**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

преподаватели

\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Голубева

\_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Мухина

«18»\_\_апреля\_\_2025 г.

**ОТЧЕТ ПО**

**ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Ведяйкин А.В. 18.04.2025 г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Батракова В.В. 18.04.2025 г.

Специальность, группа: 09.02.07, 43П

Нижний Новгород

2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc195713631)

[АНАЛИЗ ПРЕДПРИЯТИЯ 4](#_Toc195713632)

[1.1. Описание деятельности предприятия 4](#_Toc195713633)

[1.2. Аппаратное и программное обеспечение предприятия 5](#_Toc195713634)

[1.3. Организация обмена данных между сотрудниками 6](#_Toc195713635)

[1.4. Использование антивирусного ПО и других средств защиты 6](#_Toc195713636)

[ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 7](#_Toc195713637)

[2.1. Этапы разработки 7](#_Toc195713638)

[2.2. Описание предметной области 8](#_Toc195713639)

[2.3. Создание даталогической модели БД 9](#_Toc195713640)

[2.4. Выбор программных средств для разработки приложения 10](#_Toc195713641)

[2.5. Создание и заполнение БД 11](#_Toc195713642)

[2.6. Реализация информационной системы 14](#_Toc195713643)

[ЗАКЛЮЧЕНИЕ 20](#_Toc195713644)

[ПРИЛОЖЕНИЕ А 21](#_Toc195713645)

[ПРИЛОЖЕНИЕ Б 39](#_Toc195713646)

# ВВЕДЕНИЕ

Производственная практика по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем проходила в ООО «Кодерлайн» в период с 22.03.2025г. по 18.04.2025г.

Целью производственной практики является закрепление знаний общих и профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Для достижения цели были поставлены следующие задачи:

1. изучить техническое задание и сформировать алгоритм разработки;
2. разработать базу данных в соответствии с техническим заданием;
3. разработать пользовательский интерфейс;
4. реализовать функционал приложения;
5. реализовать функционал приложения;
6. провести рефакторинг и оптимизировать код.

Практическим результатом прохождения практики является отчёт по выполнению заданий с представленными в нём рисунками, таблицами и диаграммами.

# АНАЛИЗ ПРЕДПРИЯТИЯ

## Описание деятельности предприятия

«Кодерлайн» - аккредитованная IT компания, которая входит в пятёрку крупнейших партнеров фирмы «1С».

Команда фирмы автоматизируем бизнес-процессы на небольших, средних и крупных предприятиях с 2014 года. Внедряет и сопровождает программные продукты фирмы 1С - ERP, Управление Холдингом, Документооборот и другие. В штате компании более 500 высококвалифицированных сотрудников — программистов, консультантов, бизнес-аналитиков, методологов и руководителей проектов.

Эксперты компании обладают значительным опытом отраслевой автоматизации в нефтегазовой промышленности, медиаиндустрии, строительстве, оптовой и розничной торговле, легкой и пищевой промышленности, аренде и продаже недвижимости и других отраслях.

Успешный опыт внедрений в крупнейших компаниях страны подтвержден отзывами таких предприятий как «Газпром», «Лукойл», «Ростех», «ПЭК», «Пепсико» и других. Каждый год мы заявляем о себе в конкурсе фирмы 1С «Проект года». В 2022 году победили в номинации «Самый масштабный проект» с проектом автоматизации  Государственной корпорации «Ростех» на 21 500 рабочих мест.

Высокая квалификация специалистов ООО «Кодерлайн» и опыт автоматизации крупных предприятий отражаются в высоких статусах компании. Являются «Центром КОРП» и «Центром компетенций ERP» в торговле, производстве и строительстве.

Организационная структура предприятия представлена на рисунке 1.



Рисунок 1 – Организационная структура предприятия

## Аппаратное и программное обеспечение предприятия

На рабочем месте сотрудника предприятия ООО «Кодерлайн» отдела стажеров и практикантов установлены ноутбуки Maibenben Maibook M. Характеристики комплектующих рабочего ноутбука представлены на рисунке 2.



Рисунок 2 – Аппаратное обеспечение рабочего места сотрудника

Для работы на местах сотрудников установлено следующее программное обеспечение: 1С:Предприятие, Microsoft 365 Copilot.

## Организация обмена данных между сотрудниками

## Использование антивирусного ПО и других средств защиты

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Этапы разработки

Для разработки информационной системы «Плюсик» были установлены этапы, работы каждого этапа, ответственность за выполнение этапов и сроки необходимые на каждый этап. Этапы разработки представлены в таблице 1.

Таблица 1 – Этапы разработки информационной системы

|  |  |  |  |
| --- | --- | --- | --- |
| Этап | Работы этапа | Ответственность | Сроки |
| Определение требований | Выявление и документирование потребностей | Батракова Валерия | 22.03.2025  -  24.03.2025 |
| Анализ существующих систем и выявление проблем | Батракова Валерия |
| Определение функциональных и нефункциональных требований к продукту | Батракова Валерия |
| Проектирование | Разработка архитектуры системы | Батракова Валерия | 25.03.2025  -  27.03.2025 |
| Проектирование модулей, интерфейсов и взаимодействия между ними | Батракова Валерия |
| Определение технологий, инструментов и методологий разработки | Батракова Валерия |
| Разработка макетов прикладного решения | Батракова Валерия |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Этап | Работы этапа | Ответственность | Сроки |
| Разработка | Разработка функционала прикладного решения | Батракова Валерия | 28.03.2025  -  08.04.2025 |
| Реализация системных функций в разрабатываемом прикладном решении | Батракова Валерия |
| Разработка графического интерфейса разрабатываемого прикладного решения | Батракова Валерия |
| Тестирование | Проверка функциональности, производительности, безопасности и других характеристик | Батракова Валерия | 09.04.2025  -  18.04.2025 |
| Выявление и устранение ошибок и дефектов | Батракова Валерия |
| Обеспечение качества и соответствие требованиям | Батракова Валерия |

## Описание предметной области

Информационная система «Плюсик» отвечает за мониторинг успеваемости и учебной активности обучающихся. В базе данных приложения должна храниться информация о студентах, группах, улучшениях, достижениях студентов и бонусной системе.

У каждого студента есть ФИО, дата рождения, личный рейтинг и баланс валюты, которую он может потратить в «магазине» на улучшения.

Рейтинг студента может быть увеличен администратором по фиксированному списку причин или вручную.

Каждый студент является частью группы, у которой есть наименование и дата набора.

В «магазине» хранятся улучшения, у которых есть наименование и стоимость, измеряемая внутренней валютой.

Студенты могут выполнять определенные задания и получать за них достижения в БД, о которых хранятся данные о наименовании, иконке и вознаграждении.

В результате проведенного анализа предметной области базы данных «Плюсик» легко перечислить основные сущности этой БД. Основные таблицы БД представлены на рисунке 3.

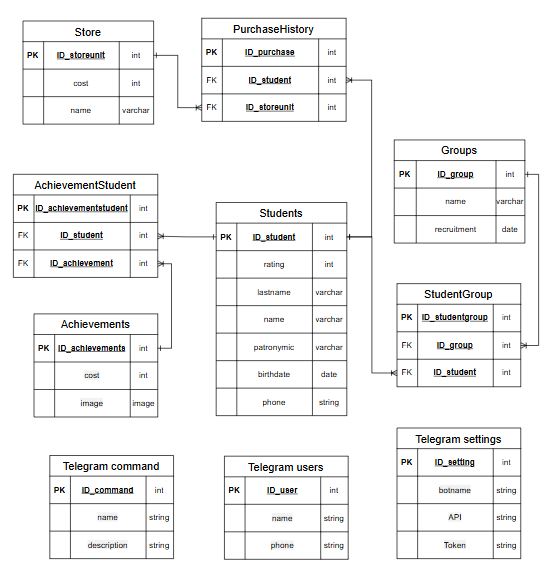


Рисунок 3 - ER-диаграмма

## Создание даталогической модели БД

Даталогическая модель отражается графически в виде схемы базы данных, где указываются имена сущностей, их атрибуты и связи между сущностями.

Даталогическая модель для базы данных «» представлена на рисунке 3.

