СИСТЕМА ТЕСТИРОВАНИЯ СТУДЕНТОВ СРЕДСТВАМИ TELEGRAM-БОТА

ВЫПОЛНИЛ: СТУДЕНТ ГРУППЫ 1944
БЕСПАЛОВ МАКСИМ СЕМЕНОВИЧ

РУКОВОДИТЕЛЬ: ДОЦЕНТ, КАНД.ТЕХН.НАУК
ШАХОМИРОВ АНДРЕЙ ВИКТОРОВИЧ

ЦЕЛИ И ЗАДАЧИ РАБОТЫ

Цель работы: Разработка системы тестирования для студентов с использованием Telegram-бота.

Задачи работы:

- **Провести анализ предметной области и рассмотреть существующие аналоги на рынке.**
- **Спроектировать базу данных и все необходимые возможности бота.**
- **Разработать Telegram-бота.**
- **Подключить базу данных.**
- **Реализовать функционал для создания тестов и их управления.**
- **Обеспечить возможность студентам проходить тестирования через бота.**

АКТУАЛЬНОСТЬ ТЕМЫ

Предметная область онлайн тестирования студентов охватывает проведение тестов, оценку знаний и успеваемости студентов с использованием интернет-технологий. Онлайн тестирование стало популярным в образовательной среде благодаря доступности, автоматизации процесса и гибкости.

СРАВНЕНИЕ С АНАЛОГАМИ НА РЫНКЕ

Параметры	Готовые решения на рынке	Система тестирования средствами Telegram-бота
Удобство использования	-	+
Разнообразие функций при создании тестов	+	-
Ведение учета успеваемости студента по каждому тесту	+	+
Более эффективная организация процесса тестирования за счет дополнительных функций	-	+
Автоматическая генерация созданных тестов	+	+
Легкость в настройке всех функций решения	-	+

