

УДК 004.048

**PANTRYPHARM – ФАРМАЦЕВТИЧЕСКИЙ ПОМОЩНИК****А. В. Езвенков***Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь**Рассмотрена проблема инвентаризации лекарственных препаратов и предложен автоматизированный метод решения в виде реализации облачной информационной системы с портативными клиентскими решениями.***Ключевые слова:** стартап, интеграция рекламы, абстракция от ОС, наносервисы.**PANTRYPHARM – PHARMACEUTICAL ASSISTANT****A. V. Ezvenkov***Sukhoi State Technical University of Gomel, the Republic of Belarus**The problem of drug inventory is considered and an automated solution method is proposed in the form of an implementation of a cloud information system with portable client solutions.***Keywords:** startup, advertising integration, OS abstraction, nano services.

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 такими встроенными функциями проекта, как:

- личный инвентарь (база данных) медикаментов с информацией о покупке, 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 изображена на рис. 1



Рис. 1. Схема представления системы пользователю

Главной пользой для владельца проекта может быть сотрудничество и данная модель, идея и тематика, которые представлены в проекте и полностью подходят под этот критерий. Сотрудничество с фармацевтическими компаниями в обмен на интеграцию рекламы, популяризацию тех или иных медикаментов/фармацевтических средств; а также получение дохода от процента с продаж (при условии наличия функции покупки медикаментов той или иной компании) как процента для площадки-распространителя.

Стоит также отметить сотрудничество с популярными платформами, интеграцию с которыми можно реализовать путем разработки расширений как отдельных клиентов для данной системы.

На рис. 2 изображена схема представления возможных моделей проектирования для разработчика.



Рис. 2. Схема представления возможных моделей проектирования для разработчика

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 не должно быть важно, запускаются ли ваши *Serverless* приложения на *Linux*, *Windows* или *custom OS*. Все, что вас должно волновать, – 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.

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

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

- *Cloud-functions* – для реализации нано сервисов, 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.

С помощью *API*-маркировки изображений *ML Kit Firebase* мы можем распознавать сущности в изображении без необходимости предоставлять какие-либо дополнительные контекстные метаданные, используя *API* на устройстве или облачный *API*. Данные технологии используются для распознавания упаковки препарата без необходимости заполнять форму для добавления его в инвентарь.

Маркировка изображений дает представление о содержании изображений. При использовании *API* мы получаем список распознанных сущностей: люди, вещи, места, действия и т. д. (в частности, упаковки различных фармацевтических препаратов). Каждая найденная метка поставляется с оценкой, которая указывает на уверенность модели *ML* в своей актуальности. С помощью этой информации можно выполнять такие задачи, как автоматическое создание метаданных и модерация контента.

УДК 004.891:004.93

### **ОБНАРУЖЕНИЕ ПЛОДОВ ОГУРЦА ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА СБОРА МНОГОФУНКЦИОНАЛЬНЫМ АГРАРНЫМ РОБОТОМ НА ОСНОВЕ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ**

**К. С. Курочка, В. С. Ермашкевич, Д. А. Дорошко, Н. И. Расшивалов**

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

*Описан процесс реализации алгоритма по обнаружению плодов огурца на основе сверточных нейронных сетей для их сбора с помощью многофункционального аграрного робота.*

**Ключевые слова:** сверточная нейронная сеть, обнаружение, робот, огурцы.