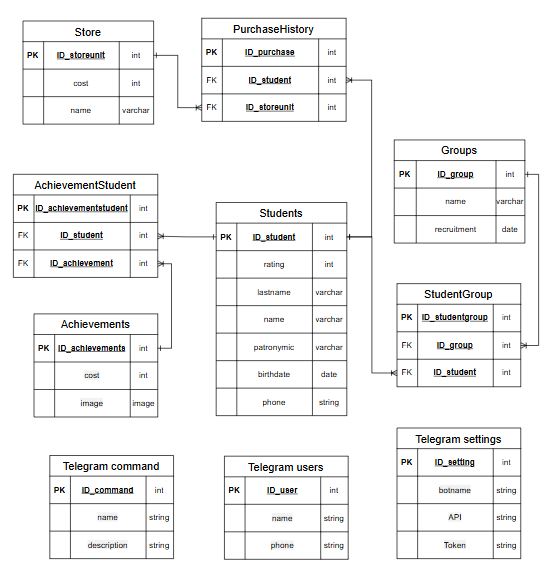


Рисунок 3 – Даталогическая модель

## Выбор программных средств для разработки приложения

Для разработки приложения «Плюсик» была выбрана платформа 1С:Предприятие. Вот некоторые плюсы использования «1С:Предприятие»:

1. Возможность доработки: добавление столбцов в базу данных или создание своих правил составления документов.
2. Автоматизация задач: составление бухгалтерских, налоговых отчётов, расчёт зарплаты, управление закупками и продажами.
3. Гибкость: подходит для разных целей и задач бизнеса, включая бухгалтерский, управленческий и налоговый учёт.
4. Готовые решения: типовые конфигурации для разных сфер бизнеса, например, «1С:Бухгалтерия» и «1С:Документооборот».
5. Разные варианты работы: локальная версия, «1С:Фреш» и облачная 1С.
6. Подключение дополнительных сервисов: отправка отчётности, автоматизация работы с контрагентами и другие функции.
7. Администрирование: разграничение прав доступа, самостоятельное обновление конфигурации и настройка интерфейса.

Для разработки макета приложения была выбрана Figma. Figma — онлайн-редактор, в котором удобно проектировать интерфейсы, создавать макеты сайтов, мобильных приложений, презентации, иллюстрации, логотипы и анимацию. Figma предлагает ряд преимуществ, которые делают его отличным выбором для дизайнеров мобильных приложений. Вот несколько основных причин, почему многие профессионалы выбирают Figma:

* Интуитивный интерфейс и простота использования. Figma обладает интуитивно понятным интерфейсом, что делает его легким в освоении даже для новичков. В нем можно легко создавать и редактировать элементы дизайна, использовать шаблоны и библиотеки, добавлять взаимодействия и анимации, а также экспортировать проекты в различные форматы. Все это делает процесс разработки мобильного приложения более эффективным и продуктивным.
* Плагины — дополнительные приложения внутри редактора, которые расширяют его возможности. Например, плагин Unsplash позволяет добавлять изображения на макет.
* Режим демонстрации работы дизайна приложения для лучшего понимания поведения элементов на странице.
* Высокая производительность и оптимизация работы даже при разработке крупных проектов.

## Создание и заполнение БД

Исходя из проведенного анализа в пункте 2.2 необходимо создать 5 справочников: Студенты, Группы, Достижения, Повышение рейтинга и Команды Telegram результатом их создания и заполнения представлен на рисунках 4 – 7.

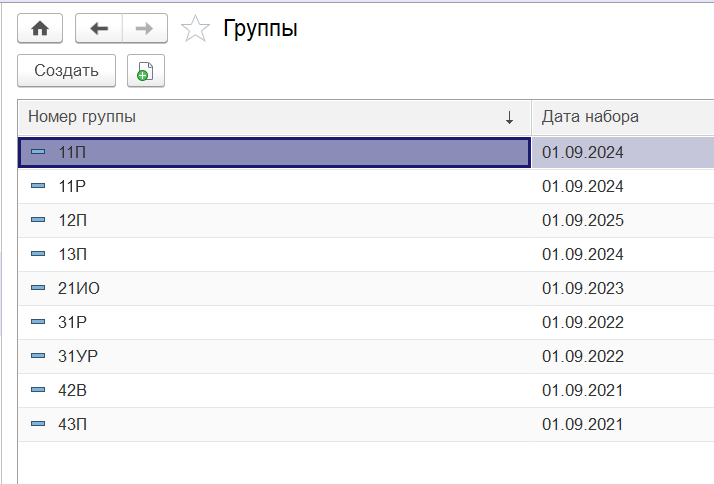


Рисунок 4 – Справочник «Группы»

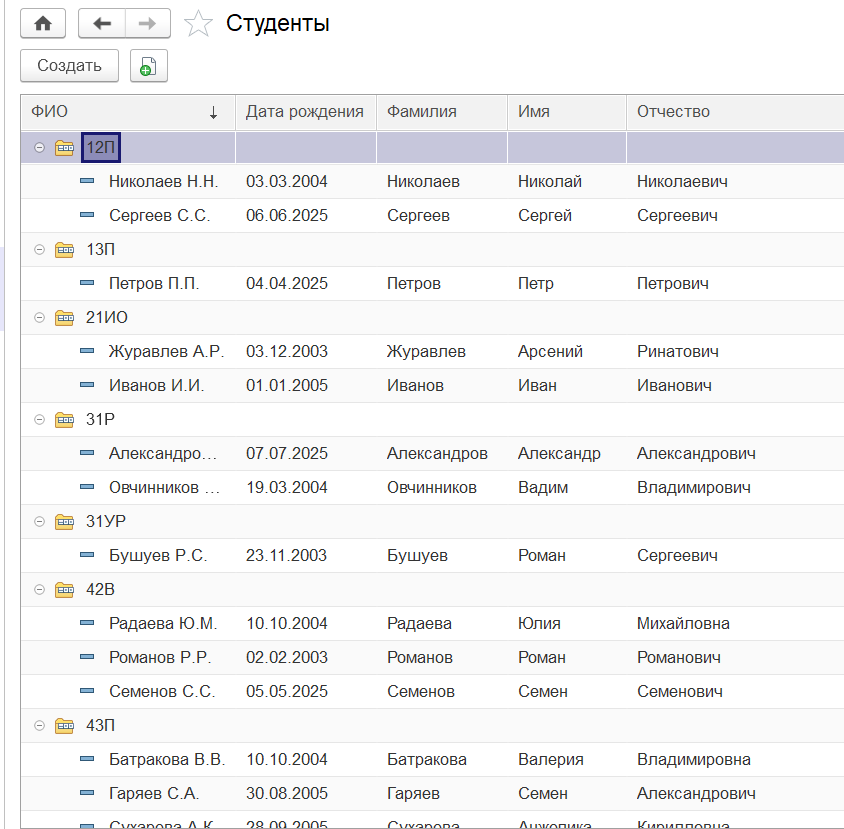


Рисунок 5 – Справочник «Студенты»

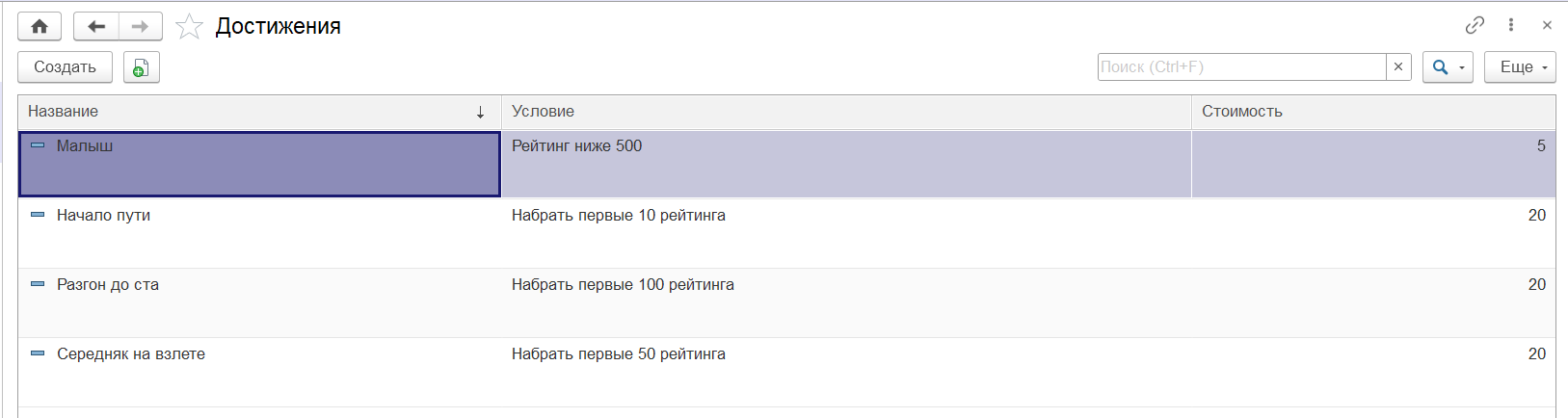


Рисунок 6 – Справочник «Достижения»