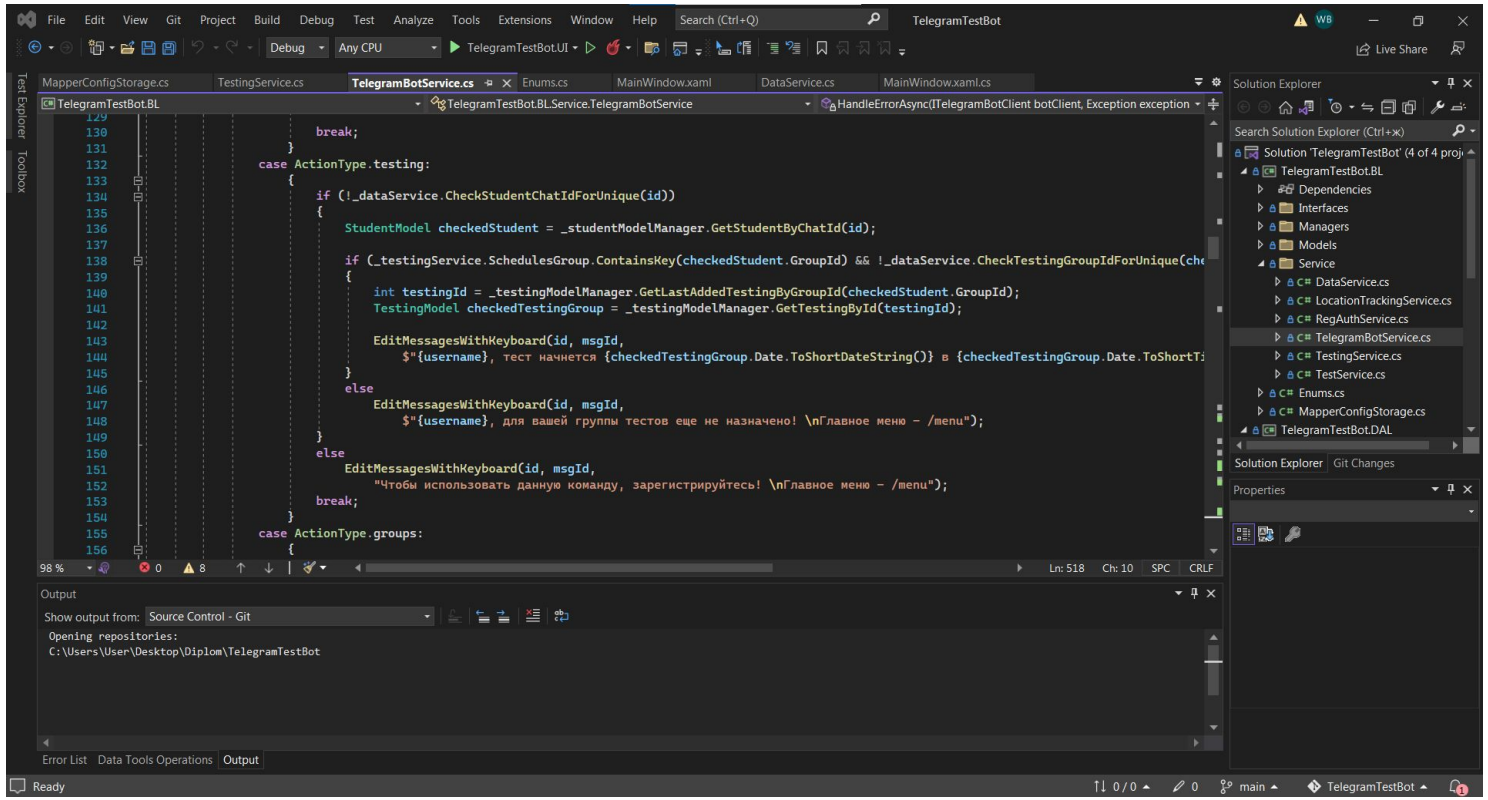
СРЕДА РЕАЛИЗАЦИИ

ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ РАЗРАБОТКИ:

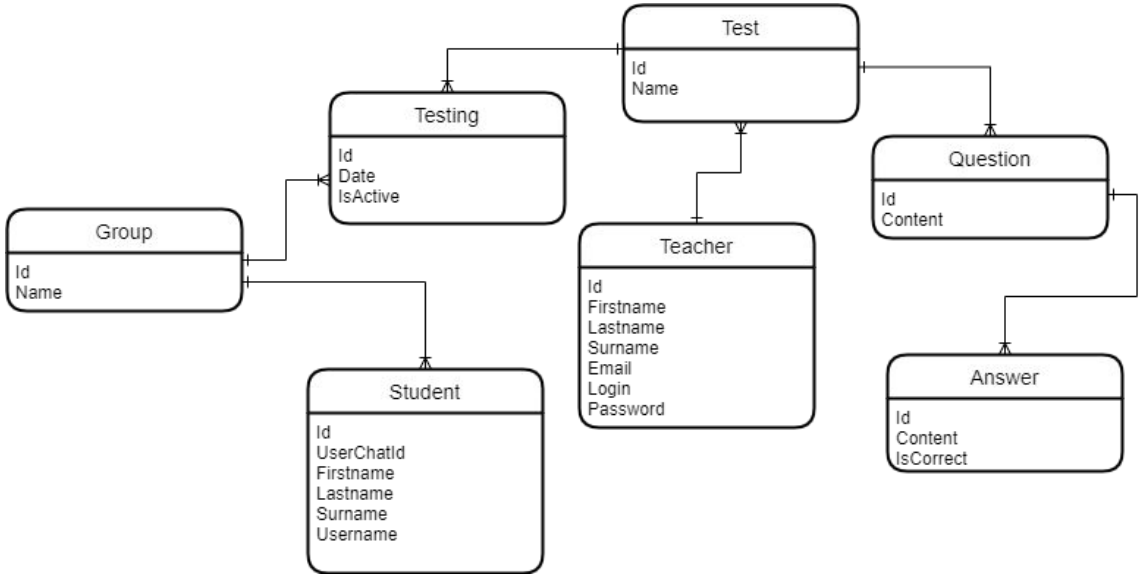
- MS VISUAL STUDIO
- MSSQL SERVER

ПОДКЛЮЧАЕМЫЕ БИБЛИОТЕКИ ЧЕРЕЗ NuGET:

- TELEGRAM.BOT
- DAPPER
- AUTOMAPPER



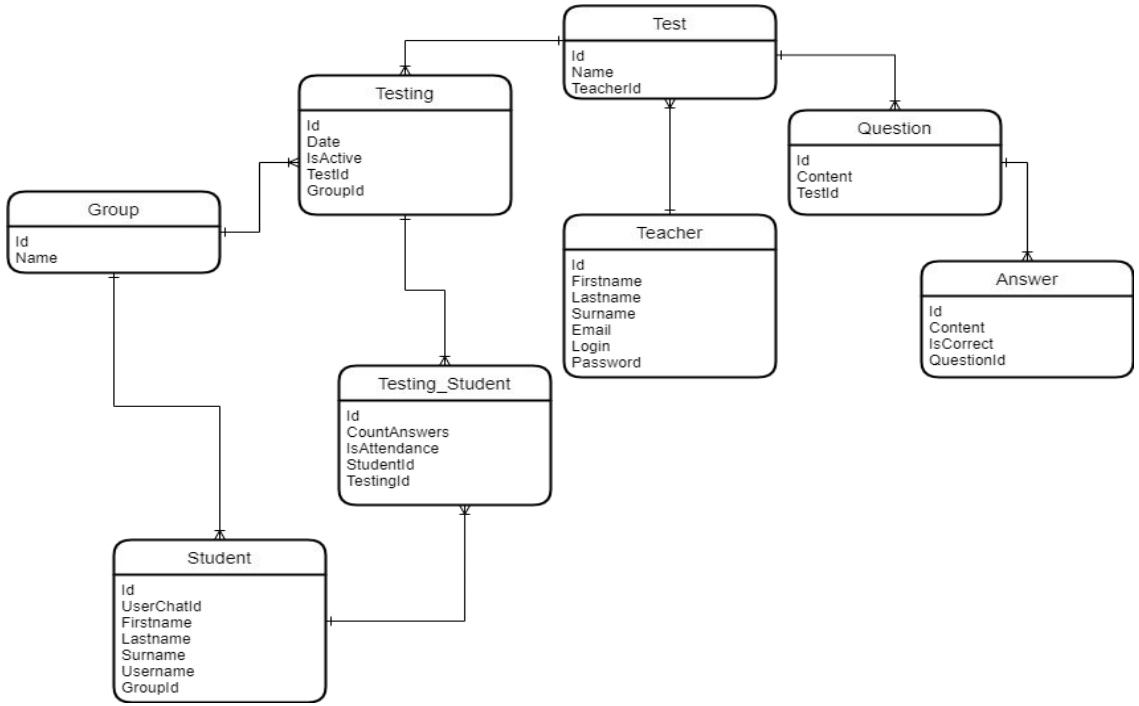
ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ



ER-диаграмма базы данных

Формирование отношений происходит с помощью правил один ко многим или многие ко многим

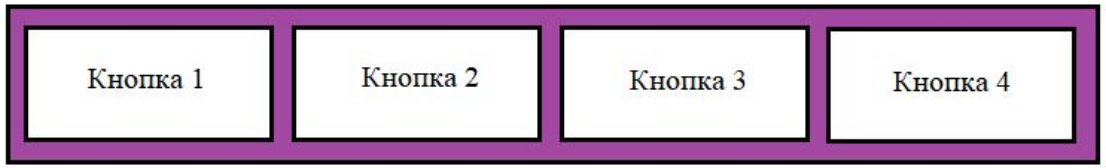
Схема базы данных после формирования отношений



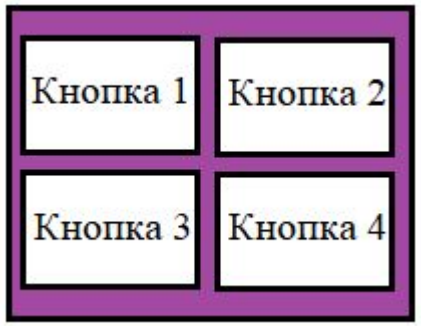
ПРОЕКТИРОВАНИЕ TELEGRAM-БОТА

Для пользователя бота, а именно, студента, должно быть главное меню, в котором будут отображаться все возможные функции. Для этого были выделены основные: регистрация, список преподавателей, информация о будущем тестировании и выбор группы.

