1. Обзор аналогов
2. Facebook

Facebook – одна из самых крупных социальных сетей в мире. На 2020 год входит в пятерку самых посещаемых сайтов мира. Основатель – Марк Цукерберг. Подробно остановимся на двух важных блоках: блок регистрации и профиль.

Страница регистрации представлена на рисунке 1.1.

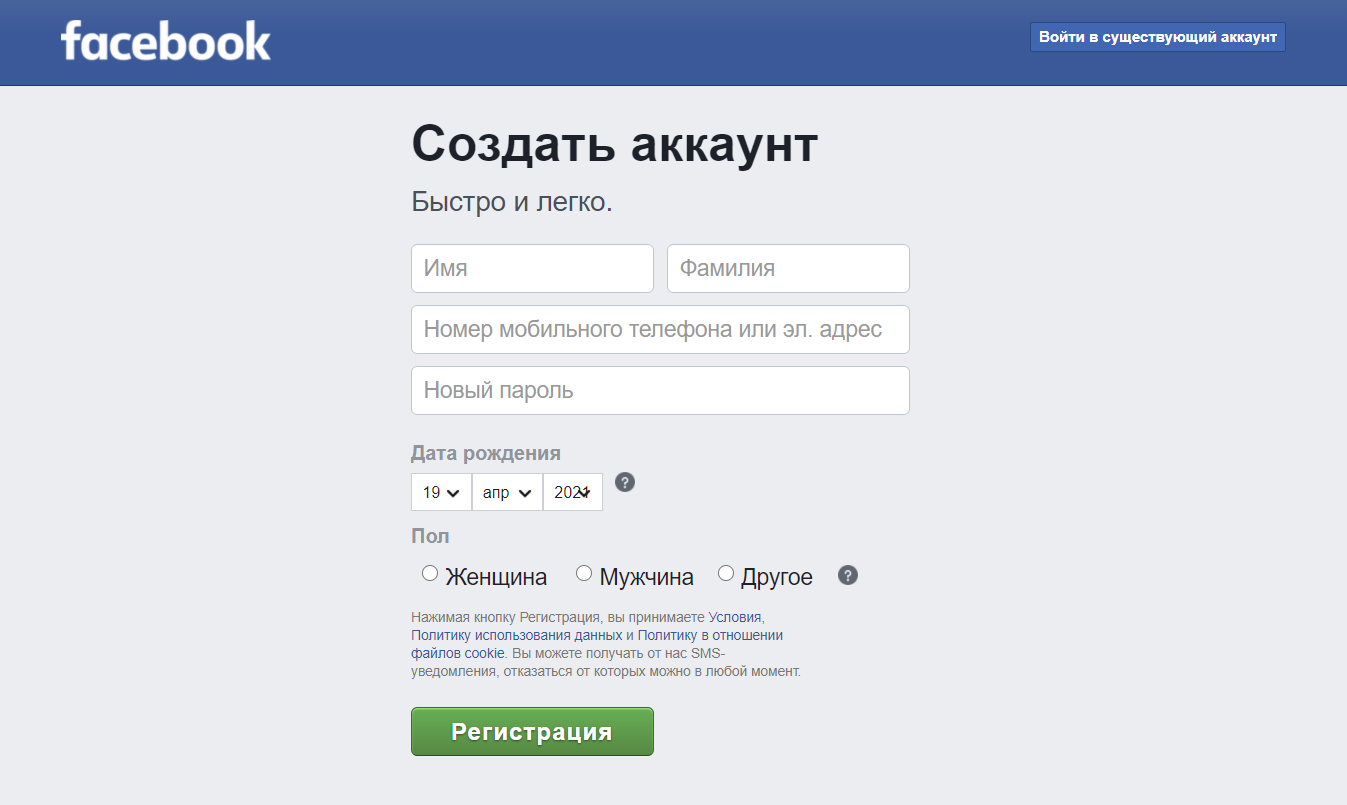


Рисунок 1.1 – Страница регистрации

Блок регистрации разработан удобно и позволяет ввести всю необходимую информацию о пользователе: имя, фамилия, мобильный телефон или почта, пароль, дата рождения, пол. Указаны ссылки на “Условия”, “Политику использования данных” и “Политику в отношении файлов cookie”. В процессе регистрации пользователю оказывается помощь в виде всплывающих подсказок и предупреждений.

