



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПISKA К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Справочник рецептов коктейлей»

Студент ИУ7-61Б
(Группа)

(Подпись, дата) Д.Н. Михеев
(И.О.Фамилия)

Руководитель курсового проекта

(Подпись, дата) А.С. Григорьев
(И.О.Фамилия)

Москва, 2022 г.

Аннотация

Курсовая работа «Справочник рецептов коктейлей». В ходе выполнения работы были разработаны база данных, в которой хранится информация о коктейлях, ингредиентах, инструментах, пользователях и их списках избранных коктейлей, и приложение, которое позволяет взаимодействовать с данными.

Для реализации приложения использовался язык программирования Python и среда разработки IDLE 3.9.2. В качестве СУБД использовалась объектно-реляционная СУБД PostgreSQL и программа для администрирования баз данных в PostgreSQL pgAdmin.

Расчетно-пояснительная записка содержит 31 страницу, 19 рисунков, 2 таблицы, 3 источника, 1 приложение.

Содержание

Аннотация.....	2
Введение	4
1 Аналитический раздел.....	5
1.1 Постановка задачи	5
1.2 Формализация данных	5
1.3 Типы пользователей.....	6
1.4 Описание существующих СУБД	6
1.4.1 Основные функции СУБД	6
1.4.2 Классификация БД по модели данных	7
1.5 Выводы из аналитического раздела.....	9
2 Конструкторский раздел.....	10
2.1 Проектирование базы данных.....	10
2.1.1 Таблицы.....	10
2.1.2 Триггеры.....	13
2.1.3 Ролевая модель	14
2.2 Требования к программе.....	15
3 Технологический раздел.....	17
3.1 Средства реализации.....	17
3.2 Создание объектов БД	17
3.2.1 Создание таблиц.....	17
3.2.2 Создание триггеров	19
3.2.3 Создание ролей	20
3.3 Интерфейс программы.....	20
4 Экспериментальный раздел.....	26
4.1 Сравнение времени выполнения от количества параметров категорий	26
Вывод	28
Заключение	29
Список использованных источников	30
Приложение А	31
Приложение 1. Презентация.....	31

Введение

В современном мире существует множество баров, в котором можно попить разнообразные коктейли из разных стран. Но если коктейль человеку понравился, у человека нет возможности его записать или поделиться им с другими. Было бы удобно сделать справочник рецептов коктейлей, чтобы следить за новинками, создавать новые коктейли и сохранять рецепты понравившихся.

Цель данной работы – реализовать приложение для базы данных, которое поможет человеку или начинающему бармену сохранять коктейли и делиться новыми коктейлями с другими пользователями.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- 1) формализовать задание, определить необходимый функционал;
- 2) провести анализ СУБД;
- 3) описать структуру базы данных, включая объекты, из которых она состоит;
- 4) спроектировать приложение для доступа к БД;
- 5) создать и заполнить БД;
- 6) реализовать интерфейс для доступа к БД;
- 7) разработать программное обеспечение, которое позволит пользователю получать и изменять информацию о существующих точках питания и их меню.
- 8) решить исследовательскую задачу.

1 Аналитический раздел

1.1 Постановка задачи

Необходимо разработать программу для отображения информации о коктейлях и способах их приготовления. В частности, об ингредиентах, о необходимых предметах, о порядке действий. Пользователь должен иметь возможность формировать персональный список понравившихся коктейлей для дальнейшего быстрого просмотра информации о них. Также пользователь должен иметь возможность создавать свои рецепты, которые он может отправить на модерацию для добавления в базу данных для всех пользователей.

Также нужно добавить возможность регистрации в приложении, возможность поиска коктейлей по названию или по его части и возможность поиска по категориям.

1.2 Формализация данных

База данных должна хранить информацию о:

- коктейле;
- ингредиентах;
- инструментах;
- пользователях и их избранном списке напитков.

Таблица 1.1 – категории и сведения о данных

Категория	Сведения
Коктейль	Название, категории, рейтинг, рецепт, идентификатор автора
Ингредиенты	Название, категории, описание
Инструменты	Название, категории, описание
Пользователь	Логин, пароль, список избранных коктейлей

1.3 Типы пользователей

Из задачи ясно, что для создания личного списка нужна авторизация пользователей. Это делит пользователей на авторизованных и неавторизованных.

Для управления базой данных принято решение ввести роль администратора.

Таблица 1.2 – типы пользователей и их функционал

Тип пользователя	Функционал
Неавторизованный	Просмотр информации общей базы коктейлей. Регистрация, авторизация.
Авторизованный	Просмотр информации общей базы коктейлей, просмотр информации списка любимых коктейлей, редактирование списка любимых коктейлей, создание заявки на добавления нового коктейля в базу данных.
Администратор	Изменение информации общей базы коктейлей. Добавление, изменение, удаление коктейлей или пользователей в общей базе данных.

1.4 Описание существующих СУБД

1.4.1 Основные функции СУБД

Система управления базами данных, сокр. СУБД — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Основными функциями СУБД являются:

- управление данными во внешней памяти;
- управление данными в оперативной памяти с использованием дискового кэша;

- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД.

1.4.2 Классификация БД по модели данных

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных[3].

Существует 3 основных типа моделей организации данных:

- дореляционная;
- реляционная;
- постреляционная.

В иерархической модели данных используется представление базы данных в виде древовидной структуры, состоящей из объектов различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка к потомку, при этом возможна ситуация, когда объект-предок имеет несколько потомков, тогда как у объекта-потомка обязателен только один предок.

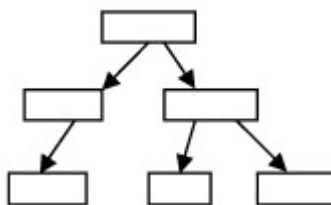


Рис. 1.1 – структура иерархической модели данных

В сетевой модели данных, в отличие от иерархической, у потомка может иметься любое число предков. Сетевая БД состоит из набора экземпляров

определенного типа записи и набора экземпляров определенного типа связей между этими записями.

