

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет» РТУ МИРЭА**

Институт КБСП направление 09.03.02 «Информационные системы и технологии»

Кафедра КБ-14 «Цифровые технологии обработки данных» Дисциплина «Программные средства манипулирования данными»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе (проект) на тему:**

Разработка базы данных для сервиса по продаже игр

Студент Луговой И. И. Ф.И.О.

подпись, дата

Зарин Н. Н. Ф.И.О.

подпись, дата

.

Группа БСБО-05-20

Работа защищена на оценку

Руководитель работы Котилевец И.Д. Ф.И.О.

подпись, дата

Члены комиссии Ф.И.О.

подпись, дата

Ф.И.О.

подпись, дата

Москва 2022г.



**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет» РТУ МИРЭА**

Институт КБСП направление 09.03.02 «Информационные системы и технологии»

Кафедра КБ-14 «Цифровые технологии обработки данных»

Дисциплина «Программные средства манипулирования данными»

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студент: 3 курса, группа: БСБО-05-20

1. Тема: Разработка базы данных для сервиса по продаже игр
2. Срок представления проекта (работы) к защите г.
3. Исходные данные для разработки
4. Содержание пояснительной записки

Содержание Техническое задание Основная часть Графический интерфейс Заключение

Список использованных источников Приложения

Руководитель работы Котилевец И.Д. Ф.И.О.

подпись, дата

Задание принял к исполнению

подпись, дата

Ф.И.О.

Москва 2022г.

Оглавление

[Введение 2](#_Toc122762499)

[Глава 1. ER-модель 3](#_Toc122762500)

[Глава 2. Сущности 4](#_Toc122762501)

[Глава 3. Основная часть 10](#_Toc122762502)

[Глава 4. Реализация практической части курсовой работы 15](#_Toc122762503)

[Заключение 31](#_Toc122762504)

[Список используемых источников 32](#_Toc122762505)

# Введение

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, которая поддерживает одну или более областей применения. С понятием базы данных тесно связано понятие системы управления базой данных.

Целью данной курсовой работы является проектирование и реализация базы данных в указанной предметной области.

В качестве СУБД для реализации базы данных была использована СУБД реляционного типа - pgsql

В рамках курсовой работы были поставлены следующие задачи:

1. Анализ предметной области: сервис по продаже игр.
2. Проектирование БД в исследуемой предметной области (включающее проектирование структуры таблиц, ключевых полей и схемы данных).
3. Приобретение навыков по созданию запросов различных типов.
4. Создание GUI-приложения

# Изображение выглядит как текст, черный, табличка Автоматически созданное описаниеГлава 1. ER-модель

Рисунок 1.1 ER-модель

# Глава 2. Сущности

|  |  |  |
| --- | --- | --- |
| **№** | **Название сущности** | **Назначение** |
| **1** | user\_login | Содержит данные для авторизации пользователей |
| **2** | user\_info | Содержит данные о пользователях |
| **3** | user\_library | Содержит данные об играх, купленных пользователем |
| **4** | role | Содержит данные о ролях в БД |
| **5** | user\_role | Служит для связи пользователя и его роли |
| **6** | country | Содержит данные о странах |
| **7** | city | Содержит данные о городах |
| **8** | images | Содержит изображения, используемые сервисом |
| **9** | game | Содержит данные об играх доступных для покупки |
| **10** | genre | Содержит данные о жанрах игр |
| **11** | game\_genre | Связывает игры и их жарны |
| **12** | publisher | Содержит данные об издателях игр |
| **13** | game\_purchase | Содержит данные о транзакциях |

**2.1 Атрибуты и подробное описание каждой сущности**

|  |  |  |
| --- | --- | --- |
| user\_login | | |
| id | Serial | Id аккаунта |
| username | Varchar(128) | Логин пользователя |
| password | Varchar(256) | Хешированный пароль |

|  |  |  |
| --- | --- | --- |
| user\_info | | |
| id | Serial | Id пользователя |
| user\_id | Int | Ссылается на id аккаунта из другой сущности |
| first\_name | Varchar(128) | Имя пользователя |
| last\_name | Varchar(128) | Фамилия пользоватедя |
| country\_id | Int | Ссылается на id страны из другой сущности |
| city\_id | Int | Ссылается на id города из другой сущности |
| image\_id | Int | Ссылается на id картинки из другой сущности |

|  |  |  |
| --- | --- | --- |
| user\_library | | |
| id | Serial | Id библиотеки пользователя |
| user\_id | Int | Ссылается на id пользователя |
| game\_id | Int | Ссылается на id игры |

|  |  |  |
| --- | --- | --- |
| role | | |
| id | Serial | Id роли пользователя |
| name | Varchar(128) | Название роли |

|  |  |  |
| --- | --- | --- |
| user\_role | | |
| id | Serial | Id связи пользователь - роль |
| person\_id | Int | Id ссылается на пользователя |
| role\_id | Int | Id ссылается на роль |

|  |  |  |
| --- | --- | --- |
| country | | |
| id | Serial | Id страны |
| name | Varchar(128) | Название страны |

|  |  |  |
| --- | --- | --- |
| city | | |
| id | Serial | Id города |
| name | Varchar(128) | Название города |
| country\_id | Int | Id ссылается на страну, в которой находится город |

|  |  |  |
| --- | --- | --- |
| images | | |
| id | Serial | Id изображения |
| image | Text | Изображение в формате base64 |

|  |  |  |
| --- | --- | --- |
| game | | |
| id | Serial | Id игры |
| name | Varchar(128) | Название игры |
| description | Varchar(2048) | Описание игры |
| price | Float | Стоимость игры |
| publisher\_id | int | Id ссылается на издателя игры |

|  |  |  |
| --- | --- | --- |
| dt\_release | Timestamp | Дата релиза игры |
| image\_id | int | Id ссылается на изображение |