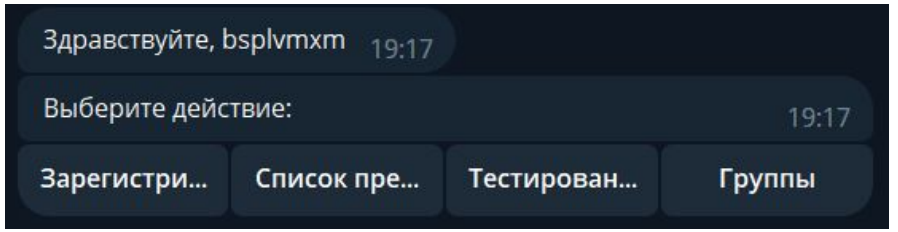
1.



2.



3.



ИСПОЛЬЗОВАНИЕ NuGET ПАКЕТОВ

```
1 reference
public class StudentManager : IStudentManager
{
    2 references
    public void AddStudent(StudentDTO newStudent)
    {
        using (var connection = new SqlConnection(ServerSettings._connectionString))
        {
            connection.Open();

            connection.QuerySingle<StudentDTO>
                (
                    StoredProcedures.Student_Add,
                    param: new
                    {
                        UserChatId = newStudent.UserChatId,
                        FirstName = newStudent.Firstname,
                        LastName = newStudent.Lastname,
                        SurName = newStudent.Surname,
                        Username = newStudent.Username,
                        GroupId = newStudent.GroupId
                    },
                    commandType: System.Data.CommandType.StoredProcedure
                );
        }
    }
```

Dapper

```
cfg.CreateMap<StudentDTO, StudentModel>()
.ForMember("Id", opt => opt.MapFrom(s => s.Id))
.ForMember("UserChatId", opt => opt.MapFrom(s => s.UserChatId))
.ForMember("Firstname", opt => opt.MapFrom(s => s.Firstname))
.ForMember("Lastname", opt => opt.MapFrom(s => s.Lastname))
.ForMember("Surname", opt => opt.MapFrom(s => s.Surname))
.ForMember("Username", opt => opt.MapFrom(s => s.Username))
.ForMember("GroupId", opt => opt.MapFrom(g => g.GroupId))
.ReverseMap();
```

AutoMapper

Telegram.Bot

```
1 reference
public TelegramBotService(Action<string> onMessage)
{
    _botClient = new TelegramBotClient(_dataService.token);
    _onMessage = onMessage;
}

1 reference
public void StartBot(string pass)
{
    if (pass == "12345")
        _botClient.StartReceiving(HandleUpdateAsync, HandleErrorAsync);
}

1 reference
private async Task HandleUpdateAsync(ITelegramBotClient botClient, Update update, CancellationToken cancellationToken)
{
    if(update.Message != null && update.Message.Text == "/start" || update.Message?.Text == "/menu")...
    else if (update.CallbackQuery != null)...
    else if (update.Message?.Text != null && DataService.UserAnswers.ContainsKey(update.Message.Chat.Id))...
    else if (update.Message?.Text != null && DataService.UserAnswersForGroup.ContainsKey(update.Message.Chat.Id))...
    else if (update.Message != null && update.Message.Type == MessageType.Location && !_dataService.UsersWithGeo.Contains(update.
    else if (update.Message != null && update.Message.Text != null)...
}

1 reference
private async Task HandleErrorAsync(ITelegramBotClient botClient, Exception exception, CancellationToken cancellationToken)
{
    await Task.CompletedTask;
}
```

