

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ	4
1.1 Анализ исходных требований.....	4
1.2 Описание предметной области	5
1.3 Сравнительный анализ аналогов и прототипов	5
1.4 Используемые библиотеки и фреймворки	5
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6
2.1 Функциональная структура административной части.....	6
2.2 Описание структуры данных	6
2.3 Безопасность и защита данных.....	8
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА	9
3.1 Реализация административной части.....	9
3.2 Реализация пользовательской части	10
3.3 Тестирование	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

На данный момент все больше и больше образовательных заведений в разных странах используют современные технологии, в том числе чат-боты для упрощения образовательных процессов. Согласно опросам, большой процент учащихся и сотрудников удовлетворены тем, что имеют возможность взаимодействовать с различными программами и уверены в том, что эти программы улучшают и облегчают их жизнь.

Чат-боты используются в различных сферах, но мы рассмотрим их непосредственное применение и пользу в сфере образования. Педагоги образовательных учреждений имеют возможность быстро устанавливать контакт с учащимися и могут передавать информацию в компактном виде. Важным аспектом является то, что боты работают по сценарию и могут автоматизировано давать ответы на популярные вопросы.

Чат-боты упрощают жизнь и преподавателей, и студентов, предоставляя удобные инструменты для решения повседневных заданий. Боты, несомненно, упрощают нашу жизнь, использование интеллектуальных систем обучения делают их безгранично перспективной технологией.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

1.1 Анализ исходных требований

Telegram-бота по рассылке расписания и оповещений через мессенджер является темой магистерской диссертации.

Данный проект представляет собой серверную часть.

К основным функциям Telegram-бота можно отнести:

- Регистрация и аутентификация.
- Получение своего расписания на сегодня и завтра.
- Получение расписания определенной группы на определенный день.
- Получение расписания определенного преподавателя на определенный день.
- Совершение рассылки по определенным группам людей.
- Отправка информации о посещении пар, а также хранение и запись пропусков в онлайн-таблицу.
- Возможность подписки на рассылку с расписанием.

Общие требования к приложению:

1. Система хранения данных – для системы хранения необходимо использовать одну из современных реляционных СУБД.
2. Надежность – вся информация должна правильно редактироваться, к тому же действия обязаны протекать в логичном русле.
3. Адаптивность – архитектура приложения должна быть легко изменяемая.
4. Безопасность – защищенность данных должна быть на высоком уровне и доступ к ним должен строго контролироваться.
5. Мультипользовательность – приложение должно поддерживать работоспособность даже при большой нагрузке клиентами.
6. Комфорт использования – структура приложения должна быть понятна пользователю.

На основании вышесказанного сделано постановка задач для проектирования:

- Реализация правил и ограничений автоматизируемых операций.
- Разработка и проектирование реляционной СУБД.
- Разработка парсера для занесения расписания в СУБД.

1.2 Описание предметной области

Данного Telegram-бота можно разделить на несколько ролей:

- Администратор.
- Преподаватель.
- Студент.
- Студент-староста.

Техническое обслуживание и администрирование Telegram-бота является ключевой задачей администратора. А также разработка нового функционала и усовершенствование имеющегося.

Преподаватель должен иметь возможность отправки оповещений, получения своего расписания, а также запроса расписания других преподавателей и студенческих групп.

Студент должен иметь возможность получения своего расписания, а также запроса расписания других преподавателей и студенческих групп.

Студент-староста должен иметь возможность получения своего расписания, запроса расписания других преподавателей и студенческих групп, а также возможность отправки данных о посещении пар.

1.3 Сравнительный анализ аналогов и прототипов

В связи с проектированием Telegram-бота для определенных нужд, рассмотрение аналогов производиться не будет.

1.4 Используемые библиотеки и фреймворки

Для реализации Telegram-бота будем использовать язык программирования Python. Для взаимодействия с Telegram будет применяться библиотека aiogram. Для создания и взаимодействия с СУБД будет одна из следующих библиотек в зависимости от базы данных pyodbc (PosgreSQL, MSSQL) или sqlite3 (SQLite).

Для административной части будет использоваться C#, а также платформа разработки Windows Forms. Для взаимодействия с СУБД — Entity Framework.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Функциональная структура административной части

Разрабатываемая административная часть предназначена для администрирования и модификации данных. Данная часть предоставляет администратору возможность вставки, модификации и удаления необходимых данных в базе.

