

Содержание

Введение.....	3
1 Анализ предметной области.....	6
2 Проектирование приложения	8
3 Разработка программного обеспечения.....	15
3.1 Описание технологического стека разработки	15
3.2 Описание алгоритма работы	16
3.3 Описание интерфейса пользователя	18
4 Тестирование веб-приложения.....	20
4.1 План тестирования	16
4.2 Оценка результатов проведения тестирования	18
Заключение.....	22
Список использованных источников.....	24

					ОКЭИ 09.02.07. *024 8				
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпис</i>	<i>Дата</i>					
<i>Разраб.</i>		<i>Исаева Д.С.</i>			Отчет по курсовой работе		<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>		<i>Адамович Н.В</i>						2	
<i>Реценз.</i>							Отделение - очное гр. 4ВБ2		
<i>Н. Контр.</i>									
<i>Утверд.</i>									

Введение

Интернет и веб-технологии прошли долгий путь с момента их изобретения. От первых электронных писем и простых веб-страниц мы пришли к тому, что имеем сегодня – к сложным веб-приложениям, мобильным приложениям и социальным сетям.

Изначально интернет использовался только для обмена данными между учеными и исследовательскими центрами. С течением времени он стал доступен для всех, и сейчас мы используем его для общения, поиска информации, покупки товаров и многого другого.

Веб-сайты – это ресурсы, расположенные в интернете и доступные для просмотра через браузер. Они могут содержать различные элементы, такие как текст, изображения, видео, аудио и другие. Веб-сайты используются для различных целей, таких как предоставление информации, продажа товаров, общение и другие.

В современном динамичном мире ресторанной индустрии эффективная коммуникация между компаниями-поставщиками (продукты, оборудование, услуги) и ресторанами играет ключевую роль в успехе обеих сторон. Традиционные методы коммуникации, такие как телефонные звонки и электронная почта, часто оказываются недостаточно эффективными, приводя к задержкам, недопониманиям и потере времени. В условиях растущей конкуренции и необходимости оперативного реагирования на изменения рынка, потребность в оптимизированной и прозрачной системе коммуникации становится особенно актуальной. Данная курсовая работа посвящена разработке информационного веб-портала, призванного улучшить взаимодействие между компаниями и ресторанами, обеспечивая своевременный обмен информацией, повышение эффективности сотрудничества и укрепление деловых отношений.

Портал может значительно упростить процесс коммуникации между компанией и ресторанами. Это особенно актуально в условиях, когда скорость и точность передачи информации критически важны для принятия решений.

На сегодняшний день наблюдается значительный рост числа ресторанов и связанных с ними предприятий, что, в свою очередь, стимулирует конкуренцию на рынке. Успех ресторана все больше зависит не только от качества кухни и обслуживания, но и от эффективного управления ресурсами, включая закупку продуктов, оборудования и привлечение новых клиентов. Для компаний в свою очередь, крайне важно поддерживать тесные контакты с ресторанами, обеспечивая своевременную доставку товаров, предоставление качественного сервиса и гибкое реагирование на запросы клиентов. Отсутствие эффективной коммуникации приводит к следующим негативным последствиям:

– потеря времени и ресурсов: запросы и ответы, проходящие через множество посредников, затягивают процесс и увеличивают затраты на коммуникацию;

- недопонимание и ошибки: неясная или неполная информация может привести к ошибкам в заказах, поставках и других аспектах сотрудничества;
- уменьшение прибыли: задержки и недоразумения приводят к снижению эффективности работы ресторанов и уменьшению прибыли как для них самих, так и для компаний-поставщиков;
- ухудшение деловых отношений: негативный опыт коммуникации может нанести вред долгосрочным отношениям между компаниями и ресторанами.

Предлагаемый информационный веб-портал направлен на решение вышеупомянутых проблем, предоставляя комплексную платформу для эффективного взаимодействия между компаниями и ресторанами

Внедрение такого портала позволит компаниям и ресторанам значительно сократить время, затрачиваемое на коммуникацию, снизить вероятность ошибок, повысить эффективность сотрудничества и укрепить деловые отношения. В конечном итоге, это приведет к росту прибыли и конкурентоспособности как ресторанов, так и компаний-поставщиков. Данная курсовая работа представляет собой исследование и разработку такого портала, анализ его потенциальной эффективности и описание архитектуры и функциональных возможностей. В ходе работы будут рассмотрены вопросы выбора технологий, разработки пользовательского интерфейса, обеспечения безопасности и масштабируемости.

Разработка информационного веб-портала как средства оптимизации коммуникации между компаниями и ресторанами – это актуальная задача, направленная на повышение эффективности работы обеих сторон и создание благоприятной среды для развития бизнеса в ресторанной индустрии.

Одним из основных средств создания web-страниц являются HTML-теги. HTML (HyperText Markup Language) – это язык разметки гипертекста, который позволяет структурировать текст и элементы на web-странице. С помощью HTML-тегов можно определить, как будет выглядеть текст, какие элементы будут отображаться, как будут располагаться элементы на странице и т.д.

Объектом исследования является web-программирование.

Предметом исследования является разработка информационного портала для коммуникации между компанией и ресторанами.

Целью данной курсовой работы является разработка концепции информационного портала, который будет служить связующим звеном между компаниями и ресторанами. Такой портал может включать в себя функционал для обмена информацией, упрощения процессов заказа, предоставления актуальных новостей и специальных предложений, а также возможность обратной связи.

Для достижения поставленной цели необходимо решение следующих основных задач:

- проведение анализа исходных данных;
- составить техническое задание на разработку сайта;
- разработать дизайн сайта с учетом удобства навигации, информативности, привлекательности и простоты использования;
- описание хода разработки веб-сайта;
- создание веб-сайта;

- разработка сайта с использованием современных программных средств;
- создание базы данных для сайта;
- проведение тестирования веб-сайта;
- оценить эффективность разработанного сайта.

