

# The Blooms Bridge software

Telegram-бот «Автоматизация процесса записи клиентов в компанию по Личному  
бренду»

## Документация

«Beta release **v.1.0.1**»

Выдана: \_\_\_\_\_

Салимли Айзек Мухтар Оглы

Принято: \_\_\_\_\_

Демидова Елена Николаевна

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Передача репозитория</b>	<b>5</b>
1.1 Идея хранения . . . . .	5
<b>2 СУБД</b>	<b>6</b>
<b>3 ER-диаграмма</b>	<b>7</b>
3.1 Чтение ER-диаграммы . . . . .	7
<b>4 Схема БД</b>	<b>8</b>
<b>5 Обеспечение постоянной работы бота</b>	<b>9</b>
5.1 Постоянный localhost . . . . .	9
5.2 Схема работы на локальном компьютере . . . . .	9
5.3 Сервер . . . . .	10
5.4 Схема развертки на сервере . . . . .	10
5.5 Хост . . . . .	11
5.6 Схема развертки на хосте . . . . .	11
<b>Контакты компаний предоставляющие место на сервере</b>	<b>12</b>
<b>Контакты компаний предоставляющие хосты</b>	<b>13</b>
<b>Контакты автора</b>	<b>14</b>

## Введение

В данной документации:

1. Текстом описаны процессы и подпроцессы работы бота
2. Графически описаны процессы и подпроцессы работы бота
3. Схема базы данных
4. ER-диаграмма
5. Дальнейшие действия по обеспечению стабильности работы телеграм-бота
6. Контакты

Краткая конфигурация:

- **Метод проектирования ПО:** Экстремальное программирование (XP)
- **Тестирование ПО:** Инспекция кода, метод белого ящика, живое тестирование
- **Система контроля версий:** Git
- **Репозиторий:** Есть, приватный
- **СУБД:** MySQL
- **Права на передачу прав в СУБД:** Салимли Айзек
- **Права на передачу репозитория:** Салимли Айзек
- **Хранение СУБД:** Ноутбук программиста
- **Хранение токена:** Ноутбук программиста
- **Проект:** Собран
- **Версия:** Beta v.1.0.1

## Краткое наименование и описание терминов:

1. Телеграм-бот (далее ТГ-Бот) - программное обеспечение интегрированное в приложение телеграма. Бот будет служить главной сущностью автоматизации процесса записи. Бот работает посредством TelegramAPI и идентификатора бота через TelegramAPI-token.
2. API - Application Programming Interface это "программный инструмент"набор правил, с помощью которого одно программное обеспечение может взаимодействовать с другим более масштабным программным обеспечением.
3. TelegramAPI - API предоставляемое самим телеграммом, в контексте реализуемого бота, для нас API - будет выступать библиотекой языка Python 3.13.
4. База данных (далее БД) - упорядоченное хранилище (обычно хранится на SSD/HDD), где хранятся необходимые данные. В базе данных используется специальные структуры данных которые помогают осуществлять быстрый поиск, быстрые запросы, быстрое получение данных из нее.
5. Система управления базами данных (далее СУБД) - Программное обеспечение управления базами данных с помощью языка манипуляции данными и языка определения данных. В случае данной реализации использовалась СУБД - MySQL.
6. Парсер - Программное обеспечение или модуль который считывает текст или другой поток данных и преобразует его в желаемую форму или формат текста. В реализации используется собственный парсер написанный на языке программирования Java.
7. ER-диаграмма (далее ER) - Графическое представление отношений сущностей.
8. Схема базы данных (далее схема БД) - Графическое представление реляционных отношений в реляционных баз данных (SQL).
9. Сервер - это компьютер или программа, которая обрабатывает запросы от других устройств (клиентов) в сети и предоставляет им необходимые данные или услуги.
10. Администратор базы данных (далее администратор БД) - человек, отвечающий за стабильность, масштабирование, структуру базы данных.
11. Репозиторий - это место (физическое или виртуальное), где хранятся и управляются файлы, обычно с помощью систем контроля версий.
12. Система контроля версий git (далее Git) - это система контроля версий, которая позволяет сохранять изменения в файлах, отслеживать историю этих изменений и совместно работать над проектами.
13. GitHub - это онлайн-платформа, основанная на Git, предназначенная для хранения проектов, совместной работы над кодом и обмена результатами.
14. Язык программирования Python 3.13 (далее Python) - интерпретируемый (и компилируемый) язык программирования высокого уровня, с большим выбором библиотек.
15. Язык программирования Java 20.1 (далее Java) - компилируемый в байт-код, кроссплатформенный высокоуровневый язык программирования. Компилирует байт-код в свою виртуальную машину JVM. Что позволяет языку быть платформо-независимым.
16. Сборка проекта - Сбор файлов проекта, файлов моделирования для более удобной работы с приложением.
17. Conda build (далее Conda) - Инструмент сборки проекта для языка Python.
18. Maven project (далее mvn) - Инструмент сборки проекта для языка Java.