Система администратора создается на Windows Forms и работает независимо от Telegram-бота. Для проектирования необходимо четко понимать какая информация нуждается в редактировании. Установлено, что в нашем случае это информация о расписании.

Принцип работы бота схематически отражен на рисунке 2.1.

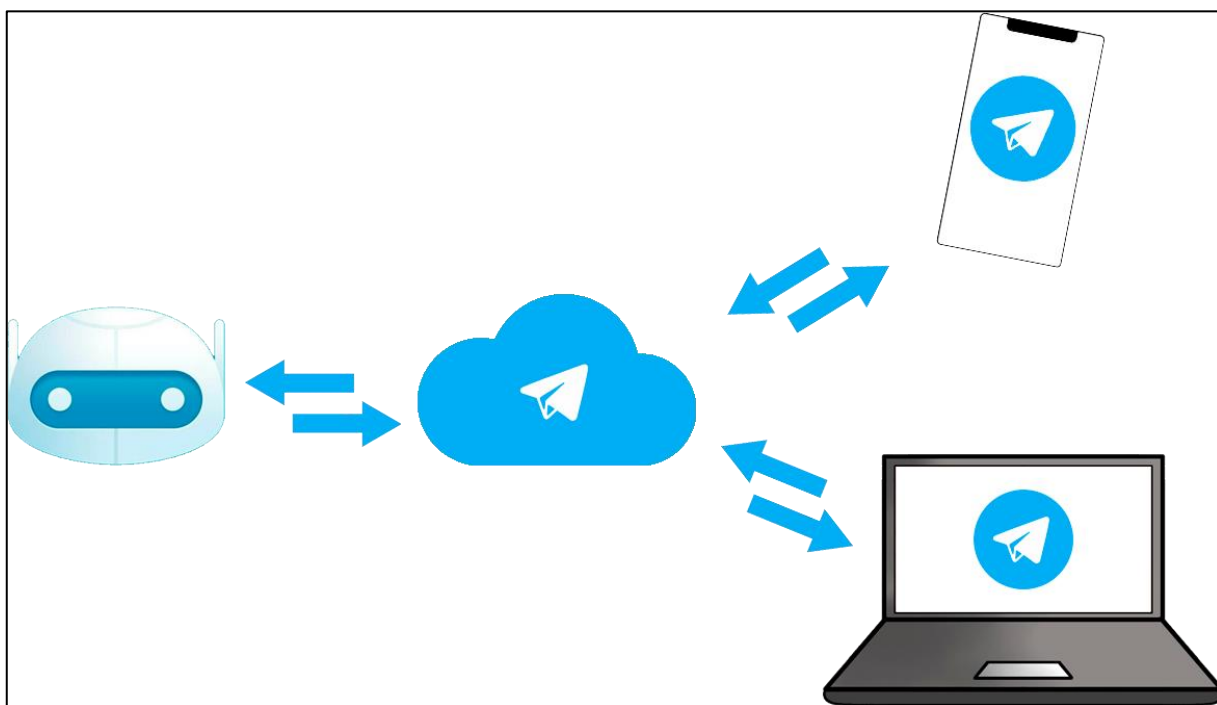


Рисунок 2.1 – Схема работы

2.2 Описание структуры данных

Для того, чтоб хранить информацию требуется разработать СУБД. Для реализации была выбрана база данных MSSQL.

Реляционная база данных — это набор данных с предопределенными связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке — значение атрибута.

Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей. К этим данным можно получить доступ многими способами, и при этом реорганизовывать таблицы БД не требуется.

Особое внимание к транзакциям — это важная часть нашей СУБД, потому что многопользовательская сторона диктует данное требование.

Рисунок 2.1 представляет собой схему базы данных.

