ВВЕДЕНИЕ

В условиях стремительного развития информационных технологий возрастает потребность в профессиональных платформах, способствующих обмену знаниями, коллективному решению задач и налаживанию профессиональных связей. Особенно это актуально для специалистов в сфере информационных технологий, таких как программисты, дизайнеры, аналитики и другие. Одним из эффективных решений является создание специализированных форумов и социальных сетей, ориентированных на профессиональное сообщество.

Сервис IT-форум - это современная веб-платформа, предназначенная для объединения специалистов в сфере IT. Основной целью данного проекта является создание онлайн-пространства, где пользователи смогут делиться опытом, задавать и отвечать на вопросы, участвовать в обсуждениях, следить за хакатонами, а также выстраивать личные и профессиональные связи с другими участниками сообщества.

Сервис предоставляет удобные инструменты для взаимодействия:

* форум;
* система постов и новостей;
* возможность подключения аккаунта GitHub для демонстрации своих проектов;
* система заявок в друзьях;
* система личных сообщений между друзьями;
* автоматический парсинг актуальных IT-хакатонов.

Таким образом, проект направлен на формирование активного профессионального сообщества и развитие культуры взаимопомощи среди IT-специалистов.

Целью данной работы является разработка специализированного веб-приложения – IT-форума, предоставляющего функциональные возможности для общения, обмена знаниями и формирования профессиональных связей среди участников.

Задачи практики:

* изучение предметной области;
* разработка концептуальной модели веб-приложения;
* реализация основных пользовательских сценариев (регистрация, авторизация, создание и просмотр постов, участие в форумах и так далее);
* интеграция с GitHub для просмотра проектов пользователей;
* реализация системы чатов и заявок в друзья;
* тестирование разработанного функционала;
* оформление технической и пользовательской документации.

Итоговой выполненной работой по окончанию преддипломной практики должен являться веб-сервис, направленный на объединение IT-специалистов и обеспечение эффективного взаимодействия между ними.

1. Постановка задачи
2. Описание предметной области

IT-форум — это специализированная платформа, объединяющая профессионалов в сфере информационных технологий. Он предоставляет своим пользователям возможность обмениваться знаниями, задавать вопросы, обсуждать проекты и делиться опытом как начинающих специалистов, так и опытных профессионалов. Платформа включает тематические разделы для публикации постов и вопросов, а также личные чаты для общения с друзьями.

Информационная система создаётся для упрощения коммуникации и обмена информацией между участниками IT-сообщества. Пользователь заходит на сайт, где может ознакомиться с актуальными новостями, постами, вопросами и ответами, а также просматривать данные, интегрированные с сервисами вроде GitHub (например, коммиты, структуру репозиториев и содержимое проектов). В личном кабинете хранятся только основные данные, необходимые для работы с сервисом (например, данные регистрации и базовая информация профиля, а также указанные навыки, которыми обладает пользователь).

Пользователями системы являются:

* клиент;
* администратор.

Клиент должен иметь в системе следующие возможности:

* регистрация в системе;
* вход в систему (и по почте, и по логину, и по GitHub-логину);
* просмотр новостей;
* просмотр постов;
* просмотр актуальных IT-хакатонов;
* просмотр пользователей;
* просмотр профиля и репозиториев других пользователей;
* просмотр заявок в друзья;
* просмотр своего профиля и репозиториев;
* редактирование своих персональных данных;
* возможность создавать новости;
* возможность создавать посты;
* возможность создавать вопрос на форуме;
* возможность закрывать свои вопросы на форуме;
* просмотр вопросов других пользователей на форуме;
* просмотр ответов на свои и на вопросы других пользователей на свои вопросы;
* возможность принимать или отклонять заявки в друзья;
* возможность отправлять сообщения в чатах;
* возможность отправлять файлы в чатах;
* возможность удалять свои отправленные сообщения в чатах;
* возможность скачивать файлы из чатов;
* возможность выйти из аккаунта.

