Техническое задание на разработку Веб-приложения «Социальная сеть для профессионалов»

1 Введение

1.1 Цель

Целью разработки данной системы является создание платформы для профессионалов, ориентированной на обмен знаниями, развитие сети контактов и интеграцию с профессиональными ресурсами.

1.2 Актуальность и назначение проекта

В условиях современного IT-рынка специалисты в определённых областях часто сталкиваются с проблемой поиска единомышленников и обмена опытом. Данная система позволит:

* упростить поиск профессиональных контактов;
* обеспечить площадку для обмена опытом и решения профессиональных вопросов.

1.3 Основные пользователи системы

Система предназначена для использования сотрудниками компании:

* администратор: управление пользователями, модерирование контента, сбор статистики по количеству зарегистрированных пользователей.
* пользователь: регистрация, авторизация, создание профиля, редактирование данных, подключение репозиториев github, взаимодействие с другими пользователями через чат и форум, просмотр профилей и хакатонов.

2 Цели и задачи проекта

2.1 Основные цели

* создание платформы для профессионалов, которая обеспечит возможность обмена знаниями, развития сети контактов и интеграции с профессиональными ресурсами.

2.2 Конкретные задачи

* разработка функциональности для администратора;
* разработка функциональности для пользователя.

3 Требования к системе

3.1 Функциональные требования

1. Администратор:

* блокировка пользователей;
* удаление пользователей;
* проверка и одобрение материалов;
* сбор статистики активности.

1. Пользователь:

* регистрация и авторизация;
* управление профилем;
* подключение репозиториев github;
* работа с репозиториями (скачивание репозиториев, просмотр коммитов, просмотр файлов репозитория);
* чат, форум, просмотр списка хакатонов.

3.2 Нефункциональные требования

* производительность: быстрая загрузка страниц.

4 Требования к пользовательскому интерфейсу

4.1 Основные экраны

* главная страница: новости, посты.
* профиль: информация о пользователе, навыках, github-репозитории.
* регистрация: форма для регистрации пользователей;
* вход: форма для авторизации пользователей;
* чаты: список чатов, блок с сообщениями;
* форум: вопросы и ответы пользователей, с кнопками «ответить» (для всех пользователей, кроме инициатора вопроса), «посмотреть ответы» (для всех пользователей) и «закрыть вопрос» (для инициатора вопроса);
* пользователи: список пользователей, с информацией о их навыках, при клике на пользователя, открывается его профиль;
* it-хакатоны: список актуальных соревнований в it-индустрии, в которых можно принять участие;
* панель администратора: список пользователей с кнопками «удалить», «заблокировать»;
* модерация контента (новости, посты) «принять», «отклонить»,
* статистика зарегистрированных пользователей в виде диаграммы.

4.2 Навигация

* для пользователя: разделы «главная», «регистрация», «вход», «мой профиль», «чаты», «пользователи», «форум», «it-хакатоны»
* для администратора: разделы «главная», «регистрация», «вход», «мой профиль», «чаты», «пользователи», «форум», «it-хакатоны», «пользователи», «модерация контента», «статистика».

4.3 Юзабилити

* простота интерфейса и доступность меню навигации.

5 Требования к технической реализации

5.1 Языки и технологии

* backend: node.js, mysql2.
* frontend: react.
* база данных: sql.

5.2 Архитектура системы

* хранение данных в реляционной базе данных.

5.3 Интеграции

* внутренние уведомления: react, websocket.

6 Требования к безопасности

6.1 Аутентификация и авторизация

* разграничение прав доступа между администраторами и пользователями.

6.2 Шифрование данных

* хранение паролей: хэширование через bcrypt.

6.3 Политики доступа

* защита от sql-инъекций и xss.

7 Ограничения и допущения

7.1 Технические ограничения

* нет.

7.2 Финансовые ограничения

* бюджет: до 1000 ₽, включая тестирование и документацию.

7.3 Сроки выполнения

* полная реализация: 1,5 месяца.

8 Требования к тестированию и приемке

8.1 Типы тестирования

* функциональное и тестирование безопасности.

8.2 Критерии приемки

* выполнение всех требований и прохождение 75% тестов.

9 Требования к документации

9.1 Пользовательская документация

* руководство пользователя.

9.2 Техническая документация

* описание архитектуры и api.

10 План реализации

10.1 Этапы разработки

1. анализ и проектирование — 2 недели;
2. разработка прототипа — 1 неделя;
3. полноценная разработка и тестирование — 5 недель;
4. внедрение и обучение — 2 недели.

10.2 Сроки выполнения этапов

* начало: 28 октября 2024 года;
* завершение: 20 декабря 2024 года.

10.3 Ответственные лица

* менеджер проекта: Карпов Д.В;
* ведущий разработчик: Карпов Д.В.

Use-Case диаграммы

1 UML (с английского аббревиатура расшифровывается как UnifiedModelingLanguage — унифицированный язык моделирования) — это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем. Для разрабатываемой системы было решено спроектировать следующие диаграммы:

* диаграмма прецедентов;
* диаграмма классов.

