СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc170377611)

[1 ХАРАКТЕРИСТИКА ОБЪЕКТА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ 4](#_Toc170377612)

[1.1 Анализ предметной области 5](#_Toc170377613)

[1.2 Программная и техническая архитектура ИС предприятия 5](#_Toc170377614)

[1.3 Разработка и анализ требований к программному модулю 9](#_Toc170377615)

[1.4 Выработка требований к программному обеспечению и программному модулю 9](#_Toc170377616)

[2 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 12](#_Toc170377617)

[2.1 Проектирование программного продукта 12](#_Toc170377618)

[2.2 Функциональная схема программного продукта 14](#_Toc170377619)

[2.3 Проектирование пользовательского интерфейса 15](#_Toc170377620)

[2.4 Кодирование программного обеспечения 16](#_Toc170377621)

[2.5 Организация обработки исключений. Отладка модулей программного проекта 18](#_Toc170377622)

[3 ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ 20](#_Toc170377623)

[3.1 Тестирование программного обеспечения 20](#_Toc170377624)

[3.1.1 Выбор и обоснование методики проведения тестирования 20](#_Toc170377625)

[3.1.2 Схема выполнения экспериментального тестирования 20](#_Toc170377626)

[3.2 Инспектирование компонент программного продукта на предмет соответствия стандартам кодирования 21](#_Toc170377627)

[3.3 Проведение функционального тестирования готового программного продукта 24](#_Toc170377628)

[3.4 Документирование результатов тестирования 25](#_Toc170377629)

[ЗАКЛЮЧЕНИЕ 27](#_Toc170377630)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 28](#_Toc170377631)

[ПРИЛОЖЕНИЯ 1 – Листинг кода 29](#_Toc170377632)

# ВВЕДЕНИЕ

В рамках практической подготовки в виде производственной практики необходимо разработать Телеграм – Бота для поддержки консультантов магазинов. Основной целью этого проекта является создание инструмента, который поможет консультантам быстро находить необходимые товары и повысит эффективность их работы.

Задачи проекта включают в себя разработку функционала ПМ (производственного модуля) бота, который будет обеспечивать консультантов актуальной информацией о наличии товаров в магазине. Кроме того, необходимо предусмотреть поддержку и тестирование ПМ для обеспечения его стабильной работы и соответствия требованиям пользователей.

Разработка Телеграм – Бота для консультантов магазинов значительно улучшит процесс консультирования клиентов, сократит время на поиск товаров и повысит уровень обслуживания. Этот инструмент станет надежным помощником для персонала магазинов, способствуя оптимизации и повышению эффективности работы консультантов, что в свою очередь приведет к улучшению сервиса для клиентов.

# ХАРАКТЕРИСТИКА ОБЪЕКТА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

ООО НПФ «Форус» является одной из ведущих ИТ-компаний России, активно развивающей рынок информационных технологий с 1992 года. Компания имеет головной офис в г. Иркутске и представительства в 16 регионах России, а также в Монголии (Улан-Баторе). В компании работает более 400 специалистов, из которых более 300 сертифицированы в области «1С».

Основным видом деятельности компании является разработка компьютерного программного обеспечения. У компании зарегистрировано 25 видов деятельности по ОКВЭД. Она имеет связи с двумя компаниями и четырьмя совладельцами. Генеральным директором является Денисова Валентина Сергеевна, а размер уставного капитала составляет 10 000₽.

Компания принимала участие в 99 тендерах, а в отношении неё было возбуждено 3 исполнительных производства и участвовала в 14 арбитражных делах.

Направления деятельности компании включают:

* Создание централизованных облачных информационных систем управления финансово-хозяйственной деятельностью органов исполнительной власти и бюджетных учреждений.
* Цифровизация производственных и торговых предприятий и внедрение ERP-систем.
* Реализация международных проектов в различных странах.
* Дистрибьюция программного обеспечения «1С» и других вендоров.
* Обеспечение информационной безопасности.
* Разработка и внедрение машинного обучения, искусственного интеллекта и других современных технологий в лаборатории R&D.

## Анализ предметной области

Компания "Форус" специализируется на предоставлении консультационных услуг в области управления бизнесом и информационных технологий. Они работают с клиентами различных отраслей, помогая им оптимизировать бизнес-процессы, внедрять новые технологии и повышать эффективность работы.

Одной из ключевых проблем, с которой сталкиваются их консультанты, является необходимость оперативного взаимодействия с клиентами для предоставления консультаций, ответов на вопросы и решения возникших проблем.

Использование бота в Мессенджере Телеграм, интегрированного с базой данных в системе 1С УНФ, позволит консультантам "Форус" быстро получать необходимую информацию и оперативно отвечать на запросы клиентов. Это также обеспечит удобный канал для сбора обратной связи от клиентов, что поможет компании улучшить качество предоставляемых услуг.

Таким образом, внедрение данного бота позволит "Форус" оптимизировать процессы обслуживания клиентов, улучшить качество консультаций и повысить уровень удовлетворенности клиентов, что в свою очередь способствует росту бизнеса компании.

## Программная и техническая архитектура ИС предприятия

Организация ООО НПФ «Форус» использует разнообразные операционные системы, включая Windows и Linux, для управления компьютерами и серверами. Для хранения данных и их управления они применяют различные системы управления базами данных (СУБД), такие как Microsoft SQL Server, PostgreSQL, Oracle и MySQL. Компания также использует программное обеспечение от Microsoft, включая Microsoft Outlook для электронной почты, Битрикс24 для управления задачами и как корпоративный мессенджер.

Важным решением является выбор программного обеспечения 1С, которое компания использует для эффективного учета финансовых операций, управления банковскими операциями, расчетов по заработной плате и налогам, а также для планирования проектов и учета контактов с клиентами.

Для создания веб-сайтов компания применяет языки программирования HTML, JavaScript и CSS. Фреймворки, такие как Angular, React и Vue.js, упрощают процесс разработки веб-приложений. CMS, такие как WordPress, Drupal и Joomla, предоставляют удобные средства для управления контентом на сайтах.

Разработчики используют инструменты типа Git для управления версиями кода, а также специализированные программы, такие как IDE (Visual Studio, IntelliJ IDEA, Eclipse, PyCharm) для написания и отладки кода. Для работы с базами данных они применяют различные СУБД, такие как MySQL, PostgreSQL, Oracle и Microsoft SQL Server, с соответствующими клиентами для удобства работы. Графические редакторы типа Adobe Photoshop или Sketch используются для создания графики и дизайна веб-сайтов.

