МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Индивидуальное домашнее задание

по дисциплине

“Базы данных”

Студент АС-21-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С. М.

(подпись, дата)

Руководитель

Доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексеев В. А.

(подпись, дата)

Липецк 2023

Содержание отчета

2. Цель работы, задание

3. Характеристика предметной области

3.1. Описание предметной области

3.2. Характеристика пользователей ИС

3.3. Требования к отчетам

4. Модель данных (PowerDesigner)

4.1. Диаграмма концептуальной модели данных

4.2. Диаграмма физической модели данных

5. Разработка прикладного приложения

5.1. Платформа разработки

(описание используемых аппаратных и программных средств для разработки)

5.2. Схема взаимодействия с БД

(схематичное пояснение, как в выбранном фреймворке реализуются типовые операции работы с данными)

5.2.1. Связь приложения с БД (connection)

5.2.2. Отображение таблицы с данными

5.2.3. Модификация данных

5.2.4. Вызов хранимой процедуры

5.3. Разработка пользовательского интерфейса

(скриншоты с представлением настройки компонентов для отображения и редактирования данных – выборочно 2-3 шт.)

5.4. Текст программы

(иллюстрация манипуляций с данными, включая вызов хранимых процедур БД - выборочно, не более 7 страниц).

6. Тестирование прикладного приложения

(Скриншоты, отображающие работу программы. Обязательно должно быть

проиллюстрировано формирование отчетов, предусмотренных в лабораторной работе №1).

**2. Цель работы**

Получение первичных навыков разработки прикладных приложений для БД, освоение фреймворков для работы с БД.

Задание

Реализовать прикладное приложение, обеспечивающее просмотр и редактирование содержимого спроектированной в ходе лабораторного практикума БД, выполнение и просмотр результатов запросов, вызов хранимых процедур.

**3. Характеристика предметной области**

**3.1 Описание предметной области**

Выбранная АИС: “Система управления учебным процессом в высшем учебном заведении”

Назначение Автоматизированной Информационной Системы (АИС):

Целью АИС является автоматизация и улучшение управления учебным процессом в высшем учебном заведении с целью повышения его эффективности и обеспечения более качественного образования для студентов. АИС предназначена для сбора, хранения, обработки и предоставления информации, необходимой для всех участников учебного процесса, включая студентов, преподавателей и администраторов, а также для автоматизации административных задач, связанных с учебным процессом.

Перечень решаемых задач:

* Учет студентов: АИС должна позволять учреждению вести учет всех студентов, включая личные данные, контактную информацию, учетные записи и другие сведения.
* Учет преподавателей: Система должна поддерживать информацию о преподавателях, их квалификации, учебных нагрузках и контактных данных.
* Расписание занятий: АИС должна автоматизировать процесс создания и управления расписанием практик, лекций и других учебных мероприятий.
* Учет успеваемости и оценок: Система должна позволять вводить и отслеживать оценки, успеваемость студентов и предоставлять студентам и преподавателям доступ к этой информации.
* Учет учебных предметов и программ: АИС должна содержать информацию о предметах, учебных программах и учебных планах.

АИС в данной предметной области помогает учебным заведениям более эффективно управлять учебным процессом, улучшать качество образования и обеспечивать удобное взаимодействие между всеми участниками образовательной среды.

**3.2 Характеристика пользователей ИС**

1) Студенты:

Решаемые задачи:

1. Просмотр расписания занятий.
2. Отслеживание успеваемости и оценок.

2) Преподаватели:

Решаемые задачи:

1. Ввод и редактирование оценок и успеваемости студентов.
2. Просмотр расписаний.

3) Администраторы системы:

Решаемые задачи:

1. Управление доступом и безопасностью системы.
2. Поддержка пользователей и решение технических проблем.

**3.3 Требования к отчетам**

1. Отчет об успеваемости студентов:  
Цель отчета: позволить студентам просмотреть свои баллы по предметам, по которым он обучаются

Требования:

1. Отображать все оценки студента, полученные в период семестра  
2. У каждой выводимой оценки должен быть преподаватель, который её ставил, и средний балл студента по этому предмету.