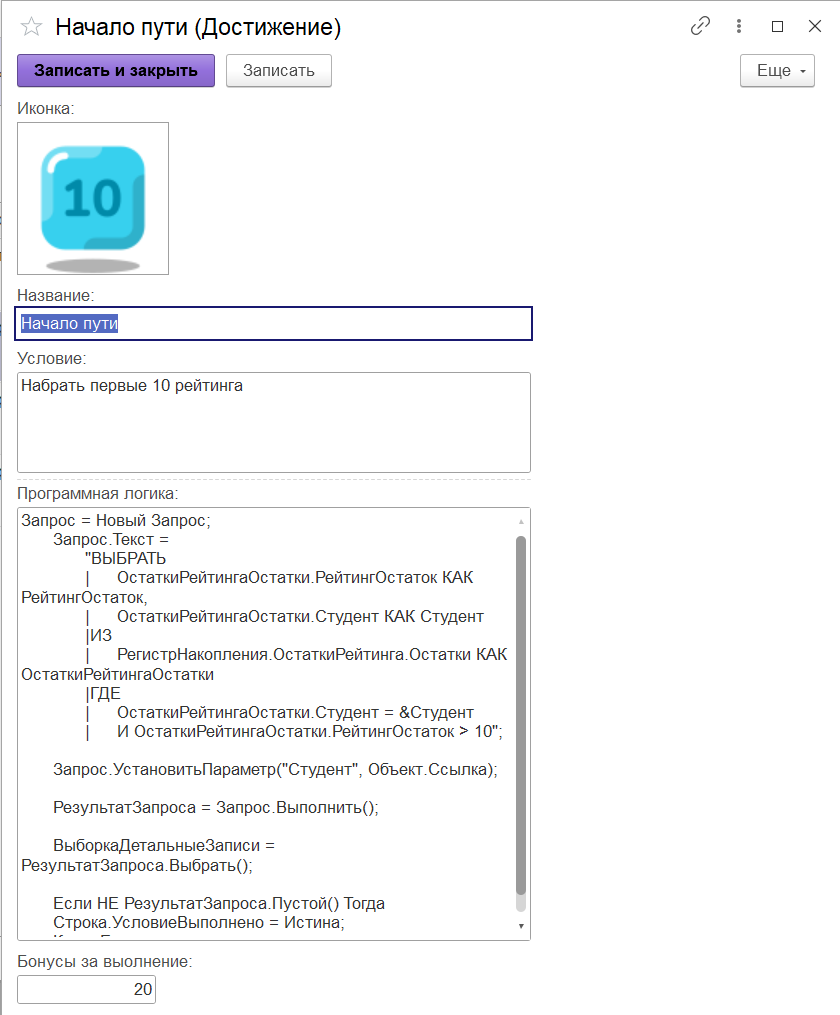


Рисунок 7 – Элемент справочника «Достижения»

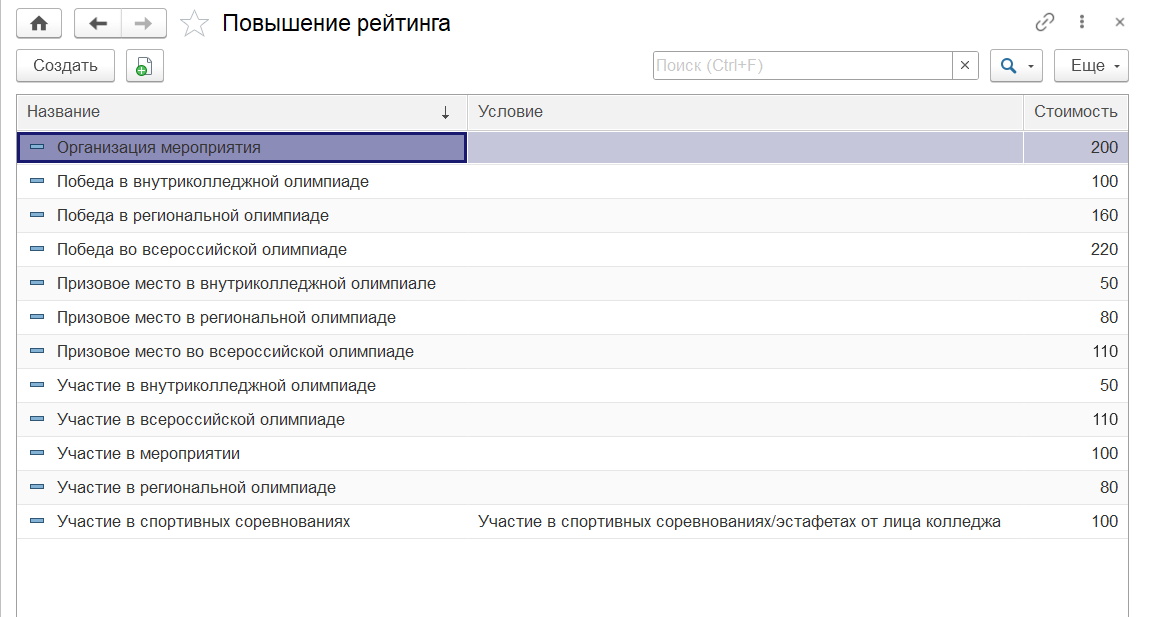


Рисунок 8 – Справочник «Повышение рейтинга»

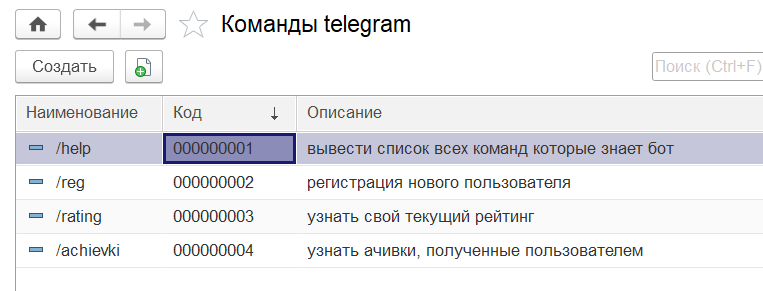


Рисунок 9 – Справочник «Команды Telegram»

## Реализация информационной системы

Согласно проведенному анализу технического задания и возможностей программного обеспечения, было разработано десктопное приложение «Плюсик» на базе 1С:Предприятие, а именно разработаны подсистемы «Telegram», в которой хранятся настройки бота, история сообщений, пользователи и команды, «Бонусная система», в которой находятся инструменты работы с бонусами пользователей – магазин, достижения и т.д., «Рейтинговая система», где осуществляются операции по начислению рейтинга пользователям, и «Учебная часть», где хранится информация о группах и студентах, на главной странице были организованы вкладки «Рейтинг», «Достижения» и «Telegram», на которых информация об обучающихся представлена в наглядном виде и есть инструменты для работы с соответствующими разделами системы, представленные на рисунках 10-17 соответственно.

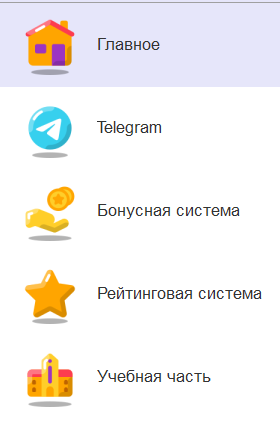


Рисунок 10 – Подсистемы конфигурации

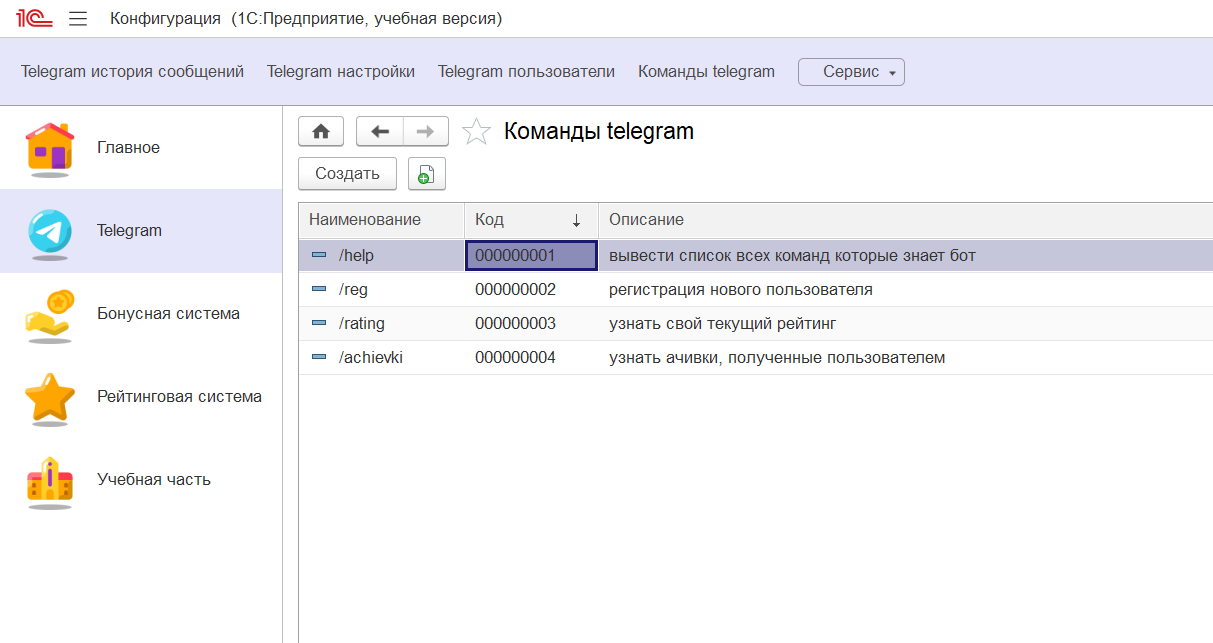


Рисунок 11 – Подсистема «Telegram»

