

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Разработка и реализация ИС «Мобильное приложение для
дистрибуции билетов на мероприятия»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 561 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Цветкова Даниила Константиновича

Научный руководитель
доцент, к.э.н

Ю. В. Мельникова

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2023

Введение. На сегодняшний день, мобильный трафик уже давно превышает трафик с персональных компьютеров. StatCounter является одним из крупнейших сервисов веб-аналитики и собирает данные с миллиардов веб-страниц по всему миру каждый месяц. По данным, представленным на этом сервисе, мировое соотношение трафика с мобильных устройств и ПК на март 2023 года составляет примерно 62% к 38%. Это означает, что более половины всех пользователей Интернета по всему миру используют мобильные устройства для доступа в Интернет.

Это имеет значительное влияние на веб-разработку и маркетинг, так как сайты теперь должны быть адаптированы для мобильных устройств, чтобы обеспечить оптимальный пользовательский опыт. Однако, мобильная версия сайта какого-либо сервиса в большинстве случаев используется для того, чтобы заинтересовать пользователя скачать мобильное приложение сервиса.

Также существует множество сервисов, которые стали популярными без веб-сайта, благодаря мобильным приложениям. Некоторые из них: Uber, Instagram, Snapchat, Tinder, TikTok.

Это лишь небольшой список сервисов, которые стали популярными без веб-сайта, используя только мобильное приложение. Такие сервисы предлагают удобный способ доступа к услугам и контенту на мобильных устройствах и могут оказаться весьма успешными.

1. **Лучшая производительность:** Мобильные приложения работают быстрее, чем мобильные версии сайтов, потому что они могут хранить данные локально на устройстве пользователя. Это позволяет уменьшить количество запросов к серверу и ускорить загрузку страниц.
2. **Улучшенный пользовательский опыт:** Мобильные приложения имеют удобный интерфейс, специально разработанный для мобильных устройств, что делает их более удобными для использования, чем мобильные версии сайтов. Кроме того, мобильные приложения могут использовать функции устройства, такие как камера, микрофон и GPS-навигация, для улучшения пользовательского опыта.
3. **Возможность работы без интернета:** Мобильные приложения могут быть разработаны с возможностью работы без интернета. Это позво-

ляет пользователям использовать приложение даже в местах с плохим интернет-соединением.

4. **Большая лояльность пользователей:** Пользователи, которые устанавливают мобильные приложения, склонны использовать их чаще и дольше, чем мобильные версии сайтов. Это может привести к большей лояльности пользователей и увеличению продаж.
5. **Большой контроль над функциональностью:** Приложения позволяют разработчикам иметь более точный контроль над функциональностью и производительностью приложения. Они также позволяют лучше защищать данные пользователей и предотвращать атаки хакеров.

Актуальность темы данной бакалаврской работы заключается в том, что с развитием технологий и повсеместным использованием мобильных устройств, все больше людей предпочитают приобретать товары и услуги онлайн. В настоящее время многие люди используют мобильные приложения для покупки билетов на различные мероприятия, такие как концерты, спортивные события и культурные мероприятия. В связи с этим, разработка мобильного приложения для продажи билетов на мероприятия становится актуальной задачей для различных компаний и предпринимателей, которые хотят обеспечить своим клиентам удобный и быстрый доступ к покупке билетов. Кроме того, разработка такого приложения может способствовать развитию индустрии развлечений и увеличению числа людей, которые посещают мероприятия, что может привести к экономическим выгодам как для компаний, так и для регионов, в которых проводятся мероприятия. Еще одним аспектом актуальности данной работы является возможность для неопытных людей легко и удобно продавать свои билеты. Такое приложение может позволить людям, которые не имеют опыта в организации мероприятий, продать свои билеты непосредственно покупателям, что может уменьшить затраты на аренду места для продажи билетов и обеспечить удобство для продавцов. Кроме того, такое приложение может устранить необходимость в организаторах мероприятий, которые обычно занимаются продажей билетов, что может уменьшить затраты на проведение мероприятий и сделать билеты доступнее для покупателей. Все эти факторы подчеркивают актуальность и важность разработки мобильного приложения для продажи билетов на мероприятия.