|  |  |  |
| --- | --- | --- |
| genre | | |
| id | Serial | Id жанра |
| name | Varchar(128) | Название жанра |

|  |  |  |
| --- | --- | --- |
| game\_genre | | |
| id | Serial | Id связи игра - жанр |
| game\_id | int | Id ссылается на игру |
| genre\_id | int | Id ссылается на жанр |

|  |  |  |
| --- | --- | --- |
| publisher | | |
| id | Serial | Id издателя |
| name | Varchar(128) | Название издателя |
| country\_id | int | Id ссылается на страну издателя |
| user\_id | int | Id ссылается на пользовательский аккаунт пользователя |

|  |  |  |
| --- | --- | --- |
| game\_purchase | | |
| id | Serial | Id транзакции |
| game\_id | Int | Id ссылается на игру |
| user\_id | Int | Id ссылается на пользователя, совершившего покупку |
| timestamp | Timestamp | Id ссылается на жанр |

# Глава 3. Основная часть

**3.1 Анализ предметной области**

Задачей курсовой работы является создание базы данных для поиска работы. В ходе выполнения поставленных задач на основе написанной базы данных были написаны десктопное и мобильное приложение, первое из которых выступает в роли админ панели для управления содержимым магазина, второе является основной сервиса, через который происходит покупка игр.

В системе существует 4 типа пользователей:

* + 1. Администратор (admin)
    2. Пользователь (user)
    3. Издатель (publisher)
    4. Роль для регистрации (reg\_master)

Каждая роль имеет свои привилегии, а пользователям, которые появляются в системе, присваивается своя определенная роль. Роли обладают своими возможностями, которые разграничены с помощью функций.

Программное обеспечение системы должно осуществлять выполнение следующего функционала:

* Навигация по базе данных и просмотр содержания таблиц
* Создание, удаление и редактирование записей
* Регистрация и авторизация пользователей с различными ролями
* Иметь встроенную защиту от sql-инъекций
* Хэширование паролей

Подробное описание функционала каждой роли:

Перед определением возможностей ролей необходимо ограничить доступ к таблицам следующими командами:

ALTER TABLE user\_login ENABLE ROW LEVEL SECURITY;

ALTER TABLE user\_info ENABLE ROW LEVEL SECURITY;

ALTER TABLE user\_role ENABLE ROW LEVEL SECURITY;

ALTER TABLE role ENABLE ROW LEVEL SECURITY;

ALTER TABLE images ENABLE ROW LEVEL SECURITY;

ALTER TABLE game ENABLE ROW LEVEL SECURITY;

ALTER TABLE publisher ENABLE ROW LEVEL SECURITY;

ALTER TABLE genre ENABLE ROW LEVEL SECURITY;

ALTER TABLE game\_genre ENABLE ROW LEVEL SECURITY;

ALTER TABLE user\_library ENABLE ROW LEVEL SECURITY;

ALTER TABLE country ENABLE ROW LEVEL SECURITY;

ALTER TABLE city ENABLE ROW LEVEL SECURITY;

ALTER TABLE game\_purchase ENABLE ROW LEVEL SECURITY;

Основной ролью является пользователь. Он может взаимодействовать лишь с определенным с определенными таблицами и записями в них, ему запрещен INSERT во все таблицы.

CREATE POLICY select\_id ON user\_login FOR SELECT TO "user"

USING

(

LOWER(username) = current\_user

);

CREATE POLICY update\_user\_login ON user\_login FOR UPDATE TO "user", publisher

USING

(

LOWER(username) = current\_user

)WITH CHECK (true);

CREATE POLICY select\_user\_info ON user\_info FOR SELECT TO "user", publisher

USING

(true);

CREATE POLICY update\_user\_info ON user\_info FOR UPDATE TO "user", publisher

USING

(true)WITH CHECK (true);

CREATE POLICY select\_user\_library ON user\_library FOR SELECT TO "user", publisher

USING

(

(SELECT LOWER(username) FROM user\_login WHERE user\_login.id = user\_library.user\_id) = current\_user

);

CREATE POLICY select\_role ON role FOR SELECT TO "user"

USING

(

name = 'user'

);

Издатель по сути своей является обычным пользователем, и необходима эта роль упрощения взаимодействия с издателями.

Роль для регистрации необходима для корректного и безопасного добавления пользователей в БД. Ей ограничен доступ в большинство таблиц.

GRANT SELECT ON get\_countries TO reg\_master;

GRANT SELECT ON user\_login TO reg\_master;

GRANT INSERT ON user\_login TO reg\_master;

GRANT SELECT ON user\_info TO reg\_master;

GRANT INSERT ON user\_info TO reg\_master;

GRANT SELECT ON user\_role TO reg\_master;

GRANT INSERT ON user\_role TO reg\_master;

GRANT SELECT ON role TO reg\_master;

GRANT SELECT ON country TO reg\_master;

GRANT SELECT ON city TO reg\_master;

От роли регистратора выполняется следующая процедура

CREATE OR REPLACE PROCEDURE create\_user(login VARCHAR(128), password\_text VARCHAR(256), first\_name\_text VARCHAR(128), last\_name\_text VARCHAR(128), country TEXT, city TEXT)

AS $$

DECLARE

id\_login INTEGER;

id\_info INTEGER;

id\_role INTEGER;

BEGIN

IF (SELECT COUNT(\*) FROM user\_login WHERE username = login) THEN

RAISE EXCEPTION 'Такой пользователь уже существует';

ELSE

INSERT INTO user\_login(username, password) VALUES (login, crypt(password\_text, gen\_salt('md5'))) RETURNING id INTO id\_login;

INSERT INTO user\_info(user\_id, first\_name, last\_name, country\_id, city\_id, image\_id)