Страница с личной информацией о пользователе загромождена различными инструментами, что можно проследить на рисунке 1.2. Эргономичность этой страницы находится на среднем уровне. Стоит отметить цветовую гамму. Создана она с использованием бело-серых тонов, в некоторых местах присутствует голубой цвет. Данное решение хорошо сочетается и не напрягает глаза пользователей.

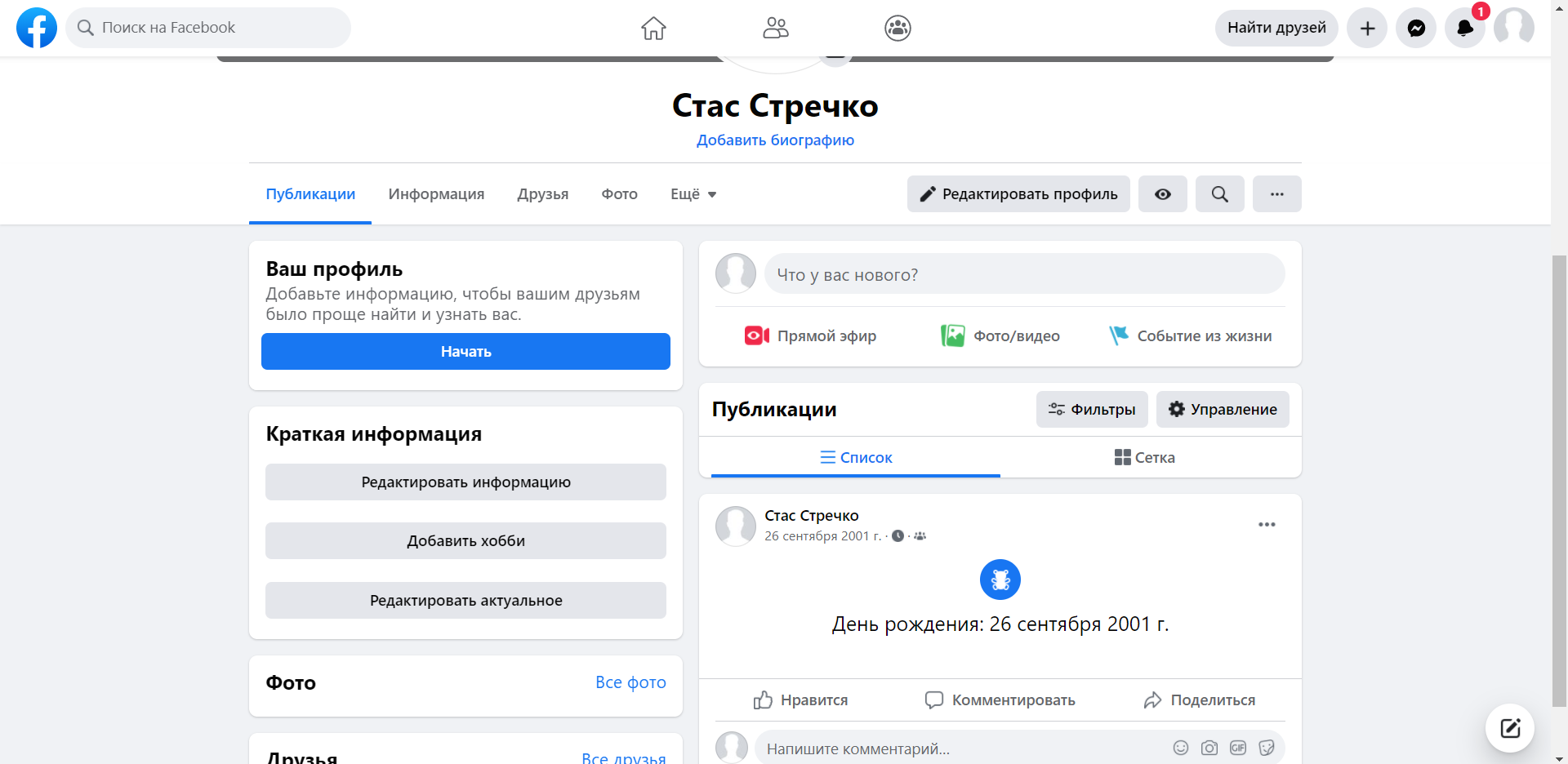


Рисунок 1.2 – Страница с личной информацией

Данная социальная сеть содержит огромное количество контента и обладает высокой функциональностью (изменение статуса, загрузка видеозаписей и фотографий, создание групп, обмен сообщениями с друзьями и прочее).

Важно сказать о том, что Facebook предоставляет пользователям возможность контролировать уровень доступа к информации, опубликованной в профиле, а также определять, кто имеет доступ к тем или иным данным.

На сайте также можно пожаловаться на неприятных пользователей или заблокировать их.

Плюсы:

* удобная регистрация;
* огромное количество контента;
* высокая функциональность;
* контроль уровня доступа к информации;
* возможность вести свой блог;
* приятная цветовая гамма.

Минусы:

* загруженность профиля;
* посредственность эргономичности страниц.

1. Instagram

Второй на очереди социальной сетью является Instagram.com. Instagram – приложение для обмена фотографиями с элементами социальной сети, позволяющее снимать фотографии и применять к ним различные фильтры. Авторы-создатели: Кевин Систром и Майк Кригер.

Блок регистрации можно увидеть на рисунке 1.3.

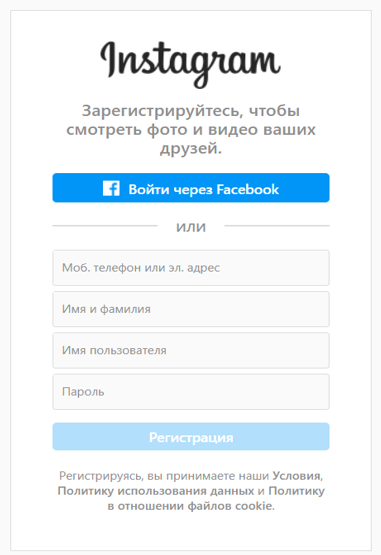


Рисунок 1.3 – Страница регистрации

В данной форме вводится необходимый минимум информации для корректной работы пользователя на сайте, а именно имя и фамилия, пароль, телефон/почта и имя пользователя. Также указаны ссылки на “Условия”, “Политику использования данных” и “Политику в отношении файлов cookie”.

Блок профиля пользователя приводится на рисунке 1.4.

