|  |  |
| --- | --- |
| C:\Users\T'rain\Desktop\логотип 2016 УКРТБдля документов.jpg | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БАШКОРТОСТАН  Государственное бюджетное профессиональное образовательное учреждение  Уфимский колледж радиоэлектроники, телекоммуникаций и безопасности |

Тема: **«**Разработка чат-бота для системы технической поддержки OsTicket»

**ДИПЛОМНАЯ РАБОТА**

Дипломант: Садыков Артур Константинович

(фамилия, имя, отчество)

Руководитель работы: Казанцев Андрей Валерьевич

(фамилия, имя, отчество)

**Индивидуальный график выполнения ВКР**

|  |  |
| --- | --- |
| Студент | Садыков Артур Константинович |
| Группа | 9ИСП-423К-18 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Утверждение задания** | **I предзащита** | **II предзащита** | **III предзащита** | **% оригинальности** |
| «\_\_» \_\_20\_\_ г. | «\_\_» \_\_20\_\_ г. | «\_\_» \_\_20\_\_ г. | «\_\_» \_\_20\_\_ г. |  |
| Заполненное задание на ВКР | 50% ВКР | 80% ВКР | 100% ВКР.  Сдача на рецензию |
| Отметка о выполнении  \_\_\_\_\_\_\_\_\_ | Отметка о выполнении  \_\_\_\_\_\_\_\_\_ | Отметка о выполнении  \_\_\_\_\_\_\_\_\_ | Отметка о выполнении  \_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_ |

Рассмотрено

на заседании кафедры

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Протокол № \_\_ от «\_\_» \_\_\_\_\_20\_\_ г.

Зав. кафедрой \_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_/

|  |  |
| --- | --- |
| C:\Users\T'rain\Desktop\логотип 2016 УКРТБдля документов.jpg | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БАШКОРТОСТАН  Государственное бюджетное профессиональное образовательное учреждение  Уфимский колледж радиоэлектроники, телекоммуникаций и безопасности |

УТВЕРЖДАЮ

Зам. директора, к.т.н.

\_\_\_\_\_\_\_ Туктарова Л. Р. «\_\_» \_\_\_\_\_\_\_\_\_\_ 2022г.

**ТЕМА ДИПЛОМНОЙ РАБОТЫ**

|  |
| --- |
| «Разработка чат-бота для системы технической поддержки |
| OsTicket» |
|  |

**З А Д А Н И Е**

**на дипломную работу**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| студенту группы | 9ИСП-423К-18 | | | |
|  | (номер группы) | | | |
| Садыкову Артуру Константиновичу | | | | |
| (фамилия, имя, отчество) | | | | |
| специальность | 09.02.07 Информационные системы и программирование | | | |
|  | (код и наименование специальности) | | | |
| тема предложена | | ФГБОУ УГАТУ | | |
|  | (наименование организации) | | | |
| руководитель работы | | | Казанцев Андрей Валерьевич | |
|  | (фамилия, имя, отчество) | | | |
| Дата выдачи задания | | | | «4» апреля 2022 г. |
| Дата окончания выполнения работы | | | | «1» июня 2022 г. |

**ОТЗЫВ** [[1]](#footnote-1)⋅

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| на дипломный проект (работу) студента | | | | | |  |
|  | | | | (наименование колледжа) | | |
|  | | | | | | |
| (Ф.И.О.) | | | | | | |
| специальности | |  | | | | |
|  | | (код и наименование специальности) | | | | |
| Группа |  | | Тема | |  | |
|  |  | |  | |  | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнение проекта (работы) заслуживает | | | |  | | оценки |
| Место работы и должность рецензента | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| Фамилия, имя, отчество | |  | | | | |
|  | |  | | | | |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г. | | | | | | |
| Подпись |  | | | |

М.П.