2. Отчет о расписании и аудиториях:

Цель отчета:   
Предоставить студентам и преподавателям информацию о расписании занятий и занятых аудиториях.

Требования:

1. Персонализированный доступ для студентов и преподавателей.
2. Подробная информация о каждом занятии, включая название предмета, преподавателя и аудиторию.

3. Отчет об успеваемости и учебной активности студентов:

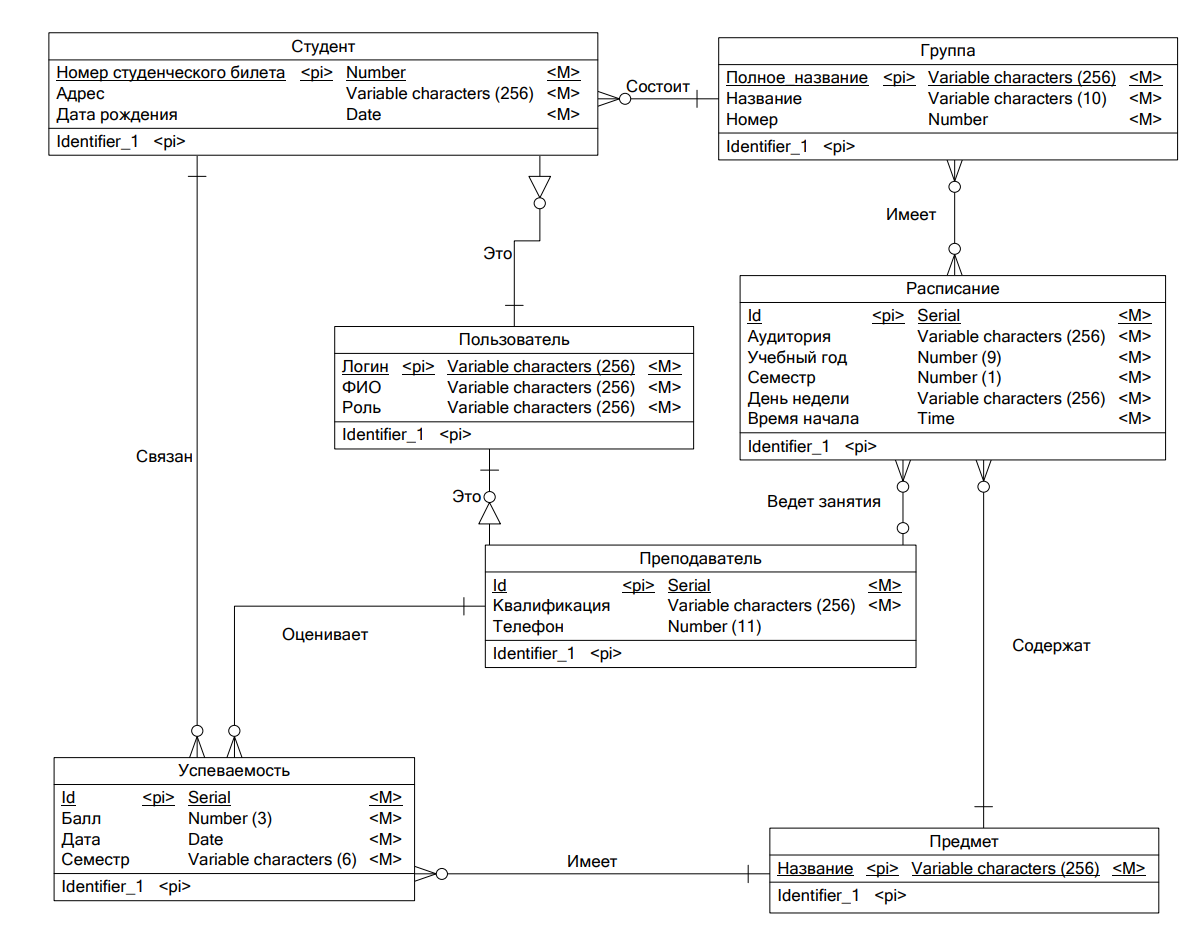
Цель отчета:   
Предоставить преподавателям информацию о посещаемости студентов и их активности в учебном процессе.