Главным недостатком сетевой модели данных являются жесткость и высокая сложность схемы базы данных, построенной на основе этой модели. Так как логика процедуры выбора данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Иначе говоря, если будет необходимо изменить структуру данных, то нужно будет изменять и приложение.

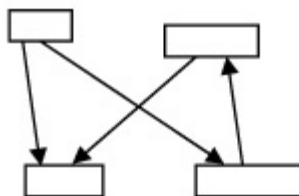


Рис. 1.2 – структура сетевой модели данных

Реляционная модель данных является совокупностью данных и состоит из набора двумерных таблиц. При табличной организации отсутствует иерархия элементов. Таблицы состоят из строк – записей и столбцов – полей. На пересечении строк и столбцов находятся конкретные значения. Для каждого поля определяется множество его значений. За счет возможности просмотра строк и столбцов в любом порядке достигается гибкость выбора подмножества элементов.

Реляционная модель является удобной и наиболее широко используемой формой представления данных.

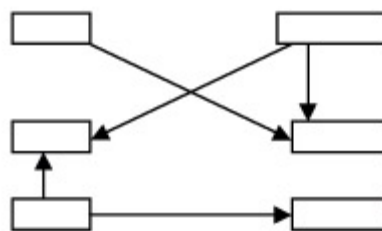


Рис. 1.3 – структура реляционной модели данных

Наиболее популярными реляционными СУБД являются Oracle, Microsoft SQL Server и PostgreSQL[1].

1.5 Выводы из аналитического раздела

В данном разделе была проанализирована поставленная задача и рассмотрены способы ее реализации. Также были рассмотрены разные типы СУБД. В качестве используемой модели в данной работе была выбрана реляционная модель.

В данной работе существует необходимость хранить рисунки с форматом jpeg. Сам файл будет храниться в файловой системе на сервере. Ссылка на него будет связана с номером коктейля, ингредиента или инструмента.

2 Конструкторский раздел

2.1 Проектирование базы данных

2.1.1 Таблицы

База данных должна хранить рассмотренные в таблице 1.1 данные. В соответствии с этой таблицей можно выделить следующие таблицы:

- таблица коктейлей Cocktails;
- таблица ингредиентов Ingredients;
- таблица инструментов для приготовления Instruments;
- развязочная таблица коктейлей и ингредиентов Ing_C;
- развязочная таблица коктейлей и инструментов Ins_C;
- таблица пользователей Users;
- таблица новых коктейлей News;
- таблица любимых коктейлей Favourites.

Таблица **Cocktails** должна хранить информацию о коктейлях:

- CocktailID – уникальный идентификатор коктейлей, PK, uniqueidentifier;
- Name – имя коктейля, varchar, not NULL;
- Tags – категории, varchar, not NULL;
- Rate – рейтинг, float, not NULL;
- Receipt – рецепт коктейля по шагам, varchar, not NULL;
- IDAuthor – автор коктейля, varchar;

Таблица **Ingredients** должна хранить информацию об ингредиентах:

- IngredientID – уникальный идентификатор ингредиента, varchar, PK, uniqueidentifier;
- Name – имя ингредиента, varchar, not NULL;

- Tags – категории ингредиента, varchar, not NULL;
- Description – описание ингредиента, varchar;

Таблица **Instruments** должна хранить информацию об инструментах:

- InstrumentID - уникальный идентификатор инструмента, varchar, PK, uniqueidentifier;
- Name – имя инструмента для приготовления, varchar, not NULL;
- Tags – категории инструмента, varchar, not NULL;
- Description – описание, varchar;

Таблица **Ins_C** должна хранить информацию о количестве необходимых инструментов при приготовлении коктейля:

- ID_Ins_C - уникальный идентификатор связи инструмента и коктейля, varchar, PK, uniqueidentifier;
- CocktailID – идентификатор коктейля, varchar, not NULL, FK;
- InstrumentID – идентификатор инструмента, varchar, not NULL, FK;
- Counter – количество инструментов, varchar, not NULL;

Таблица **Ing_C** должна хранить информацию о количестве нужных ингредиентов для приготовления коктейля:

- ID_Ing_C – уникальный идентификатор связи ингредиента и коктейля, varchar, PK, uniqueidentifier;
- CocktailID – идентификатор коктейля, varchar, not NULL, FK;
- IngredientID – идентификатор ингредиента, varchar, not NULL, FK;
- Capacity – количество граммов/миллилитров ингредиента, varchar, not NULL;

Таблица **Users** должна хранить информацию о всех пользователях:

- UserID – идентификатор пользователя, varchar, PK;
- PassN – права пользователя, short int, not NULL;

- Login – логин пользователя для входа в систему, varchar, not NULL;
- Password – пароль пользователя для входа в систему, varchar, not NULL;

Таблица **Favourites** должна хранить информацию о избранных коктейлях пользователя:

- FavoriteID – уникальный идентификатор таблицы, varchar, PK, uniqueidentifier;
- UserID - идентификатор пользователя, varchar, not NULL, FK;
- CocktailID – идентификатор коктейля, varchar, not NULL, FK;

Таблица **News** должна хранить информацию о коктейлях пользователей:

- New_ID – уникальный идентификатор коктейлей пользователя, PK, uniqueidentifier;
- UserID – идентификатор пользователя или автора, varchar, not NULL;
- Name – имя коктейля, varchar, not NULL;
- Tags – категории, varchar, not NULL;
- Receipt – рецепт коктейля по шагам, varchar, not NULL;
- Data_info_ins – идентификаторы инструментов и их количество, varchar, not NULL;
- Data_info_ing – идентификаторы ингредиентов и их объём, varchar, not NULL;
- Image – путь к картинке на компьютере, varchar, not NULL;
- Admit – флаг возможности добавить в таблицу (False, после прохождения цензуры становится True), bool, default False;