Администратор в системе может иметь следующие возможности:

* управление пользователями (заблокировать, удалить);
* модерирование новостей (опубликовать или нет);
* модерирование постов (опубликовать или нет);
* удаление новостей;
* удаление постов;
* просмотр статистики зарегистрированных пользователей за всё время;
* регистрация в системе;
* вход в систему (и по почте, и по логину, и по GitHub-логину);
* просмотр новостей;
* просмотр постов;
* просмотр актуальных IT-хакатонов;
* просмотр пользователей;
* просмотр профиля и репозиториев других пользователей;
* просмотр заявок в друзья;
* просмотр своего профиля и репозиториев;
* редактирование своих персональных данных;
* возможность создавать новости;
* возможность создавать посты;
* возможность создавать вопрос на форуме;
* возможность закрывать незакрытые вопросы на форуме;
* просмотр вопросов других пользователей на форуме;
* просмотр ответов на свои и на вопросы других пользователей на свои вопросы;
* возможность принимать или отклонять заявки в друзья;
* возможность отправлять сообщения в чатах;
* возможность отправлять файлы в чатах;
* возможность удалять свои отправленные сообщения в чатах;
* возможность скачивать файлы из чатов;
* возможность выйти из аккаунта.

Информация, хранимая в базе данных информационной системы танцевальной студии:

* сведения о пользователях;
* информация о новостях;
* информация о постах;
* информация о репозиториях пользователях;
* сведения о чатах;
* сведения о сообщениях в чатах;
* сведения о друзьях;
* информация о вопросах на форуме;
* информация о репозиториях пользователя.

Данные о клиенте хранятся в системе и характеризуются следующими полями:

* код клиента;
* имя;
* почта;
* пароль;
* github-имя;
* роль;
* аватар;
* навыки;
* статус блокировки.

После регистрации пользователь получает доступ ко всем возможностям платформы: он может просматривать и публиковать посты, задавать вопросы на форуме и отвечать на них, отправлять и принимать заявки в друзья, вести переписку в личных чатах, а также просматривать GitHub-профили пользователей с их репозиториями, коммитами и структурой проектов.

Для создания новости нужно заполнить форму, и данные о новости характеризуется следующим образом:

* код новости;
* название;
* описание новости;
* статус новости (на модерации, принята или отклонена);
* ссылка на новость в интернете;
* код автора;
* изображение;
* дата создания новости;

Для создания поста нужно заполнить форму, и данные о посте характеризуется следующим образом:

* код поста;
* название;
* описание поста;
* статус поста (на модерации, принята или отклонена);
* код автора;
* изображение;
* дата создания поста;

Авторизированный клиент может создать свой вопрос на форуме, где каждый вопрос характеризуется:

* код вопроса;
* код клиента;
* вопрос;
* дата создания;
* описание вопроса;
* статус.

Также авторизированный клиент может оставить ответ на вопрос пользователя на форуме, где каждый ответ характеризуется:

* код ответа;
* код вопроса;
* код клиента;
* ответ;
* дата создания.

Также авторизированный клиент может отправить заявку в друзья другому пользователю, где каждая заявка характеризуется:

* код клиента;
* код клиента, которому отправлена заявка (код друга);
* статус;
* дата отправки заявки.

Также авторизированный клиент может отправлять сообщения в чатах с друзьями, где каждый чат с другом заявка характеризуется:

* код чата;
* код клиента 1;
* код клиента 2 (собеседник);
* дата создания чата.

И где каждое сообщение характеризуется:

* код сообщения;
* код клиента;
* сообщение;
* медиа (если есть отправленные файлы);
* дата отправки сообщения.

Администратор может управлять пользователями, блокируя их в системе, или удаляя их из системы. Также администратор может модерировать контент на главной страницы, он может как принимать новости и посты, так и отклонять их. Также у администратора есть возможность удалять новости и посты. А также администратор может просматривать статистику зарегистрированных пользователей за всё время.

Ограничения в информационной системе:

* клиент, который не зарегистрировался в системе, не может создавать новости, посты, вопросы на форумах, общаться с другими пользователями, а также не может просматривать профили и репозитории других пользователей.

UML диаграммы

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.