**РЕЦЕНЗИЯ**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| на дипломный проект (работу) студента | | | | | | |  |
|  | | | | | (наименование колледжа) | | |
|  | | | | | | | |
| (Ф.И.О.) | | | | | | | |
| специальности | |  | | | | | |
|  | | (код и наименование специальности) | | | | | |
| Наименование темы | | |  | | | | |
|  |  | | |  | |  | |
|  | | | | | | | |

[[2]](#footnote-2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнение проекта (работы) заслуживает | | | |  | | оценки |
| Место работы и должность рецензента | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| Фамилия, имя, отчество | |  | | | | |
|  | |  | | | | |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г. | | | | | | |
| Подпись |  | | | |

М.П.

|  |  |
| --- | --- |
| C:\Users\T'rain\Desktop\логотип 2016 УКРТБдля документов.jpg | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БАШКОРТОСТАН  Государственное бюджетное профессиональное образовательное учреждение  Уфимский колледж радиоэлектроники, телекоммуникаций и безопасности |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

К ДИПЛОМНОМУ ПРОЕКТУ (РАБОТЕ)

УКРТБ.ДП(ДР)09.02.01.001 ПЗ

(обозначение документа)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| К защите допущен | | Дипломант |  |  |
|  |  |  | (подпись) | (и., о., фамилия) |
| Зам.директора |  | Руководитель |  |  |
|  | Л. Р. Туктарова | проекта (работы) |  |  |
| (подпись) | (и., о., фамилия) |  | (подпись) | (и., о., фамилия) |
|  | | Консультанты: |  |  |
| (дата) | | - по ИКТ |  |  |
| Рецензент |  |  | (подпись) | (и., о., фамилия) |
|  |  | - по нормоконтролю текстовой и | | |
| (подпись) | (и., о., фамилия) | графической части | |  |
|  | |  |  |  |
| (дата) | |  | (подпись) | (и., о., фамилия) |

Уфа 2022

**Содержание**

Изм.ен.

Лист

№ докум.

Подпись

Датааааа

Лист

2

УГКР.ДП080504.022 ПЗ

.000ПЗ

Разраб.

Провер.

Экон.контр

Н. Контр.

*.*

Утверд.

Лит.

Листов

9ИСП-423К-18

1. **Общая часть**
   1. **Описание предметной области**

Боты – это специальные программы, выполняющие различные функции и упрощающие жизнь их пользователей. Написанные для платформы Telegram, они предназначены для выполнения самых разных функций: от получения новостей до поиска информации и даже торговли акциями. Главное задачей бота является автоматический ответ после введенной ему пользователем команды. При этом, работая непосредственно через интерфейс Telegram, программа имитирует действия живого юзера, за счет чего пользование таким ботом гораздо удобно и понятно.

Именно поэтому, многие компании, развивающие бизнес через интернет, используют возможности ботов по нескольким причинам:

1. Они позволяют задействовать очередной канал коммуникации с целевой аудиторией (в России Телеграмм пользуется около 10 миллионов человек).
2. Они быстро выполняют однообразную работу, позволяя разгрузить наемных сотрудников, тем самым экономя деньги компании.

В Telegram используется один общий вид ботов, которых от обычных юзеров отличает только наличие приставки «bot» в имени. Сами же боты делятся на несколько направлений:

* **Чат-боты**. Представляют из себя простейший чат, имитирующий общение на заданную пользователем тематику.
* **Боты-информаторы**. Отдельный вид ботов, главная цель которых — информирование пользователя о тех или иных событиях (новости, мероприятия, публикации и т. п.).
* **Игровые боты.** Боты, в которых можно поиграть в различные игры. По большей части, это текстовые версии разных игр времен Combats.ru
* **Боты-ассистенты.** Боты, разработанные различными онлайн-сервисами как дополнение к основной веб-версии.

На самом деле, четкого деления нет, т. к. некоторые боты содержат сразу несколько механик и успешно выполняют множество пользовательских задач. С их помощью можно переводить, обучаться, тестировать, искать информацию, играть в игры и даже пользоваться другими сервисами и взаимодействовать с вещами, имеющими выход в глобальную сеть (популярный нынче «интернет-вещей»).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Благодаря этому боты стали карманными помощниками, пользоваться которыми можно, даже не покидая мессенджер. Они предоставляют возможность решать элементарные задачи при помощи мгновенных команд,

Изм.

Лист

№ докум.

Подпись

Дата

Лист

при всех этих программах не нуждаются в установке и не занимают отдельное место в памяти вашего девайса.

Алгоритм работы бот-утилит достаточно прост. Сообщения, команды и запросы, отправленные пользователями, передаются на программное обеспечение, запущенное на серверах разработчиков. Посреднический анонимный сервер Telegram обрабатывает шифрование и осуществляет обратную связь между утилитой и пользователем.