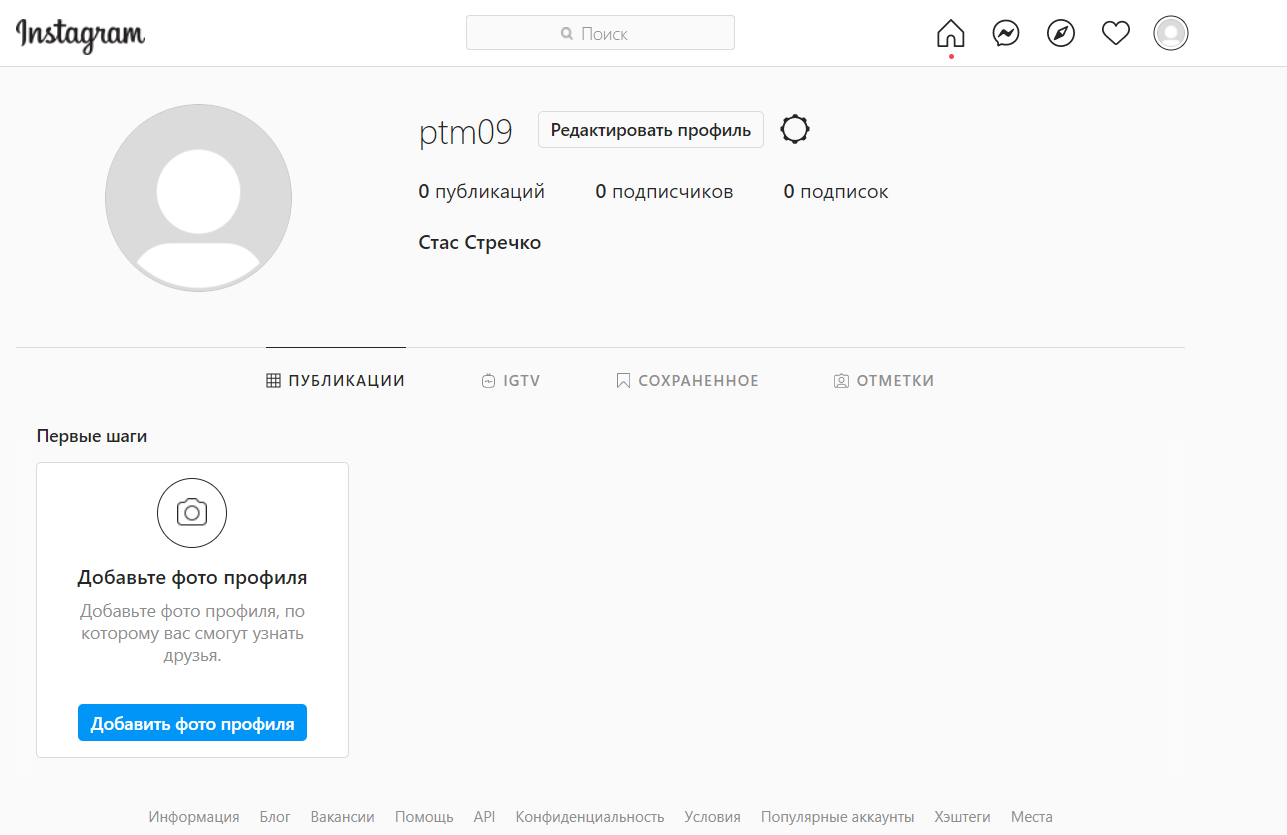


Рисунок 1.4 – Страница профиля

Страница имеет простой незагруженный дизайн. Выполнен он с использованием бело-серых тонов, некоторые элементы голубого цвета. Блок профиля содержит всю необходимую информацию о пользователе: количество публикаций, количество подписчиков и подписок, публикации, сохраненные данные, отметки. Стоит учесть, что данная страница предоставляет возможность редактировать и настраивать профиль.

Главными достоинствами Instagram являются простота и доступность. Не стоит также забывать о том, что в Instagram можно обмениваться сообщениями в Директе. Однако стоит отметить, что в Instagram нельзя создавать группы.

Плюсы:

* приятный и простой дизайн;
* доступность системы.

Минусы:

* отсутствие большого функционала;

## Схема базы данных

Реляционная модель

Реляционная модель данных — это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов). Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

В реляционной модели, в отличие от иерархической или сетевой, не существует физических отношений. Вся информация хранится в виде таблиц (отношений), состоящих из рядов и столбцов. А данные двух таблиц связаны общими столбцами, а не физическими ссылками или указателями. Для манипуляций с рядами данных существуют специальные операторы.

В отличие от двух других типов СУБД, в реляционных моделях данных нет необходимости просматривать все указатели, что облегчает выполнение запросов на выборку информации по сравнению с сетевыми и иерархическими СУБД. Это одна из основных причин, почему реляционная модель оказалась более удобна. В реляционной модели, как объекты, так и их отношения представлены только таблицами, и ничем более.

Распространённые реляционные СУБД: Oracle, Sybase, DB2, Ingres, Informix и MS-SQL Server.

Для создания базы данных был выбран Microsoft SQL Server 2019.

Модель базы данных — тип модели данных, которая определяет логическую структуру базы данных и принципиально определяет, каким образом данные могут быть сохранены, организованы и обработаны.

Текущая база данных состоит из одиннадцати таблиц, ее структура представлена на рисунке 2.1.

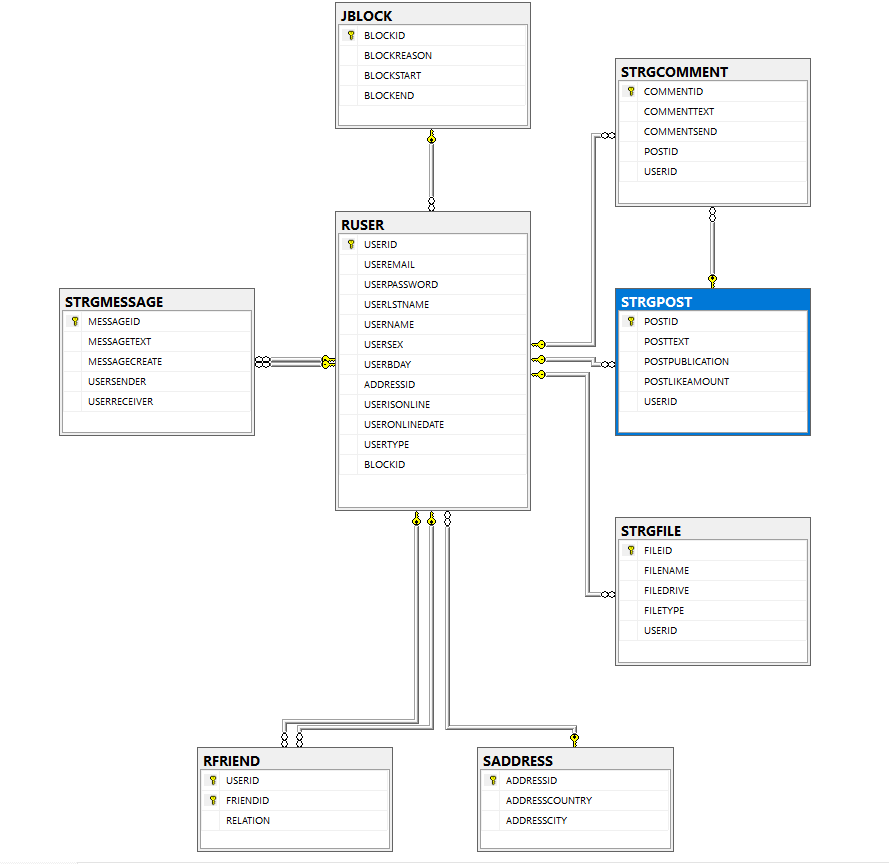


Рисунок 2.1 – Структура базы данных

Для лучшего понимая названий таблиц, введем условные обозначения на первые символы:

S – справочник;

R – реестр;

J – журнал;

STRG – хранилище.

На рисунке 2.2 представлена структура таблицы RUSER, которая содержит информацию о добавленных пользователях.

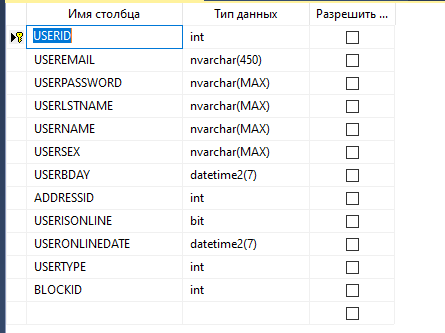


Рисунок 2.2 – Структура таблицы «RUSER»

Таблица содержит поля:

* *USERID*– идентификатор пользователя — ключевое поле;
* *USEREMAIL* – электронная почта пользователя;
* *USERPASSWORD* – пароль от учетной записи пользователя;
* *USERLSTNAME* – фамилия пользователя;
* *USERNAME* – имя пользователя;
* *USERSEX* – пол пользователя;
* *USERBDAY* – дата рождения пользователя;
* *USERISONLINE* – дата рождения пользователя;
* *USERONLINEDATE* – дата рождения пользователя;
* *USERTYPE* – дата рождения пользователя;
* *ADDRESSID* – адрес пользователя — внешний ключ;
* *BLOCKID*– идентификатор блокировки для пользователя — внешний ключ;

На рисунке 2.3 представлена структура таблицы SADDRESS, которая содержит информацию об адресе пользователя.