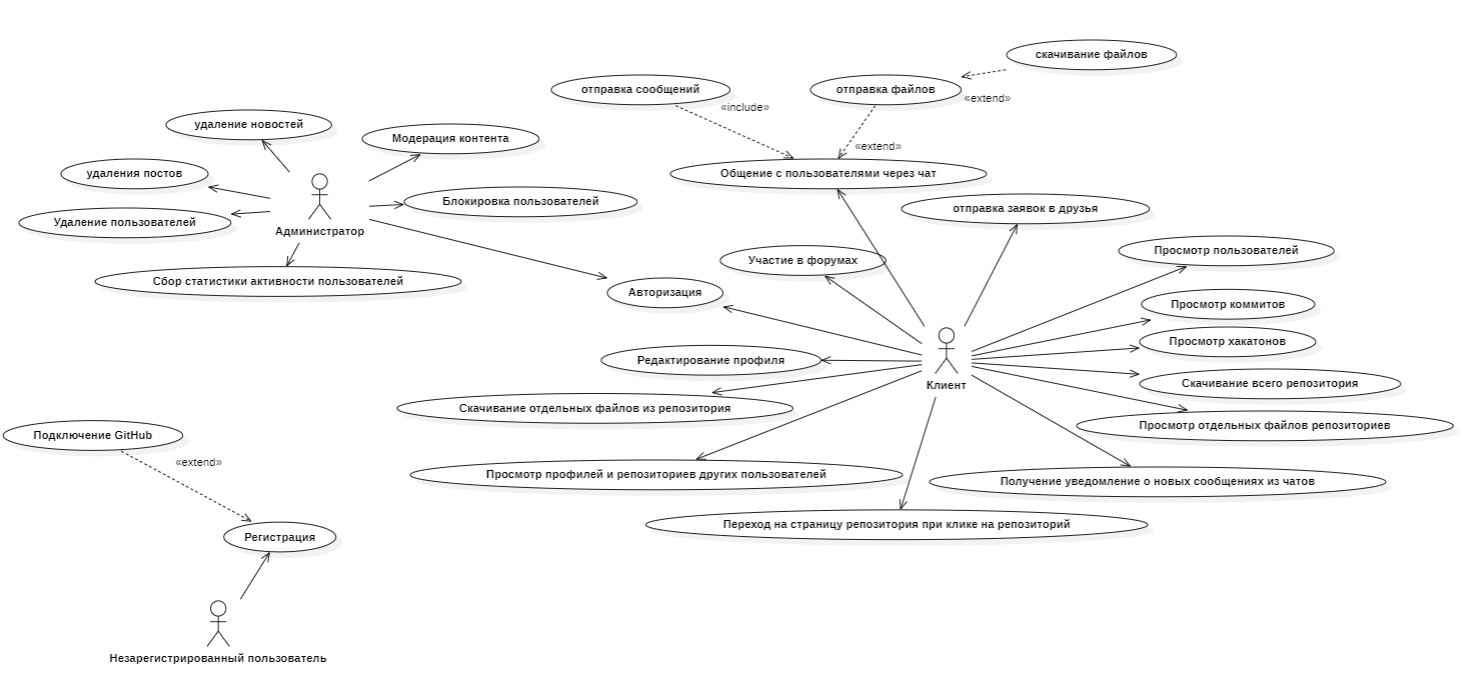


Рисунок 1 – Диаграмма прецедентов

Диаграмма классов (англ. classdiagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Диаграмма классов представлена на рисунке 2.

? Хотел спросить нужна ли мне диаграмма классов? Если да, то что именно нужно в ней указать?

Логическое моделирование

Логическое моделирование - это процесс создания структурированной модели, которая описывает данные, их отношения, ограничения и правила обработки данных на более детальном уровне, чем концептуальное моделирование.

Система предназначена для танцоров, преподавателей по танцам и администраторам платформы.

Логическое моделирование сайта представлено на рисунке 3.