# 1 Передача репозитория

На данный момент репозиторий находится в приватном доступе. В нем скрыты все ключи и токены и скрипт создания БД.

Ссылка на репозитории автора: [github.com MathematicLove](https://github.com/MathematicLove)

## 1.1 Идея хранения

Весь рабочий код данной версии находится полностью под контролем программиста, на ветке main. На рис.1 показана иерархия системы контроля Git.

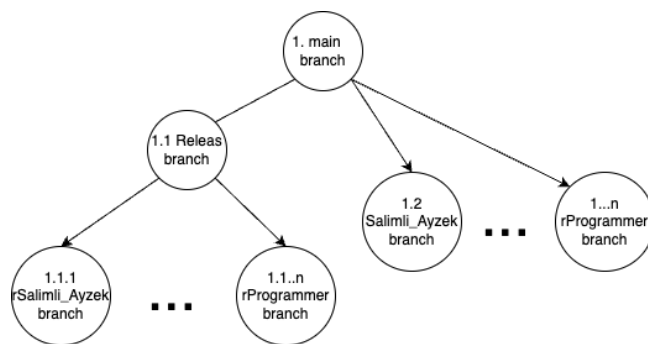


Рис. 1: Иерархия контроля версий.

1. Ветка main: главная ветка, где находится само приложение (еще не выпущенное в эксплуатацию, например новые доработки). В ветку main отправляют общий код который пишут/ редактируют программисты. В ветке main, код контролирует главный программист.
2. Ветка release: ветка в которой хранится код программы готовый к эксплуатации.
3. Ветка 1.1.1: ветка на которой работает (вносит правки или дополнения в версию готовую к эксплуатации) - Салимли Айзек.
4. Ветки 1.1... n: Ветки других возможных программистов, которые вносят правки/дополнения в версию готовую к эксплуатации.
5. Ветка 1.2: Ветка на которой работает программист который вносит правки/дополнения в версию **не** готовой к эксплуатации (например новая версия бота).
6. Ветка 1... n: Ветки других возможных программистов, которые вносят правки/дополнения в версию **не** готовой к эксплуатации.

Если программист/программисты готовы к выпуску продукции, они отправляют все свои доработки в ветку release. После чего, весь код тестируется заказчиком в живую. При согласии, версия публикуется на host/сервер. При передаче репозитория, главный программист (Салимли А.) отдает весь приватный репозиторий заказчику (Демидова Е. Н.). После чего создает свою отдельную ветку, где будут вноситься доработки/правки.

## 2 СУБД

Для хранения данных клиентов, будет использоваться СУБД - MySQL. Таблицы в которых будут храниться данные:

- Client - Таблица клиентов. Хранит: ID, Фамилию клиента, Имя клиента, Отчество клиента (если есть), номер телефона клиента.
- Appointment - Таблица записей. Хранит: ID, ID-клиента, дату записи, время записи, полное ФИО клиента.
- AppointmentStatus - Таблица статусов записей. Хранит: ID, ID-записи, статус записи, номер клиента, полное ФИО клиента.
- Logs - Таблица ошибок. Хранит: ID, дату ошибки, текст ошибки.

В СУБД поступают следующие шифрованные данные:

1. ФИО
2. Номер телефона

Для надежности используется тройное шифрование алгоритмами криптографии: AES, DES3, ChaCha20. При получении данных данные расшифровываются в обратном порядке.