Выбор темы «Информационный портал для коммуникации между компанией и ресторанами» обусловлен ее практической значимостью и актуальностью в условиях современного рынка. Разработка подобного портала представляет собой сложную, но высоко востребованную задачу, решение которой может существенно улучшить бизнес-процессы в ресторанной индустрии. Данная работа позволит углубить знания в области веб-разработки, проектирования баз данных и управления проектами, а также получить практический опыт в создании реального веб-приложения.

Таким образом, данный проект позволит не только установить эффективные каналы коммуникации между компаниями и ресторанами, но и создать платформу для развития бизнеса, улучшения качества предоставляемых услуг и удовлетворенности клиентов.

1 Анализ предметной области

С развитием технологий и цифровых платформ веб-сайты становятся неотъемлемой частью бизнеса, особенно в сфере общепита. Одной из актуальных задач в данной области является создание эффективного информационного портала, который будет служить связующим звеном между компанией (например, поставщиком продуктов или услуг) и ресторанами. Такой портал должен обеспечивать удобный доступ к информации о новостях, изменениях в правилах и стандартах, а также предоставлять другие важные данные, способствующие успешной интеграции бизнес-процессов.

Предметная область включает двух ключевых участников: компанию и рестораны. Их потребности и цели во взаимодействии формируют основу для функциональных требований портала.

Компания: компания нуждается в эффективном инструменте для распространения информации среди ресторанов, контроля соблюдения стандартов и оперативного реагирования на запросы. Ключевые потребности включают:

- централизованное управление информацией: возможность легко публиковать и обновлять новости, правила, стандарты и другую документацию;
- контроль доступа: управление правами доступа разных пользователей внутри Компании (разделение на администратора, директоров и пользователей);
- эффективная обратная связь: возможность получать вопросы и отзывы от ресторанов, оперативно на них реагировать;
- интеграция с существующими системами.

Рестораны: рестораны нуждаются в удобном и оперативном доступе к актуальной информации, необходимой для эффективной работы. Ключевые потребности включают:

- быстрый и легкий доступ к информации: удобный интерфейс для поиска и просмотра новостей, правил, стандартов и другой документации;
- архив документов: доступ к истории публикаций для поиска необходимой информации в любое время;
- простота использования: интуитивно понятный интерфейс, не требующий специальной подготовки.

Предметная область включает следующие ключевые сущности и их взаимосвязи:

- два типа пользователей: администратор (Компания) и пользователи ресторанов;
- новости: текстовые сообщения с возможностью добавления изображений, содержащие актуальную информацию для ресторанов;
- правила: документы, регламентирующие взаимоотношения между Компанией и Ресторанами (например, контрактные условия, требования к качеству);
- стандарты: документы, описывающие требования к качеству продукции, услугам или процессам;

– документы: общий термин, охватывающий новости, правила и стандарты. Может включать различные типы файлов (PDF, DOCX, JPG и т.д.);

– категории: для структурирования новостей, правил и стандартов по тематическим рубрикам;

– роли и права доступа: система управления правами доступа пользователей к различным функциям портала.

Функциональные требования для компании, администратора и директоров ресторанов:

– добавление, редактирование и удаление новостей (с возможностью загрузки изображений);

– добавление, редактирование и удаление правил и стандартов (с возможностью загрузки файлов);

– управление пользователями (добавление, редактирование, удаление, назначение ролей);

– просмотр статистики использования портала (количество посещений, просмотров новостей и т.д.).

Для авторизированных пользователей:

– просмотр новостей, правил и стандартов;

– возможность поиска и фильтрации информации по ключевым словам, дате публикации и типу документа.

Нефункциональные требования:

– надежность: портал должен быть устойчив к отказам и обеспечивать бесперебойную работу;

– производительность: портал должен быстро загружаться и обрабатывать запросы пользователей;

– масштабируемость: портал должен быть способен обрабатывать большое количество пользователей и данных;

– безопасность: портал должен защищать информацию от несанкционированного доступа;

– удобство использования: UI должен быть интуитивно понятным и удобным для пользователей;

– соответствие стандартам: портал должен соответствовать современным стандартам веб-разработки.

2 Проектирование приложения

В процессе разработки программного обеспечения ключевое значение имеет этап проектирования, на котором определяется архитектура, функциональные и нефункциональные требования к создаваемому приложению. Грамотно выполненное проектирование заложит прочный фундамент для дальнейшего этапа реализации и обеспечит соответствие продукта ожиданиям пользователей и бизнес-требованиям.

Функциональные требования для администратора:

- добавление, редактирование и удаление новостей (с возможностью загрузки изображений). Администратор должен иметь возможность создавать новую новость с указанием заголовка, текста, категории и даты публикации. Для каждой новости предусмотрена функция загрузки изображения. Поддерживаемые форматы должны включать JPEG, PNG и GIF. Администратор должен иметь возможность перемещения новостей в архив, а также полного удаления, с подтверждением действия, чтобы избежать случайного удаления;

- добавление, редактирование и удаление правил и стандартов (с возможностью загрузки файлов). Возможность создания новых документов, содержащих правила и стандарты с указанием заголовка, текста и даты публикации. Функция загрузки файлов (например, PDF, DOCX) для дополнительных материалов, связанных с правилами и стандартами. Возможность перемещения документов в архив или полного удаления с подтверждением, чтобы предотвратить случайные действия;

- управление пользователями (добавление, редактирование, удаление, назначение ролей). Администратор может создать нового пользователя, указав необходимые данные: имя, фамилию, email, пароль и роль (например, пользователь, модератор, редактор).

Для авторизованных пользователей:

- просмотр новостей, правил и стандартов. Пользователи смогут просматривать актуальные новости, а также ознакомиться с действующими правилами и стандартами. Это поможет им быть в курсе важных обновлений и изменений;

- возможность поиска и фильтрации информации по ключевым словам, дате публикации и типу документа. Возможность эффективно искать и фильтровать информацию по ключевым словам, дате публикации и типу документа. Это упростит навигацию и поможет быстро находить нужные материалы;

- предложение новостей: пользователи смогут предлагать свои новости или события для публикации. Это дает возможность каждому внести вклад в информационное пространство и делиться важными событиями с другими пользователями.