ООО НПФ «Форус» оснащает свои офисы компьютерной техникой, включая моноблоки, ноутбуки и стационарные компьютеры. Они предпочитают моноблоки от HP, но также используют моноблоки фирмы Acer. Процессоры, установленные в их компьютерах, варьируются от Intel Pentium CPU G640 2.80GHz до Core i5-1235U и Core i3-1215U. ОЗУ составляет от 4 ГБ до 32 ГБ. Графические карты в основном интегрированы в процессор (Intel Graphics), но у ноутбуков также имеются дискретные видеокарты Nvidia GTX 1050, Nvidia GTX 1650, Nvidia RTX 2060. В качестве носителей внешней памяти используются SSD емкостью от 100ГБ до 250ГБ для операционной системы и HDD емкостью от 500ГБ до 1ТБ для хранения информации. Стационарные компьютеры оснащены процессорами Intel Core i5-7000, 16ГБ оперативной памяти в двухканальном режиме, видеокартами AMD Radeon RX 5000 и Nvidia GTX 1050ti, а также только HDD емкостью 500ГБ. Мониторы поставляются от компаний Acer, Asus и LG. Компания использует принтеры, включая многофункциональные устройства (МФУ) и лазерные принтеры (всего 7 штук) для печати и ксерокопирования документов. Также имеется струйный принтер для печати на фотобумаге и цветной бумаге. Для мероприятий, обучения и собраний используются проекторы, установленные в каждой переговорной комнате.

Программная архитектура представлена на Рисунке 1, а техническая архитектура представлена на Рисунке 2.

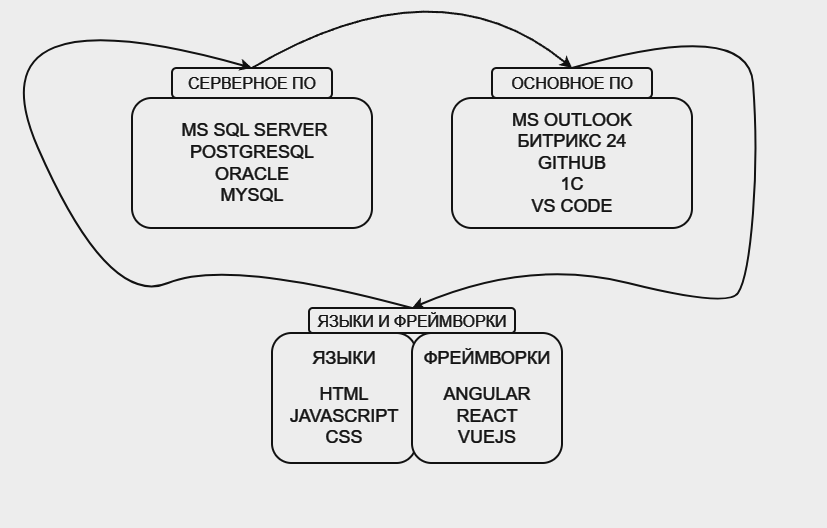


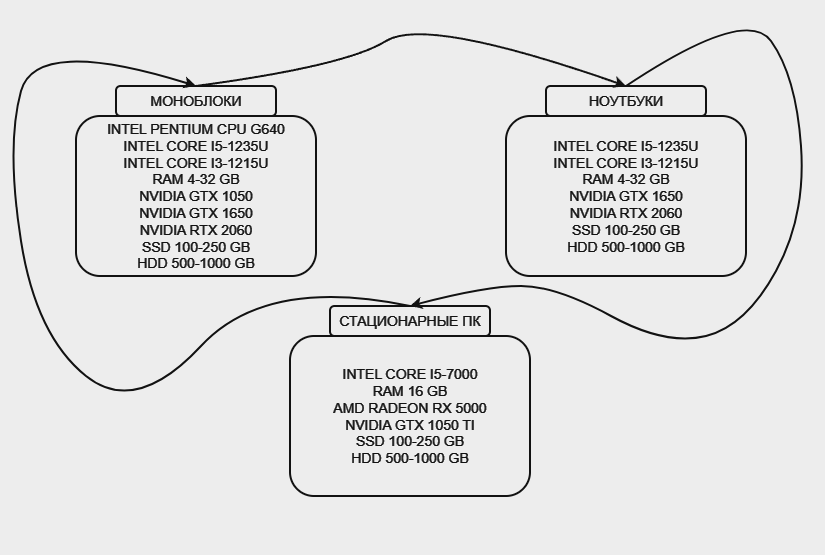
Рисунок 1 – Программная архитектура

Рисунок 2 – Техническая архитектура

## Разработка и анализ требований к программному модулю

Бот должен быть интегрирован с Мессенджером Телеграм и базой данных в системе 1С УНФ.

Задачи внедрения бота включают создание "единого окна" управления взаимодействиями с консультантами, получение обратной связи от них, а также рассмотрение потенциальных выгод для консультантов, магазина или фирмы, получение конкурентных преимуществ, увеличение количества каналов информирования клиентов, а также повышение качества обслуживания и лояльности клиентов.

Бот должен выполнять следующие бизнес-процессы: обработку запросов на поиск товаров, предоставление информации о статусе товаров, а также информацию о доступности специальных и дополнительных услуг.

## Выработка требований к программному обеспечению и программному модулю

Технические характеристики системы бота представлены в таблице 1.

Таблица 1 – Технические характеристики системы бота.

| Характеристика | Значение |
| --- | --- |
| Количество сообщений | Не менее 2000 |
| Время реакции бота на обращение пользователя, секунд | Не более 3 |
| Распределенная нагрузка, при которой сохраняется заявленная производительность, диалогов/сутки | Не менее 2000 |
| Количество пользователей | Не менее 50 |

Требования к системе.

Бот должен содержать следующие компоненты:

* Удобный интерфейс. Конечный вид бота, который должен быть прост в использовании и иметь приятный внешний вид;
* Модуль обработки запросов. Этот модуль отвечает за прием запросов от консультанта через интерфейс мессенджера;
* Логика обработки запросов. Этот компонент определяет логику обработки запросов от консультанта и формирования ответов. Например, он может определять какие данные нужно получить из базы данных и как их представить консультанту;
* Модуль взаимодействия с базой данных 1С. Этот модуль отвечает за отправку запросов к базе данных 1С через API и получение необходимой информации.

