Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ТЕХНИЧЕСКИЙ ПРОЕКТ**

Telegram-бот «шведский стол»

Киров,

2024

Содержание

[1. Бизнес-процессы 2](#_Toc182784854)

[2. Варианты использования 4](#_Toc182784855)

[3. Пользовательский интерфейс 6](#_Toc182784856)

[4. Передача и хранение данных 13](#_Toc182784857)

[4.1 Передача данных 13](#_Toc182784858)

[4.2 Хранение данных 13](#_Toc182784859)

[4.3 Выбор целевой СУБД 16](#_Toc182784860)

[5. Технологический стек 19](#_Toc182784861)

[6. Разработка алгоритма решения задачи 20](#_Toc182784862)

[7. Алгоритмы решения задач 26](#_Toc182784863)

[7.1 Авторизация 26](#_Toc182784864)

[7.2 Одобрение заявки на авторизацию 26](#_Toc182784865)

[7.3 Создание заказа 26](#_Toc182784866)

[7.4 Одобрение заявки на создание заказа 26](#_Toc182784867)

# Бизнес-процессы

На рисунке 1 представлена контекстная диаграмма IDEF0 для Telegram-бота.

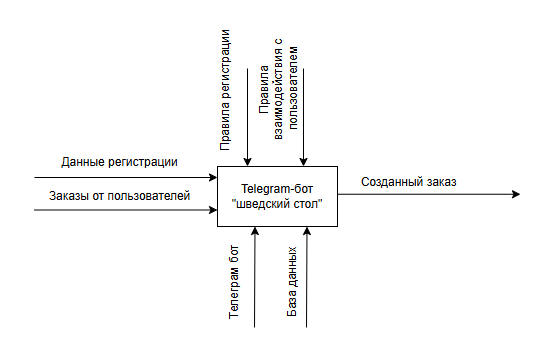


Рисунок 1 – контекстная диаграмма Telegram-бота

Внешней входящей информацией являются:

* Данные регистрации;
* Заказы от пользователей.

Управляющим воздействием будут являться:

* Правила регистрации;
* Правила взаимодействия с пользователем.

Механизмом управления будут являться:

* Телеграм-бот;
* База данных.

Внешней исходящей информацией будет являться:

* Созданный заказ.

Для разбиения сложного процесса на составляющие подфункции применяется декомпозиция.

Диаграмма декомпозиции, представляет контекстную функцию, разложенную на 3 составляющих более простых функций.

1. Создание аккаунта в telegram-боте «шведский стол».
2. Вход в telegram-бот
3. Обработка заказов

На рисунке 2 показана декомпозиция контекстной диаграммы telegram-бота.

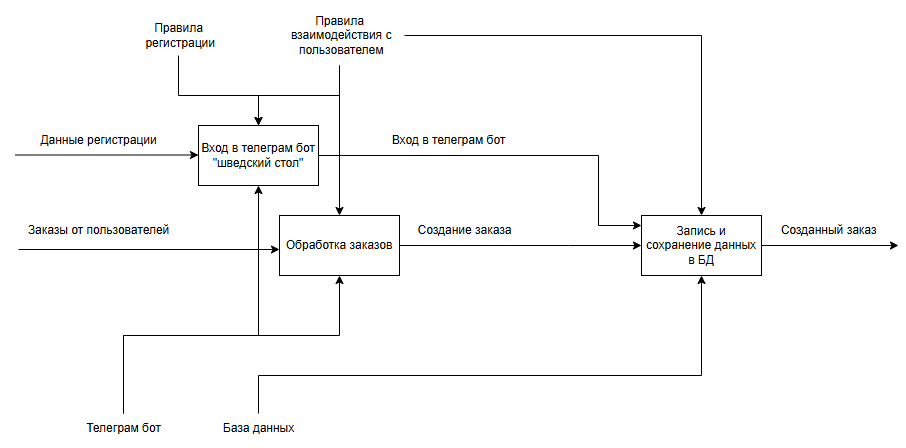


Рисунок 2 – Декомпозиция контекстной диаграммы

# 2. **Варианты использования**

Диаграмма прецедентов для Telegram-бота для шведского стола представлена на Рисунке 3.

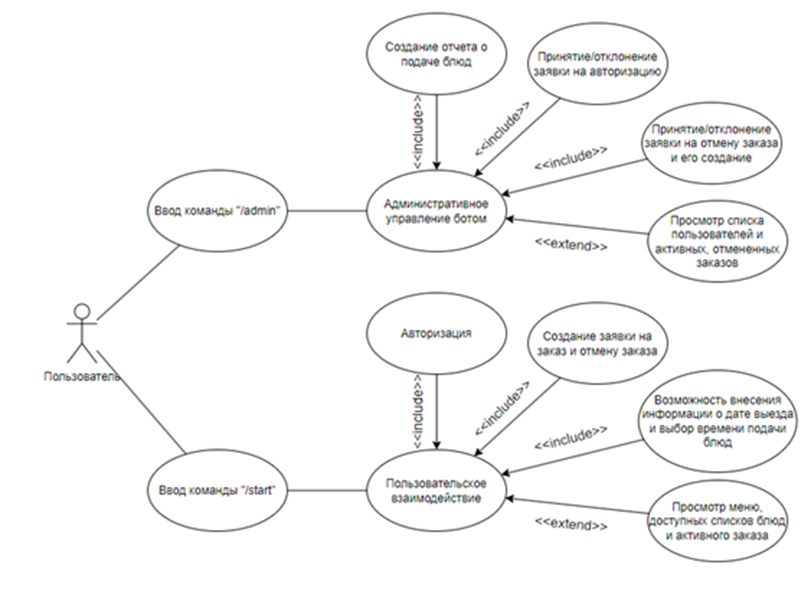


Рисунок 3 – Диаграмма прецендентов

Исходя из данной диаграммы возможны 2 варианта использования:

* Ввод команды “/admin”.
* Ввод команды “/start”.

Далее представлены отдельные диаграммы для ранее перечисленных вариантов использования.

Диаграмма прецендента «Ввод команды “/admin”» представлена на Рисунке 4.

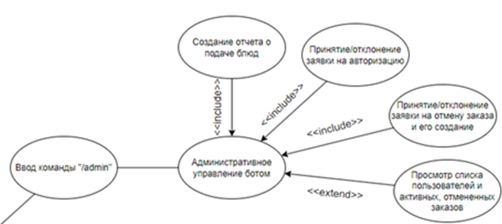


Рисунок 4 – Диаграмма прецедента «Ввод команды “/admin”»

Диаграмма прецендента «Ввод команды “/start”» представлена на Рисунке 5.

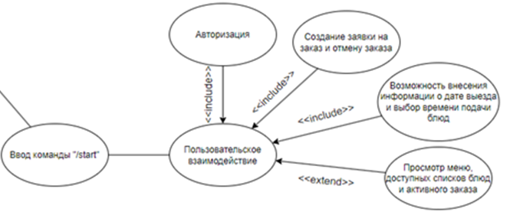


Рисунок 5 – Диаграмма прецедента «Ввод команды “/start”»

# 3. Пользовательский интерфейс

Ниже представлены экранные формы с пояснениями.

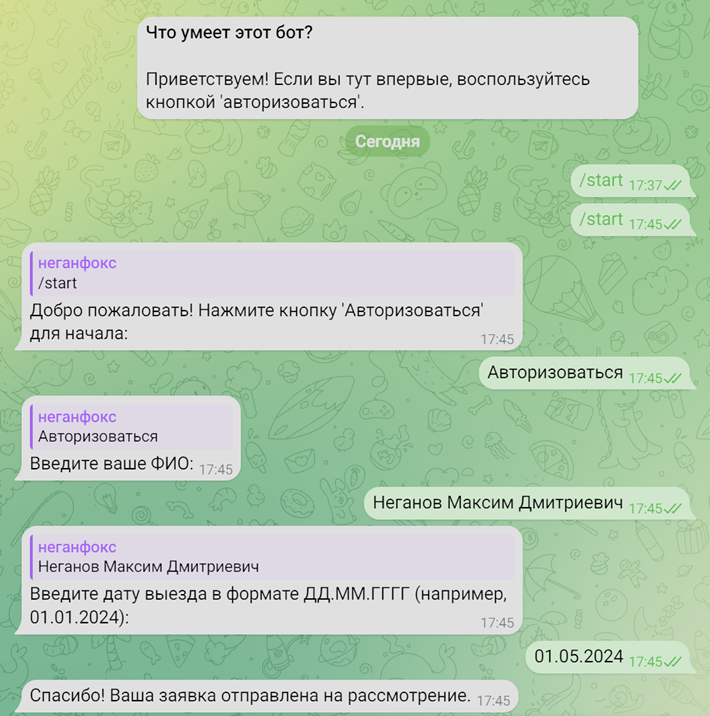


Рисунок 6 – Вид регистрации пользователя в Telegram-боте

При запуске telegram-бота предлагается авторизоваться, с указанием ФИО и даты выезда, если ранее аккаунт не был создан. После одобрения заявки на авторизацию, данные пользователя записываются в базу данных. Сам telegram-бот написан при помощи библиотеки telebot с использованием datetime и xlwt для создания отчета.

В telegram-боте есть три главные кнопки «Меню», «Заказать», «Отменить заказ», при нажатии на которые происходят соответствующие действия.

При нажатии на кнопку «Меню», бот присылает сообщение со списком доступным блюд и их номером. В меню можно подобрать для себя комплект пищи для завтрака, обеда или ужина.

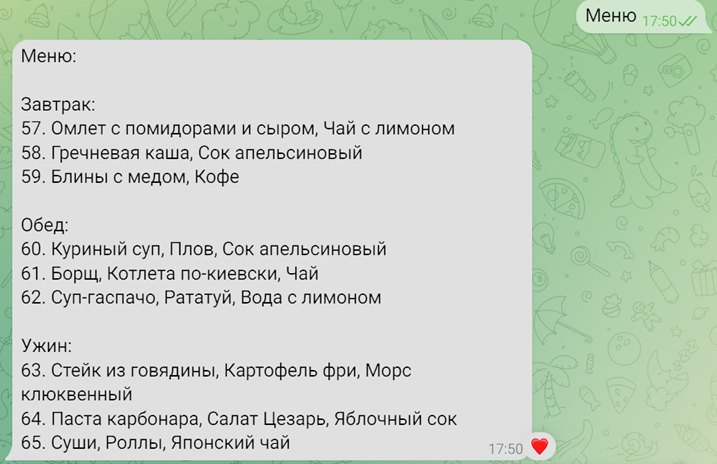


Рисунок 7 – Вид просмотра доступного меню

При нажатии на кнопку «Заказать» предлагается выбрать время приема пищи на завтрак, обед или ужин. После выбора времени приема пользователю доступен просмотр меню на выбранное время и выбор номер блюда для заказа. После одобрения заявки, заказ пользователя заносится в базу данных.

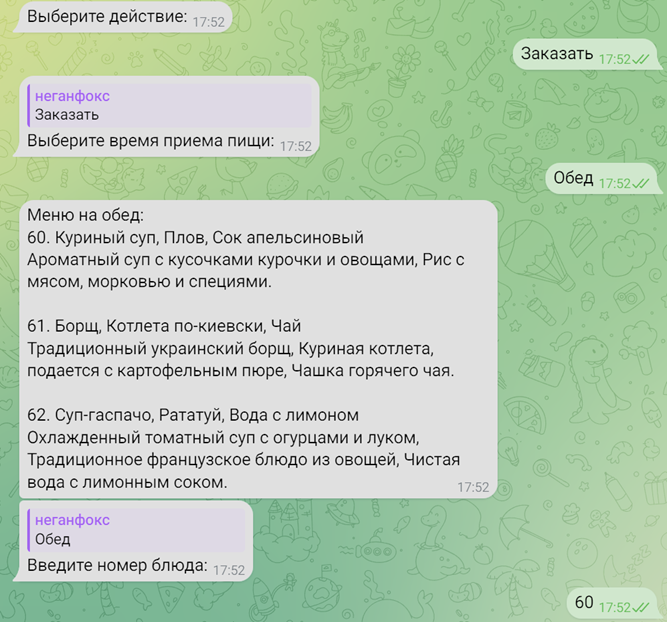


Рисунок 8 – Вид создания заказа пользователем

При нажатии на кнопку «Отменить заказ» бот высылает пользователю его активные заказы и предлагает выбор заказа для отмены. Если истекло время отмены заказа, то совершить данное действие не получится.

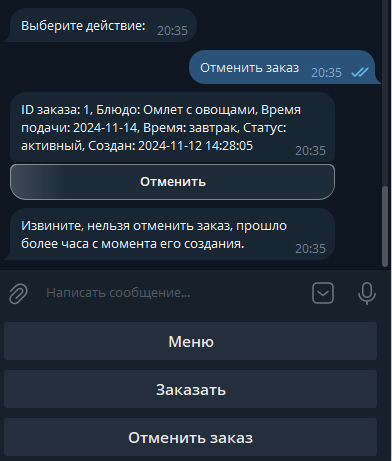


Рисунок 9 – Вид отмены заказа пользователем

Далее админ-панель, вход осуществляется через команду «/admin» с вводом пароля.

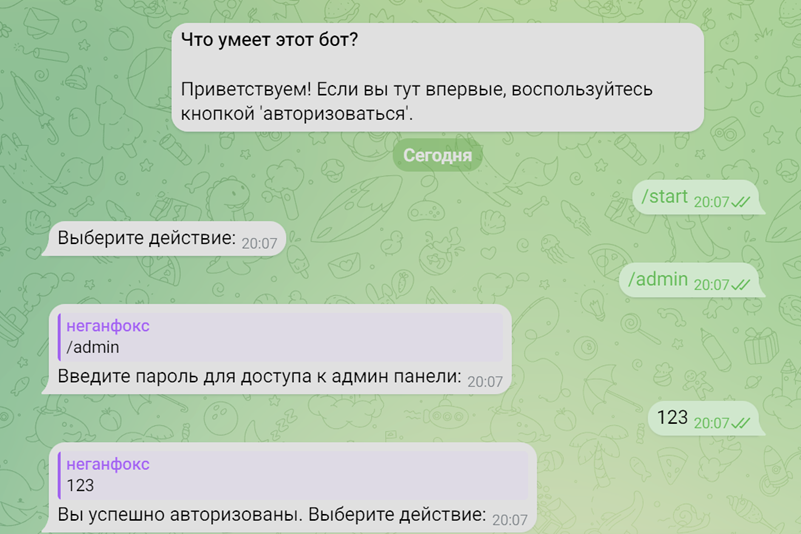


Рисунок 10 – Вид авторизации в админ-панель

После успешного входа становятся доступны следующие кнопки: просмотр заявок, просмотр заказов, пользователи, сделать отчёт.

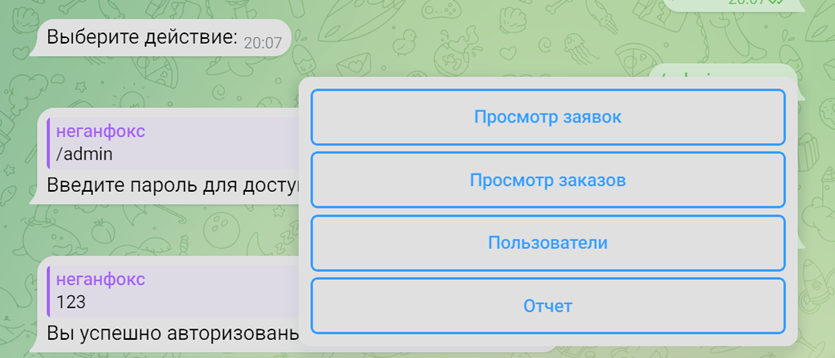


Рисунок 11 – Вид меню админ-панели

При нажатии на кнопку «просмотр заявок» бот высылает все заявки, требующие одобрения или отклонения со стороны администратора, в базе данных пользователь до одобрения его заявки помечен статусом «в ожидании», после одобрения «активный». Если заявка пользователя одобрена, администратору предлагается выбор свободного стола со свободными местами, стол будет прикреплен к пользователю.

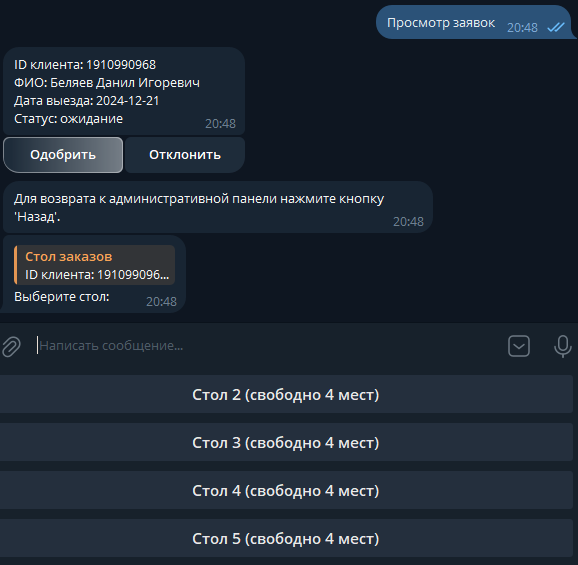


Рисунок 12 – Вид просмотра заявок в админ-панели

При нажатии на кнопку «просмотр заказов» администратору предлагается выбор между просмотром активных заказов, заказов в ожидании и отмененных заказов. В активных заказов можно просматривать действующие заказы пользователей и отменять их. В заказах в ожидании можно просматривать заказы, требующие одобрения со стороны администратора для их активации. В отмененных заказах можно подтверждать удаление заказа от пользователя.

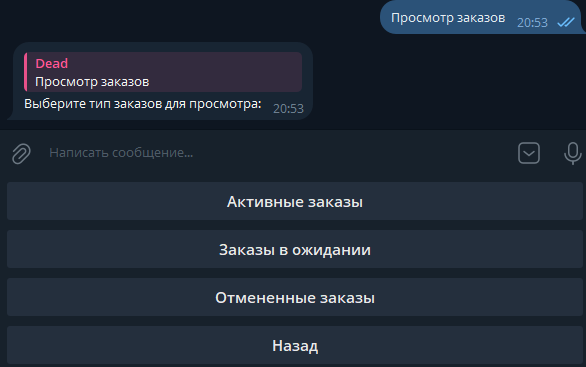


Рисунок 13 – Вид просмотра заказов в админ-панели

При нажатии на кнопку «Пользователи» бот высылает список всех активных пользователей, чья дата выезда еще не истекла и предлагает возможность удалить их аккаунт с подтверждением. При удалении пользователя все его существующие заказы удаляются из базы данных, а также освобождается стол и место, которое он занимал.

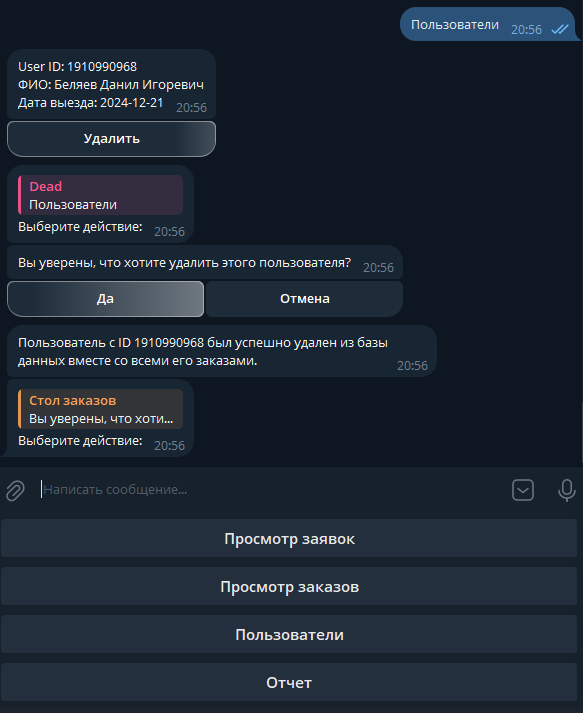


Рисунок 14 – Вид просмотра пользователей в админ-панели

При нажатии на кнопку «Отчет» бот высылает сгенерированный отчет в формате .xls.

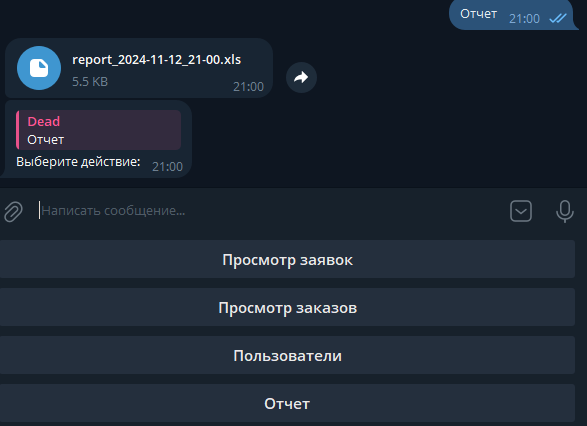


Рисунок 15 – Вид получения сгенерированного отчета в админ-панели

В сгенерированном отчете имеется информация по общему количество пользователей, заказов и блюд в меню, а также заказанные блюда с их количеством и названием. Ниже располагается информация о заказах пользователя с его данными, названием блюда и временем. Информация в отчете берется запросом с таблиц пользователей, блюд и столов в базе данных.

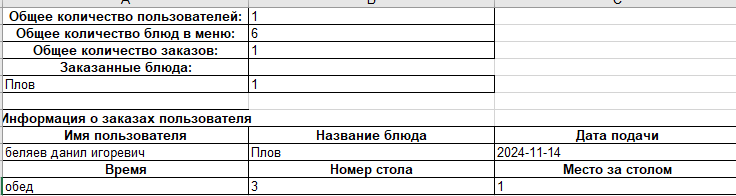


Рисунок 16 – Вид сгенерированного отчета

# 4. Передача и хранение данных

## 4.1 Передача данных

Диаграмма DFD для Telegram-бота для шведского стола представлена на Рисунке 17.

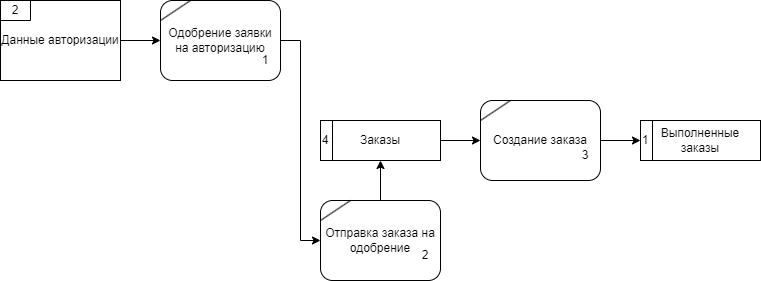


Рисунок 17 – Диаграмма DFD

На данной диаграмме DFD представлен процесс деятельности пользователя с точки зрения передвижения данных по системе. Из сущности «Данные авторизации» информация идёт в процесс «Одобрение заявки на авторизацию», далее идет процесс «Просмотр меню», которые далее доставляются в сущность «Заказы», после чего из сущности «Заказы» попадаем в процесс «Создание заказа» и «Создание заказа» попадает в сущность «Выполненные заказы»

## 4.2 Хранение данных

В таблице 1 выделен набор сущностей с их свойствами для telegram-бота «шведский стол».

Таблица 1 – Набор сущностей и их свойств

| **Сущность** | **Свойство** | **Тип** | **Уникальность** | **Обязательность заполнения** | **Ограничения** |
| --- | --- | --- | --- | --- | --- |
| **Dishes** | dish\_id | SERIAL | Да | Да | Первичный ключ, только числа |
|  | dish\_name | VARCHAR(255) | Нет | Да | Нет |
|  | dish\_description | TEXT | Нет | Нет | Нет |
|  | meal | VARCHAR(50) | Нет | Да | Завтрак, обед или ужин |
| **Orders** | order\_id | SERIAL | Да | Да | Первичный ключ, только числа |
|  | user\_id | INTEGER | Нет | Да | Внешний ключ, ссылка на users(user\_id) |
|  | delivery\_date | DATE | Нет | Да | Только даты |
|  | time | VARCHAR(50) | Нет | Да | Только строки |
|  | status | VARCHAR(50) | Нет | Да | Статус заказа |
|  | created\_at | TIMESTAMP | Нет | Нет | По умолчанию текущая дата и время |
| **Order\_dishes** | order\_id | INTEGER | Нет | Да | Внешний ключ, ссылка на orders(order\_id) |
|  | dish\_id | INTEGER | Нет | Да | Внешний ключ, ссылка на dishes(dish\_id) |
| **Users** | user\_id | BIGINT | Да | Да | Первичный ключ, только числа |
|  | full\_name | VARCHAR(255) | Нет | Нет | Нет |
|  | departure\_date | DATE | Нет | Нет | Только даты |
|  | table\_number | INTEGER | Нет | Нет | Ссылка на tables(table\_number) |
|  | seat\_number | INTEGER | Нет | Нет | Только числа |
|  | status | VARCHAR(50) | Нет | Нет | Статус пользователя |
| **Tables** | table\_number | SERIAL | Да | Да | Первичный ключ, только числа |
|  | seats\_count | INT | Нет | Да | Только числа |
|  | occupied\_seats\_count | INT | Нет | Да | Только числа |

Тип связей между Users и Orders: Один ко многим — у одного пользователя может быть несколько заказов.

Тип связей между Orders и Order\_dishes: Один ко многим — один заказ может включать несколько блюд.

Тип связей между Dishes и Order\_dishes: Один ко многим — одно блюдо может быть частью нескольких заказов.

Тип связей между Tables и Users: Один ко многим — один стол может быть занят несколькими пользователями.

Логическая ER диаграмма представлена на рисунке 18.

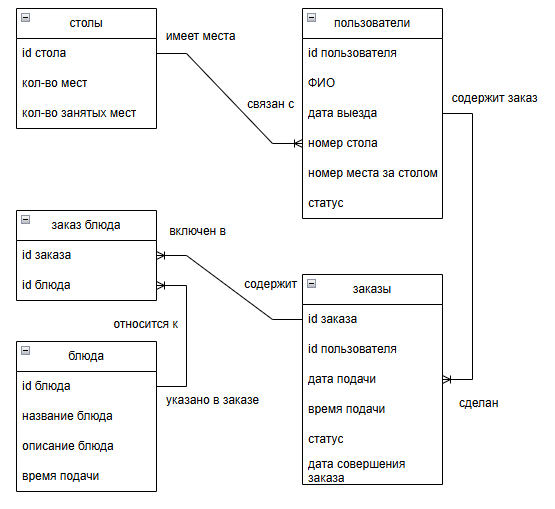


Рисунок 18 – Логическая ER диаграмма базы данных.

На ER диаграмме изображено 5 таблиц: Пользователи, заказы, блюда, заказ блюда, столы.

В таблице «пользователи» - содержатся данные о всех пользователях, которые есть в telegram-боте «шведский стол». Таблица содержит следующие данные:

* id пользователя;
* ФИО;
* Дата выезда;
* Номер стола;
* Номер места за столом;
* Статус.

В таблице «столы» - содержатся данные о всех столах с количеством общих и занятых мест. Таблица содержит следующие данные:

* Id стола;
* Кол-во мест;
* Кол-во занятых мест.

В таблице «заказы» - содержатся данные о всех заказах блюд от пользователей. Таблица содержит следующие данные:

* id заказа;
* id пользователя;
* Дата подачи;
* Время подачи;
* Статус;
* Дата совершения заказа.

В таблице «блюда» - содержатся данные о всех блюдах с описанием и времени, на которое подается блюдо. Таблица содержит следующие данные:

* id блюда;
* Название блюда;
* Описание блюда;
* Время подачи.

В таблице «заказ блюда» - содержатся данные о заказе с закрепленным к нему блюдом. Таблица содержит следующие данные:

* id заказа;
* id блюда.

## 4.3 Выбор целевой СУБД

В качестве целевой СУБД была выбрана PostgreSQL. PostgreSQL — это мощная объектно-реляционная система управления базами данных, которая обеспечивает высокий уровень производительности и гибкости. Она обладает отличной масштабируемостью, поддерживает сложные запросы, транзакции, расширенные функции работы с данными и стандарты SQL. Достоинствами PostgreSQL являются надежность, открытый исходный код и активное сообщество, что делает её удобной для сложных проектов и обработки больших объемов данных.

Описание полей созданных таблиц представлены в таблице 2.

Таблица 2 – Описание полей таблиц

| **Таблица** | **Атрибут** | **Тип** | **Размер** |
| --- | --- | --- | --- |
| **tables** | table\_number | SERIAL | - |
|  | seats\_count | INTEGER | - |
|  | occupied\_seats\_count | INTEGER | - |
| **users** | user\_id | BIGINT | - |
|  | full\_name | VARCHAR | 255 символов |
|  | departure\_date | DATE | - |
|  | table\_number | INTEGER | - |
|  | seat\_number | INTEGER | - |
|  | status | VARCHAR | 50 символов |
| **dishes** | dish\_id | SERIAL | - |
|  | dish\_name | VARCHAR | 255 символов |
|  | dish\_description | TEXT | - |
|  | meal | VARCHAR | 50 символов |
| **orders** | order\_id | SERIAL | - |
|  | user\_id | INTEGER | - |
|  | delivery\_date | DATE | - |
|  | time | VARCHAR | 50 символов |
|  | status | VARCHAR | 50 символов |
|  | created\_at | TIMESTAMP | - |
| **order\_dishes** | order\_id | SERIAL | - |
|  | dish\_id | INTEGER | - |

Физическая ER диаграмма представлена на рисунке 19.

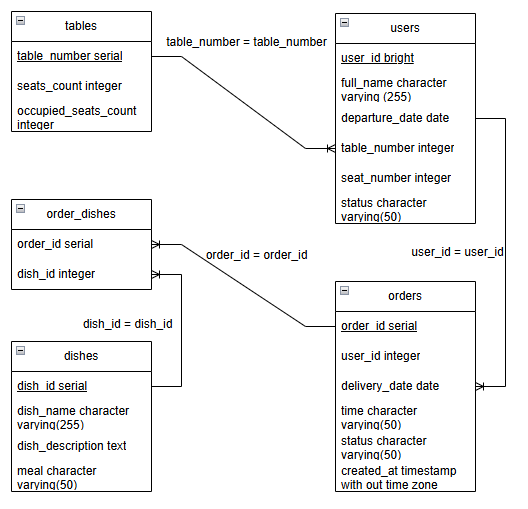


Рисунок 19 – Физическая ER диаграмма базы данных.

# 5. Технологический стек

Для реализации проекта Telegram-бот «шведский стол» используются следующие языки программирования и технологии:

* Python – мультипарадигмальный высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.
* PostgreSQL – свободная объектно-реляционная система управления базами данных (СУБД). Существует в реализациях для множества UNIX-подобных платформ, включая различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.
* Библиотеки и фреймворки. В проекте будут использоваться специализированные библиотеки и фреймворки, которые обеспечат создание функционала подключения telegram-бота и отчетов. В том числе: xlwt для создания отчета. Telebot для реализации telegram-бота. Psycopg2 для работы с базой данных postgresql. Datetime для работы с временем.

# Разработка алгоритма решения задачи

Алгоритм: **Авторизация пользователя.**

Схема алгоритма «Авторизация пользователя» представлена на рисунке 20.

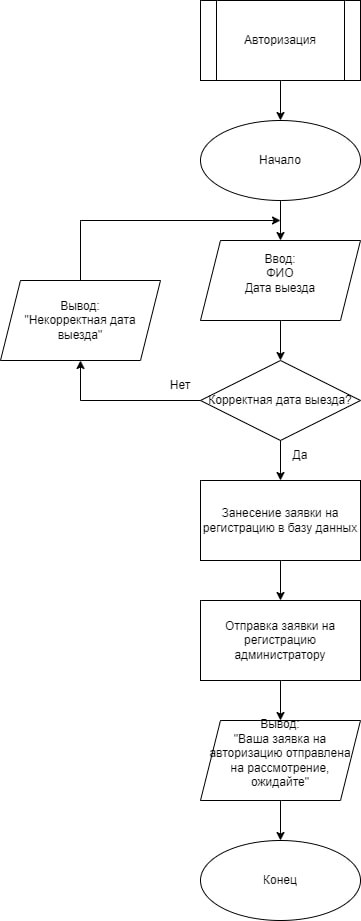


Рисунок 20 - Схема алгоритма «Авторизация пользователя»

Алгоритм: **Создание заказа.**

Схема алгоритма «Создание заказа» представлена на рисунке 21.

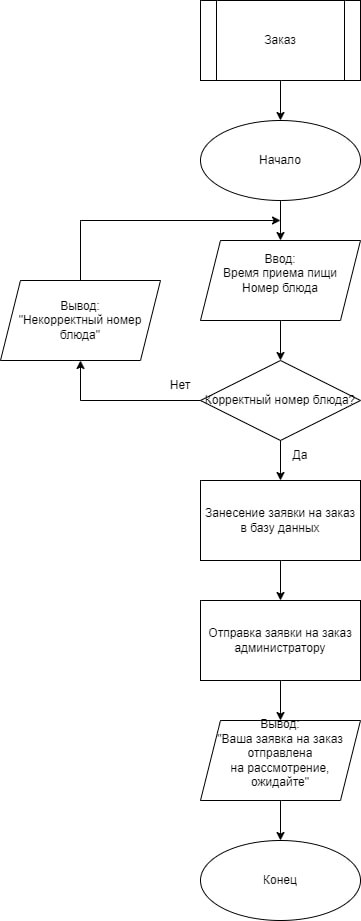


Рисунок 21 - Схема алгоритма «Создание заказа»

Алгоритм: **Отмена заказа.**

Схема алгоритма «Отмена заказа» представлена на рисунке 22.