### 3 ER-диаграмма

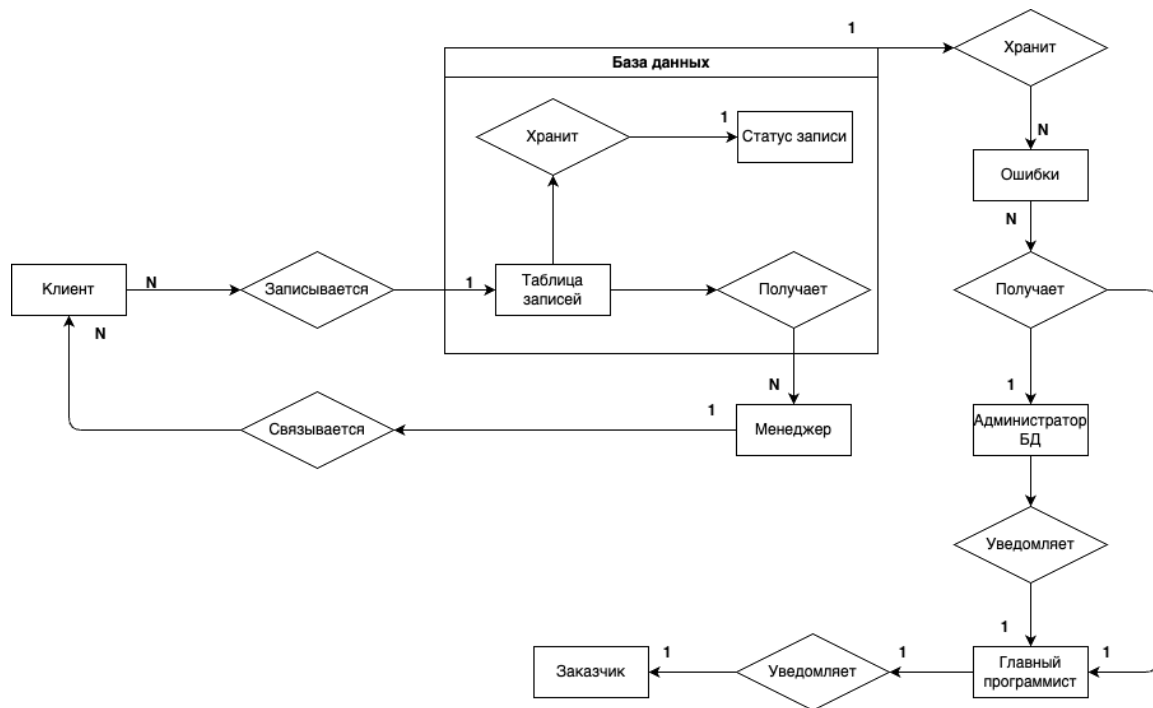


Рис. 2: ER-диаграмма.

#### 3.1 Чтение ER-диаграммы

Клиенты записываются на прием в компанию по Личному бренду, запись записывается в таблицу записей (хранится в БД) которая хранит статус этой записи, которая так же хранится в БД. Менеджер получив запись может связаться с клиентом, если клиент проигнорирует уведомление-вопрос от бота (придет ли клиент вовремя на запись). База данных хранит в себе таблицу ошибок. Эти ошибки получает администратор БД, либо если ошибка связана с логикой программы/запросов, ошибку получает программист. Администратор БД может уведомить главного программиста об ошибках. Главный программист уведомляет об ошибке заказчика, далее работает над ее исправлением.

## 4 Схема БД

На Рис.3 изображена схема БД.

Название базы данных: Booking system

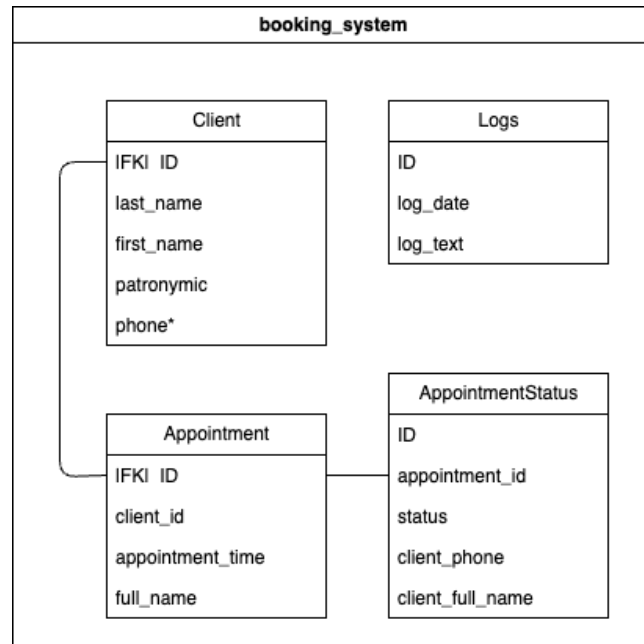


Рис. 3: Схема базы данных.