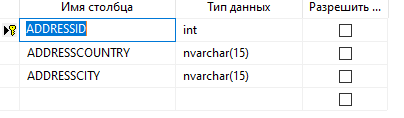


Рисунок 2.3 – Структура таблицы «SADDRESS»

Таблица содержит поля:

* *ADDRESSID* – идентификатор адреса — ключевое поле;
* *ADDRESSCOUNTRY* –  страна;
* *ADDRESSCITY* – город.

На рисунке 2.4 представлена структура таблицы JBLOCK, которая содержит информацию о заблокированных пользователях.

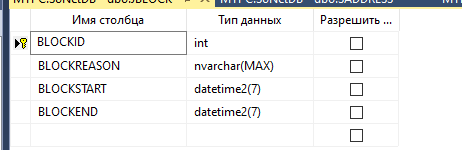


Рисунок 2.4 – Структура таблицы «JBLOCK»

Таблица содержит поля:

* *BLOCKID* – идентификатор записи о блокировке — ключевое поле;
* *BLOCKREASON* – причина блокировки;
* *BLOCKSTART –* дата и время начала блокировки;
* *BLOCKEND –* дата и время конца блокировки;

На рисунке 2.5 представлена структура таблицы STRGPOST, которая содержит информацию о публикациях.

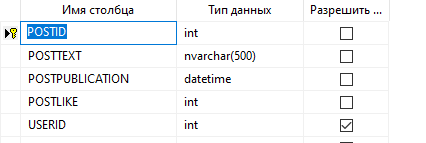


Рисунок 2.5 – Структура таблицы «STRGPOST»

Таблица содержит поля:

* *POSTID* – идентификатор публикации — ключевое поле;
* *POSTTEXT* – текст публикации;
* *POSTPUBLICATION –* дата и время создания публикации;
* *POSTLIKE* – количество отметок «Мне нравится» на публикации;
* *USERID* – идентификатор пользователя, который создал публикацию — внешний ключ.

На рисунке 2.6 представлена структура таблицы STRGCOMMENT, которая содержит информацию о комментариях под публикациями.

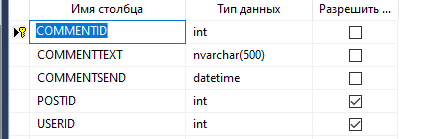


Рисунок 2.6 – Структура таблицы «STRGCOMMENT»

Таблица содержит поля:

* *COMMENTID* – идентификатор комментария — ключевое поле;
* *COMMENTTEXT* – текст комментария;
* *COMMENTSEND* – дата и время отправки комментария;
* *POSTID* – идентификатор поста, к которому относится комментарий — внешний ключ;
* *USERID* – идентификатор пользователя, который оставил комментарий — внешний ключ.

На рисунке 2.7 представлена структура таблицы STRGMESSAGE, которая содержит информацию об отправленных сообщениях.

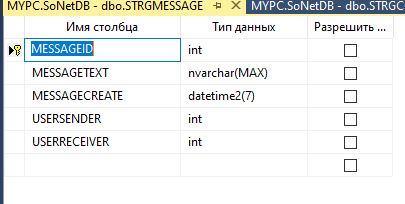


Рисунок 2.7 – Структура таблицы «STRGMESSAGE»

Таблица содержит поля:

* *MESSAGEID* – идентификатор записи о сообщении — ключевое поле;
* *MESSAGETEXT* – текст сообщения;
* *MESSAGECREATE*– дата и время создания сообщения;
* *USERSENDER* – идентификатор пользователя, который отправил сообщение — внешний ключ;
* *USERRECIEVER –* идентификатор пользователя, который получил сообщение — внешний ключ;

На рисунке 2.8 представлена структура таблицы RFRIEND, которая содержит информацию об отношениях между пользователями.

