СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc170282003)

[1 ХАРАКТЕРИСТИКА ОБЪЕКТА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ 4](#_Toc170282004)

[1.1 Анализ предметной области 5](#_Toc170282005)

[1.2 Программная и техническая архитектура ИС предприятия 6](#_Toc170282006)

[1.3 Разработка и анализ требований к программному модулю 10](#_Toc170282007)

[1.4 Выработка требований к программному обеспечению и программному модулю 10](#_Toc170282008)

[2 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 13](#_Toc170282009)

[2.1 Проектирование программного продукта 13](#_Toc170282010)

[2.2 Функциональная схема программного продукта 14](#_Toc170282011)

[2.3 Проектирование пользовательского интерфейса 15](#_Toc170282012)

[2.4 Кодирование программного обеспечения 17](#_Toc170282013)

[2.5 Организация обработки исключений. Отладка модулей программного проекта 17](#_Toc170282014)

[3 ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ 18](#_Toc170282015)

[3.1 Тестирование программного обеспечения 18](#_Toc170282016)

[3.1.1 Выбор и обоснование методики проведения тестирования 18](#_Toc170282017)

[3.1.2 Схема выполнения экспериментального тестирования 18](#_Toc170282018)

[3.2 Инспектирование компонентов программного продукта на предмет соответствия стандартам кодирования 19](#_Toc170282019)

[3.3 Проведение функционального тестирования готового программного продукта 22](#_Toc170282020)

[3.4 Документирование результатов тестирования 22](#_Toc170282021)

[ЗАКЛЮЧЕНИЕ 25](#_Toc170282022)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 26](#_Toc170282023)

[ПРИЛОЖЕНИЯ 28](#_Toc170282024)

# ВВЕДЕНИЕ

Цель практической подготовки в виде производственной практики по профессиональному модулю 01 Разработка модулей программного обеспечения для компьютерных систем – является разработка HTTP-сервиса на 1С, который будет отвечать на запросы, заданые из Node.js.

Основные задачи:

* Создать HTTP-сервис на 1С, в которой заранее будет установлена УНФ;
* Опубликовать созданный HTTP-сервис и разместить его с помощью ПО Apache;
* Реализовать программу в Node.js, которая будет отправлять запрос в созданную УНФ на 1С и выводить ответ.

Чат-Бот станет незаменимым помощником для персонала магазинов, облегчая их рабочие обязанности и повышая общий уровень обслуживания клиентов. Использование такого инструмента позволит сократить время на поиск товаров, улучшить качество консультирования и обеспечить более быструю и точную информацию о наличии товаров в магазине. Разработка Чат-Бота для консультантов магазинов является ключевым шагом к повышению эффективности работы персонала и улучшению сервиса для клиентов.

# ХАРАКТЕРИСТИКА ОБЪЕКТА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

ООО НПФ «Форус» входит в ТОП 100 крупнейших ИТ-компаний России (по данным CNews, TAdviser, Эксперт-РА). Активно развивает рынок информационныхтехнологий с 1992 года. Головной офис в г. Иркутск, представительства в 16 регионах России и в Монголии (Улан-Батор). В компании более 400 специалистов, более 300 из них сертифицированы в «1С».

Компания ООО НПФ ФОРУС, адрес: Иркутская обл., г. Иркутск, ул. Ямская, д. 1/1 офис 1 зарегистрирована 01.10.2002. Организации присвоены ИНН 3812023430, ОГРН 1023801752633, КПП 381101001. Основным видом деятельности является разработка компьютерного программного обеспечения, всего зарегистрировано 25 видов деятельности по ОКВЭД. Имеет связи с 2 компаниями.

Количество совладельцев: 4, генеральный директор - Денисова Валентина Сергеевна. Размер уставного капитала 10 000₽.Компания ООО НПФ ФОРУС принимала участие в 99 тендерах. В отношении компании было возбуждено 3 исполнительных производства. ООО НПФ ФОРУС участвовало в 14 арбитражных делах: в 7 в качестве истца, и в 3 в качестве ответчика.