Требования:

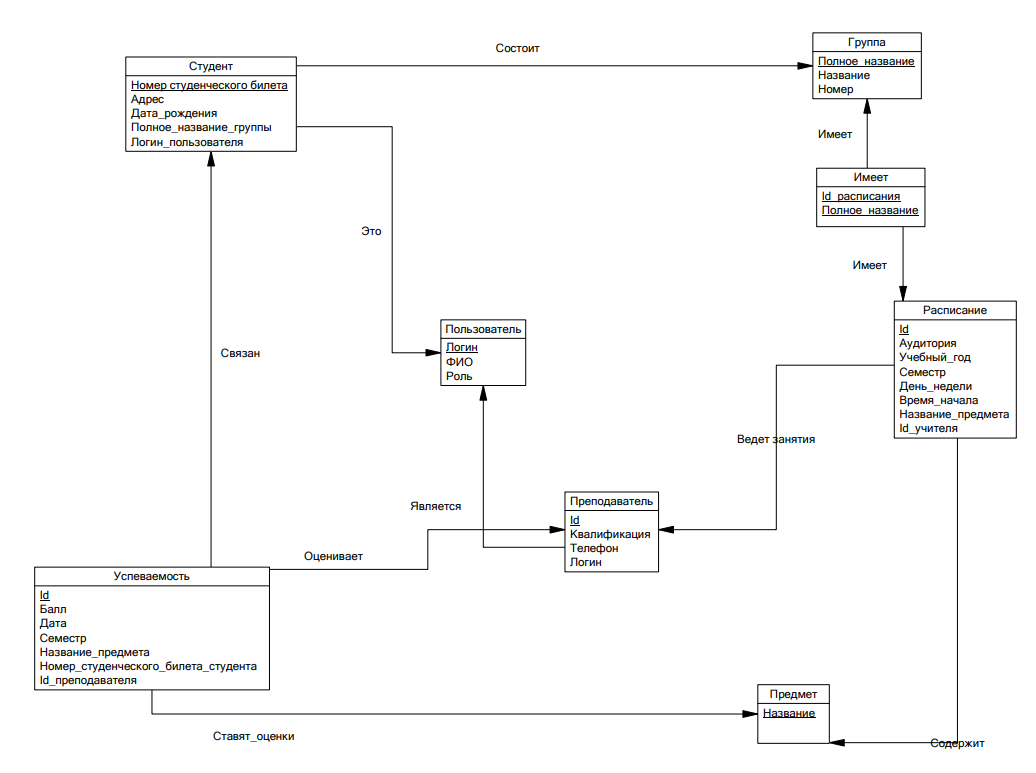
1. Возможность выбора периода (например, текущий семестр или учебный год).
2. Подробная информация о оценках, включая дату, название предмета и преподавателя.

**4. Модель данных**

**4.1 Концептуальная диаграмма**



**4.2 Физическая диаграмма**

****

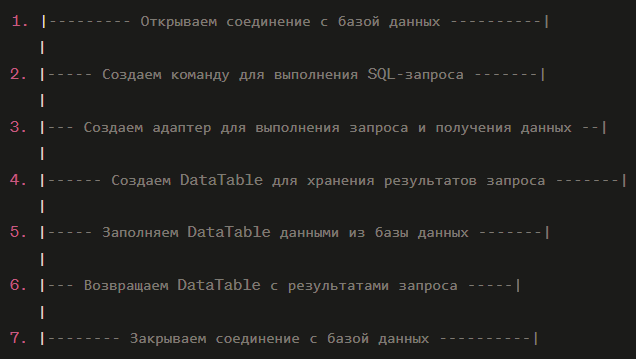
**5. Разработка прикладного приложения**

**5.1 Платформа разработки**

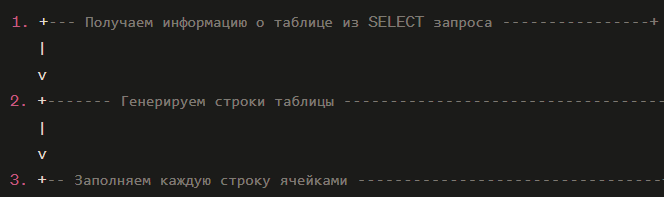
Unity 2021, C#; Visual Studio 2022; Unity Npgsql.

