# **6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

В данном разделе приводится руководство пользователя для приложения «Электронный журнал». При разработке данного приложения учитывались следующие минимальные системные требования:

Таблица 6.1 – Минимальные системные требования

|  |  |
| --- | --- |
| Операционная система | Windows 7/10 64 bit |
| Объем ОЗУ | 1 ГБ |
| Видеокарта | Nvidia 760 |
| Разрешение экрана | 1024 x 768 |
| Свободного места на жестком диске | 141,7 МБ |

Оптимальными же требованиями к разработанному приложению будут считаться значения, соответствующие персональному компьютеру, на котором производилась разработка программного обеспечения.

Таблица 6.2 – Рекомендуемые системные требования

|  |  |
| --- | --- |
| Операционная система | Windows 10 64 bit |
| Объем ОЗУ | 8 ГБ |
| Видеокарта | Nvidia 1650 |
| Разрешение экрана | 3840 x 2160 |
| Свободного места на жестком диске | 1. ГБ |

## **6.1 Установка приложения**

Поскольку для приложения создан установочный файл, то данный процесс не предоставляет никакой сложности для пользователя. Первым делом ему необходимо выбрать установочный файл на своем экране. На ярлыке данного файла установлен логотип разработанного приложения, а название соответствует фразе «Setup Diary.exe». Размер инсталлятора – 36,5 МБ.



Рисунок 6.1 – Установочный файл приложения

Далее необходимо выбрать язык, на котором будет производиться установка (русский либо английский).

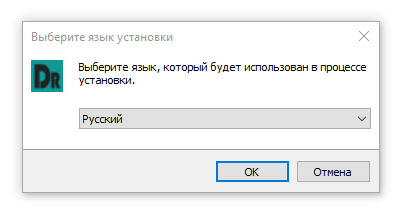


Рисунок 6.2 – Выбор языка установщика приложения

После выбора языка, установщик потребует пароль приложения. Он установлен лишь с целью защиты приложения от распространения без ведома разработчика. Пароль приложения – «BSUIR».

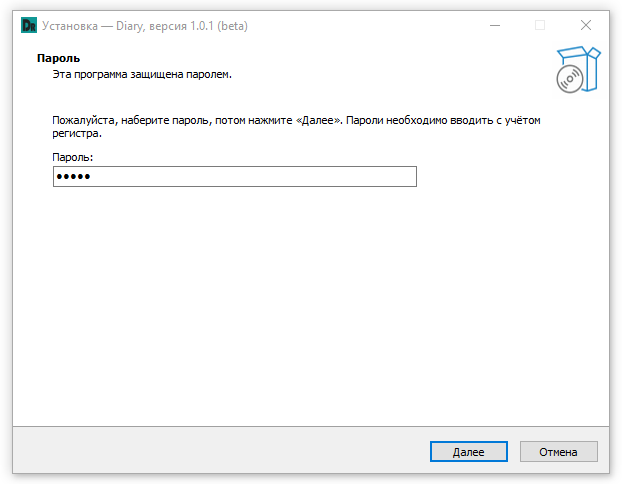


Рисунок 6.3 – Окно запроса пароля приложения

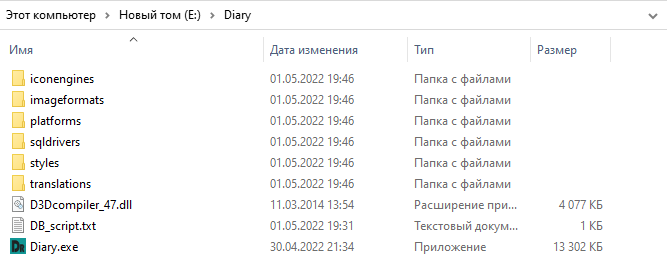


Рисунок 6.4 – Файл «DB\_script.txt», создаваемый после установки приложения

## **6.2 Создание базы данных PostgreSQL**

Первоначально, для работы с приложением необходимо создать новую базу данных для разработанного приложения. Ее можно создать, используя стандартные средства, предоставляемые Postgres, такими как pgAdmin (версии 4). Это стандартная утилита, поставляемая компанией Postgres, предоставляющая графический интерфейс для упрощенного управления базами данных. Для создания новой базы данных необходимо включить данную утилиту.