Требования к надежности.

Бот должен функционировать без отказов – не менее 99,8%. Время простоя бота без учета проведения регламентных работ не должно превышать 17,5 часов в год.

Общие требования к надежности бота:

* сохранение всех функций в случае выхода из строя любого отдельного компонента;
* обеспечение сохранности данных при наступлении аварийной ситуации;
* сохранение работоспособности и обеспечение автоматического восстановления функций при возникновении нештатных ситуаций;
* использование механизмов обеспечения целостности операционных данных в условиях параллельного доступа на чтение и запись.

Требования к программному обеспечению.

К клиентскому программному обеспечению предъявляются следующие требования:

* бот должен управляться пользователями через кроссплатформенное веб-приложение;
* клиентская часть веб-интерфейса должна поддерживать работу со следующими веб-браузерами на десктопах:
* Chrome версии 90 и выше;
* Safari версии 14 и выше;
* Mozilla Firefox версии 68 и выше;
* Opera версии 66 и выше;
* Yandex браузер версии 20 и выше.
* на мобильных устройствах:
* Android 9 и выше в веб-браузере Chrome или в мобильном приложении;
* iOS 13 и выше в веб-браузерах Safari или в мобильном приложении;

Вывод по главе: была представлена информация о структуре организации, программной и технической архитектуре предприятия, а также проведен анализ сферы деятельности, на основе которого были определены цели, задачи и особенности проекта.

# РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

## Проектирование программного продукта

Диаграммы деятельности используются для описания динамических аспектов поведения системы на различных уровнях абстракции, включая классы, интерфейсы, компоненты и узлы.

При разработке программного обеспечения планирование и моделирование помогают упростить программирование. Для концептуального моделирования функциональности применяется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

На Рисунке 3 представлен пример диаграммы вариантов использования системы Телеграм – Бота.

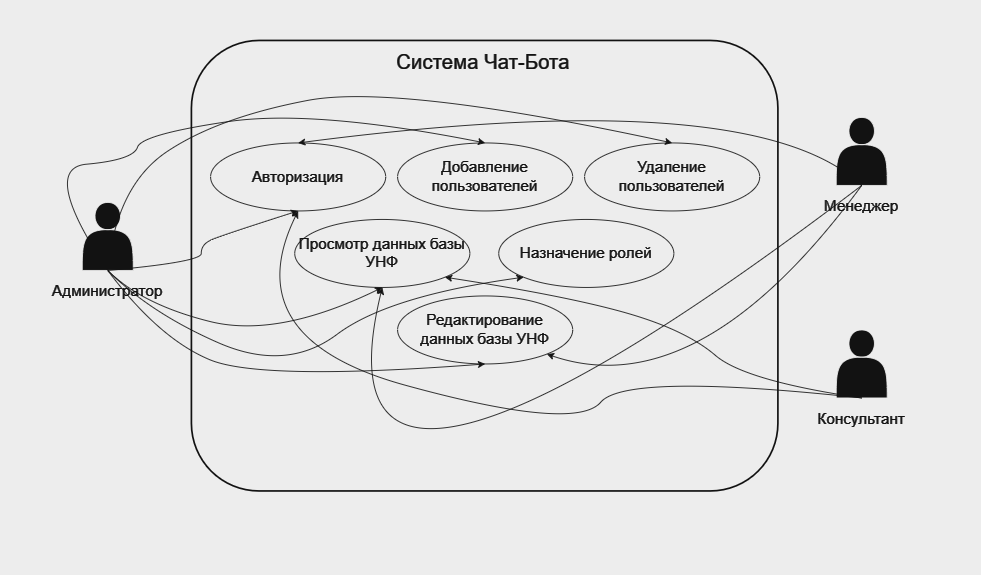


Рисунок 3 – Диаграмма вариантов использования

На Рисунке 4 представлен пример диаграммы деятельности системы Телеграм – Бота.

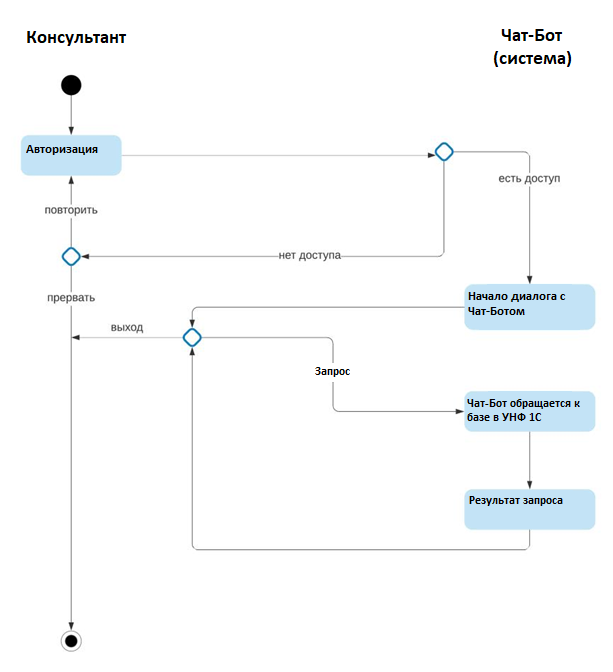


Рисунок 4 – Диаграмма детальности

## Функциональная схема программного продукта

Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Настоящий стандарт распространяется на условные обозначения (символы) в схемах алгоритмов, программ, данных и систем и устанавливает правила выполнения схем, используемых для отображения различных видов задач обработки данных и средств их решения. Стандарт не распространяется на форму записей и обозначений, помещаемых внутри символов или рядом с ними и служащих для уточнения выполняемых ими функций.

Ниже представлена функциональная схема данных системы Телеграм – Бота на Рисунке 5.

