Пояснительная записка

Оглавление

[Введение 2](#_Toc155896423)

[Функциональная диаграмма продукта 3](#_Toc155896424)

[Проектирование 5](#_Toc155896425)

[Архитектурная диаграмма 5](#_Toc155896426)

[Диаграмма последовательности UML 5](#_Toc155896427)

[Концептуальная модель и физическая модель базы данных 7](#_Toc155896428)

[Требования к системному ПО 8](#_Toc155896429)

# Введение

На текущий момент существует множество различных мест, достопримечательностей, которые можно посетить в Санкт-Петербурге. Информацию о таких местах можно найти в сети Интернет с помощью браузера или с помощью онлайн-карт таких, как Яндекс.Карты.

При таком поиске пользователь тратит достаточно продолжительное время на выбор заведения, изучения информации о нем и т.д. Выборка мест и заведений будет происходит за счет наименования, а общая возможная заинтересованность пользователем в месте будет определяться отзывами о нем и его описанием, что также затрачивает время пользователя.

В качестве разрешения данной проблемы может выступать телеграмм бот, который будет выводить пользователю, основываясь на его предпочтениях и рекомендательной системе, структурированную информацию о местах и заведениях для посещения.

Целью работы является создание телеграмм бота, благодаря которому пользователи смогут получить информацию о местах и достопримечательностях города Санкт-Петербург на основе своих предпочтений.

Для выдачи информации пользователю необходимо воспользоваться следующими компонентами:

1. Тематики, предлагаемые телеграмм ботом и выбранные лично пользователем;
2. Рекомендательная система на основе схожих интересов между пользователями.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. спроектировать функциональную диаграмму продукта;
2. спроектировать и обосновать архитектуру продукта;
3. определить требования к системному ПО;

# Функциональная диаграмма продукта

Ниже представлена диаграмма функциональная диаграмма продукта в нотации IDEF0. Как видно из рисунка 1, есть несколько сущностей, влияющих на конечный проект:

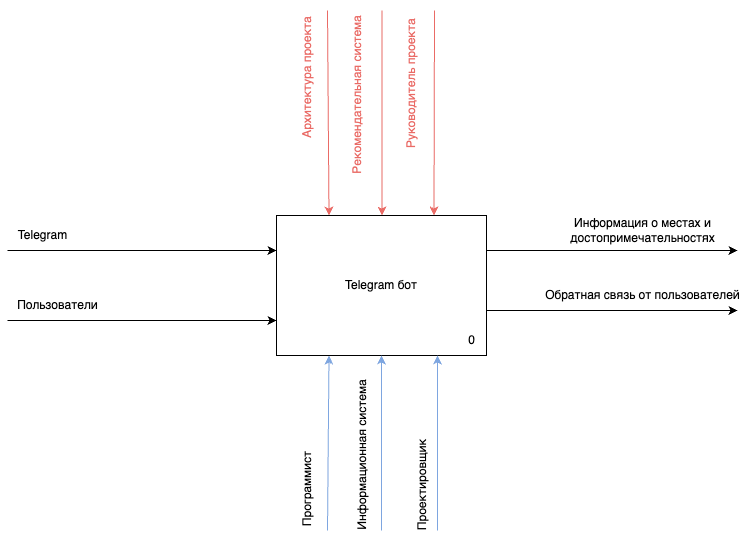
1. Входные данные определяются двумя сущностями: telegram и пользователи. Пользователи могут высылать информацию в telegram бот; telegram высылает различные сигналы в telegram бот.
2. Выходные данные: информация о местах и достопримечательностях, которая высылается пользователям, и обратная связь от пользователях, которая поставляется рекомендательной системе.
3. Контроль продукта составляют архитектура проекта, рекомендательная система и руководитель проекта.
4. ****За механизм продукта отвечают программист, информационная система и проектировщик.

Рисунок 1 – Функциональная диаграмма продукта

Ниже представлена функциональная диаграмма продукта, описывающая бизнес-процессы. Так существуют следующие бизнес-процессы:

1. Выбор пользователем категорий. В данном процессе подразумевается, что пользователь регистрируется в системе, выбирает интересующие его категории мест и достопримечательностей.
2. Процесс обработки рекомендательной системы. Рекомендательная система выбирает наиболее интересную для пользователя информацию, основываясь на собранной ранее информации о пользователях, их оценках о местах и достопримечательностях.
3. Публикация информации для пользователя. Как только рекомендательная система определила информацию о каком месте необходимо публиковать, telegram бот высылает информацию пользователю.
4. Сбор обратной связи. После того, как информация о месте выслана, telegram бот предоставляет возможность пользователю оценить информацию, место или достопримечательность. После информация об этом переходит в рекомендательную систему для использования в будущем.