Направления деятельности:

* Создание централизованных облачных информационных систем управления финансово-хозяйственной деятельностью органов исполнительной власти и бюджетных учреждений. Более 10 реализованных проектов федерального и регионального уровня. Двукратный победитель конкурса «1С: Проект года» 2017, 2018.
* Цифровизация производственных и торговых предприятий и внедрение ERP-систем от Калининграда до Владивостока. Является рамках программы гос. поддержки для реализации проектов цифровизации производственных предприятий.
* Реализация международных проектов с 2015 года: Монголия, Индонезия, Вьетнам. Первое международное внедрение английской версии программы 1С:ЕRP в крупнейшем холдинге Монголии на заводе Cocа Cola.
* Дистрибьюция программного обеспечения «1С» и других вендоров, поставка оборудования через партнёрскую сеть в 19 городах России, >1 400 партнёров.
* Обеспечение информационной безопасности.
* Крупнейший контакт-центр по сопровождению «1С» за Уралом, >10 000 клиентов.
* Разработка и внедрение машинного обучения, искусственного интеллекта, бизнес-аналитики и других современных технологий в собственной лаборатории R&D.

Основной вид деятельности по ОКВЭД:

* 62.01 Разработка компьютерного программного обеспечения.

Дополнительные виды деятельности по ОКВЭД:

* 18.11 Печатание газет
* 46.43 Торговля оптовая бытовыми электротоварами
* 46.43.3 Торговля оптовая грампластинками, аудио- и видеомагнитными лентами, компакт-дисками (CD) и цифровыми видеодисками (DVD) (кроме носителей без записей)
* 46.49.32 Торговля оптовая газетами и журналами
* 46.51 Торговля оптовая компьютерами, периферийными устройствами к компьютерам и программным обеспечением
* 46.52.3 Торговля оптовая аудио- и видеомагнитными лентами и дискетами, магнитными и оптическими дисками, компакт-дисками (CD), цифровыми видеодисками (DVD) и прочими техническими носителями информации без записей
* 46.66 Торговля оптовая прочей офисной техникой и оборудованием
* 46.90 Торговля оптовая неспециализированная
* 47.41 Торговля розничная компьютерами, периферийными устройствами к ним и программным обеспечением в специализированных магазинах
* 47.62.1 Торговля розничная газетами и журналами в специализированных магазинах  
  47.63.1 Торговля розничная музыкальными записями, аудиолентами, компакт-дисками и кассетами в специализированных магазинах
* 47.63.2 Торговля розничная лентами и дисками без записей в специализированных магазинах
* 58.13 Издание газет
* 62.02 Деятельность консультативная и работы в области компьютерных технологий  
  62.03 Деятельность по управлению компьютерным оборудованием
* 62.09 Деятельность, связанная с использованием вычислительной техники и информационных технологий, прочая
* 63.11 Деятельность по обработке данных, предоставление услуг по размещению информации и связанная с этим деятельность
* 63.11.1 Деятельность по созданию и использованию баз данных и информационных ресурсов
* 68.20 Аренда и управление собственным или арендованным недвижимым имуществом  
  69.20.2 Деятельность по оказанию услуг в области бухгалтерского учета
* 70.22 Консультирование по вопросам коммерческой деятельности и управления  
  72.19 Научные исследования и разработки в области естественных и технических наук прочие
* 73.11 Деятельность рекламных агентств
* 73.20.1 Исследование конъюнктуры рынка

## 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. Также важно собирать обратную связь от клиентов для оценки качества предоставляемых услуг и постоянного совершенствования работы.

Использование бота в Мессенджере Телеграм, интегрированного с базой данных в системе 1С УНФ, позволит консультантам "СмартКонсалт" 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 "СмартКонсалт" 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

На компьютерах предприятия ООО НПФ «Форус» используют различные операционные системы, такие как Windows и Linux, для запуска и управления компьютерами и серверами. Для хранения и управления данными также используем различные системы управления базами данных (СУБД), такие как Microsoft SQL Server, PostgreSQL, Oracle и MySQL. Кроме того, они используют всеми привычное программное обеспечение компании Microsoft. В основном в компании используют Microsoft Outlook для коммуникаций по почте, Битрикс24, как систему управления задачами, а также как корпоративный мессенджер. ООО НПФ «Форус» выбрало программное обеспечение 1С, во-первых, потому что они являются официальным дистрибьютором этого продукта и так же потому, что оно обладает множеством функций, необходимых для эффективного учета финансовых операций и для продуктивной и полноценной работы компании. С помощью 1С они ведут учет доходов и расходов, управляют банковскими операциями, осуществляют расчеты по заработной плате и налогам, а также проводят другие финансовые операции. Кроме того, данное программное обеспечение предоставляет инструменты для планирования и учета проектов, включая управление временем, бюджетом, ресурсами и задачами. Также они используют 1С для управления контактами с клиентами, учета продаж, создания и отправки счетов, а также отслеживания заказов и других операций, связанных с обслуживанием и поддержкой клиентов, как пример 1С: CRM.