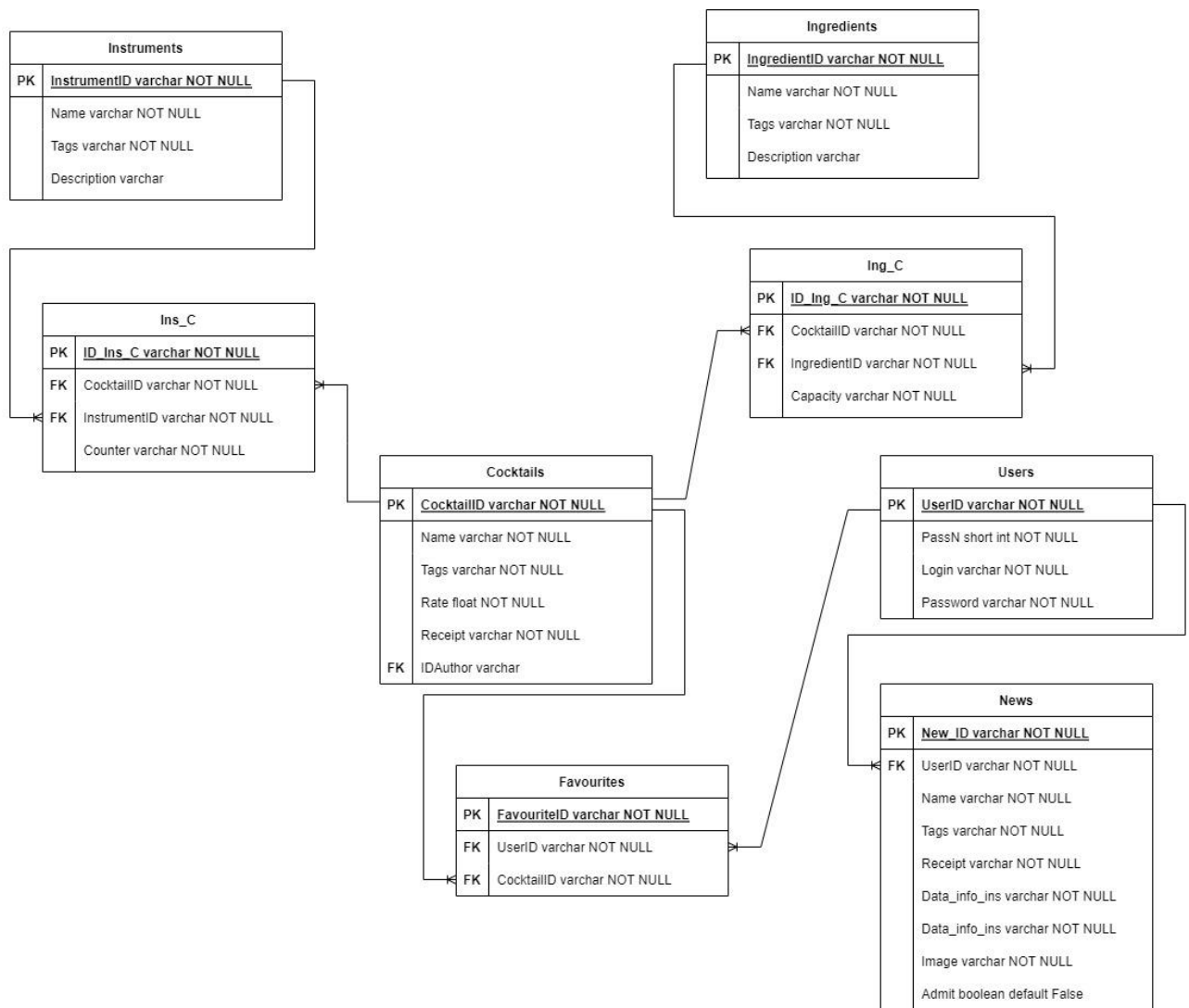


Рис. 2.1 – Диаграмма базы данных

2.1.2 Триггеры

Для поддержания целостности БД необходимо реализовать триггеры на удаление данных:

1. DML-триггер before delete, определенный на таблице Cocktails. До удаления коктейля из таблицы всех коктейлей, нужно удалить все записи с конкретным идентификатором из таблиц Ing_C, Ins_C и Favourites.
2. DML-триггер before delete, определенный на таблице Users. До удаления коктейля из таблицы всех коктейлей, нужно удалить все записи с конкретным идентификатором из таблиц Favourites.

На рисунке 2.2 представлены блок-схемы данных триггеров.

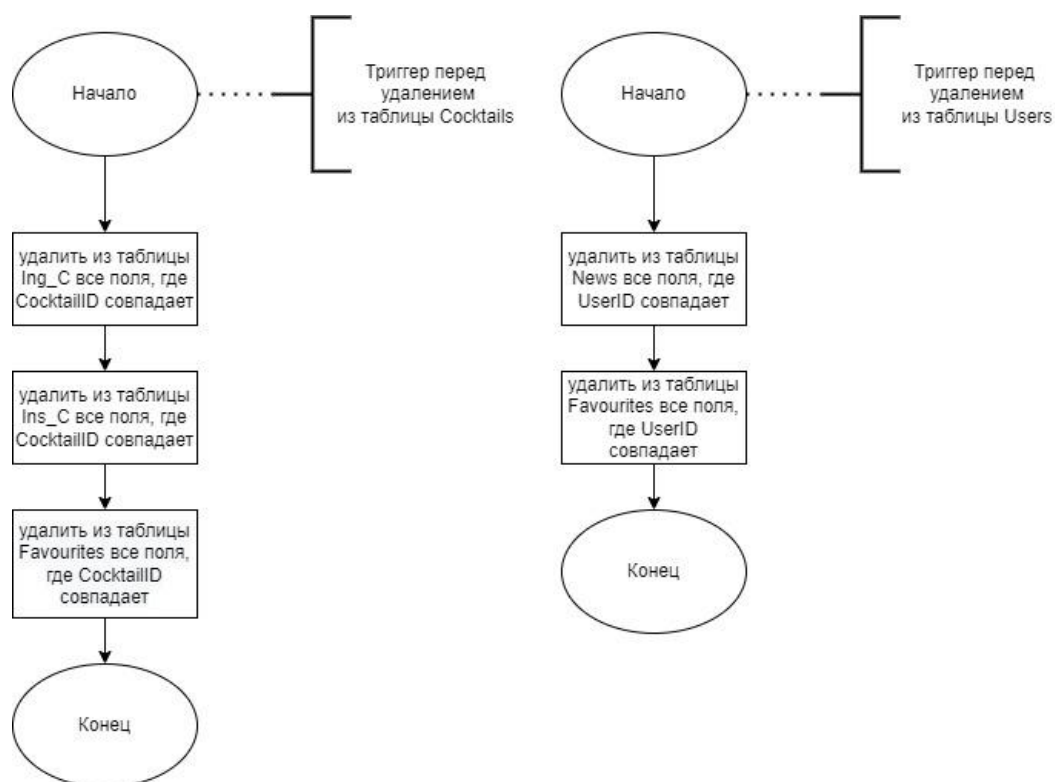


Рис. 2.2 – Схемы триггеров базы данных

2.1.3 Ролевая модель

Для безопасной работы с данными вводится 3 типа пользователей:

1. Неавторизованный пользователь (гость), которые могут просматривать таблицу коктейлей.
2. Авторизованный пользователь, которые могут просматривать таблицы коктейлей, редактировать список избранного и добавлять новый запрос коктейля.
3. Администратор, который может просматривать таблицы, редактировать список избранного и цензурировать запросы пользователей.

2.2 Требования к программе

Возможности незарегистрированного пользователя (гостя):

- регистрация;
- авторизация;
- просмотр списка всех коктейлей общей базы данных;
- просмотр выбранного коктейля и информации о нём;
- поиск максимально непохожих коктейлей по категориям;
- просмотр выбранного инструмента и информации о нём;
- просмотр выбранного ингредиента и информации о нём;
- поиск коктейля по выбранным категориям;
- поиск коктейля по названию (или его части).

Возможности зарегистрированного пользователя:

- авторизация;
- просмотр списка всех коктейлей общей базы данных;
- просмотр списка понравившихся коктейлей;
- добавление коктейля в список понравившихся;
- удаление коктейля из списка понравившихся;
- организация запроса на добавление коктейля в общую базу;
- просмотр выбранного коктейля и информации о нём;
- поиск максимально непохожих коктейлей по категориям;
- просмотр выбранного инструмента и информации о нём;
- просмотр выбранного ингредиента и информации о нём;
- поиск коктейля по выбранным категориям;
- поиск коктейля по названию (или его части).

Возможности администратора:

- авторизация;
- вывод списка всех коктейлей общей базы данных;
- просмотр списка понравившихся коктейлей пользователя;
- возможность подтверждения или отказа коктейля пользователя;
- добавление коктейля в список понравившихся;
- удаление коктейля из списка понравившихся;
- просмотр выбранного коктейля и информации о нём;
- поиск максимально непохожих коктейлей по категориям;
- просмотр выбранного инструмента и информации о нём;
- просмотр выбранного ингредиента и информации о нём;
- поиск коктейля по выбранным категориям;
- поиск коктейля по названию (или его части).

3 Технологический раздел

3.1 Средства реализации

В качестве языка программирования был выбран Python, так как:

- Имеются навыки создания компьютерного приложения с использованием tkinter.
- Для Python написано очень много библиотек, что позволит расширить возможности пользователей[2].

В качестве среды разработки была выбрана IDLE 3.9.2, так как:

- Редактор кода использует подсветку синтаксиса.
- При большом количестве функций, IDLE подсказывает какие параметры должны передаваться в ту или иную функцию.

3.2 Создание объектов БД

3.2.1 Создание таблиц

Ниже, на листинге 3.1-3.8, приведён код создания таблиц БД.

```
create table if not exists Cocktails (  
    CocktailID varchar primary key,  
    Name varchar not NULL,  
    Tags varchar not NULL,  
    Rate float not NULL,  
    Receipt varchar not NULL,  
    IDAuthor varchar default NULL);
```

Листинг 3.1: Создание таблицы Cocktails

```
create table if not exists Instruments (  
    InstrumentID varchar primary key,  
    Name varchar not NULL,  
    Tags varchar not NULL,  
    Description varchar);
```

Листинг 3.2: Создание таблицы Instruments

```
create table if not exists Users (
    UserID varchar primary key,
    PassN int not NULL,
    Login varchar not NULL,
    Password varchar not NULL);
```

Листинг 3.3: Создание таблицы Users

```
create table if not exists Ingredients (
    IngredientID varchar primary key,
    Name varchar not NULL,
    Tags varchar not NULL,
    Description varchar);
```

Листинг 3.4: Создание таблицы Ingredients

```
create table if not exists Favourites (
    FavoriteID varchar primary key,
    UserID varchar not NULL,
    CocktailID varchar not NULL,
    foreign key (CocktailID) references Cocktails(CocktailID),
    foreign key (UserID) references Users(UserID));
```

Листинг 3.5: Создание таблицы Favourites

```
create table if not exists News (
    New_ID varchar primary key,
    UserID varchar not NULL,
    Name varchar not NULL,
    Tags varchar not NULL,
    Receipt varchar not NULL,
    Data_info_ins varchar not NULL,
    Data_info_ing varchar not NULL,
    foreign key (UserID) references Users(UserID));
```

Листинг 3.6: Создание таблицы News

```
create table if not exists Ins_C (
    ID_Ins_C varchar primary key,
    CocktailID varchar not NULL,
    InstrumentID varchar not NULL,
    Counter varchar not NULL,
    foreign key (CocktailID) references Cocktails(CocktailID),
    foreign key (InstrumentID) references Instruments(InstrumentID));
```

Листинг 3.7: Создание таблицы Ins_C

```

create table if not exists Ing_C (
    ID_Ing_C varchar primary key,
    CocktailID varchar not NULL,
    IngredientID varchar not NULL,
    Capacity varchar not NULL,
    foreign key (CocktailID) references Cocktails (CocktailID),
    foreign key (IngredientID) references Ingredients (IngredientID));

```

Листинг 3.8: Создание таблицы Ing_C

3.2.2 Создание триггеров

Ниже, на листинге 3.9-3.10, приведён код создания триггеров БД.

DML-триггер, который срабатывает до удаления из таблицы Cocktails:

```

create or replace function fn_before_delete_user()
returns trigger
as
$body$
begin
delete from favourites where userid = old.userid;
return old;
end;
$body$
language plpgsql;

create trigger tr_delete_user before delete on users
for each row execute function fn_before_delete_user();

```

Листинг 3.9: Создание таблицы Ing_C

DML-триггер, который срабатывает до удаления из таблицы Users:

```

create or replace function fn_before_delete_user()
returns trigger
as
$body$
begin
delete from favourites where userid = old.userid;
return old;
end;
$body$
language plpgsql;

create trigger tr_delete_user before delete on users
for each row execute function fn_before_delete_user();

```

Листинг 3.10: Создание таблицы Ing_C

3.2.3 Создание ролей

Ниже, на листинге 3.11-3.12, приведён код создания пользователей и «guest827ccb0eea8a706c4c34a16891f84e7b» (гость).

Роль с правами гостя на просмотр таблицы Cocktails:

```
CREATE USER guest827ccb0eea8a706c4c34a16891f84e7b WITH PASSWORD '12345';  
GRANT select on Cocktails, Ingredients, Instruments, Ing_C, Ins_C to guest;
```

Листинг 3.11: Создание роли гостя

Создание новых пользователей с правами пользователя на просмотр таблицы Cocktails и редактирование таблицы Favourites и News (вместо %s соответственно подставляются логин, пароль, логин, логин, логин):

```
CREATE USER %s WITH PASSWORD %s;  
GRANT select on Cocktails, Ingredients, Instruments, Ing_C, Ins_C,  
Favourites, News to %s;  
GRANT insert on Favourites, News to %s;  
GRANT delete on Favourites to %s;
```

Листинг 3.12: Создание пользователя с именем и паролем

3.3 Интерфейс программы

На рисунке 3.1 – 3.12 представлены примеры работы программы.

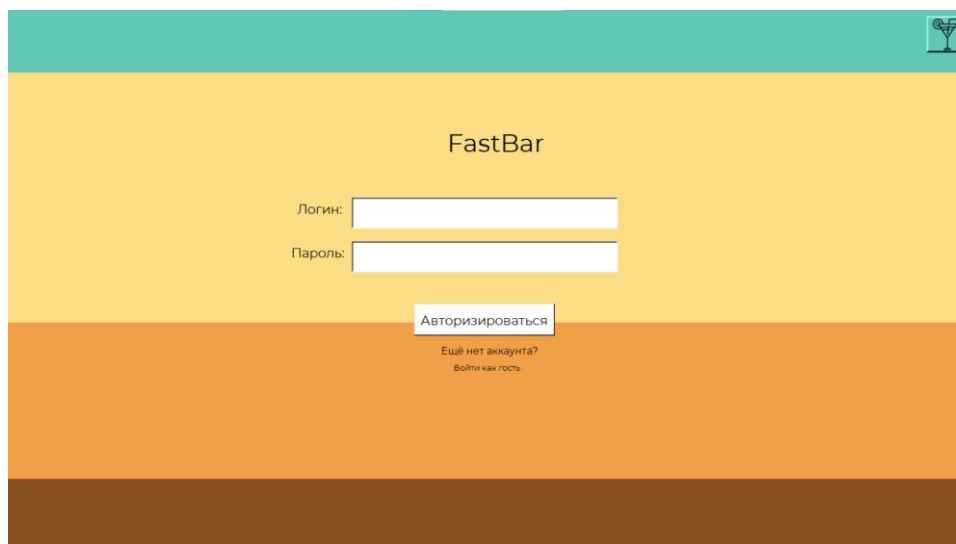


Рис. 3.1 – Начальное окно авторизации

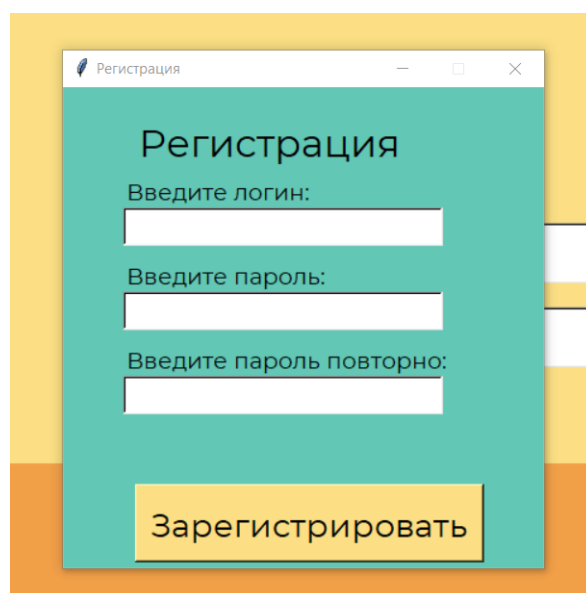


Рис. 3.2 – Окно регистрации пользователей

☐ безалкогольные ☐ слабоалкогольные

☒ крепкие













Image	Name	Tags	Rate	Author
	0.10	крепкое, травяные, горькие, на водке, шорт дринки	3.0	
	12 млпъ	крепкое, фруктовые, на роме, пончи, оранжевые	2.8	
	22.13	крепкое, сладкие, на дижне, шорт дринки, розовые	2.9	
	48 капель	крепкое, цитрусовые, сладкие, на водке, шорт дринки	3.7	
	50 оттенков улуна	крепкое, кислые, цитрусовые, на дижне, шорт дринки	3.3	
	560 Steit Стрит	крепкое, горькие, на виски, мартини, красные	4.5	
	866	крепкое, кислые, цитрусовые, травяные, горькие	0.0	

- ☐ на абсенте
- ☐ на ядре
- ☐ на праге
- ☐ на мискале
- ☐ на бурбоне
- ☐ на молоке
- ☐ на вермуте
- ☐ на гиве
- ☐ на вине
- ☐ на лиско
- ☐ на виски
- ☐ на роме
- ☐ на воде
- ☐ на саке
- ☐ на газировке
- ☐ на самбуке
- ☐ на граппе
- ☐ на сэдзу
- ☐ на дижне
- ☐ на содовой
- ☐ на игристом
- ☐ на соке
- ☐ на каламандосе
- ☐ на текиле
- ☐ на кашаса
- ☐ на хересе
- ☐ на коньяке
- ☐ на чае