Цель данной бакалаврской работы - разработка приложения для продажи билетов на мероприятия, которое обеспечивает быструю и удобную покупку билетов, а также возможность продажи билетов между пользователями.

Для реализации цели поставлены задачи:

1. Проанализировать технологии разработки мобильных приложений
2. Спроектировать информационную систему для дистрибуции билетов
3. Спроектировать БД для системы
4. Реализовать информационную систему

Основная часть В данной научной работе была разработана информационная система для продажи электронных билетов на мероприятия. Основной целью работы было создание эффективного и удобного приложения, способного обеспечить пользователям возможность приобретения билетов в онлайн-режиме.

В первой главе работы было проведено проектирование информационной системы с использованием языка моделирования UML. В соответствии с рисунком 1, была составлена диаграмма вариантов использования, которая позволила определить основные функциональные возможности приложения и взаимодействие между пользователями и системой. Данная диаграмма взаимодействия описывает основные роли пользователей и их взаимодействие в приложении по продаже билетов.

Это позволило внимательно рассмотреть различные сценарии использования приложения и спланировать его функциональность таким образом, чтобы оно полностью удовлетворяло потребностям пользователей. Далее была создана диаграмма последовательности, которая иллюстрирует последовательность действий при продаже билетов. Эта диаграмма помогла визуализировать и описать процесс взаимодействия между пользователями и системой на более детальном уровне. Также была разработана диаграмма активности, которая позволяет моделировать потоки действий и процессы, происходящие в системе. Она помогла определить основные шаги и этапы взаимодействия пользователя с системой.

Во второй главе работы была осуществлена разработка баз данных. Был проведен обзор существующих баз данных и выбрана наиболее подходящая для данного проекта. Определение подходящей базы данных играет важную

