**Аналитическая записка**

1. **Наименование проекта**

Название проекта: Telegram-бот для шведского стола.

1. **Целевая аудитория**

«Telegram-бот для шведского стола» предназначен для людей, работающих в сфере обслуживания, которые нуждаются в автоматизации некоторых процессов, а точнее для перевода с письменного вида заполнения бумаг о заказе блюда на онлайн-формат через Telegram-бота.

1. **Назначение проекта**

Назначением проекта «Telegram-бот для шведского стола» является Telegram-бот, который обеспечивает удобный и эффективный процесс заказа еды из определенного меню и управления бронированием столов для пользователей, а также административные функции для администратора.

1. **Описание взаимодействия с потенциальным пользователем**

Все пользовательские истории представлены в Таблице 1.

Таблица 1 – Пользовательские истории

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Роль** | **Действие** | **Цель** |
| 1 | Заказчик | Авторизоваться | Использование функционала бота для заказчика |
| 2 | Заказчик | Меню | Просмотр меню для заказа |
| 3 | Заказчик | Заказать | Заказ блюда из меню |
| 4 | Заказчик | Отменить заказ | Отмена актуального заказа |
| 5 | Администратор | «/admin» | Использование функционала бота для администратора |
| 6 | Администратор | Просмотр заявок | Просмотр заявок на авторизацию пользователей |
| 7 | Администратор | Просмотр заказов | Просмотр заказов блюд пользователей |
| 8 | Администратор | Заказы в ожидании | Просмотр неодобренных заказов пользователей |
| 9 | Администратор | Отмененные заказы | Просмотр отмененных заказов пользователей |
| 10 | Администратор | Пользователи | Просмотр созданных аккаунтов пользователями |
| 11 | Администратор | Отчет | Формирование отчета |

Все пользовательские сценарии представлены в Таблице 2.

Таблица 2 – Пользовательские сценарии

|  |  |  |
| --- | --- | --- |
| **№** | **Пользователь** | **Система** |
| 1 | Нажатие кнопки «Авторизоваться» | 1. Запрос ФИО пользователя  2. Запрос даты выезда пользователя  3. Отправка заявки на рассмотрение администратору |
| 2 | Нажатие кнопки «Меню» | 1. Вывод меню блюд для заказа |
| 3 | Нажатие кнопки «Заказать» | 1. Запрос времени приема пищи пользователя  2. Вывод меню блюд для заказа по времени приема пищи пользователя  3. Запрос номеров блюд для заказа пользователя |
| 4 | Нажатие кнопки «Отменить заказ» | 1. Вывод активных заказов пользователя  2. Запрос номера заказа пользователя для отмены  3. Запрос подтверждения отмены заказа пользователя |
| 5 | Ввод «/admin» | 1. Запрос пароля пользователя  2. Авторизация пользователя под администратором |
| 6 | Нажатие кнопки «Просмотр заявок» | 1. Вывод активных заявок пользователей на авторизацию  2. Вывод кнопок «Одобрить» и «Отклонить» |
| 7 | Нажатие кнопки «Одобрить» | 1. Вывод кнопок с свободными столами и местами для пользователя |
| 8 | Нажатие кнопки «Отклонить» | 1. Удаление заявки пользователя на авторизацию |
| 9 | Нажатие кнопки «Просмотр заказов» | 1. Вывод кнопок «Заказы в ожидании» и «Отмененные заказы» |
| 10 | Нажатие кнопки «Заказы в ожидании» | 1. Вывод активных заказов пользователей  2. Вывод кнопок «Одобрить» и «Отклонить» |
| 11 | Нажатие кнопки «Одобрить» | 1. Одобрение заказа пользователя |
| 12 | Нажатие кнопки «Отклонить» | 1. Отклонение заказа пользователя |
| 13 | Нажатие кнопки «Отмененные заказы» | 1. Вывод списка отклоненных заказов  2. Вывод кнопки «Одобрить» |
| 14 | Нажатие кнопки «Пользователи» | 1. Вывод списка авторизованных пользователей  2. Вывод кнопки «Удалить» |
| 15 | Нажатие кнопки «Удалить» | 1. Удаление пользователя из базы данных |
| 16 | Нажатие кнопки «Отчет» | 1. Формирование отчета  2. Вывод «.xls»-файла |
| 17 | Нажатие кнопки «Назад» | 1. Возвращение на шаг назад |

Диаграмма Use-Case представлена на Рисунке 1.

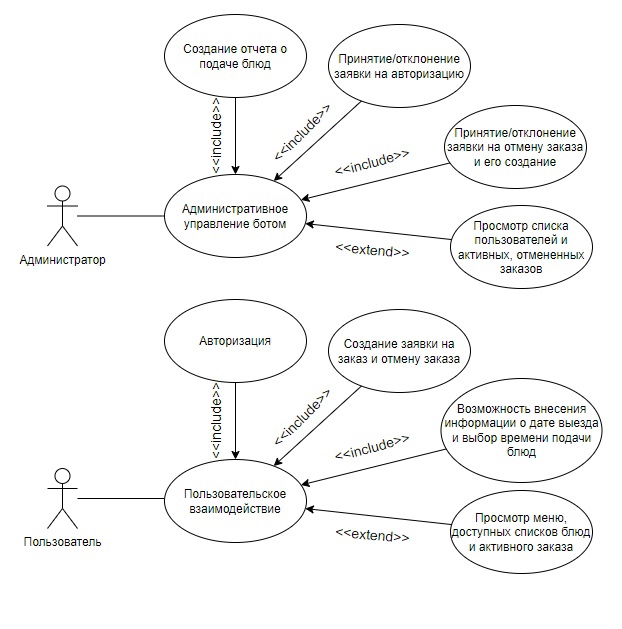


Рисунок 1 – Диаграмма Use-Case

Диаграмма Use-Case изображает систему Telegram-бота «Шведский стол» с двумя основными акторами: пользователь и администратор. Пользователь может авторизоваться и совершать заказ блюд и их отмену через создание заявки, также ему доступно внесение информации о дате выезда и просмотр информации своего заказа и списка блюд. Администратор отвечает за управление записями пользователей, принятие и отклонение их заявок на авторизацию и заказы, а также имеет возможность просматривать списки заказов с любыми состояниями (отмененные, активные), данные пользователей. Между действиями есть связи включения («include») и расширения («extend»), обозначающие, какие процессы включены или расширены в рамках других действий.

1. **Основной функционал**

Все функциональные требования представлены в таблице 3.

Таблица 3 – Функциональные требования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название функции** | **Входные параметры** | **Выходные параметры** | **Описание функции** |
| 1 | Инициализация главного меню | Команда «/start» | Главное меню пользователя | Эта функция проверяет статус пользователя и инициализирует главное меню при его первом взаимодействии с ботом. |
| 2 | Авторизация пользователя | ФИО, дата выезда | Заявка на регистрацию | Функция собирает данные пользователя для авторизации (ФИО и дата выезда) и отправляет их на обработку в систему. |
| 3 | Просмотр меню | Время приема пищи | Список блюд по времени | Пользователь выбирает время приема пищи (завтрак, обед, ужин), после чего функция выводит доступное меню для указанного времени. |
| 4 | Оформление заказа | Время приема пищи, номера блюд | Подтверждение заказа | Функция позволяет пользователю оформить заказ, выбрав блюда из меню, и сохраняет данные о заказе в системе. |
| 5 | Отмена заказа | Идентификатор заказа | Подтверждение отмены | Пользователь может просмотреть свои активные заказы и отменить любой из них в течение ограниченного времени. |
| 6 | Администратор: просмотр заявок | Команда «/admin» и пароль | Список заявок на регистрацию | Функция позволяет администратору просматривать заявки на авторизацию, а также одобрять или отклонять их. |
| 7 | Администратор: просмотр заказов | Команда администратор | Список заказов (активные, отмененные) | Администратор может просматривать активные и отмененные заказы пользователей, а также управлять ими. |
| 8 | Формирование отчета | Команда администратора | Отчет в формате «.xls» | Администратор может сгенерировать отчет о текущих пользователях, заказах и блюдах, который будет выгружен в формате Excel. |
| 9 | Удаление пользователя по дате выезда | Дата выезда пользователя | Удаление пользователя | Функция автоматически удаляет пользователя из системы по истечении даты выезда, вместе с его заказами. |

1. **Возможные аналоги, решения**

Существуют аналоги ИС «Telegram-бот для шведского стола», рассмотрим некоторые из них.

* 1. **Аналог 1. Физическое меню шведского стола – обыкновенный распечатанный лист**

Вид сервиса представлен на Рисунке 2.