Важным аспектом является описание функциональных требований, которые можно визуализировать с помощью диаграммы прецедентов (use case diagram). Эта

диаграмма позволяет наглядно представить взаимодействие пользователей с системой и выявить ключевые сценарии использования приложения (Приложение Б). Прецеденты используются для описания конкретных функций или задач, которые система должна выполнять. Они представляют собой сценарии взаимодействия акторов с приложением и помогают понять, какие функции являются критически важными для пользователей.

Нефункциональные требования:

- надежность: портал должен быть способен выдерживать различные виды сбоев, таких как отключение сервера или проблемы с сетью, без потери доступности для пользователей. Важно реализовать механизмы резервного копирования и восстановления данных, а также обеспечить безотказную работу ключевых функций. Проведение регулярного мониторинга состояния системы для быстрого реагирования на возможные проблемы;

- производительность: время загрузки страниц должно составлять не более 2-3 секунд для оптимального пользовательского опыта. Портал должен эффективно обрабатывать запрашиваемые данные, минимизируя задержки и время отклика на действия пользователя. Наличие механизма кэширования для ускорения загрузки и обработки часто запрашиваемой информации;

- масштабируемость: портал должен быть способен обработать увеличенное количество пользователей и транзакций без ухудшения производительности. Необходимо предусмотреть возможность расширения инфраструктуры (например, добавление серверов или увеличение ресурсов) при росте нагрузки. Изоляция компонентов системы для упрощения обновлений и интеграции новых функций;

- безопасность: все данные, передаваемые между пользователями и порталом, должны быть защищены с помощью шифрования (например, использование HTTPS). Реализация многоуровневой аутентификации и авторизации для защиты от несанкционированного доступа. Проведение регулярных аудитов безопасности и обновления системы для защиты от новых уязвимостей;

- удобство использования: интерфейс должен быть интуитивно понятным, позволяя пользователям легко находить необходимую информацию и выполнять действия. Должны быть предусмотрены адаптивные элементы интерфейса для обеспечения возможности работы с порталом на различных устройствах (мобильных, планшетах, ПК). Регулярные пользовательские тестирования для выявления и устранения проблем с удобством использования интерфейса;

- соответствие стандартам: портал должен быть разработан с соблюдением современных стандартов веб-разработки, включая семантическую разметку, доступность (например, WCAG), и лучшие практики безопасности. Необходима кроссбраузерная совместимость, чтобы все пользователи могли получить доступ к функциональности портала независимо от используемого браузера. Следует учитывать требования к локализации и интернационализации, чтобы обеспечить поддержку различных языков и культур.

3 Разработка программного обеспечения

3.1 Описание технологического стека разработки

Технологический стек разработки, состоит из HTML, CSS, JavaScript, SQLite, VisualStudioCode и Figma. Они представляет собой мощный набор инструментов для создания веб-приложений.

HTML — это стандартный язык разметки для создания веб-страниц. Он позволяет структурировать содержание, используя теги для определения заголовков, параграфов, списков, изображений, ссылок и других элементов. HTML — это основа любого веб-документа и обеспечивает семантическую структуру.

Преимущества HTML:

- структурирование контента: HTML позволяет организовать текст, изображения, видео и другие элементы на веб-странице, создавая логическую структуру;
- простота изучения: HTML обладает простой и понятной синтаксической структурой, что делает его доступным для новичков;
- совместимость: HTML поддерживается всеми современными веб-браузерами и устройствами, что обеспечивает высокую степень совместимости.
- доступность: HTML позволяет создавать доступные веб-страницы, которые легко читаются программами для людей с ограниченными возможностями;
- интерактивность: HTML поддерживает использование различных типов медиа и форм, что делает страницу интерактивной.

CSS — это язык стилей, который используется для оформления и визуального представления HTML-документов. С помощью CSS можно задавать цвета, шрифты, отступы, размеры и многие другие стилистические параметры. CSS также позволяет адаптировать дизайн под различные устройства и экраны, используя медиа-запросы.

Преимущества CSS:

- разделение содержания и оформления: CSS позволяет отделить внешний вид веб-страницы от её структуры, что упрощает обновление и поддержку сайта;
- управление стилями: с помощью CSS можно легко изменять шрифты, цвета, отступы, размеры и другие стили элементов на странице;
- адаптивный дизайн: CSS позволяет создавать адаптивные и отзывчивые дизайны, что обеспечивает корректное отображение сайта на разных устройствах (мобильных, планшетах, десктопах);
- кросс-браузерная совместимость: CSS стили хорошо поддерживаются в большинстве современных браузеров, что позволяет создавать единый стиль на разных платформах;
- переиспользование стилей: CSS дает возможность создавать стили один раз и применять их к нескольким элементам или страницам, что сокращает объем кода и упрощает его поддержку;

– анимации и переходы: CSS поддерживает создание анимаций и эффектов перехода, что может значительно улучшить пользовательский интерфейс и восприятие сайта.

SQLite — это легковесная СУБД (система управления базами данных), которая используется для хранения данных в виде файлов. Она не требует установки на сервер и подходит для разработки небольших приложений и прототипов. SQLite идеально вписывается в проекты, где необходима простая и быстрая работа с данными без сложной архитектуры.

Преимущества SQLite:

– легковесность: SQLite – это встроенная база данных, которая не требует отдельного серверного процесса. Вся база данных хранится в одном файле, что делает ее лёгкой в развертывании и использовании;

– простота использования: SQLite имеет простой и понятный интерфейс. Пользователи могут начать работать с ней без необходимости в сложной настройке;

– поддержка ACID: SQLite поддерживает свойства ACID (Атомарность, Согласованность, Изолированность, Долговечность), что гарантирует надежную работу с данными;

– кросс-платформенность: SQLite доступна на различных платформах и может работать на любом устройстве, которое поддерживает файловую систему;

– отсутствие необходимости в установке: так как SQLite является встроенной БД, нет необходимости в установке и управлении сервером баз данных;

– хорошая производительность: для небольших и средних приложений SQLite обеспечивает хорошую производительность при обработке запросов;

– поддержка SQL: SQLite использует стандартный язык SQL для взаимодействия с данными, что облегчает переход от других СУБД.