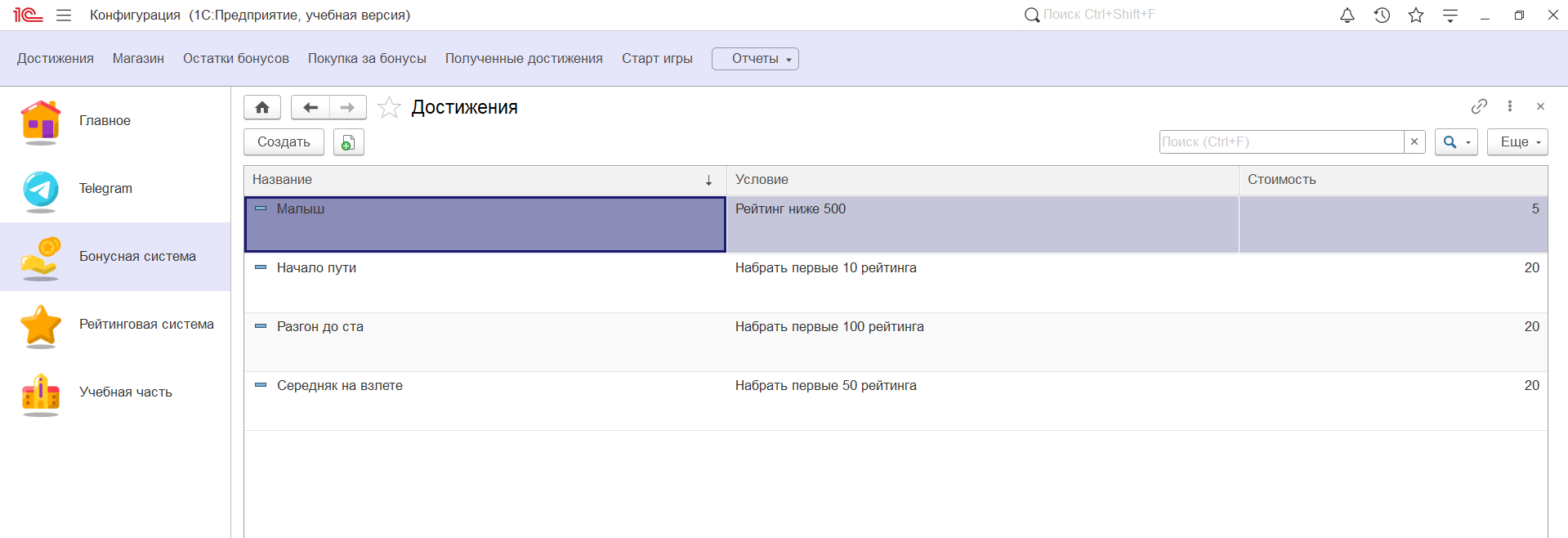


Рисунок 12 – Подсистема «Бонусная система»

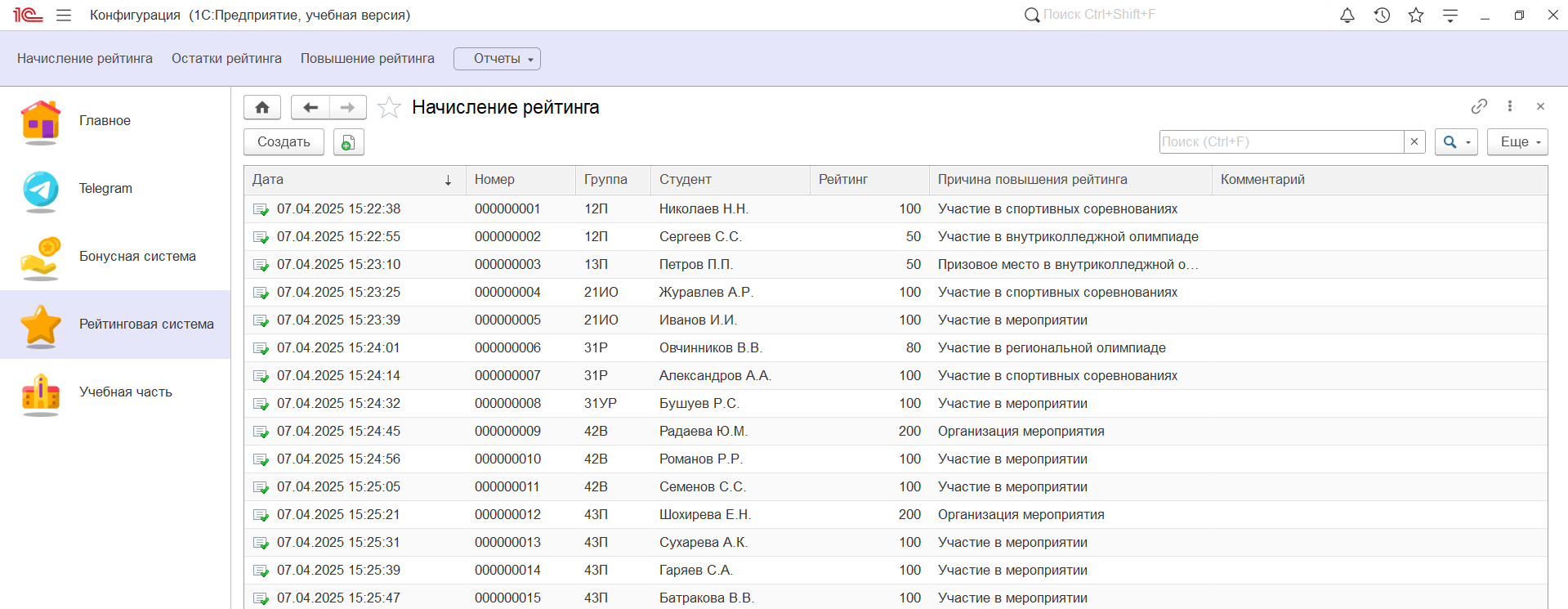


Рисунок 13 – Подсистема «Рейтинговая система»

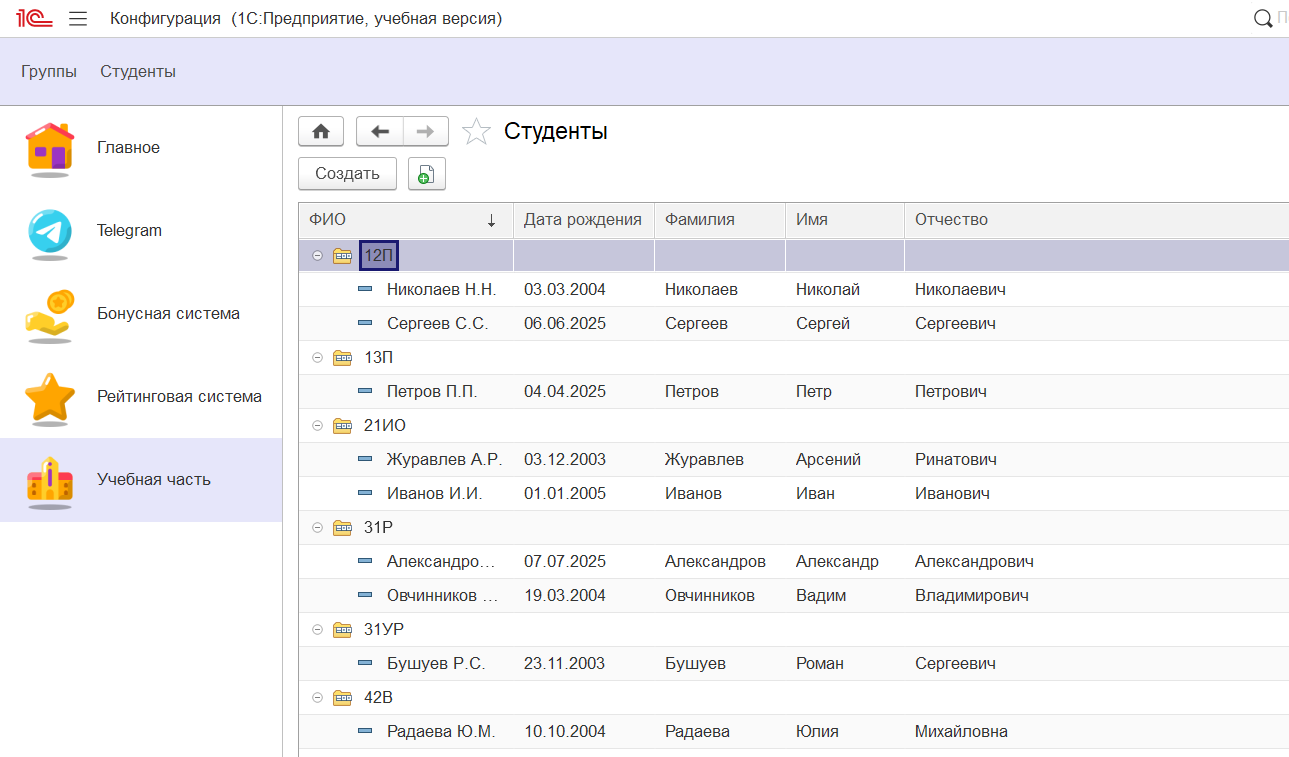


Рисунок 14 – Подсистема «Учебная часть»