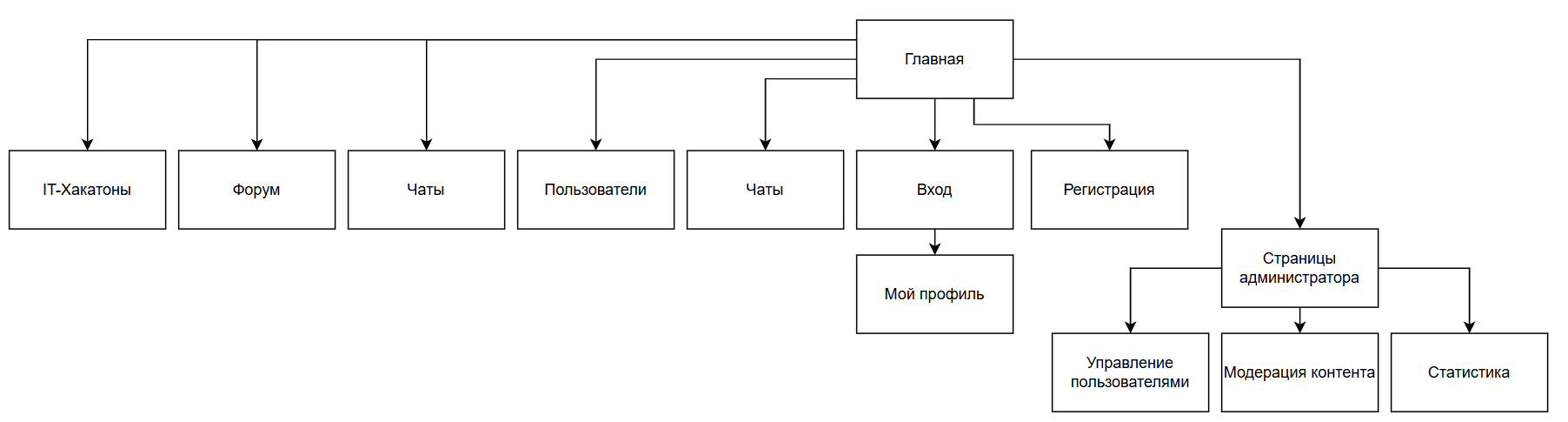


Рисунок 3 – Логическая структура веб-приложения

Описание структуры базы данных

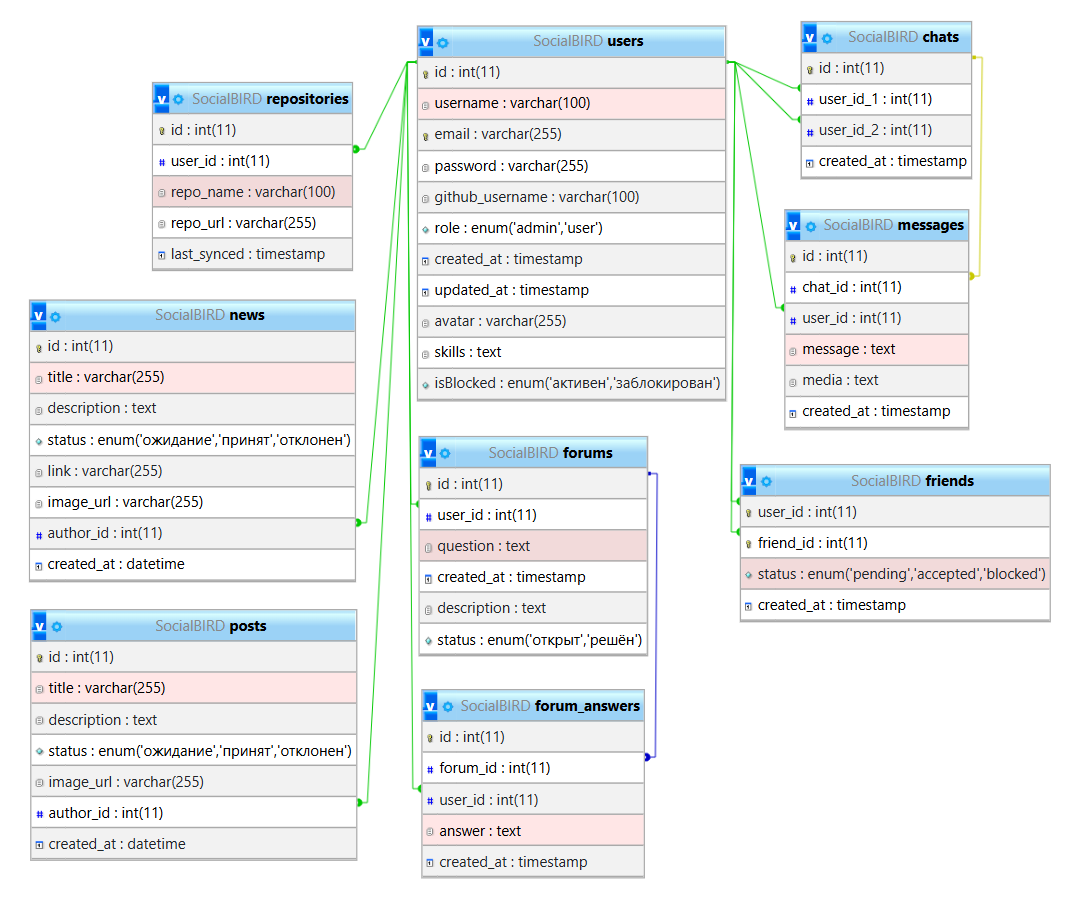
Схема отношений базы данных представлена на рисунке 4.  