JavaScript — это язык программирования, который позволяет добавлять интерактивность на веб-страницы. С его помощью можно обрабатывать события, управлять элементами DOM (Document Object Model), делать асинхронные запросы к серверу (например, через AJAX) и многое другое. JavaScript широко используется для создания динамических и отзывчивых интерфейсов.

Преимущества JavaScript:

– кросс-платформенность: JavaScript работает на всех современных браузерах и может быть использован как на клиентской, так и на серверной стороне (с помощью Node.js);

– асинхронное программирование: возможность выполнения асинхронных операций позволяет улучшать отзывчивость приложений, что делает его идеальным для разработки веб-приложений;

– обширная экосистема: существует множество библиотек и фреймворков, таких как React, Angular и Vue.js, которые упрощают разработку сложных интерфейсов;

– удобство работы с данными: JavaScript хорошо подходит для работы с JSON, что упрощает взаимодействие с RESTful API и другими формами обмена данными;

– динамическая типизация: JavaScript использует динамическую типизацию, что позволяет разрабатывать код быстрее, хотя и требует внимательности к типам.

– поддержка событий: JavaScript отлично справляется с обработкой событий, что позволяет создавать интерактивные и отзывчивые пользовательские интерфейсы;

– сообщество и поддержка: огромное сообщество разработчиков и регулярные обновления означают, что у вас всегда будет доступ к последним технологиям и поддержке.

VS Code — это популярный редактор кода, который поддерживает множество языков программирования и предоставляет многофункциональные инструменты для разработчиков. Он включает в себя поддержку расширений, подсветку синтаксиса, автозаполнение, отладку и интеграцию с системами контроля версий, такими как Git. VS Code является удобной средой для написания кода на HTML, CSS и JavaScript.

Figma — это облачный инструмент для дизайна интерфейсов и прототипирования. С его помощью дизайнеры могут создавать макеты веб-страниц, мобильных приложений и других продуктов. Figma позволяет командам совместно работать над проектами в реальном времени и предоставляет мощные инструменты для создания интерактивных прототипов, что облегчает процесс разработки.

Совместное использование этих технологий позволяет создать полноценное веб-приложение:

- HTML используется для создания структуры страницы;
- CSS оформляет страницу, обеспечивая привлекательный вид;
- JavaScript добавляет интерактивность, позволяя пользователям взаимодействовать с приложением;
- SQLite обеспечивает хранение данных, которые могут быть использованы в приложении;
- VS Code служит инструментом для разработки и написания кода;
- Figma помогает в проектировании пользовательского интерфейса и создании прототипов.

3.2 Описание алгоритма работы

Разработка программного обеспечения (ПО) — это процесс создания и поддержки программных приложений, который включает несколько этапов. Один из основных аспектов разработки ПО — это грамотное проектирование алгоритмов.

Алгоритм работы для информационного портала коммуникации между сотрудниками и ресторанами может быть следующим:

- пользователь авторизуется на сайте и попадает на главную страницу;
- пользователь может просматривать новости как своей так и другой компании;
- пользователь может просматривать общие коммуникации для всех ресторанов;
- пользователь может посмотреть доступные или прошедшие мероприятия;
- пользователь сам может заполнить форму и предложить новость.

Хранение данных в приложении будет реализовано с использованием базы данных для хранения информации о мероприятиях, пользователях, коммуникации, новостей. Для этого планируется использовать реляционную базу данных – SQLite.

Так как на сайте и на странице администратора представлены формы для отправки данных в бд, они реализованы следующим образом:

- создание пользовательского интерфейса: на первой стадии мы разработали HTML-форму, которая содержит поля для ввода данных и кнопку для отправки данных в бд;

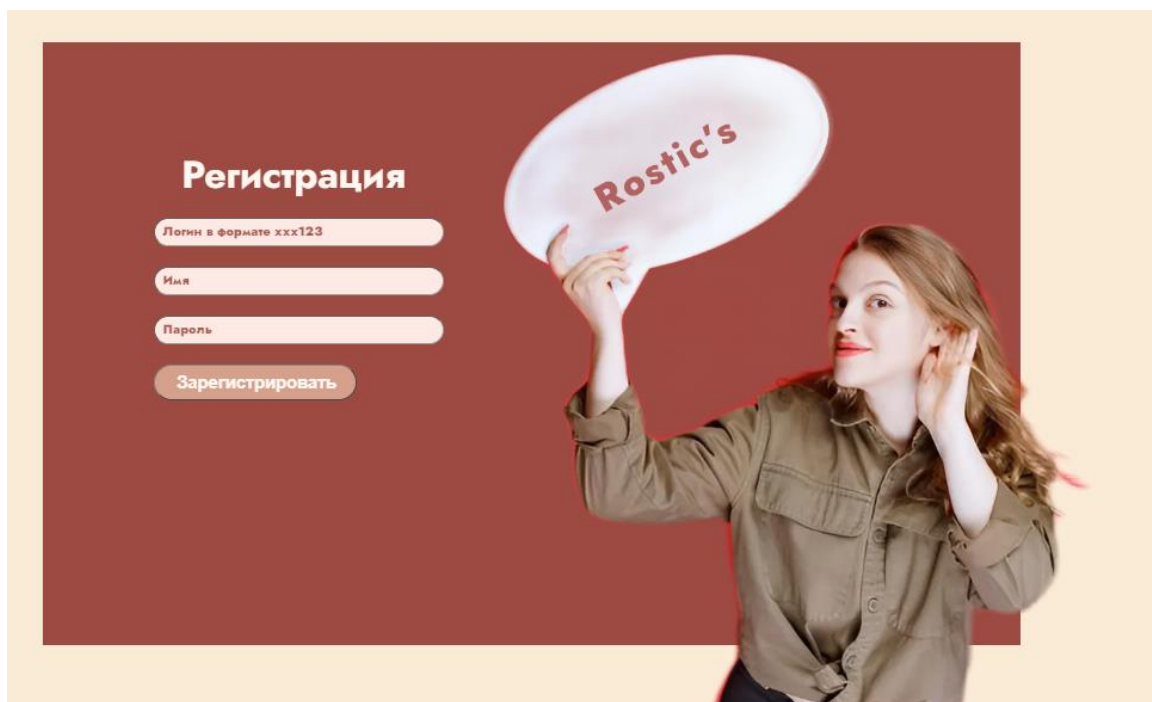


Рисунок 1 – Форма для добавление пользователей

- сбор данных из формы: после того как пользователь заполнит форму и нажмет кнопку отправки, JavaScript используется для захвата данных, введенных в полях формы. Этот процесс включает в себя проверку корректности данных перед отправкой;
- отправка данных на сервер: для передачи собранных данных на сервер используется AJAX-запрос (обычно через fetch или XMLHttpRequest). Данные отправляются в формате JSON, который удобен для обработки на сервере;

– обработка данных на сервере: на серверной стороне создается обработчик, который принимает данные от клиента. Этот обработчик может быть реализован на Node.js, Python или другом языке программирования, поддерживающем работу с SQLite. Здесь осуществляется подключение к базе данных и выполнение SQL-команды для вставки полученных данных в соответствующие таблицы;

– работа с базой данных SQLite: после выполнения SQL-команды данные сохраняются в базе данных SQLite. Обработчик возвращает ответ клиенту, подтверждая успешное добавление данных или сообщая об ошибке;

```
1  const express = require("express");
2  const bodyParser = require("body-parser");
3  const session = require("express-session");
4  const { Sequelize, DataTypes } = require("sequelize");
5  const bcrypt = require("bcrypt");
6  const path = require("path");
7
8  const app = express();
9  const port = 3000;
10
11 // Настройка базы данных SQLite и подключение к ней
12 const sequelize = new Sequelize({
13   dialect: "sqlite",
14   storage: "db.sqlite",
15 });
```

Рисунок 2 – Подключение к БД

– получение и отображение данных для администратора: для вывода данных из базы данных на страницу администратора создается еще один AJAX-запрос, который запрашивает данные из таблиц. Сервер обрабатывает этот запрос, извлекает нужные данные из базы и отправляет их обратно клиенту;

Рисунок 3 – Код отображения данных на странице

– отрисовка данных на странице (рисунок 4): полученные данные отображаются на странице администратора с помощью JavaScript. Это может быть сделано путем динамического создания HTML-элементов или использования библиотек для работы с таблицами (например, DataTables), что позволяет отображать и форматировать данные в удобном для восприятия виде.

Рисунок 4 – Отображение таблицы на странице администратора

Диаграмма классов (Приложение Г) представляет собой статическую модель проектируемого информационного портала для коммуникации между компанией-поставщиком и сетью ресторанов. Она визуализирует структуру системы, отображая ключевые классы, их атрибуты, методы и взаимосвязи. Диаграмма служит основой для дальнейшей разработки программного обеспечения, обеспечивая ясное понимание архитектуры и структуры данных. В ней представлены основные сущности предметной области и их взаимоотношения, что позволяет оценить целостность и корректность модели.

3.3 Описание интерфейса пользователя

Приложение предоставляет авторизованным пользователям доступ к новостям, важной информации (коммуникациям), форме обратной связи, расписанию мероприятий, информации о компании и её контактными данными.

Первое что видит пользователь заходя на сайт, это страница авторизации, где пользователь должен ввести свои данные, которые предоставляет компания для входа на сайт.

Форма авторизации представляет поле «Логин» – текстовое поле для ввода личного неповторяющегося имени пользователя. Валидация на стороне клиента: обязательное поле. Поле «Имя» – текстовое поле для ввода обычного имени пользователя. Поле «Пароль» – поле для ввода пароля. Тип поля: password для скрытия вводимых символов. Валидация на стороне клиента: обязательное поле. Кнопка «Войти» Активируется после заполнения обоих полей. При нажатии отправляет данные формы на сервер для проверки.

После авторизации пользователь попадает на главную страницу, на которой представлены: шапка сайта (рисунок 5) – логотип компании: при клике перенаправляет на главную страницу; ссылки на остальные страницы сайта ;кнопка «Выход»: расположена справа. При нажатии завершает сеанс пользователя и перенаправляет на страницу авторизации.

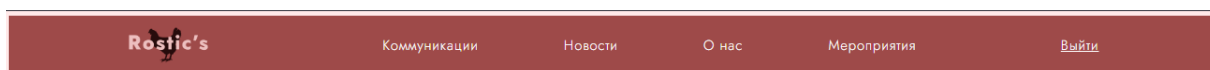


Рисунок 5 – Шапка сайта

Основной контент главной страницы это главная новость или коммуникация компании (рисунок 6). Так же на нем присутствует кнопка «перейти к ценностям», которая перенаправит пользователя на страницу самой новости.

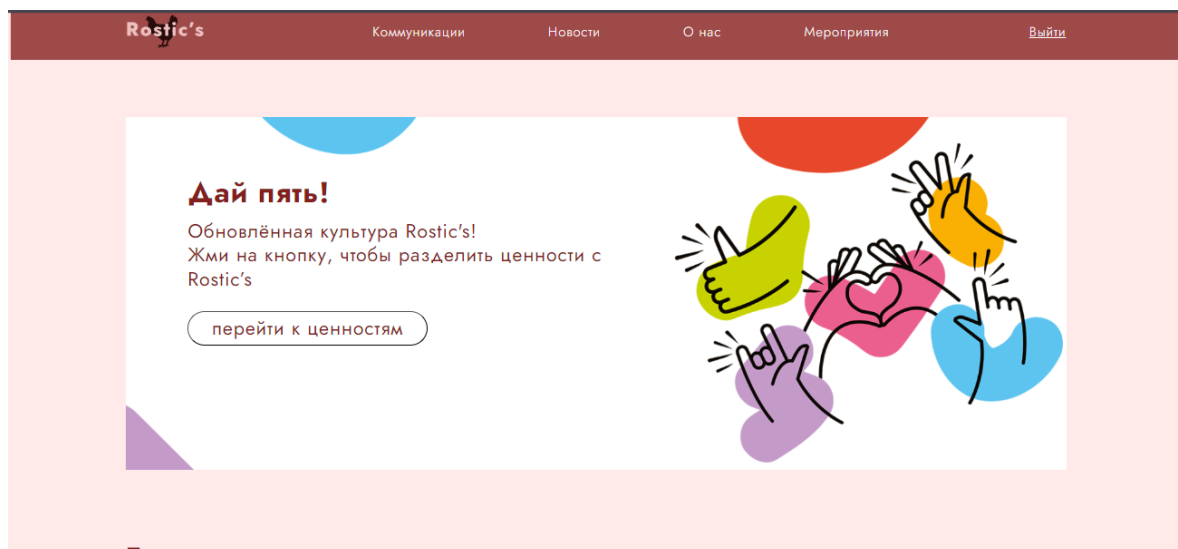


Рисунок 6 – Первый блок главной страницы

Блок «Главное», на котором представлены 3 последние новости компании, каждая новость включает заголовок новости и картинку (Приложение Д).

Блок «О нас» включает краткое описание и историю компании. Контактная информация или подвал сайта предоставляет контактную информацию и реквизиты компании (рисунок 7).

Контакты

Название компании: ООО "ОренРест"

Юр.адрес: 450078, Республика Башкортостан, г. Уфа, ул. Владивостокская д.2

Телефон: 13-34-05

Email: office@grufa.ru

Реквизиты

ИНН: 0278202810

КПП: 027801001

ОГРН: 1130280050196

Организационно-правовая форма: Общество с ограниченной ответственностью

Рисунок 7 – Подвал сайта

Краткая карта сайта представлена на рисунке 8.

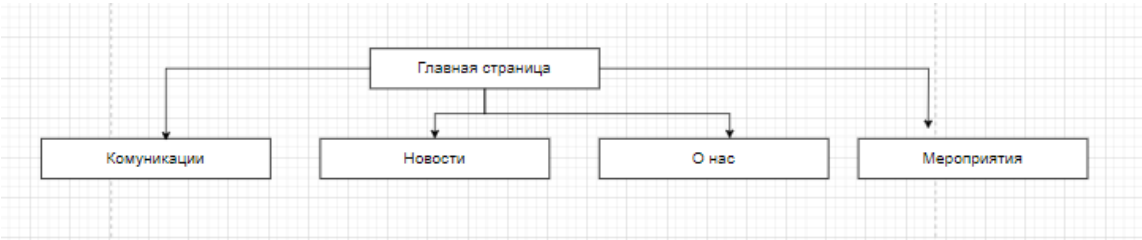


Рисунок 8 – Карта сайта

Также пользователь может оставить обратную связь или задать интересующий вопрос (рисунок 9).

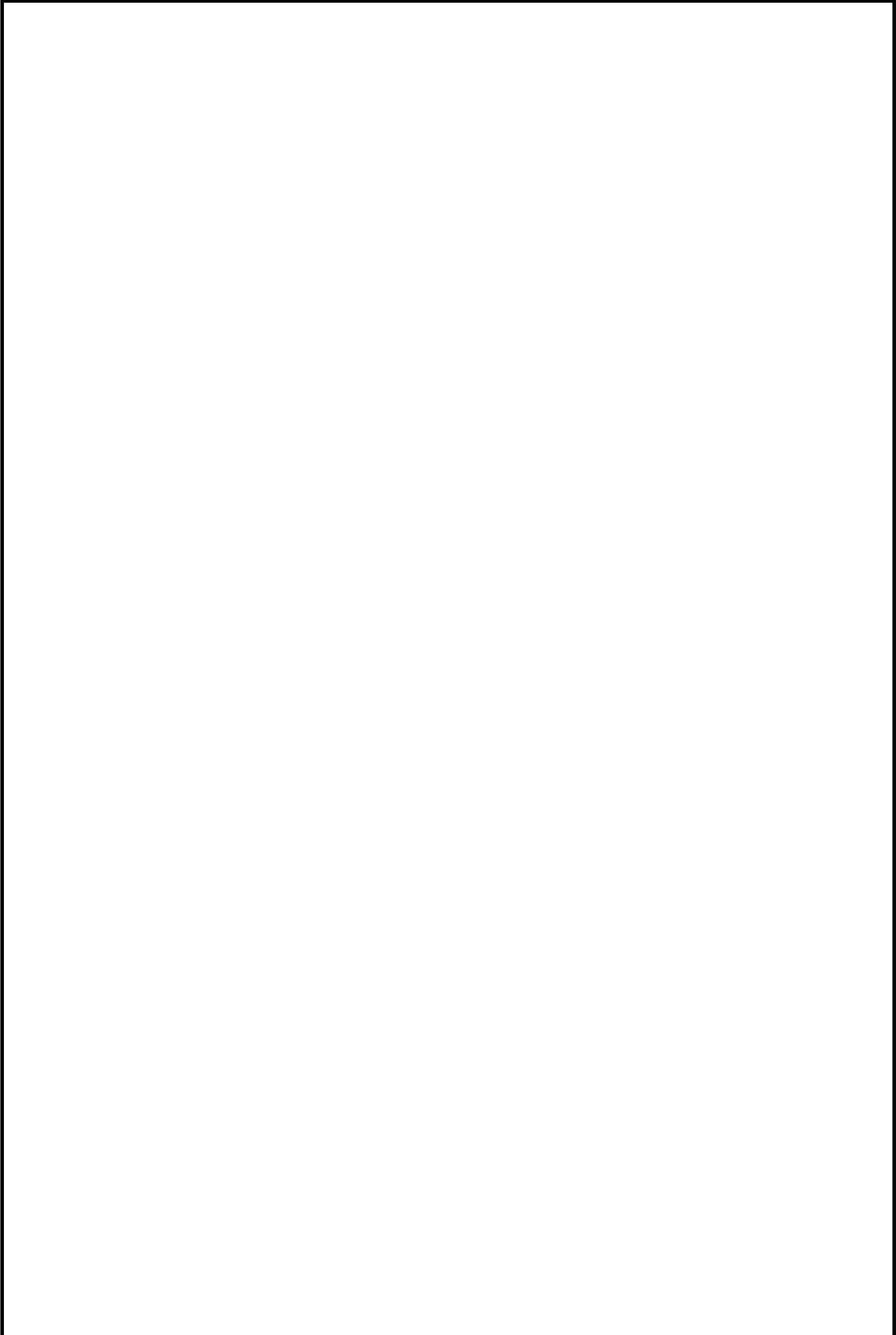
Остались вопросы?

Имя

Email

Ваш вопрос

Рисунок 9 – Форма обратной связи



					<i>ОКЭИ 09.02.07.*024 8 *</i>	Лист
						17
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Приложение А (обязательное)

Информационная модель

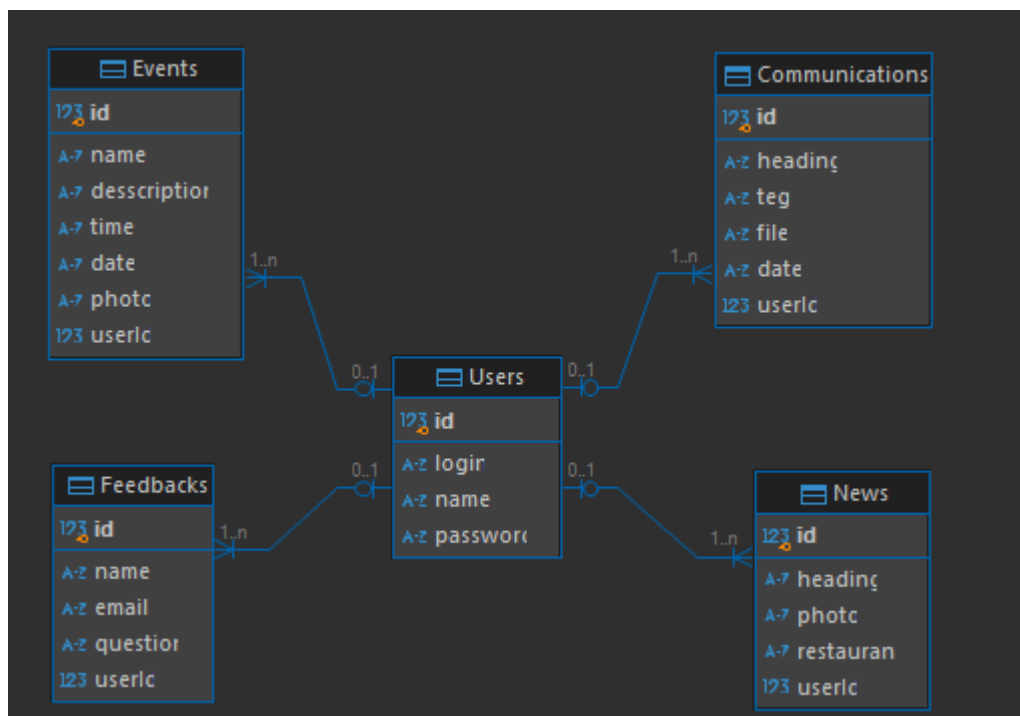


Рисунок А.1 – Информационная модель

Приложение Б (обязательное)

Диаграмма прецедентов

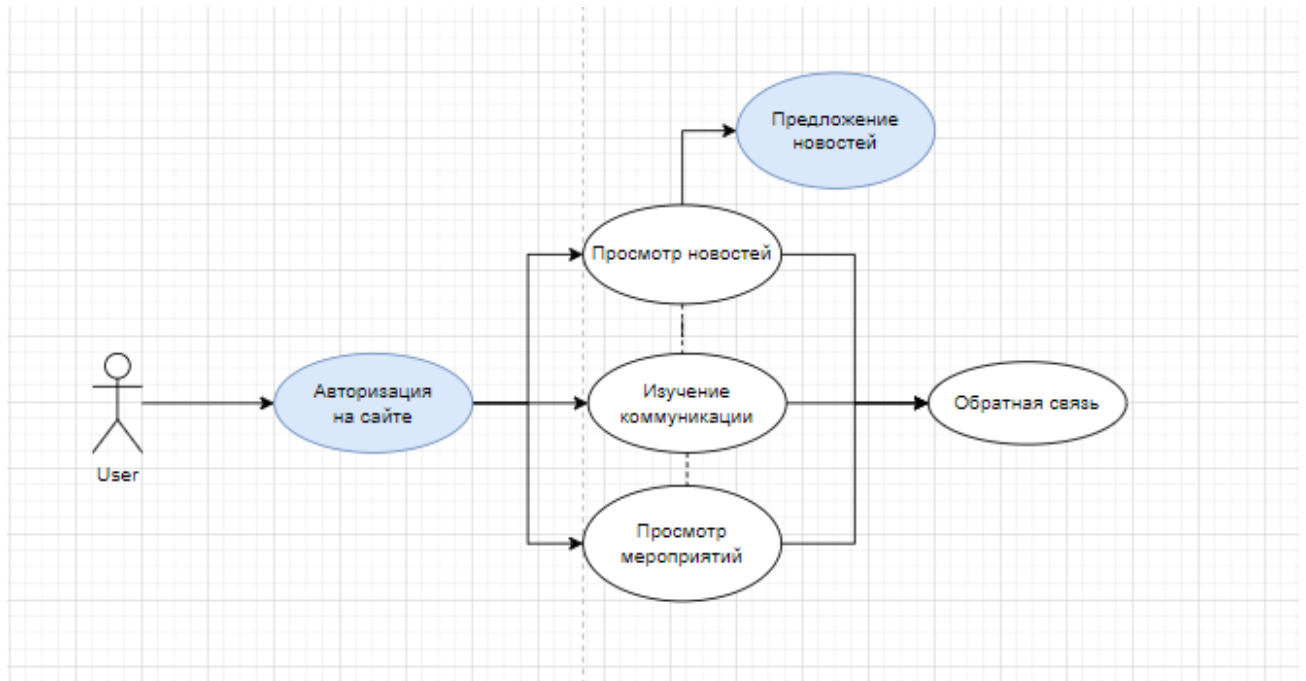


Рисунок Б.1 – Диаграмма прецедентов пользователя

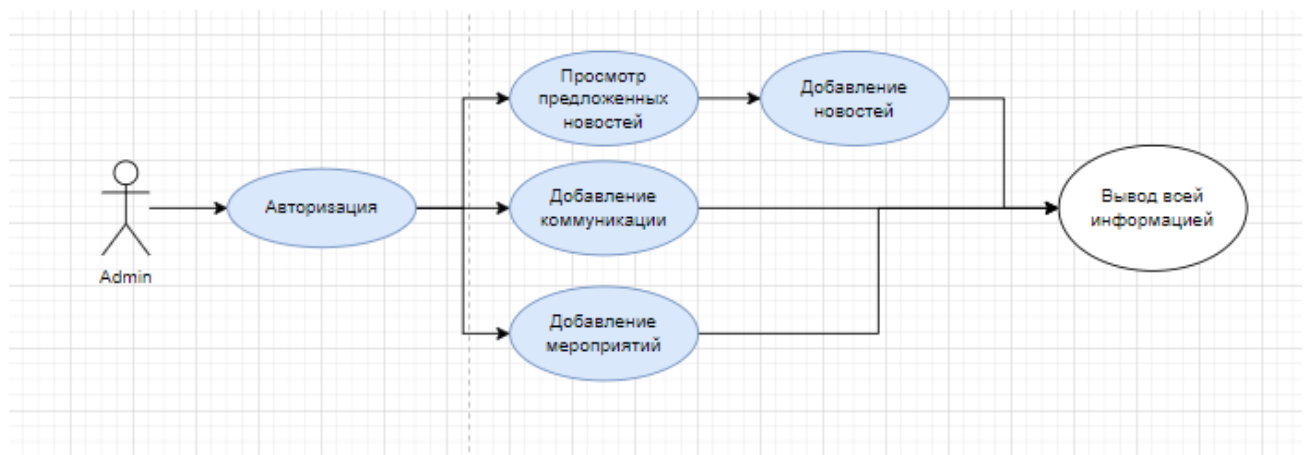


Рисунок Б.2 – Диаграмма прецедентов администратора

Приложение Г (обязательное)

Диаграмма классов

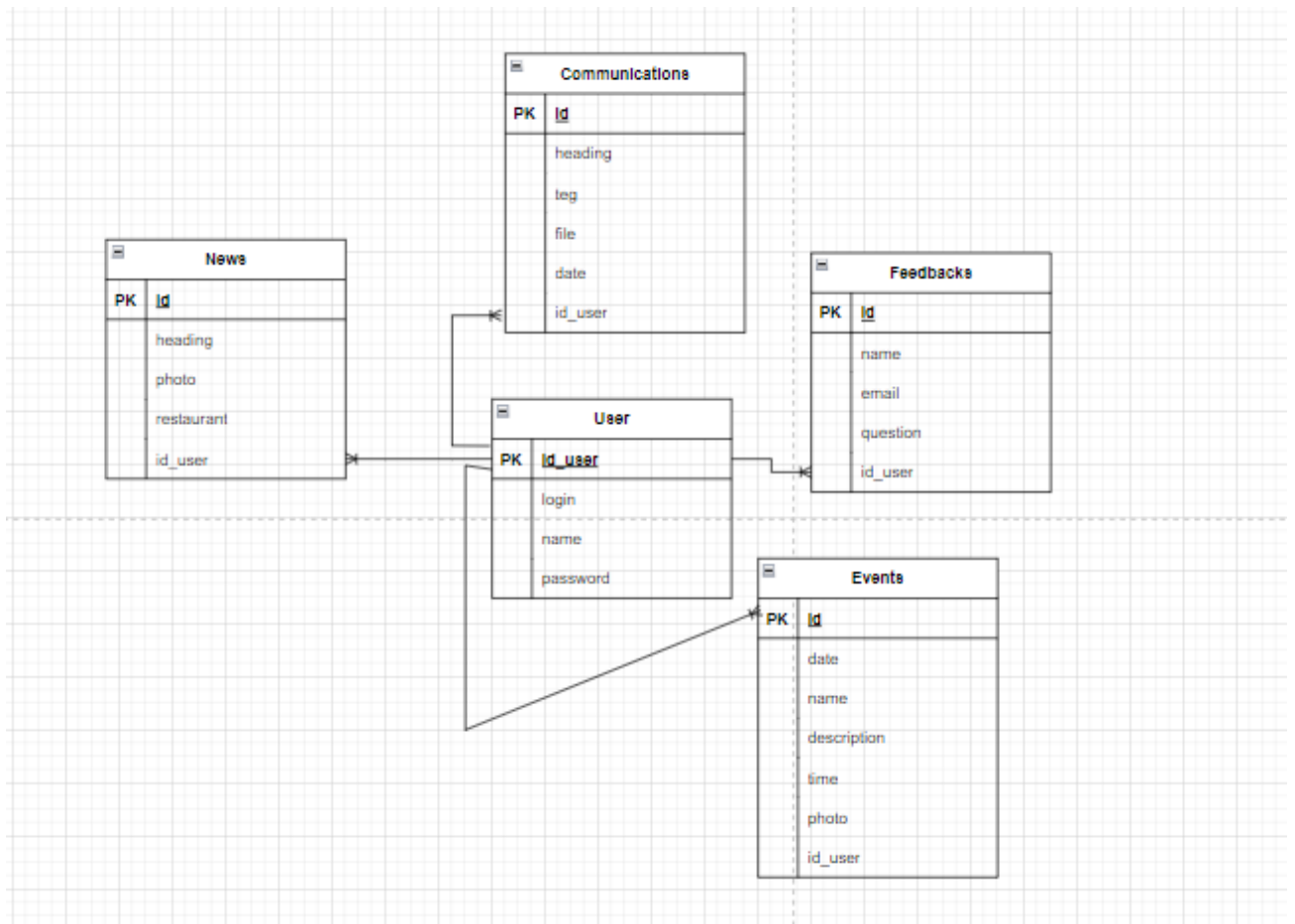


Рисунок Г.1 – Диаграмма классов