## 5 Обеспечение постоянной работы бота

### 5.1 Постоянный localhost

Это способ при котором реализуемый телеграм-бот локально на компьютере, и запускается в среде разработки или через командную строку. Компьютер при этом должен быть включен постоянной основе, и не будет отключать запущенную программу (код бота), а так же компьютер будет подключен к интернету. То есть телеграм-бот будет доступен пользователям только когда включен компьютер и запущен код бота.

Плюсы:

1. Бесплатно
2. Удобно для тестирования
3. Легкое внедрение правок/дополнений

Минусы:

1. Работа компьютера 24/7
2. Если компьютер будет отключен от интернета - бот отключится
3. Случайное закрытие среды разработки/командной строки приведет к отключению бота

### 5.2 Схема работы на локальном компьютере

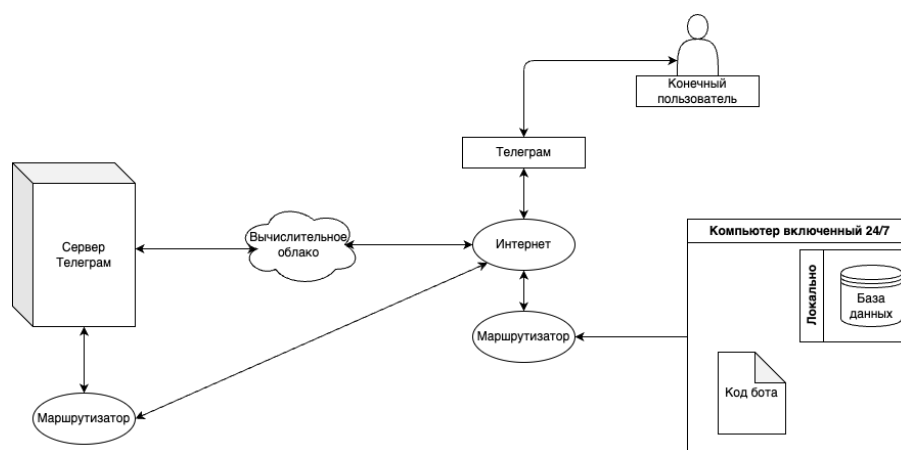


Рис. 4: Схема работы на локальном компьютере.

### 5.3 Сервер

Этот способ гарантирует постоянную работу на удаленном сервере которым владеет компания по обеспечению облачных вычислений. Например: Google Cloud, Yandex Cloud, GoogleSDK. Нужно арендовать виртуальный сервер, например VPS. Далее заходим на сервер удаленно, загружаем туда код бота и запускаем. Сервер постоянно включен и подключен к интернету. При сбое сервера ответственность берет компания предоставляющая услугу.

Плюсы:

1. Надежность - сервер постоянно включен, подключен к интернету и есть резервное копирование
2. Можно выбрать разные мощности для работы (в случае бота, не требуется мощный сервер)

Минусы:

1. Платно (относительно не дорого)
2. При внесении правок, нужно разбираться с удаленным сервером

