

Рис. 3. Интерфейс мобильного приложения для мастера отгрузки

Таким образом, была разработана модель программного комплекса оперативно-автоматического управления очередью, который позволит предприятию и водителям уменьшить временные издержки и увеличить пропускную способность предприятия.

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ ПРОДАЖИ ТЕАТРАЛЬНЫХ БИЛЕТОВ

Н. С. Заяц

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

Научный руководитель Т. Л. Романькова

Любая коммерческая организация стремится максимизировать свою прибыль и уменьшить расходы, а для этого необходимо рационально подходить к организации рабочего процесса. В наши дни добиться данной цели возможно благодаря применению современных информационных технологий. Автоматизация бизнес-процессов помогает владельцам бизнеса упростить администрирование и контроль за своими ресурсами, а конечным пользователям упрощает взаимодействие с бизнесом.

В организации, занимающейся продажей билетов, автоматизация особенно важна. Большие расстояния между местами проведения мероприятий, наличие длинных очередей и негибкого рабочего графика существенно затрудняет жизнь конечным клиентам. А владельцам подобных организаций все сложнее работать с клиентской базой, своевременно реагировать на различные происшествия, предоставлять качественный и персонализированный сервис.

Таким образом, главная цель работы – автоматизация основных бизнес-процессов для оптимизации обработки информации, повышения качества обслужи-

вания и упрощения принятия управленческих решений.

Очень важным является вопрос доступности создаваемого продукта. В мире существует около 5 операционных систем для персональных компьютеров и примерно столько же – для мобильных девайсов. 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.

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

- рассылка корреспонденции (информация о новых мероприятиях, подтверждения оформления билетов и т. п.);
- учет клиентов (хранение и обработка информации о клиентах и их учетных записях);
- осуществление операций бронирования и покупки на выбранное пользователем мероприятие;
- управление мероприятиями (возможность создания, редактирования и удаления информации о мероприятиях);
- возможность загрузки фотографий для мероприятий;
- поиск необходимых данных;
- 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, что очень важно при выполнении финансовых операций.

Логическая модель предметной области приведена на рис. 1.

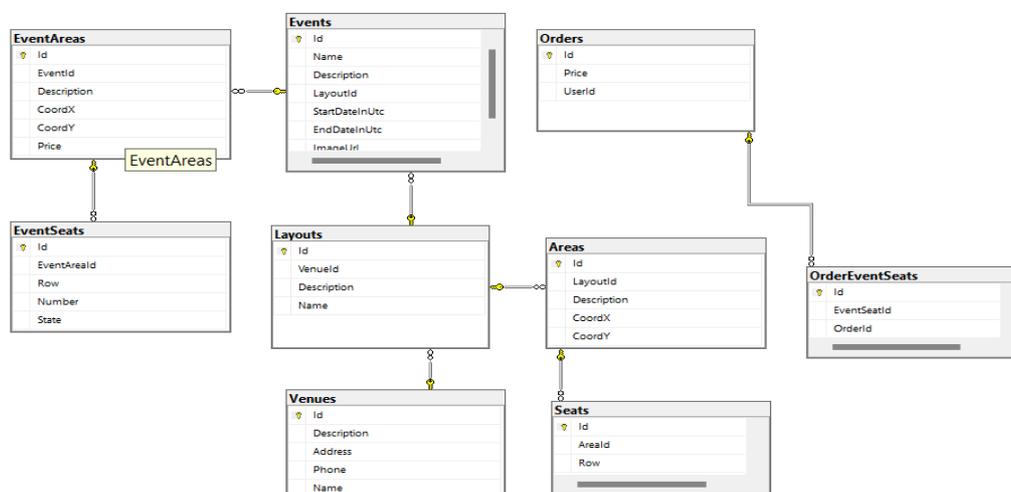


Рис. 1. Логическая модель данных разрабатываемого продукта

Логическая модель данных для выполнения операций авторизации и аутентификации представлена на рис. 2.

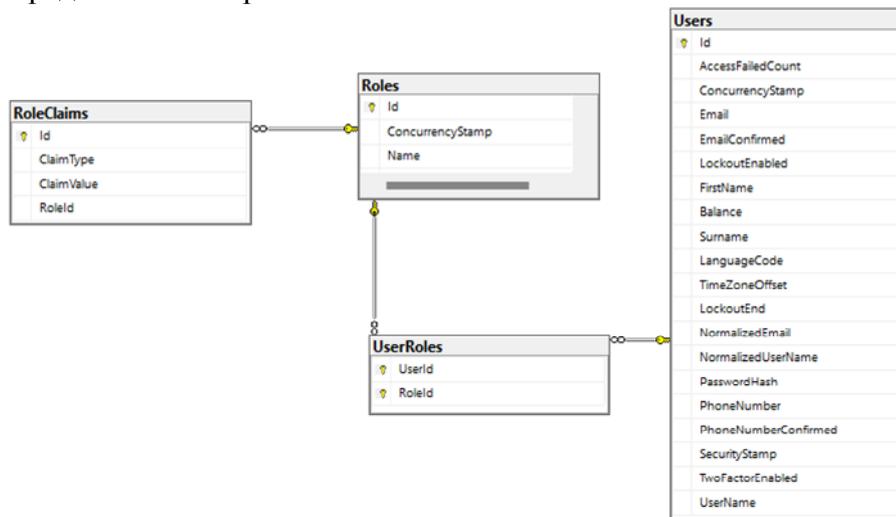


Рис. 2. Логическая модель данных для операций авторизации и аутентификации

Для сохранения возможности для дальнейшего развития приложения наиболее подходящим архитектурным стилем является применение REST API. Основными плюсам 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 единой загрузкой, 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 javascript и фреймворк 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 положительно 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. 3.

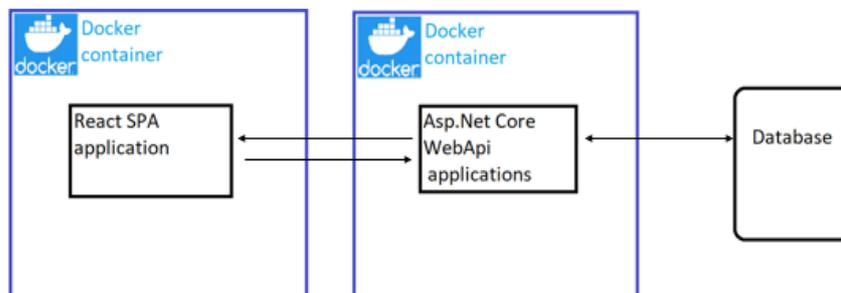


Рис. 3. Общая схема приложения

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.