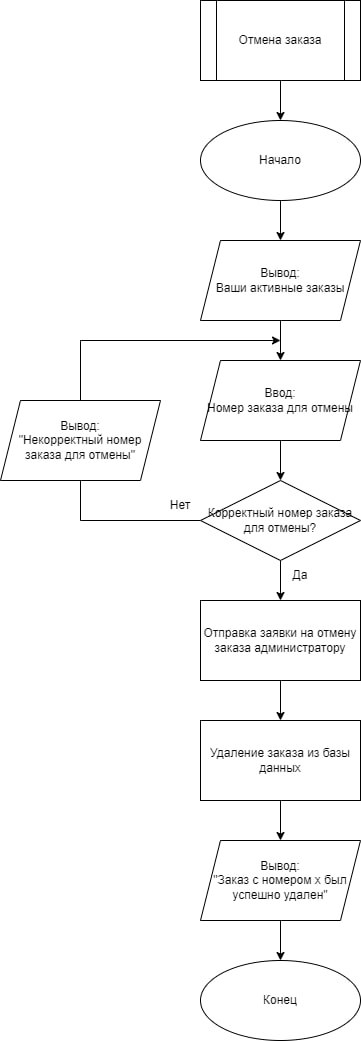


Рисунок 22 - Схема алгоритма «Отмена заказа»

Алгоритм: **Авторизация в админ-панель.**

Схема алгоритма «Авторизация в админ-панель» представлена на рисунке 23.

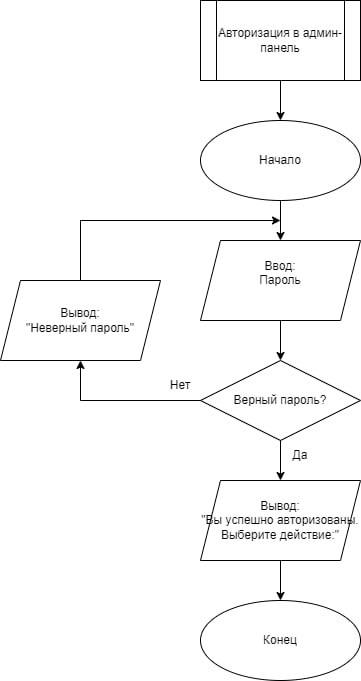


Рисунок 23 - Схема алгоритма «Завершение работы по заявке»

Так же для понимания как реагирует система был составлен пользовательский сценарий, который указан в таблице A.

Таблица A – Пользовательские сценарий

| **№ функции** | **Пользователь** | **Система** |
| --- | --- | --- |
| 1 | Регистрация | 1. Пользователь вводит команду для начала регистрации. 2. Вводит ФИО и дату выезда. 3. Заявка отправляется в админ-панель на обработку. |
| 2 | Просмотр меню | 1. Пользователь нажимает кнопку «Меню». 2. Система отображает доступное меню, разделенное на завтрак, обед и ужин. |
| 3 | Создание заказа | 1. Пользователь выбирает блюда и время подачи заказа. 2. Вводит номера выбранных блюд. 3. Заказ отправляется на обработку администратору. |
| 4 | Отмена заказа | 1. Пользователь нажимает кнопку «Отменить заказ». 2. Система отображает информацию о текущем заказе. 3. Пользователь вводит номер заказа для отмены. |
| 5 | Авторизация в админ-панель | 1. Ввод специальной команды и пароля для входа. 2. Система предоставляет доступ к функциям: просмотр заявок, заказов, пользователей и создание отчетов. |
| 6 | Одобрение заявки | 1. Администратор просматривает заявку с ID пользователя и его данными. 2. Нажимает кнопку «Одобрить». 3. Система назначает пользователю место за столом. |
| 7 | Просмотр заказов | 1. Администратор просматривает заказы с информацией о клиентах и блюдах. 2. Администратор может одобрить или отклонить заказ. |

# 7. Алгоритмы решения задач

## 7.1 Авторизация

Пользователь должен ввести ФИО и дату выезда. Если поля не заполнены или заполнены некорректно отображает сообщение об этом. Поля считаются заполненными некорректно если пользователь ввёл некорректную дату выезда. В случае корректного заполнения всех полей в базе данных в таблице «users» создаётся запись, содержащая введённые данные.

## 7.2 Одобрение заявки на авторизацию

После авторизации статус аккаунта пользователя меняется на «В ожидании». Администратор должен просмотреть заявки на авторизацию и одобрить нужную заявку. Дальше администратор выбирает стол и место пользователя, данная информация попадает в базу данных, в таблицу «tables», после этого статус аккаунта пользователя меняется на «Активный».

## 7.3 Создание заказа

Пользователь может просмотреть меню и начать создание заказа. Сначала пользователь должен выбрать время приёма пищи, после ввести номера блюд для заказа. Если поля не заполнены или заполнены некорректно отображает сообщение об этом. Поля считаются заполненными некорректно если пользователь ввёл несуществующие номера блюд для заказа. В случае корректного заполнения всех полей в базе данных в таблице «orders» создаётся запись, содержащая введённые данные.

## 7.4 Одобрение заявки на создание заказа

После создания заказа статус заказа пользователя меняется на «В ожидании». Администратор должен просмотреть заказы в ожидании и одобрить нужный заказ. После этого статус заказа пользователя меняется на «Активный».