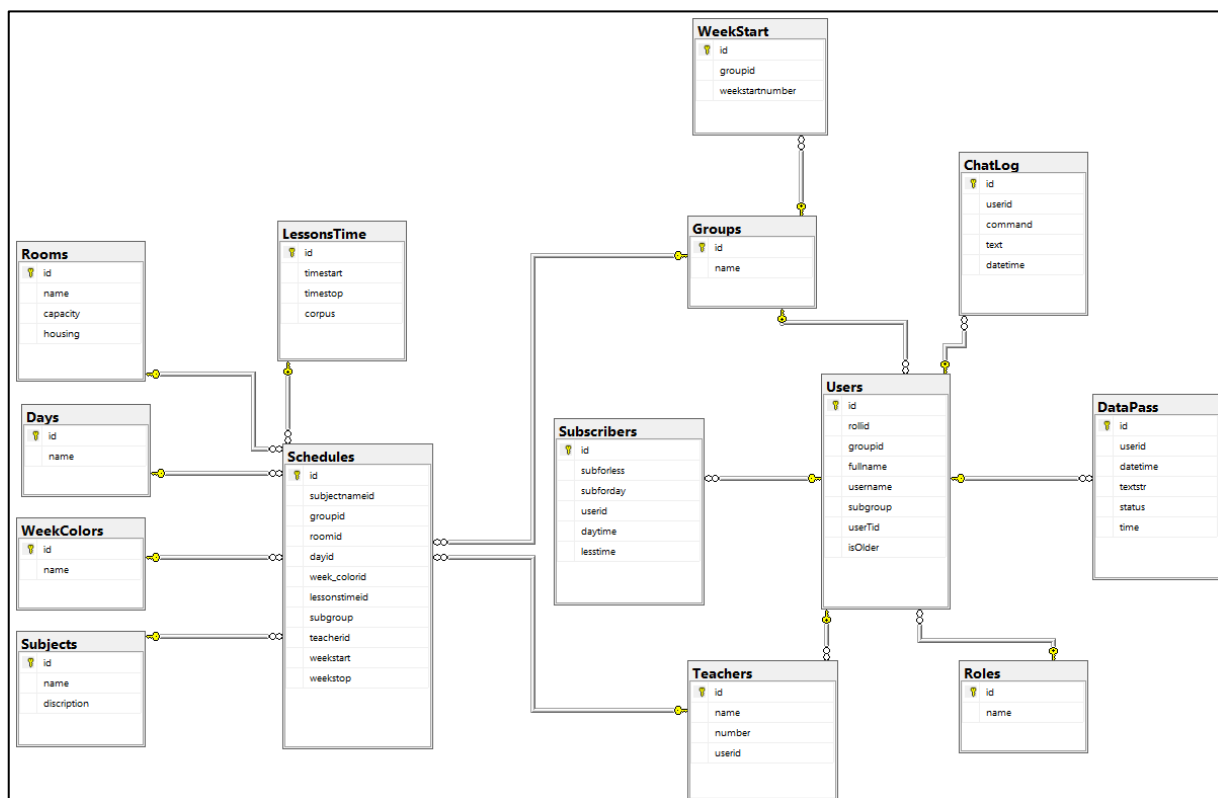


Рисунок 2.1 – Схема базы данных

Для того чтоб иметь эффективный и инновационный метод доступа к данным в административной части будет применяться Entity Framework. Целью Entity является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования при работе в объектно-ориентированных средствах в реляционной базе данных. Разработчик может использовать Entity как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой данных.

Для работы с базой данных со стороны Telegram-бота была использована библиотека `pyodbc`. `pyodbc` — это модуль Python с открытым исходным кодом, который упрощает доступ к базам данных ODBC. Однако по сравнению с Entity Framework в библиотеке `pyodbc` нам необходимо использовать низкоуровневое программирование и писать все запросы к базе данных вручную.

2.3 Безопасность и защита данных

Безопасность данных зиждется на следующих постулатах:

- Данные должны быть устойчивы к изменению извне неавторизованными пользователями. Система должна быть устойчива к различным атакам. Для того чтобы сохранять копии данных используется резервное копирование.
- Доступ к данным должен предоставляться согласно аутентификации пользователей по ролям.

