УДК 004.777:614

ВЕБ-ПРИЛОЖЕНИЕ МЕДИЦИНСКОГО СЕРВИСА

Чвыров И.А.

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

Научный руководитель: Прудник А.М. – канд. техн. наук, доцент, доцент кафедры ИПиЭ

Аннотация. Представлено веб-приложение медицинского сервиса в микросервисной архитектуре на языке Python, которое позволит снизить нагрузку на амбулаторную часть системы здравоохранения и изолировать пациентов от нежелательных визитов в данные учреждения. Актуальность состоит в том, что сервисы подобного рода в настоящее время очень востребованы.

Ключевые слова: веб-сайт, веб-приложение, микросервис, архитектура, разработка, Python, PostgreSQL, FastAPI.

Введение. В настоящее время у каждого человека есть множество приложений на их устройствах, которые позволяют получать разного рода информацию и товары, не выходя из дома. Покупки в магазинах, бронирование различных услуг, заказ билетов в кинотеатр, доставка еды – все это перешло в онлайн.

Пандемия также ускорила переход в онлайн многих процессов, которые традиционно не затрагивала цифровизация. В частности, оказались востребованы программные продукты, которые с одной стороны позволяют снизить нагрузку на амбулаторное звено здравоохранения, а с другой – изолировать пациентов и врачей, снизив тем самым риск заражения вирусными инфекциями пациентов. Постепенно они прочно вошли в нашу жизнь.

Основная часть. Целью работы является создание веб-приложения в микросервисной архитектуре на языке Python, с помощью которого пользователи смогут получать рецепты на очки, онлайн не выходя из дома.

Для выполнения данной работы должны быть решены следующие задачи:

– изучить микросервисный подход в разработке архитектуры веб-приложений, произведено сравнение его с другими существующими видами архитектур, а также выявление его положительных и отрицательных сторон;

– спроектировать модель данных;

– выделить микросервисы на основе полученных доменов;

– спроектировать базы данных для каждого из микросервисов;

– спроектировать внешний API для взаимодействия с сервисом;

– имплементировать выделенные микросервисы;

– наладить процессы взаимодействия между микросервисами.

Микросервисная архитектура – это оптимальный подход к разработке веб-приложения. В микросервисной архитектуре приложение делится на небольшие и автономные компоненты (микросервисы) с определенными интерфейсами [1].

Помимо этого, к системе также существует ряд требований:

1. Пациент должен иметь возможность регистрироваться и авторизовываться в приложении.

2. Пациент должен иметь возможность проходить опросник, вносить данные предыдущих рецептов на очки, оставлять дополнительные комментарии для докторов.

3. Пациент должен иметь возможность просматривать рецепты на очки, выписанные доктором.

4. Доктор должен иметь возможность просматривать данные, полученные от пациента.

5. Доктор должен иметь возможность выписывать рецепты и оставлять комментарии на странице пациента.

6. Администратор должен иметь возможность просмотра статистики, демонстрирующей, как много кейсов было обработано доктором.

Отталкиваясь от требований, выделены следующие микросервисы. Микросервис для пациентов (User Service) хранит данные о пациентах и итоговых рецептах на очки, выписанных доктором. Микросервис для докторов (Doctor Service) хранит часть данных о пациентах, офтальмологических данных, необходимых для принятия решения и выписанные рецепты на очки. В микросервисе для администраторов (Admin Service) собрана информация докторах и о том, сколько пациентов они рассмотрели.

Для разработки пользовательского микросервиса использовался язык программирования Python и его популярный современный веб-фреймворк FastAPI. Для взаимодействия с базой данных использовалась ORM система SQLAlchemy. Веб-фреймворк FastAPI предоставляет встроенную валидацию входных параметров, тела HTTP-запроса, тела HTTP-ответа [2].

Клиентское приложение для пациентов написано на языке JavaScript при использовании библиотеки React. Приложение имеет следующие страницы: страница регистрации пользователя, страница авторизации пользователя, основная страница приложения, на которой расположены форма опросника о текущем состоянии пользователя и форма для загрузки данных предыдущего выписанного рецепта, представленная на рисунке 1.