- ☐ горькие
- ☐ соленые
- ☐ кислые
- ☐ сухие
- ☐ кофейные
- ☐ травяные
- ☐ матные
- ☐ пропитанные
- ☐ овощные
- ☐ фруктовые
- ☐ ликантные
- ☐ цветочные
- ☐ пряные
- ☐ цитрусовые
- ☐ сладкие
- ☐ шоколадные
- ☐ слабозольные
- ☐ ядочные

Рис. 3.3 – Таблица всех коктейлей

[illegible]

Рис. 3.4 – Таблица коктейлей из избранного

☐ безалкогольные
☐ крепкие

☐ слабоалкогольные
☐ крепкие

ap

☐ на абрикос
☐ на апельсине
☐ на вишне
☐ на чернике
☐ на персике
☐ на яблоне
☐ на киви
☐ на лимоне
☐ на сливе
☐ на клубнике
☐ на малине
☐ на смородине
☐ на еженике
☐ на вишне
☐ на чернике
☐ на персике
☐ на яблоне
☐ на киви
☐ на лимоне
☐ на сливе
☐ на клубнике
☐ на малине
☐ на смородине
☐ на еженике

☐ на лимоне
☐ на сливе
☐ на клубнике
☐ на малине
☐ на смородине
☐ на еженике
☐ на вишне
☐ на чернике
☐ на персике
☐ на яблоне
☐ на киви
☐ на лимоне
☐ на сливе
☐ на клубнике
☐ на малине
☐ на смородине
☐ на еженике
☐ на вишне
☐ на чернике
☐ на персике
☐ на яблоне
☐ на киви
☐ на лимоне
☐ на сливе
☐ на клубнике
☐ на малине
☐ на смородине
☐ на еженике








Image	Name	Taste	Rate	Author
	48 капель	крепкие, цитрусовые, сладкие, на воде, шорт дри๊งк	3.7	
	Апельсиновый гоголь-моголь	слабоалкогольные, сливочные, пряные, цитрусовые, сладкие	0.0	
	Апельсиновый коктейль	слабоалкогольные, цитрусовые, на лимоне, лонги, фizzes	0.0	
	Апельсиновый чейзер	крепкие, сладкие, пряные, цитрусовые, на тепле	3.1	
	Апрель в Париже	слабоалкогольные, цветочные, на вине, шорт дри๊งк, прозрачные	0.0	
	Барбекю по-будапештски	слабоалкогольные, фруктовые, кислые, на диане, лонги	3.9	
	БерриMyAe	крепкие, ягодные, пряные, на виски, лонги	3.0	