### 5.4 Схема развертки на сервере

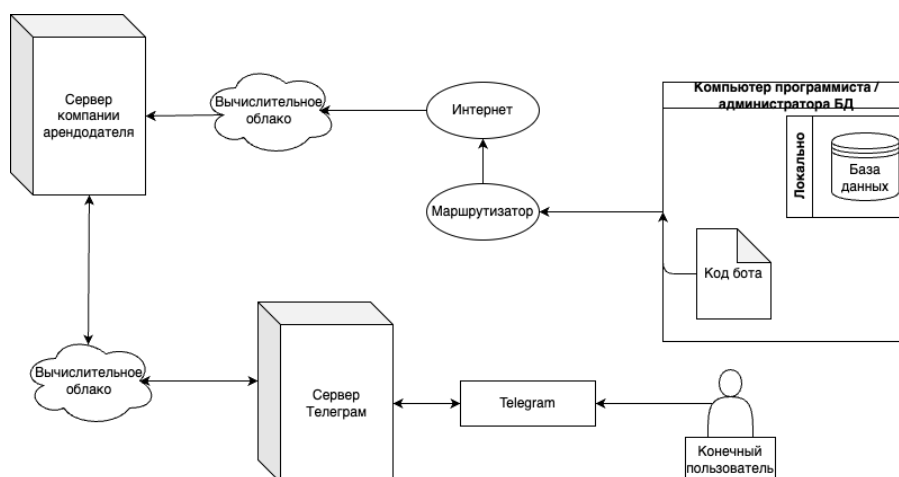


Рис. 5: Схема развертки на сервер.

## 5.5 Хост

Здесь бот запускается на специальных платформах, которые предназначены именно для размещения приложений (ботов, сайтов). Нужно зарегистрироваться на платформе, например Render. Загрузить код бота (например через GitHub). Платформа автоматически запустит бота.

Плюсы:

1. Легко настроить
2. Минимум усилий программиста

Минусы:

1. Платно
2. Время отклика больше чем от сервера

## 5.6 Схема развертки на хосте

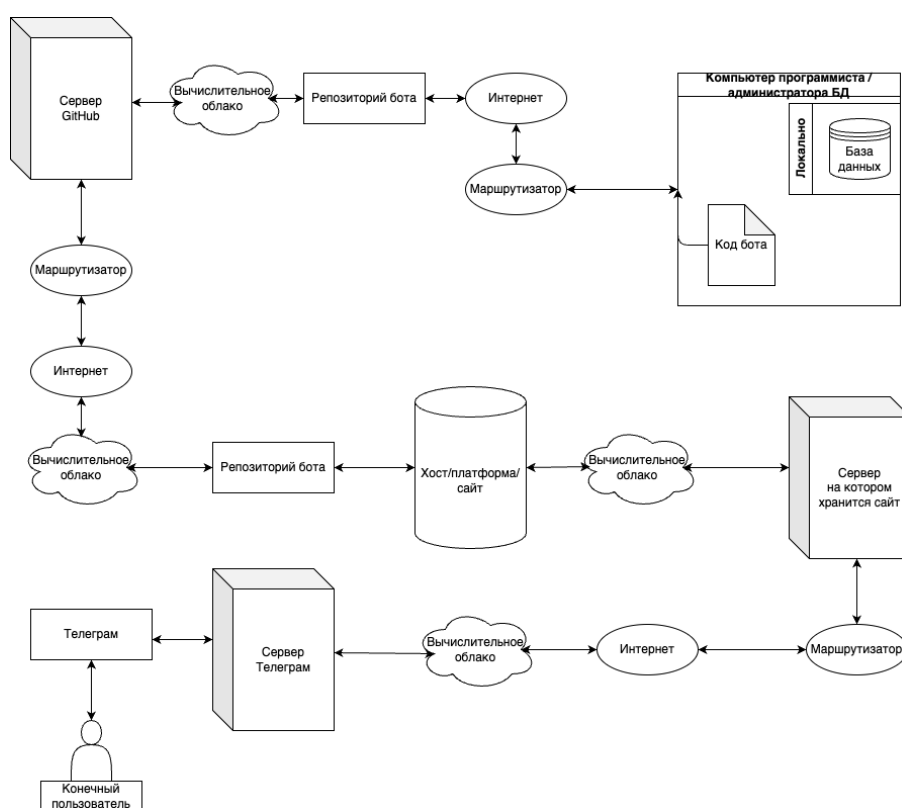


Рис. 6: Схема развертки на хосте.