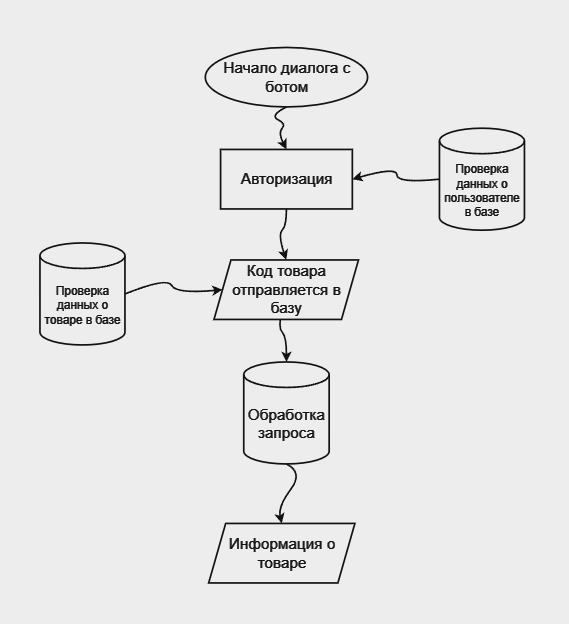
****

Рисунок 5 – Схема данных Телеграм – Бота

## Проектирование пользовательского интерфейса

Бот разрабатывался в такой IDE как Visual Studio Code. Где бот, и его библиотека данных, писалась и хранилась на коде JavaScript.

Visual Studio Code – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

База, откуда Чат-Бот брал все необходимые данные, хранилась и разрабатывалась в такой система как 1С УНФ.

1С – российская компания, основанная в 1991 году и занимающаяся разработкой, изданием и поддержкой компьютерных программ, баз данных делового и домашнего назначения, а также компьютерных игр. Организационно-правовая форма: общество с ограниченной ответственностью.

Основной продукт – программная система «1С: Предприятие», изначально созданная как расширяемая бухгалтерская система с собственным встроенным языком программирования, в дальнейшем охватившая многие функции продуктов классов ERP, CRM, HRM, SCM.

Ниже представлена таблица 2, где описаны команды и их назначения Телеграм – Бота.

Таблица 2 – Команды и их назначения.

| Объекты | Свойство | Значение |
| --- | --- | --- |
| Команда /menu | Общие команды бота | Появление возможных команд |
| Команда /code | Отправка товара по артикулу | Информация по товару и его цене |
| Команда /remainder | Excel файл с остатками на складе | Таблица товаров по остаткам на складах |

На Рисунке 6 изображен перечень кнопок для взаимодействия с ботом в чате.



Рисунок 6 – Схема кнопок для взаимодействия с ботом

## Кодирование программного обеспечения

Программный код команд /menu, /code и /remainder (написанные на JS) представлены в таблице 3.

Таблица 3 – Программный код команд /menu, /code и /remainder.

| Объекты | Событийно-управляемые процедуры |
| --- | --- |
| /menu | bot.onText(/\/menu/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Список команд: \n/code - поиск товара по артиклу\n/name - поиск товара по наименованию\n/remainder - отчёт товаров по остаткам на складах');  }); |
| /code | bot.onText(/\/code/, async (msg) => {    const chatId = msg.chat.id;    const http = require('http');    const options = {      host: 'localhost',      path: '/bot/hs/telegram/test',      headers: {        'Authorization': 'Basic 0JDQtNC80LjQvdC40YHRgtGA0LDRgtC+0YA6'      }    };    http.get(options, (res) => {      let data = '';      res.setEncoding('utf8');      res.on('data', (chunk) => {        data += chunk;      });      res.on('end', () => {        try {          const jsonData = JSON.parse(data);          if (jsonData.text) {            bot.sendMessage(chatId, jsonData.text);          } else {            bot.sendMessage(chatId, "Неверный формат данных");          }        } catch (error) {          bot.sendMessage(chatId, "Ошибка обработки данных");        }      });    }).on('error', (err) => {      console.log("Error: " + err.message);    });  }); |
| /remainder | bot.onText(/\/remainder/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Вот Excel файл с остатками на складе: ');  }); |

Далее идет программный код со стороны 1С для связки библиотеки NodeJS, на которой и написан сам бот, а также поиск кода по товару в базе.

Программный код публикации HTTP-Сервиса представлен в таблице 4.

Таблица 4 – Программный код HTTP-Сервиса.

| Объекты | Событийно-управляемые процедуры |
| --- | --- |
| Функция testGET | Функция testGET(Запрос)     Товар = Справочники.Номенклатура.НайтиПоРеквизиту("Артикул",Артикул);   Если НЕ ЗначениеЗаполнено(Товар) Тогда            СтруктураОтвета = Новый Структура("text", "Не работает");       //КодТовара = Запрос.ПараметрыURL["CustomerCode"];  //Товар = ПолучитьНазваниеТовара(КодТовара);   Возврат ВернутьУспехИОтвет(СтруктураОтвета);     Иначе //3. заполним структуру данных о клиенте            ДанныеТовара = Новый Структура;       ДанныеТовара.Вставить("Code", Товар.Код);            ДанныеТовара.Вставить("Name", Товар.Наименование);       ДанныеТовара.Вставить("Artikul", Товар.Артикул);          // ДанныеКлиента.Вставить("Email", Товар.Вес);     //4. сериализуем данные о клиенте в JSON            ЗаписьJSON = Новый ЗаписьJSON;       ЗаписьJSON.УстановитьСтроку();            ЗаписатьJSON(ЗаписьJSON, ДанныеТовара);       СтрокаДляОтвета = ЗаписьJSON.Закрыть();            //5. сформируем ответ       Ответ = Новый HTTPСервисОтвет(200);            Ответ.Заголовки.Вставить("Content-type", "application/JSON; charset=utf-8");       Ответ.УстановитьТелоИзСтроки(СтрокаДляОтвета, КодировкаТекста.UTF8, ИспользованиеByteOrderMark.НеИспользовать);   Возврат Ответ; КонецЕсли; КонецФункции |
| Функция ВернутьУспехИОтвет | Функция ВернутьУспехИОтвет(СтруктураОтвет) Экспорт  Ответ = Новый HTTPСервисОтвет(200);  СтрокаJSON = ПолучитьСтрокуJSON(СтруктураОтвет);  Ответ.УстановитьТелоИзСтроки(СтрокаJSON);  Ответ.Заголовки.Вставить("Content-type", "application/json");  Возврат Ответ; КонецФункции |
| Функция ПолучитьСтрокуJSON | Функция ПолучитьСтрокуJSON(Значение) Экспорт  ЗаписьJSON = Новый ЗаписьJSON;  ЗаписьJSON.УстановитьСтроку();  ЗаписатьJSON(ЗаписьJSON, Значение);  Возврат ЗаписьJSON.Закрыть();  КонецФункции |

## Организация обработки исключений. Отладка модулей программного проекта

Была проведена отладка кода в Visual Studio Code, в результате которой была выявлена только одна незначительная ошибка, которая была сразу же ликвидирована.

