики

Преимущества:

- Уменьшение количества времени на разработку. Bootstrap дает готовые решения: их использование позволяет сократить время, затрачиваемое на создание макета сайта.
- Адаптивность. Bootstrap позволяет создавать адаптивные сайты.

- **Кросс-браузерность.** Сайты, сделанные с использованием Bootstrap, будут одинаково отображаться во всех современных браузерах.
- **Легкость в использовании и быстрота в освоении.** Bootstrap легко использовать в разработке, в нем легко разобраться.
- **Понятный код.** Bootstrap позволяет писать качественный и понятный код, который легко поймет другой разработчик.

Backend - Node.js, представляет собой серверную платформу на JavaScript. На базе платформы Node.js создано множество фреймворков, включая такие популярные, как Express.

Платформа Node.js, и, соответственно, основанные на ней фреймворки, отличаются нетребовательностью к ресурсам и масштабируемостью. Именно поэтому Node.js — это платформа, к которой часто прибегают те, кто пользуется микросервисными архитектурами. Эта платформа хорошо подходит для разработки микросервисов ещё и из-за существования системы модулей Node.js, которые можно представить себе в виде строительных блоков серверных приложений.

- Код клиентских и серверных частей проектов легче поддерживать в согласованном состоянии, так как и там и там используется один и тот же язык.
- Один и тот же код можно совместно использовать и на клиенте, и на сервере.
- Благодаря существованию модулей Node.js, которые, в сущности, представляют собой особым образом оформленные фрагменты кода, разработчики могут с удобством использовать в своих проектах чужой код, а так же собственные наработки.
- Платформа Node.js, и, соответственно, основанные на ней фреймворки, отличаются нетребовательностью к ресурсам и масштабируемостью. Именно поэтому Node.js — это *платформа, к которой часто прибегают те, кто пользуется микросервисными архитектурами.*
- Эта платформа хорошо подходит для разработки микросервисов ещё и из-за *существования системы модулей Node.js*, которые можно представить себе в виде строительных блоков серверных приложений.
- В Node.js JavaScript код компилируется в машинный код, что позволяет получить гораздо более высокую производительность, чем при интерпретации кода. Сообщество JavaScript-разработчиков видит постоянное улучшение производительности Node.js за счёт того, что Google постоянно работает над совершенствованием V8.

- Node.js — это опенсорсный проект, вокруг которого собралось огромное сообщество разработчиков.

Конкуренты: SpringBoot, Django

<https://habr.com/ru/company/ruvds/blog/519478/>

СУБД – Postgres, так как

- Бесплатное открытое ПО с открытым исходным кодом, соответствующее стандарту SQL. Эта СУБД является очень мощной системой.
- Существует довольно большое сообщество, в котором просто найти ответы на вопросы
- Большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими.
- Объектность - PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого

Релизы

Среда разработки – VisualCode

Система контроля версий - GitHub

Релизы будут публиковаться ночью по местному времени (03:00). Бэкапы будут создаваться перед каждым обновлением и каждую ночь (02:00).

Скрипт проведения собеседования

1. Просмотр GitHub/GitLab для ознакомления с предыдущими проектами соискателя.
2. Небольшое тестовое задание для соискателя.
3. Начало собеседования с рассказа о ценностях компании и ее проектах. Кандидат, который приходит на собеседование еще не наш сотрудник и нам важно заинтересовать его.
4. Обсуждение предыдущих проектов соискателя, сложностей их реализации и выбранных решений.
5. Собеседование по технической части

Вопросы технической части:

Для Frontend

- HTML + CSS+ JavaScript (понимание языка и его особенностей, структуры данных; строки, числа, объекты и массивы; замыкание и области видимости (как

работает и что значит); методы объекта (Bind, Call и Apply); контекст выполнения и стек вызовов функций)

- Фреймворки, Vue

- 1) На каком шаблоне проектирования основан Vue?
- 2) Как передать данные компоненту?
- 3) Назовите хуки жизненного цикла компонента в Vue.js?
- 4) Как обновить состояние компонента в Vue ?
- 5) Что такое вычисляемые свойства и когда их нужно использовать?
- 6) Как подключить внешний css файл в Vue ?
- 7) Как подключить jQuery плагин?
- 8) Как зарегистрировать компонент в Vue.js ?
- 9) Что такое Vuex ?
- 10) Как сделать условный рендеринг компонента?
- 11) Есть ли в Vue.js поддержка data binding? Если да, как его использовать?
- 12) Как реализовать маршрутизацию на стороне клиента в Vue?
- 13) Как программно сделать редирект в Vue Router?
- 14) Как защитить какой-то маршрут от несанкционированного доступа?

- Понимать, как работает система контроля версий GIT.

Для Backend

- Что такое Node.JS?
- Почему люди используют Node.JS?
- В чем разница между фронт-энд и бэк-энд разработкой?
- Что такое stubs?
- Опишите callback hell
- Что такое Event?

- Что такое event-driven программирование?
- В чем смысл однопоточности Node.JS?
- Что такое worker processes?
- Что такое Express JS?