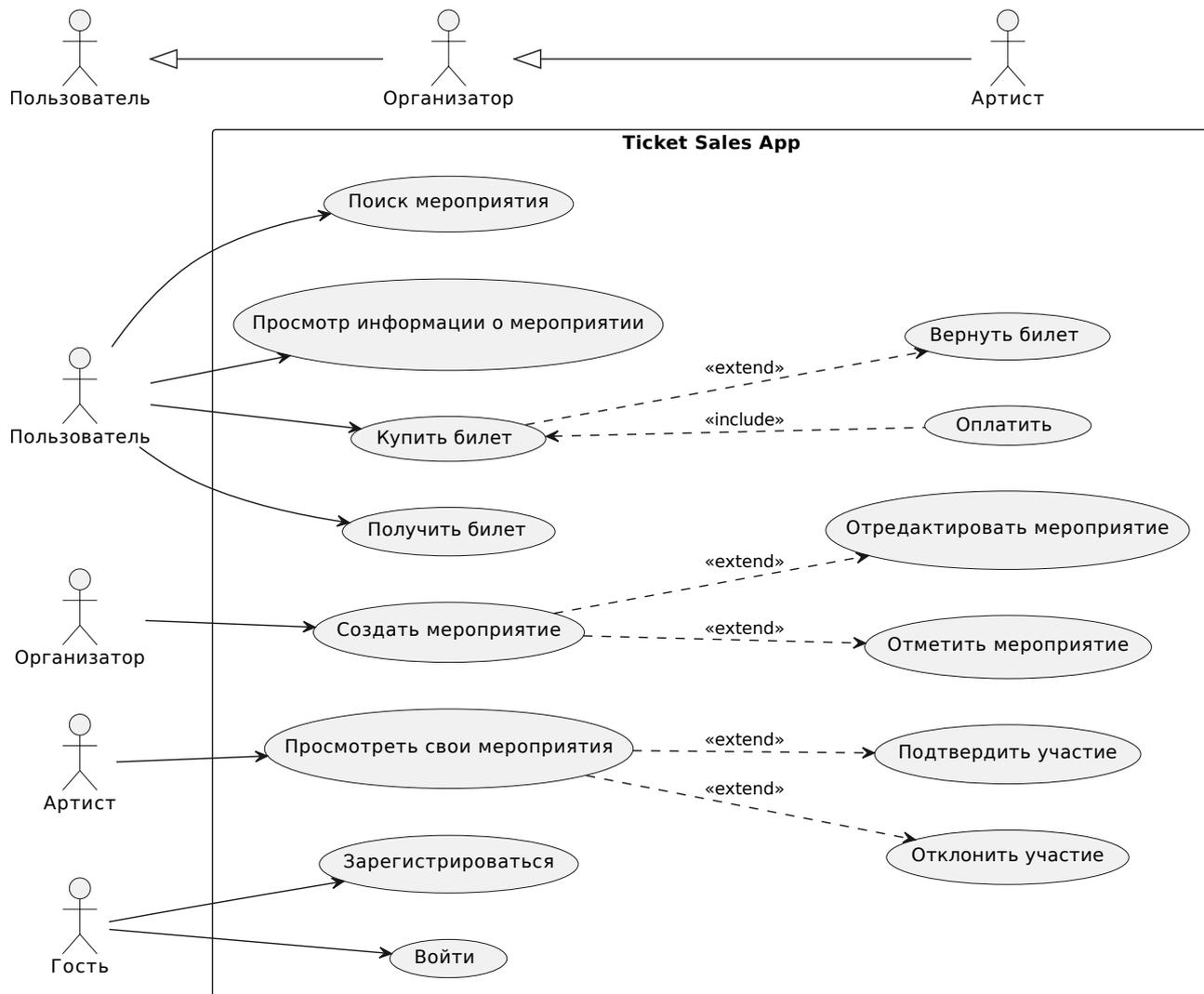


Рисунок 1 — Диаграмма вариантов использования

роль в создании стабильной и эффективной системы. Затем была построена ER-диаграмма, которая отражает структуру и взаимосвязи между сущностями системы. ER-диаграмма позволила определить основные сущности и атрибуты, а также связи между ними. Она служит основой для разработки базы данных и обеспечивает целостность данных в системе. ER-диаграммы, или диаграммы сущность-связь, представляют собой графический инструмент для моделирования структуры баз данных. Они полезны в процессе проектирования, поскольку обеспечивают наглядное представление о сущностях, атрибутах и связях между ними. ER-диаграммы являются эффективным способом определения ключевых сущностей, уточнения требований и установления базовой структуры базы данных. Они также позволяют выявить связи между сущностями и определить ограничения целостности данных. В целом, ER-диаграммы играют важную роль в разработке информационных систем и способствуют успешной реализации проектов баз данных.

В информационной системе был использован Spring Boot в качестве фреймворка для разработки серверной части приложения. Spring Boot предоставляет удобные инструменты и функциональность для создания масштабируемых и гибких приложений.

Одной из ключевых особенностей, реализованной в информационной системе, был механизм безопасности с использованием JWT. JWT - это открытый стандарт (RFC 7519), который определяет компактный и самодостаточный формат для представления информации в виде токена. Токены JWT используются для аутентификации и авторизации пользователей в приложениях.

В контексте разработки информационной системы, использование JWT вместе с механизмом безопасности Spring Boot предоставляет ряд преимуществ. Один из ключевых аспектов состоит в том, что JWT позволяет реализовать механизм без сохранения состояния (stateless), что означает, что сервер не хранит информацию о состоянии аутентификации пользователя. Вместо этого, после успешной аутентификации, сервер создает токен JWT, содержащий информацию о пользователе и его правах. Этот токен затем передается клиенту и включается в каждый последующий запрос. Сервер может проверить подлинность и целостность токена, используя секретный

ключ, и на основе этой проверки определить, имеет ли пользователь доступ к запрашиваемым ресурсам.

С использованием Spring Boot, реализация механизма безопасности с JWT становится еще более удобной и эффективной. Spring Security, модуль фреймворка Spring, предоставляет интеграцию с JWT, позволяя легко настроить проверку токенов, авторизацию и аутентификацию. Благодаря гибкости Spring Boot и Spring Security, разработчики могут легко настроить различные аспекты безопасности, такие как управление правами доступа, защита ресурсов и обработка аутентификационных запросов.

