Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Казанский (Приволжский) федеральный университет»

Институт вычислительной математики и информационных технологий

**Техническое задание**

**«Приложение рекомендации: Рестораны, Бары, Кофейни»**

Выполнили:

Кузьмина А.И.,

Квятковский В.Г.,

Валетдинов Д.Р,

Берсенев С.А.

Казань, 2025 год

**ОГЛАВЛЕНИЕ**

[**1. Введение 3**](#_sz7kgk9l8tir)

[1.1 Цель 3](#_9q87agr2j35p)

[1.2 Краткая сводка возможностей 3](#_xlzfmeec3894)

[1.3 Определения, акронимы и сокращения 3](#_ma1ych2nagym)

[1.4 Краткое содержание 3](#_7p9i2kejbp8x)

[**2. Обзор системы 4**](#_y6wz7zvl4n36)

[2.1 Определение типов пользователей системы 4](#_tc2we3w4su67)

[2.3 Обзор вариантов использования 5](#_dobfjoq2mplo)

[2.4 Нефункциональные требования 7](#_i6trfta065ht)

[**3. Детальные требования 7**](#_j8mshxg8nztj)

[3.1 Функциональные требования 7](#_gesg0afks9ca)

[3.2 Требования к пользовательскому интерфейсу 18](#_djnzyfiilvda)

[3.3 Требования к логической структуре БД 25](#_nk3lf82l6c4i)

# **Введение**

## 1.1 Цель

Цель документа - предоставить информацию о полном функционале создаваемого приложения для подбора рекомендаций баров, ресторанов, кофеен. В документе содержится информация о возможностях, сценариях использования и структуре системы.

## 1.2 Краткая сводка возможностей

Приложение разработано для подбора баров, ресторанов и кофеен на основе анкеты, заполненной пользователем. В приложении представлены следующие функциональные возможности:

* Регистрация/Вход в аккаунт
* Анкетирование на основе интересов пользователя (тип/категория/кухня/средний чек)
* Добавление заведений в избранное, скрытие не понравившихся заведений
* Подбор рекомендаций на основе понравившихся заведений
* Просмотр подробной информации о заведении
* Генерация отчета об аккаунте пользователя

## 1.3 Определения, акронимы и сокращения

БД - База данных

ЛК - Личный кабинет (Аккаунт пользователя)

СУБД - Система управления базами данных

## 1.4 Краткое содержание

Во второй главе описываются типы пользователей, окружение системы, обзор вариантов использования и нефункциональные требования.

В третьей главе описываются функциональные требования, требования к пользовательскому интерфейсу и логической структуре БД.

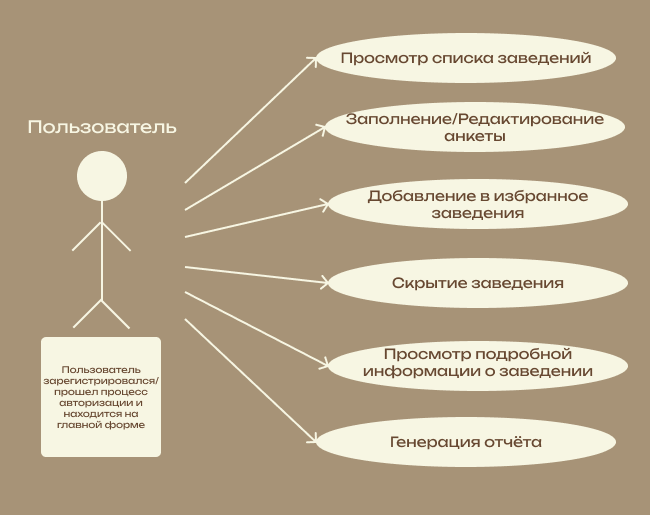
# **Обзор системы**

## 2.1 Определение типов пользователей системы

Пользователь: Регистрация, Вход в аккаунт, просмотр списка заведений, заполнение/изменение анкеты, добавление в избранное, скрытие заведений, просмотр подробной информации о заведении, генерация отчета.

Администратор: Вход в аккаунт (определенные логин и пароль), просмотр списка заведений, заполнение/изменение анкеты, добавление в избранное, скрытие заведений, просмотр подробной информации о заведении, генерация отчета, изменение файла со списком заведений.

2.2 Окружение системы



## 2.3 Обзор вариантов использования

1. Как пользователь, я хочу создать аккаунт, чтобы пользоваться приложением и работать с заведениями.
2. Как пользователь, я хочу войти в свой аккаунт, чтобы продолжить использование приложения с моими данными.
3. Как пользователь, я хочу просмотреть список всех заведений, чтобы подобрать интересное место без фильтрации.
4. Как пользователь, я хочу заполнить анкету, чтобы получить список заведений, соответствующих моим интересам (например, только рестораны с итальянской кухней).
5. Как пользователь, я хочу изменить ранее заполненную анкету, чтобы обновить свои предпочтения (например, если я теперь ищу кофейню вместо ресторана).
6. Как пользователь, я хочу просматривать подробную информацию о заведении, чтобы понять, подходит ли оно мне.
7. Как пользователь, я хочу добавить заведение в избранное, чтобы сохранить понравившиеся места и получать рекомендации на их основе.
8. Как пользователь, я хочу скрыть заведение, которое мне не понравилось, чтобы оно больше не отображалось в списке.
9. Как пользователь, я хочу получать рекомендации заведений на основе моего избранного, чтобы быстрее находить интересные мне места.
10. Как пользователь, я хочу скачать отчёт в формате Excel, чтобы сохранить свои действия и предпочтения в приложении.

Варианты использования администратора:

1. Как администратор, я хочу войти в аккаунт с использованием специального логина и пароля, чтобы получить доступ к расширенному функционалу управления данными заведений.
2. Как администратор, я хочу изменить файл, из которого загружается список заведений, чтобы обновить данные (например, при устаревании текущего списка).
3. Как администратор, я хочу просмотреть список всех заведений, чтобы подобрать интересное место без фильтрации.
4. Как администратор, я хочу заполнить анкету, чтобы получить список заведений, соответствующих моим интересам (например, только рестораны с итальянской кухней).
5. Как администратор, я хочу изменить ранее заполненную анкету, чтобы обновить свои предпочтения (например, если я теперь ищу кофейню вместо ресторана).
6. Как адмнистратор, я хочу просматривать подробную информацию о заведении, чтобы понять, подходит ли оно мне.
7. Как администратор, я хочу добавить заведение в избранное, чтобы сохранить понравившиеся места и получать рекомендации на их основе.
8. Как администратор, я хочу скрыть заведение, которое мне не понравилось, чтобы оно больше не отображалось в списке.
9. Как администратор, я хочу получать рекомендации заведений на основе моего избранного, чтобы быстрее находить интересные мне места.
10. Как администратор, я хочу скачать отчёт в формате Excel, чтобы сохранить свои действия и предпочтения в приложении.

## 2.4 Нефункциональные требования

Язык программирования: C#

Используемые инструменты: Visual Studio, Windows Forms

База данных: локальная  
СУБД: MS SQL

# 3. **Детальные требования**

## 3.1 Функциональные требования

User Story 1

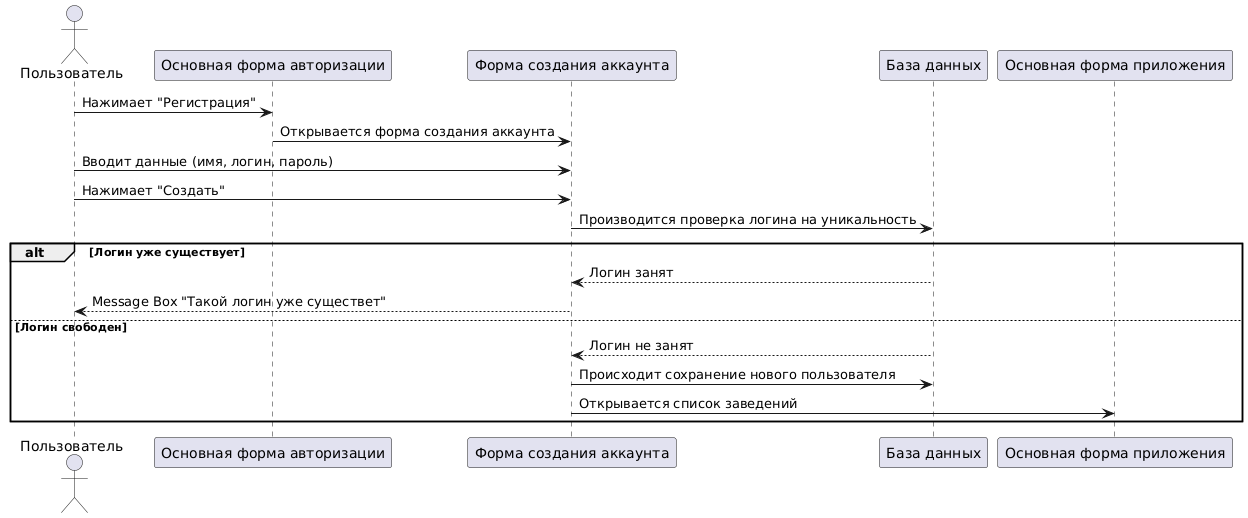
*Как пользователь, я хочу создать аккаунт, чтобы пользоваться приложением и работать с заведениями.*

Исполнитель: Пользователь

Цель: Создать аккаунт

Сценарий:

1. Пользователь нажимает кнопку “Регистрация”, открывается форма создания аккаунта.
2. Пользователь вводит данные (Имя,Логин, Пароль) и нажимает кнопку “Создать”
3. Производится проверка логина на уникальность: а) логин уже существует - пользователю предлагается поменять его; б) логин свободен - происходит сохранение нового пользователя
4. Открывается Основная форма приложения



UML диаграмма последовательностей User Story 1

User Story 2

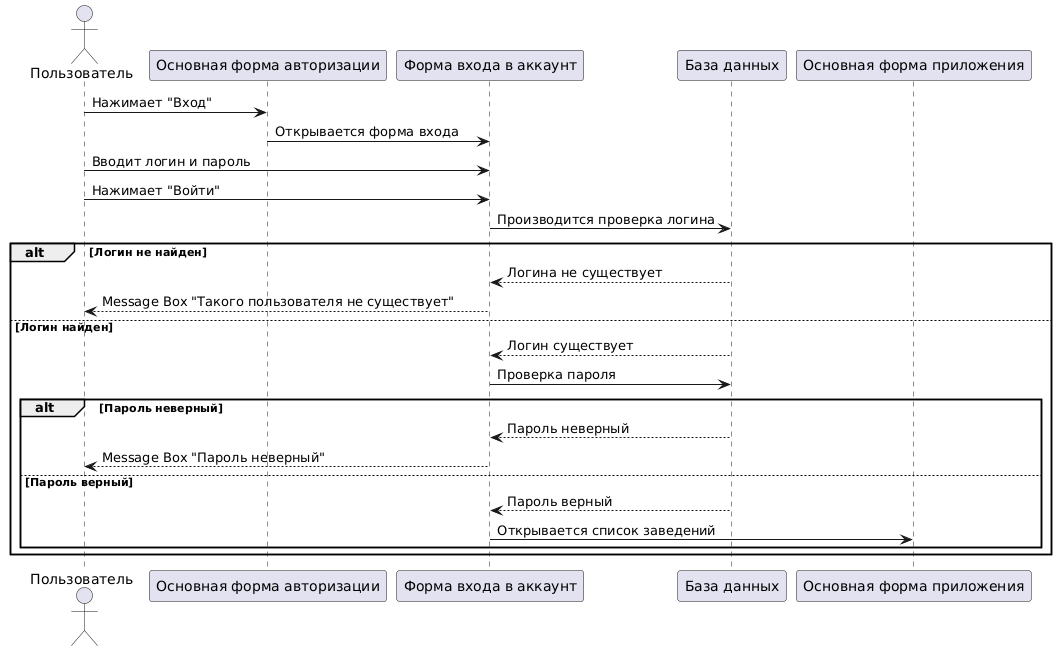
*Как пользователь, я хочу войти в свой аккаунт, чтобы продолжить использование приложения с моими данными.*

Исполнитель: Пользователь

Цель: Войти в уже существующий аккаунт

Сценарий:

1. Пользователь находится на основной форме авторизации, нажимает кнопку “Вход”.
2. Открывается форма входа в аккаунт.
3. Вводит логин и пароль, нажимает кнопку “Вход”.
4. Происходит проверка данных: а) если логин/пароль неверны — сообщение об ошибке; б) если данные верны — открывается основная форма приложения



UML диаграмма последовательностей User Story 2

User Story 3

*Как пользователь, я хочу просмотреть список всех заведений, чтобы подобрать интересное место без фильтрации.*

Исполнитель: Пользователь

Цель: Просмотр всех заведений

Сценарий:

1. Пользователь входит в приложение/проходит регистрацию.
2. На основной форме отображается таблица “Список заведений”.
3. Пользователь просматривает все заведения.



UML диаграмма последовательностей User Story 3

User Story 4

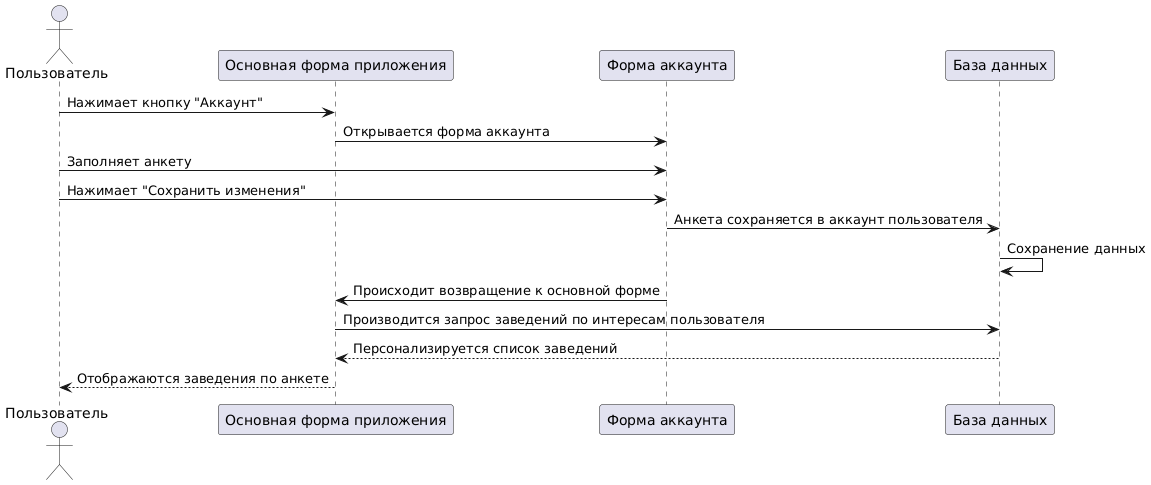
*Как пользователь, я хочу заполнить анкету, чтобы получить список заведений, соответствующих моим интересам.*

Исполнитель: Пользователь

Цель: Заполнение анкеты

Сценарий:

1. Пользователь нажимает кнопку аккаунт и переходит в ЛК.
2. Отмечает предпочитаемые типы, категории, кухни и средний чек, нажимает “Сохранить изменения”.
3. Система сохраняет анкету.
4. Происходит возвращение к основной форме.
5. Производится запрос интересующих заведений и персонализируется список заведений.
6. В таблице отображаются заведения по анкете.



UML диаграмма последовательностей User Story 4

User Story 5

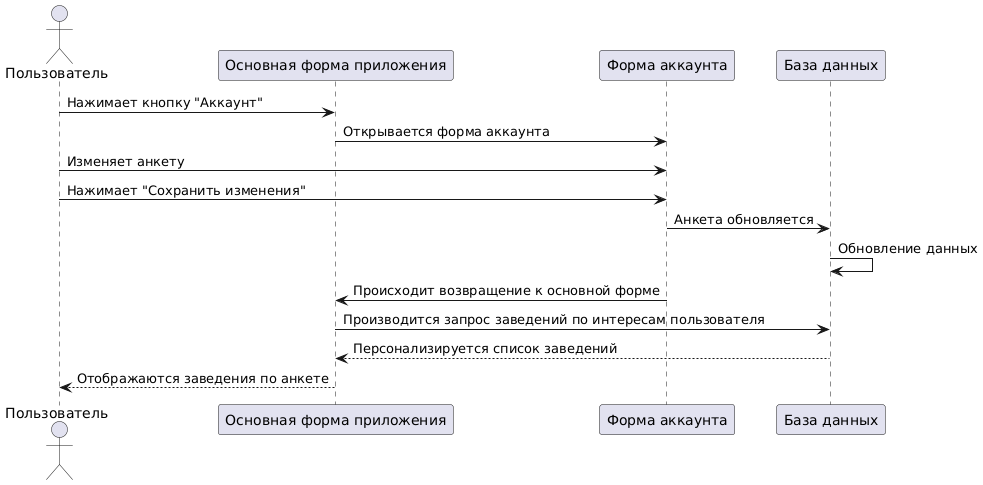
*Как пользователь, я хочу изменить ранее заполненную анкету, чтобы обновить свои предпочтения (например, если я теперь ищу кофейню вместо ресторана).*

Исполнитель: Пользователь

Цель: Изменить ранее заполненную анкету

Сценарий:

1. Пользователь нажимает кнопку “Аккаунт” и открывается форма аккаунта.
2. Пользователь изменяет анкету и нажимает кнопку “Сохранить изменения”.
3. Анкета и база данных обновляется.
4. Происходит возвращение к основной форме.
5. Производится запрос интересующих заведений и персонализируется список заведений.
6. В таблице отображаются заведения по анкете.



UML диаграмма последовательностей User Story 5

User Story 6

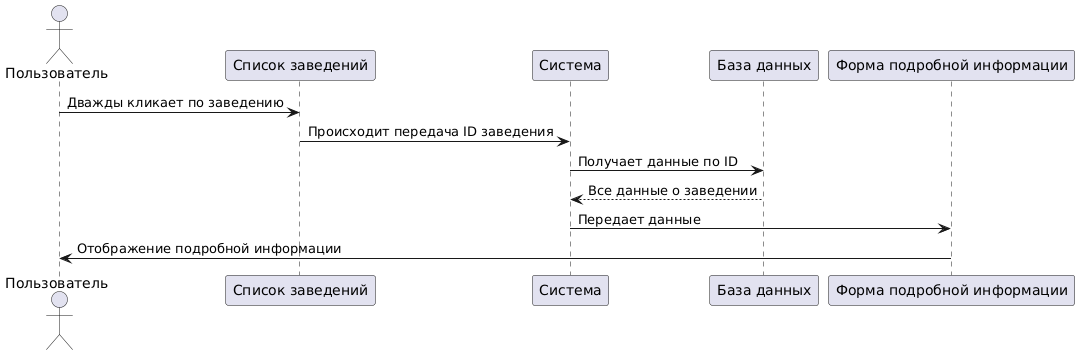
*Как пользователь, я хочу просматривать подробную информацию о заведении, чтобы понять, подходит ли оно мне.*

Исполнитель: Пользователь

Цель: Получить подробную информацию о заведении

Сценарий:

1. Пользователь выбирает заведение, информацию о котором хочет увидеть.
2. Нажимает 2 раза на выбранное заведение в таблице.
3. Открывается форма подробной информации о заведении.
4. Система достает из БД все данные, связанные с этим заведением и отображает их на форме.



UML диаграмма последовательностей User Story 6

User Story 7

*Как пользователь, я хочу добавить заведение в избранное, чтобы сохранить понравившиеся места и получать рекомендации на их основе.*

Исполнитель: Пользователь

Цель: Добавление заведения в избранное

сценарий:

1. Пользователь смотрит побробную информацию о заведении.
2. Пользователь ставит отметку, что ему нравится это заведение (нажимает CheckBox).
3. Система сохраняет заведение в список избранных для этого пользователя.



UML диаграмма последовательностей User Story 7

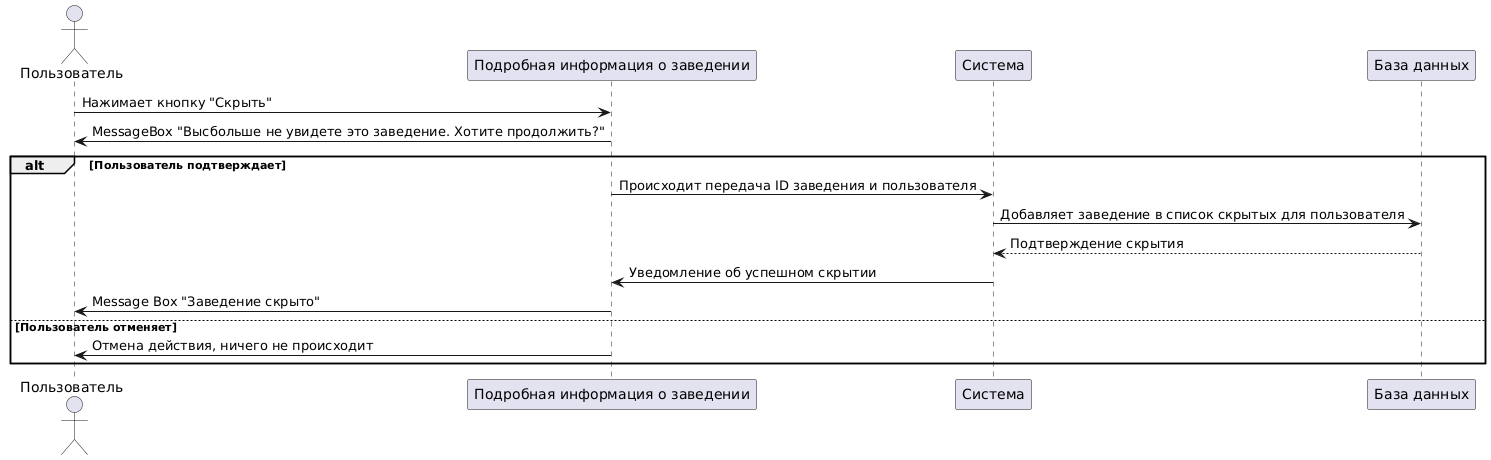
User Story 8

*Как пользователь, я хочу скрыть заведение, которое мне не понравилось, чтобы оно больше не отображалось в списке.*

Исполнитель: Пользователь  
Цель: Скрыть заведение

Сценарий:

1. Пользователь находится на форме подробной информации о заведении.
2. Пользователь нажимает на кнопку “Скрыть”.
3. Система добавляет заведение в список скрытых для этого пользователя.
4. Добавленные в этот список объекты не отображаются в списках предлагаемых заведений.



UML диаграмма последовательностей User Story 8

User Story 9

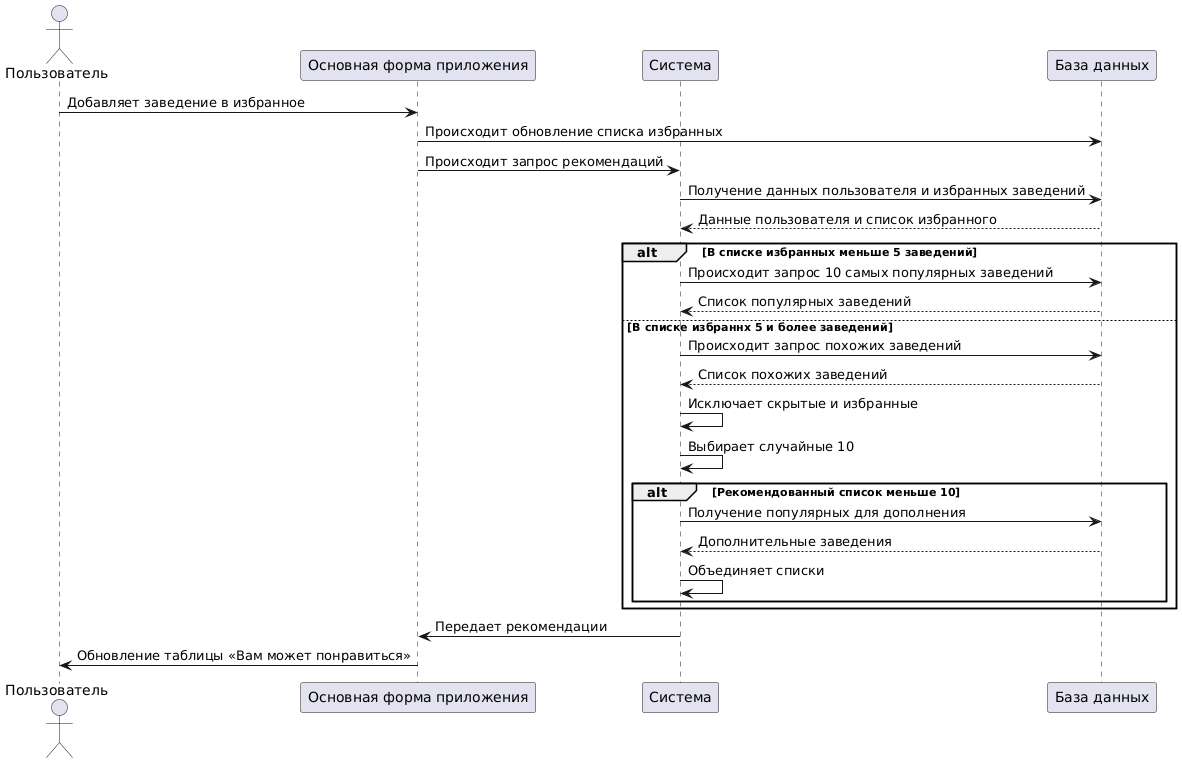
*Как пользователь, я хочу получать рекомендации заведений на основе моего избранного, чтобы быстрее находить интересные мне места.*

Исполнитель: Пользователь

Цель: Получение рекомендации заведений на основе избранного

Сценарий:

1. Пользователь добавляет понравившиеся заведения в избранные.
2. Система получает данные пользователя и его избранные заведения.
3. Система анализирует количество заведений в избранном.
4. Если избранных меньше 5 — система выбирает 10 самых популярных заведений.
5. Если избранных 5 и больше — система находит похожие заведения, исключает скрытые и избранные, выбирает случайные 10.
6. Если рекомендаций меньше 10 — система дополняет список популярными заведениями.
7. Система отображает рекомендации в блоке «Вам может понравиться».



UML диаграмма последовательностей User Story 9

User Story 10

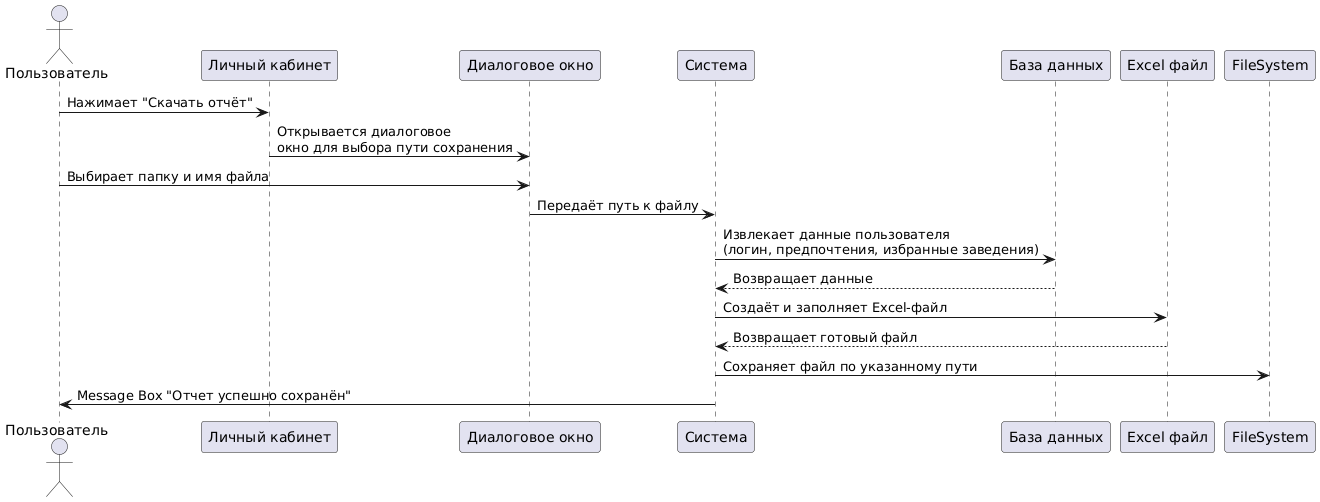
*Как пользователь, я хочу скачать отчёт в формате Excel, чтобы сохранить свои действия и предпочтения в приложении.*

Исполнитель: Пользователь

Цель: Скачать Excel отчет

Сценарий:

1. Пользователь нажимает кнопку "Скачать отчёт".
2. Открывается диалоговое окно выбора места сохранения файла.
3. Пользователь выбирает папку и указывает имя файла.
4. Программа извлекает из базы данных: логин пользователя, предпочтения (типы, категории, средний чек, кухни) и список избранных заведений.
5. Все данные формируются в таблицу Excel с соответствующими заголовками.
6. Файл сохраняется по выбранному пути.
7. Пользователь получает сообщение о сохранении отчета.



UML диаграмма последовательностей User Story 10

Admin Story 1

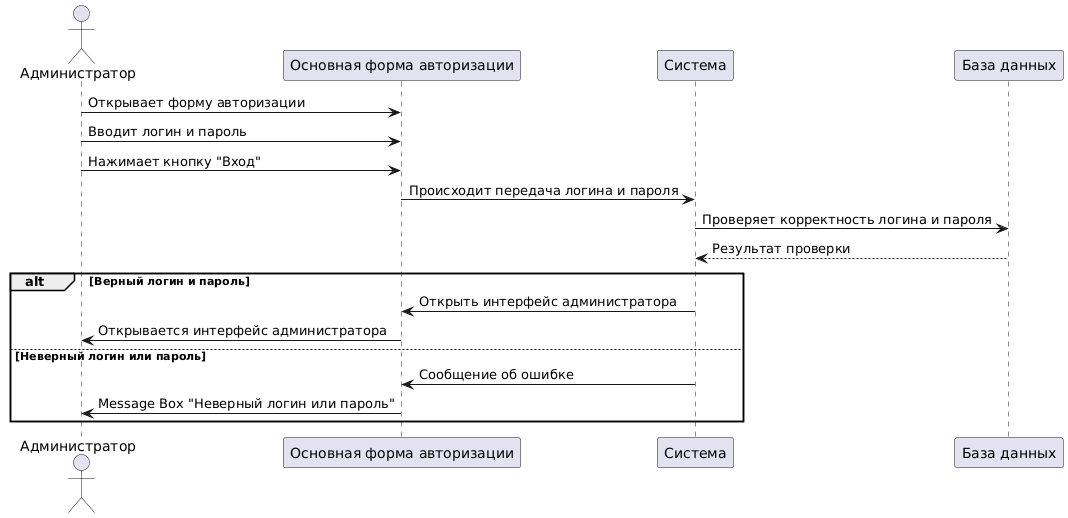
*Как администратор, я хочу войти в аккаунт с использованием специального логина и пароля, чтобы получить доступ к расширенному функционалу управления данными заведений.*

Исполнитель: Администратор

Цель: Войти в аккаунт администратора

Сценарий:

1. Администратор находится на главной форме авторизации.
2. Нажимает на кнопку “Вход”.
3. Вводит специальный логин и пароль для админов.
4. Происходит проверка на правильность логина и пароля: а) если логин/пароль неверны — сообщение об ошибке; б) если данные верны — открывается интерфейс администратора.



UML диаграмма последовательностей Admin Story 1

Admin Story 2

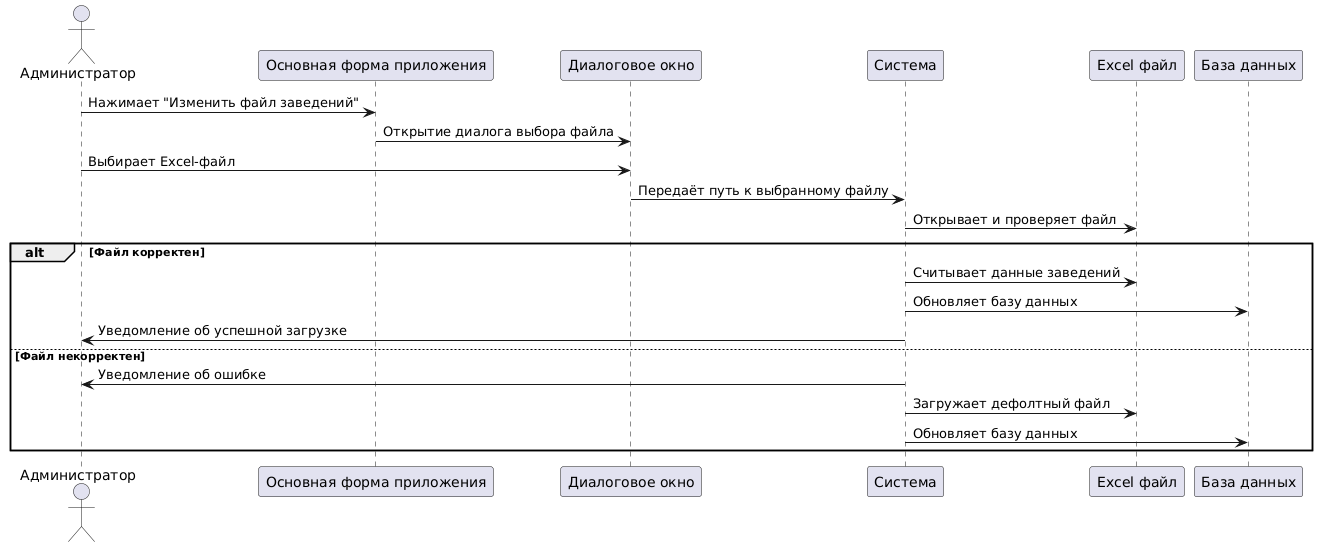
*Как администратор, я хочу изменить файл списка заведений, чтобы обновить данные (например, при устаревании текущего списка).*

Исполнитель: Администратор

Цель: Изменение файла со списком заведений

Сценарий:

1. Администратор находится на Основной форме приложения, нажимает на кнопку “Изменить файл заведений”.
2. Открывается диалоговое окно, администратор выбирает какой файл хочет загрузить.
3. После выбора файла начинается выгрузка списка заведений из файла.
4. Система проверяет данные: а) если файл корректен, то обновляет базу данных; б) если файл некорректен, то сообщается об ошибке и используется дефолтный файл.



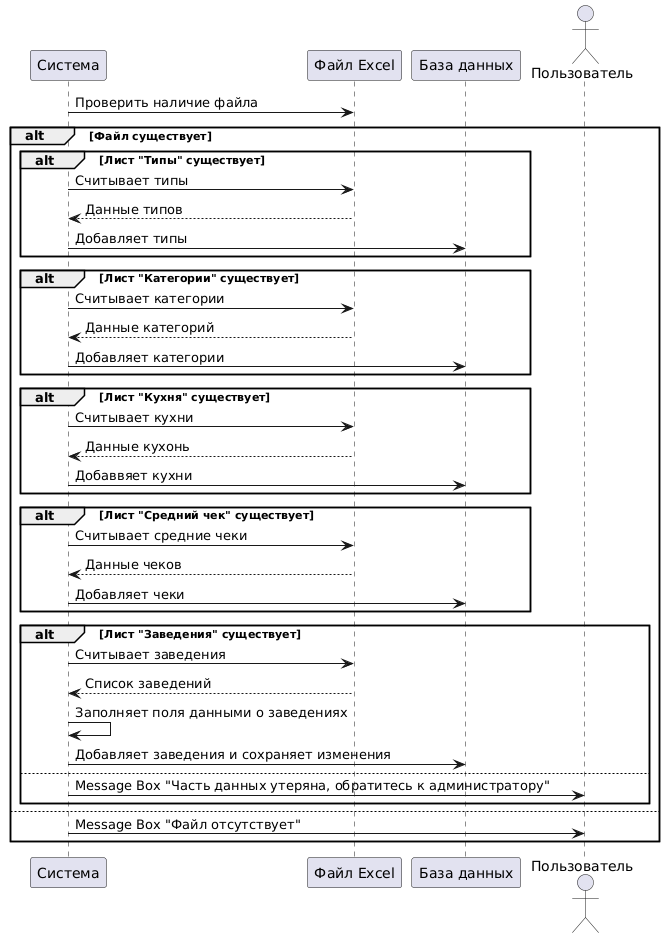
UML диаграмма последовательностей Admin Story 2

Выгрузка списка заведений из файла  
Исполнитель: Система

Цель: Получение текущего списка заведений

Сценарий:

1. Выгрузка происходит при существовании файла.
2. Система считывает категории, типы, кухни и средние чеки из Excel-файла при наличии этих листов в файле.
3. Система добавляет считанные данные в базу данных.
4. Система считывает заведения с файла и заполняет все его поля данными из файла и заполненным ранее категориям, типам, кухням и чекам при наличии листа с заведениями в файле.
5. Система добавляет заведения в базу данных и сохраняет изменения.
6. В случае утери некоторых данных система уведомляет пользователя о потере данных.

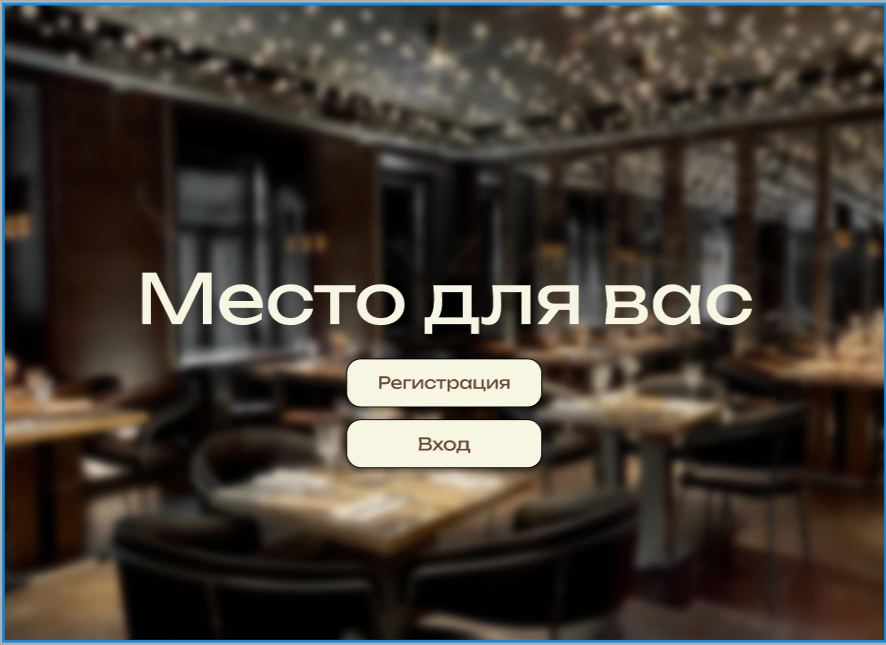


UML диаграмма последовательностей Выгрузки списка заведений из файла

## 3.2 Требования к пользовательскому интерфейсу

Когда пользователь запускает приложение, он видит основную форму авторизации (рисунок 1).На окне авторизации:

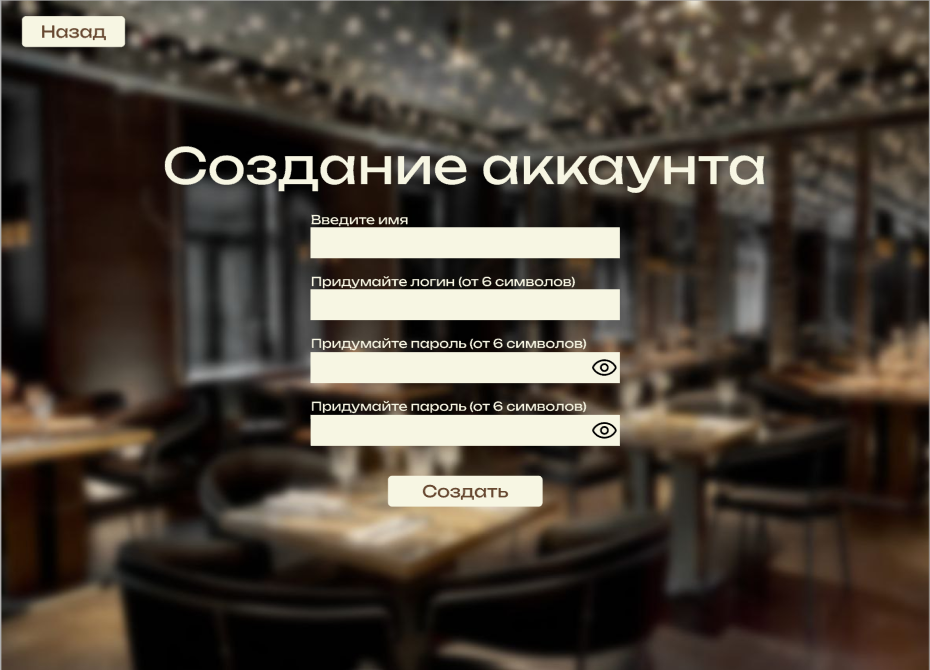
* Label с названием приложения
* Button “Регистрация”
* Button “Вход”



Основная форма авторизации (Рисунок 1)

При нажатии на кнопку “Регистрация” откроется окно “Создание аккаунта” (Рисунок 2). На окне создания аккаунта:

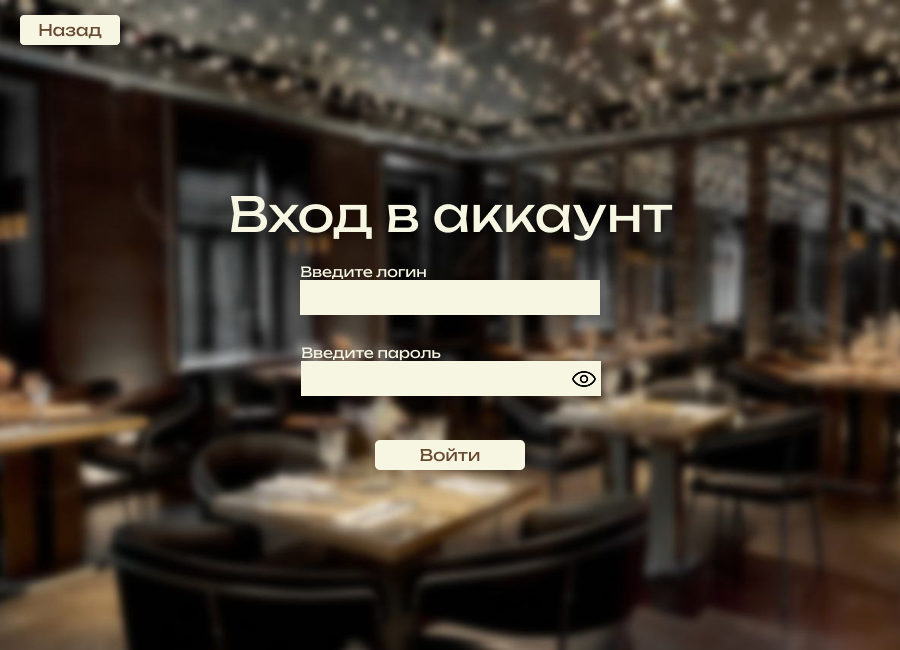
* 4 TextBox-а для заполнения информации - Имя пользователя, логин, пароль, повторение пароля
* Button для сохранения информации о имени, логине и пароле
* Button для возвращения на основную форму авторизации
* 2 Buttona для изменения видимости пароля (видимый - сам пароль, скрытый - \*\*\*\*\*\*)



Создание аккаунта (Рисунок 2)

При нажатии на кнопку “Вход” откроется окно “Вход в аккаунт” (Рисунок 3). На окне входа:

* 2 TextBox-а для ввода логина и пароля
* Button для входа в аккаунт
* Button для возвращения на основную форму авторизации (рисунок 1)
* Button для изменения видимости пароля (видимый - сам пароль, скрытый - \*\*\*\*\*\*)



Вход в аккаунт (Рисунок 3)

После создания аккаунта или входа в уже существующий, пользователь видит Главное окно приложения (рисунок 4). Предложение предназначено для отображения пользователю списка рекомендаций в виде заведений того или иного типа, категории и проч.

Алгоритм рекомендаций работает следующим образом:

Рекомендации выстраиваются на основе избранных (понравившихся) заведений у пользователя.

* При первом входе пользователю отображается весь список заведений в таблице “список заведений”, в таблице “Вам может понравиться” отображаются 10 самых популярных заведений (те, что чаще всего просматривают другие пользователи).
* Когда в избранном 0-4 заведения - в таблице “Вам может понравиться” отображаются 10 самых популярных заведений.
* Когда в избранном 5 заведений и более - система выбирает 10 заведений случайным образом из списка похожих (например, при 10 избранных заведениях формируется список из примерно 20 похожих, из которых отображаются любые 10).
* Если рекомендаций меньше 10, система добавляет популярные заведения, которых нет в избранном или в списке скрытых, чтобы заполнить блок до 10 позиций.

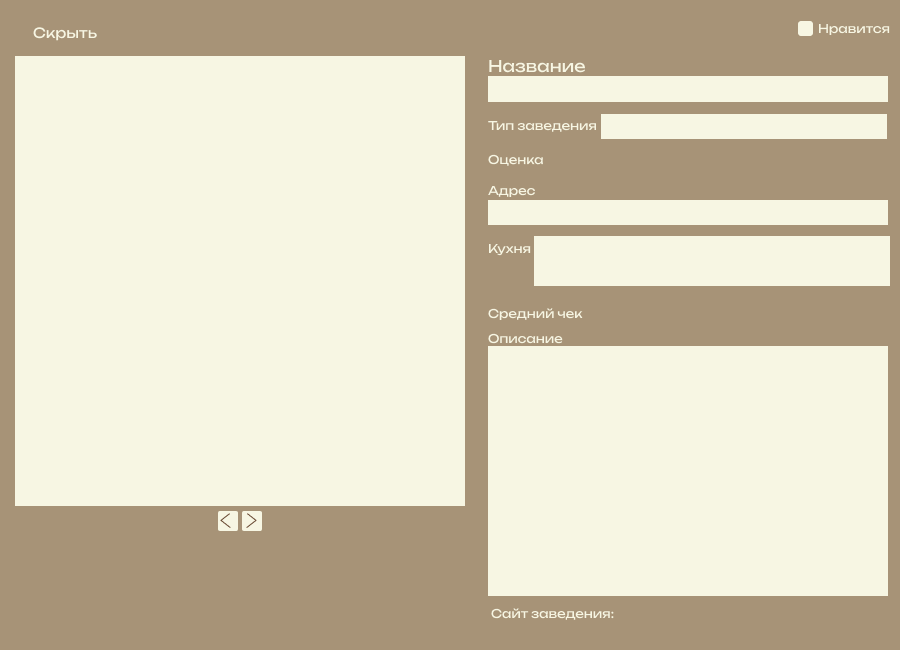
У каждого заведения 2-3 похожих. Похожие добавляются вручную в таблицу. Похожие заведения определяются на основе совпадений по типу, кухне, категории или среднему чеку. Список рекомендаций обновляется после каждого изменения избранного.



Главное окно приложения (рисунок 4)

На форме:

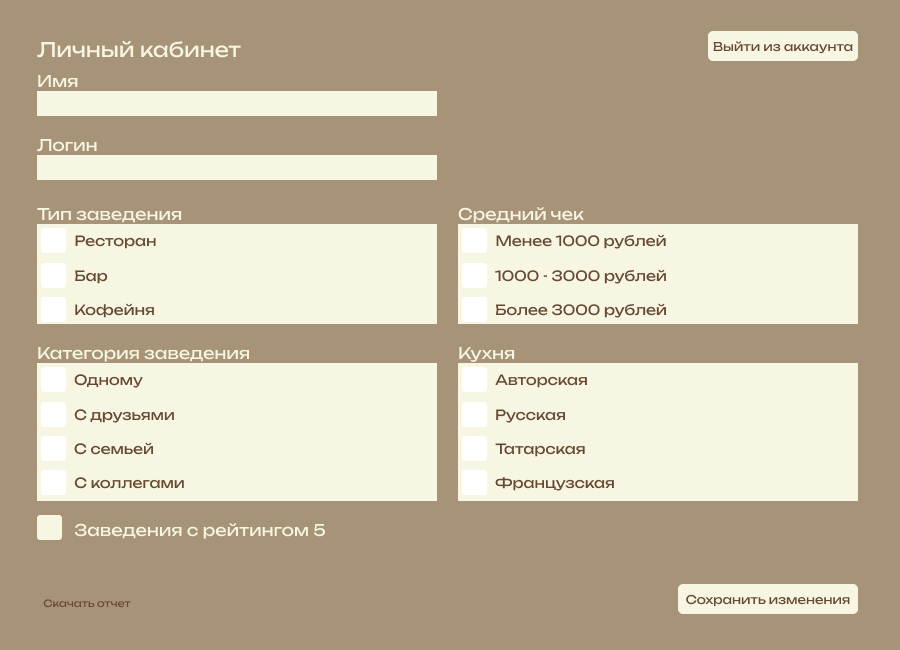
* DataGridView “Список заведений” с отображением списка всех заведений подходящих под выбранные пользователем категории (про категории будет сказано чуть позже). DataGridView имеет следующие столбцы: Название, Тип заведения, Рейтинг. При двойном клике на заведение открывается форма с подробной информацией о заведении (Рисунок 5)
* DataGridView “Вами может понравится” с отображением списка рекомендованных пользователю на основе алгоритма, описанного выше. DataGridView имеет следующие столбцы: Название заведения, Рейтинг. Сортировка в таблице автоматически и всегда по популярности заведения При двойном клике на заведение открывается форма с подробной информацией о заведении (Рисунок 5)
* 4 RadioButton для сортировки: По популярности, По названию (в алфавитном порядке), По типу, По рейтингу
* CheckBox для отображения только избранного (при пустом избранном должна высветиться надпись “Сначала добавьте в избранное”)
* Button “Аккаунт” для просмотра информации об аккаунте. при нажатии происходит переход на окно Личный кабинет (рисунок 6)



Подробная информация про заведение (Рисунок 5)

На форме:

* PictureBox для отображения фотографий выбранного заведения
* 2 кнопки Button Для листания фотографий
* 7 TextBox для отображения названия,типа, оценки, адреса заведения, кухни, среднего чека, описания заведения
* LinkLabel Для отображения ссылки на сайт заведения
* CheckBox для добавления заведения в избранное (понравилось заведение - лайк).
* button “скрыть” - после нажатия заведение больше не будет отображаться ни в одном из datagridview



Аккаунт пользователя (Личный кабинет) (Рисунок 6)

На форме:

* TextBox для Имени (можно поменять)
* TextBox для логина (нельзя поменять)
* CheckedListBox для выбора типа заведения
* CheckedListBox для выбора категории заведения
* CheckedListBox для выбора кухни в заведении
* CheckedListBox для выбора среднего чека
* CheckBox для выбора отображения заведений с рейтингом 5
* Button “Выйти из аккаунта” для выхода из аккаунта. При нажатии происходит переход на Основная форма авторизации (Рисунок 1)
* Button “Сохранить изменения” сохраняет изменения произошедшие Личном кабинете (замена имени, выбор категории/типа заведения)
* Button “Скачать отчет” для скачивания отчета в формате Excel

Изначально при входе в аккаунт заполнены только поля имени и логина.

Таблица для анкетирования

| Тип | Категория | Средний чек | Кухня |
| --- | --- | --- | --- |
| Бар | Одному | < 1000 | Авторская |
| Ресторан | С друзьями | 1000 - 3000 | Русская |
| Кофейня | С семьей | > 3000 | Татарская |
|  | Просто посидеть |  | Французская |
|  | Деловая встреча |  | Итальянская |
|  | С коллегами |  | Азиатская |
|  | Праздник |  | Грузинская |
|  | Коворкинг |  | Нет специализации |
|  |  |  | Европейская |
|  |  |  | Вегетарианская |
|  |  |  | Национальная |
|  |  |  | Халяльная |
|  |  |  | Кавказская |
|  |  |  | Смешанная |

У администратора определенные логин и пароль, также после входа в ЛК администратору открывается Главная форма, но на ней есть Button“Изменить файл заведений”, при нажатии на него открывается диалоговое окно с возможностью выбора Excel файла. Если файл заполнен некорректно, пустой или отсутствует, то используется файл по умолчанию.  
Если администратор загрузил новый файл, то после этого у всех пользователей список заведений выгружается из этого файла.

Главная форма администратора ниже (рисунок 7)



Главное окно администратора(Рисунок 7)

## 

## 3.3 Требования к логической структуре БД

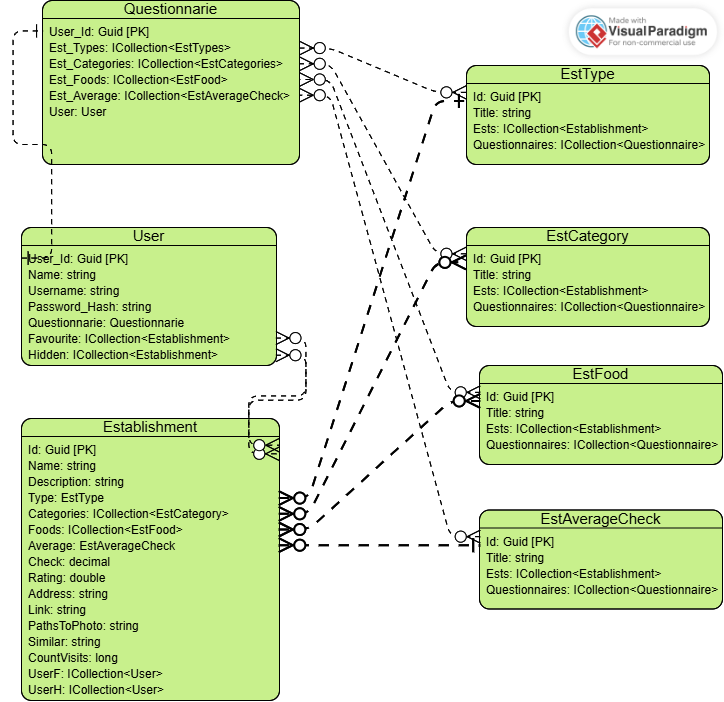


Схема Базы Данных (Рисунок 8)

Таблица **Establishment**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| Id | Guid [PK] | Уникальный идентификатор заведения |
| Name | string | Название заведения |
| Description | string | Описание заведения |
| Type | EstType | Тип заведения |
| Categories | ICollection<EstCategories> | Категории заведения |
| Foods | ICollection<EstFoods> | Кухни заведения |
| Check | decimal | Значение среднего чека заведения |
| Average | EstAverageCheck | Средний чек заведения |
| Rating | double | Рейтинг заведения |
| Address | string | Адрес заведения |
| Link | string | Ссылка на сайт заведения |
| PathsToPhoto | string | Название изображений заведения |
| Similar | string | Список похожих заведений |
| CountVisits | long | Счетчик количества посещений |
| UserF | ICollection<User> | Навигационное свойство на пользователей, разместивших заведение в избранное |
| UserH | ICollection<User> | Навигационное свойство на пользователей, разместивших заведение в скрытое |

Таблица **User**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| User\_Id | Guid [PK] | Уникальный идентификатор пользователя |
| Name | string | Имя пользователя |
| UserName | string | Логин пользователя |
| Password\_Hash | string | Хэш пароля пользователя |
| Questionnaire | Questionnaire | Анкета пользователя |
| Favorite | ICollection<Establishment> | Список избранных заведений |
| Hidden | ICollection<Establishment> | Список скрытых заведений |

Таблица **EstType**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| Id | Guid [PK] | Уникальный идентификатор типа |
| Title | string | Название типа |
| Ests | ICollection<Establishment> | Навигационное свойство на заведения с этим типом |
| Questionnaires | ICollection<Questionnaire> | Пользовательские предпочтения типа заведения |

Таблица **EstCategory**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| Id | Guid [PK] | Уникальный идентификатор категории |
| Title | string | Название категории |
| Ests | ICollection<Establishment> | Навигационное свойство на заведения с этой категорией |
| Questionnaires | ICollection<Questionnaire> | Пользовательские предпочтения в категориях |

Таблица **EstAverageCheck**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| Id | Guid [PK] | Уникальный идентификатор категории среднего чека |
| Title | string | Значение среднего чека |
| Ests | ICollection<Establishment> | Навигационное свойство на заведения с этим средним чеком |
| Questionnaires | ICollection<Questionnaire> | Пользовательские предпочтения в среднем чеке |

Таблица **EstFood**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| Id | Guid [PK] | Уникальный идентификатор категории кухни |
| Title | string | Название кухни заведения |
| Ests | ICollection<Establishment> | Навигационное свойство на заведения с этой кухней |
| Questionnaires | ICollection<Questionnaire> | Пользовательские предпочтения в кухне |

Таблица **Questionnaire**

| **Название** | **Тип** | **Описание** |
| --- | --- | --- |
| User\_Id | Guid [PK] | Идентификатор пользователя |
| Est\_Types | ICollection<EstType> | Предпочитаемые типы |
| Est\_Categories | ICollection<EstCategory> | Предпочитаемые категории |
| Est\_Foods | ICollection<EstFood> | Предпочитаемая кухня |
| Est\_Average | ICollection<EstAverageCheck> | Предпочитаемый средний чек |
| User | User | Пользователь, которому принадлежит анкета |