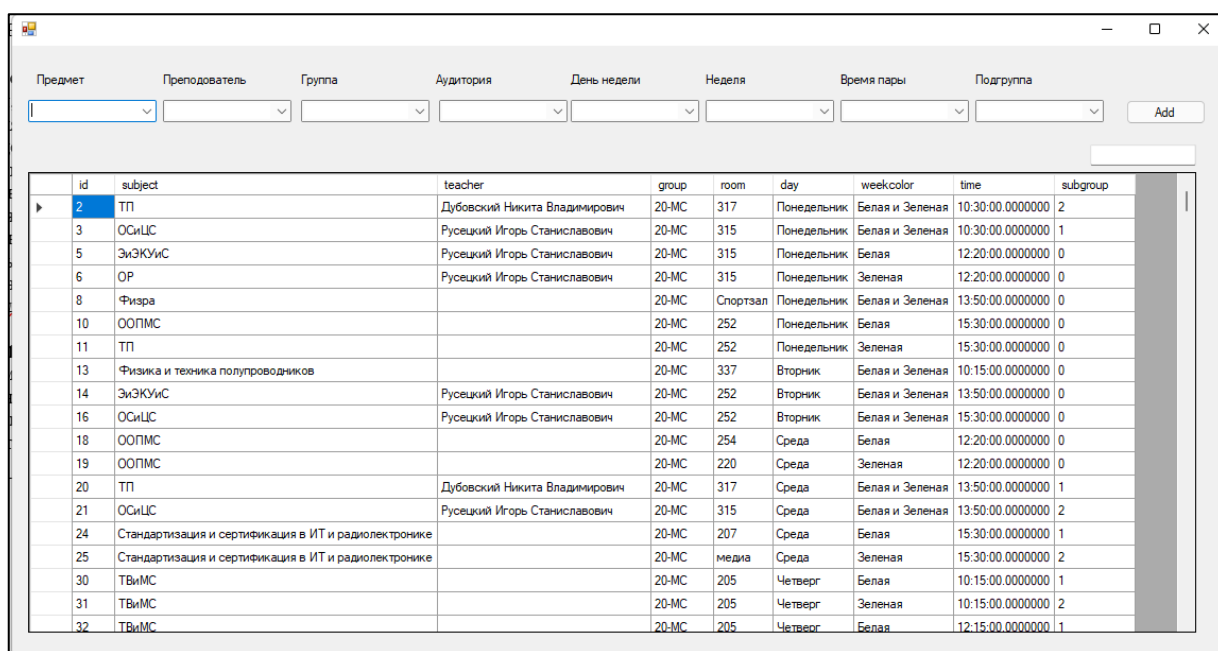
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

3.1 Реализация административной части

В рамках магистерской диссертации рассматриваемого приложения реализация происходит соответственно заявленным требованиям к проектированию и разработке.

Ниже представлены результаты реализации административной части. Было разработано главное и единственное окно, в котором администратор может наблюдать данные, которые были асинхронно подгружены с сервера в приложение. При изменении данных их обновление происходит динамически.

При проектировании и реализации административной части был использована платформа графического интерфейса Windows Forms. Окно административной части отображено на рисунке 3.1.



The screenshot shows a Windows Forms application window titled 'Административная часть'. At the top, there are several dropdown menus for filtering data: 'Предмет' (Subject), 'Преподаватель' (Teacher), 'Группа' (Group), 'Аудитория' (Auditorium), 'День недели' (Day of the week), 'Неделя' (Week), 'Время пары' (Time), and 'Подгруппа' (Subgroup). Below these filters is a table with the following columns: id, subject, teacher, group, room, day, weekcolor, time, and subgroup. The table contains 20 rows of data, including subjects like 'ТП', 'ОСиЦС', 'ЭиЭКУиС', 'ОР', 'Физра', 'ООПМС', 'Физика и техника полупроводников', 'Стандартизация и сертификация в ИТ и радиоэлектронике', and 'ТВыМС'.