Рисунок 4 – Схема отношений базы данных

Описание входной информации

Входная информация служит для осуществления деятельности системы, в которую входят данные пользователя при регистрации.

Информация о пользователе формируется на основе:

* имя пользователя;
* почта;
* пароль;
* github-имя.

Так же входной информацией для системы являются данные при создании новостей пользователем:

* название новости;
* описание новости;
* ссылка на новость;
* изображение.

Так же входной информацией для системы являются данные при создании поста пользователем:

* название поста;
* описание поста;
* изображение.

Так же входной информацией для системы являются данные при создании вопроса пользователем на форуме:

* название темы;
* описание проблемы.

Описание выходной информации

Выходная информация — информация, которая возникает в результате обработки человеком или устройством входной информации.

Основные требования к выходной информации сводятся к обеспечению пользователей сайта данными в удобной для них форме.

После обработки всей входной информации на выходе получается документ, содержащий в себе данные о занятости записи на занятие.

? Если у меня нет выходной информации, мне нужно её указывать?

Описание структуры базы данных

Описание структуры базы данных представлено в таблицах 11.1 – 11.8

Таблица 11.1 – Users (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | id | INTEGER(4) | Первичный ключ, уникальное значение |
| Имя | username | VARCHAR(100) | Обязательное поле |
| Почта | email | VARCHAR(255) | Обязательное поле, уникальное значение |
| 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 |
| Изображение | avatar | VARCHAR(255) | Необязательное поле |
| Навыки | skills | TEXT | Необязательное поле |
| Статус блокировки | isBlocked | ENUM(‘активен, ‘заблокирован’) | Обязательное поле |

Структура таблицы posts представлена в таблице 11.2.

Таблица 11.2 – 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 | status | ENUM(‘ожидание’, ‘принят’, ‘отклонен’) | Обязательное поле |
| Изображение | image\_url | VARCHAR(255) | Обязательное поле |
| Код автора | author\_id | INTEGER(4) | Внешний ключ (к users) |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |

Структура таблицы news представлена в таблице 11.3.

Таблица 11.3 – News (Новости)

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

Структура таблицы messages представлена в таблице 11.4.

Таблица 11.4 – Messages (Сообщения)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код | 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 | message | TEXT | Обязательное поле |
| Медиа | media | TEXT | Необязательное поле |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |

Структура таблицы friends представлена в таблице 11.5.

Таблица 11.5 – Friends (Друзья)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | user\_id | INTEGER(4) | Первичный ключ, уникальное значение |
| Код друга | friend\_id | INTEGER(4) | Обязательное поле |
| Статус | status | ENUM(‘pending’,’accepted’,’blocked’) | Обязательное поле |
| Дата отправки заявки | created\_at | TIMESTAMP | Обязательное поле |

Структура таблицы forum\_answers представлена в таблице 11.6.

Таблица 11.6 – Forum\_answers (Ответы на форуме)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код ответа | id | INTEGER(4) | Первичный ключ, уникальное значение |
| 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 11.7.

Таблица 11.7 – Forums (Форум)

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

Структура таблицы forum\_answers представлена в таблице 11.8.

