

скрипт для запуска всех необходимых программ и компонентов, что упрощает администрирование и перенос приложения на новые устройства. К тому же данный подход позволит достаточно просто развернуть приложение на облачных сервисах.

Общая схема структуры приложения, учитывающая вышеперечисленные инструменты и подходы, изображена на рис. 3.

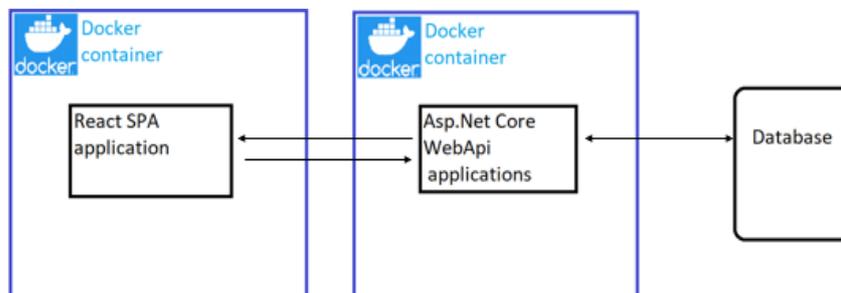


Рис. 3. Общая схема приложения

Таким образом, в рамках данной работы создается приложение, которое обеспечивает авторизацию и аутентификацию пользователей, учет клиентов, предоставляет пользователям интерфейс на подходящем для них языке, позволяет просматривать список мероприятий и покупать на них билеты без посещения точек продажи билетов и стояния в очередях.

Главными преимуществами для владельцев бизнеса являются невысокая цена разработки за счет отказа от ненужных на первоначальных этапах технологий и расширений, простота установки за счет использования контейнеризации, применение современных подходов проектирования, таких, как REST И SPA, благодаря чему достигается большая дифференциация ответственности между отдельными модулями, что при возникновении необходимости расширения текущей функциональности позволит это сделать без существенных временных и денежных затрат.

## «PANTRYPHARM» – ФАРМАЦЕВТИЧЕСКИЙ ПОМОЩНИК

**А. В. Езвенков**

*Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель Н. В. Самовендюк

Целью данного проекта является разработка удобной системы приложений, которая позволит пользователям получать справочную информацию о медикаментах (используя открытые API или путем сотрудничества с фармацевтическими компаниями, что будет являться плюсом для стартапа), пользоваться встроенными функциями проекта, такими, как:

- личный инвентарь (база данных) медикаментов с информацией о закупке, сроке годности и удобным добавлением медикамента (используя ИИ и ML);
- доступ к платформе через любые приложения (не только web-решение), используя как расширения для популярных платформ Apple health, Samsung health; так и отдельные приложения на платформах iOS, Android.

Схема представления системы пользователю дана на рис. 1.



Рис. 1. Схема представления системы пользователю

*Польза для предпринимателя, владельца или руководителя проекта.* Самой главной пользой для владельца проекта может быть сотрудничество, и данная модель, идея и тематика которой представлены в проекте, полностью подходят под этот критерий. Сотрудничество как с фармацевтическими компаниями в обмен на интеграцию рекламы, популяризацию тех или иных медикаментов/фармацевтических средств, и получение дохода от процента с продаж (при условии наличия функции покупки медикаментов той или иной компании) как процента для площадки-распространителя. Также стоит отметить сотрудничество с популярными платформами, интеграцию с которыми можно реализовать путем разработки расширений как отдельных клиентов для данной системы.

Схема представления возможных моделей проектирования представлена на рис. 2.

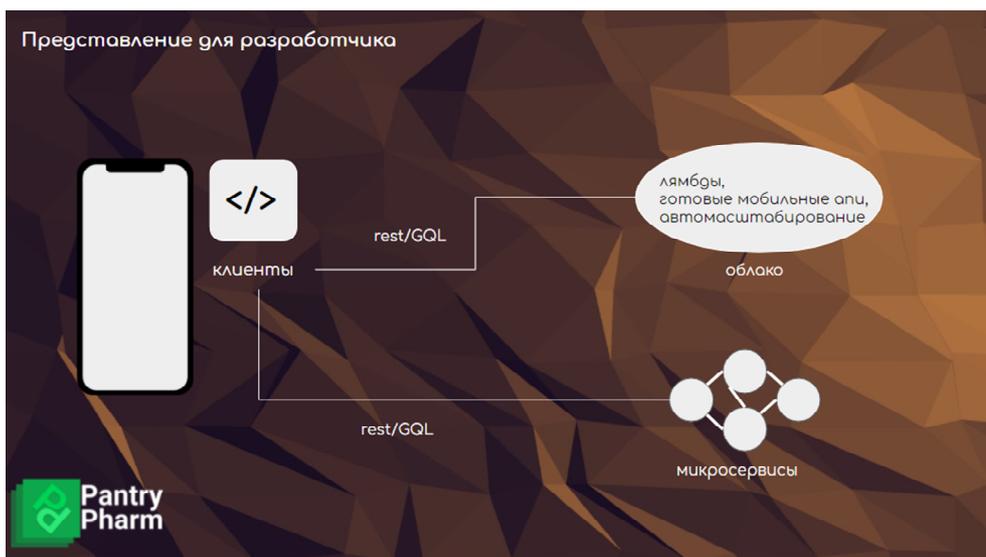


Рис. 2. Схема представления возможных моделей проектирования для разработчика

*Особенности разработки.* В качестве архитектуры для данного проекта была выбрана следующая модель: *Any client + serverless*. Выбор данной модели для разработки примечателен многочисленными плюсами для масштабирования, сопровождения и управления системой:

- максимальная эластичность. Быстрое масштабирование от нуля до тысяч параллельно работающих функций;
- полная абстракция от операционной системы или любого софта, использующегося для выполнения приложения. Вам неважно, запускаются ли ваши *Serverless* приложения на *Linux*, *Windows* или *custom OS*. Все, что вас волнует, это способность платформы выполнять *Python/Java/Ruby/YouNameIt* код и сопутствующие библиотеки для этого ЯП;
- при правильном проектировании функций легче построить слабо связанную архитектуру, при которой ошибка в одной функции не скажется на работоспособности всего приложения;
- ниже порог входа для новоприбывших. Понять «наносервис» из 100–500 строк (а это и есть обычный размер функции в *Serverless*) для нового разработчика в команде гораздо проще, чем понять *legacy* проект с миллионом строк и сложных связей.

*Технологии для построения базовой версии модели.* В качестве основных технологий для построения веб-решения будут использованы фреймворки *angular* (как удобное по структуре решение для построения *UI*) и *firebase platform* (как идеальный инструмент связки веб-решения на *node.js* и облачной платформы *google cloud*).

В качестве основных технологий для построения *back-end* составляющей системы будут использованы инструменты *google cloud* платформы:

- *Cloud-functions* – для реализации наносервисов, соответствующих *serverless* модели.
- *Firebase-ml* – для реализации технологий нейронных сетей, вспомогательного инструмента для таргетной рекламы, рекомендаций и умного добавления медикаментов.
- *Pub-sub* – для реализации *message query* между наносервисами и компонентами системы.
- *Firebase/firestore – nosql* бд для хранения данных.
- *Security* – для защиты пользовательских данных.
- *Pay-api* – для обработки и осуществления транзакций, используя платформу *google*.
- *Cloud-storage* – для хранения открытых данных пользователей.
- *Firebase-analytics* – для составления аналитики по всей системе.

## ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ РЕШЕНИЯ ПРОБЛЕМ РАСПОЗНОВАНИЯ ЖЕСТОВ

М. А. Рябиков

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель В. С. Захаренко

С развитием персонального компьютера, также наблюдалась эволюция пользовательского интерфейса. От текстового командного до графического интерфейса: от простой клавиатуры до мыши, геймпада, электронной ручки, видеокамеры и т. д.