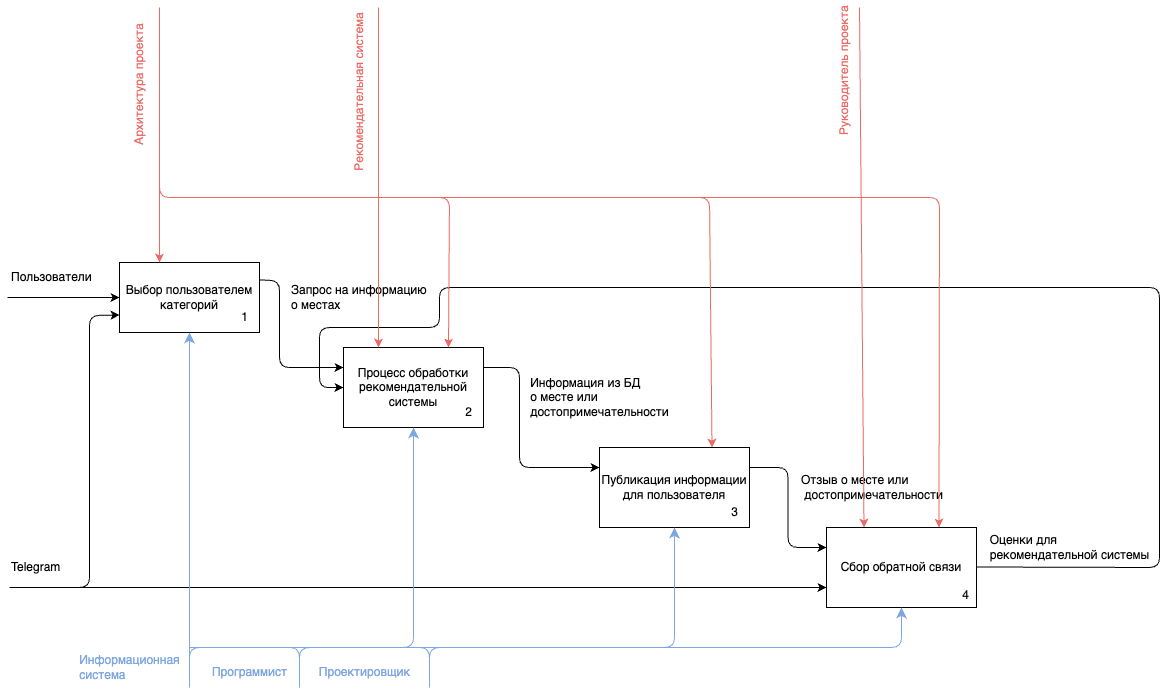
****

Рисунок 2 – Функциональная диаграмма продукта (бизнес-процессы)

# Проектирование

## Архитектурная диаграмма

На рисунке 3 ниже представлена архитектурная диаграмма продукта. На диаграмме присутствуют четыре сущности:

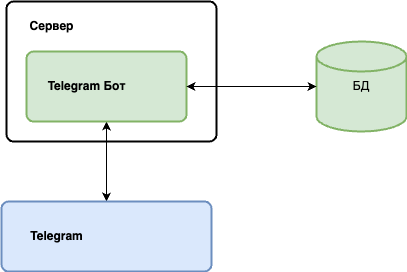
1. сервер, на котором запущен telegram бот;
2. telegram бот – скрипты, которые осуществляют всю необходимую работу для функционирования telegram бота;
3. база данных, которая хранит информацию о пользователях, рекомендательной системы и о местах и достопримечательностях;
4. telegram, к которому высылается вся информация от telegram бота.

Рисунок 3 – Архитектурная диаграмма

В случае реализации MVP данная архитектура является достаточной, если учитывать время и ресурсы, которыми обладает команда для реализации продукта.

В дальнейшем при анализе готового решения текущая архитектура позволяет производить масштабирование системы, а в каких-то системных изменений – команда не понесет значительных временных и иных ресурсов для внесения изменений.

## Диаграмма последовательности UML

Ниже представлена диаграмма последовательностей UML. Диаграмма отражает в другом виде функциональную диаграмму IDEF0. Данная диаграмма указывает дополнительные сведения о том, что в определенные процессы происходит обращение к базе данных (БД) и к рекомендательной системе.