Рис. 3.5 – Поиск по названию коктейля

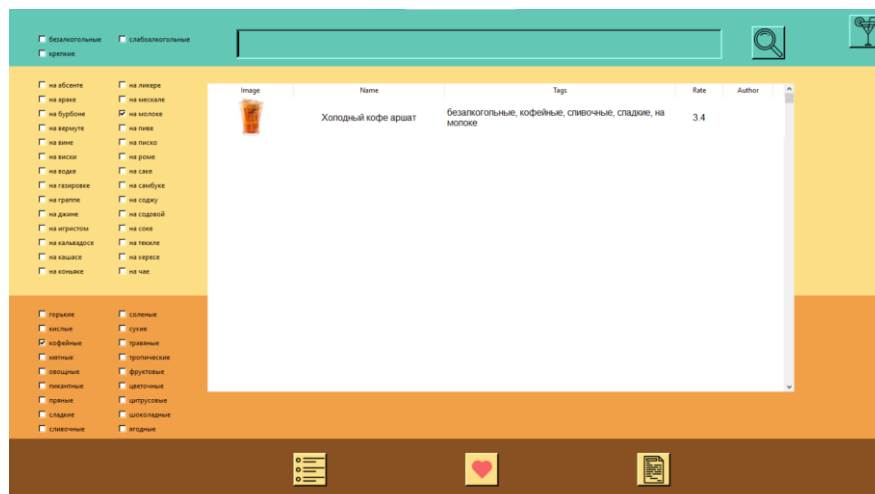


Рис. 3.6 – Поиск по категориям коктейля

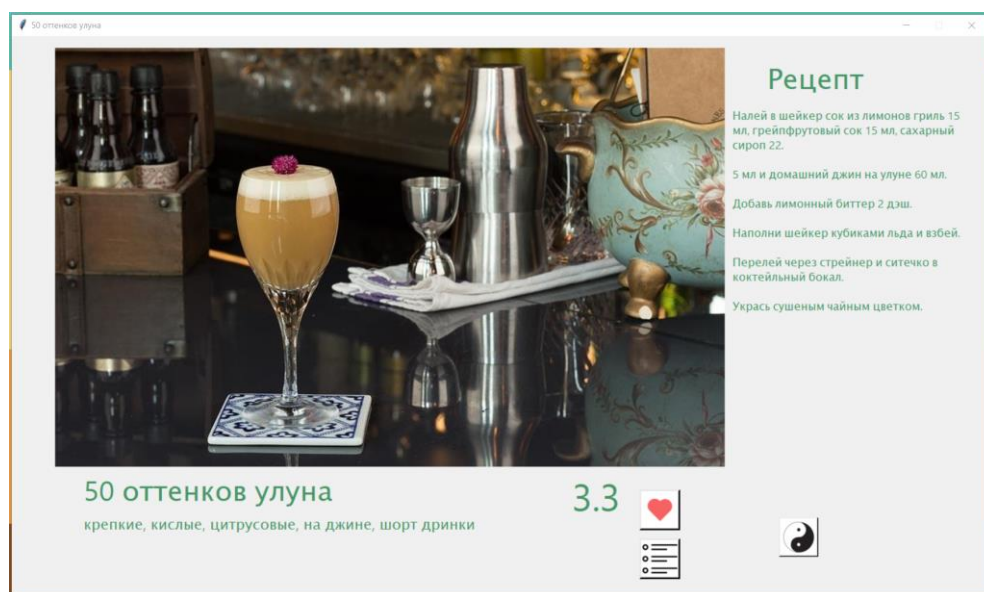


Рис. 3.7 – Окно с информацией о коктейле

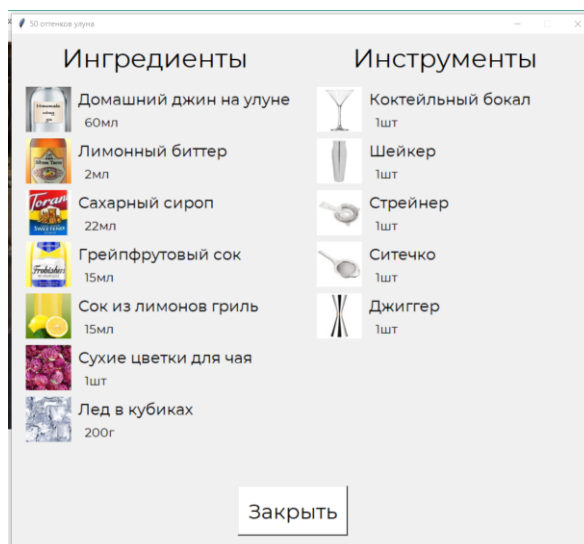


Рис. 3.8 – Окно с информацией об ингредиентах и необходимых инструментов для коктейля

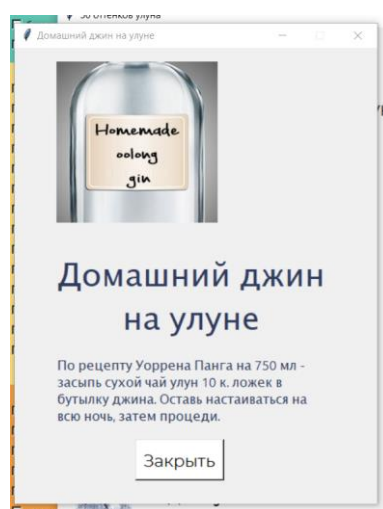


Рис. 3.9 – Окно с информацией об конкретном ингредиенте/инструменте

Новый запрос коктейля

Введите имя напитка:

Введите его тэги (до 5):

Введите рецепт напитка:

Введена картинка:

Введите нужные инструменты:

Введите нужные ингредиенты:

Выбрать картинку

Сохранить

Рис. 3.10 – Запрос пользователя о добавлении нового коктейля



Рис. 3.11 – Просмотр всех отложенных заявок на добавление

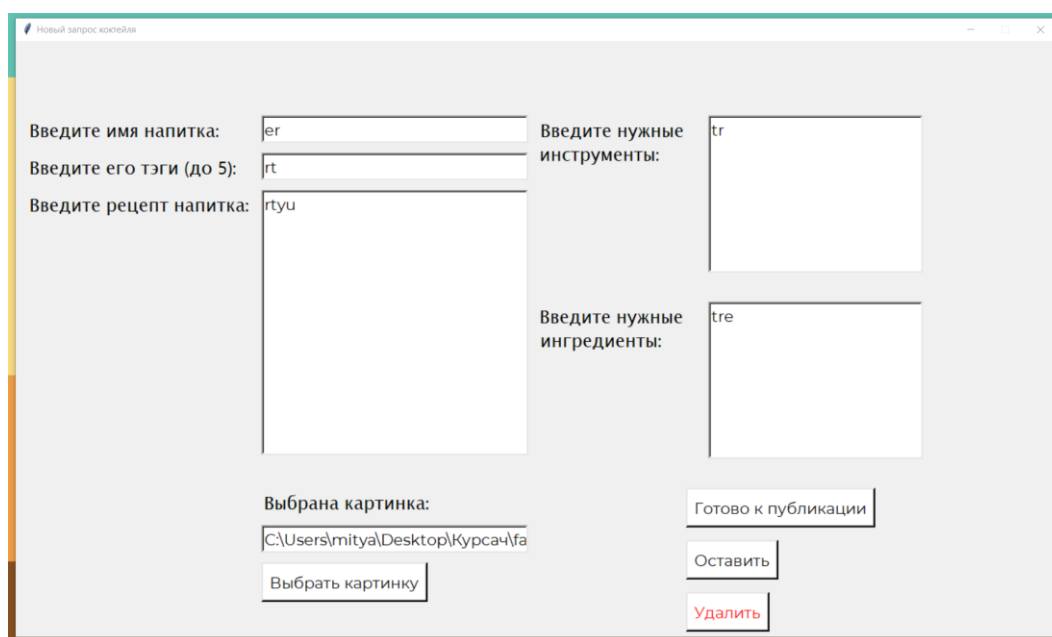


Рис. 3.12 – Просмотр конкретной заявки на добавление

4 Экспериментальный раздел

4.1 Сравнение времени выполнения от количества параметров категорий

В данном разделе проводится анализ времени выполнения запросов в зависимости от количества параметров категорий. Исследование проводилось на таблицах, состоящих из более 1000 записей. Проводится анализ времени выполнения запросов в зависимости от количества параметров категорий.