Для создания сайтов применяются следующие языки программирования: HTML, JavaScript и CSS. HTML используется для разметки содержимого, JavaScript - для добавления интерактивности, а CSS - для стилизации и оформления. Фреймворки, такие как Angular, React и Vue.js, предлагают разработчикам комплексный набор инструментов и библиотек, которые значительно упрощают процесс создания веб-приложений. Они позволяют эффективно применять язык программирования HTML и создавать большие и интерактивные веб-сайты.

CMS, такие как WordPress, Drupal и Joomla, предоставляют удобные интерфейсы для создания, управления и обновления контента на веб-сайтах. Они позволяют пользователям без особых знаний в программировании легко управлять своими веб-сайтами и поддерживать их актуальность. Разработчики могут выбрать из различных редакторов кода, таких как Visual Studio Code, Sublime Text или Atom, чтобы писать и редактировать код веб-страниц. Эти редакторы предлагают широкий набор инструментов и функций, которые помогают улучшить производительность и эффективность разработки. Для создания графики и дизайна веб-сайтов используют графические редакторы, такие как Adobe Photoshop или Sketch. Компания использует самые лучшие инструменты для создания привлекательных и профессиональных изображений и макетов, которые могут быть использованы на веб-страницах.

Также разработчики в компании не могут обойтись без платформ управления версиями, таких как Git. Они помогают отслеживать изменения в коде и облегчают сотрудничество между разработчиками. Кроме того, для работы над проектами на компьютерах разработчиков устанавливаются специализированные программы, такие как интегрированные среды разработки (IDE) - Visual Studio, IntelliJ IDEA, Eclipse и PyCharm. IDE предоставляют широкий набор инструментов для написания, отладки и тестирования кода. Для работы с базами данных программисты используют подходящие под их нужды системы управления базами данных (СУБД), например MySQL, PostgreSQL, Oracle или Microsoft SQL Server. Кроме того, для удобной работы с выбранной СУБД установлены соответствующие клиенты, такие как MySQL Workbench или pgAdmin.

ООО НПФ «Форус» использует преимущественно моноблоки, но также у них имеются ноутбуки и иногда стационарные компьютеры. В основном используются моноблоки компании HP, но также имеются моноблоки фирмы Acer. Снабжены они процессорами: Intel Pentium CPU G640 2.80GHz, Core i5-1235U, Core i3-1215U. Количество оперативной памяти (ОЗУ) варьируется от 4 ГБ до 32 ГБ. Видеокарты зачастую интегрированы в процессор (Intel Graphics), однако у ноутбуков в компании с дискретной видеокартой Nvidia GTX 1050, Nvidia GTX 1650, Nvidia RTX 2060. В качестве носителей внешней памяти выступают твердотельные накопители (SSD) объемом от 100ГБ до 250ГБ, на которых установлена операционная система и жесткие диски (HDD) объемом от 500ГБ до 1ТБ, на них хранится вся остальная информация (документация, ПО, и тд). Стационарные компьютеры снабжены процессорами Intel Core i5-7000, оперативной памятью в размере 16ГБ в двухканальном режиме, видеокартами AMD Radeon RX 5000 и Nvidia GTX 1050ti, накопители внешней памяти имеются только жесткие диски (HDD) объемом 500ГБ, мониторы от компании Acer, Asus и LG.

Предприятием используются принтеры, в основном это многофункциональные устройства (МФУ) и лазерные их всего 7, они используются для печати и ксерокопирования документации. Для печати на фотобумаги и некоторых цветных бумаг имеется струйный принтер. Для некоторых мероприятий, обучению технике безопасности и собраний используется проекторы, которые установлены в каждых переговорных комнатах и комнатах для собраний.