**5.2 Схема взаимодействия с БД**

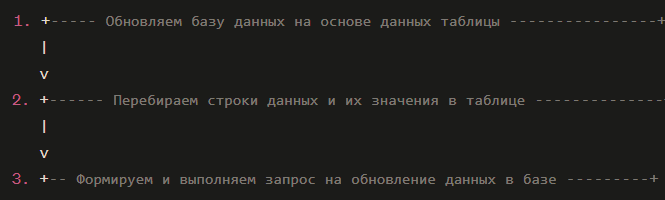
**5.2.1 Связь приложения с БД**

****

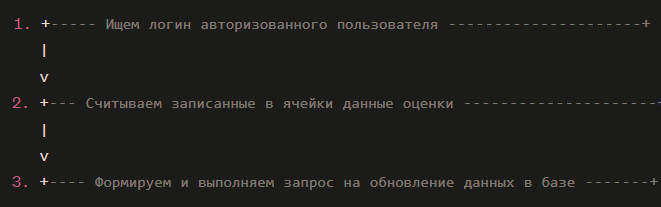
**5.2.2 Отображение таблицы с данными**



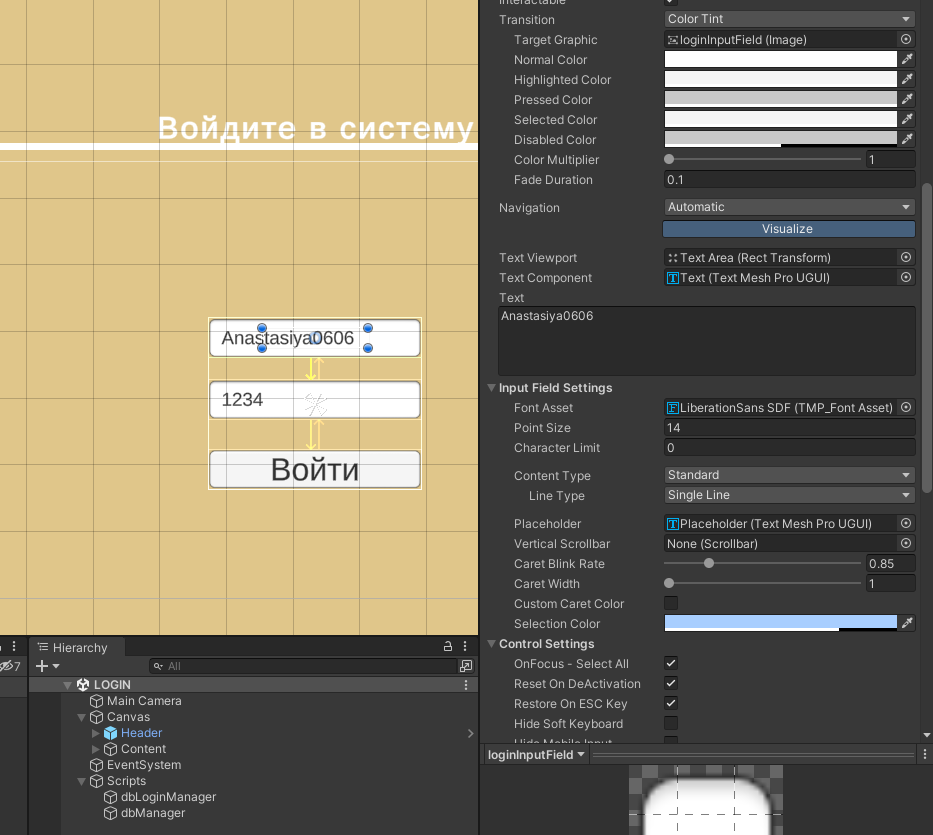
**5.2.3 Модификация данных**

****

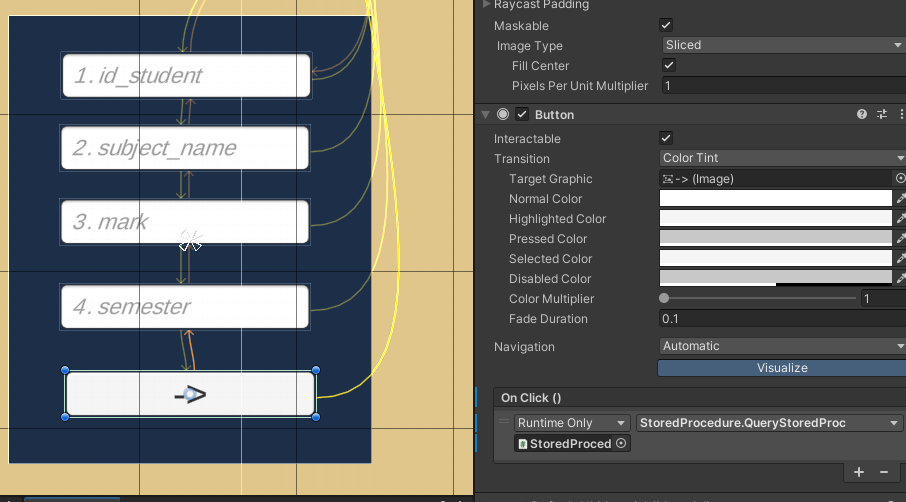
**5.2.4 Вызов хранимой процедуры**



**5.3 Разработка пользовательского интерфейса**



Сцена авторизации пользователя (компонент ввода текста с клавиатуры, в данном случае – логина)



Префаб хранимой процедуры (компонент “кнопка”, который вызывает ивент OnClick(), который, в свою очередь, вызывает публичный метод скрипта, реализующего вызов ХП)

**5.4 Текст программы**

1) Метод, который реализует вызов другого метода, генерирующего таблицу на основе полученного запроса

private void ViewMySchedule()

{

string query =

"SELECT " +

"g.full\_name\_group AS group\_name, " +

"sch.class\_schedule AS classroom, " +

"sch.weekday\_schedule AS weekday, " +

"sch.start\_time\_schedule AS start\_time, " +

"sch.name\_subject AS subject\_name, " +

"t.qualification\_teacher AS teacher\_qualification, " +

"u.full\_name\_user AS teacher\_full\_name " +

"FROM Student s " +

"JOIN group\_data g " +

"ON s.full\_name\_group = g.full\_name\_group " +

"JOIN group\_has\_schedules ghs " +

"ON g.full\_name\_group = ghs.full\_name\_group " +