Если запуск pgAdmin будет первым, то приложение выведет на экран окно «Set Master Password», в котором мы должны задать «мастер-пароль». Это нужно для дополнительного шифрования паролей. После установки мастер-пароля все существующие сохраненные пароли будут повторно зашифрованы с использованием мастер-пароля.

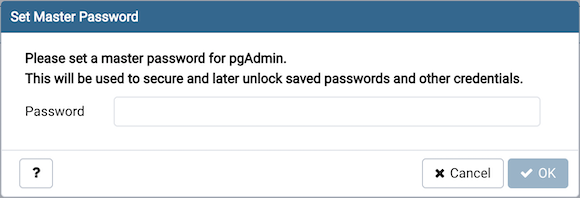


Рисунок 6.5 – Окно «Set Master Password» в pgAdmin 4

Если же вход в приложение не первый, то оно предложит ввести «мастер-пароль», выведя на экран окно «Unlock Saved Passwords».

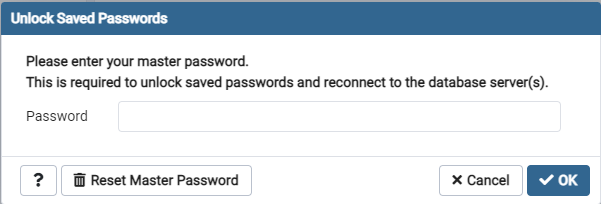


Рисунок 6.6 – Окно «Unlock Saved Passwords» в pgAdmin 4

После авторизации в приложении можно создавать новую базу данных. Для этого необходимо выполнить действия, пошагово продемонстрированные на рисунке 6.3:

Шаг 1. Правой кнопкой мыши нажать около слова «Databases» (1).

Шаг 2. В появившемся меню навести на кнопку «Create» (2).

Шаг 3. В новом меню выбрать пункт «Database…» (3).

Шаг 4. Далее, в новом окне (4), необходимо указать имя базы данных (DiaryDB), которое будет использоваться в дальнейшем (5).

Шаг 5. Последний же шаг, подтверждение создания новой базы данных.

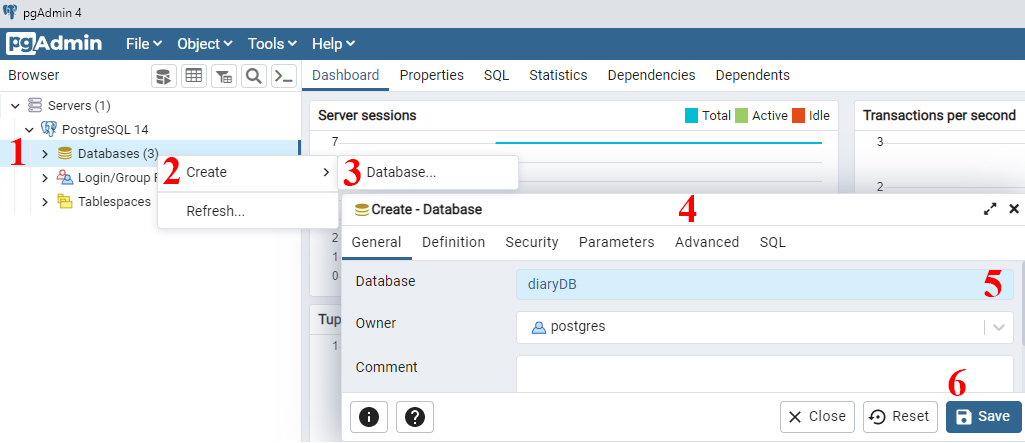


Рисунок 6.7 – Создание новой базы данных посредством pgAdmin 4

## **6.3 Создание пользователей базы данных PostgreSQL с правами доступа**

Следующий шаг - создание пользователей и ролей для последних. Производится это с помощью скрипта, который прописывается в панели «pgAdmin 4». Для этого выбирается база данных с именем «DiaryDB». Затем после нажатия левой кнопкой мыши по ее названию появляется дополнительна панель. Выбирается пункт «CREATE Script» и прописывается скрипт, описанный ниже. Также, данный скрипт добавлен в текстовый файл, поставляемый с установщиком приложения, с названием «DB\_script.txt».

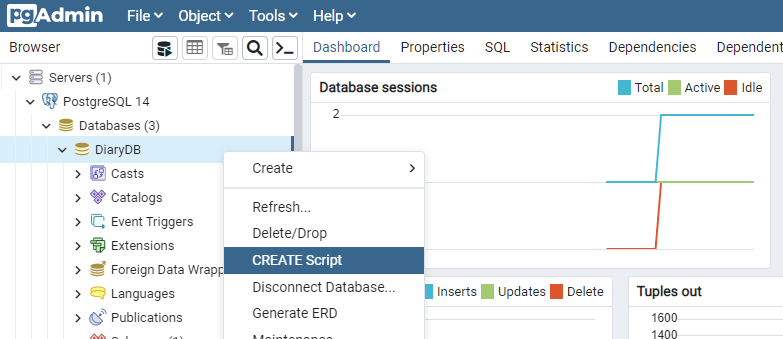


Рисунок 6.8 – Панель написания скриптов в pgAdmin 4

CREATE USER auth\_user WITH ENCRYPTED PASSWORD 'pgsecret';

GRANT USAGE ON SCHEMA public to auth\_user;

ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO auth\_user;

GRANT CONNECT ON DATABASE "DiaryDB" to auth\_user;

ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO auth\_user;

GRANT USAGE ON SCHEMA public to auth\_user;

GRANT SELECT ON ALL SEQUENCES IN SCHEMA public TO auth\_user;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO auth\_user;

CREATE USER teacher WITH ENCRYPTED PASSWORD 'teacherDBpassword';

GRANT USAGE ON SCHEMA public to teacher;

ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO teacher;

## **6.4 Создание таблиц и связей базы данных PostgreSQL с использованием Oracle SQL Developer Data Modeler**

Созданная заранее ER-модель базы данных преобразуется в реляционную модель согласно шагам, детально описываемым в методическом пособии к лабораторным работам [13]. В результате выполнения всех шагов, можно получить файл формата «DLL» (Dynamic Link Library — «библиотека динамической компоновки»). Выполнив преобразования (не все типы данных, существующие в приложении Oracle SQL Developer Data Modeler, поддерживаются другими приложениями, например, Data Grip) получаем следующий скрипт, по созданию базы данных. Ниже представлены несколько таблиц и связь между ними, чтобы не загромождать описываемый раздел. Полный скрипт будет описан в приложении.

CREATE TABLE ids (

student\_id INTEGER NOT NULL,

teacher\_id INTEGER NOT NULL

);

ALTER TABLE ids ADD CONSTRAINT ids\_pk PRIMARY KEY ( student\_id, id );

CREATE TABLE teacher (

full\_name\_t VARCHAR,

gender VARCHAR,

id INTEGER NOT NULL,

rank VARCHAR,

phone\_number VARCHAR,

subject\_id INTEGER NOT NULL,

birth\_date\_t DATE

);

ALTER TABLE teacher ADD CONSTRAINT teacher\_pk PRIMARY KEY ( id );

CREATE TABLE student (

full\_name\_st VARCHAR,

gender VARCHAR,

id INTEGER NOT NULL,

studying\_group\_id INTEGER NOT NULL,

birth\_date\_st DATE,

stud\_parent\_id INTEGER NOT NULL

);

ALTER TABLE student ADD CONSTRAINT student\_pk PRIMARY KEY ( id );

ALTER TABLE ids

ADD CONSTRAINT ids\_student\_fk FOREIGN KEY ( student\_id )

REFERENCES student ( id );

ALTER TABLE ids

ADD CONSTRAINT ids\_teacher\_fk FOREIGN KEY ( teacher\_id )

REFERENCES teacher ( id );

С помощью запроса, описанного ниже, можно увидеть, что таблицы действительно создались:

select \* from information\_schema.tables where table\_schema='public';

Результат выполнения запроса предоставлен на рисунке 6.6. Отображение оформлено с использованием консоли приложения DataGrip и выполнением вышеописанного запроса.

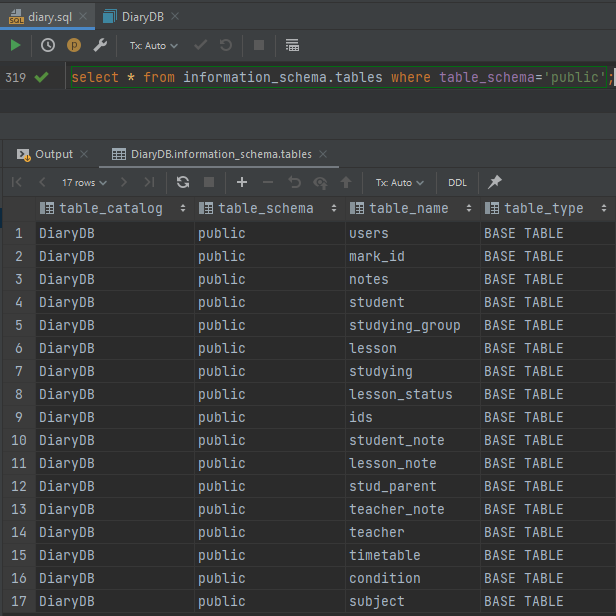


Рисунок 6.9 - Результат проверки создания таблиц в базе данных

Последний шаг по работе с базой данных – ее включение. Этот шаг необходим, если база данных отключалась, например в случае обновления оборудования. Сделать это можно с помощью команды

net start postgresql-x64-14

На данный момент полностью подготовлено для использования. В дальнейшем первый действием всегда будет – включить приложение (двойной клик по ярлыку приложения). После чего пользователь увидит информационное сообщение о дальнейших действиях: авторизация через сторонний сервис, предоставление выданного токена приложению. Данное действие упоминаться не будет, для уменьшения повторений в тексте.

## **6.5 Использование приложение «родителями»**

После авторизации в приложении пользователь увидит окно информации об его «ребенке» («учащемся»). Возможности «родителя» ограничиваются отображением полного имени, номера учебного класса, комментариев, отметок (как ежедневных, так и средних), контактной информации об учебном заведении.

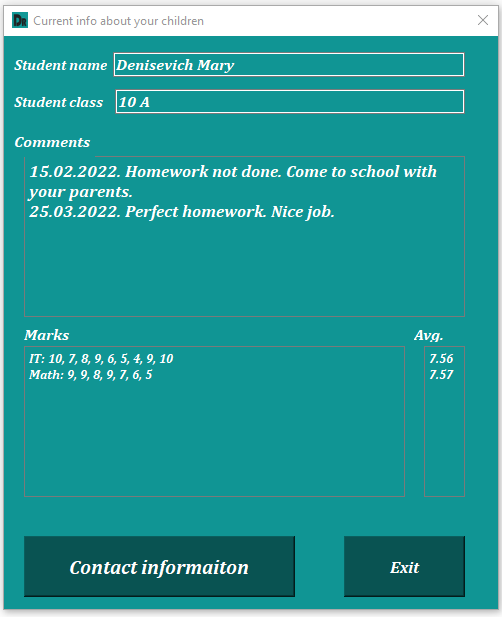


Рисунок 6.10 - Использование приложения «родителями»

Контактная информация может быть получена нажатием на кнопку с названием «Contact information».

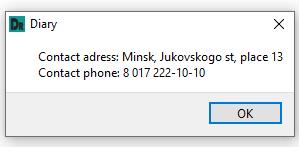


Рисунок 6.11 – Окно контактной информации для «родителей»

Для завершения работы с приложением необходимо нажать на кнопку «Exit» (рисунок 6.10) или «X» в правом углу экрана.

## **6.6 Использование приложение «учащимися»**

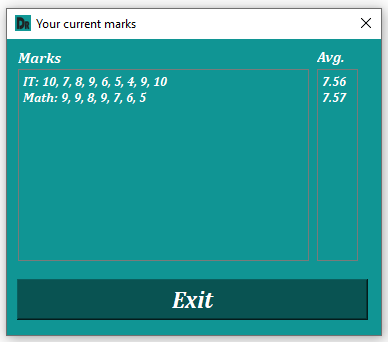


Рисунок 6.12 - Использование приложения «учащимися»

После авторизации «учащемуся» сразу отображается информация, доступная для него: отметки (ежедневные и средние).