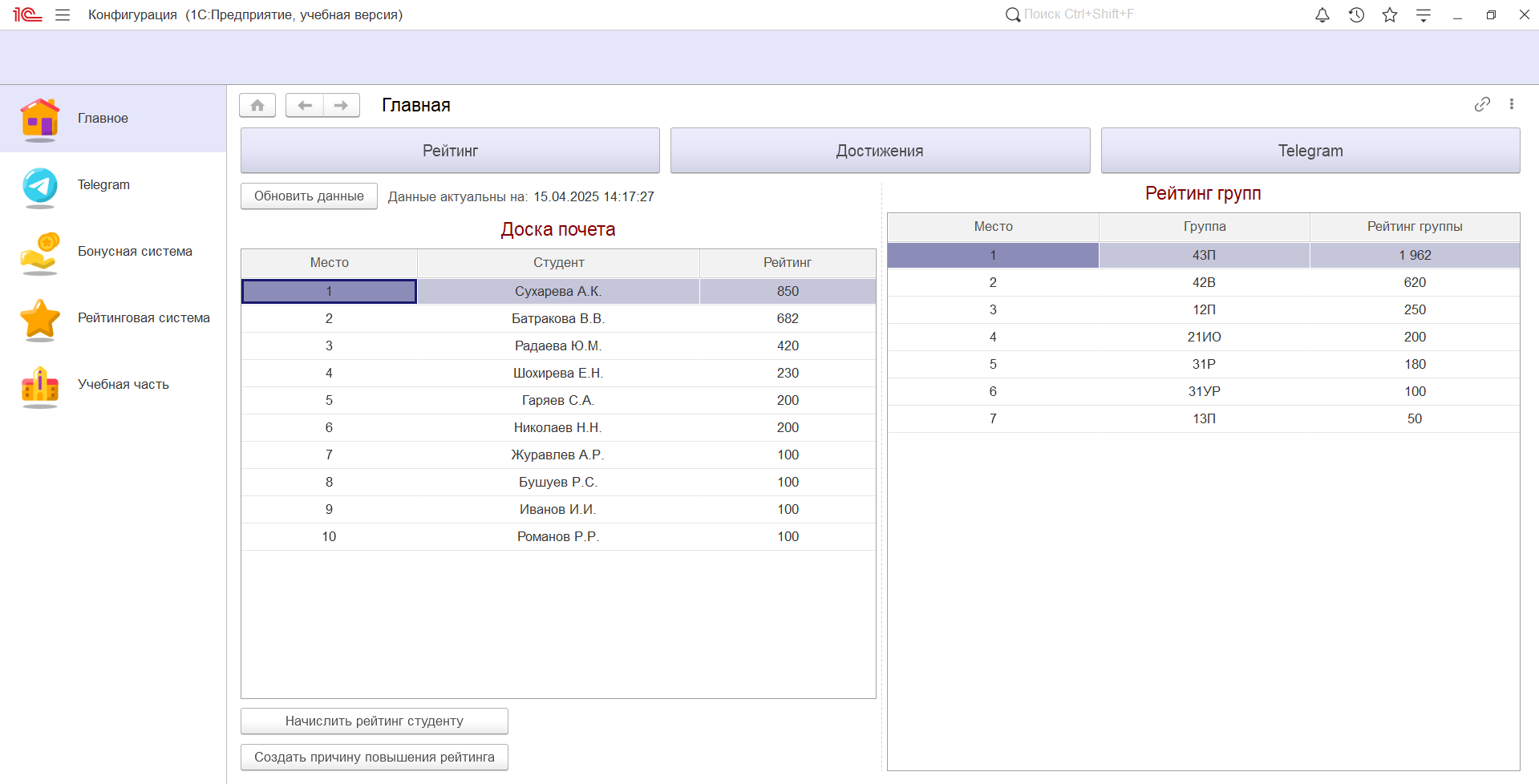


Рисунок 15 – Вкладка «Рейтинг» главной страницы

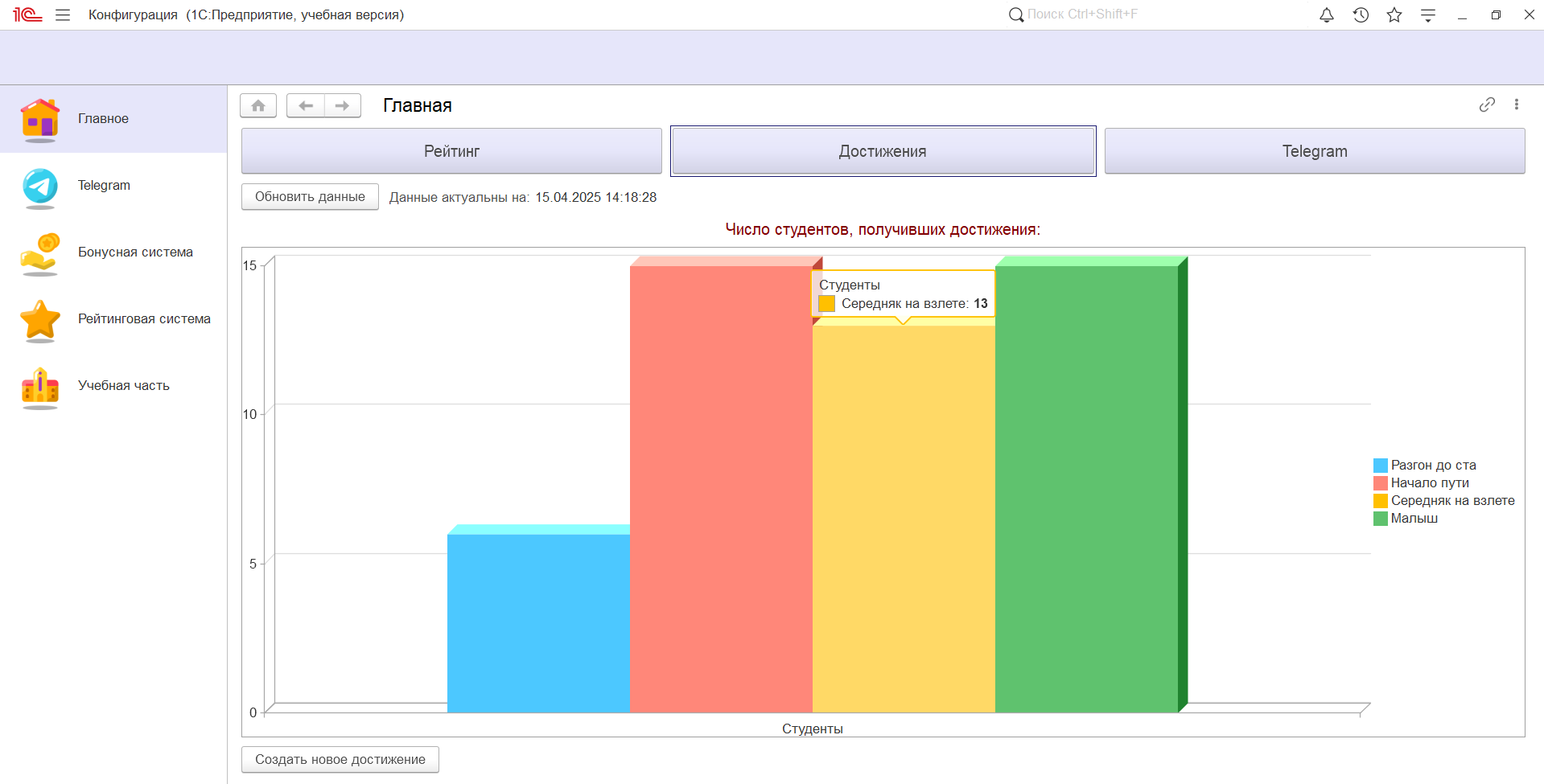


Рисунок 16 – Вкладка «Достижения» главной страницы

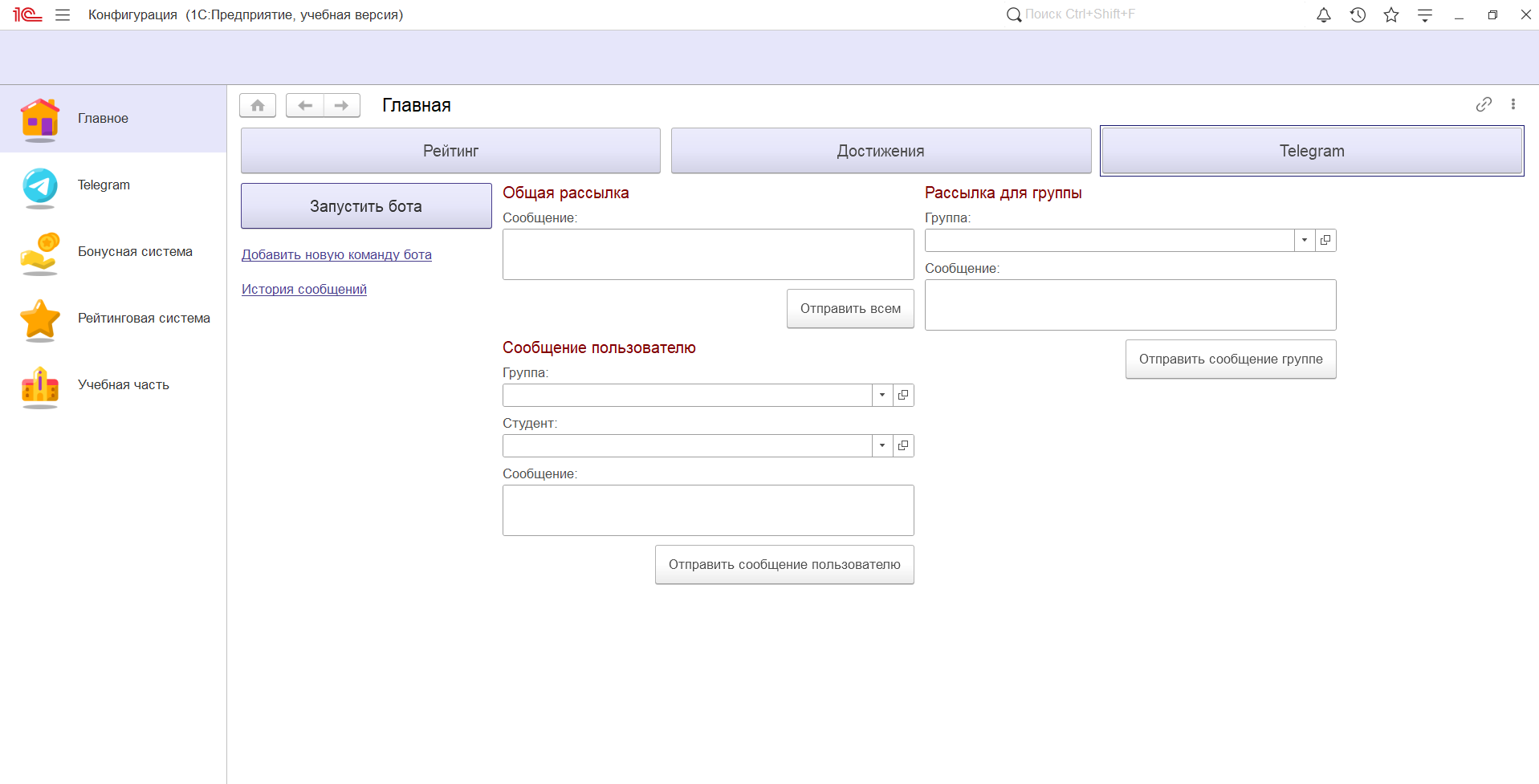


Рисунок 17 – Вкладка «Telegram» главной страницы

Для взаимодействия обучающихся с системой был разработан Telegram-бот «Плюсик», функционал которого представлен на рисунках 18 и 19, а программный код его реализации представлен в приложении А.

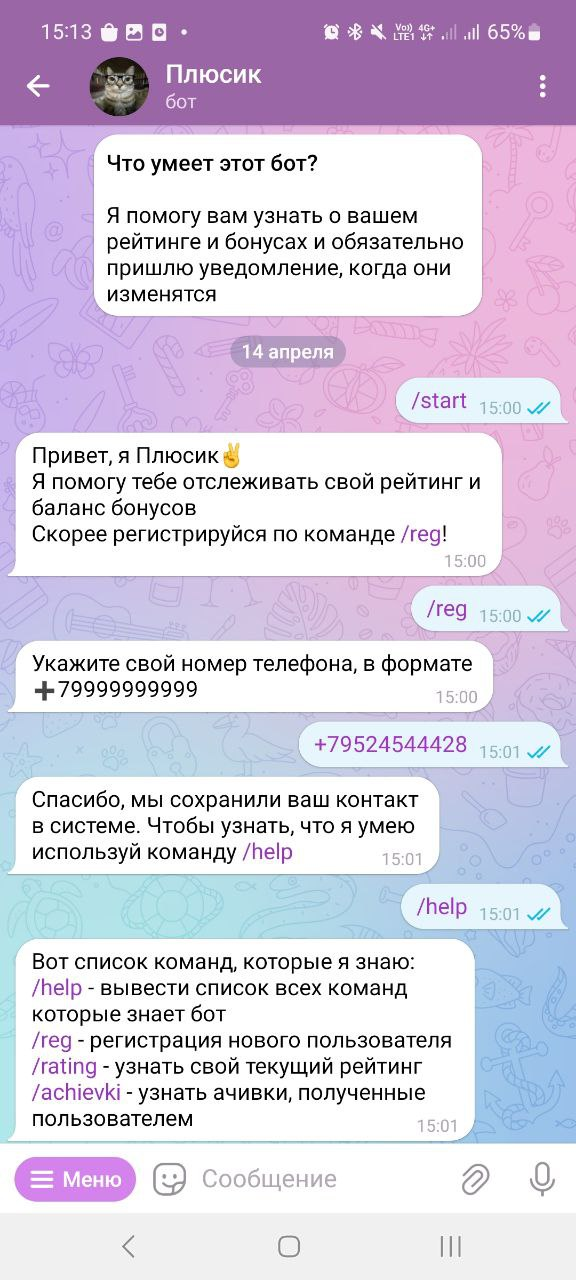


Рисунок 18 – Telegram-бот Плюсик

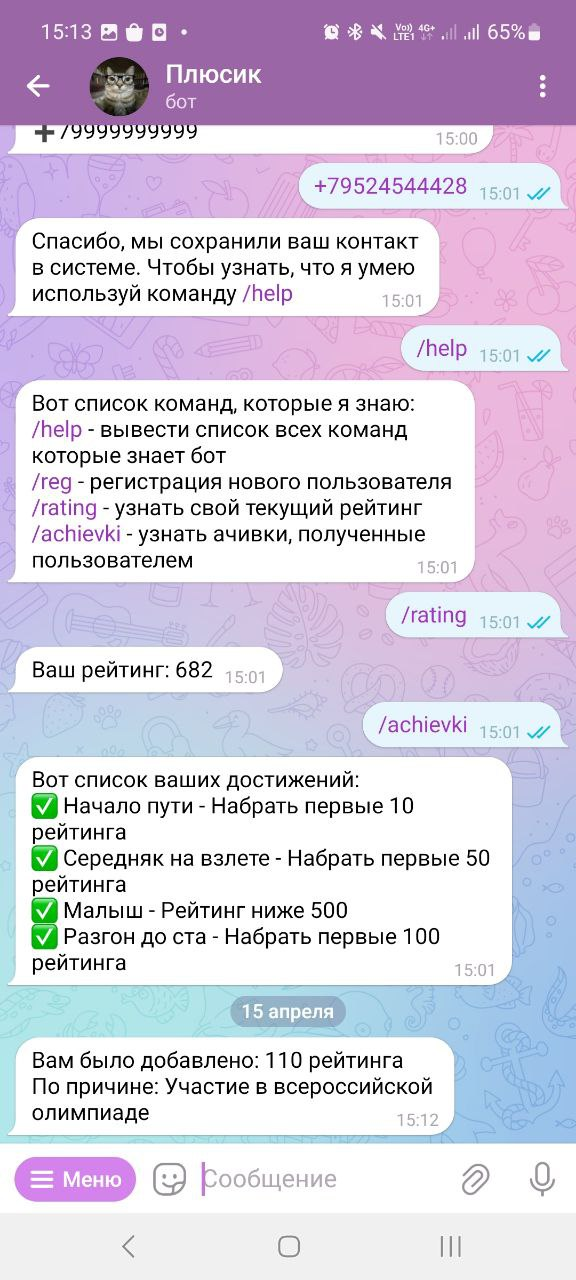


Рисунок 19 – Telegram-бот Плюсик

Для обеспечения соответствия заявленным требованиям были разработаны 5 тест-кейсов для модуля отправки сообщений в Telegram, представленные в приложении В и приложении Б программный код 2 соответственно.

# ЗАКЛЮЧЕНИЕ

В результате прохождения производственной практики по ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» в ООО «Кодерлайн» были успешно выполнены все поставленные задачи. В результате практики были закреплены необходимые профессиональные компетенции, такие как формирование алгоритмов разработки программных модулей, их разработка и отладка, тестирование, рефакторинг и оптимизация программного кода, а также разработка модулей для мобильных платформ.