Взаимодействие между пользователем и ботом выглядит следующим образом:  
*Пользователь бота отдает ему команду* -> *Бот передает команду на сервер* -> *Программа на сервере обрабатывает полученный от бота запрос* -> *Сервер отдает ответ боту* -> *Бот выводит ответ на экран приложения пользователю*.

И этот цикл повторяется раз за разом, когда вы нажимаете на кнопки и взаимодействуете с любым телеграмм-ботом.

Вы общаетесь с серверами с помощью простого HTTPS-интерфейса, который представляет собой упрощенную версию API Telegram. По-другому этот интерфейс можно назвать программным каталогом или бот-алгоритмом. Новые bot-утилиты создаются с помощью специальной утилиты @BotFather, который значительно упрощает процесс разработки.

OsTicket - это система организация технической поддержки пользователей в режиме реального времени, включая отслеживание потоков электронной почты, вход на почтовый сервер по протоколу POP3, неограниченное количество адресов email, методы избежания зацикливания автоответчиков, и др.

**OsTicket** - это альтернатива более дорогих и сложных систем поддержки клиентов. Простой в настройке, лёгкий и надёжный Web-интерфейс. Но самое главное, это абсолютно бесплатно.

OsTicket сделана чтобы упростить прием запросов, а также улучшить эффективность обслуживания ваших клиентов, предоставляя персоналу все необходимые средства для быстрой и эффективной поддержки. Некоторые основные возможности включают:

* **Web и Email Поддержка**  
  Запросы могут быть созданы по email, через онлайн форму или по

телефону (создается персоналом). Гибкая конфигурация и отображение.

* **Авто оповещение**  
  Автоматическая отправка сообщения, когда открыта новая заявка или

получено сообщение. Настраиваемые почтовые шаблоны.

* **Готовые ответы**  
  Готовые ответы для часто задаваемых вопросов.
* **Внутренние Сообщения**  
  Добавление внутренних сообщений к заявкам для персонала.
* **Темы обращения**  
  Настройка тем обращений для веб заявок.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

* **Предупреждения и уведомления**

Сотрудники и клиенты получают оповещения по электронной почте о ходе работ над заявками. Уведомления имеют гибкие настройки.

* **Контроль доступа**  
  Полный контроль над персоналом по средствам групп и отделов.
* **Назначение и передача заявок**  
  Назначение заявок сотруднику или отделу.
* **Без регистрации**  
  Нет необходимости в регистрации пользователей (для входа используется ID/email).
* **История**

Сохраняются все запросы и ответы.

* 1. **Обзор аналогичных программных продуктов**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Боты «Телеграм» универсальны и выполняют множество задач на любой вкус. Найти информацию, скачать книгу, фильм, музыку, развлечь — это лишь маленький перечень возможностей ботов.

**Скачать видео с YouTube**— [@Save\_youbot](https://t.me/Save_youbot)

Это бот для скачивания видео с YouTube. С его помощью вы сможете загрузить видео любого объема прямиком в Telegram

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Рисунок 1 – Телеграм-бот для скачивания видео с YouTube.

**Yandex.Translate**— [@YTranslateBot](https://t.me/YTranslateBot)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание Инструмент, интегрированный с переводчиком Яндекса.

Рисунок 2 – Переводчик интегрированный с Telegram ботом.

**BotFather** — [@BotFather](https://t.me/BotFather)

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описаниеТелеграм-бот для создания других телеграмм-ботов

Рисунок 3 – Телеграм-бот для создания телегам-ботов

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**2. Специальная часть**

**2.1 Аналитическая часть**

**2.1.1 Постановка задачи**

Создание бота включает в себя несколько этапов:

* скачивание и установка набора инструментов для разработки;
* загрузка и установка необходимых зависимостей;
* настройка среды программирования;
* регистрация бота;
* создание проекта;
* определение и создание команд;
* настройка и внедрение команд;
* написание кода и интеграция с внешними сервисами;
* тестирование и отладка.

**2.1.2 Разработка алгоритма работы бота**

Бот обрабатывает команды, поступающие ему от пользователя:

* «/start» – бот должен активизироваться и начать свою работу.
* «/auth {id}» - при активации бота пользователь авторизуется через данную команду вводя id, где id – идентификатор пользователя системы.
* «/all\_tickets» - при вводе данной команды, пользователь получает список доступных ему заявок в системе.

Изм.