Для завершения работы с приложением необходимо нажать на кнопку «Exit» (рисунок 6.12) или «X» в правом углу экрана.

## **6.7 Выставление отметок и комментариев «учащемуся»**

Процесс выставления отметки и комментария является одним из самых важных процессов разработанного приложения «Электронный журнал». Для его осуществления необходимо произвести некоторый набор действий:

Шаг 1. Нажать на кнопку (1) «Add marks» (рисунок 6.13).

Шаг 2. В поле (1) ввести имя «учащегося». После чего нажать на кнопку (2) «Accept» (рисунок 6.14).

Шаг 3. В поле (1) ввести значение отметки. В поле (2) по необходимости ввести комментарий к отметке. После чего подтвердить ввод нажатием на кнопку (3) «Evaluate student» (рисунок 6.15).

Шаг 4. Подтвердить выбор (рисунок 6.16).

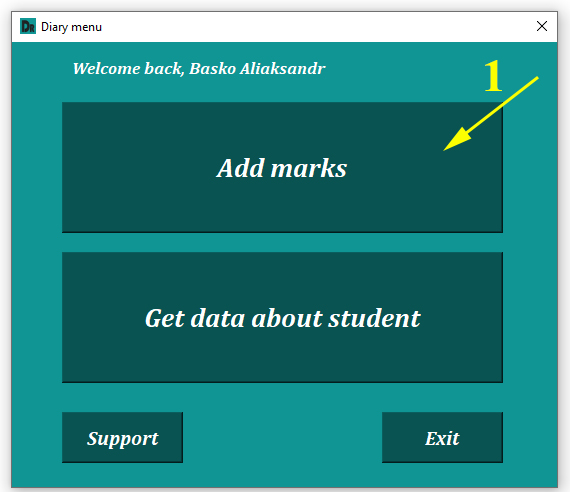


Рисунок 6.13 – Процесс выставления отметки. Шаг 1

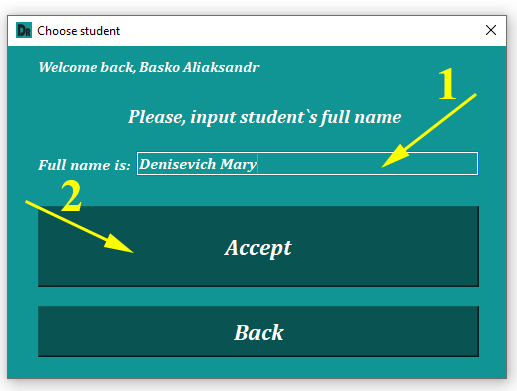


Рисунок 6.14 - Процесс выставления отметки. Шаг 2

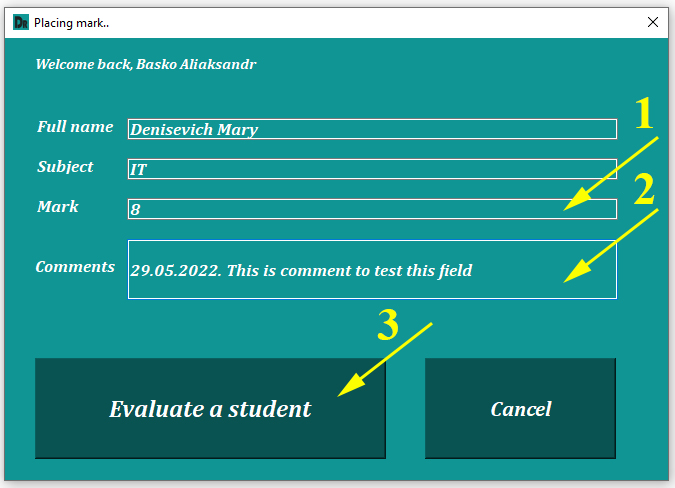


Рисунок 6.15 - Процесс выставления отметки. Шаг 3

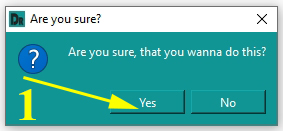


Рисунок 6.16 - Процесс выставления отметки. Шаг 4

## **6.8 Отображение информации о группе учащихся**

Для получения общей информации об учащихся работа пользователя сводится к минимуму, благодаря интуитивно понятному интерфейсу.

Шаг 1. Первым делом пользователю необходимо нажать на кнопку (1) «Get data about student». Рисунок 6.17.

Шаг 2. Следующей следует кнопка уточнения – «Data about class», позволяющая получить информацию о группе «учащихся». Рисунок 6.18.

Шаг 3. Далее необходимо конкретизировать выбор группы.

В случае, если необходимо получить информацию обо всех учащихся, представленных в базе данных, то необходимо сразу нажать на кнопку (3) «Find».

В случае, если же необходимо получить информацию об учащихся конкретной группы необходимо ввести в поле (1) «Class number» численное значение класса, а в поле (2) «Class letter» его буквенное обозначение. После чего подтвердить выбор с помощью кнопки (3) «Find».

Стоит отметить факты того, что не имеет значения регистр вводимого в поле буквенного обозначения (2). Также, в случае отсутствия информации об группе, пользователю будем предоставлено соответствующее сообщение.

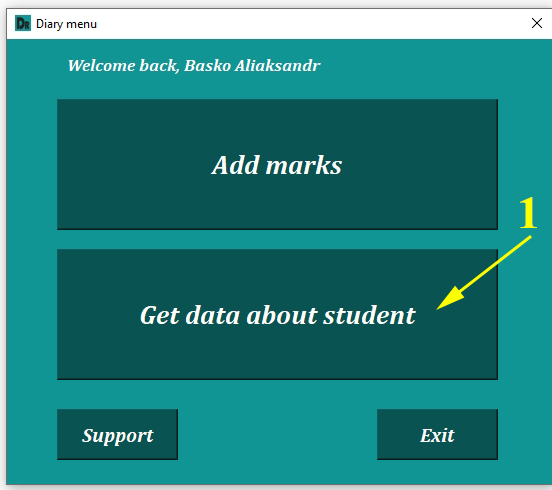


Рисунок 6.17 - Отображение информации о группе учащихся. Шаг 1

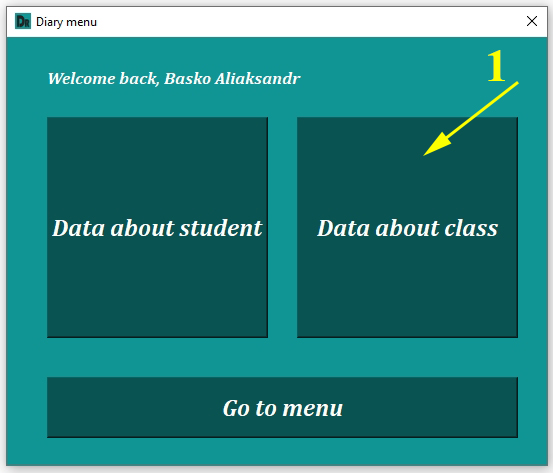


Рисунок 6.18 - Отображение информации о группе учащихся. Шаг 2

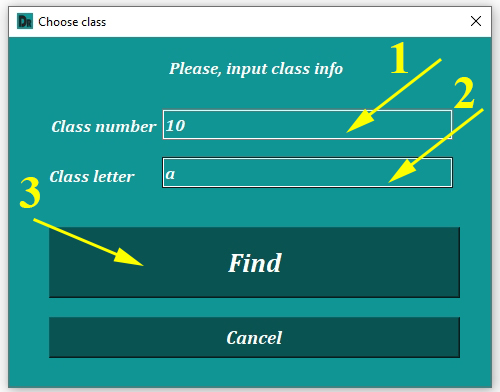


Рисунок 6.19 - Отображение информации о группе учащихся. Шаг 3

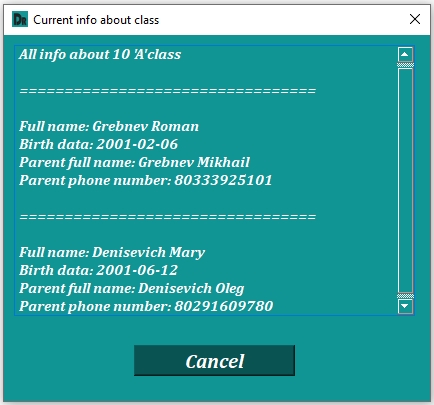


Рисунок 6.20 - Отображение информации об учащихся

## **6.9 Отображение информации о конкретном учащемся**

Процесс уточнения информации о конкретном учащемся также не предоставляет проблем для конечного пользователя.

Шаг 1. Первым делом пользователю необходимо нажать на кнопку (1) «Get data about student». Рисунок 6.17.

Шаг 2. Следующей следует кнопка уточнения – «Data about student», позволяющая получить информацию о конкретном учащемся. Рисунок 6.21.

Шаг 3. В поле (1) «Full name» необходимо ввести полное имя искомого учащегося. Ввод осуществляется в формате: «Фамилия Имя». После ввода необходимо подтвердить ввод, нажав на кнопку (2) «Find». Рисунок 6.22.

Результатом будет отображение подробной информации подобно изображенной на рисунке 6.23. В блоке (1) отображается информация об учащемся. В блоке (2) предоставлена информация о родителе учащегося. В отделе (3) отображаются все отметки учащегося, а в отделе (4) их среднее значение по каждому из учебных предметов. В графе (5) отображены комментарии преподавателей. Кнопка (6) отвечает за возврат на предыдущий экран (предыдущее меню).

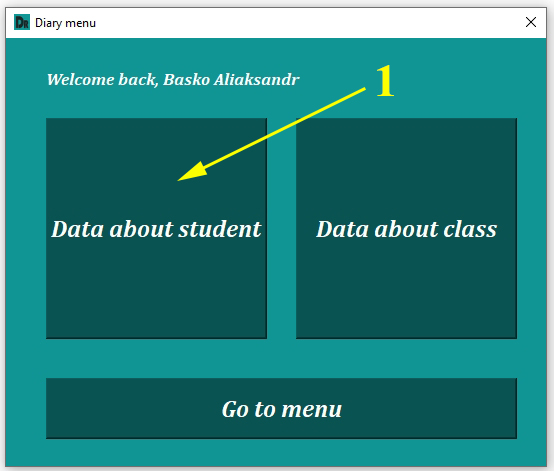


Рисунок 6.21 - Отображение информации о конкретном учащемся.

Шаг 2

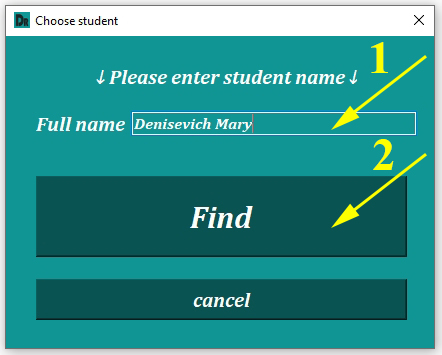


Рисунок 6.22 - Отображение информации о конкретном учащемся.

Шаг 3

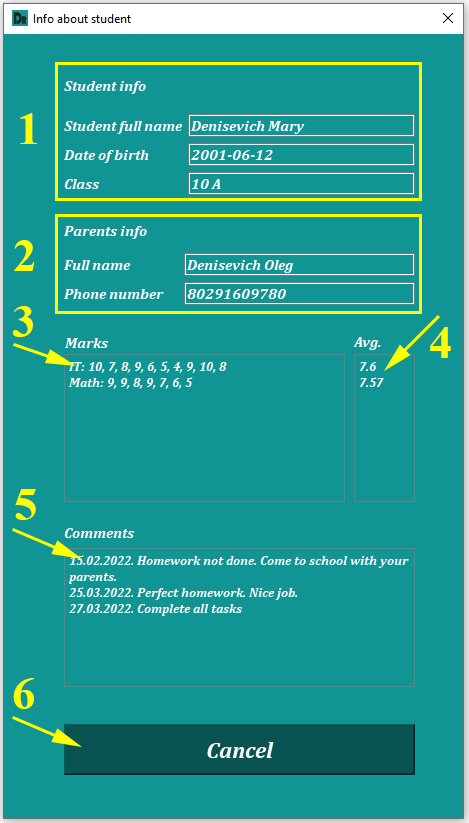


Рисунок 6.23 - Отображение информации о конкретном учащемся. Пример результата