

## ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ ДНЕВНИКА ДИАБЕТИКА

Э. С. Суховенко

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

Научный руководитель Д. В. Прокопенко

*Рассмотрено применение интеллектуальных технологий для автоматизации дневника диабетика. Автор предоставляет подход к решению задачи, основанный на REST и SPA, которые позволяют существенно упростить масштабирование приложения, а также его производительность, однако ведут к значительному усложнению развертывания приложения.*

**Ключевые слова:** автоматизация, web-приложение, программный комплекс, база данных, REST API.

В настоящее время внедрение технологий автоматизации является повсеместным и уже закрепилось как важная часть жизни людей. Обычный человек не должен думать о том, как он покупает билеты на поезд или же, как он делает заказ доставки еды, ему главное, чтобы все эти процессы были интуитивно понятны и просты.

На западе в направлении создания информационных систем для контроля состояния пациентов уже достигнуты значительные результаты. В США сейчас различные медицинские информационные технологии прочно вошли в жизнь людей.

Беларусь пока отстает в этом вопросе, но уже наметились первые шаги в направлении внедрения электронных историй болезни, которые открывают широчайшие перспективы для совершенствования системы здравоохранения и повышения качества оказания медицинской помощи. Таким образом, актуальными являются исследования, связанные с проектированием, разработкой и внедрением информационных технологий в процесс контроля состояния организма в период заболевания.

Главная цель работы – автоматизация ведения дневника диабетика, обработка информации об уровне сахара пациента, оповещения пациента о необходимости принятия инсулина.

Очень важным является вопрос доступности создаваемого продукта. В мире существует около пяти операционных систем для персональных компьютеров и примерно столько же – для мобильных устройств. Поэтому создание программного комплекса под конкретную систему существенно снизит популярность, а соответственно и прибыль компании. К тому же важна возможность работать и пользоваться услугами из разных мест. Оптимальное решение в подобной ситуации – создание web-приложения с выделенной логикой, а для доступности в местах без интернета – мобильное приложение.

Сам же разрабатываемый программный продукт должен обеспечивать автоматизацию следующих бизнес-процессов:

- уведомление пациента о необходимости принятия инсулина;
- учет клиентов (хранение и обработка информации о клиентах и их учетных записях);
- предоставление информации о болезни (диабете);
- учет показателей здоровья пациента, таких как давление и пульс;
- ведение дневника питания (количество калорий, белков, жиров и углеводов, принятых за день);

– возможность просматривать важные показатели здоровья за определенный период, на основании которых приложение будет составлять рекомендации по питанию;

– каждый пользователь приложения должен иметь доступ только к определенному функционалу, который определяется его ролью.

В качестве базы данных оптимальным выбором в плане производительности является использование комбинированного подхода хранения данных, т. е. использование реляционной базы данных при операциях добавления и редактирования данных, и нереляционной – при операциях чтения данных. Однако минус данного подхода – увеличение стоимости и времени разработки за счет реализации механизмов синхронизации данных. Поэтому для удешевления разработки для хранения данных была выбрана реляционная база данных MS SQL, так как она позволяет в полной мере применять механизм транзакций.

Логическая модель данных для выполнения операций авторизации и аутентификация представлена на рис. 1.

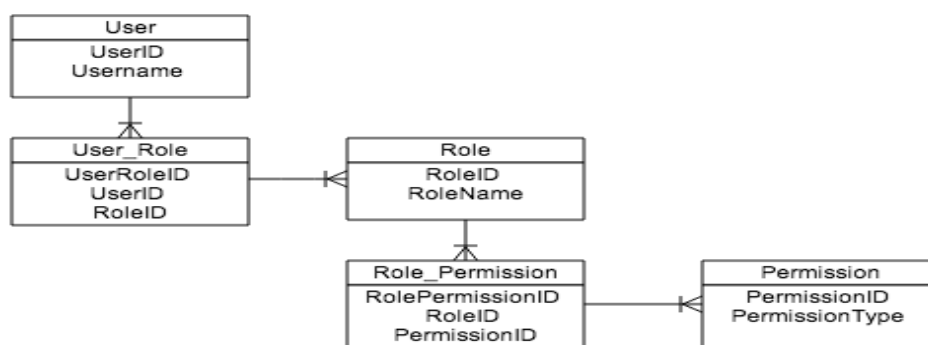


Рис. 1. Логическая модель данных для операций авторизации и аутентификации

Для сохранения возможности для дальнейшего развития приложения наиболее подходящим архитектурным стилем является применение REST API. Основными плюсами подобного решения будет упрощение поддержки приложения за счет строгой детерминированности элементов приложения, а также широкие возможности для расширения, так как большинство современных инструментов по созданию пользовательского интерфейса позволяют осуществлять интернет-запросы. Также данный подход позволяет внедрить единую систему авторизации и унификации на основе JWT токенов.

Для графического интерфейса наиболее подходящим подходом в данном случае становится использование single page application. Суть данного подхода состоит в том, что web-приложение загружается единожды, и по мере необходимости данные загружаются за счет выполнения запросов к серверному приложению. Это позволяет повысить скорость взаимодействия с сервером за счет того, что пользователю будут отправляться только маленькие кусочки информации, а не вся страница целиком. Также сама обработка данных будет происходить на компьютере пользователя, а не на сервере.

В проекте для реализации данного подхода выбран язык программирования type-script и библиотека React, так как они имеют большое распространение, что снижает затраты на разработку, а также реализуют удобный компонентный подход.

Вышеперечисленные подходы позволяют существенно упростить масштабирование приложения, а также его производительность, однако ведут к значительному усложнению развертывания приложения.

Оптимальным выходом в данной ситуации является использование контейнеризации, а один из самых распространенных инструментов для этого – Docker. Главные преимущества данного подхода состоят в том, что, как и при виртуализации, мы получаем возможность запустить приложение изолированно. Однако в отличие от виртуализации, при данном подходе используется ядро установленной на компьютере операционной системы, а не эмулируется новое, что положительным образом сказывается на производительности. Также инструмент Docker позволяет создать удобный 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. 2.

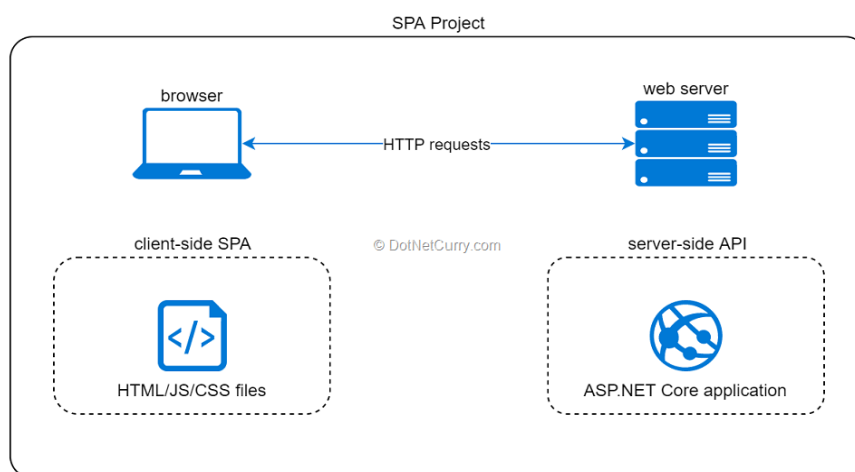


Рис. 2. Общая схема приложения

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.