ИСПОЛЬЗОВАНИЕ АССИХРОННОСТИ

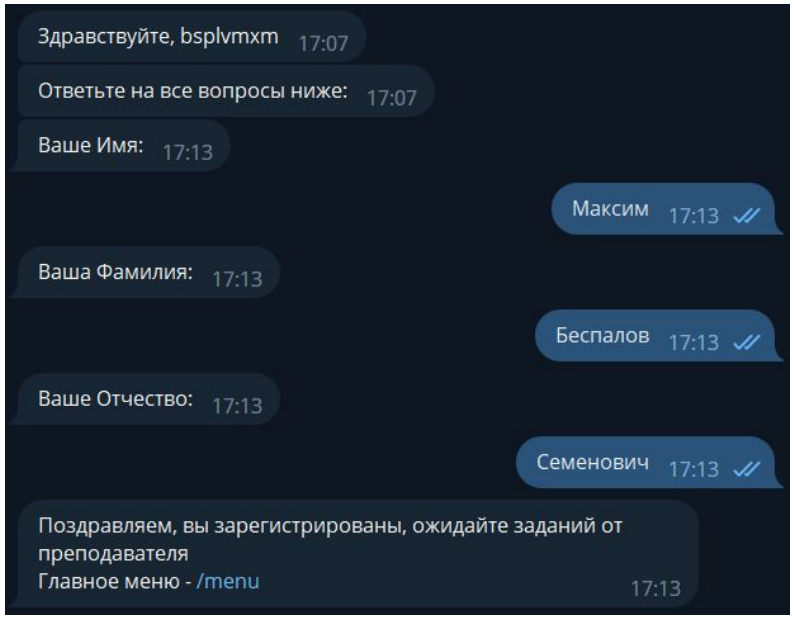
```
1 reference
private async void SendActionMenu(long chatId)
{
    var inlineKeyboard = new InlineKeyboardMarkup(new[]
    {
        InlineKeyboardButton.WithCallbackData("Зарегистрироваться", "reg"),
        InlineKeyboardButton.WithCallbackData("Список преподавателей", "teachers"),
        InlineKeyboardButton.WithCallbackData("Тестирование", "testing"),
        InlineKeyboardButton.WithCallbackData("Группы", "groups")
    });

    await _botClient.SendTextMessageAsync(new ChatId(chatId), "Выберите действие:", replyMarkup: inlineKeyboard);
}
```

Асинхронное программирование позволяет выполнить блок кода без остановки (или блокировки) всего потока, в котором выполняется действие.

```
1 reference
private async void UpdateStudentAfterConfirmAttendance(int studentId, int groupId, long userId)
{
    int testingId = _testingModelManager.GetLastAddedTestingByGroupId(groupId);
    int testingStudentId = _testingStudentModelManager.GetTestingStudentByStudentIdByTestingId(studentId, testingId);
    TestingStudentModel test = _testingStudentModelManager.GetTestingStudentById(testingStudentId);

    test.IsAttendance = true;

    _testingStudentModelManager.UpdateTestingStudentById(test);

    var inlineKb = new InlineKeyboardMarkup(new[]
    {
        InlineKeyboardButton.WithCallbackData("Начать тестирование", "test")
    });

    await _botClient.SendTextMessageAsync(userId,
        "Вы прошли тест с геопозицией, можете начать основное тестирование.", replyMarkup: inlineKb);
}
```

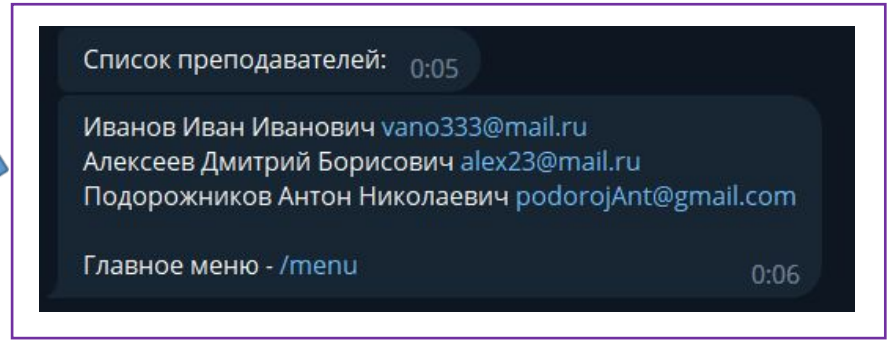
ДЕМОНСТРАЦИЯ РАЗРАБОТАННОЙ СИСТЕМЫ

Возможности пользователя бота(студенты):