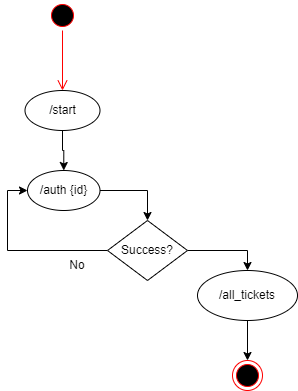
Лист

№ докум.

Подпись

Дата

Лист

Рисунок 4 – UML-диаграмма алгоритма работы бота

**2.1.3 Разработка диаграммы Use Case программы**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

На диаграмме Use Case представлен актер и его роли, которые он

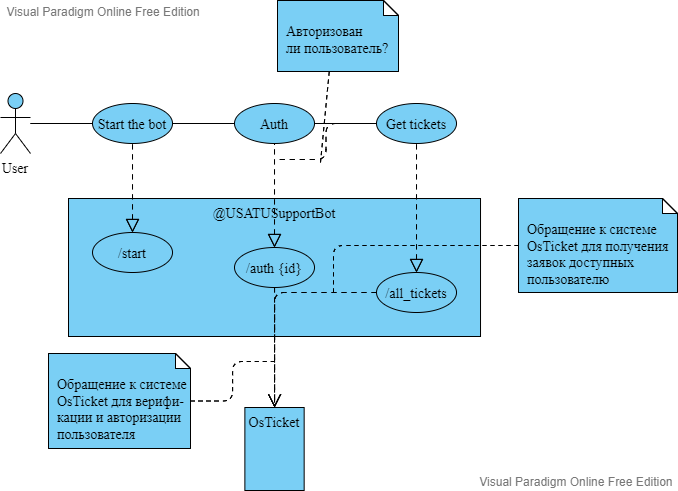
****выполняет. На рисунке 4 представлен 1 актер: user. Он может запустить бота, затем авторизоваться, а после получить список всех доступных ему заявок в системе.

Рисунок 5 – Use Case диаграмма

**2.1.4 Обоснование выбора языка программирования**

Для написания телеграмм-бота можно использовать разные технологии. Наибольшей популярностью для создания телеграмм-ботов пользуются Python, Java и еще с десяток языков программирования из рейтинга TIOBE. В конечном итоге я оставил только два языка программирования которые полностью удовлетворяли мои потребности.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++.

Java - это полнофункциональный объектно-ориентированный язык, который поддерживает все три «столпа» объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Он имеет прекрасную поддержку компонентов, надежен и устойчив благодаря использованию «сборки мусора», обработки исключений, безопасности типов. Язык Java разрабатывался с нуля и вобрал в себя много полезных свойств таких языков, как C++, Visual Basic, а также Pascal и Delphi.

При этом необходимость обратной совместимости с предыдущими версиями отсутствовала, что позволило языку Java избежать многих отрицательных сторон своих предшественников. Telegram позволяет использовать любой язык программирования. Из всех языков программирования я решил выбрать Java, так как это очень перспективный язык и достаточно простой для освоения. Основная идея Java — это универсальность. Можно реализовать все на одном языке и для любой платформы. Он неплохо подходит как для начинающих в качестве первого языка, так и для реализации крупных коммерческих проектов.

В качестве языка программирования для реализации данного дипломного проекта был выбран: Java.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**2.1.5 Обоснование выбора инструментальных средств**

Для разработки на Java существует несколько популярных сред разработки: Eclipse и IntelliJ Idea.

Eclipse — свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Наиболее известные приложения на основе Eclipse Platform — различные «Eclipse IDE» для разработки ПО на множестве языков (например, наиболее популярный «Java IDE», поддерживавшийся изначально, не полагается на какие-либо закрытые расширения, использует стандартный открытый API для доступа к Eclipse Platform).

В данной работе была использована среда разработки – IntelliJ Idea. Эта программа обеспечивает высокое качество кода на протяжении всего цикла жизни ПО, от проектирования до внедрения. Какие бы приложения ни разрабатывались: для Windows, Windows Phone, Android, Ios и других платформ, Idea — это универсальное решение. Сегодня это основное и самое эффективное средство разработки решений для платформы Java. Idea продолжает традиции, заложенные предыдущими версиями продукта, и содержит множество новинок, предназначенных как для упрощения создания приложений для различных платформ отдельными, так и для повышения производительности всей проектной команды. Поэтому мною была выбрана именно эта среда разработки.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**2.2 Практическая часть**

**2.2.1 Разработка Telegram-бота**

Исходя из поставки задач были проведены следующие работы:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

* установка IntelliJ Idea;
* установка JDK;
* создание Maven проекта;
* добавление зависимостей;
* написание кода;
* тестирование и исправление ошибок

Устанавливаем IntelliJ Idea с последующей настройкой – рисунок 6.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – установка IntelliJ Idea.