На Рисунке 7 представлена отладка модулей программного проекта.

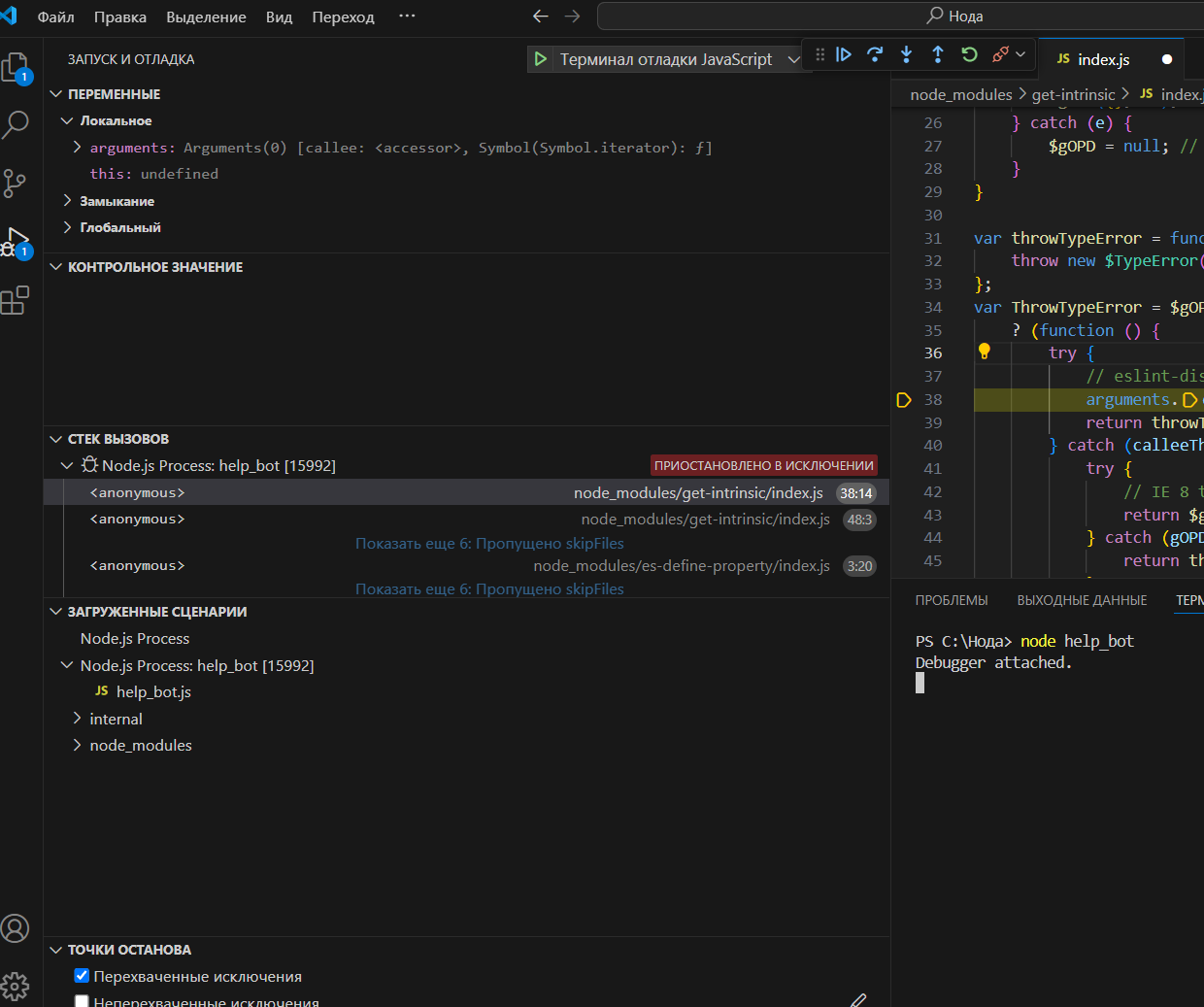


Рисунок 7 – Отладка модулей программного проекта

Вывод по главе: было завершено проектирование программного продукта, создания диаграмм использования, подробной структуры данных для системы Телеграм – Бота, разработки пользовательского интерфейса, написания кода программного обеспечения, а также реализации обработки исключений и отладки модулей программного проекта, можно утверждать, что разработка программных модулей завершена в полном объеме.

# ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

## Тестирование программного обеспечения

Тестирование программного продукта — это процесс проверки и оценки программного обеспечения с целью выявления дефектов, ошибок и недостатков, а также убеждения в соответствии продукта требованиям и ожиданиям пользователей.

Цели тестирования программного продукта:

* Выявление дефектов
* Уверенность в качестве
* Оценка функциональности
* Проверка безопасности

Задачи тестирования программного продукта:

* Выполнение тестов
* Выявление и отслеживание дефектов
* Анализ результатов тестирования

### Выбор и обоснование методики проведения тестирования

Были выбраны три методики проведения тестирования:

* Тестирование чёрным ящиком;
* Экспериментальное тестирование;
* Функциональное тестирование.

Данные методики были выбраны для реализованного Чат-Бота, так как лучше всего проверяют работоспособность и надёжность продукта.

### Схема выполнения экспериментального тестирования

Прежде чем начинать экспериментальное тестирование необходимо собрать группу пользователей, которые будут тестировать данного Чат-Бота. Количество привлеченных пользователей должно быть не менее пяти человек.

Пользователям будет представлен список некоторых вопросов, относящихся к графическому интерфейсу.

В таблице 5 представлены сведения о пользователях.