"JOIN Schedule sch ON ghs.Id\_schedule = sch.Id\_schedule " +

"LEFT JOIN Teacher t ON sch.Id\_teacher = t.Id\_teacher " +

"LEFT JOIN user\_data u ON t.login\_user = u.login\_user " +

$"WHERE s.login\_user = '{UserManager.Login}';";

List<Dictionary<string, string>> scheduleData = DatabaseManager.Instance.ExecuteQueryList(query);

// Отображение данных в DatabaseTableViewer

if (DatabaseTableViewer != null)

{

DatabaseTableViewer.ToggleTable(scheduleData);

}

}

2) Метод, который реализует вызов хранимой процедуры

public void QueryStoredProc()

{

string login = UserManager.Login;

int idTeacher = DatabaseManager.Instance.GetTeacherIdByLogin(login);

string query = $"CALL update\_student\_performance({idNumberStudent.text}, '{subjectName.text}', " +

$"{mark.text}, '{semester.text}', {idTeacher})";

Debug.Log(query);

DatabaseManager.Instance.ExecuteQueryList(query);

}

3) Метод, реализующий обновление БД

private void UpdateDatabase(string currEditableTableColumnName, List<Dictionary<string, string>> data)

{

foreach (Dictionary<string, string> row in data)

{

string primaryKeyValue = row[currEditableTablePrKeyName];

// Проверка наличия данных

if (!row.TryGetValue(currEditableTableColumnName, out string newValue))

{

Debug.LogError($"Column '{currEditableTableColumnName}' not found in the dictionary.");

continue;