## Контакты компаний предоставляющие место на сервере

- Timeweb: <https://timeweb.com>  
Цена: от 169 руб/мес. (VPS)  
Характеристики: от 1 до 16 ГБ RAM, SSD NVMe от 15 до 160 ГБ, интернет 200 Мбит/с
- Sprinthost: <https://sprinthost.ru>  
Цена: от 144 руб/мес. (виртуальный хостинг)  
Характеристики: от 5 до 25 ГБ SSD, неограниченный трафик
- Beget: <https://beget.com>  
Цена: от 199 руб/мес. (виртуальный хостинг)  
Характеристики: от 10 ГБ SSD, неограниченные базы данных, ежедневное резервное копирование
- Fornex: <https://fornex.com>  
Цена: от 199 руб/мес. (VPS-Start)  
Характеристики: 1 CPU, 1 ГБ RAM, 30 ГБ SSD, 100 Мбит/с
- SmartApe: <https://smartape.ru>  
Цена: от 1.61 USD/мес. (VPS)  
Характеристики: 1 CPU, 512 МБ RAM, 5 ГБ SSD, 100 Мбит/с
- AdminVPS: <https://adminvps.ru>  
Цена: от 320 руб/мес. (VPS)  
Характеристики: от 1 CPU, от 1 ГБ RAM, SSD от 20 ГБ, 100–200 Мбит/с
- McHost: <https://mchost.ru>  
Цена: от 144 руб/мес. (виртуальный хостинг)  
Характеристики: SSD 5–50 ГБ, безлимитный трафик, ежедневные бэкапы
- RuVDS: <https://ruvds.com>  
Цена: от 250 руб/мес. (VPS)  
Характеристики: 1 CPU, 512 МБ RAM, 10 ГБ SSD, 100 Мбит/с
- EuroByte: <https://eurobyte.ru>  
Цена: от 199 руб/мес. (VPS-Start)  
Характеристики: 1 CPU, 1 ГБ RAM, 30 ГБ SSD, 100 Мбит/с
- HandyHost: <https://handyhost.ru>  
Цена: от 200 руб/мес. (виртуальный хостинг)  
Характеристики: SSD от 5 ГБ, неограниченный трафик, ежедневное резервное копирование

## Контакты компаний предоставляющие хосты

- Heroku: <https://www.heroku.com>

**Цена:** Платные тарифы начинаются от 7\$ в месяц.

**Характеристики:** Стандартный план включает 512 МБ RAM и 1 веб-дино. Более высокие тарифы предлагают больше ресурсов и возможностей.

- Render: <https://www.render.com>

**Цена:** Платные тарифы от 7\$ в месяц.

**Характеристики:** 512 МБ RAM, 0.5 CPU для бесплатного тарифа. Платные тарифы предлагают больше ресурсов и возможностей.

- Railway: <https://www.railway.app>

**Цена:** Платные тарифы от 10\$ в месяц.

**Характеристики:** 1 vCPU, 1 ГБ RAM для начальных тарифов. Более высокие тарифы предлагают больше ресурсов.

- Glitch: <https://www.glitch.com>

**Цена:** Платный тариф ("Glitch Pro") за 8\$ в месяц.

**Характеристики:** 512 МБ RAM, 200 МБ дискового пространства для бесплатного тарифа. Платный тариф предоставляет дополнительные возможности.

- Replit: <https://www.replit.com>

**Цена:** Платные тарифы от 7\$ в месяц.

**Характеристики:** 0.2-0.5 vCPU, 0.5-1 ГБ RAM для бесплатного тарифа. Платные тарифы предлагают больше ресурсов и возможностей.

- Amvera Cloud: <https://amvera.ru/>

**Цена:** Бесплатный тариф отсутствует. Платные тарифы от 200 рублей в месяц.

**Характеристики:** 1 vCPU, 512 МБ RAM, 10 ГБ SSD для базового тарифа. Более высокие тарифы предлагают больше ресурсов.

## Контакты автора

- Телефон: +7 (921) 945-67-03
- Электронная почта: [thisisnauchno@gmail.com](mailto:thisisnauchno@gmail.com)
- Корпоративная почта: [ayzek@thebloomsbridge.com](mailto:ayzek@thebloomsbridge.com)
- Telegram: @undefined\_1010
- WhatsApp: +7 (921) 945-67-03
- Сайт компании, предоставляющей услуги ПО: [thebloomsbridge.io](http://thebloomsbridge.io)
- Сайт с репозиториями автора: [github.com/MathematicLove](https://github.com/MathematicLove)

Салимли Айзек Мухтар оглы

The Blooms Bridge Software, Machine Learning Engineer.