

УДК 004.05

## РЕФАКТОРИНГ АРХИТЕКТУРЫ КЛИЕНТСКОЙ ЧАСТИ ВЕБ ПРИЛОЖЕНИЯ ОБРАБОТКИ МЕДИЦИНСКИХ ДАННЫХ

Сулейманов Р.И. (Университет ИТМО)

Научный руководитель – старший преподаватель, доцент Береснев А.Д.  
(Университет ИТМО)

**Введение.** Каждый ведущий разработчик сам выбирает архитектуру приложения, в зависимости от многих факторов, таких как используемые библиотеки, бизнес логика, количество сущностей и своих привычек.

В самом начале данного проекта, никто не говорил мне, что я буду не один. Я сам писал компоненты, знал где и что лежит, сделал удобную для себя архитектуру. Но со временем команда росла, создавались новые компоненты, страницы, функции и прочее. К сожалению мои коллеги не достаточно опытны, в связи с чем, удобная архитектура превратилась в комок грязи (Big ball of mud). Стало сложнее писать компоненты, расширять функционал и просто разбираться в коде. Правильно построенная архитектура сэкономит нервы разработчиков, их время разработки, а также позволит легче войти в проект новым разработчикам.

**Основная часть.** Рассмотрена текущая архитектура с ее положительными и отрицательными сторонами. Проанализированы возможные альтернативы из топа самых популярных методов построения системы среди разработчиков. Были рассмотрены архитектуры:

- 1) Domain Driven Design and vertical slices: VS позволяет поддерживать единую ответственность на высоком уровне, если мы «минимизируем связь между слайсами и максимизируем связь в срезе»
- 2) Clear architecture: основная суть данной архитектуры заключается в том, что приложение делится на круговые слои. Чем дальше слой от центра, тем больше он знает. В идеале, внутренние слои, не знают ничего даже о фреймворке, который мы используем, потому что это бизнес логика.
- 3) Feature Sliced Design: модульная, бизнес ориентированная технология построения фронтенд приложений.

После анализа на уровень зацепленности и связности выбрана архитектура Feature Sliced Design, так как она имеет самые лучшие показатели, легче всего масштабируется, архитектура гибкая, ее легко изменить под новые условия бизнес, имеет контролируемые и явные связи, легко ориентироваться в пространстве имен и прочие достоинства. Составлен план миграции с текущей архитектуры на новую, состоящий из 4 шагов. И реализован сам переход. В ходе которого было добавлено около 100 новых файлов, но увеличился коэффициент переиспользования компонентов, появилось разделение на модули и исправлены все ошибки допущенные в ходе разработки.

**Выводы.** Проведен анализ популярных архитектур клиента веб приложений, улучшен порог входа новых разработчиков и масштабируемость системы.

### Список использованных источников:

1. Feature-Sliced Design [электронный ресурс]. URL <https://feature-sliced.design/> (дата обращения: 19.02.2023)
2. The Clean Architecture [электронный ресурс]. URL: <https://habr.com/ru/post/499078/> (дата обращения: 19.02.2023)
3. Архитектура фронтенда и какой она должна быть [электронный ресурс]. URL: <https://habr.com/ru/post/667214/> (дата обращения: 19.02.2023)