}

string updateQuery = $"UPDATE {currEditableTableName} SET {currEditableTableColumnName} = '{newValue}' " +

$"WHERE {currEditableTablePrKeyName} = '{primaryKeyValue}';";

Debug.Log(updateQuery);

DatabaseManager.Instance.ExecuteQuery(updateQuery);

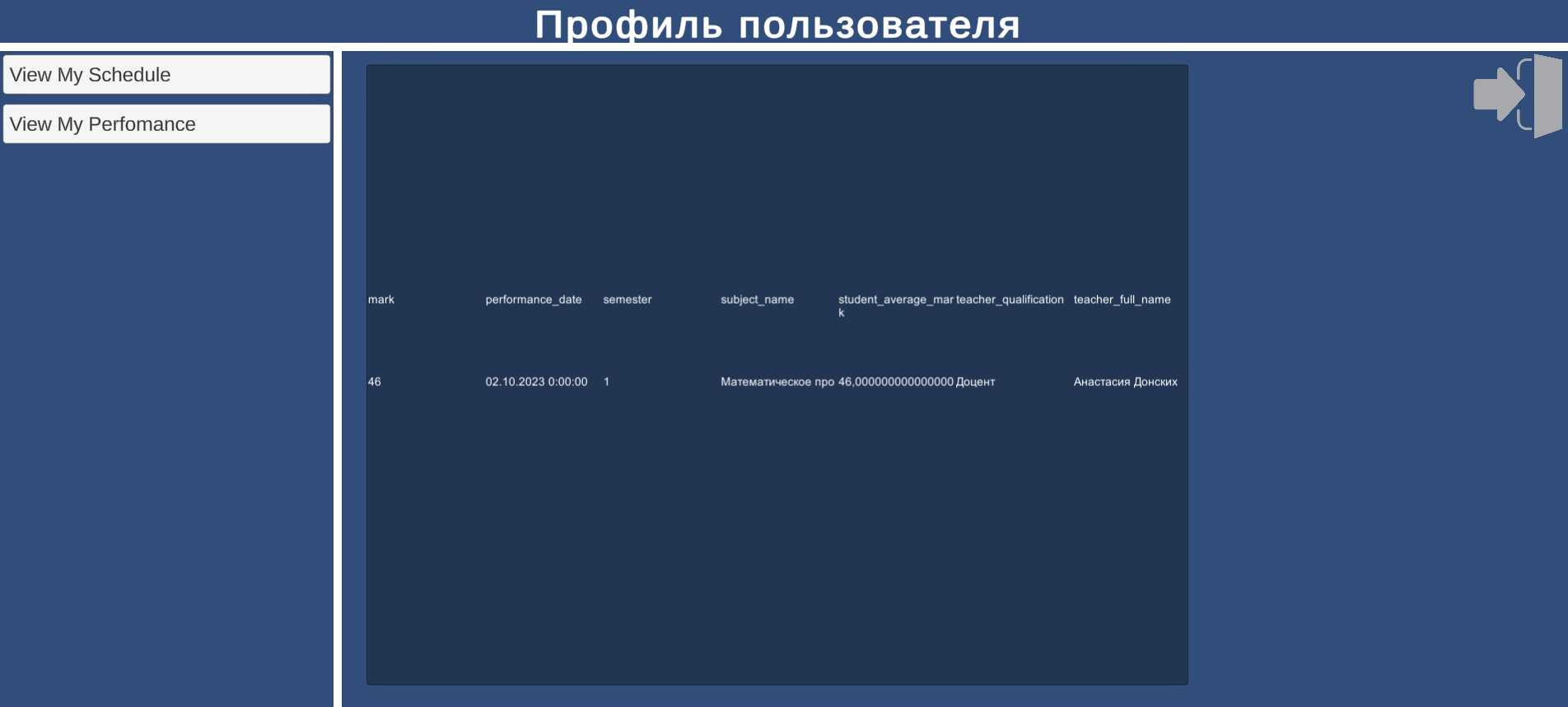