В результате, внедрение механизма безопасности с использованием JWT и Spring Boot позволяет обеспечить безопасность информационной системы, аутентифицировать пользователей и предоставлять доступ к ресурсам на основе их прав. Это повышает уровень доверия пользователей и обеспечивает защиту конфиденциальности и целостности данных.

При обзоре различных технологий для создания мобильных приложений автор приходит к выводу, что выбор оптимальной технологии зависит от конкретных целей, задач, бюджета и аудитории проекта. Не существует универсального решения, которое было бы идеальным для всех случаев. Однако, на основе проведенного обзора, автор приходит к убеждению, что нативная разработка под платформу Android с использованием языка программирования Java и XML для разметки интерфейса является эффективным и надежным способом создания качественных мобильных приложений.

Нативная разработка на Android позволяет полностью использовать возможности платформы, обеспечивая высокую производительность, стабильность и оптимизацию приложения. Это также упрощает процесс тестирования, отладки и поддержки приложения, благодаря использованию стандартных инструментов и сред разработки для Android.

Комбинация технологий XML, Java, PostgreSQL, Spring Boot Security и JWT в реализации дипломного проекта предлагает оптимальное решение с высокой степенью гибкости, безопасности и мощности. XML обеспечивает гибкую структуру данных, а Java предоставляет мощные возможности обработки и управления данными. PostgreSQL является надежной и масштаби-

руемой системой управления базами данных, а Spring Boot Security и JWT обеспечивают безопасность приложения на различных уровнях.

Таким образом, на основе проведенного обзора, автор приходит к заключению, что комбинация нативной разработки под платформу Android с использованием технологий XML, Java, PostgreSQL, Spring Boot Security и JWT является оптимальным выбором для реализации дипломного проекта, обеспечивая создание надежного, безопасного и функционального мобильного приложения.

Это позволило выбрать наиболее подходящие инструменты для создания надежной и функциональной системы. Затем была выполнена реализация базы данных, включая создание таблиц и связей между ними. Разработка базы данных играет важную роль в обеспечении эффективного хранения и доступа к данным.

В процессе разработки баз данных для дипломного проекта были применены различные подходы и методы, обеспечивающие эффективное управление данными и обеспечение их целостности. Ниже представлен обзор ключевых особенностей разработанных баз данных:

1. **Уникальные ограничения:** В таблицах `roles`, `payment_statuses`, `photo_categories` и других были определены уникальные ограничения для соответствующих полей, таких как `role_name`, `payment_status_name` и `photo_category_name`. Это гарантирует, что каждое значение в этих полях будет уникальным, предотвращая дублирование и обеспечивая целостность данных.
2. **Триггеры:** Триггер `sold_tickets_trigger` обеспечивает проверку наличия билетов перед их продажей. Он гарантирует, что количество продаваемых билетов не превышает доступное предложение. Триггеры помогают автоматизировать сложные проверки и действия в БД, обеспечивая целостность данных и предотвращая ошибки или нежелательные операции.
3. **Ограничения уникальности:** Уникальные ограничения, такие как `UNIQUE(country_id, city_name)` в таблице `cities`, обеспечивают уникальность комбинации значений в указанных полях. Это предотвращает нарушение целостности данных и дублирование информации, что

особенно важно при работе с местоположениями, поскольку они гарантируют, что каждый город имеет уникальное название в пределах определенной страны. Это помогает избежать путаницы и ошибок при обработке данных о городах и странах.

4. **Роли и безопасность:** Таблица `roles` представляет собой важный аспект системы управления пользователями. Она позволяет определить различные роли пользователей, такие как администраторы, организаторы и артисты. Путем привязки пользователей к соответствующим ролям и использования связи через таблицу `user_role` обеспечивается контроль доступа и безопасность. Это позволяет надежно управлять разрешениями и ограничениями для каждой роли пользователей в системе.
5. **Связи между сущностями:** БД содержит связи между различными сущностями, такими как организаторы, артисты и события. Например, таблица `organizers` связана с таблицей `users` через внешний ключ `user_id`, что позволяет связывать организаторов с конкретными пользователями. Аналогично, таблица `artists` связана с таблицей `users`, что позволяет связывать артистов с соответствующими пользователями. Это обеспечивает структурированность и связность данных, а также упрощает выполнение запросов и операций связанных с этими сущностями.
6. **Управление продажей билетов:** Таблицы `event_tickets_types`, `sold_tickets` и `used_tickets` позволяют эффективно управлять продажей и использованием билетов на события. С помощью этих таблиц можно отслеживать типы билетов, их стоимость, доступное количество и статус оплаты. Также предусмотрены связи с другими таблицами, такими как `events`, `users` и `payment_statuses`, что позволяет следить за покупкой, использованием и оплатой билетов, а также управлять этими процессами.