Рисунок 2 – Физическое меню шведского стола

Достоинствами данного аналога является простота восприятия, удобен для людей, не обладающих навыками пользования смартфона.

* 1. **Аналог 2. «PANORAMA» – премиум-ресторан, занимающийся обслуживанием клиентов в сфере еды**

Интерфейс сервиса «PANORAMA» представлен на Рисунке 3.

****

Рисунок 3 – Интерфейс сервиса «PANORAMA»

Достоинством сервиса является бронирование столиков, простота восприятия интерфейса и возможность посмотреть меню на сайте.

* 1. **Аналог 3. «Сулугуни» - заведение, специализирующееся на приготовлении блюд кавказской кухни**

Интерфейс сервиса «Сулугуни» представлен на Рисунке 4.

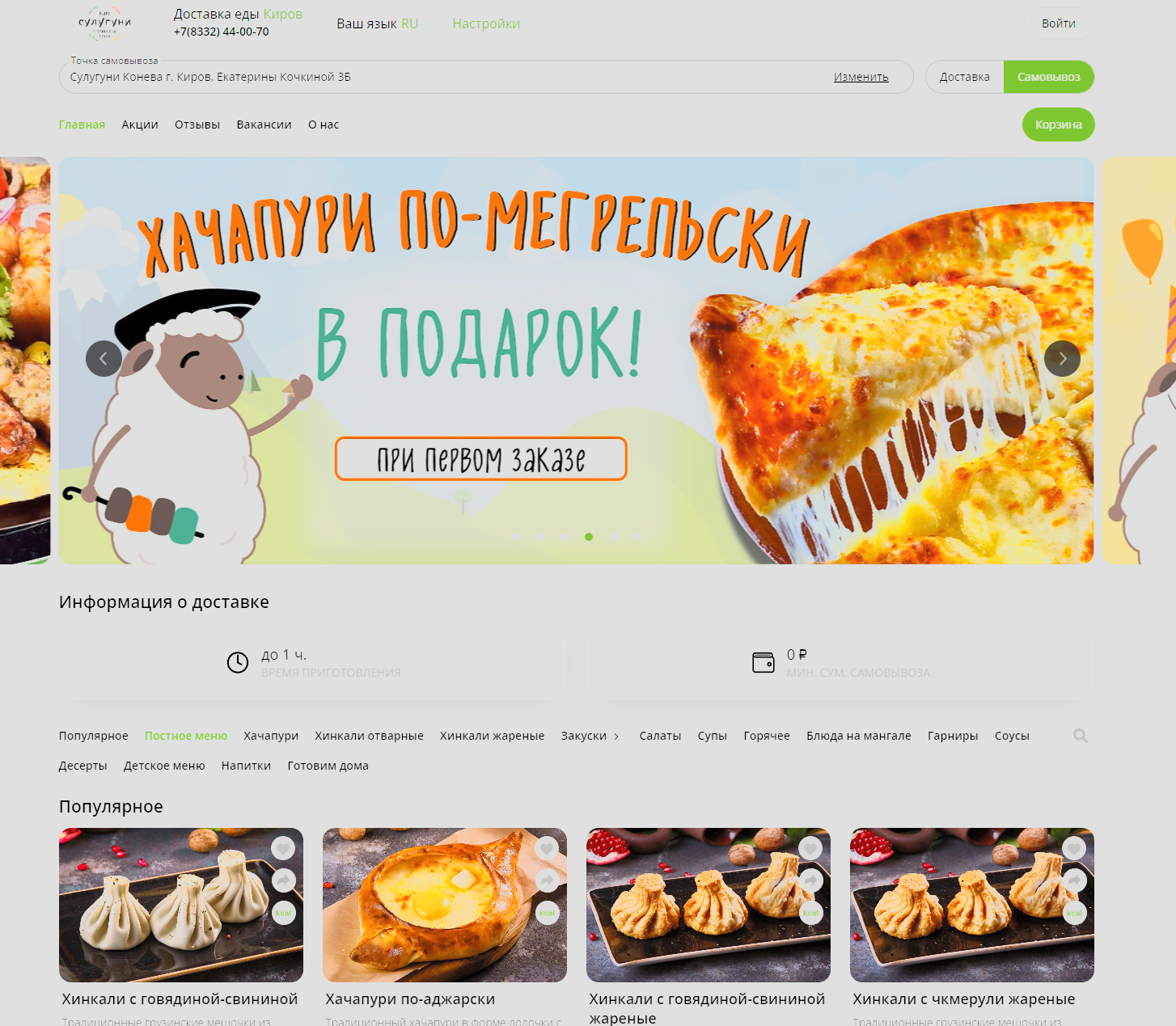


Рисунок 4 – Интерфейс сервиса «Сулугуни»

Достоинством сервиса является возможность дистанционного заказа для доставки или самовывоза.