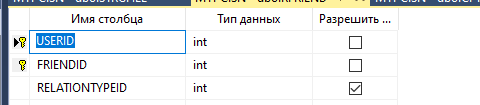


Рисунок 2.8 – Структура таблицы «RFRIEND»

Таблица содержит поля:

* *USERID* – пользователь, который инициировал отношения — ключевое поле — внешний ключ;
* *FRIENDID*– пользователь, в отношении которого начались отношения — ключевое поле — внешний ключ;
* *RELATION* – запись о типе отношений;

На рисунке 2.9 представлена структура таблицы STRGFILE, которая содержит информацию о файлах пользователя.

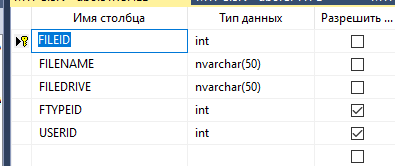


Рисунок 2.9 – Структура таблицы «STRGFILE»

Таблица содержит поля:

* *FILEID* – идентификатор записи о файле – ключевое поле;
* *FILENAME* – имя файла;
* *FILEDRIVE* – идентификатор файла на диске;
* *FTYPEID* – идентификатор типа файла;
* *USERID* – идентификатор пользователя, которому принадлежит файл — внешний ключ;

1. **Схемы взаимодействия**

Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

В данной программе существует роль пользователя и система администраторов. Диаграмма взаимодействия представлена на рисунке 3.1.

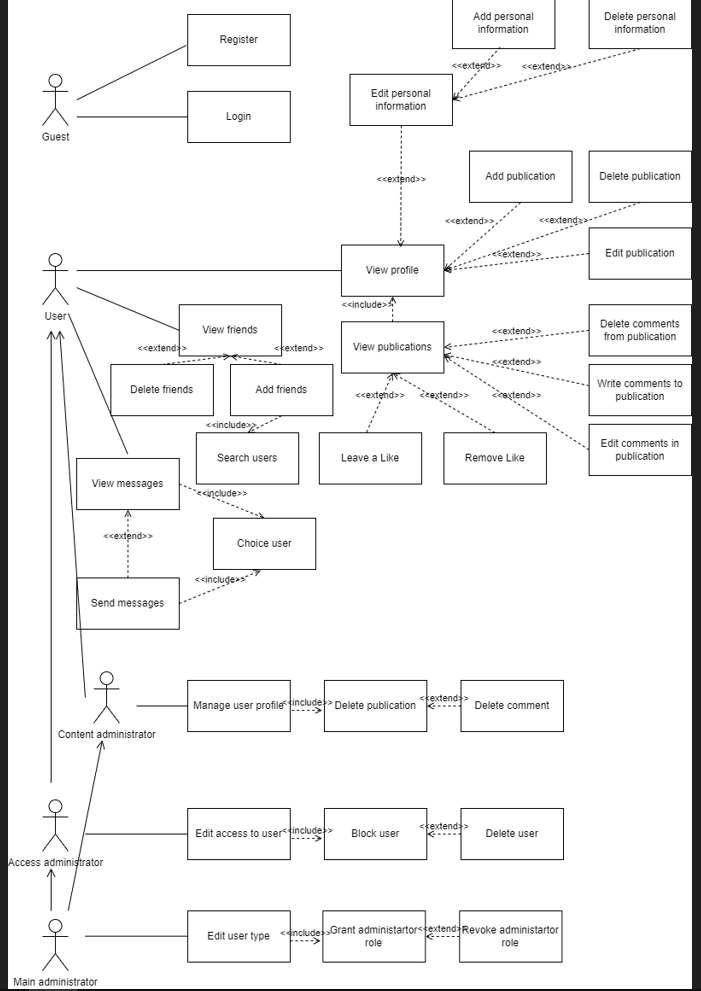


Рисунок 3.1 – Схема взаимодействия

На диаграмме присутствуют Guest, User, Content administrator, Access administrator и Main administrator. Видно, что администраторы также содержат в себе все функции обычного пользователя.

1. **Блок-схема**

Блок-схема — распространенный тип схем, описывающих [алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

Блок-схема алгоритма добавления публикации представлена на рисунке 4.1.



Рисунок 4.1 – Блок-схема добавления публикации