VALUES (get\_user\_id\_by\_login(login), first\_name\_text, last\_name\_text, get\_county\_id(country), get\_city\_id(city), 1) RETURNING id INTO id\_info;

INSERT INTO user\_role(person\_id, role\_id) VALUES (get\_user\_id\_by\_login(login), get\_role\_id('user')) RETURNING id INTO id\_role;

EXECUTE format('CREATE USER %I WITH ENCRYPTED PASSWORD %L', LOWER(login), password\_text::VARCHAR);

EXECUTE format('GRANT "user" to %I', LOWER(login));

IF (id\_info IS NULL OR id\_login IS NULL OR id\_role IS NULL) THEN

ROLLBACK;

RAISE EXCEPTION 'Что-то пошло не так, попробуйте снова';

ELSE

COMMIT;

END IF;

END IF;

END

$$LANGUAGE plpgsql;

Админ имеет полный доступ к базе данных. Он может добавлять, изменять, удалять необходимые данные.

CREATE POLICY adminAll ON user\_login FOR All TO admin

USING (true);

CREATE POLICY adminAll ON user\_info FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON user\_role FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON user\_library FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON role FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON images FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON game FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON publisher FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON genre FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON game\_genre FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON country FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON city FOR ALL TO admin

USING (true);

CREATE POLICY adminAll ON game\_purchase FOR ALL TO admin

USING (true);

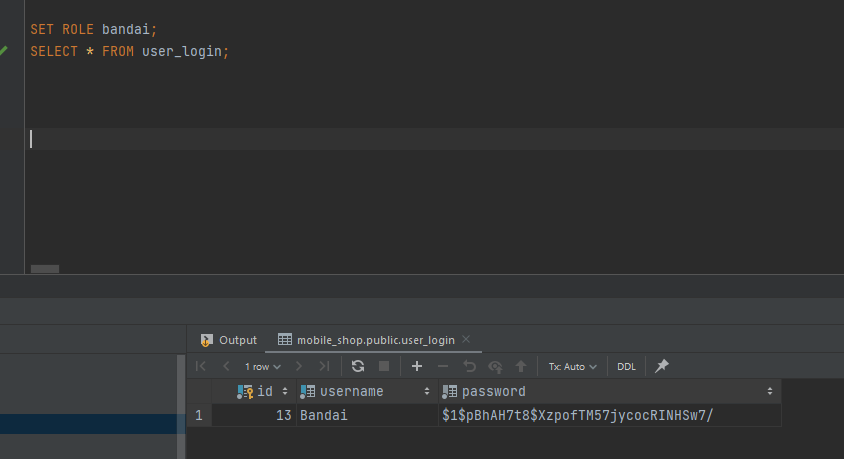
Пример того, что роли имеют ограниченный функционал

Рис. 3.1

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.2

# Глава 4. Реализация практической части курсовой работы

* 1. **Выбор технологий разработки системы**

После проектирования базы данных переходим к реализации пользовательского GUI. Для построения графического интерфейса было решено использовать C# с фреймворком DevExpress для работы с формами и Android Studio для десктопного и мобильного приложений, соответственно.

* 1. **Защита от SQL-инъекций**

Для защиты от SQL-инъекций мы реализовали весь функционал базы данных с помощью функций. Это позволило нам строго ограничить возможности пользователей и, соответственно, потенциальных злоумышленников. Помимо это были использованы параметризированные запрос и prepared statements, что позволило нам еще сильнее обезопасить БД.

* 1. **Вспомогательный функционал для работы с базой данных и веб-сайтом**

Функция для авторизации пользователя

CREATE OR REPLACE FUNCTION auth\_user\_get\_id(login\_text VARCHAR, password\_text VARCHAR)

RETURNS INTEGER

LANGUAGE plpgsql AS

$func$

DECLARE

ret INTEGER;

BEGIN

SELECT id INTO ret FROM user\_login WHERE username = login\_text AND password = crypt(password\_text, password);

IF (ret = 0) THEN

RAISE EXCEPTION 'Неверные данные';

ELSE

RETURN ret;

END IF;

END

$func$;

Процедура добавления игры

CREATE OR REPLACE PROCEDURE create\_game(name\_text VARCHAR(128), description\_text VARCHAR(2048), price\_text FLOAT, publisher\_text VARCHAR(128), date\_release TIMESTAMP, image\_base64 TEXT)

AS $$

DECLARE

image\_id\_ret INTEGER;

game\_id\_ret INTEGER;

BEGIN

IF (SELECT COUNT(\*) FROM game WHERE name = name\_text) THEN

RAISE EXCEPTION 'Такая игра уже существует';

ELSE

INSERT INTO images(image) VALUES (image\_base64) RETURNING id INTO image\_id\_ret;

INSERT INTO game(name, description, price, publisher\_id, dt\_release, image\_id) VALUES (name\_text, description\_text, price\_text, get\_publisher\_id\_by\_title(publisher\_text), date\_release, image\_id\_ret) RETURNING id into game\_id\_ret;

IF (image\_id\_ret IS NULL OR game\_id\_ret IS NULL) THEN

RAISE EXCEPTION 'Что-то пошло не так, попробуйте снова';

END IF;

END IF;

END

$$LANGUAGE plpgsql;

Триггер на проверку дубликатов жанра

CREATE TRIGGER check\_for\_duplicates

BEFORE INSERT OR UPDATE on genre

FOR EACH ROW EXECUTE FUNCTION check\_for\_duplicates();

CREATE OR REPLACE FUNCTION check\_for\_duplicates() RETURNS TRIGGER AS

$$BEGIN

IF(0 = (SELECT COUNT(\*) FROM genre WHERE name LIKE NEW.name)) THEN

RETURN NEW;

ELSE RAISE EXCEPTION 'Такой жанр уже есть в базе данных';

END IF;

END

$$ LANGUAGE plpgsql;

Индексы для поля name в таблице city

CREATE INDEX ON city (name);

**4.3 Разработка пользовательского интерфейса**

На данном скриншоте представлена главная страница мобильного приложения – магазин.