Устанавливаем JDK – рисунок 7.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – установка JDK.

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описание Создаем Maven проект – рисунок 8.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Рисунок 8 – создание Maven проекта.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст

Автоматически созданное описание Добавляем необходимые зависимости в pom.xml – рисунок 9.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Рисунок 9 – добавление необходимых зависимостей.

**2.2.2 Регистрация Телеграмм-бота**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описаниеЧто бы бот был доступен в телеграмм его необходимо зарегистрировать в другом бот, BotFather. При регистрации необходимо указать уникальный идентификатор для бота - @USATUSupportBot, в данном случае, и имя бот, USATU Support.

Рисунок 10 – регистрация бота в телеграмм.

После регистрации BotFather отправит в ответ уникальный идентификатор телеграмм-бота в системе телеграмм, в данном случае - *5185789510:AAEBENUnLBle0bkpd1EMXbjM9E\_i0YypLP4*

**2.2.3 Формирование запросов Telegram-бота**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Что бы наш бот мог обрабатывать команды, поступающие ему от пользователя, необходимо написать соответствующий код.

Для начала следует научить бота приветствовать пользователя, для этого он должен уметь обрабатывать команду «/start», ниже представлен код, обрабатывающий данную команду.

@Slf4j  
@Component  
public final class StartAbility implements AbilityExtension {  
  
 private static final InputFile *STICKER* = new InputFile("CAACAgIAAxkBAAEErPRidqr-GJExJkWzOovKMIR9pAF8MgACMAEAAqtXxAtP62N9ym6CbyQE");  
   
 private static final String *GREETING* = """  
 Привет \*\*{0}\*\*!  
 Введи команду \*/auth [OsTicketUserId]\*  
 (\*[OsTicketUserId]\* - идентификатор пользователя системы OsTicket).  
 """;  
  
 @SuppressWarnings("unused")  
 public Ability start() {  
 return Ability.*builder*()  
 .action(this::sendSticker)  
 .post(this::sendGreeting)  
 .name(*START*.getText())  
 .privacy(*PUBLIC*)  
 .locality(*USER*)  
 .build();  
 }  
  
 private void sendSticker(MessageContext ctx) {  
 try {  
 ctx.bot().sender().sendSticker(SendSticker.*builder*()  
 .chatId(ctx.chatId().toString())  
 .sticker(*STICKER*)  
 .build());  
 } catch (TelegramApiException e) {  
 *log*.error("An error occurred during sending a sticker to the user.", e);  
 }  
 }  
  
 private void sendGreeting(MessageContext ctx) {  
 try {  
 ctx.bot().sender().execute(SendMessage.*builder*()  
 .text(*format*(*GREETING*, ctx.user().getFirstName()))  
 .chatId(ctx.chatId().toString())  
 .parseMode(*MARKDOWN*)  
 .build());  
 } catch (TelegramApiException e) {  
 *log*.error("An error occurred during sending a greeting to the user.", e);  
 }  
 }  
  
}

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание На рисунке 11 представлен результат работы данного кода.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Рисунок 11 – приветствие пользователя.

Далее необходимо научить бота обрабатывать команду «/auth {id}», чтобы пользователь мог регистрироваться в нем. Ниже представлен код, обрабатывающий данную команду.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

@Component  
@RequiredArgsConstructor  
public final class AuthAbility implements AbilityExtension {  
  
 private final Mapper<User, TelegramUser> userMapper;  
 private final UserInfoRepository userInfoRepository;  
 private final AuthenticationService authService;  
  