Примеры запросов при разном количестве параметров приведены на листинге 4.1 (вместо %s стоят значения категорий):

```
SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
ORDER BY name;

SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
WHERE tags LIKE %s
ORDER BY name;

SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
WHERE tags LIKE %s AND tags LIKE %s
ORDER BY name;

SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
WHERE tags LIKE %s AND tags LIKE %s AND tags LIKE %s
ORDER BY name;

SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
WHERE tags LIKE %s AND tags LIKE %s AND tags LIKE %s AND tags
LIKE %s
ORDER BY name;

SELECT cocktailid, name, tags, rate, idauthor
FROM cocktails
WHERE tags LIKE %s AND tags LIKE %s AND tags LIKE %s AND tags
LIKE %s AND tags LIKE %s
ORDER BY name;
```

Листинг 4.1: Запросы на разное количество категорий

Количество миллисекунд при поиске по категориям коктейля в таблице:

При пяти выбранных категориях: 3.011
При четырех выбранных категориях: 3.011
При трех выбранных категориях: 2.000
При двух выбранных категориях: 2.989
При одной выбранной категории: 7.009
При неограниченности в категориях: 9.010

Рис. 4.1 – Количество миллисекунд при разном количестве категорий

Было решено, сделать несколько таких экспериментов и посчитать среднее значение вручную, так как PostgreSQL для оптимизации работы оптимизирует данные. Так как все значения x являются целыми числами, то локальный минимум функции находится в точке $x = 4$. Т.е. при количестве категорий равной четырем запросы выполняются быстрее всего.

Это объяснено двумя аргументами:

- при 4 выбранных категориях остается до 5 коктейлей из всей таблицы коктейлей. Так как очень маловероятно повторение всех 4 категорий у двух разных коктейлей, скорее всего, пятая категория будет гарантировать совпадение множеств категорий, что предоставляет нам такой же результат, но с увеличенным временем запроса.
- оптимизация этих запросов самим PostgreSQL делается через создание индексов, не все данные кэшируются и ему приходится выполнять работы не в четырех плоскостях, а в пяти.

То есть, в какой-то момент стоимость проверки запроса повышается по сравнению с предыдущими результатами, что превышает стоимость добавления в базу новых строк.

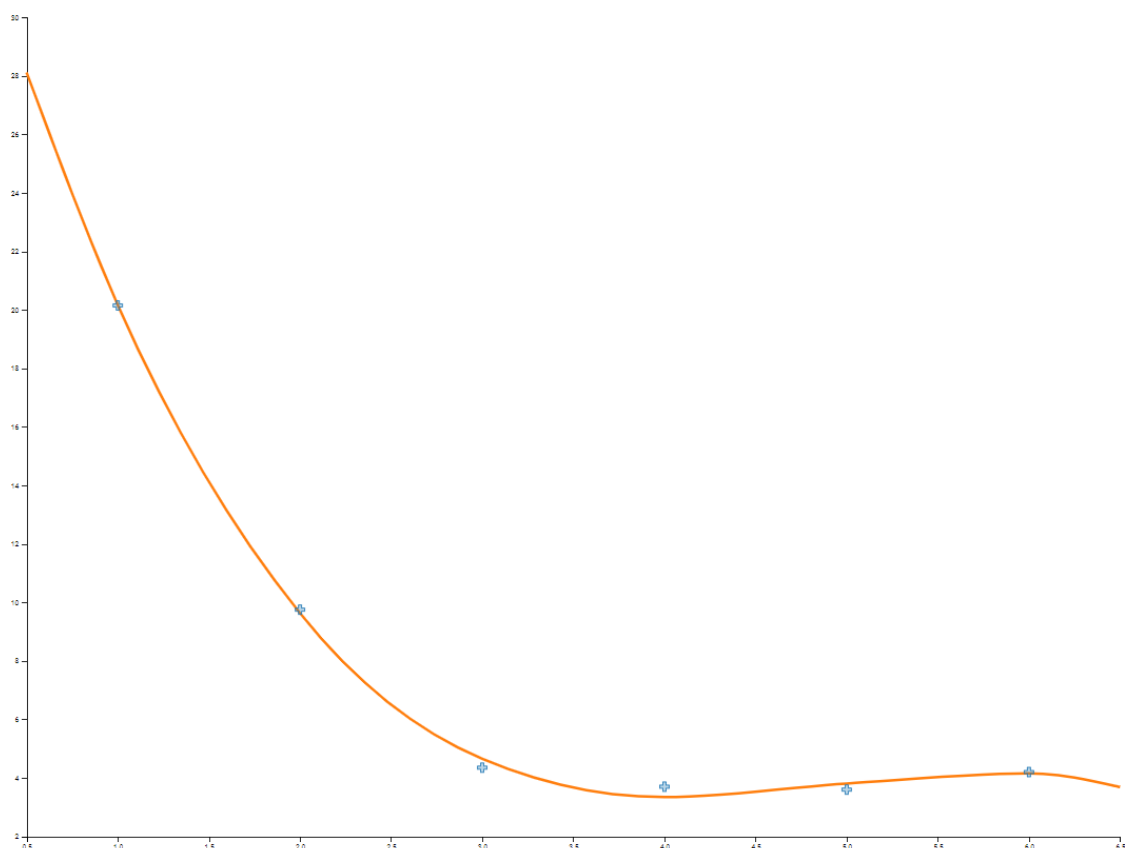


Рис. 4.3 – График аппроксимации по средним значениям миллисекунд

Вывод

В результате исследования было выяснено, что существует локальный минимум количества категорий (равный 4), который обосновывается тем, что, с одной стороны, не добавляет сравнительно большое количество записей в временную таблицу, но, с другой стороны, обеспечивает оптимальность запроса относительно стоимости проверки. Так как данные для этой БД были взяты с сайта коктейлей, а не написаны мной, число категорий у любого коктейля, ингредиента, инструмента было равно 5. При последующих работах с БД необходимо проводить анализ оптимального количества категорий при создании своей БД.

Заключение

В результате работы было разработано приложение справочника коктейлей, а также решены следующие задачи:

- сформулированы требования к разрабатываемому приложению;
- проанализированы варианты представления данных и выбран подходящий вариант для решения задачи;
- спроектирована база данных, описаны ее сущности и связи
- реализовано программное обеспечение на основе разработанной структуры приложения;
- проведено исследование производительности в зависимости от количества параметров категорий;

Список использованных источников

1. PostgreSQL [Электронный ресурс] URL: <https://www.postgresql.org/> (дата обращения: 30.05.2022)
2. Кузин А.В. Базы данных. – 5-е изд., испр. – М. : Издательский центр «Академия», 2012. – 320 с.
3. Мартин, Грубер. "Понимание SQL./Пер." *Лебедева ВН–М* (1993).

Приложение А

Приложение 1. Презентация