Общими недостатками рассмотренных аналогов является то, что в большинстве примеров невозможно реализовать заказ дистанционно, а также присутствие рекламных баннеров на сайтах.

Проанализировав три аналога ИС «Telegram-бот для шведского стола», можно сделать вывод, что оно должно включать Telegram-бот, в котором будет присутствовать возможность сделать дистанционно заказ, например, на завтрак, обед или ужин. А также полное отсутствие рекламы.

1. **Предполагаемые к использованию технологии и модели**

Для реализации объекта используется следующий набор программных средств.

Python – универсальный язык, широко используется во всем мире для самых разных целей - базы данных и обработка текстов, встраивание интерпретатора в игры, программирование GUI и быстрое создание прототипов (RAD). Его синтаксис прост и читаем, что делает его отличным выбором для разработки как небольших, так и крупных проектов. Python обладает обширной экосистемой библиотек, что позволяет разработчикам эффективно решать разнообразные задачи.

PyCharm – это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для графического отладчика и работы с кодом. Она обладает богатым набором функций, включая подсветку синтаксиса, авто-дополнение кода, откладку, интеграцию с системами контроля версий и многое другое. PyCharm поддерживает множество языков программирования, включая Python.

PostgreSQL – мощная открытая реляционная система управления базами данных, которая широко применяется по всему миру для разнообразных задач. Она используется для хранения и обработки данных в больших корпоративных приложениях, веб-разработке, научных исследованиях и аналитике данных. PostgreSQL отличается высокой надежностью, масштабируемостью и поддержкой расширенных функциональных возможностей, включая геоинформационные и временные запросы, а также полнотекстовый поиск.

pgAdmin 4 – это кроссплатформенный графический инструмент для администрирования и управления серверами баз данных PostgreSQL. Разработан командой PostgreSQL и предоставляет разработчикам и администраторам полный набор инструментов для работы с базами данных. pgAdmin 4 предлагает удобный интерфейс для выполнения SQL-запросов, просмотра и редактирования данных, а также управления структурой базы данных.

В ходе практики были использованы несколько сторонних библиотек для более эффективной разработки. Например, telebot, psycopg2, datetime, xlwt, представляющие такие возможности, как создание и управление ботами в мессенджере Telegram, работа с PostgreSQL, датой и временем, файлами Excel в формате XLS.

Библиотека telebot в Python представляет собой популярную библиотеку на Python для создания и управления ботами в мессенджере Telegram. Она предоставляет разработчикам удобный и гибкий инструментарий для взаимодействия с Telegram Bot API, позволяя создавать ботов с разнообразным функционалом, от обработки текстовых сообщений до отправки медиафайлов и использования клавиатур для взаимодействия с пользователем. Благодаря гибкой архитектуре и расширяемым функциональным возможностям, telebot позволяет разработчикам создавать сложные и масштабируемые боты для различных задач, включая автоматизацию бизнес-процессов, предоставление информационных сервисов, управление заказами и т.д.

Библиотека psycopg2 в Python представляет собой одну из наиболее популярных библиотек на Python для работы с PostgreSQL. Она предоставляет разработчикам мощные и гибкие инструменты для взаимодействия с базами данных PostgreSQL, позволяя выполнять SQL-запросы, управлять транзакциями, обрабатывать ошибки и многое другое.

Библиотека datetime в Python предоставляет классы и функции для работы с датами и временем. Она является частью стандартной библиотеки Python и предоставляет удобные средства для работы с датами, временем, интервалами и таймзонами.

Библиотека xlwt в Python предназначена для создания файлов Excel в формате XLS. Она позволяет программно создавать и модифицировать таблицы Excel, добавляя в них различные типы данных, стили, форматирование и другие элементы. Она представляет собой мощный и гибкий инструмент для создания и управления файлами Excel на Python.