id	subject	teacher	group	room	day	weekcolor	time	subgroup
2	ТП	Дубовский Никита Владимирович	20-МС	317	Понедельник	Белая и Зеленая	10:30:00.0000000	2
3	ОСиЦС	Русецкий Игорь Станиславович	20-МС	315	Понедельник	Белая и Зеленая	10:30:00.0000000	1
5	ЭиЭКУиС	Русецкий Игорь Станиславович	20-МС	315	Понедельник	Белая	12:20:00.0000000	0
6	ОР	Русецкий Игорь Станиславович	20-МС	315	Понедельник	Зеленая	12:20:00.0000000	0
8	Физра		20-МС	Спортзал	Понедельник	Белая и Зеленая	13:50:00.0000000	0
10	ООПМС		20-МС	252	Понедельник	Белая	15:30:00.0000000	0
11	ТП		20-МС	252	Понедельник	Зеленая	15:30:00.0000000	0
13	Физика и техника полупроводников		20-МС	337	Вторник	Белая и Зеленая	10:15:00.0000000	0
14	ЭиЭКУиС	Русецкий Игорь Станиславович	20-МС	252	Вторник	Белая и Зеленая	13:50:00.0000000	0
16	ОСиЦС	Русецкий Игорь Станиславович	20-МС	252	Вторник	Белая и Зеленая	15:30:00.0000000	0
18	ООПМС		20-МС	254	Среда	Белая	12:20:00.0000000	0
19	ООПМС		20-МС	220	Среда	Зеленая	12:20:00.0000000	0
20	ТП	Дубовский Никита Владимирович	20-МС	317	Среда	Белая и Зеленая	13:50:00.0000000	1
21	ОСиЦС	Русецкий Игорь Станиславович	20-МС	315	Среда	Белая и Зеленая	13:50:00.0000000	2
24	Стандартизация и сертификация в ИТ и радиоэлектронике		20-МС	207	Среда	Белая	15:30:00.0000000	1
25	Стандартизация и сертификация в ИТ и радиоэлектронике		20-МС	медиа	Среда	Зеленая	15:30:00.0000000	2
30	ТВыМС		20-МС	205	Четверг	Белая	10:15:00.0000000	1
31	ТВыМС		20-МС	205	Четверг	Зеленая	10:15:00.0000000	2
32	ТВыМС		20-МС	205	Четверг	Белая	12:15:00.0000000	1

Рисунок 3.1 – Окно административной части

А также часть возможностей была вынесена в самого Telegram-бота. Таким образом у администратора появилась возможность задавать роль вновь зарегистрированному пользователю, совершать рассылку и перенос данных из СУБД в онлайн-таблицу, что отображено на рисунке 3.2 – 3.3.