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

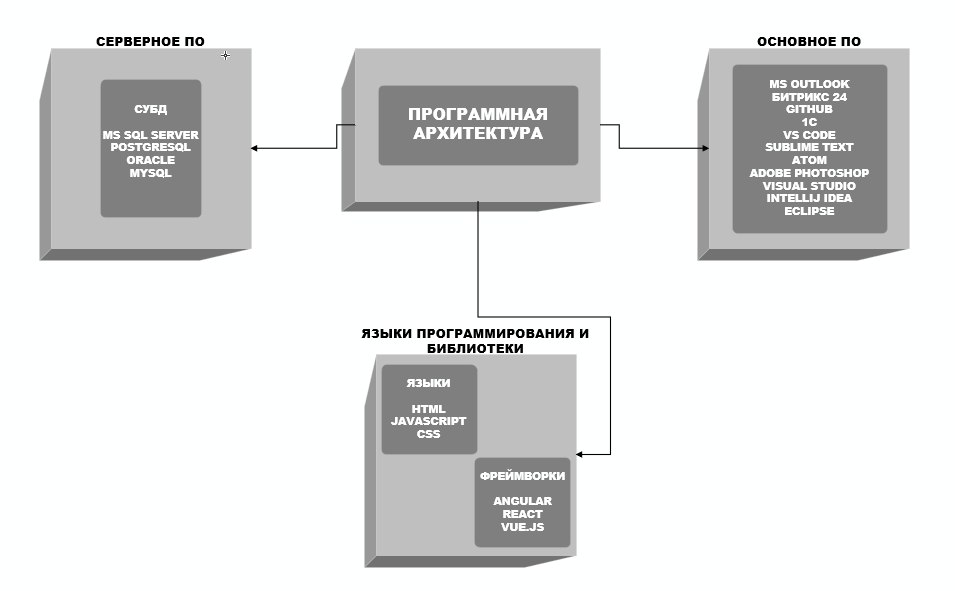


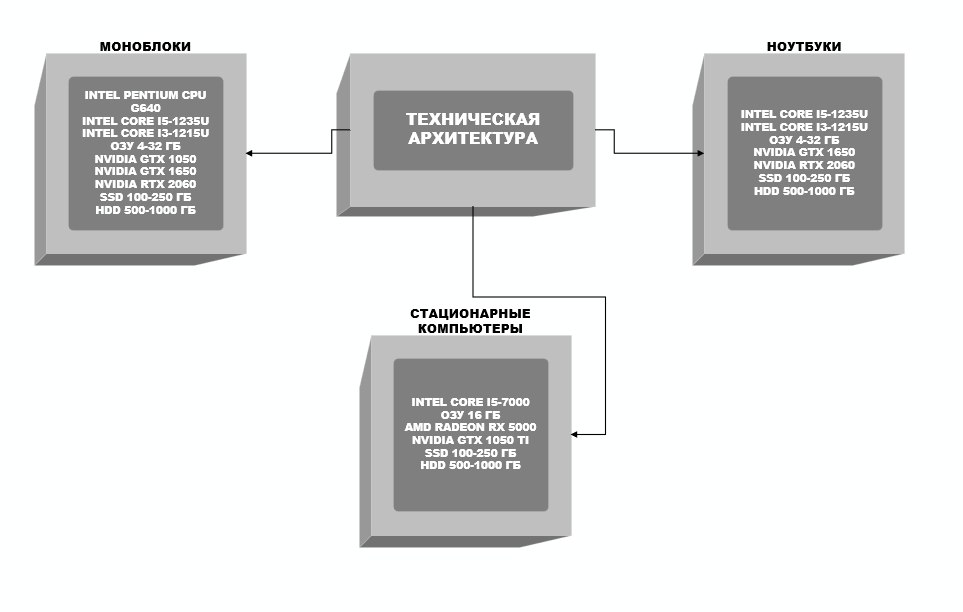
Рисунок 1 – Программная архитектура

Рисунок 2 – Техническая архитектура

## Разработка и анализ требований к программному модулю

Чат-бот должен обеспечить дополнительный канал поддержки консультантов. Чат-бот должен быть интегрирован с Мессенджером Телеграм и базой данных в системе 1С УНФ.

Задачи для внедрения чат-бота:

* создание «единого окна» управления взаимодействиями с консультантами;
* получение обратной связи от консультанта;
* потенциальные выгоды проекта как для консультанта, так и для магазина или фирмы;
* получение конкурентных преимуществ;
* увеличение количества каналов информирования клиентов;
* повышение качества обслуживания и лояльности клиентов.

Чат-бот должен обеспечивать выполнение следующих бизнес-процессов:

* обработка запроса на поиск товаров;
* обработка запроса информации о статусе товара;
* обработка запроса доступности специальных и дополнительных услуг.

## Выработка требований к программному обеспечению и программному модулю

Технические характеристики системы

| **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

Чат-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

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

Общие требования к надежности Чат-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:

* чат-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+ в веб-браузере Chrome;
* iOS13+ в веб-браузерах Safari;

Вывод по главе: была описана информация об организации, программная и техническая архитектура предприятия, а также был проведён анализ предметной области, из которого были выведены цели, задачи и характеристики проекта.

# РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

## Проектирование программного продукта

Для описания того, как система работает на разных уровнях абстракции, используются диаграммы деятельности. Они помогают упростить процесс программирования при разработке программного обеспечения.

Для концептуального моделирования функциональности используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов - диаграмма деятельности.

На Рисунке 3 представлен пример диаграммы вариантов использования системы "Чат-Бот".

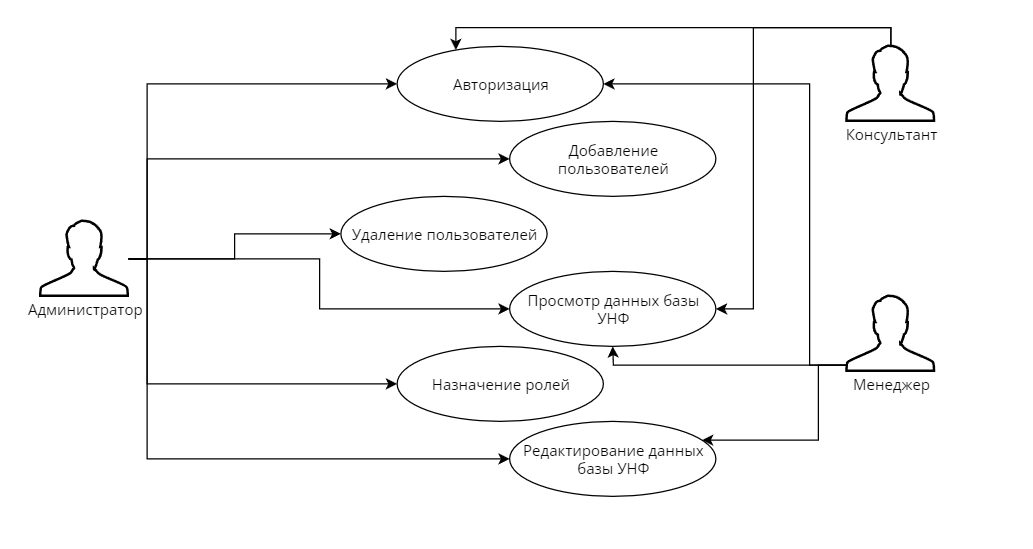


Рисунок 3 – Диаграмма вариантов использования

На Рисунке 4 представлен пример диаграммы деятельности системы «Чат-Бот».

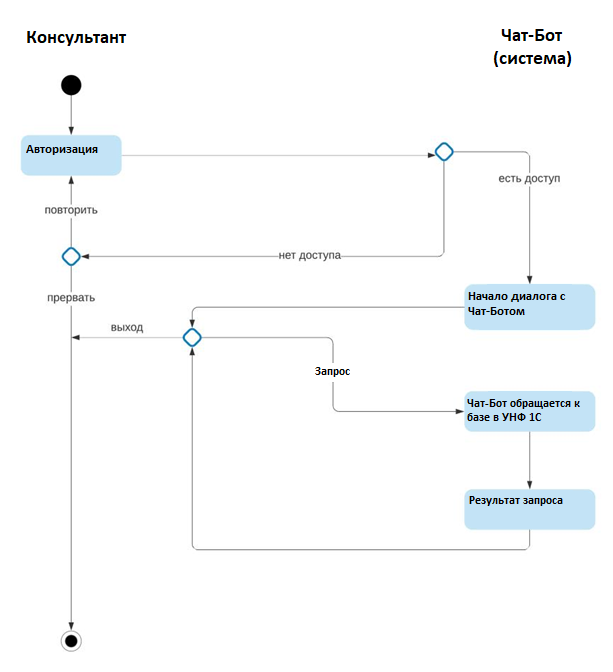


Рисунок 4 – Диаграмма детальности

## Функциональная схема программного продукта

Единая система программной документации включает в себя схемы алгоритмов, программ, данных и систем, а также условные обозначения и правила их выполнения.

Данный стандарт распространяется на условные обозначения (символы), используемые в схемах алгоритмов, программ, данных и систем, и устанавливает правила выполнения схем. Он предназначен для отображения различных видов задач обработки данных и средств их решения. Стандарт не регулирует форму записей и обозначений, которые могут находиться внутри символов или использоваться рядом с ними для уточнения выполняемых ими функций.

Ниже представлена функциональная схема данных системы Чат-Бота на Рисунке 5.