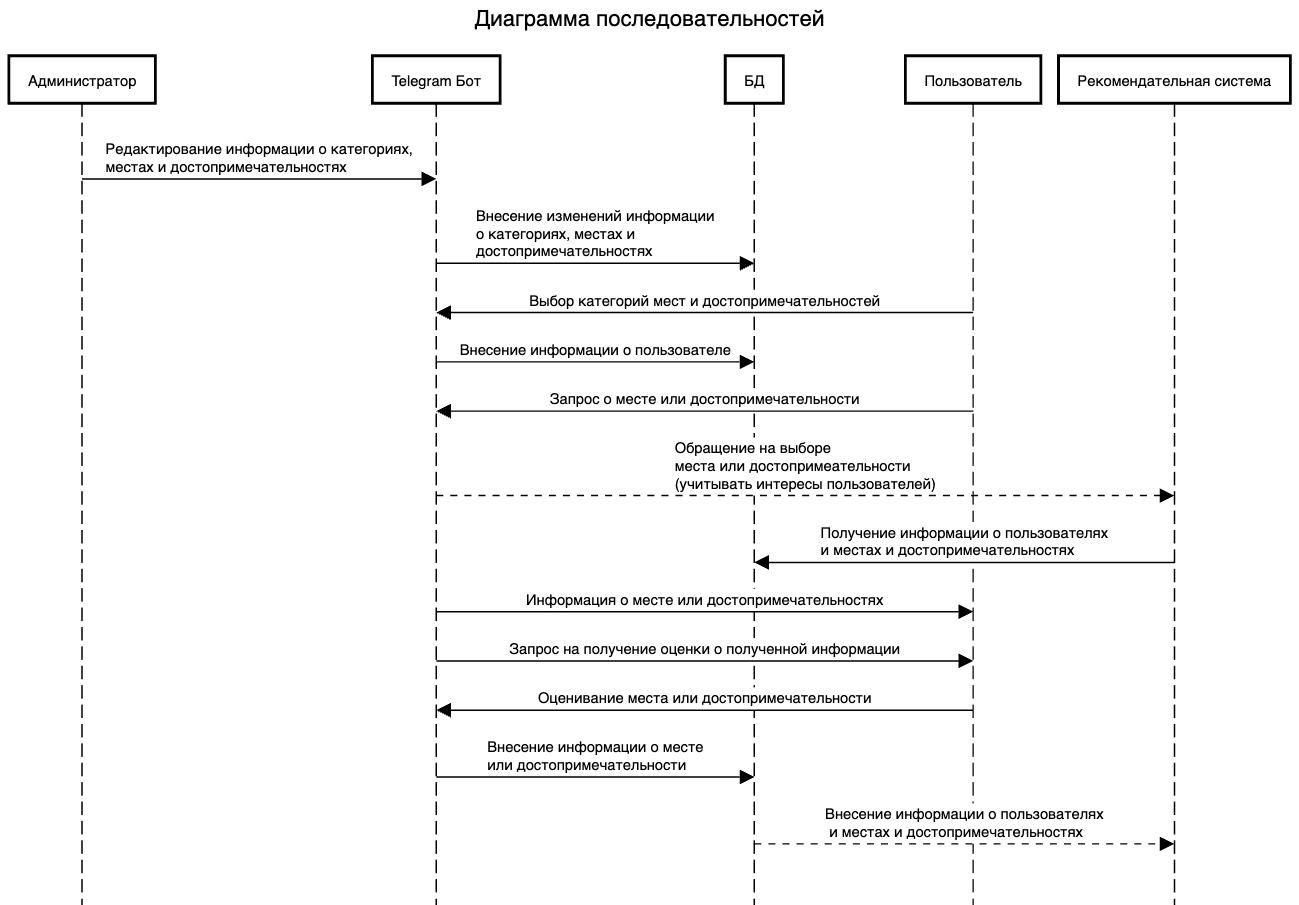


Рисунок – Диаграмма последовательностей UML

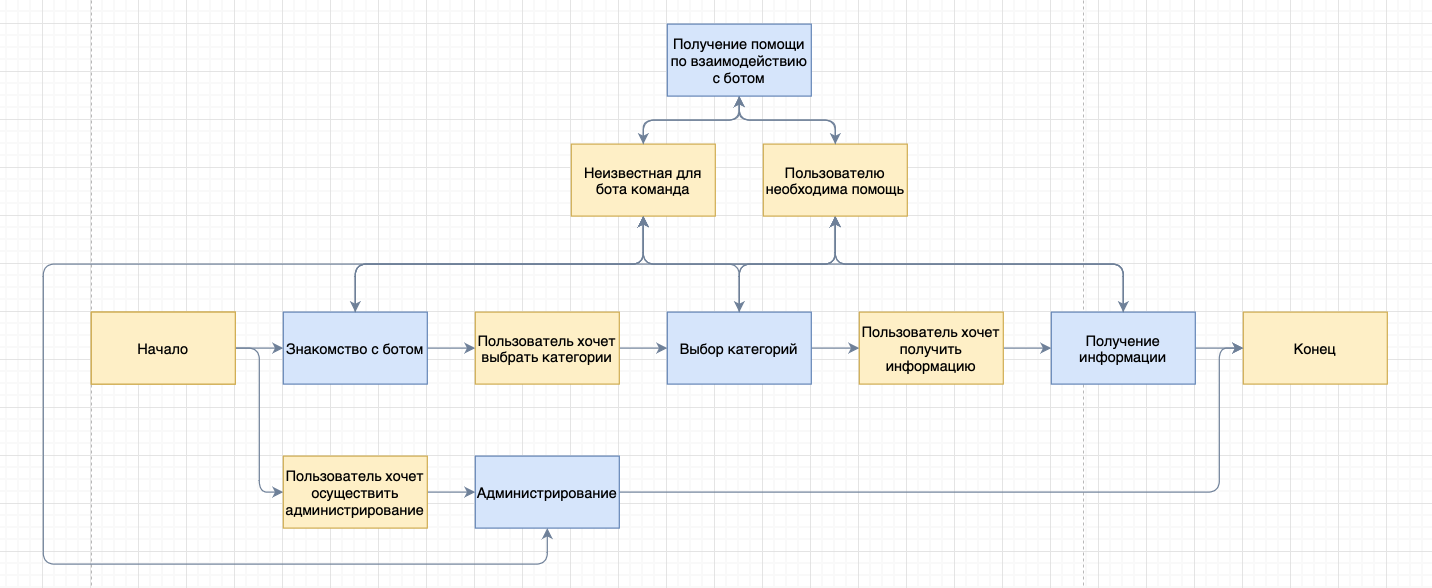
Дополнительно была реализована карта сценариев, которая отражает диалог между пользователем и telegram ботом. Диаграмма представлена рисунком 5 ниже:

Рисунок 5 – Карта сценариев

## Концептуальная модель и физическая модель базы данных

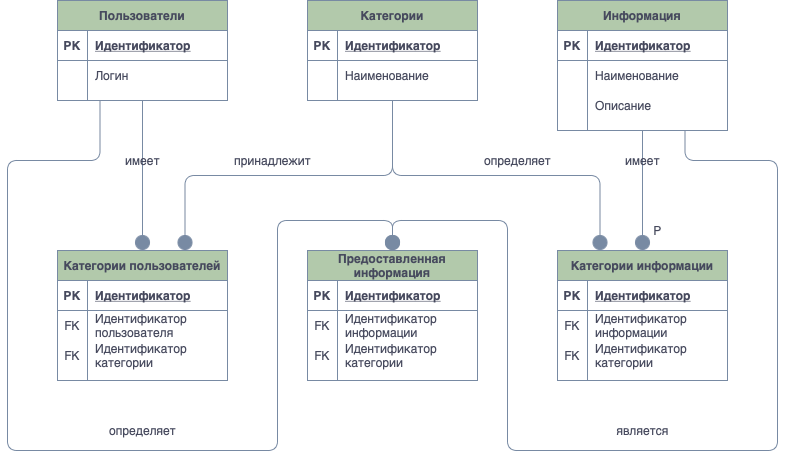
Концептуальная модель и физическая модель базы данных представлена в виде логической модели базы данных. На нижеуказанной модели (рисунок 6) можно увидеть сущности базы данных, которые будут реализованы, их поля, а и также отношения и их мощности.

Рисунок 6 – Логическая модель базы данных

# Требования к системному ПО

Продукт, телеграм-бот, располагается на сервере. Для работы сервера вводятся минимальные требования:

1. Операционная система – Ubuntu 16.04, Debian 10;
2. Процессор – i5-3370;
3. ОЗУ – 4 GB;
4. HHD или SSD объемом 20 GB.

Telegram бот написан с использованием:

1. Python 3.8;
2. Библиотек Python 3.8:
   1. telegram;
   2. sqlite3.