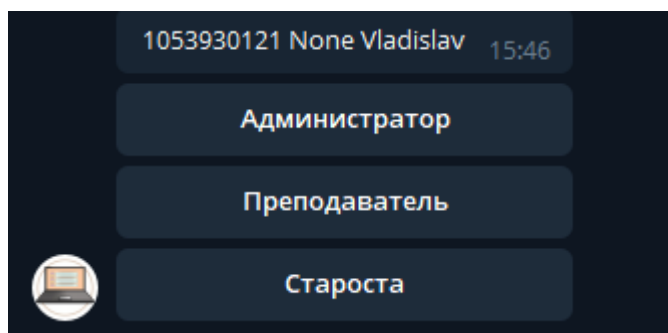


Рисунок 3.2 – Выбор роли

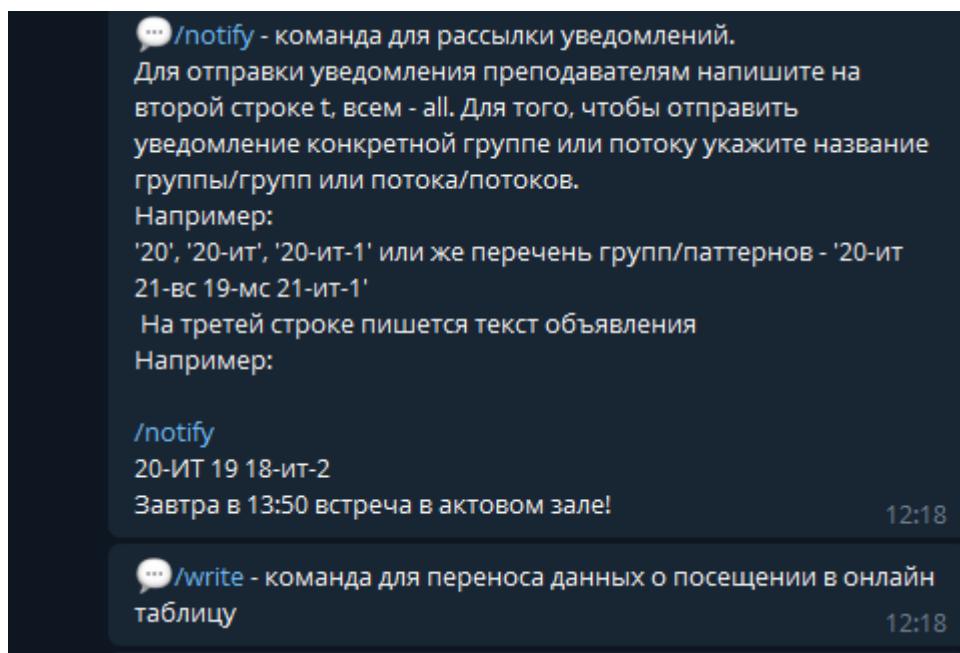


Рисунок 3.3 – Функция совершения рассылки и переноса данных

3.2 Реализация пользовательской части

В данном разделе предоставлено описание результатов разработки модуля пользовательской части, реализованной как Telegram-bot.

Внешний вид и иконка Telegram-бота нарисованная вручную отображены на рисунках 3.4, 3.5.

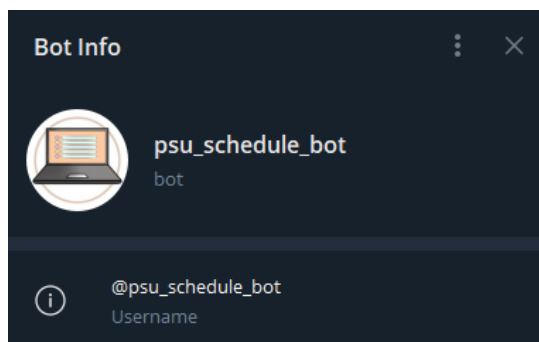


Рисунок 3.4 – Внешний вид

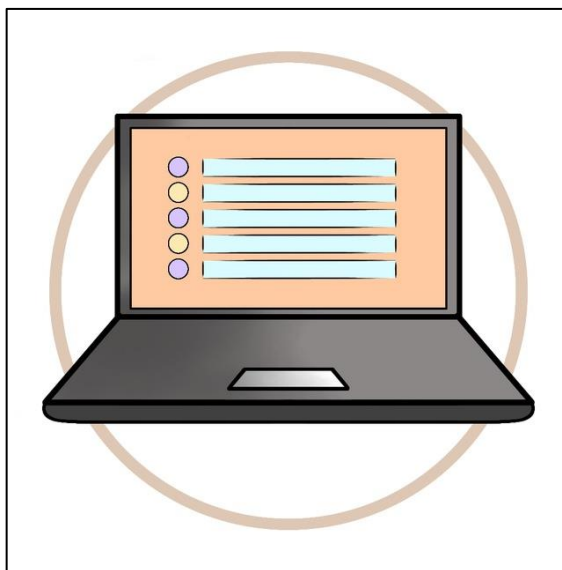


Рисунок 3.5 – Иконка

Реализованные команды, разделенные по ролевому доступу, отображенные на рисунке 3.6.

























	<code>/start</code>		Начало работы с ботом
	<code>/help</code>		Описание основных команд бота
	<code>/registration</code>		Регистрация
	<code>/subscribe_day</code>		Ежедневная подписка на расписание
	<code>/subscribe_less</code>		Подписка на уведомление перед парой
	<code>/schedule_now</code>		Расписание на сегодня
	<code>/schedule_tomorrow</code>		Расписание на завтра
	<code>/schedule_teacher</code>		Расписание преподавателя
	<code>/schedule_group</code>		Расписание группы
	<code>/visit</code>		Отправка данных о посещении
	<code>/write</code>		Запись данных в онлайн таблицу
	<code>/notify</code>		Рассылка сообщений

Рисунок 3.6 – Команды Telegram-бота

Тем временем пользовательская часть делится на 3 роли:

1. Студент.
2. Студент-староста.
3. Преподаватель.

Студент может:

1. Зарегистрироваться как участник группы, рисунок 3.7.
2. Подписаться на 2 вида рассылки, рисунок 3.8.
3. Получить свое расписание на сегодня и на завтра, рисунок 3.9.
4. Получить расписание определенного преподавателя или группы на определённый день недели, рисунок 3.10.

Студент-староста отличается от обычного студента лишь тем, что может отправлять отсутствующих, рисунок 3.11.

Преподаватель умеет все что умеет студент, а также дополнительно имеет возможность совершать рассылку и отправлять данные о пропусках в онлайн таблицу, рисунок 3.12.