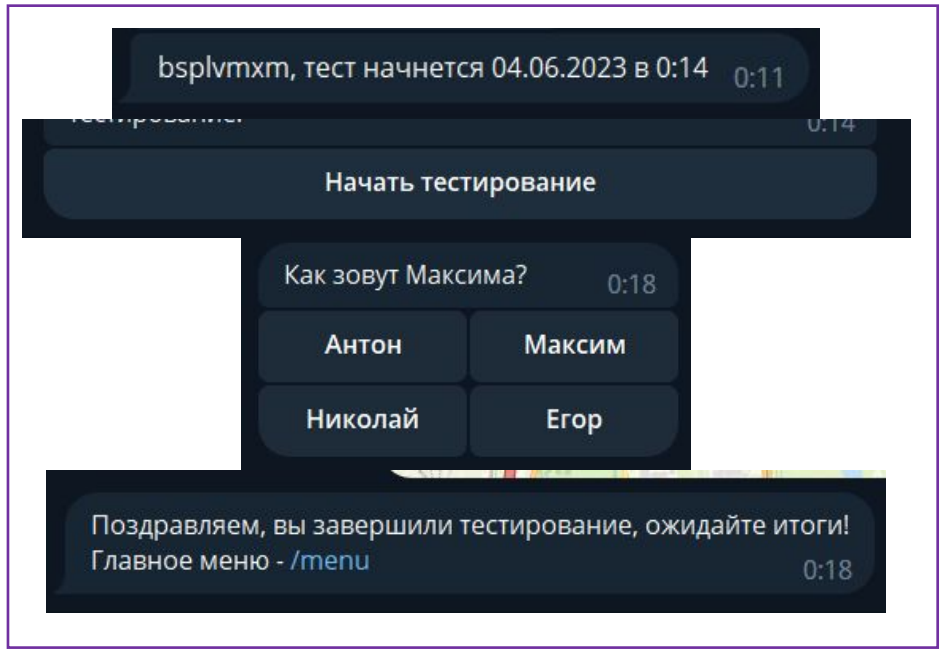


Регистрация

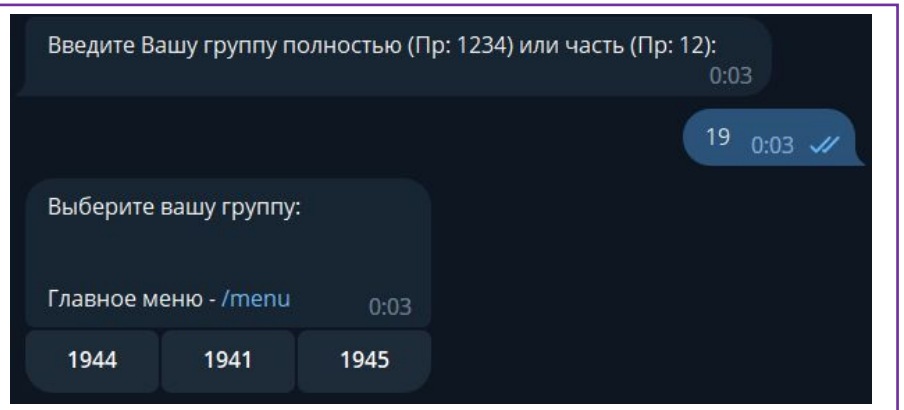
Вывод списка преподавателей



Прохождение тестирования

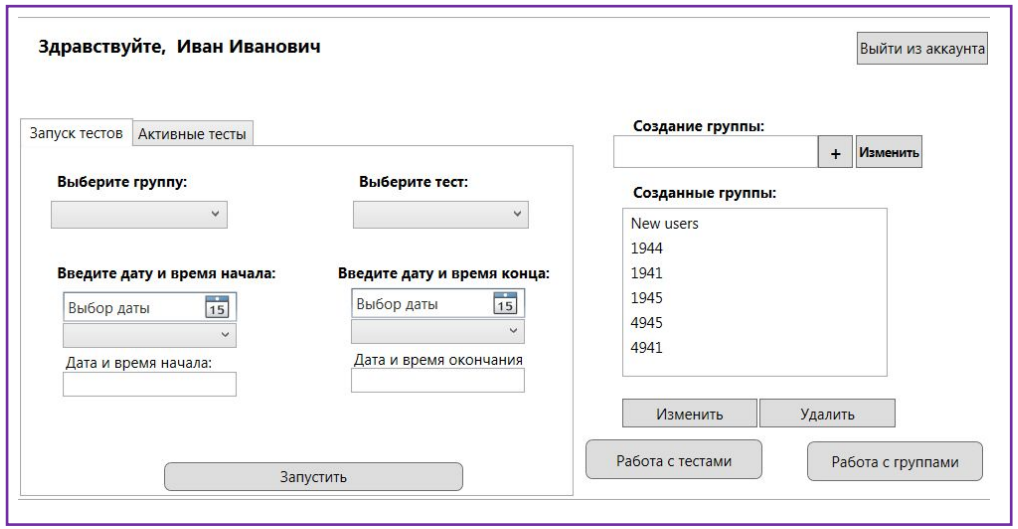


Выбор группы

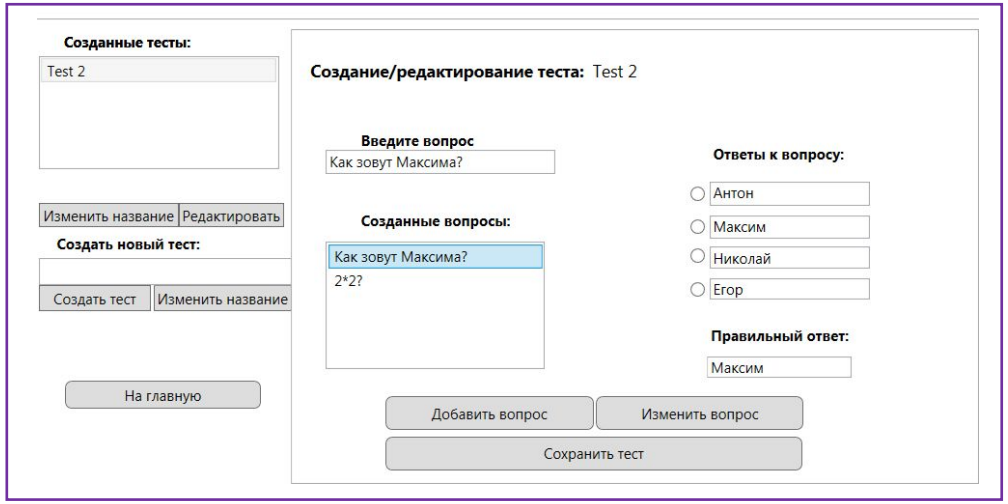


ДЕМОНСТРАЦИЯ РАЗРАБОТАННОЙ СИСТЕМЫ

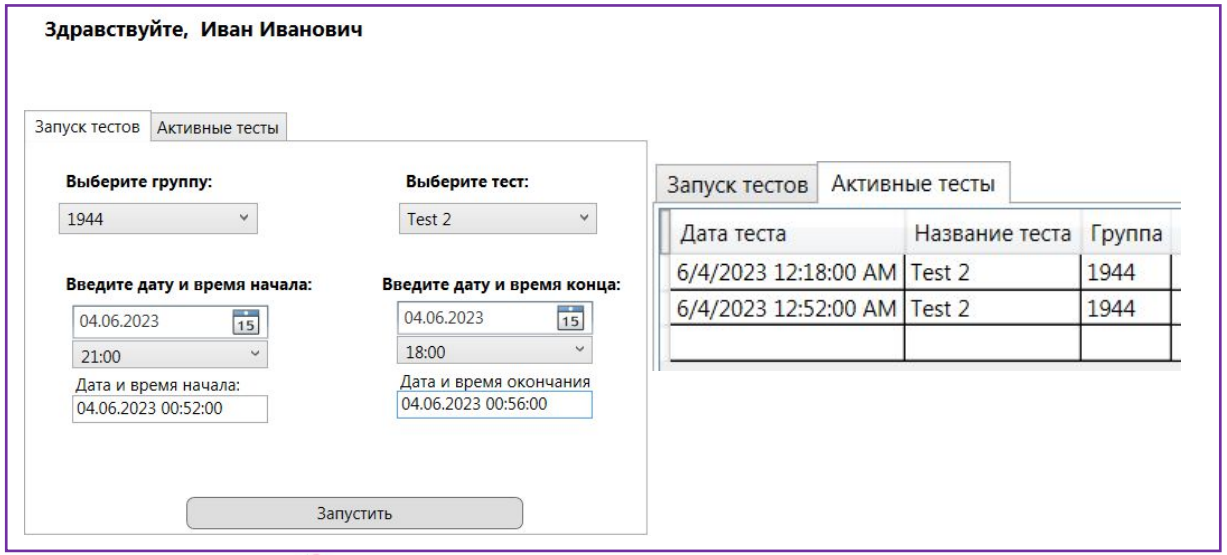
Возможности пользователя управляющего приложения(преподаватели):



Запуск тестирования и просмотр активных тестов



Главное меню управляющего приложения



Создание теста

ДЕМОНСТРАЦИЯ РАЗРАБОТАННОЙ СИСТЕМЫ

База данных:

Results | Messages

	Id	UserChatId	Firstname	Lastname	Surname	Username	GroupId	IsDeleted
1	1	503440584	Максим	Семенович	Беспалов	bsplvmxm	2	0

Results | Messages