Практическая работа включала в себя составление технического задания, разработку базы данных, создание пользовательского интерфейса, реализацию функционала приложения, проведение отладки и тестирования, а также рефакторинг и оптимизацию кода. Все эти этапы были выполнены в соответствии с требованиями технического задания.

Практическим результатом прохождения практики стал отчёт, содержащий подробные описания выполненных заданий, а также иллюстрации в виде рисунков, таблиц и диаграмм, которые наглядно демонстрируют процесс разработки и результаты работы.

# ПРИЛОЖЕНИЕ А

Программный код А.1 – Методы Telegram-бота

Процедура ПрочитатьСообщенияПользователя() Экспорт

СтруктураНастроек = ПолучитьНастройкиTelegram();

// Установка соединения

Источник = "bot"+СтруктураНастроек.Token + "/getUpdates" ;

HTTPСоединение = Новый HTTPСоединение( СтруктураНастроек.API,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Источник);

HTTPЗапрос.Заголовки.Вставить("Content-type", "application/json");

HTTPОтвет = HTTPСоединение.Получить(HTTPЗапрос);

Если HTTPОтвет.КодСостояния=200 Тогда

Данные = ДесериализоватьJSON(HTTPОтвет.ПолучитьТелоКакСтроку());

Если Данные.ok И Данные.result.Количество()>0 Тогда

//записываем историю сообщений

Для каждого СтрокаМассива Из Данные.result Цикл

СообщениеID = СтрокаМассива.update\_id;

Если ПроверитьНаличиеСообщенияПоID(СообщениеID) Тогда

Продолжить;

КонецЕсли;

Если СтрокаМассива.Свойство("message") Тогда

//Обработаем и запишем сообщение

Команда = СтрокаМассива.message.text;

ЗаписатьИсториюСообщений(СтрокаМассива.message,СообщениеID,Команда,СтрокаМассива.message.from.id);

ОбработатьОтветПользователя(Команда, СтруктураНастроек, СтрокаМассива.message, СообщениеID);

Иначе

// обработаем кнопки

КонецЕсли;

КонецЦикла;

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Процедура ОбработатьОтветПользователя(Команда, СтруктураНастроек, СтруктураСообщения, СообщениеID)

Отзыв = ЗапросОтзыва(СтруктураСообщения.from.id);

Если Команда = "/reg" Тогда

ОтветСистемы = "Укажите свой номер телефона, в формате ➕79999999999";

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли СтрНайти(Команда, "+7") Тогда

ОтветСистемы = "Спасибо, мы сохранили ваш контакт в системе. Чтобы узнать, что я умею используй команду /help";

Телефон =Команда;

ВернутьСоздатьПользователяTelegram(СтруктураСообщения.chat.first\_name, СтруктураСообщения.from.id , Телефон);

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли Команда = "/rating" Тогда

СписокЗаказов = ПолучитьРейтинг(СтруктураСообщения.from.id);

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + СписокЗаказов;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли Команда = "/achievki" Тогда

СписокЗаказов = ПолучитьДостижения(СтруктураСообщения.from.id);

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + СписокЗаказов;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли Отзыв.ЭтоОтзыв Тогда

ОтветСистемы = "Спасибо за отзыв";

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли Команда = "/start" Тогда

ОтветСистемы = "Привет, я Плюсик✌"+Символы.ПС+"Я помогу тебе отслеживать свой рейтинг и баланс бонусов"+Символы.ПС+"Скорее регистрируйся по команде /reg!";

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

ИначеЕсли Команда = "/help" Тогда

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| КомандыTelegram.Наименование КАК Наименование,

| КомандыTelegram.Описание КАК Описание

|ИЗ

| Справочник.КомандыTelegram КАК КомандыTelegram";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

ОтветСистемы = "Вот список команд, которые я знаю: "+Символы.ПС;

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

ОтветСистемы = ОтветСистемы +ВыборкаДетальныеЗаписи.Наименование+" - "+ВыборкаДетальныеЗаписи.Описание+Символы.ПС;

КонецЦикла;

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

Иначе

ОтветСистемы = "Такой команды нет";

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(СтруктураСообщения.chat.id, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ОтветСистемы;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ОтветСистемы, Команда, СообщениеID, СтруктураСообщения.from.id);

КонецЕсли;

КонецПроцедуры

Функция ЗапросОтзыва(ID)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Telegram\_ИсторияСообщенийСрезПоследних.Сообщение КАК Сообщение,

| Telegram\_ИсторияСообщенийСрезПоследних.Команда КАК Команда

|ИЗ

| РегистрСведений.Telegram\_ИсторияСообщений.СрезПоследних КАК Telegram\_ИсторияСообщенийСрезПоследних

|ГДЕ

| Telegram\_ИсторияСообщенийСрезПоследних.Пользователь = &Пользователь

| И Telegram\_ИсторияСообщенийСрезПоследних.ВидСообщения = &ВидСообщения

| И Telegram\_ИсторияСообщенийСрезПоследних.Команда = ""Отзыв""

|

|УПОРЯДОЧИТЬ ПО

| Telegram\_ИсторияСообщенийСрезПоследних.Период УБЫВ";

Запрос.УстановитьПараметр("ВидСообщения", Перечисления.Telegram\_ВидСообщения.Исходящее);

Запрос.УстановитьПараметр("Пользователь", ID);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.Команда = "Отзыв" Тогда

Структура = Новый Структура;

Структура.Вставить("ЭтоОтзыв", Истина);

Структура.Вставить("НомерЗаказа", ВыборкаДетальныеЗаписи.Сообщение);

Возврат Структура;

КонецЕсли;

КонецЦикла;

Структура = Новый Структура;

Структура.Вставить("ЭтоОтзыв", Ложь);

Возврат Структура;

КонецФункции

#Область ОбщиеФункции

Функция ПолучитьНастройкиTelegram() Экспорт

СтруктураНастроек = Новый Структура;

СтруктураНастроек.Вставить("ИмяБота","");

СтруктураНастроек.Вставить("Token","");

СтруктураНастроек.Вставить("API","");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Telegram\_Настройки.ИмяБота КАК ИмяБота,

| Telegram\_Настройки.API КАК API,

| Telegram\_Настройки.Token КАК Token

|ИЗ

| РегистрСведений.Telegram\_Настройки КАК Telegram\_Настройки";

Выборка = Запрос.Выполнить().Выбрать();

Если Выборка.Количество()>0 Тогда

Выборка.Следующий();

ЗаполнитьЗначенияСвойств( СтруктураНастроек, Выборка);

КонецЕсли;

Возврат СтруктураНастроек;

КонецФункции // ()

Функция ДесериализоватьJSON(СтрокаJSON)

Чтение = Новый ЧтениеJSON;

Чтение.УстановитьСтроку(СтрокаJSON);

Данные = ПрочитатьJSON(Чтение,Ложь);

Чтение.Закрыть();

Возврат Данные;

КонецФункции // ДесериализоватьJSON()

Функция ПроверитьНаличиеСообщенияПоID(IDСообщения)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Telegram\_ИсторияСообщений.Период КАК Период

|ИЗ

| РегистрСведений.Telegram\_ИсторияСообщений КАК Telegram\_ИсторияСообщений

|ГДЕ

| Telegram\_ИсторияСообщений.IDСообщения = &IDСообщения";

Запрос.УстановитьПараметр("IDСообщения", IDСообщения);

Выборка = Запрос.Выполнить().Выбрать();

Если Выборка.Количество()>0 Тогда

Возврат Истина;

Иначе

Возврат Ложь;

КонецЕсли;

КонецФункции // ()

Процедура ЗаписатьИсториюСообщений(СтруктураСообщения,IDСообщения,Команда,IDПользователя)

МенеджерЗаписи = РегистрыСведений.Telegram\_ИсторияСообщений.СоздатьМенеджерЗаписи();

МенеджерЗаписи.Период = ТекущаяДата();

МенеджерЗаписи.Сообщение = СокрЛП(СтруктураСообщения.text);

МенеджерЗаписи.Команда = Команда;

МенеджерЗаписи.Пользователь = IDПользователя;

МенеджерЗаписи.ВидСообщения = Перечисления.Telegram\_ВидСообщения.Входящее;

МенеджерЗаписи.IDСообщения = IDСообщения;

МенеджерЗаписи.Записать(Истина);

КонецПроцедуры

Процедура ЗаписатьИсториюОтветов(ОтветСистемы, Команда, IDСообщения, IDПользователя) Экспорт

МенеджерЗаписи = РегистрыСведений.Telegram\_ИсторияСообщений.СоздатьМенеджерЗаписи();

МенеджерЗаписи.Период = ТекущаяДата();

МенеджерЗаписи.Сообщение = ОтветСистемы;

МенеджерЗаписи.Команда = Команда;