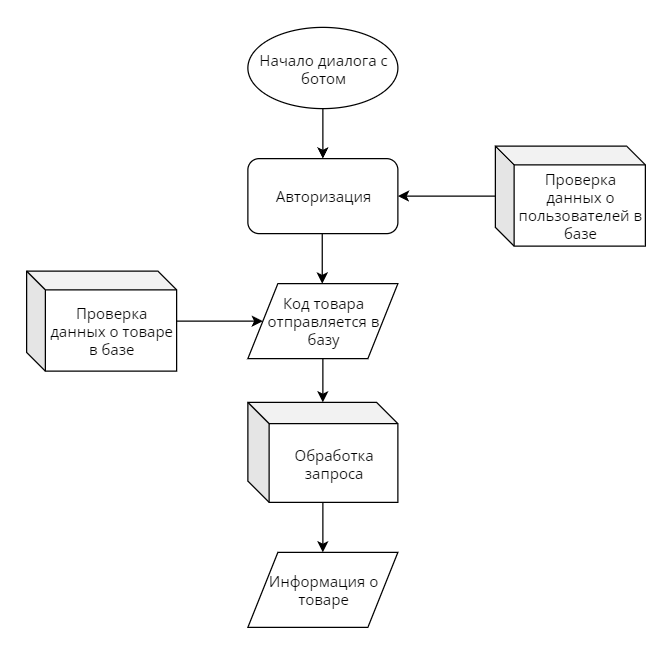
****

Рисунок 5 – Схема данных системы Чат-Бота

## Проектирование пользовательского интерфейса

Чат-Бот был разработан в среде разработки Visual Studio Code, где бот и его библиотека данных были написаны и хранились в JavaScript.

Visual Studio Code – это текстовый редактор, созданный Microsoft для Windows, Linux и macOS. Он предназначен для "легкой" кроссплатформенной разработки веб- и облачных приложений, включает отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Редактор имеет широкие возможности для настройки: пользовательские темы, сочетания клавиш и файлы конфигурации. Visual Studio Code распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

База данных, из которой Чат-Бот извлекал необходимую информацию, хранилась и разрабатывалась в системе 1С УНФ.

1С – это российская компания, основанная в 1991 году, специализирующаяся на разработке, издании и поддержке компьютерных программ, баз данных для делового и домашнего использования, а также компьютерных игр. Основной продукт компании – программная система "1С: Предприятие", изначально созданная как расширяемая бухгалтерская система с собственным встроенным языком программирования. В дальнейшем она охватила многие функции продуктов классов ERP, CRM, HRM, SCM. Название компании происходит от названия собственной поисковой программы: не более 1 секунды требовалось для получения информации.

Ниже представлена Таблица 1, описывающая команды и их значение в диалоге между Чат-Ботом и пользователем.

Таблица 2 – Команды, Свойства и Значения Чат-Бота

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

Ниже представлен рисунок 6, на котором изображены перечень кнопок для взаимодействия с ботом в чате.



Рисунок 6 – Схема кнопок для взаимодействия с ботом

## Кодирование программного обеспечения

Программный код публикации HTTP-Сервиса представлен в таблице 3.

Таблица 3 – 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 |

Далее идет программный код со стороны Node.js для связки написаного бота с 1С, на которой реализована библиотека.

Программный код команд /menu, /code и /remainder представлены в таблице 4.

Таблица 4 – 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: ');  }); |

## 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 сразуже ликвидирована.

На рисунке 7 представлена отладка модулей программного проекта.

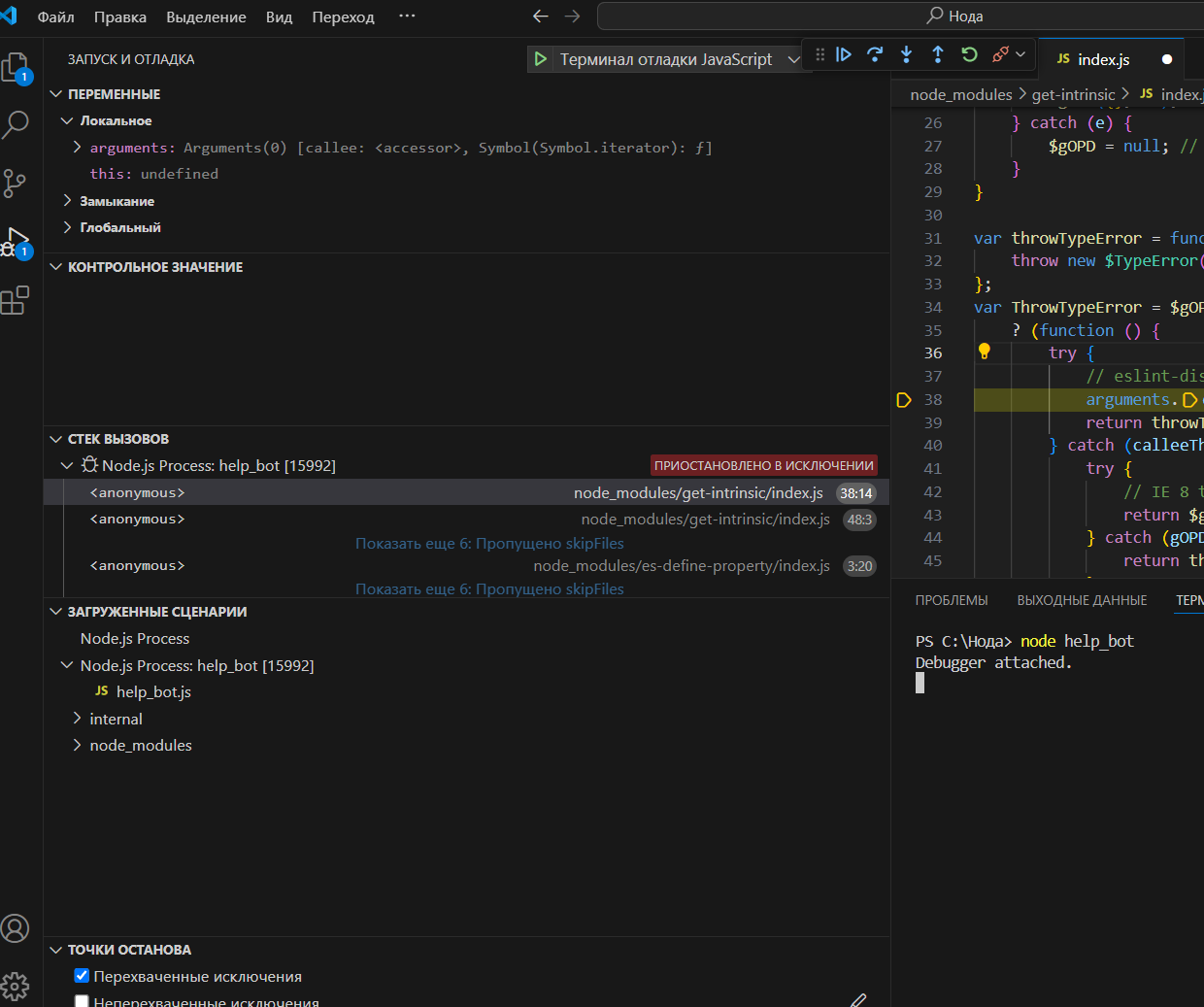


Рисунок 7 – Отладка модулей программного проекта

Вывод по главе:

После проведения проектирования программного продукта, создания диаграмм вариантов использования, детальности и схемы данных для системы Чат-Бота, а также разработки пользовательского интерфейса и кодирования программного обеспечения, а также выполнения организации обработки исключений и отладки модулей программного проекта, можно сказать что разработка програмных модулей выполнена полностью.

# 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 компонентов 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 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 | 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 |
| 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 Чат-Бота. Результат представлен 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 артиклу | Ввести артикул  Пример (000001) | Название товара: Диван Dandy 2.0 Артикул: 000001 | Название товара: Диван Dandy 2.0 Артикул: 000001 | Да |

На рисунке 8-9 представлены тесты один и два и их результат.



Рисунок 8 – Получение ответа от Чат-бота из УНФ, находящейся на 1С 

Рисунок 9 – Получение информации о товаре по его артиклу

## Документирование результатов тестирования

В результате проведения тестирования «чёрным ящиком» не были выявлены ошибки, мешающие работе Чат-Бота. Пример работы Чат бота представлен на рисунке 10.

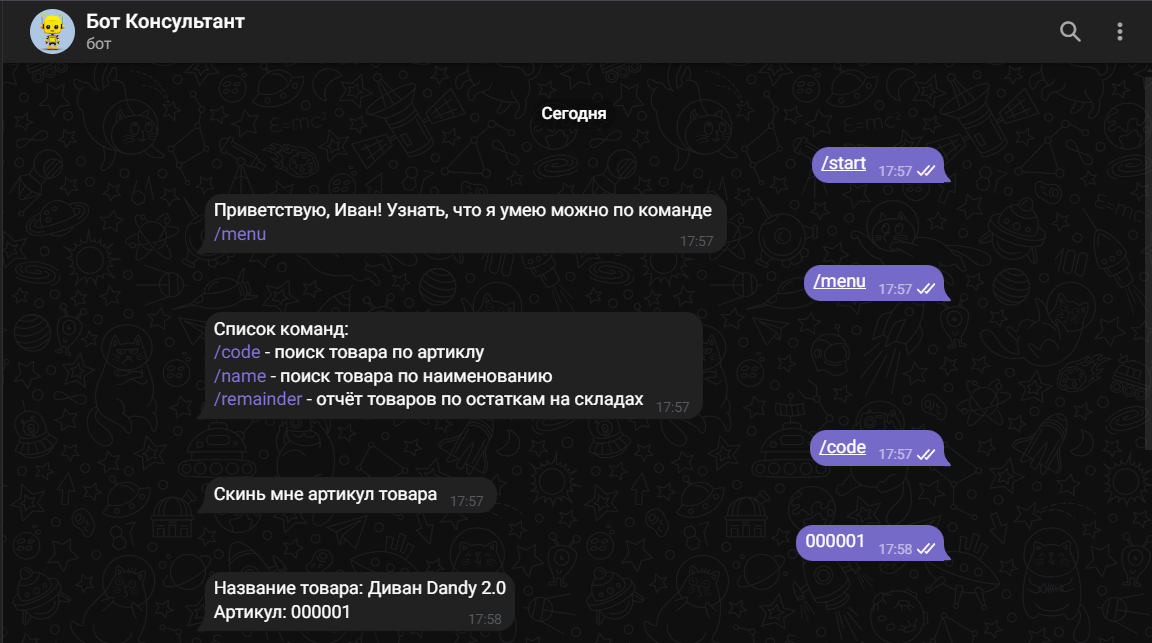


Рисунок 10 – Тестирование черным ящиком

В таблице 8 представлен результат эксперементального 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? | нет | 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 |

Исходя из средней оценки Чат-бота, рассчитанной на основе общих оценок всех пользователей с учетом ответов на все поставленные вопросы, можно сделать вывод о том, что внешние характеристики графического интерфейса находятся на довольно высоком уровне. Также в информационной системе нет помех для пользователя в виде отвлекающей информации, так как все отображаемые элементы не выглядят инородно и сочетаются с другими элементами интерфейса.

Результат функционального тестирования приведён в разделе 3.3 «Проведение функционального тестирования готового программного продукта».

Вывод по главе:

В результате проведения трех методик тестирования чат-бота - "Чёрным ящиком", эксплуатационного и функционального тестирования, были получены ценные данные о качестве работы бота.

# ЗАКЛЮЧЕНИЕ

Во время производственной практики в IT компании "Форус" я провел анализ предметной области, спроектировал и разработал Чат-Бот, который помогает консультантам находить товары по артикулу.

Цели практики были успешно достигнуты через анализ и разработку бота, что подтверждает выполнение поставленных задач. В будущем бот может быть улучшен путем добавления новых функций и улучшения дизайна.

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

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

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 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: 30.04.2022 г.);
2. 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: 22.05.2013 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: 01.10.1981 г.);
4. Дебби Валковски. Microsoft Office Visio 2003 для «чайников» = Visio 2003 For Dummies. — М: «Диалектика», 2006. — С. 336;
5. Расширения конфигураций. Адаптация прикладных решений с сохранением поддержки. Хрусталева [Электронный доступ]. — Режим доступа: <https://www.litres.ru/static/or3/view> (Дата обращения: 11.09.2023 г.);
6. 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: 17.10.2023 г.);
7. [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;
8. [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;
9. 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: 03.03.2020 г.);
10. Visual Studio Code [Электронный доступ]. — Режим доступа: [Visual Studio Code — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Visual_Studio_Code) (Дата обращения: 17.03.2024 г.);
11. Alessandro Del Sole. [Visual Studio Code Succinctly](https://www.syncfusion.com/resources/techportal/details/ebooks/Visual_Studio_Code_Succinctly). — SyncFusion Inc., 2016. — C. 128;
12. [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: Управление торговлей 8.2 с нуля. Конфигурирование и настройка. 75 уроков для начинающих / 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), 2012. — С. 240.

# ПРИЛОЖЕНИЯ

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