

# ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОГРАММНОГО СРЕДСТВА ТЕРМИНАЛА ВЫДАЧИ МЕДИКАМЕНТОВ

Корытко А.С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Существует множество различных, подходящих для разработки десктопного приложения, типов архитектур, которые успешно применяются. Одной из наиболее используемых является клиент-серверная архитектура или, в переводе на английский язык – «Front end and Back end» [1].

В данной архитектуре «Front end» частью является клиентское приложение. Эта часть системы обменивается данными с серверной, отправляя запросы пользователя в виде HTTP запросов и получая ответы в виде структурированной структуры данных.

«Back end» – это REST API интерфейс на удаленном сервере [2]. Серверное приложение обеспечивает функционирование всех клиентских приложений. Все запросы к ней должны сопровождаться действительным ключом авторизации – токеном. В данный момент времени индустриальным стандартом определения токенов является JWT – JSON Web Token. В такой токен допускается запись дополнительных данных для идентификации пользователя, что даёт, вкупе с применением защищенного HTTP соединения с SSL шифрованием, защищённый канал связи.

Сервер и клиентское приложение программного средства будут реализовывать архитектуру «Front end and Back end». Однако стоит отметить, что обязательной составляющей программного средства является собственный API интерфейс терминала, который будет отслеживать движения механических составляющих и взаимодействует с клиентским приложением посредством протокола передачи данных WebSocket [3]. Соответственно, клиент-серверную архитектуру приложения должен дополнять еще один API интерфейс, который будет сообщать клиентскому приложению состояния терминала.

Для иллюстрации архитектуры программного средства была выбрана диаграмма компонентов, представленная на рисунке 1. Программное средство состоит из API сервера, интерфейса терминала и клиентского приложения.

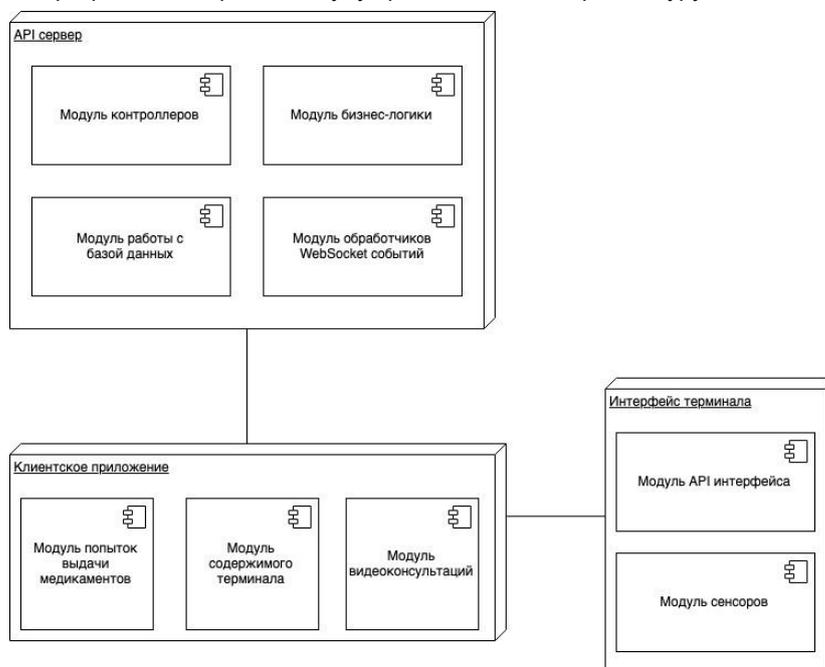


Рисунок 1 – Диаграмма компонентов

Стандартом архитектуры в области разработки является MVC, что расшифровывается как «Model-View-Controller» и переводится как «Модель-Представление-Контроллер». Архитектура MVC - схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо [4].

Исходя из требований, предъявляемых к серверному приложению, можно сделать вывод, что архитектура CQRS является наиболее подходящей, так как она позволяет восстановить систему из любого состояния, что является жизненно необходимым для терминала выдачи медикаментов.

Исходя из требований, предъявляемых к серверному приложению, можно сделать вывод, что архитектура Flux является наиболее подходящей, так как она позволяет разрабатывать расширяемые компоненты, которые могут быть применены в различных местах приложения.

Список использованных источников:

1. Front end and back end [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Front\\_end\\_and\\_back\\_end](https://en.wikipedia.org/wiki/Front_end_and_back_end). – Дата доступа: 05.10.2020.
2. API [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/API>. – Дата доступа: 05.10.2020.
3. WebSocket [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Glossary/WebSockets>. – Дата доступа: 05.10.2020.
4. Архитектура MVC [Электронный ресурс]. – Режим доступа: [https://www.artima.com/articles/dci\\_vision.html](https://www.artima.com/articles/dci_vision.html). – Дата доступа: 05.10.2020.