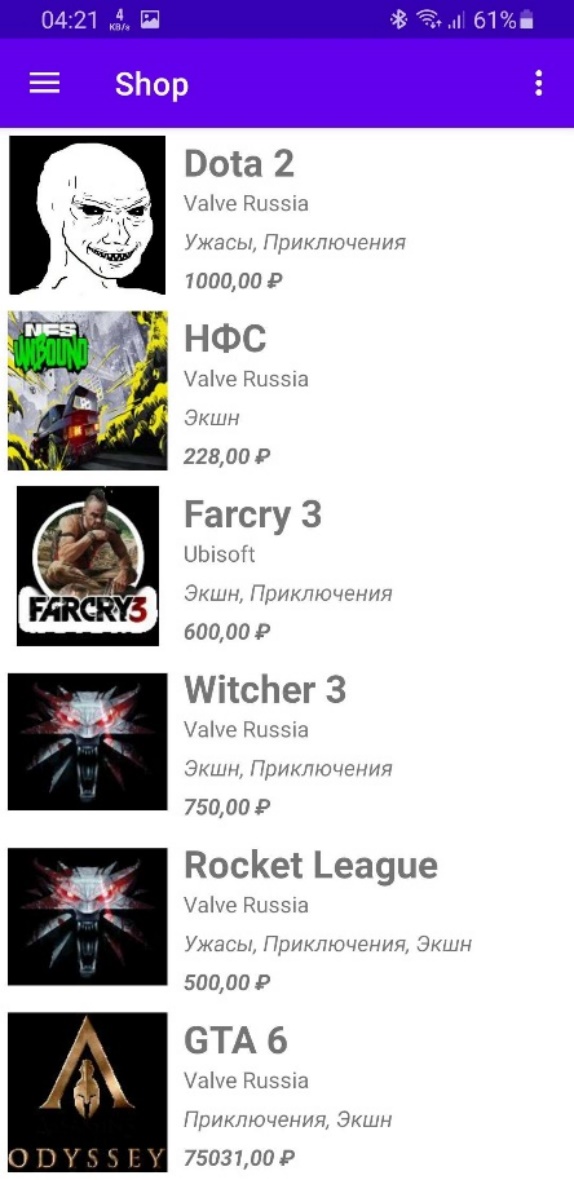


Рис. 4.1

Нажав на интересующую игру, можно открыть ее страницу.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 4.2

Впоследствии ее можно приобрести.



Рис. 4.3

После удачной покупки игра будет добавлена в библиотеку пользователя.