 @SuppressWarnings("unused")  
 public Ability start() {  
 return Ability.*builder*()  
 .action(this::authenticate)  
 .name(*AUTH*.getText())  
 .input(1)  
 .privacy(*PUBLIC*)  
 .locality(*USER*)  
 .build();  
 }  
  
 private void authenticate(MessageContext ctx) {  
 var osTicketUserId = ctx.firstArg();  
 if (!*hasText*(osTicketUserId)) return;  
  
 authService.findOsTicketUserById(osTicketUserId).ifPresentOrElse(  
 osTicketUser -> {  
 var telegramUser = userMapper.map(ctx.user());  
 saveUserInformation(telegramUser, osTicketUser);  
 sendKeyboard(ctx);  
 },  
 () -> sendMessage(ctx, "Неверный идентификатор пользователя")  
 );  
 }  
  
 @SneakyThrows  
 private void sendKeyboard(MessageContext ctx) {  
 var row = new KeyboardRow();  
 row.add(KeyboardButton.*builder*()  
 .text("/all\_tickets")  
 .build());  
  
 var keyboardMarkup = ReplyKeyboardMarkup.*builder*()  
 .keyboard(List.*of*(row))  
 .resizeKeyboard(true)  
 .build();  
  
 ctx.bot().sender().execute(SendMessage.*builder*()  
 .chatId(ctx.chatId().toString())  
 .text("Вы успешно авторизованы")  
 .replyMarkup(keyboardMarkup)  
 .build());  
 }  
  
 private UserInfo saveUserInformation(TelegramUser telegramUser, OsTicketUser osTicketUser) {  
 var userInfo = buildUserInfo(telegramUser, osTicketUser);  
 return userInfoRepository.save(userInfo);  
 }  
  
 private UserInfo buildUserInfo(TelegramUser telegramUser, OsTicketUsers.OsTicketUser osTicketUser) {  
 return UserInfo.*builder*()  
 .userId(UserPK.*builder*()  
 .osTicketId(osTicketUser.userId())  
 .telegramId(telegramUser.userId())  
 .build())  
 .createdAt(osTicketUser.createdAt())  
 .firstName(telegramUser.firstName())  
 .lastName(telegramUser.lastName())  
 .userName(telegramUser.userName())  
 .systemName(osTicketUser.name())  
 .build();  
 }  
  
 @SneakyThrows  
 private void sendMessage(MessageContext ctx, String message) {  
 ctx.bot().execute(SendMessage.*builder*()  
 .chatId(ctx.chatId().toString())  
 .text(message)  
 .build());  
 }  
  
}

На рисунке 12 представлен результат работы данного кода.

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описаниеРисунок 12 – авторизация пользователя.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Далее необходимо обучить бота возвращать заявки конкретного пользователя (пользователя что прошел успешную авторизацию). Ниже представлен код, отвечающий за это.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

@Component  
@RequiredArgsConstructor  
public final class AllTicketsAbility implements AbilityExtension {  
  
 private final UserInfoRepository userInfoRepository;  
 private final RestTemplate restTemplate = new RestTemplate();  
  
 @SuppressWarnings("unused")  
 public Ability allTickets() {  
 return Ability.*builder*()  
 .name("all\_tickets")  
 .action(this::findUserInfo)  
 .privacy(Privacy.*PUBLIC*)  
 .locality(Locality.*USER*)  
 .build();  
 }  
  
 @SneakyThrows  
 private void findUserInfo(MessageContext ctx) {  
 var telegramUserId = ctx.user().getId();  
 var user = userInfoRepository.findByUserId\_TelegramId(telegramUserId);  
 if (user.isEmpty()) return;  
 var tickets = findAllTickets(user.get());  
 if (tickets.isEmpty()) return;  
 ctx.bot().sender().execute(SendMessage.*builder*()  
 .chatId(ctx.chatId().toString())  
 .text(tickets.toString())  
 .build());  
 }  
  
 private List<Ticket> findAllTickets(UserInfo userInfo) {  
 var result = restTemplate.getForEntity("http://thisbot.ru/ost\_wbs/?apikey={apikey}&query=ticket&condition=all&sort=status&parameters={statusId}",  
 Tickets.class, Map.*of*("apikey", "AB5F9CB3B226E2F3667581C77CAB87DC", "statusId", 0));  
 if (!result.hasBody()) return *emptyList*();  
 var body = result.getBody();  
 assert body != null;  
  
 return body.getData().getTickets().stream()  
 .flatMap(Collection::stream)  
 .filter(it -> it.getUserId().equals(userInfo.userId().osTicketId()))  
 .toList();  
 }  
}

На рисунке 13 показан результат работы данного кода.