	Id	Name	IsDeleted
1	1	New users	0
2	2	1944	0
3	3	1941	0
4	4	1945	0
5	5	4945	0
6	6	4941	0

Зарегистрированный студент и группы

Активные тестирования

Results | Messages

	Id	TestId	Date	GroupId	isActive	IsDeleted
1	1	1	2023-06-03 18:25:00.000	2	0	0
2	2	1	2023-06-04 00:14:00.000	2	0	0
3	3	1	2023-06-04 00:18:00.000	2	1	0
4	4	1	2023-06-04 00:52:00.000	2	0	0

Results | Messages

	Id	Name	TeacherId	IsDeleted
1	1	Test 2	1	0

Results | Messages

	Id	Content	TestId	IsDeleted
1	1	Как зовут Максима?	1	0
2	2	2*2?	1	0

Results | Messages

	Id	Content	QuestionId	IsCorrect	IsDeleted
1	1	Антон	1	0	0
2	2	Максим	1	1	0
3	3	Николай	1	0	0
4	4	Егор	1	0	0
5	5	5	2	0	0
6	6	6	2	0	0
7	7	4	2	1	0
8	8	3	2	0	0

Созданный тест

Results | Messages

	Id	CountAnswers	StudentId	TestingId	IsAttendance	IsDeleted
1	1	1	1	1	1	0
2	2	1	1	2	1	0
3	3	2	1	3	1	0
4	4	2	1	4	1	0

Пройденные тестирования

ВЫВОДЫ

В рамках выпускной квалификационной работы была разработана система тестирования студентов с использованием Telegram-бота.

Целью работы было создание инструмента, позволяющего проводить удаленное тестирование студентов и оценивать их знания в определенных предметных областях.

Разработанная система представляет собой полноценный продукт, который может быть использован для проведения тестов как в дистанционном режиме, так и в очном. Она позволяет студентам получить доступ к тестам через Telegram-бота и проходить их непосредственно в мессенджере.

БЛАГОДАРЮ ЗА ВНИМАНИЕ!