Таблица 5 – Информация о пользователях

| ФИО | Пол | Возраст | Образование |
| --- | --- | --- | --- |
| Попов Андрей Евгеньевич | Мужской | 24 | Среднее специальное |
| Троцкий Юрий Александрович | Мужской | 24 | Среднее специальное |
| Козлов Владислав Владимирович | Мужской | 30 | Среднее специальное |
| Шелковникова Анастасия Анатольевна | Женский | 35 | Среднее специальное |
| Комаров Александр Иванович | Мужской | 34 | Среднее специальное |

## Инспектирование компонент программного продукта на предмет соответствия стандартам кодирования

Таблица 6 – Инспектирование компонент программного продукта на предмет соответствия стандартам кодирования

| № | Исходный код | Код в соответствии со стандартами языка программирования |
| --- | --- | --- |
| 1 | Функция testGET(Запрос)  СтруктураОтвет = Новый Структура("text", "Работает");  Возврат ВернутьУспехИОтвет(СтруктураОтвет);  КонецФункции  Функция ВернутьУспехИОтвет(СтруктураОтвет) Экспорт  Ответ = Новый HTTPСервисОтвет(200);  СтрокаJSON = ПолучитьСтрокуJSON(СтруктураОтвет);  Ответ.УстановитьТелоИзСтроки(СтрокаJSON);  Ответ.Заголовки.Вставить("Content-type", "application/json");  Возврат Ответ;  КонецФункции  Функция ПолучитьСтрокуJSON(Значение) Экспорт  ЗаписьJSON = Новый ЗаписьJSON;  ЗаписьJSON.УстановитьСтроку();  ЗаписатьJSON(ЗаписьJSON, Значение);    Возврат ЗаписьJSON.Закрыть();  КонецФункции | Функция testGET(Запрос)  СтруктураОтвет = Новый Структура("text", "Работает");  Возврат ВернутьУспехИОтвет(СтруктураОтвет);  КонецФункции  Функция codeGET(Запрос)  ПолучитьНазваниеТовара(КодТовара);  // Поиск товара в базе данных по его коду  Товар = Справочники.Номенклатура.НайтиПоКоду(КодТовара);  // Получение названия товара  Название = Товар.Наименование;  // Возврат названия товара  Возврат Название;  КонецФункции  Функция ВернутьУспехИОтвет(СтруктураОтвет) Экспорт  Ответ = Новый HTTPСервисОтвет(200);  СтрокаJSON = ПолучитьСтрокуJSON(СтруктураОтвет);  Ответ.УстановитьТелоИзСтроки(СтрокаJSON);  Ответ.Заголовки.Вставить("Content-type", "application/json");  Возврат Ответ;  КонецФункции  Функция ПолучитьСтрокуJSON(Значение) Экспорт  ЗаписьJSON = Новый ЗаписьJSON;  ЗаписьJSON.УстановитьСтроку();  ЗаписатьJSON(ЗаписьJSON, Значение);    Возврат ЗаписьJSON.Закрыть();  КонецФункции |
| 2 | const TelegramBot = require('node-telegram-bot-api');  const token = '7050755395:AAHCzc0HOZo42GsqHLv\_FFJtwZhcUlAWmk8';  const bot = new TelegramBot(token, {polling: true});  console.log('Bot has been started ...')  bot.onText(/\/start/, async (msg) =>  {    console.log(msg);    const chatId = msg.chat.id;    await bot.sendMessage(chatId, `Приветствую, ${msg.from.first\_name}! Узнать, что я умею можно по команде /menu`);  });  bot.onText(/\/menu/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Список команд: \n/code - поиск товара по артиклу\n/name - поиск товара по наименованию\n/remainder - отчёт товаров по остаткам на складах');  });  bot.onText(/\/code/, async (msg) => {    const chatId = msg.chat.id;    const http = require('http');    const options = {      host: 'localhost',      path: '/bot/hs/telegram/test',      headers: {        'Authorization': 'Basic 0JDQtNC80LjQvdC40YHRgtGA0LDRgtC+0YA6'      }    };    http.get(options, (res) => {      let data = '';      res.setEncoding('utf8');      res.on('data', (chunk) => {        data += chunk;      });      res.on('end', () => {        try {          const jsonData = JSON.parse(data);          if (jsonData.text) {            bot.sendMessage(chatId, jsonData.text);          } else {            bot.sendMessage(chatId, "Неверный формат данных");          }        } catch (error) {          bot.sendMessage(chatId, "Ошибка обработки данных");        }      });    }).on('error', (err) => {      console.log("Error: " + err.message);    });  }); | const TelegramBot = require('node-telegram-bot-api');  const token = '7050755395:AAHCzc0HOZo42GsqHLv\_FFJtwZhcUlAWmk8';  const bot = new TelegramBot(token, {polling: true});  console.log('Bot has been started ...')  bot.onText(/\/start/, async (msg) =>  {    console.log(msg);    const chatId = msg.chat.id;    await bot.sendMessage(chatId, `Приветствую, ${msg.from.first\_name}! Узнать, что я умею можно по команде /menu`);  });  bot.onText(/\/menu/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Список команд: \n/code - поиск товара по артиклу\n/name - поиск товара по наименованию\n/remainder - отчёт товаров по остаткам на складах');  });  bot.onText(/\/code/, async (msg) => {    const chatId = msg.chat.id;    const http = require('http');    const options = {      host: 'localhost',      path: '/bot/hs/telegram/test',      headers: {        'Authorization': 'Basic 0JDQtNC80LjQvdC40YHRgtGA0LDRgtC+0YA6'      }    };    http.get(options, (res) => {      let data = '';      res.setEncoding('utf8');      res.on('data', (chunk) => {        data += chunk;      });      res.on('end', () => {        try {          const jsonData = JSON.parse(data);          if (jsonData.text) {            bot.sendMessage(chatId, jsonData.text);          } else {            bot.sendMessage(chatId, "Неверный формат данных");          }        } catch (error) {          bot.sendMessage(chatId, "Ошибка обработки данных");        }      });    }).on('error', (err) => {      console.log("Error: " + err.message);    });  });  bot.onText(/\/name/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Скинь мне наименование товара');  });  bot.onText(/\/remainder/, async (msg) =>  {    const chatId = msg.chat.id;    await bot.sendMessage(chatId, 'Вот Excel файл с остатками на складе: ');  }); |

## Проведение функционального тестирования готового программного продукта

Было проведено несколько вариантов функционального тестирования Телеграм – Бота в таблице 7.

Таблица 7 – Результаты функционального тестирования.

| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| --- | --- | --- | --- | --- | --- |
| 1 | Получение ответа от Чат-бота из УНФ, находящейся на 1С | /code | Сообщение «Скинь мне артикул товара» | Сообщение «Скинь мне артикул товара» | Да |
| 2 | Получение информации о товаре по его артикулу | Ввести артикул  Пример (000001) | Название товара: Диван Dandy 2.0 Артикул: 000001 | Название товара: Диван Dandy 2.0 Артикул: 000001 | Да |

## Документирование результатов тестирования

В результате проведения тестирования «чёрным ящиком» не были выявлены ошибки, мешающие работе бота. Пример работы бота представлен на Рисунке 8.