Изображение выглядит как текст, снимок экрана, экран, компьютер

Автоматически созданное описаниеРисунок 13 – получение заявок.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**2.2.4 Разработка информационной структуры**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

В качестве метода хранения данных используется база данных h2.

H2 — открытая кроссплатформенная СУБД, полностью написанная на языке Java.

Несмотря на малый размер (чуть более 1 МБ) H2 поддерживает следующие возможности «из коробки»:

* Два режима работы (клиент-сервер, встроенный)
* Два режима хранения данных (файловая система, память)
* Поддержка планов выполнения запросов
* Поддержка кластеризации и репликации
* Шифрование данных
* Внешние (связанные) таблицы
* Драйвер ODBC
* Полнотекстовый поиск
* Определение доменов
* Мультиверсионный конкурентный доступ
* Поддержка последовательностей
* Поддержка ключевых слов LIMIT и OFFSET в запросах
* Временные таблицы
* Вычисляемые столбцы
* Пользовательские агрегатные функции
* Пользовательские хранимые процедуры
* Сжатие CLOB/BLOB объектов
* Работа с CSV файлами на чтение и запись
* Браузерная консоль управления
* Запуск как сервис Windows.

Основные причины почему была выбрана именно эта база данных:

* Легкая интеграция с Java-приложениями
* Кроссплатформенность
* Большая защищенность, чем у нативных приложений
* Пользовательские функции и триггеры работают очень быстро
* Поддержка Юникода.

СУБД весьма полезна в разработке, так как позволяет создавать тестовые наборы данных прямо в памяти, имеет развитые возможности запросов, многочисленные функции, совместимые с более тяжеловесными БД, легко настраивается, имеет подробную и понятную документацию. Для небольших программ СУБД возможно использовать саму по себе, так как она легко встраивается в приложение.

**2.3 Описание Telegram-бота**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Телеграм-бот имеет название «USATU Support» и имеет уникальный идентификатор в системе телеграмм @USATUSupportBot, по которому каждый пользователь телеграмм сможет воспользоваться им.

Что бы начать пользоваться ботом необходимо отправить ему команду «/start» и под начнет функционировать, отправив в ответ смайл и текстовое сообщение с приветствием.

Для полноценного использования бота пользователь должен авторизоваться в нем.

Для это необходимо отправить команду «/auth» с указанием идентификатора пользователя в системе osTicket. Если авторизация пройдет успешно – бот покажет меню дополнительных команд доступных только авторизованному пользователю. Если успешной авторизации не произошло, например пользователя нет в системе или введены не верные данные – бот покажет соответствующее сообщение.

При успешной авторизации бот представляет меню дополнительных команд, в котором есть команда «/all\_tickets». Используя данную команду, пользователь может получить все доступные ему заявки, созданные в системе osTicket. В случае если что-то пойдет не так, например произойдет сбой сети или по каким-то причинам система osTicket будет не доступна бот отправит в ответ соответствующее сообщение.

**2.4 Разработка инструкции пользователя**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описаниеЧто бы запустить телеграмм-бота введите команду «/start», как показано на рисунке ниже.

Рисунок 14 – команда «/start»

Далее, чтобы пользоваться полным функционалом бота, необходимо авторизоваться. Для этого введите команду «/auth {id}», где вместо {id} укажите идентификатор пользователя в системе osTicket, как показано на рисунке ниже.

Рисунок 15 – команда «/auth»Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Изображение выглядит как текст, снимок экрана, экран, компьютер

Автоматически созданное описание Что бы получить заявки выберите в меню команду «/all\_tickets», как показано на рисунке ниже.

Рисунок 16 – команда «/all\_tickets» **2.5 Мероприятия по охране труда и противопожарной безопасности**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

**2.5.1 Описание требований по технике безопасности при работе с вычислительной техникой.**

С развитием научно-технического прогресса немаловажную роль играет возможность безопасного исполнения людьми своих трудовых обязанностей. В связи с этим была создана и развивается наука о безопасности труда и жизнедеятельности человека.

Безопасность жизнедеятельности — это комплекс мероприятий, направленных на обеспечение безопасности человека в среде обитания, сохранение его здоровья, разработку методов и средств защиты путем снижения влияния вредных и опасных факторов до допустимых значений, выработку мер по ограничению ущерба в ликвидации последствий чрезвычайных ситуаций мирного и военного времени.

На рабочем месте должны быть предусмотрены меры защиты от возможного воздействия опасных и вредных факторов производства. Уровни этих факторов не должны превышать предельных значений, оговоренных правовыми, техническими и санитарно-техническими нормами. Эти нормативные документы обязывают к созданию на рабочем месте условий труда, при которых влияние опасных и вредных факторов на работающих либо устранено совсем, либо находится в допустимых пределах.

**2.5.2 Характеристика условий труда программиста**

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе за компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

В помещениях вычислительных центров необходимо применить систему комбинированного освещения. При выполнении работ категории высокой зрительной точности величина коэффициента естественного

освещения должна быть не ниже 1,5%, а при зрительной работе средней точности должен быть не ниже 1,0%.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Ученые считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности электромагнитного излучения от мониторов не существует, а вот ионизирующее очень опасно, если излучение имеет высокую частоту, оно способно разрушать ДНК и клетки, выбивая электроны из атомов. Такой вид излучения мы называем радиацией. Никакие приборы ни при какой мощности не смогут оказать ионизирующий эффект. Для этого нужна очень высокая частота (от 75000 гц)

**2.5.3 Эргономические требования к рабочему месту**

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники. Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям.

Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Эргономическими аспектами проектирования видеотерминальных рабочих мест являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте, характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места. Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя. Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства. Дисплей размещается в центре, системный блок размещается в предусмотренной нише стола.

Для комфортной работы стол должен удовлетворять следующим условиям: высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники; нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги; поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста; конструкция стола должна предусматривать наличие выдвижных ящиков; высота рабочей поверхности рекомендуется в пределах

680-760мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм; большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420–550 мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый. Положение экрана определяется: расстоянием считывания (0,6…0,7м); углом считывания, направлением взгляда на 20 градусов ниже горизонтали к центру экрана, причем экран перпендикулярен этому направлению. Должна также предусматриваться возможность регулирования экрана: по высоте +3 см; по наклону от -10 до +20 градусов относительно вертикали; в левом и правом направлениях. Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие: голова не должна быть наклонена более чем на 20 градусов; плечи должны быть расслаблены; – локти - под углом 80 -100 градусов; предплечья и кисти рук - в горизонтальном положении. Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы - низко, некуда положить руки и кисти, недостаточно пространство для ног.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук. Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50–60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем, когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз.

Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

1. ⋅Отзыв должен содержать:

   1. Наименование темы проекта (работы), упоминание о количестве страниц пояснительной записки и количества листов в графической части.
   2. Краткий перечень основных вопросов, изложенных в пояснительной записке.
   3. Оценку соответствия темы ВКР профилю специальности «…» и содержанию одного или нескольких профессиональных модулей.

   4. Характерные особенности работы, ее достоинства и недостатки. Обязательную характеристику проекта (работы) с точки зрения его актуальности для промышленности (рекомендован для внедрения, производством используются материалы проекта (работы), изготовлен макет и т. д.). Если тема проекта (работы) является частью общей разработки, то это должно быть обязательно отмечено.

   5. Оценку уровня освоения общих и профессиональных компетенций, знания, умения обучающегося продемонстрированные им при выполнении ВКР.

   6. Степень самостоятельности обучающегося, его личный вклад в раскрытие проблем и разработку предложений по их решению. Отношение обучающегося к выполнению ВКР, проявленные (не проявленные) им способности.

   7.Характеристику проведённой экспериментальной работы студентом по изготовлению макета.

   8.Отзыв заканчивается выводом о возможности (невозможности) допуска ВКР к защите. [↑](#footnote-ref-1)
2. Рецензия должна содержать: а) заключение о степени соответствия ВКР заявленной теме и заданию на нее; б) заключение о соответствии темы ВКР профилю специальности и содержанию одного или нескольких профессиональных модулей; в) характеристику выполнения каждого раздела проекта (работы) и степени использования дипломантом последних достижений науки, техники и новаторов производства; г) оценка качества выполнения графической части проекта (работы) и пояснительной записки; д) перечень положительных качеств проекта (работы) и его основных недостатков; е) оценку степени разработки поставленных вопросов и практической значимости работы. Общая оценка качества выполнения ВКР даётся по пятибалльной системе. [↑](#footnote-ref-2)