Разработанные базы данных предоставляют надежную основу для хранения, управления и обработки данных, необходимых для функционирования мобильного приложения. Они обеспечивают целостность, связность и безопасность данных, а также предоставляют механизмы для эффективного

управления бизнес-процессами, связанными с продажей билетов, управлением пользователями и другими сущностями приложения.

Далее была разработана серверная часть приложения, которая обрабатывает запросы от клиентов и осуществляет продажу билетов. Это включало разработку API с использованием Spring Boot и механизма безопасности на основе JWT. Наконец, было создано мобильное приложение, которое предоставляет удобный интерфейс для пользователей. XML-разметка интерфейса была использована для описания визуального вида приложения, а Java-код отвечал за взаимодействие с серверной частью и обработку данных.

Как показано в коде ниже, был создан триггер `sold_tickets_trigger`, который активируется перед каждой операцией вставки новой записи в таблицу `sold_tickets`.

```
1 CREATE OR REPLACE FUNCTION sold_tickets_trigger()
2 RETURNS TRIGGER AS $$
3 DECLARE
4     tickets_value_max INTEGER;
5     tickets_value_now INTEGER;
6 BEGIN
7     tickets_value_max := (SELECT tickets_value FROM
8     event_tickets_types WHERE event_tickets_types.ticket_type_id
9     = NEW.ticket_type_id);
10    tickets_value_now := (SELECT COUNT(*) FROM sold_tickets
11    WHERE sold_tickets.ticket_type_id = NEW.ticket_type_id);
12    IF tickets_value > tickets_value THEN
13        RETURN NEW;
14    ELSE
15        RAISE EXCEPTION 'Tickets out of stock';
16    END IF;
17
18 END;
19 $$ LANGUAGE plpgsql;
20
21 CREATE TRIGGER sold_tickets_trigger
22 BEFORE INSERT ON sold_tickets
23 FOR EACH ROW
24 EXECUTE FUNCTION sold_tickets_trigger();
```

Триггер связан с функцией `sold_tickets_trigger()`, которая реализует логику проверки наличия билетов перед их продажей. В триггере используется оператор `IF` для сравнения количества доступных билетов (полученного с помощью подзапросов) с требуемым количеством билетов для продажи. Если количество доступных билетов меньше или равно требуемому количеству, триггер выбрасывает исключение `Tickets out of stock` и операция вставки записи отклоняется.

Триггер `sold_tickets_trigger` предназначен для обеспечения целостности данных и предотвращения продажи билетов, когда их количество недостаточно. **Заключение.** Как результат выполнения данной бакалаврской работы было разработано приложение для продажи билетов на мероприятия. В процессе выполнения работы были достигнуты следующие результаты:

- Были проанализированы современные технологии разработки мобильных приложений и выбраны наиболее подходящие для данного проекта.
- Была спроектирована ИС для дистрибуции билетов, которая обеспечивает быструю и удобную покупку билетов, а также возможность продажи билетов между пользователями.
- Была спроектирована БД для системы, которая обеспечивает эффективное хранение и обработку информации о билетах и пользовательских аккаунтах.
- Была реализована ИС, которая была успешно протестирована на соответствие функциональным требованиям и отвечает всем заданным критериям качества.

Разработанное приложение, позволит увеличить продажи билетов на мероприятия, за счет того что, что пользователи могут следить за своими любимыми артистами и за счет простоты покупки билетов. А также предлагает талантливым людям, без опыта организации мероприятий, без лишних усилий организовать продажу билетов и проход на мероприятие, сосредоточившись над главной темой мероприятия.

Таким образом, цели, поставленные в работе, были полностью достигнуты.