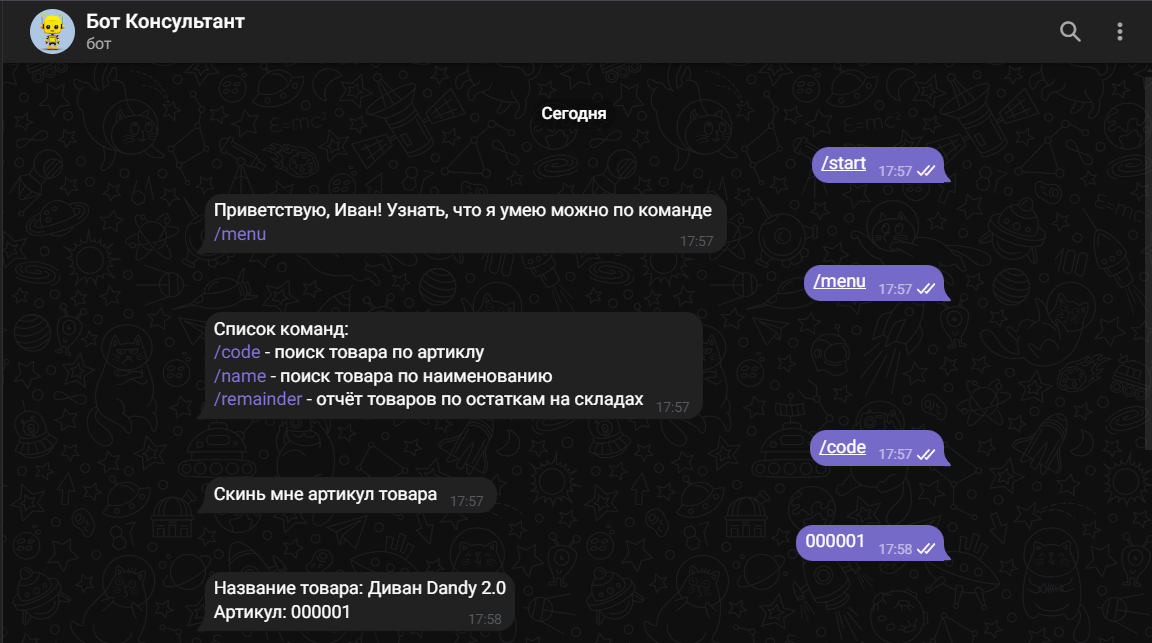


Рисунок 8 – Тестирование черным ящиком

В таблице 8 представлен результат юзабилити-тестирования.

Таблица 8 – Результаты проведения юзабилити-тестирования

| Вопрос | Попов Андрей Евгеньевич | Троцкий Юрий Александрович | Козлов Владислав Владимирович | Шелковникова Анастасия Анатольевна | Комаров Александр Иванович |
| --- | --- | --- | --- | --- | --- |
| Присутствуют ли такие дизайн-элементы, которые неприятны для зрения? | нет | нет | нет | нет | нет |
| Присутствует ли отвлекающая информация? | нет | нет | нет | нет | нет |
| Понятна ли структура Чат-Бота на интуитивном уровне? | да | да | да | да | да |
| Смогли ли вы найти интересующую вас информацию без затруднений? | да | да | да | да | да |
| Понятен ли смысл Чат-бота? | да | да | да | да | да |
| Общая оценка | 7/10 | 9/10 | 8/10 | 8/10 | 7/10 |

Средняя оценка бота, учитывающая ответы на все вопросы пользователей, указывает на высокое качество внешнего графического интерфейса. Информационная система не содержит отвлекающих элементов, так как все компоненты интерфейса гармонично сочетаются между собой.

Результаты функционального тестирования доступны в разделе 3.3 "Проведение функционального тестирования готового программного продукта".

Вывод по главе: было совершенно проведение трех популярных методов тестирования бота — "Чёрный ящик", функциональное и юзабилити тестирование, которые принесли ценные сведения о производительности бота.

# ЗАКЛЮЧЕНИЕ

Во время производственной практики в IT компании "Форус" я провел анализ предметной области, спроектировал и разработал Телеграм-бота, который поможет консультантам находить товары по артикулу и его наименованию.

Цели практики были достигнуты через анализ и разработку бота, подтверждая выполнение задач. В будущем бот может быть улучшен добавлением новых функций и улучшением дизайна.

Проект в компании принес мне ценный опыт и поддержку коллег. Практика в "Форусе" позволила познакомиться с рабочей культурой и процессами разработки ПО. Опыт в компании был интересным и полезным, позволяя применить знания на практике и улучшить навыки в IT.

Прохождение практики в такой компании дает возможность студентам получить ценный опыт работы с использованием новейших технологий и методик разработки. Важно отметить, что прохождение практики в IT компании "Форус" дает студентам возможность не только применить свои знания и навыки на практике, но и получить ценные рекомендации и советы от опытных специалистов в области IT, а также получить поддержку в дальнейшем развитии своего IT-проекта.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения [Электронный доступ]. — Режим доступа: [ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения (rostest.info)](https://rostest.info/gost/001.001.040.035/gost-20886-85/#:~:text=%D0%93%D0%9E%D0%A1%D0%A2%2020886%2D85.%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2,%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D0%BE%2D%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%) (Дата обращения: 06.06.2024 г.);
2. ГОСТ Р. 59793–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Электронный доступ]. — Режим доступа: [ГОСТ Р. 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (internet-law.ru)](https://internet-law.ru/gosts/gost/77858/?ysclid=lifap277mg519765664) (Дата обращения: 06.06.2024 г.);
3. ГОСТ 19.201-78. ЕСПД. Общие положения. [Электронный доступ]. — Режим доступа: [ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению (swrit.ru)](https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lpi5u6naom706376064) (Дата обращения: 06.06.2024 г.);
4. Microsoft Visio [Электронный доступ]. — Режим доступа: [Microsoft Visio — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_Visio) (Дата обращения: 06.06.2024 г.);
5. 1С: Предприятие 8.3 [Электронный доступ]. — Режим доступа: [1С: Предприятие 8.3 — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/1%D0%A1:%D0%9F%D1%80%D0%B5%D0%B4%D0%BF%D1%80%D0%B8%D1%8F%D1%82%D0%B8%D0%B5) (Дата обращения: 06.06.2024 г.);
6. [Виолетта Филатова](https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%BB%D0%B0%D1%82%D0%BE%D0%B2%D0%B0,_%D0%92%D0%B8%D0%BE%D0%BB%D0%B5%D1%82%D1%82%D0%B0_%D0%9E%D0%BB%D0%B5%D0%B3%D0%BE%D0%B2%D0%BD%D0%B0&action=edit&redlink=1). 1С: Предприятие 8.1 Бухгалтерия предприятия. Управление торговлей. Управление персоналом / Екатерина Кондукова. — СПб: [БХВ](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — С. 288;
7. [Алексей Гладкий](https://ru.wikipedia.org/w/index.php?title=%D0%93%D0%BB%D0%B0%D0%B4%D0%BA%D0%B8%D0%B9,_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B5%D0%B9_%D0%90%D0%BD%D0%B0%D1%82%D0%BE%D0%BB%D1%8C%D0%B5%D0%B2%D0%B8%D1%87&action=edit&redlink=1). 1С 8 с нуля: комплексное руководство для начинающих / А. Спивак. — [Феникс](https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B5%D0%BD%D0%B8%D0%BA%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. — С. 478;
8. Visual Studio Code [Электронный доступ]. — Режим доступа: [Visual Studio Code. Подробный гайд по настройке и установке плагинов — Андриканич Евгений (habr.com)](https://habr.com/ru/articles/490754/) (Дата обращения: 06.06.2024 г.);
9. Visual Studio Code [Электронный доступ]. — Режим доступа: [Visual Studio Code — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Visual_Studio_Code) (Дата обращения: 06.06.2024 г.);