Таблица 11.8 – Forum\_answers (Ответы на форуме)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код чата | id | INTEGER(4) | Первичный ключ, уникальное значение |
| Код пользователя 1 | user\_id\_1 | INTEGER(4) | Внешний ключ (к users) |
| Код пользователя 2 | user\_id\_2 | INTEGER(4) | Внешний ключ (к users) |
| Дата создания | created\_at | TIMESTAMP | Обязательное поле |

Схема отношений представлена на рисунке 11.1.

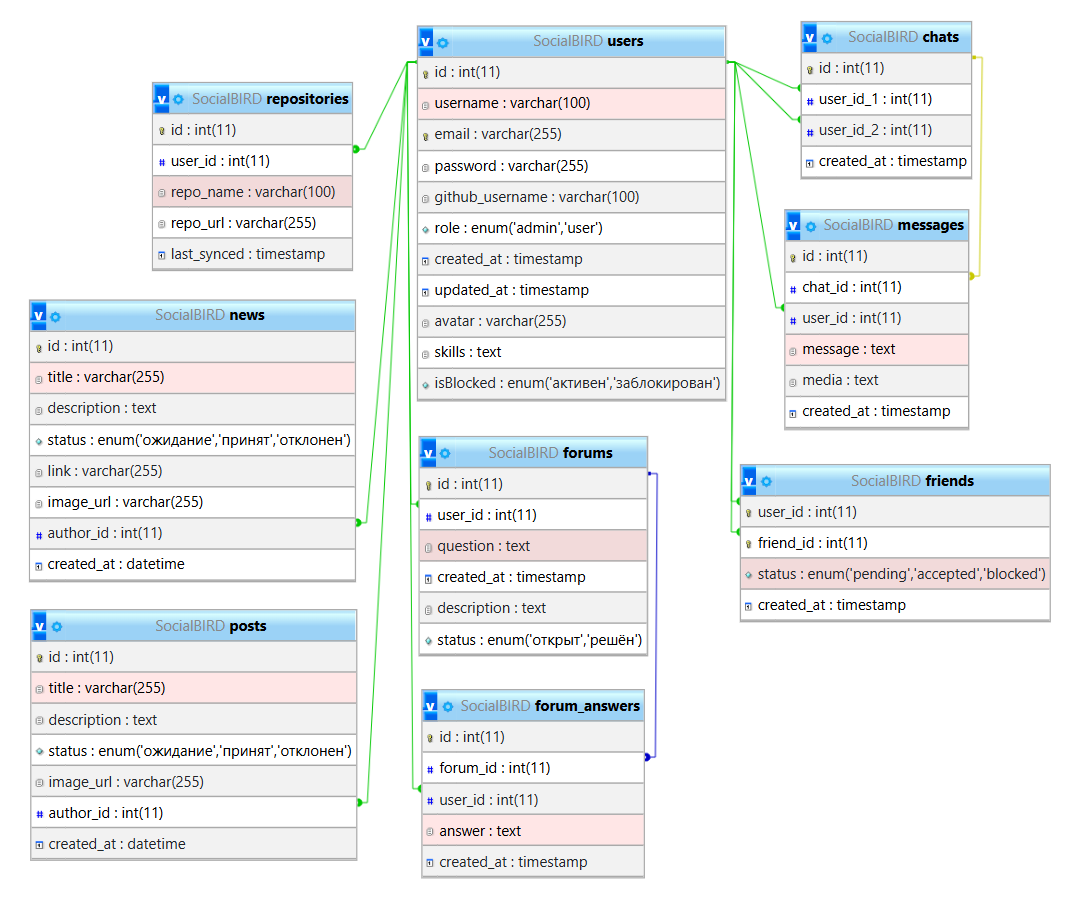


Рисунок 11.1 – Схема отношений

Контрольный пример

Контрольный пример – это таблицы с примером входных данных и предполагаемыми результатами, которые в дальнейшем используются для тестирования системы,

Эти результаты в дальнейшем должны совпадать с результатами, полученными при работе программного продукта на соответствующих входных данных. Данные должны быть подобраны таким образом, чтобы на них можно было продемонстрировать работу основных функций задачи, а при получении отчетов однозначно просматривались критерии сортировки, группировки, а также промежуточные и окончательные итоги