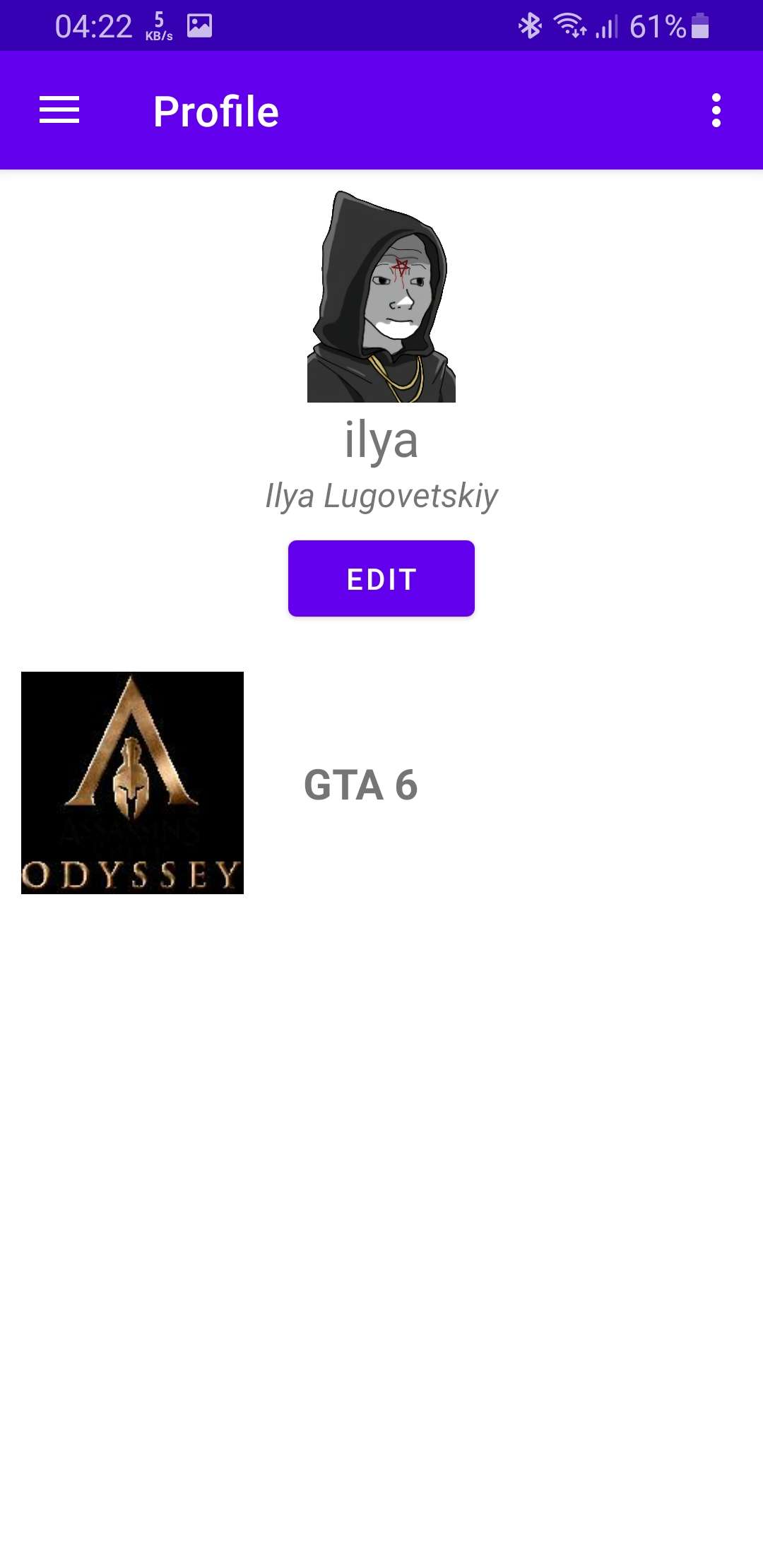


Рис. 4.4

Пользователь может изменять свой профиль.

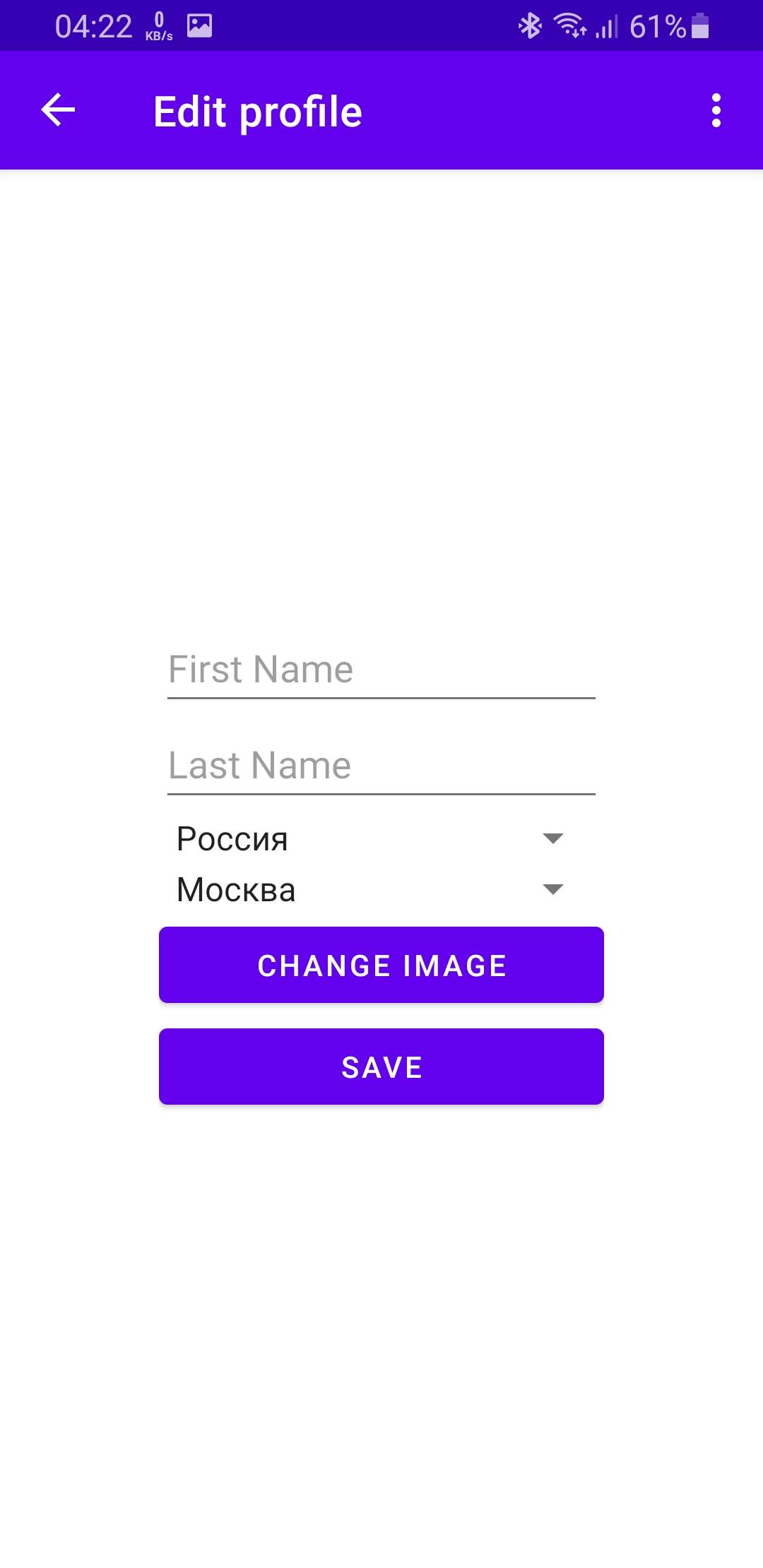


Рис. 4.5

Окно навигации выглядит следующим образом.