}

}

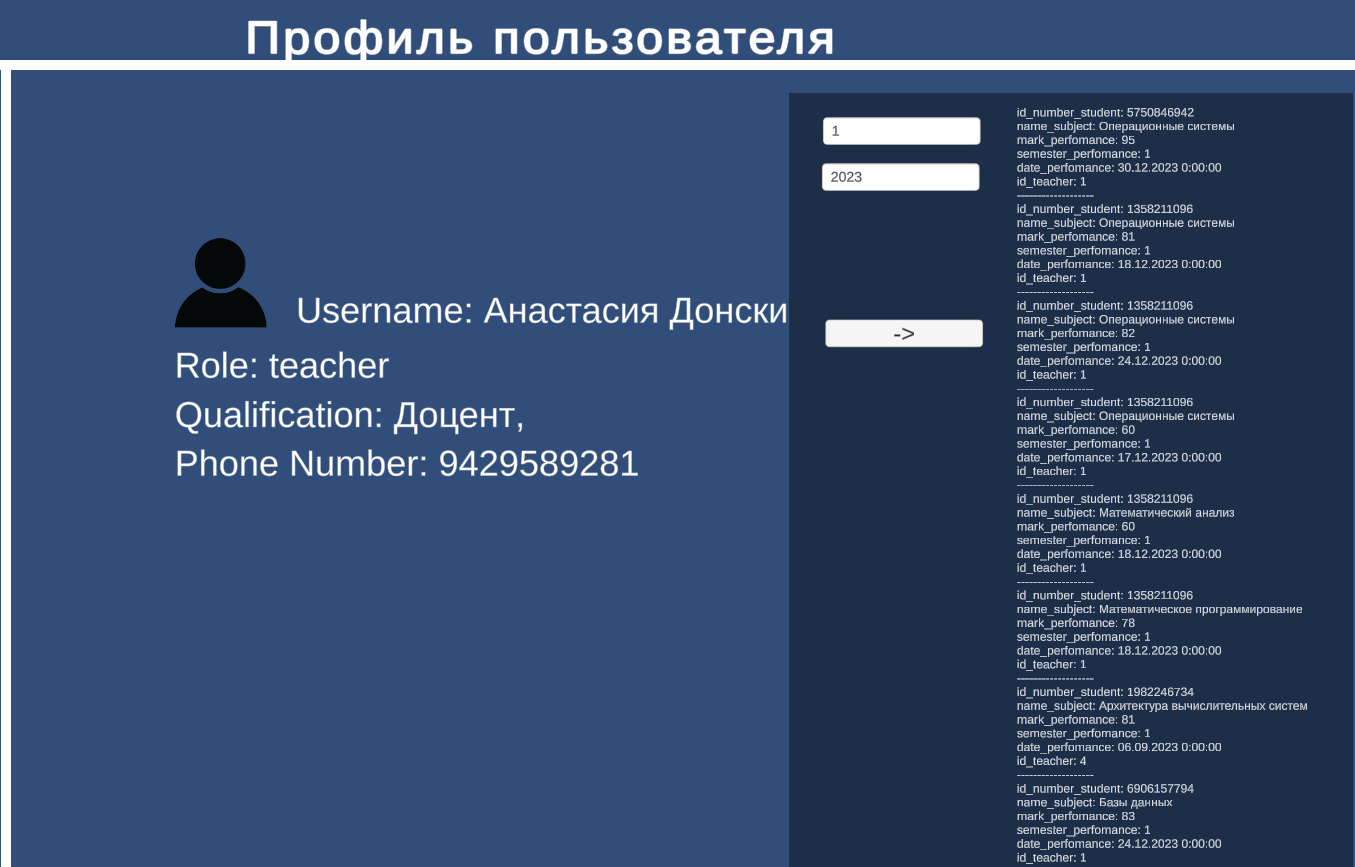
**6. Тестирование прикладного приложения**

****

1) Просмотр расписания занятий студента



2) Просмотр оценок студента

****

3) Отчет преподавателя о выставленных оценках за конкретный период времени (1 семестр, 2023)