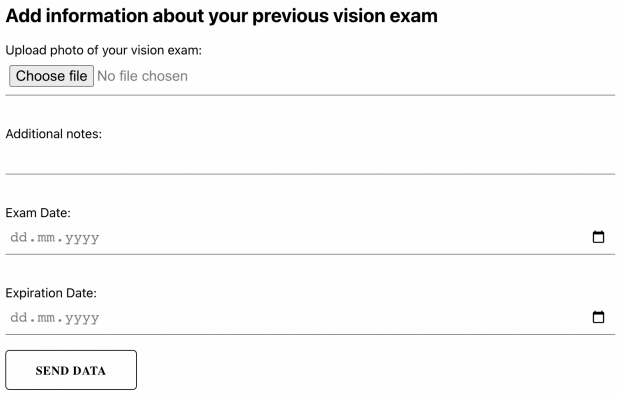

Рисунок 1 – Форма загрузки предыдущего рецепта

Для разработки «врачебного» микросервиса выбран язык программирования Python и веб-фреймворк Fast-API. Взаимодействие между клиентом и сервером будет происходить посредством языка запросов GraphQL.

Фронтенд приложение портала врачей представляет собой одностраничное приложение, написанное с применением библиотеки React и языка программирования JavaScript. Выполнено в минималистичном дизайне. В данной версии приложения доктор-офтальмолог может либо продлить имеющийся рецепт, либо отменить его, в случаях, когда пациент испытывает проблемы и ухудшение зрения.

AdminService – сервис, используемый администраторами для просмотра статистики о работе докторов. В качестве базы данных используется Redis. Redis хранит данные в формате ключ-значение [3, с. 389]. В качестве ключа используется email адрес доктора-офтальмолога, а в качестве значения – число, равное количеству рецептов, выписанных данным доктором.

Фронтенд приложение для администраторов также написано с помощью JavaScript и React. На главном экране приложения отображается таблица, в которой агрегирована информация о количестве выписанных рецептов и отказов врачей-офтальмологов, что представлено на рисунке 2.

Doctor Email	Prescription Count	Referer Count
torch.ilya.doctor@gmail.com	159	6
zagg.perso@warbyparker.com	237	9
eclair.santiago@coherent.com	262	16
antony.hopkins@me.com	305	37

Рисунок 2 – Статистика по пользователям-докторам

В данной работе представлена лишь первая версия приложения и реализован базовый необходимый функционал. Однако возможен и дополнительный функционал, такой как:

1. Поддержка выписанных рецептов для контактных линз.
2. Создание «истории рецептов» пациентов.
3. Online-тест для проверки состояния зрения пользователя.
4. Online-чат для взаимодействия пациентов с врачами.
5. Версия для слабовидящих.

Заключение. Таким образов, в ходе работы было спроектировано и реализовано веб-приложение, позволяющее автоматизировать процесс выписки рецептов на очки для пациентов и врачей-офтальмологов. Приложение такого рода позволит снизить загрузку амбулаторного сектора здравоохранения, а также снизить риски распространения вирусных инфекций путем переноса взаимодействия пациентов и докторов в онлайн.

Также были намечены пути развития приложения и представлены примеры новых функциональных возможностей, которые в будущем могут быть интегрированы в систему.

Список литературы

1. Микросервисная архитектура в разработке приложений: преимущества и недостатки [Электронный ресурс] – 2023. – Режим доступа: https://habr.com/ru/post/682628//*– Дата доступа: 03.03.2023*

2. Документация к веб фреймворку FastAPI [Электронный ресурс]. – 2023. – Режим доступа: https://fastapi.tiangolo.com/. – Дата доступа: 03.03.2023

3. *Лутц Марк Изучаем Python, том 1, 5-еизд.: Пер.с англ.–СПб.: ООО“Диалектика”, 2019. – 832 с.*

UDC 004.777:614

MEDICAL SERVICE WEB APPLICATION

Chvyrov I.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Prudnik A.M. – PhD, associate professor, associate professor of the Department of EPE

Annotation. A web application of a medical service in a micro-service architecture in Python is presented, which will reduce the burden on the outpatient part of the healthcare system and isolate patients from unwanted visits to these institutions. The relevance lies in the fact that services of this kind are currently in great demand.

Keywords: website, web application, microservice, architecture, development, Python, PostgreSQL, FastAPI.