# ПРИЛОЖЕНИЯ 1 – Листинг кода

Листинг кода JS:

const TelegramBot = require('node-telegram-bot-api');  
const token = '7050755395:AAHCzc0HOZo42GsqHLv\_FFJtwZhcUlAWmk8';  
const bot = new TelegramBot(token, {polling: true});  
  
console.log('Bot has been started ...')  
  
bot.onText(/\/start/, async (msg) =>   
{  
console.log(msg);  
const chatId = [msg.chat.id](https://web.telegram.org/a/msg.chat.id);  
await bot.sendMessage(chatId, Приветствую, ${msg.from.first\_name}! Узнать, что я умею можно по команде /menu);  
});  
  
bot.onText(/\/menu/, async (msg) =>   
{  
const chatId = [msg.chat.id](https://web.telegram.org/a/msg.chat.id);  
await bot.sendMessage(chatId, 'Список команд: \n/code - поиск товара по артиклу\n/name - поиск товара по наименованию\n/remainder - отчёт товаров по остаткам на складах');  
});  
  
bot.onText(/\/code/, async (msg) => {  
const chatId = [msg.chat.id](https://web.telegram.org/a/msg.chat.id);  
const http = require('http');  
  
const options = {  
host: 'localhost',  
path: '/bot/hs/telegram/test',  
headers: {  
'Authorization': 'Basic 0JDQtNC80LjQvdC40YHRgtGA0LDRgtC+0YA6'  
}  
};  
  
http.get(options, (res) => {  
let data = '';  
  
res.setEncoding('utf8');  
  
res.on('data', (chunk) => {  
data += chunk;  
});  
  
res.on('end', () => {  
try {  
const jsonData = JSON.parse(data);  
if (jsonData.text) {  
bot.sendMessage(chatId, jsonData.text);  
} else {  
bot.sendMessage(chatId, "Неверный формат данных");  
}  
} catch (error) {  
bot.sendMessage(chatId, "Ошибка обработки данных");  
}  
});  
}).on('error', (err) => {  
console.log("Error: " + err.message);  
});  
});  
  
bot.onText(/\/name/, async (msg) =>   
{  
const chatId = [msg.chat.id](https://web.telegram.org/a/msg.chat.id);  
await bot.sendMessage(chatId, 'Скинь мне наименование товара');  
});  
  
bot.onText(/\/remainder/, async (msg) =>   
{  
const chatId = [msg.chat.id](https://web.telegram.org/a/msg.chat.id);  
await bot.sendMessage(chatId, 'Вот Excel файл с остатками на складе: ');  
});

Листинг кода 1C:

Функция testGET(Запрос)  
Товар = Справочники.Номенклатура.НайтиПоРеквизиту("Артикул",Артикул);   
Если НЕ ЗначениеЗаполнено(Товар) Тогда  
СтруктураОтвета = Новый Структура("text", "Не работает");   
//КодТовара = Запрос.ПараметрыURL["CustomerCode"];  
//Товар = ПолучитьНазваниеТовара(КодТовара);   
Возврат ВернутьУспехИОтвет(СтруктураОтвета);  
Иначе //3. заполним структуру данных о клиенте  
ДанныеТовара = Новый Структура;   
ДанныеТовара.Вставить("Code", Товар.Код);  
ДанныеТовара.Вставить("Name", Товар.Наименование);   
ДанныеТовара.Вставить("Artikul", Товар.Артикул);  
// ДанныеКлиента.Вставить("Email", Товар.Вес);   
//4. сериализуем данные о клиенте в JSON  
ЗаписьJSON = Новый ЗаписьJSON;   
ЗаписьJSON.УстановитьСтроку();  
ЗаписатьJSON(ЗаписьJSON, ДанныеТовара);   
СтрокаДляОтвета = ЗаписьJSON.Закрыть();  
//5. сформируем ответ   
Ответ = Новый HTTPСервисОтвет(200);  
Ответ.Заголовки.Вставить("Content-type", "application/JSON; charset=utf-8");   
Ответ.УстановитьТелоИзСтроки(СтрокаДляОтвета, КодировкаТекста.UTF8, ИспользованиеByteOrderMark.НеИспользовать);  
Возврат Ответ; КонецЕсли;  
КонецФункции   
  
Функция ВернутьУспехИОтвет(СтруктураОтвет) Экспорт  
Ответ = Новый HTTPСервисОтвет(200);  
СтрокаJSON = ПолучитьСтрокуJSON(СтруктураОтвет);  
Ответ.УстановитьТелоИзСтроки(СтрокаJSON);  
Ответ.Заголовки.Вставить("Content-type", "application/json");  
Возврат Ответ;  
КонецФункции   
  
Функция ПолучитьСтрокуJSON(Значение) Экспорт   
ЗаписьJSON = Новый ЗаписьJSON;  
ЗаписьJSON.УстановитьСтроку();  
ЗаписатьJSON(ЗаписьJSON, Значение);  
Возврат ЗаписьJSON.Закрыть();   
КонецФункции