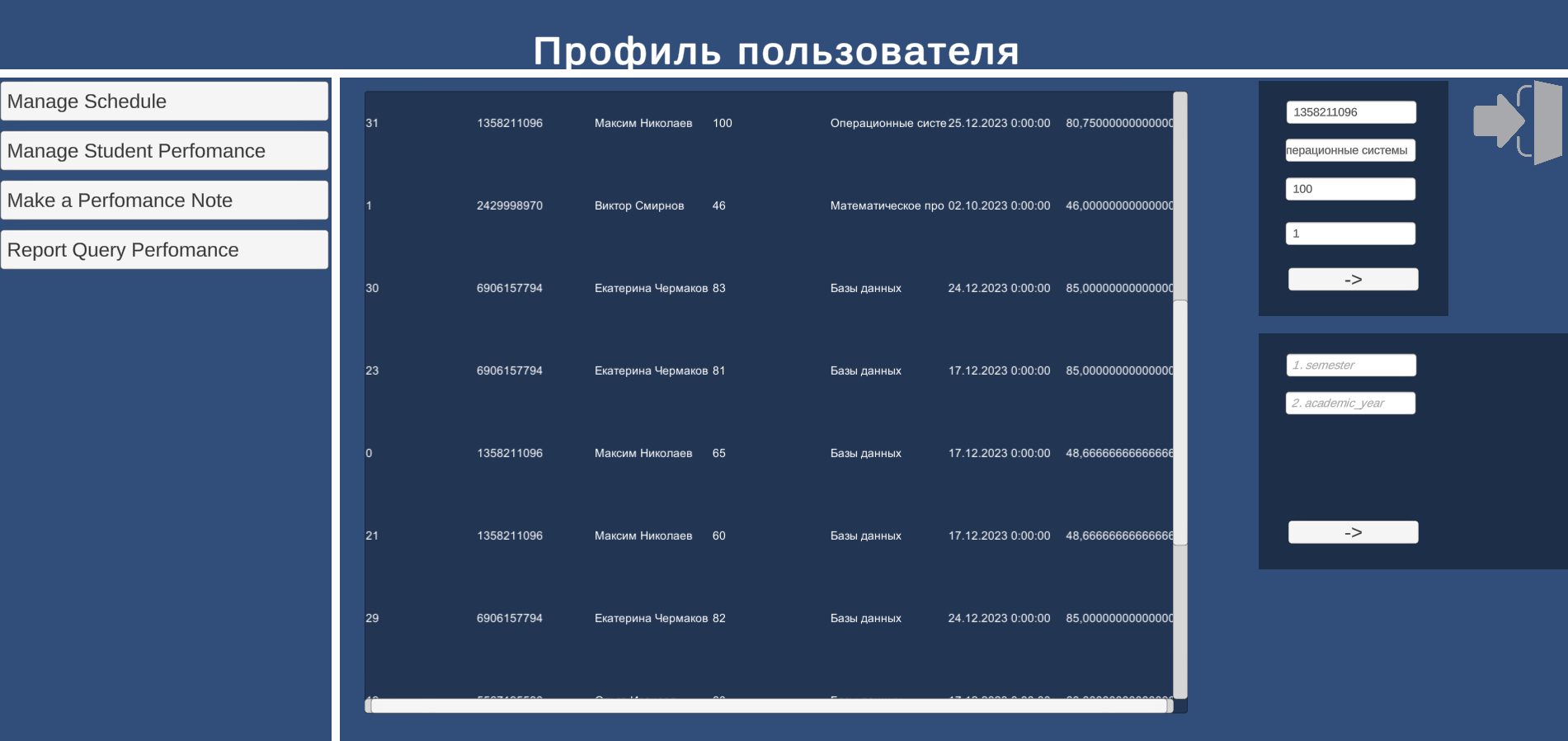
4) Отчет преподавателя об успеваемости студентов, у которых он ведет занятия



5) Вызов хранимой процедуры, которая выставит оценку студенту Максиму Николаеву.



5.1) В Логе видим, что процедура вызвалась с соответствующими значениями параметров.



5.2) Новая полученная оценка Максима, с учетом изменения среднего балла по предмету

Вывод: в ходе выполненной работы получил первичные навыки разработки прикладных приложений для БД, освоил фреймворк для работы с БД.