МенеджерЗаписи.Пользователь = IDПользователя;

МенеджерЗаписи.ВидСообщения = Перечисления.Telegram\_ВидСообщения.Исходящее;

МенеджерЗаписи.IDСообщения = IDСообщения;

МенеджерЗаписи.Записать(Истина);

КонецПроцедуры

Функция ВернутьСоздатьПользователяTelegram(ИмяПользователя, IDПользователя, Телефон = Неопределено)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Telegram\_Пользователи.Телефон КАК Телефон

|ИЗ

| РегистрСведений.Telegram\_Пользователи КАК Telegram\_Пользователи

|ГДЕ

| Telegram\_Пользователи.IDПользователя = &IDПользователя";

Запрос.УстановитьПараметр("IDПользователя", IDПользователя);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

МенеджерЗаписи = РегистрыСведений.Telegram\_Пользователи.СоздатьМенеджерЗаписи();

МенеджерЗаписи.ИмяПользователя = ИмяПользователя;

МенеджерЗаписи.Телефон = Телефон;

МенеджерЗаписи.IDПользователя = IDПользователя;

МенеджерЗаписи.Записать(ИСТИНА);

Возврат Телефон;

КонецЦикла;

МенеджерЗаписи = РегистрыСведений.Telegram\_Пользователи.СоздатьМенеджерЗаписи();

МенеджерЗаписи.ИмяПользователя = ИмяПользователя;

МенеджерЗаписи.Телефон = Телефон;

МенеджерЗаписи.IDПользователя = IDПользователя;

МенеджерЗаписи.Записать(ИСТИНА);

Возврат Телефон;

КонецФункции

Функция ПолучитьРейтинг(ID)

Сообщение = "";

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОстаткиРейтингаОстатки.РейтингОстаток КАК РейтингОстаток,

| Telegram\_Пользователи.Телефон КАК Телефон,

| ОстаткиРейтингаОстатки.Студент.Телефон КАК СтудентТелефон

|ИЗ

| РегистрСведений.Telegram\_Пользователи КАК Telegram\_Пользователи

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиРейтинга.Остатки КАК ОстаткиРейтингаОстатки

| ПО Telegram\_Пользователи.Телефон = ОстаткиРейтингаОстатки.Студент.Телефон

|ГДЕ

| Telegram\_Пользователи.IDПользователя = &IDПользователя";

Запрос.УстановитьПараметр("IDПользователя", ID);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Сообщение = Сообщение + "Ваш рейтинг: " + ВыборкаДетальныеЗаписи.РейтингОстаток + Символы.ПС;

КонецЦикла;

Сообщение = СокрЛП(Сообщение);

Если Сообщение = "" Тогда

Сообщение = "Вы еще не начали накапливать рейтинг";

КонецЕсли;

Возврат Сообщение;

КонецФункции

Функция ПолучитьДостижения(ID)

Сообщение = "";

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Telegram\_Пользователи.Телефон КАК Телефон,

| Telegram\_Пользователи.IDПользователя КАК IDПользователя,

| ПолученныеДостижения.Студент.Телефон КАК СтудентТелефон,

| ПолученныеДостижения.Достижение.Код КАК ДостижениеКод,

| ПолученныеДостижения.Достижение.Условие КАК ДостижениеУсловие

|ИЗ

| РегистрСведений.Telegram\_Пользователи КАК Telegram\_Пользователи

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ПолученныеДостижения КАК ПолученныеДостижения

| ПО Telegram\_Пользователи.Телефон = ПолученныеДостижения.Студент.Телефон

|ГДЕ

| Telegram\_Пользователи.IDПользователя = &IDПользователя";

Запрос.УстановитьПараметр("IDПользователя", ID);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Сообщение = "Вот список ваших достижений: "+Символы.ПС;

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Сообщение = Сообщение +"✅ "+ ВыборкаДетальныеЗаписи.ДостижениеКод + " - " + ВыборкаДетальныеЗаписи.ДостижениеУсловие+Символы.ПС;

КонецЦикла;

Сообщение = СокрЛП(Сообщение);

Если Сообщение = "" Тогда

Сообщение = "У вас еще нет достижений(";

КонецЕсли;

Возврат Сообщение;

КонецФункции // ()

Процедура ОтправитьСообщение(IDПользователя, ТекстСообщения) Экспорт

СтруктураНастроек = ПолучитьНастройкиTelegram();

Приемник = "bot" + СтруктураНастроек.Token + "/sendMessage?chat\_id=" + СтрЗаменить(Формат(IDПользователя, "ЧДЦ=; ЧС=; ЧРГ=."), ".", "") + "&text=" + ТекстСообщения;

HTTPСоединение = Новый HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Приемник);

Ответ = HTTPСоединение.Получить(HTTPЗапрос);

ЗаписатьИсториюОтветов(ТекстСообщения, "Оповещение", "", IDПользователя);

КонецПроцедуры

Процедура ЗапускБотов() Экспорт

ПрочитатьСообщенияПользователя();

КонецПроцедуры

#КонецОбласти

# ПРИЛОЖЕНИЕ Б

Таблица Б.1 - Аннотация теста

|  |  |
| --- | --- |
| **Название проекта** | PP2 |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Батракова Валерия |
| **Дата(ы) теста** | **11.04.2025** |

Таблица Б.2 – Тест-кейс 1

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Отправка сообщения пользователю |
| **Краткое изложение теста** | Администратор отправляет сообщение зарегистрированному в боте студенту |
| **Этапы теста** | 1. Открыть форму отправки сообщений в Telegram. 2. Заполнить поля: группа, студент и сообщение. 3. Нажать кнопку "Отправить". |
| **Тестовые данные** | Группа 43П, студент Батракова В.В., сообщение «тест» |
| **Ожидаемый результат** | Сообщение успешно доставлено пользователю |
| **Фактический результат** | Сообщение успешно доставлено пользователю |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица Б.3 – Тест-кейс 2

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Отправление сообщения пользователю, незарегистрированного в боте |
| **Краткое изложение теста** | Администратор пытается отправить сообщение студенту, который не является пользователем бота |
| **Этапы теста** | 1. Открыть форму отправки сообщений в Telegram. 2. Заполнить поля: группа, студент и сообщение. 3. Нажать кнопку "Отправить". |
| **Тестовые данные** | Группа 31Р, студент Овчинников В.В., сообщение «тест» |
| **Ожидаемый результат** | Система выдает ошибку, сообщающую, что данный студент не является пользователем бота и сообщение отправить ему невозможно |
| **Фактический результат** | Система выдает ошибку, сообщающую, что данный студент не является пользователем бота и сообщение отправить ему невозможно |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица Б.4 – Тест-кейс 3

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Отправка пустого сообщения |
| **Краткое изложение теста** | Администратор пытается отправить сообщение студенту, не заполнив поле «сообщение». |
| **Этапы теста** | 1. Открыть форму отправки сообщений в Telegram. 2. Заполнить поля: группа, студент, оставив поле сообщение незаполненным. 3. Нажать кнопку "Отправить". |
| **Тестовые данные** | Группа 43П, студент Батракова В.В. |
| **Ожидаемый результат** | Сообщение не отправляется, система выдает предупреждение о том, что отправка пустого сообщения невозможна. |
| **Фактический результат** | Сообщение не отправляется, система выдает предупреждение о том, что отправка пустого сообщения невозможна. |
| **Статус** | Зачет |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица Б.5 – Тест-кейс 4

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Отправка медиафайлов |
| **Краткое изложение теста** | Администратор пытается отправить несколько сообщений студенту подряд |
| **Этапы теста** | 1. Открыть форму отправки сообщений в Telegram. 2. Заполнить поля: группа, студент, вставив в поле сообщение изображение 3. Нажать кнопку "Отправить". |
| **Тестовые данные** | Группа 43П, студент Батракова В.В., сообщение с изображением |
| **Ожидаемый результат** | Сообщение успешно доставлено пользователю |
| **Фактический результат** | Сообщение не отправляется, предупреждение не возникает |
| **Статус** | Незачёт |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица Б.6 – Тест-кейс 5

|  |  |
| --- | --- |
| **Test Case #** | TC\_UI\_5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Отправка нескольких сообщений подрад |
| **Краткое изложение теста** | Администратор не выбирает категорию товара. |
| **Этапы теста** | 1. Открыть форму отправки сообщений в Telegram. 2. Заполнить поля: группа, студент и сообщение. 3. Нажать кнопку "Отправить". 4. Написать еще одно сообщение и отправить пользователю 5. Нажать кнопку «Отправить» |
| **Тестовые данные** | Группа 43П, студент Батракова В.В., сообщение «тест», второе сообщение «тест 2» |
| **Ожидаемый результат** | Сообщения успешно доставлены пользователю |
| **Фактический результат** | После отправки первого сообщения поля группа и студент сбрасываются, отправить новые сообщения этому же пользователю без повторного заполнения формы невозможно. |
| **Статус** | Незачёт |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |