



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 _____ Ф.И.О.
подпись, дата

Оглавление

Введение	2
Глава 1. ER-модель.....	3
Глава 2. Сущности.....	4
Глава 3. Основная часть.....	10
Глава 4. Реализация практической части курсовой работы.....	15
Заключение.....	31
Список используемых источников	32

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 - postgresql

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

Глава 1. ER-модель

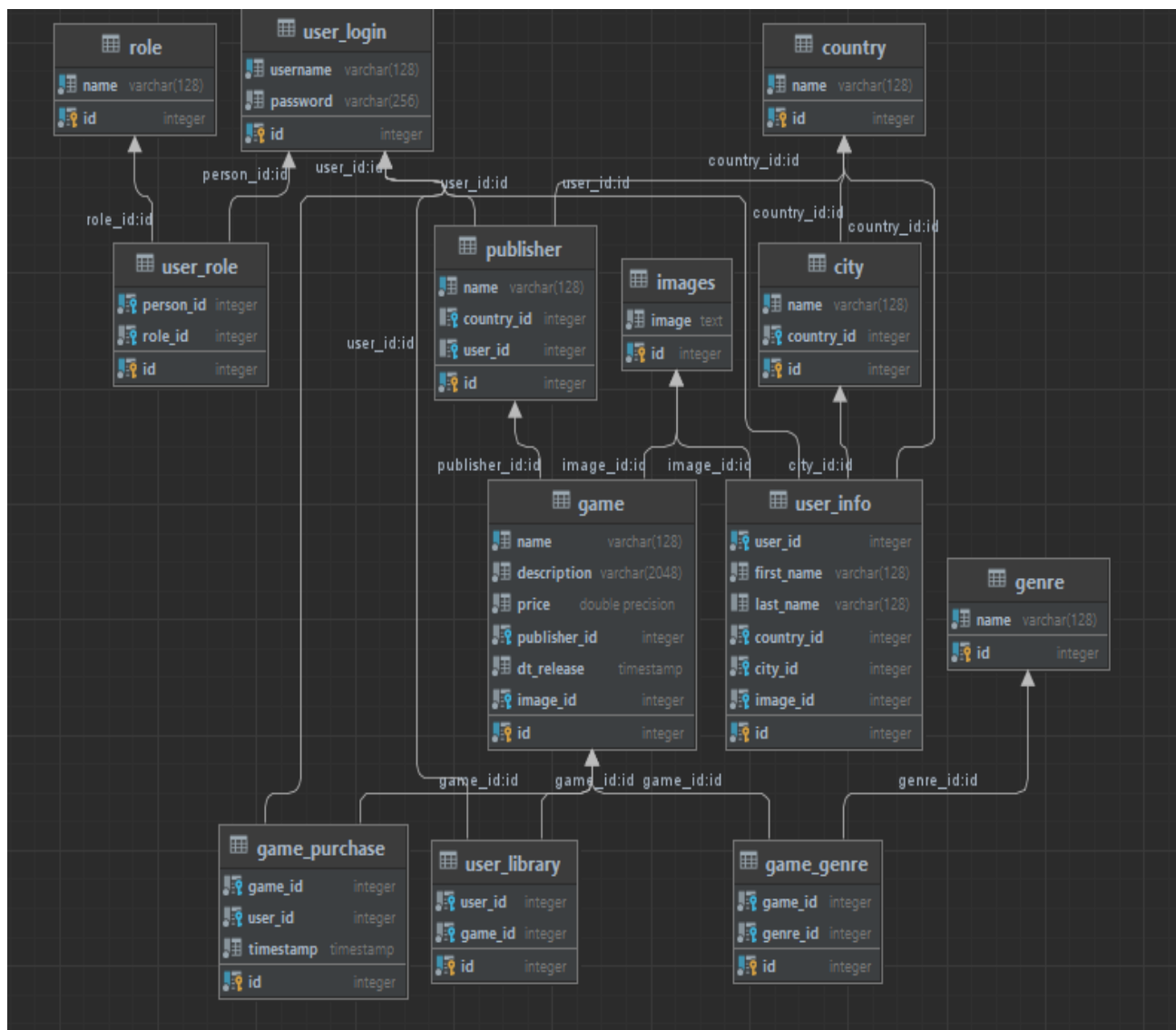


Рисунок 1.1 ER-модель

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 жанры
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 пользователя
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

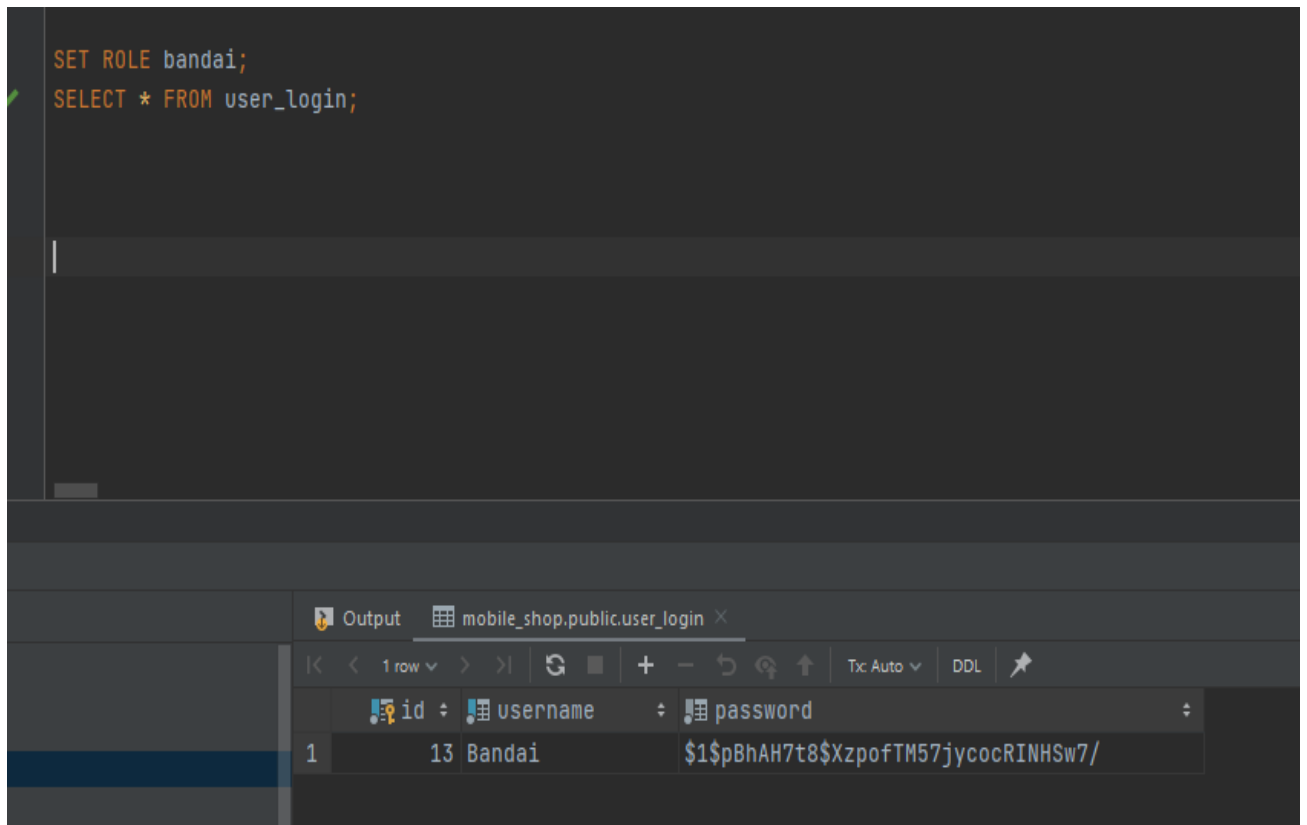


Рис. 3.1

```
-- auto-generated definition  
create user bandai;  
  
grant publisher to bandai;
```

Рис. 3.2

Глава 4. Реализация практической части курсовой работы

4.1. 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.

4.2. 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 параметризованные запрос и prepared statements, что позволило нам еще сильнее обезопасить БД.

4.3. 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.

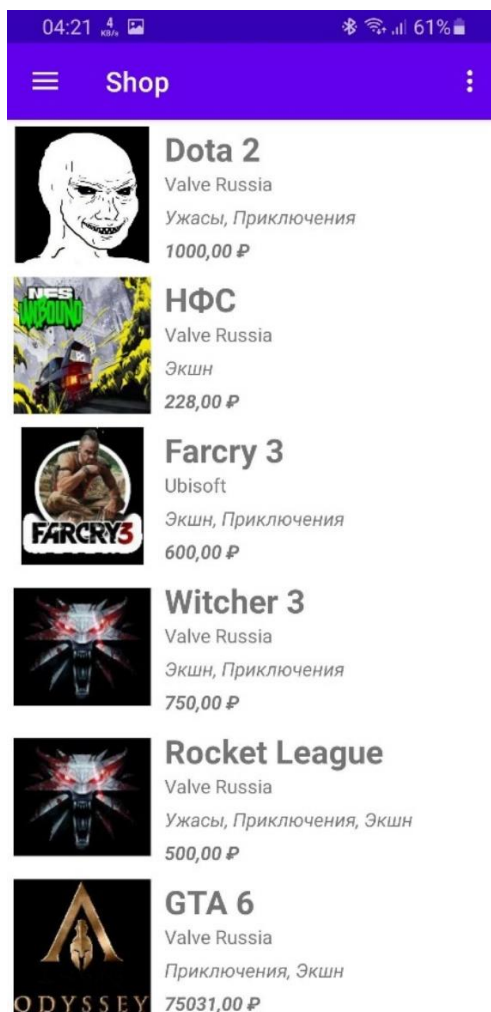


Рис. 4.1

Нажав на интересующую игру, можно открыть ее страницу.



Рис. 4.2

Впоследствии ее можно приобрести.



Witcher 3

1111222233334444

01/01 123

PAY 750,00 ₺

Рис. 4.3

После удачной покупки игра будет добавлена в библиотеку пользователя.

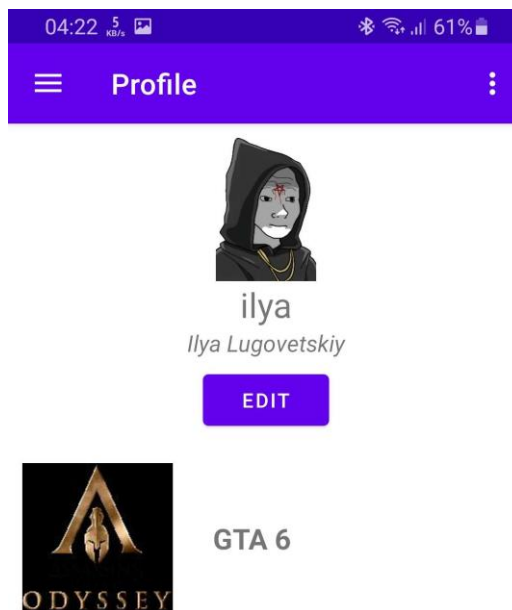


Рис. 4.4

Пользователь может изменять свой профиль.

04:22 0 KB/s 61%

← Edit profile ⋮

First Name

Last Name

Россия ▼

Москва ▼

CHANGE IMAGE

SAVE

Рис. 4.5

Окно навигации выглядит следующим образом.

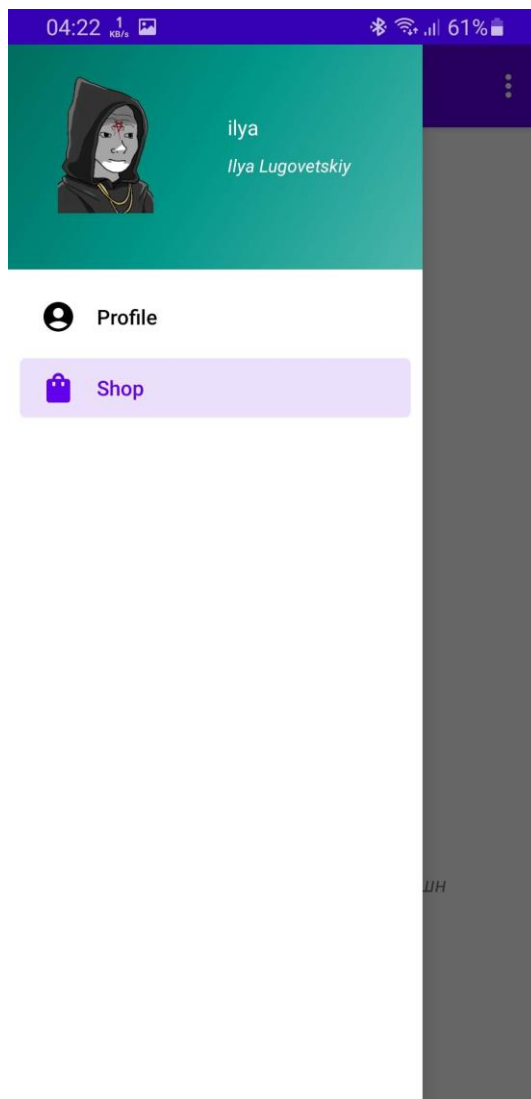


Рис. 2.6

Окна регистрации и авторизации.

04:21 11 KB/s

Bluetooth, Wi-Fi, Cellular, 61%

← Registration

Username

Password

First Name

Last Name

Россия ▼

Москва ▼

SIGN UP

Рис. 2.7



Username

Password

SIGN IN

SIGN UP

The form consists of two text input fields, one for 'Username' and one for 'Password', each with a light gray border and a small blue cursor icon at the start. Below these fields are two blue rectangular buttons with rounded corners. The top button is labeled 'SIGN IN' and the bottom button is labeled 'SIGN UP', both in white, uppercase, sans-serif font.

Рис. 2.8

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

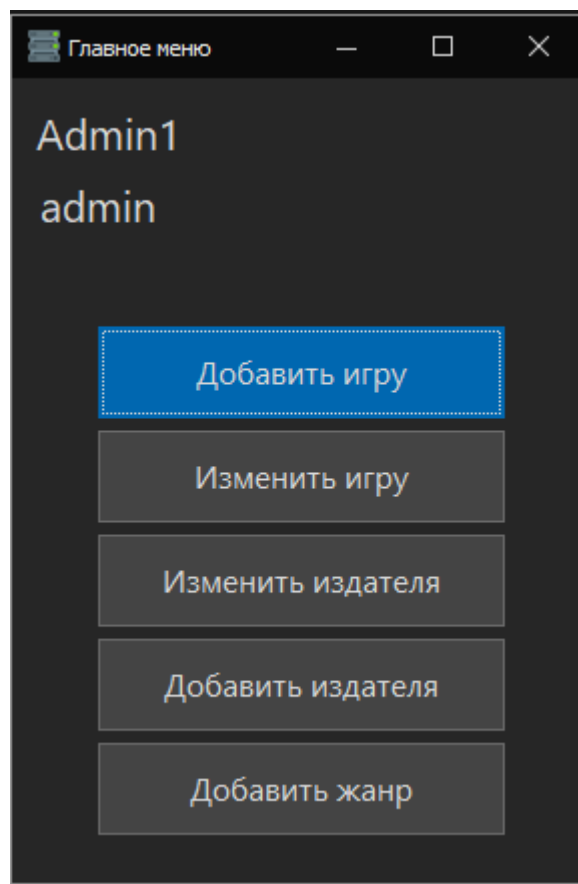
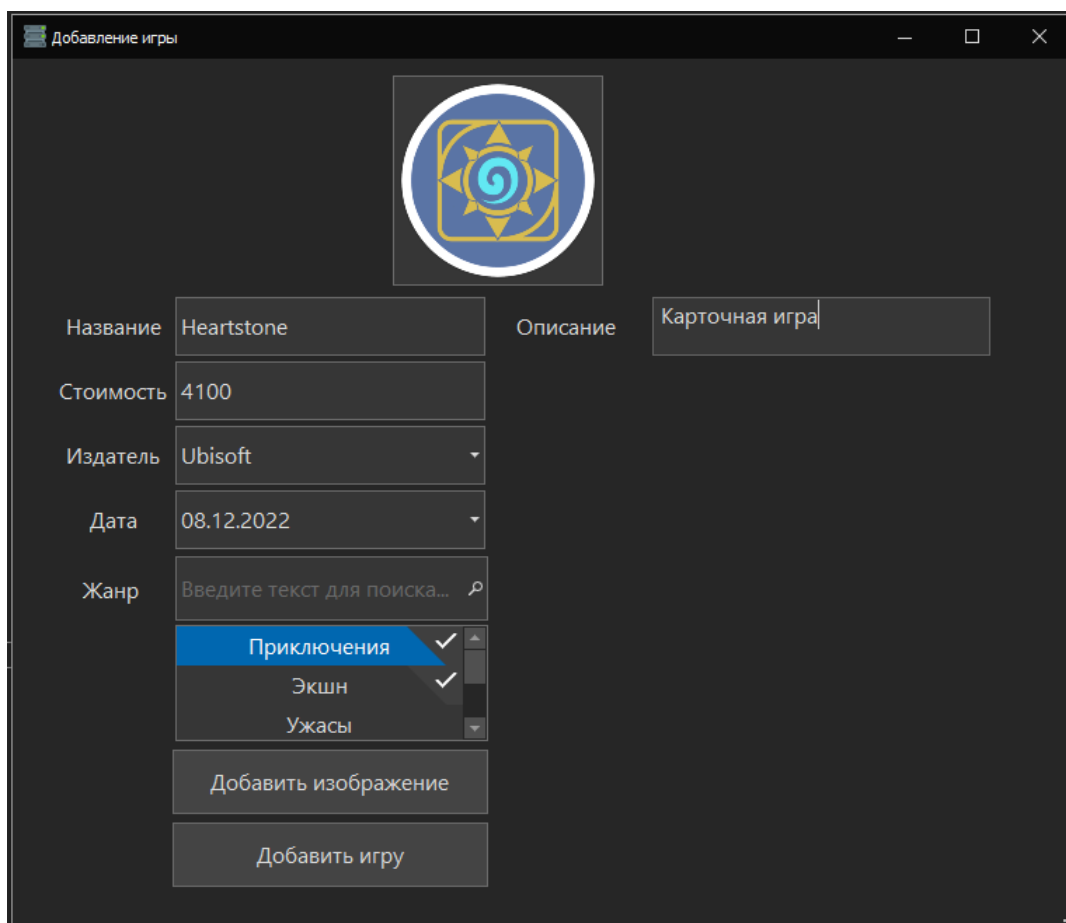


Рис. 2.9

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.



Добавление игры

Название: Heartstone

Описание: Карточная игра

Стоимость: 4100

Издатель: Ubisoft

Дата: 08.12.2022

Жанр: Приключения, Экшн, Ужасы

Добавить изображение

Добавить игру

Рис. 2.10

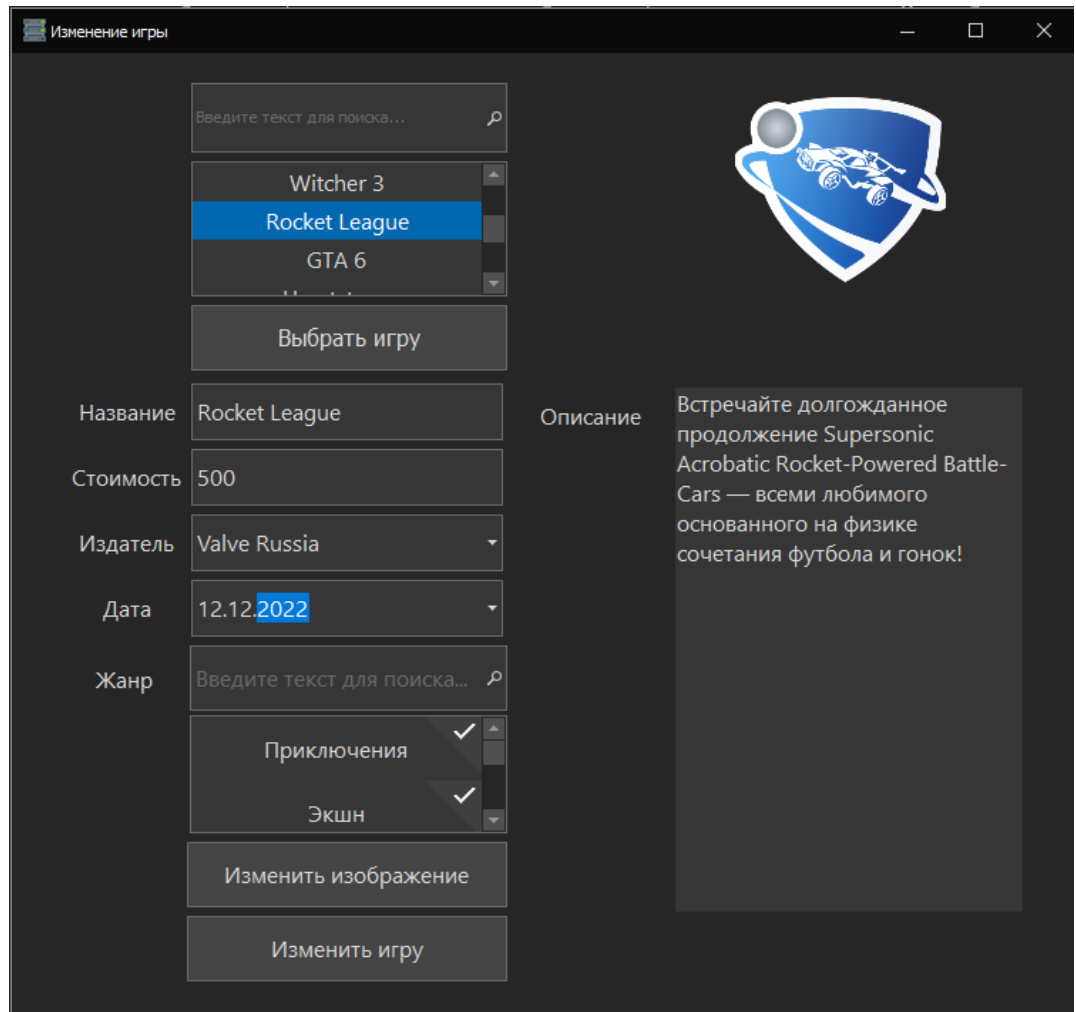


Рис. 2.11

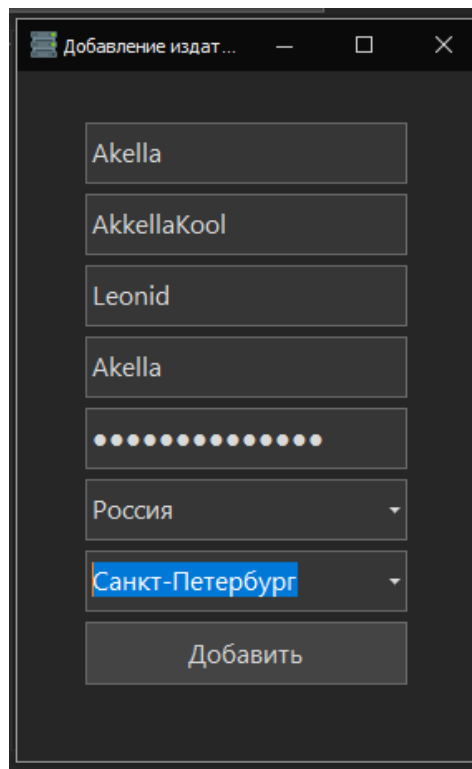


Рис 2.12

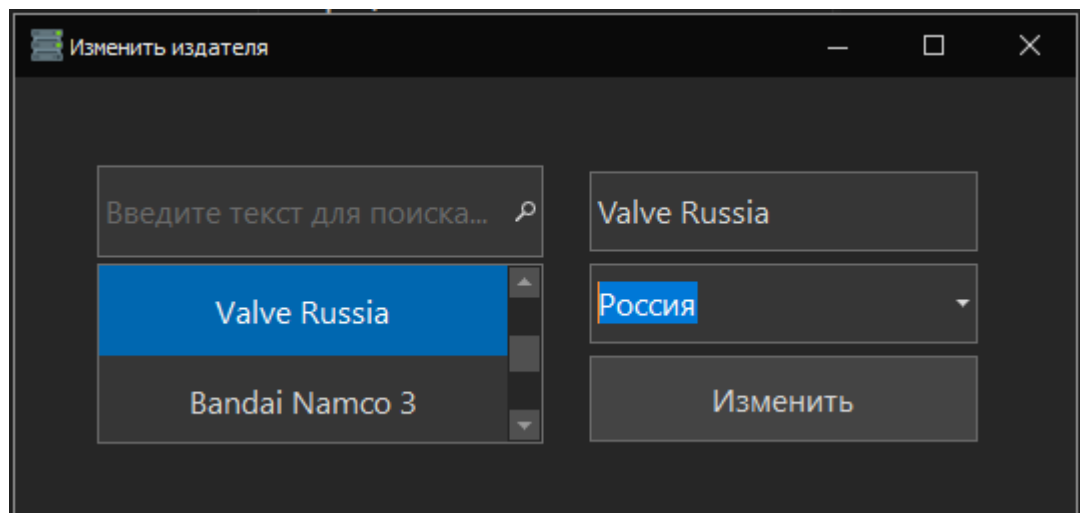


Рис. 2.13

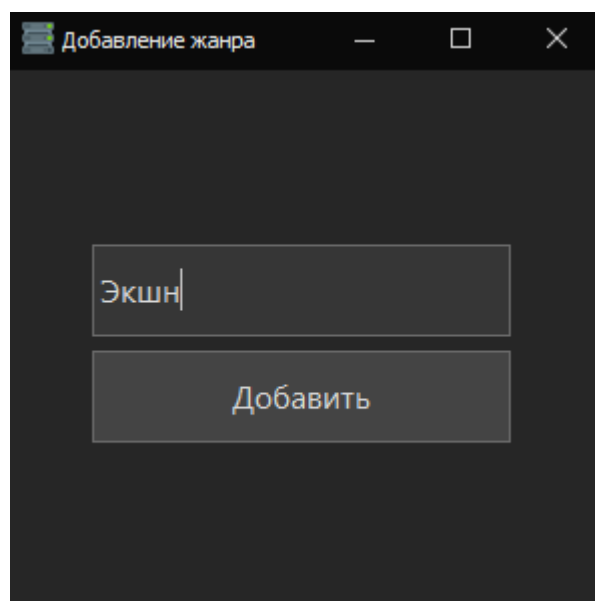


Рис 2.14

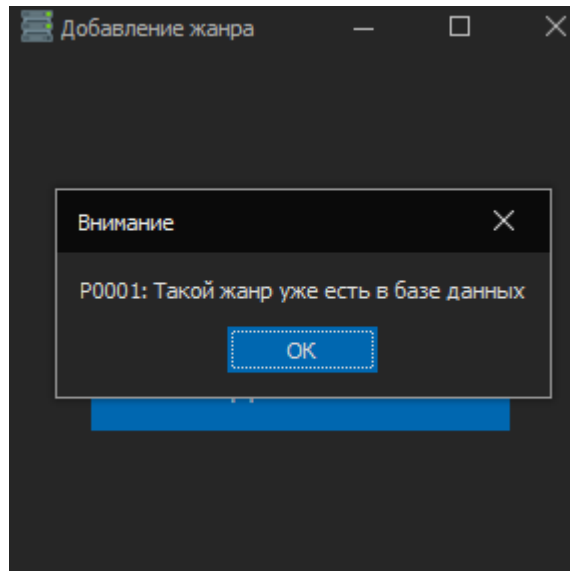


Рис. 2.15

Окно авторизации выглядит следующим образом.

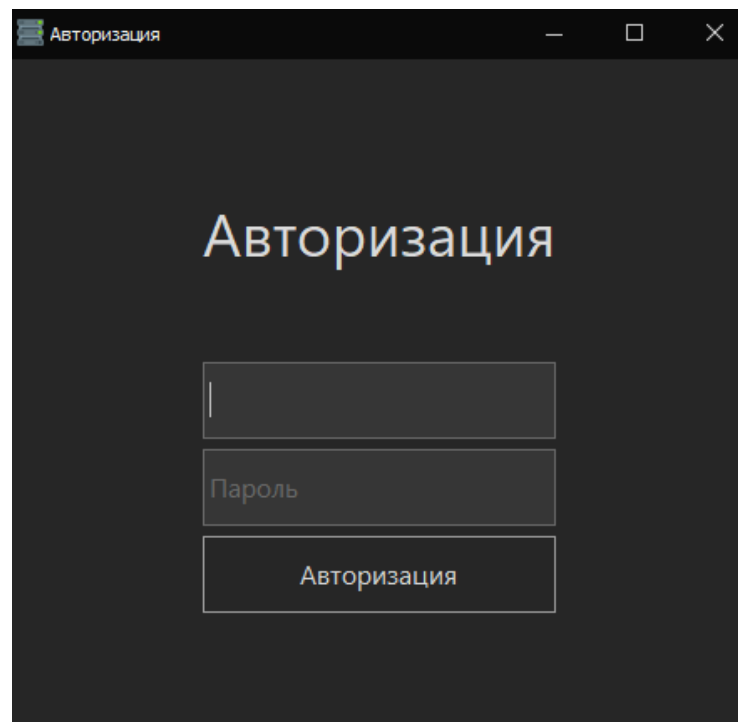


Рис. 2.16

Заключение

В рамках курсового проекта были выполнены 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**](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 с.
3. Ригс, Саймон Администрирование PostgreSQL 9. Книга рецептов Саймон Ригс , Ханну Кросинг. - М.: ДМК Пресс, 2015. - 291 с.
4. Аллан Бьюли “Изучаем SQL” – М.: Символ-Плюс, 2016 – 179 с.