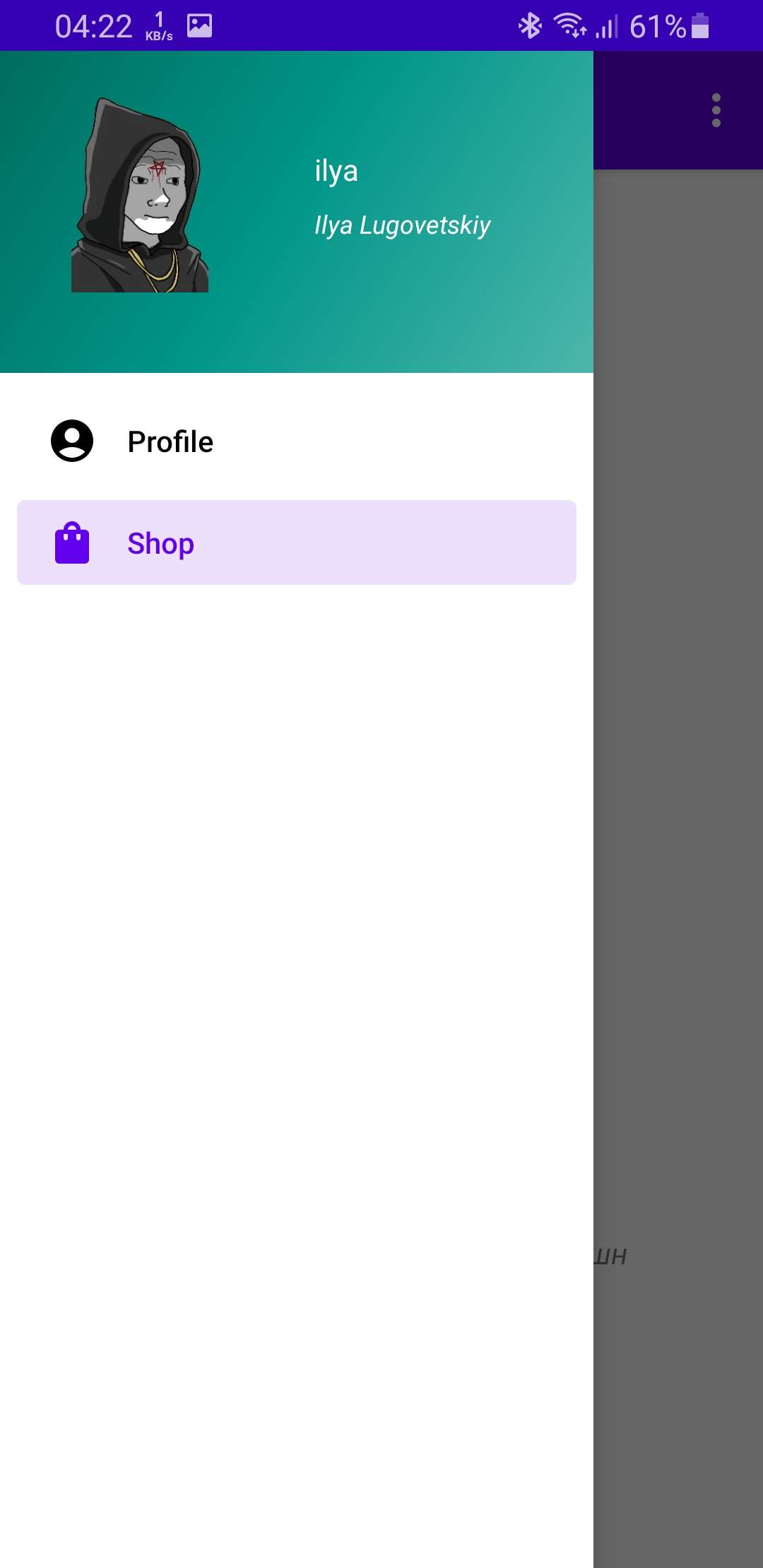


Рис. 2.6

Окна регистрации и авторизации.

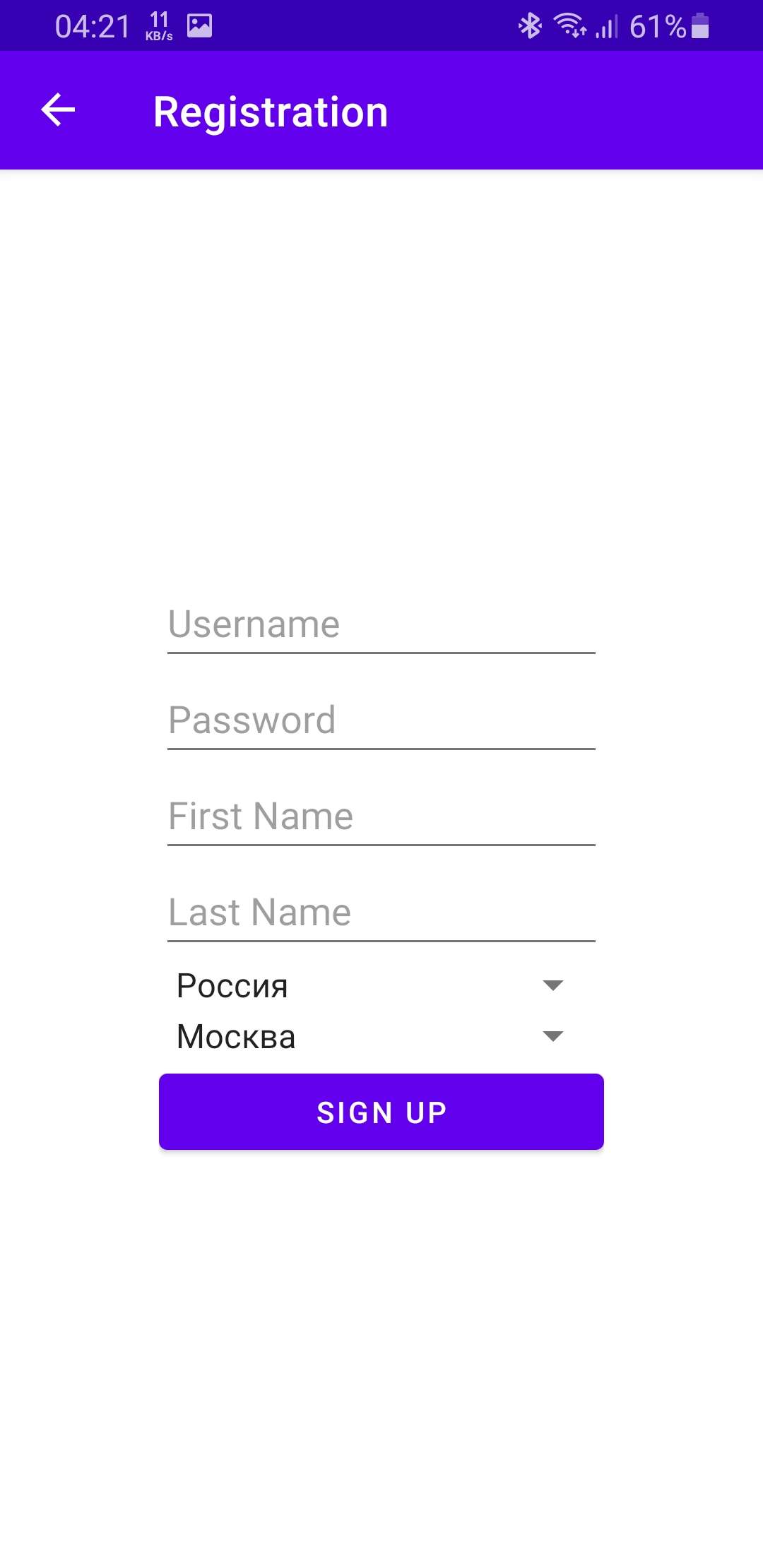


Рис. 2.7

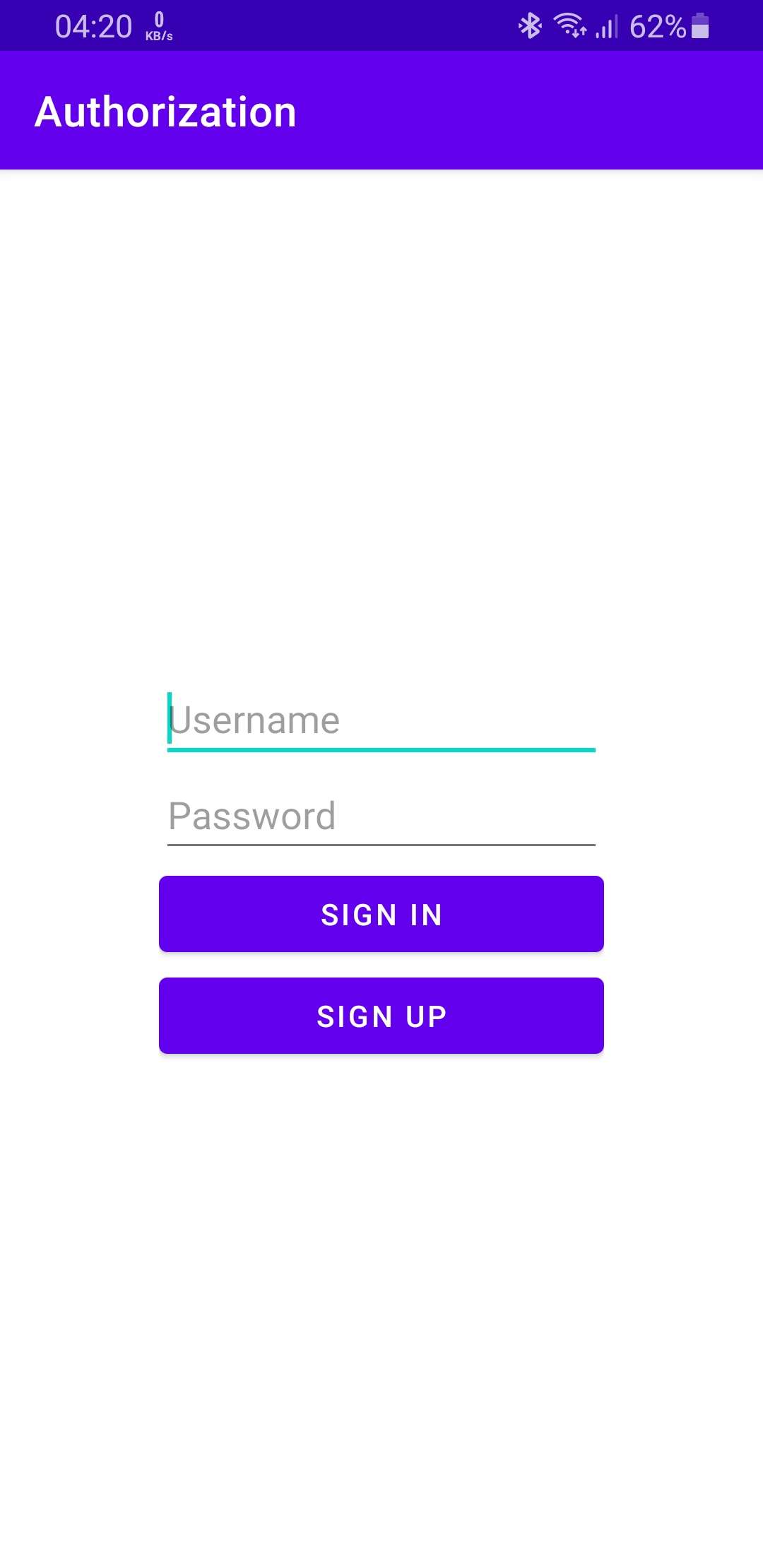


Рис. 2.8

Главное меню десктопного приложения отражает в себе весь его основной функционал.

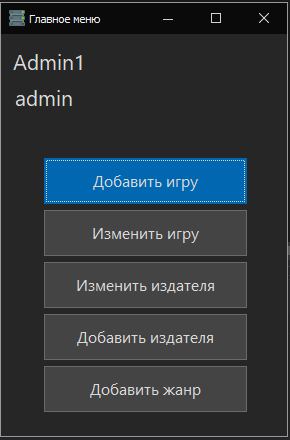


Рис. 2.9

Доступом к этому приложению обладает только админ. Основную задачу, которое оно решает – это добавление и изменения данных в БД. Администратор может добавлять/изменять игры, добавлять/изменять издателей, добавлять жанры.

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

Рис. 2.10

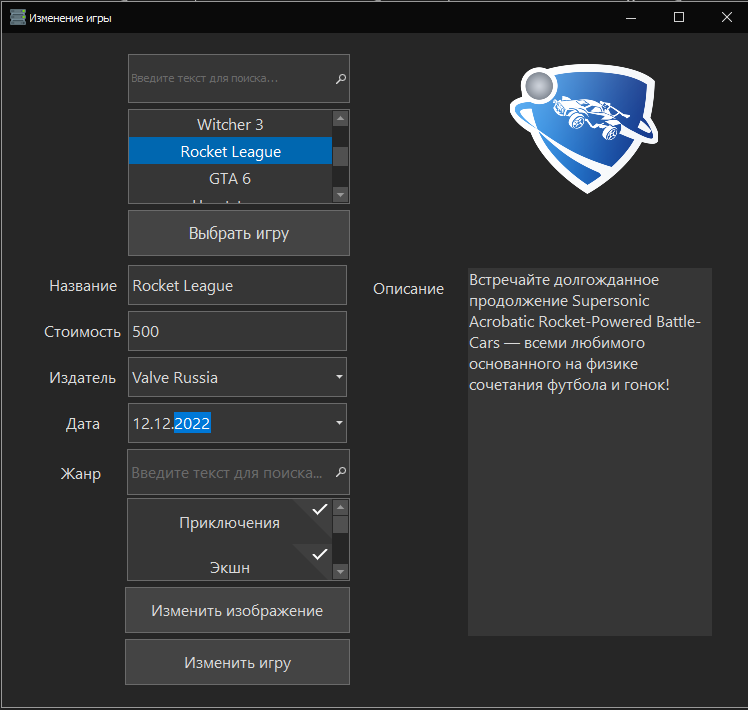


Рис. 2.11

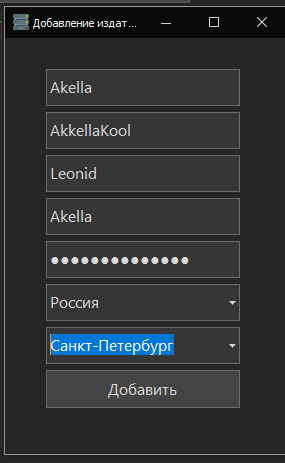


Рис 2.12

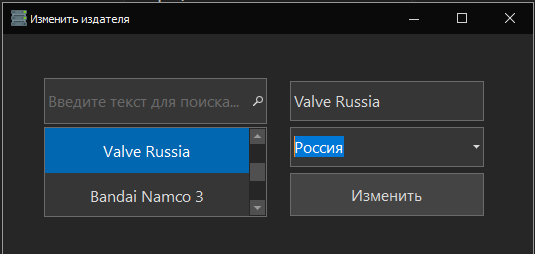


Рис. 2.13

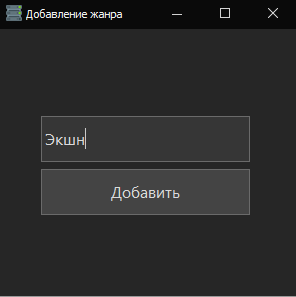


Рис 2.14

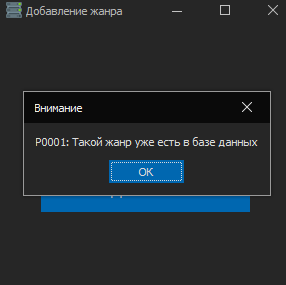


Рис. 2.15

Окно авторизации выглядит следующим образом.

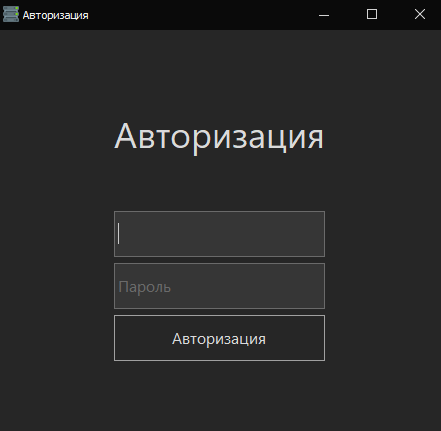


Рис. 2.16

# Заключение

В рамках курсового проекта были выполненные следующие, ранее поставленные задачи:

* Анализ предметной области: hr-сайт в области информационных технологий
* Проектирование базы данных в исследуемой предметной области (включает в себя проектирование структуры таблиц, ключевых полей и схемы данных)
* Приобретение навыков по созданию простых и составных запросов
* Создание GUI в виде веб-сайта

**Ссылки на репозитории с кодом для работы приложений:**

* **https://github.com/LZ-Software/MobileShop**

# Список используемых источников

1. PostgreSQL: Документация [Электронный ресурс] / URL: https://postgrespro.ru/docs/postgresql (дата обращения: 12.12.2021).
2. Стоунз PostgreSQL. Основы / Стоунз, Мэттью Ричард; Нейл. - М.: СПб: Символ-Плюс, 2002. - 302 c.
3. Ригс, Саймон Администрирование PostgreSQL 9. Книга рецептов Саймон Ригс , Ханну Кросинг. - М.: ДМК Пресс, 2015. - 291 c.
4. Аллан Бьюли “Изучаем SQL” – M.: Символ-Плюс, 2016 – 179 с.