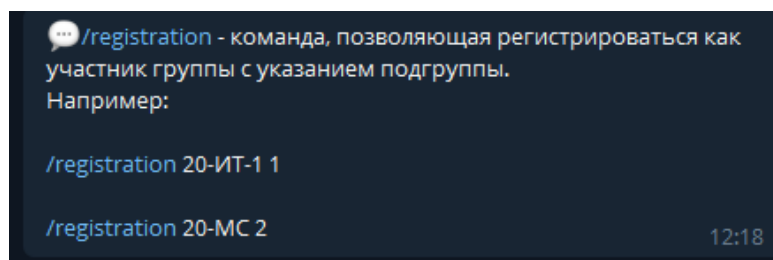


Рисунок 3.7 – Регистрация

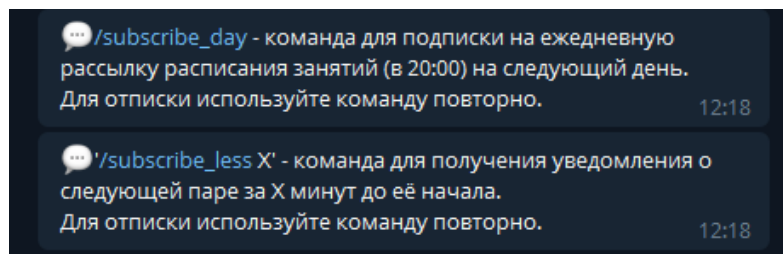


Рисунок 3.8 – Подписки

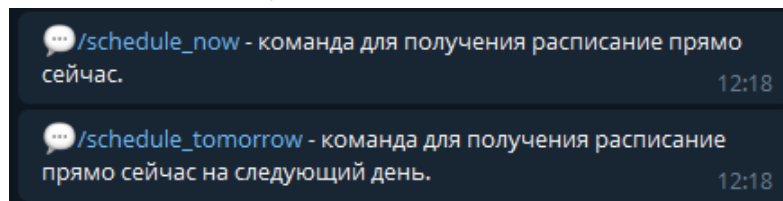


Рисунок 3.9 – Мгновенное расписание

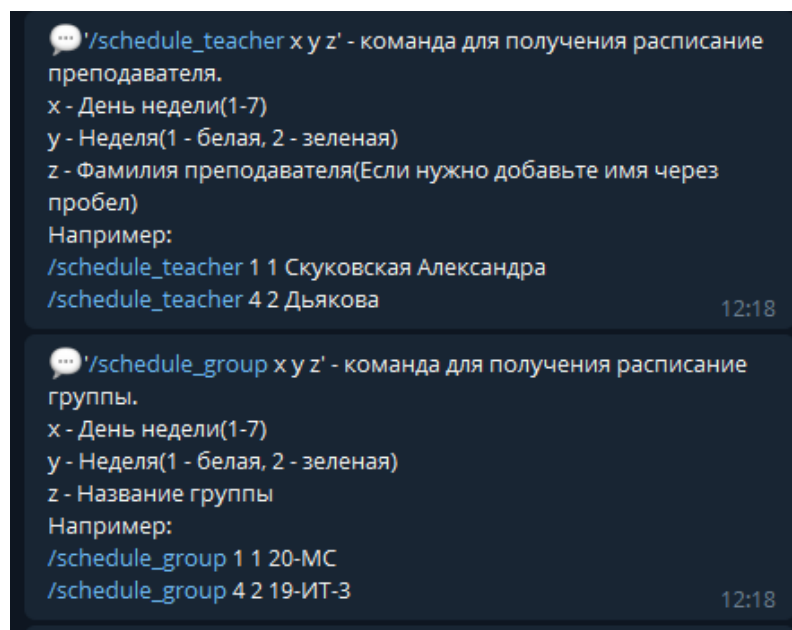


Рисунок 3.10 – Расписание определенного дня и преподавателя

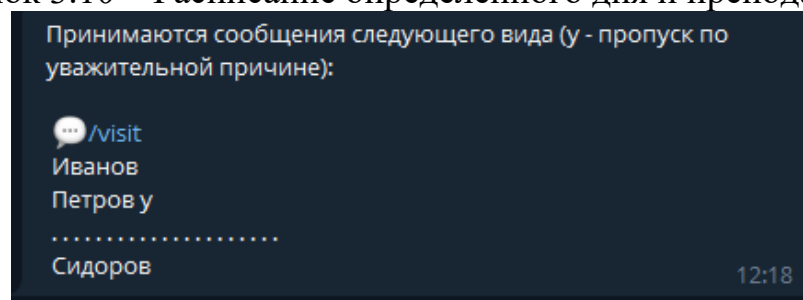


Рисунок 3.11 – Отправка посещения

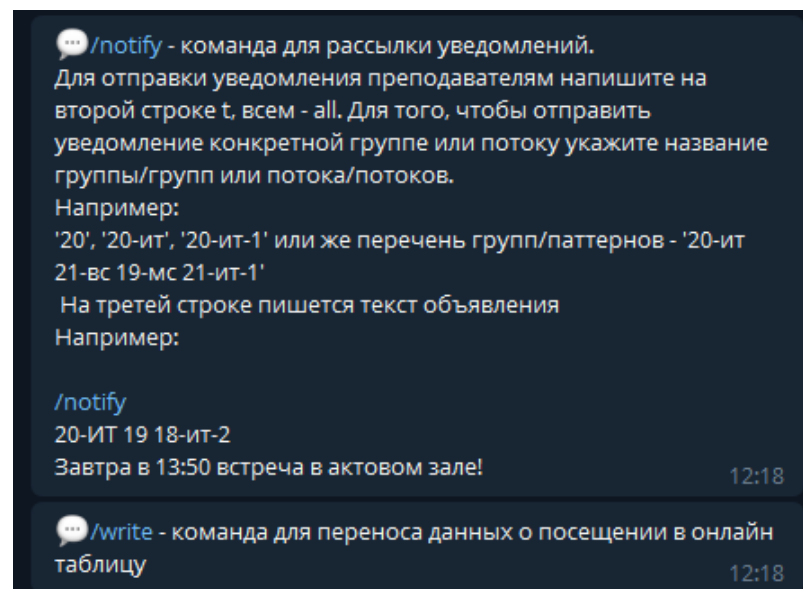


Рисунок 3.12 – Рассылка и перенос данных

Стоит уточнить что студенты и преподаватели получают свое расписание по отдельности.

3.3 Тестирование

В результате тестирования программного продукта критических ошибок не обнаружено.

Работа Telegram-бота была протестирован реальными пользователями. При нагрузке 70 пользователей ошибок и багов выявлено не было.

Telegram-бот был запущен на VPS. VPS — это услуга предоставления в аренду так называемого виртуального выделенного сервера. В плане управления операционной системой по большей части она соответствует физическому выделенному серверу. В частности: root-доступ, собственные IP-адреса, порты, правила фильтрации и таблицы маршрутизации. Отклонений от ожидаемого поведения в процессе выполнения установлено не было.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной научной работы по реализации Telegram-бота по рассылке расписания и оповещений через мессенджер было воплощено следующие:

1. Произведен анализ технической документации, по итогу которого были выделены главные достоинства предметной области.
2. Установлены требования для Telegram-бота.
3. Спроектирована СУБД.
4. Отрисован современный дизайн логотипа Telegram-бота, который отражает отношение разработчика приложения к кафедре технологий программирование УО «Полоцкого государственного университета».
5. реализован Telegram-бот с перспективой на улучшение адаптивности, повышение скорость и с минимальными требованиями к вычислительным мощностям.

В результате разработки была реализована удобная для работы административная часть, позволяющая производить работу с базой данных. Также была воплощена функциональная, удобная, простая и современная пользовательская часть, которую по достоинству оценило большинство пользователей.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Entity Framework / Microsoft Docs – Режим доступа: <https://docs.microsoft.com/en-us/aspnet/entity-framework> – Дата доступа: 15.02.2022.
2. Руководство по классическим приложениям (Windows Forms .NET) – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-5.0> – Дата доступа: 18.02.2022.
3. VPS – Режим доступа: <https://ru.wikipedia.org/wiki/VPS> - Дата доступа: 17.02.2022.
4. Welcome to aiogram's documentation! – Режим доступа: <https://docs.aiogram.dev/en/latest/> - Дата доступа: 16.02.2022.
5. pyodbc – Режим доступа: <https://pypi.org/project/pyodbc/> - Дата доступа: 16.02.2022.