Диаграмма прецедентов (диаграмма вариантов использования) -диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 1.

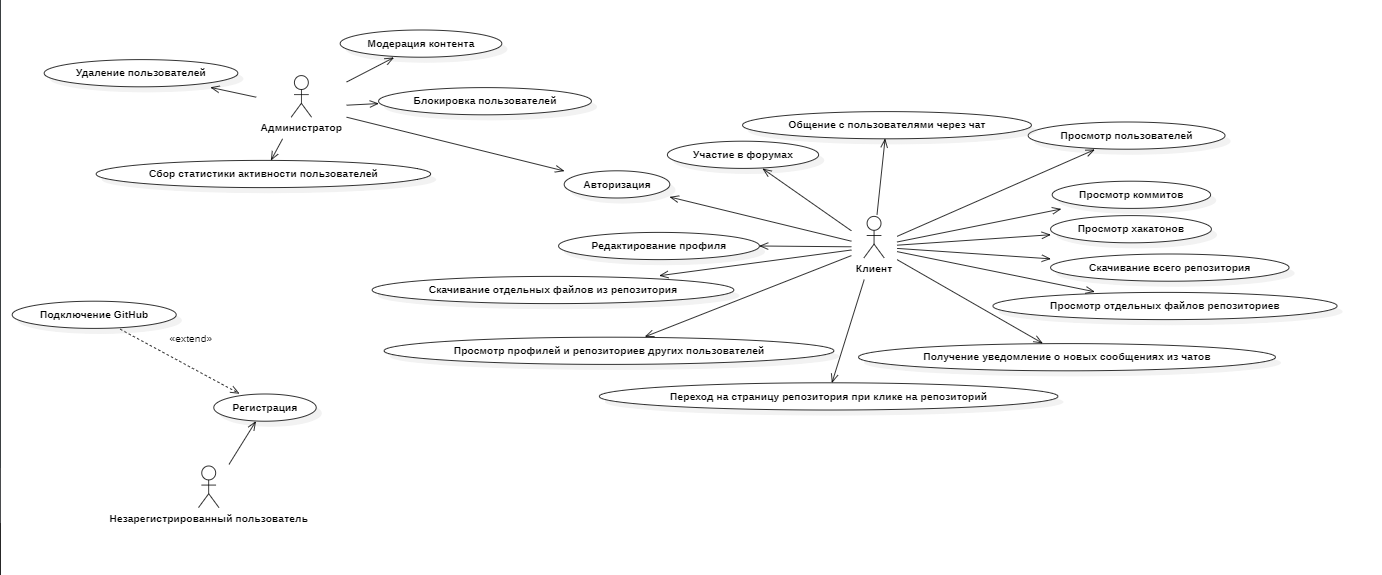


Рисунок 1 — Диаграмма прецедентов

2 Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 2.

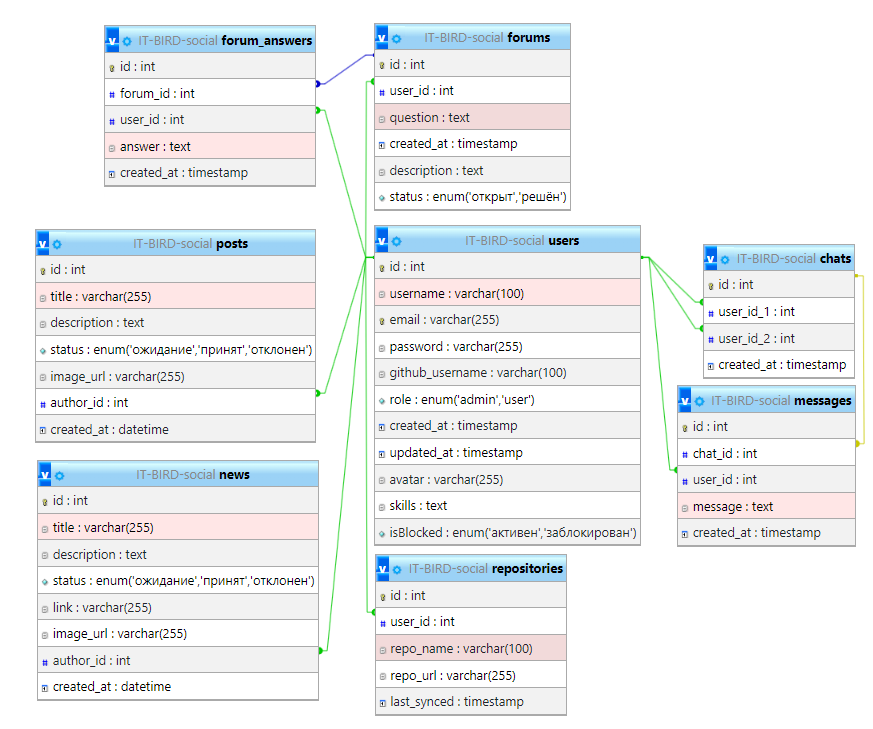


Рисунок 2 – Схема отношений базы данных

Структура таблицы users представлена в таблице 2.1.

Таблица 2.1. – Users (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | Id | INTEGER(4) | Первичный ключ, уникальное значение |
| Имя пользователя | username | VARCHAR(100) | Обязательное поле, уникальное значение |
| Почта | email | VARCHAR(255) | Обязательное поле |
| Пароль | Password | VARCHAR(255) | Обязательное поле |
| Github-имя | Github\_username | VARCHAR(100) | Необязательное поле |
| Роль | Role | ENUM(admin, user) | Обязательное поле |
| Дата создания | Created\_at | TIMESTAMP | Обязательное поле |
| Дата обновления | Updated\_at | TIMESTAMP | Обязательное поле |

Продолжение таблицы users 2.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Аватар | avatar | VARCHAR(255) | Обязательное поле |
| Навыки | Skills | TEXT | Обязательное поле |
| Статус блокировки | isBlocked | ENUM(активен, заблокирован) | Обязательное поле |

Структура таблицы repositories представлена в таблице 2.2.

Таблица 2.2 - Repositories (Репозитории)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код репозитория | id | INTEGER(4) | Первичный ключ,  Уникальное значение |
| Код пользователя | user\_id | INTEGER(4) | Внешний ключ (к users ) |
| Имя репозитория | repo\_name | VARCHAR(100) | Обязательное поле |
| Ссылка на репозиторий | repo\_url | VARCHAR(255) | Обязательное поле |
| Последняя синхронизация | last\_synced | TIMESTAMP | Обязательное поле |

Структура таблицы posts представлена в таблице 2.3.

Таблица постов 2.3- Posts (Посты)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код поста | id | INTEGER(4) | Первичный ключ, уникальное значение |
| Название поста | title | VARCHAR(255) | Обязательное поле |
| Описание поста | description | TEXT | Обязательное поле |
| Статус поста | status | ENUM(‘ожидание’, ‘принят’, ‘отклонен’); | Обязательное поле |
| Картинка постов | image\_url | VARCHAR(255) | Обязательное поле |
| Код автора | author\_id | INTEGER(4) | Внешний ключ (к Users) |
| Дата создания | created\_at | VARCHAR(100) | Обязательное поле |

Структура таблицы news представлена в таблице 2.4.

Таблица 2.4 – News (Новости)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код новости | Id\_record | INTEGER(4) | Первичный ключ, уникальное значение |
| Название новости | title | VARCHAR(255) | Обязательное поле |
| Описание новости | description | TEXT | Обязательное поле |
| Статус новости | status | ENUM(‘ожидание’, ‘принят’, ‘отклонен’); | Обязательное поле |
| Ссылка | link | VARCHAR(255) | Обязательное поле |
| Картинка новости | image\_url | VARCHAR(255) | Обязательное поле |
| Код автора | author\_id | INTEGER(4) | Внешний ключ (к Users) |

Структура таблицы messages представлена в таблице 2.5.

Таблица 2.5 - Messages (Сообщения)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код сообщения | id | INTEGER(4) | Первичный ключ, уникальное значение |
| Код чата | chat\_id | INTEGER(4) | Внешний ключ (к Chats) |
| Код пользователя | user\_id | INTEGER(4) | Внешний ключ (к Users) |
| Сообщение | images | TEXT | Обязательное поле |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |

Структура таблицы forum\_answers представлена в таблице 2.6.

Таблица 2.6 – Forum\_answers (Ответы на форуме)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код ответа | id | INTEGER(4) | Первичный ключ,  уникальное значение |

Продолжение таблицы forum\_answers 2.6.

|  |  |  |  |
| --- | --- | --- | --- |
| Код форума | forum\_id | INTEGER(4) | Внешний ключ (к Forums) |
| Код пользователя | user\_id | INTEGER(4) | Внешний ключ (к Users) |
| Ответ | answer | TEXT | Обязательное поле |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |

Структура таблицы forums представлена в таблице 2.7.

Таблица 2.7 – Forums (Вопросы на форуме)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код ответа | id | INTEGER(4) | Первичный ключ, уникальное значение |
| Вопрос | question | TEXT | Обязательное поле |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |
| Описание вопроса | description | TEXT | Обязательное поле |
| Статус | status | ENUM(‘открыт’, ‘решён’); | Обязательное поле |

Структура таблицы chats представлена в таблице 2.8.

Таблица 2.8 – Chats (Чаты)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код чата | id | INTEGER(4) | Первичный ключ,  уникальное значение |
| Код отправителя | user\_id\_1 | INTEGER(4) | Внешний ключ (к Users) |
| Код собеседника | user\_id\_2 | INTEGER(4) | Внешний ключ (к Users) |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |