

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Каталог данных Роспатента

Студент гр. 9382	_____	Субботин М.О.
Студент гр. 9382	_____	Кодуков А.В.
Студентка гр. 9382	_____	Круглова В.Д.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург

2022

ЗАДАНИЕ

Студенты:

Субботин М.О.

Круглова В.Д.

Кодуков А.В.

Группа 9382

Тема работы: разработка веб-приложения каталога данных Роспатента

Исходные данные:

База данных патентов в формате csv. Требуется реализовать веб-приложение каталога данных Роспатента с возможностью поиска и фильтрации данных.

Содержание пояснительной записки:

“Введение”, “Качественные требования к решению”, “Сценарии использования”, “Модель данных”, “Разработанное приложение”, “Выводы”, “Приложения”, “Литература”

Предполагаемый объем пояснительной записки:

Не менее 19 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 20.12.2022

Дата защиты реферата: 20.12.2022

Студент

Субботин М.О.

Студентка

Круглова В.Д.

Студент

Кодуков А.В.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В данном проекте предполагалась разработка веб-приложения с применением СУБД. Для темы “Каталог данных Роспатента” использовалась СУБД MongoDB. Было разработано веб-приложение, бэкенд которого написан на языке Python с использованием фреймворка Flask.

SUMMARY

This project was supposed to develop a web application using a DBMS. MongoDB DBMS was used for the topic “Rospatent Data Catalog”. A web application was developed, the backend of which is written in Python using the Flask framework.

СОДЕРЖАНИЕ

	Введение	6
1.	Качественные требования к решению	6
2.	Сценарии использования	7
3.	Модель данных	12
4.	Разработанное приложение	17
	Выводы	19
	Приложения	19
	Литература	19

ВВЕДЕНИЕ

Научные сотрудники пользуются базами данных патентов. Существующие веб-сервисы хранения таких баз данных не предоставляют возможности проведения различных операций с информацией.

Для решения этой проблемы предлагается разработать веб-приложение, которое удовлетворит потребность научных сотрудников в проведении поиска, фильтрации и скачивании данных.

Бэкенд веб-приложения будет написан на языке Python с использованием фреймворка Flask, а фронтенд на html с javascript скриптами. В качестве СУБД будет применяться MongoDB.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

- 1) Веб-приложение должно использовать СУБД MongoDB.
- 2) Наличие входа в систему (логин и пароль).
- 3) Возможность поиска по данным и фильтрации данных.
- 4) Возможность импорта и экспорта данных.
- 5) Возможность добавления новых записей в базу данных.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

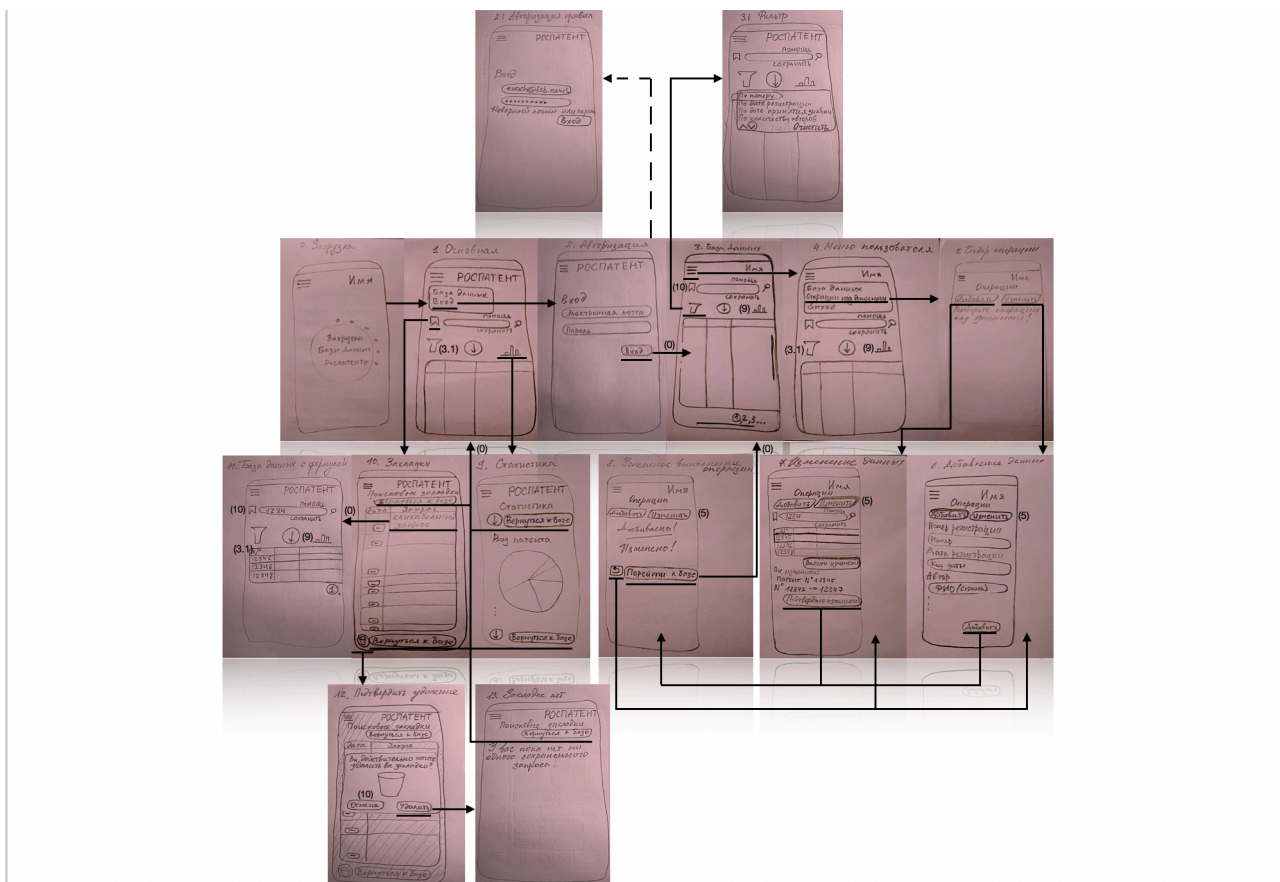


Рисунок 1. Макет UI

1. Вход пользователя в веб-приложение (представление данных)

Отображение: 0 -> 1

Действия пользователя: Пользователь переходит по ссылке на главную страницу Роспатента

Предусловия: Пользователь еще не авторизован

Основной сценарий: Пользователь видит индикатор загрузки, после чего для него открыта база данных с возможностью поиска и меню с кнопкой Вход

Альтернативные сценарии: Ошибка соединения -> 404

Результат: Пользователь видит базу данных в виде таблицы и меню, может авторизоваться либо пользоваться общими функциями: искать в базе с помощью поиска по полям, фильтровать данные, сохранять и просматривать сохраненные комбинации поиска и фильтрации, сохранять данные в отфильтрованном виде, смотреть и сохранять статистику. База данных загружена и показана, основные возможности доступны

2. Авторизация пользователя

Отображение: 1 -> 2 -> 0 -> 3

Действия пользователя: Пользователь нажимает на Вход в выпадающем меню и вводит свои данные и нажимает кнопку Вход

Предусловия: Пользователь еще не авторизован, на странице База данных

Основной сценарий: Пользователь видит поля для ввода электронной почты и пароля, нажимает кнопку Вход и после загрузки видит базу данных со всеми возможностями неавторизованного пользователя и некоторыми доступными только для тех, кто вошел

Альтернативные сценарии:

- Ошибка соединения -> 404
- Некорректный ввод логина или пароля -> на странице с авторизацией очищается поле ввода пароля и под ним появляется надпись «Неверный логин или пароль»

Результат: Пользователь видит базу данных, в правом углу экрана написано его имя, он может использовать все функции, доступные без авторизации. В добавок к этому в меню появляются Опции над данными, где авторизованный пользователь сможет добавлять или изменять данные в базе, а также, кнопка Вход -> Выход. Пользователь авторизован, ему доступны основные и особые возможности

3. Импорт данных

3.1. Добавление патента вручную

Отображение: 3 -> 4 -> 5 -> 6 -> 8

Действия пользователя: Пользователь открывает меню и нажимает на Опции над данными выбирает вариант “Добавить” и вводит информацию о патенте, после чего нажимает кнопку Добавить

Предусловия: Пользователь авторизован, на странице База Данных

Вводимые данные:

- Дата регистрации (строка) (ггггммдд)
- Дата принятия заявки (строка) (ггггммдд)
- Количество авторов (натуральное число)
- Авторы (строка) (Иванов И.И. (RU)), допустимо (Иванов Иван Иванович (SU))
- Количество правообладателей (Натуральное число)
- Правообладатели (строка) (Полное наименование вида организации «Название организации» (RU)) / (Иванов Иван Иванович (RU))
- Контакты третьих лиц (строка) Опционально
- Название программы (строка)
- Год создания (строка) (гггг) Опционально
- Дата публикации регистрации (строка) (ггггммдд)
- Количество попыток публикации (натуральное число)
- Актуальность (указывается галочкой)

Основной сценарий: Пользователь видит страницу Опции, где у него есть выбор добавить или изменить патент. Он нажимает добавить и видит поля для ввода информации, куда он вводит все необходимые данные и нажимает кнопку Добавить, после чего видит строку «Добавлено!»

Альтернативные сценарии:

- Ошибка соединения -> 404
- Не заполнено какое-то из полей -> рядом с кнопкой Добавить появляется надпись «Необходимо заполнить все поля, помеченные знаком *», также под полем, которое было пропущено, появится надпись «Необходимо заполнить это поле»

Результат: Пользователь видит строку «Добавлено!», может вернуться к базе данных, вернуться к редактированию данных о только что добавленном патенте или перейти к добавлению или изменению других. Новый патент добавлен в базу данных.

3.2. Импорт данных из файла.

На главной странице пользователь жмет на кнопку выбора файла, выбирает, затем жмет на кнопку загрузки файла на сайт, при обновлении страницы новые данные выведены в таблице.

4. Экспорт отфильтрованной базы данных

Отображение: 1 -> 10 -> 0 -> 11

Действия пользователя: Пользователь нажимает на кнопку закладки и ищет среди них последнюю добавленную формулу, нажимает на нее, затем нажимает на фильтр, выбирает нужный и нажимает скачать

Предусловия: Пользователь может быть не авторизован, на странице База данных

Основной сценарий: Пользователь видит таблицу со списком сохраненных запросов, кликая на нужный, после загрузки перемещается на страницу База данных с уже вставленной искомой формулой, показывающую данные по выбранному запросу. Нажимая на фильтр, видит варианты фильтрации и нажимает на нужный, после чего видит отфильтрованную запрошенную базу данных. Пользователь нажимает на кнопку скачать и видит, как у него в браузере началась загрузка csv файла

Альтернативные сценарии:

- Ошибка соединения -> 404
- В базе данных нет патентов, которые можно было бы найти по выбранной формуле -> вместо таблицы выводится строка «По вашему запросу ничего не найдено». При попытке скачать базу в таком случае иконка скачивания начинает трястись при нажатии (на 3 секунды загорается красным)

Результат: Пользователь видит базу данных, найденных по его запросу, в отфильтрованном виде. Файл с нужными пользователю патентами загружен на его компьютер в формате csv

5. Экспорт статистики патентов

Отображение: 1 -> 9

Действия пользователя: Пользователь нажимает на иконку статистики, нажимает на кнопку скачивания

Предусловия: Пользователь может быть не авторизован, на странице База данных

Основной сценарий: Пользователь видит страницу Статистика, где отображено соотношение основных количественных характеристик базы данных, и кнопку скачивания, нажав на которую видит, как файл со статистикой загружается на его компьютер

Альтернативные сценарии:

- Ошибка соединения -> 404

Результат: Пользователь видит статистику и возможность вернуться обратно к базе данных. Файл со статистикой загружен на его компьютер

6. Изменение патентов

Отображение: 5 -> 7 -> 9

Действия пользователя: Пользователь выбирает опцию Изменить и вводит в поиске формулу для нахождения всех патентов с номером, начинающимся на 1234 и нажимает на лупу, сохраняет формулу и меняет нужные поля для искомых патентов и нажимает кнопки Внести изменения и Подтвердить изменения

Предусловия: Пользователь авторизован, на странице Опции над данными

Основной сценарий: Пользователь видит поле для поиска патентов и после ввода формулы и нажатия на лупу появляется таблица, значения которой можно редактировать. Изменив нужные данные, пользователь нажал на кнопку Внести изменения, после чего по каждому измененному патенту выводится информация о том, что изменилось. Пользователь нажимает на кнопку Подтвердить изменения и видит строку «Изменено!»

Альтернативные сценарии:

- Ошибка соединения -> 404
- Удаление информации без ввода новой в обязательном поле -> рядом с кнопкой Внести изменения появляется надпись «Необходимо заполнить все горящие ячейки», также все несправедливо опустевшие ячейки поменяют цвет
- Не изменено ни одно поле -> кнопка Внести изменения на начинает трястись при нажатии (на 3 секунды загорается красным)

Результат: Пользователь видит строку «Изменено!», может вернуться к базе данных, вернуться к редактированию данных о только что измененных патентах или перейти к добавлению или изменению других. Изменения отразились в базе данных

7. Отмена изменений

Отображение: 9 -> 7 -> 9

Действия пользователя: Пользователь нажимает на кнопку Вернуться назад и редактирует данные, затем нажимает на кнопки Внести изменения и “Подтвердить” изменения

Предусловия: Пользователь авторизован, на странице успешного изменения патента

Основной сценарий: Пользователь видит таблицу с которой только что работал, видит историю всех изменений и кнопку Внести изменения, редактирует нужные ему поля, нажимает на кнопку Внести изменения, видит список изменений и появившуюся кнопку Подтвердить изменения, нажимает на нее и видит ту же страницу, с которой начинал редактирование

Альтернативные сценарии:

- Ошибка соединения -> 404
- Удаление информации без ввода новой в обязательном поле -> рядом с кнопкой Внести изменения появляется надпись «Необходимо заполнить все горящие ячейки», также все несправедливо опустевшие ячейки поменяют цвет
- Не изменено ни одно поле -> кнопка Внести изменения на начинает трястись при нажатии (на 3 секунды загорается красным)

Результат: Пользователь видит строку «Изменено!», может вернуться к базе данных, вернуться к редактированию данных о только что измененных патентах или перейти к добавлению или изменению других. Изменения отразились в базе данных

Операции чтения будут преобладать в данном веб-приложении, т.к. пользователи будут намного чаще просматривать информацию о патентах, чем импортировать новые патенты.

3. МОДЕЛЬ ДАННЫХ

База данных содержит информацию о патентах и пользователях, модерирующих данные.

Ниже представлена общая схема базы данных. Все перечисленные ниже поля носят информационный характер, и их тип не будет меняться в зависимости от вида базы данных.

Patents:

- registration number: string - 10 байт (для последних патентов 2022 + 6-значный номер)
- registration date: date - 8 байт
- application number: string - 10 байт
- application date: date - 8 байт
- authors: string - 32 байт (в среднем)
- authors count: int - 4 байта
- right holders: string - 105 байт (в среднем)
- contact to third parties: string - 21 байт (в среднем)
- program name: string - 80 байт (в среднем)
- creation year: date - 8 байт
- registration publish date: date - 8 байт
- registration publish number: int - 4 байта
- actual: bool - 1 byte
- publication URL: string - 71 байт (в среднем)

Users:

- login: string
- access: string

3.1. Нереляционная модель данных

На рис. 2 представлена нереляционная модель данных, которая будет реализована в СУБД MongoDB.

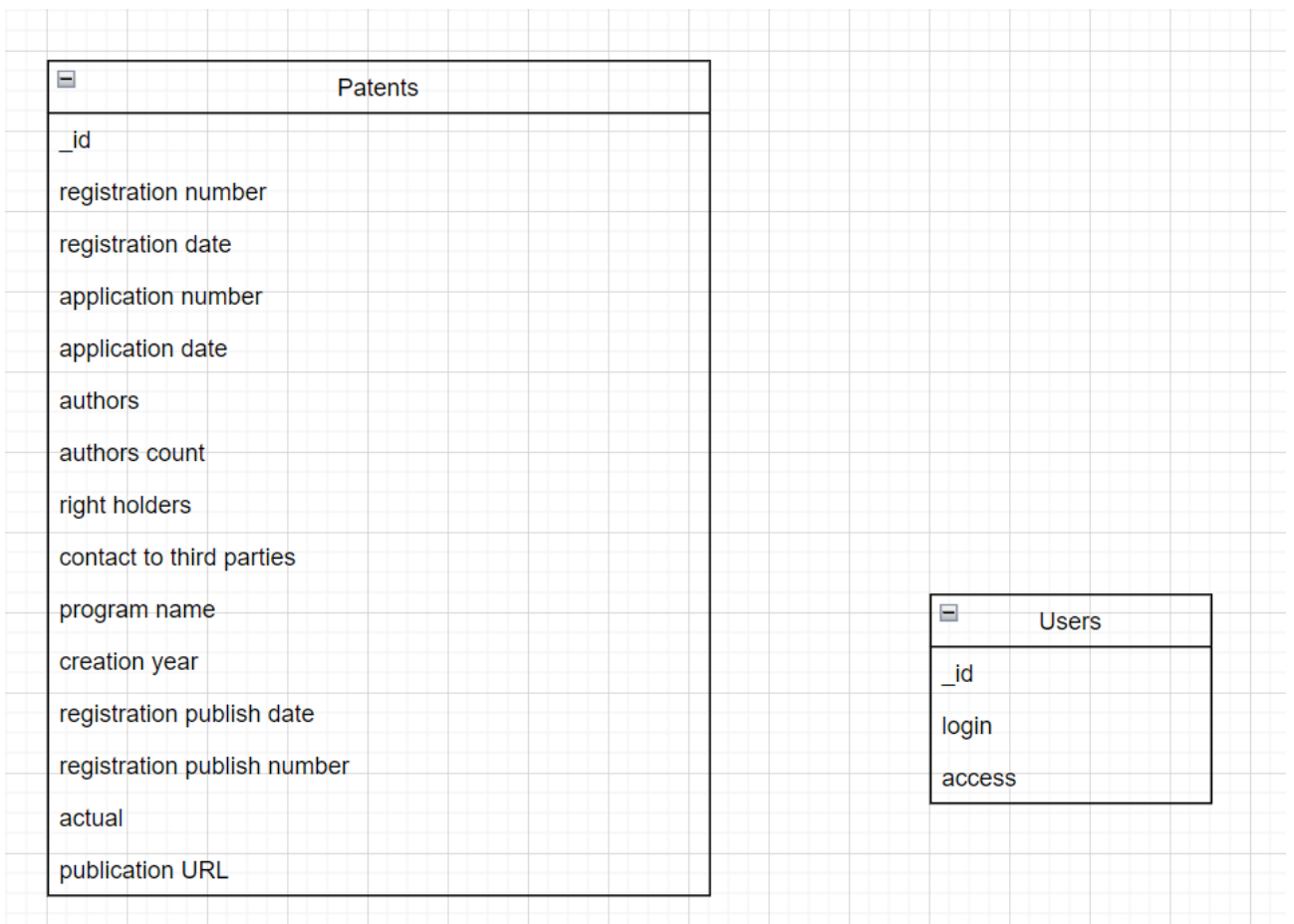


Рисунок 2. Графическое представление нереляционной модели данных

Оценка удельного объема информации, хранимой в модели

Пусть в базе данных N патентов и M пользователей.

Примем средний размер логина и пароля пользователя по 8 байт.

Размер базы данных: $N * (10 + 8 + 10 + 8 + 32 + 4 + 105 + 21 + 80 + 8 + 8 + 4 + 1 + 71) + M * (8 + 8) = 370 * N + 16 * M$.

Избыточность модели

Модель не содержит дополнительных данных, помимо исходных, а значит не является избыточной.

В базе содержатся повторяющиеся данные - имена авторов, но так как средний размер имени автора - 13 байт, а ObjectId - 12 байт, то замена авторов их идентификаторами в другой таблице не даст улучшения по памяти, но усложнит запросы. Поэтому все авторы патента хранятся в одной строке.

Направление роста модели при увеличении количества объектов каждой сущности

Размер базы данных растет линейно по каждому параметру.

Запросы к модели, с помощью которых реализуются сценарии использования

- **Добавление**

```
collection.insert_one({
    "_id" : "940436",
    "registration date" : datetime.datetime.strptime("19941006", "%Y%m%d"),
    "application number" : "940400",
    "application date" : datetime.datetime.strptime("19940926", "%Y%m%d"),
    "authors" : "",
    "authors count" : "1",
    "right holders" : 'Закрытое акционерное общество "СП ПараГраф"',
    "contact to third parties" : "",
    "program name" : "Гарнитуры шрифтовой библиотеки–ParaType (ПараТайп)",
    "creation year" : "",
    "registration publish date" : "19941220",
    "registration publish number" : 4,
    "actual" : True,
    "publication URL" :
    "http://[www1.fips.ru/fips_servl/fips_servlet?DB=EVM&DocNumber=940437](https://vk.com/away.php?to=http%3A%2F%2Fwww1.fips.ru%2Ffips_servl%2Ffips_servlet%3FDB%3DEVM%26DocNumber%3D940437&cc_key=)"
})
```

- **Удаление**

```
collection.delete_one({"_id" : "940436"})
```

- **Обновление**

```
collection.update_one({"_id" : "940436"}, {"$set" : {"application number" : "938212"}}
```

- **Поиск по полю**

```
collection.find_one({"_id" : "940436"})
```

- **Сортировка**

```
collection.find().sort("application number", pymongo.ASCENDING)
```

- **Поиск по нескольким полям**

Поиск осуществляется по полю 'program name' с неточным совпадением(вхождением) слова 'adobe', по полю 'actual' и по полю 'registration date', для которого задан диапазон [start, end] (концы включены).

```
start = datetime.datetime(1994, 12, 5)
end = datetime.datetime(1994, 12, 6)
pat = re.compile(r'.*adobe.*', re.I)
res = (collection.find({"program name" : {"$regex" : pat}, 'actual' : True,
    'registration date' : {'$lte' : end, '$gte' : start}}))
```

3.2. Аналог модели данных для SQL СУБД

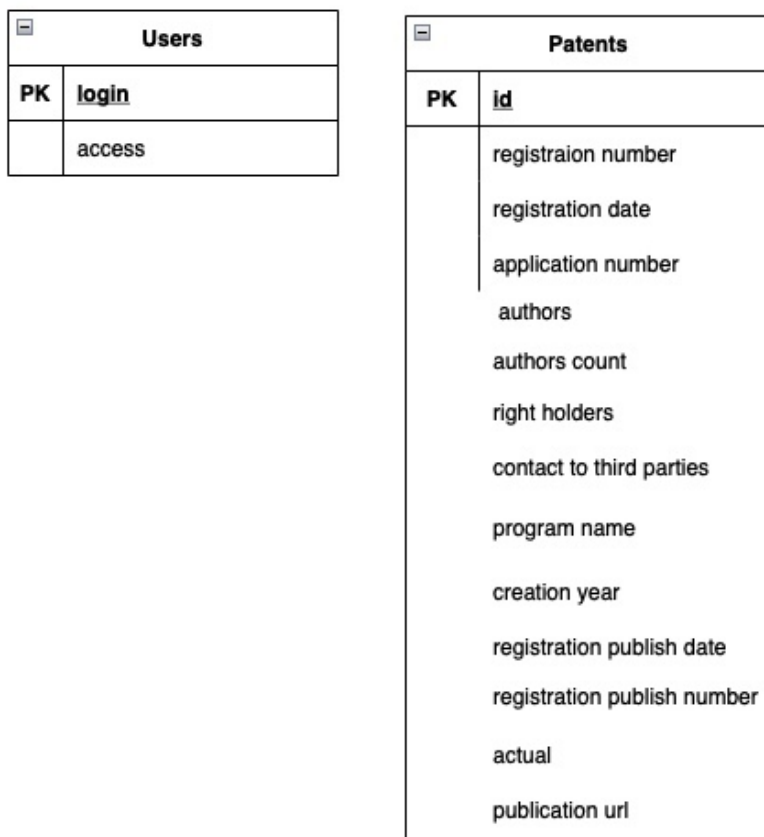


Рисунок 3. Графическое представление реляционной базы данных

Оценка удельного объема информации, хранимой в модели

Аналогично нереляционной модели.

Избыточность модели

Аналогично нереляционной модели.

Направление роста модели при увеличении количества объектов каждой сущности

Аналогично нереляционной модели.

Запросы к модели, с помощью которых реализуются сценарии использования

- **Добавление**

```
INSERT INTO Patents  
VALUES  
(...);
```

- **Удаление**

```
DELETE FROM Patents
WHERE registrastion_number == 0123456789;
```

- **Изменение поля**

```
UPDATE Patents
SET program_name = 'odnoklassniki'
WHERE registrastion_number == 0123456789;
```

- **Поиск по полю**

```
SELECT *
FROM Patents
WHERE program_name = 'odnoklassniki';
```

- **Сортировка**

```
SELECT *
FROM Patents
ORDERED BY application_number ASC;
```

- **Поиск по нескольким полям**

```
SELECT *
FROM Patents
WHERE program_name LIKE '%adobe%' AND actual = 'True' AND registration_date
BETWEEN '1994-12-05' AND '1994-12-06'
```

Сравнение моделей

В данной модели не обнаружено преимуществ нереляционной базы данных, оба варианта эквивалентны по объему и покрывают сценарии use case за один запрос.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

Веб-приложение в качестве СУБД использует MongoDB [1]. Бэкенд написан на языке Python с применением фреймворка Flask [2]. Для взаимодействия с базой данных на уровне бэкенда использовалась библиотека mongoengine [3]. Фронтенд написан на html с js скриптами. Для отображения данных использовалась библиотека таблиц DataTables [4].

В программе реализованы все основные сценарии: вход пользователя; отображение, импорт/экспорт, фильтрация данных; поиск по данным.

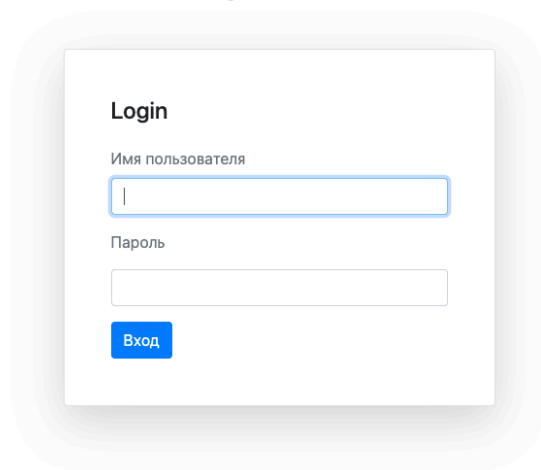
На рис. 4-6 изображены основные страницы приложения.

The screenshot shows the 'Rospatent DB' application interface. At the top, there is a navigation bar with 'База данных' and 'Операции над данными'. Below this, there are buttons for 'Добавить патент', 'Выгрузить базу данных :3', and a file upload section with 'Выберите файл', 'Файл не выбран', and 'Загрузить csv' with a file icon. The main content area features a table with columns for registration number, registration date, application number, application date, authors, and authors count. Above the table, there are search and filter controls: 'Show 10 entries', a 'Search:' input field, and filters for 'registration date', 'application date', and 'authors' with date pickers and search inputs. The table contains six rows of data.

registration number	registration date	application number	application date	authors	authors count
940274	1994-07-11	940178	1999-12-20	Косарева Ю.А. (RU) Усов А.Ю. (RU) Бондаренко В.С. (RU)	3
940436	1994-10-06	940400	1994-09-26	nan	1
940437	1994-10-06	940401	1994-09-26	nan	1
950092	1995-03-15	950090	1995-01-30	Зиновьев П.А. (RU) Жиляев А.И. (RU)	2
950108	1995-03-20	0000950019	1995-01-19	nan	1
950396	1995-11-09	0000950377	1995-10-19	Тюхов Борис Петрович (RU)	3

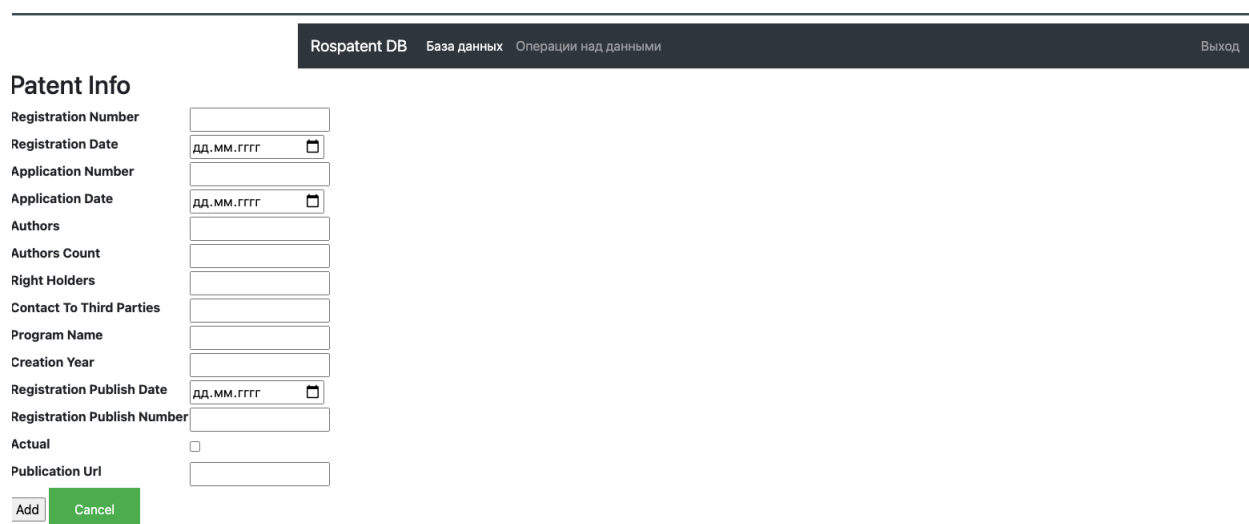
Рисунок 4. Отображение данных со всеми основными возможностями.

Rospatent DB



The login form is titled "Login" and is centered on the page. It contains two input fields: "Имя пользователя" (Username) and "Пароль" (Password). Below the password field is a blue button labeled "Вход" (Login).

Рисунок 5. Страница входа пользователя.



The "Patent Info" form is located on the left side of the page. It contains the following fields and controls:

- Registration Number: Text input
- Registration Date: Date input (ДД.ММ.ГГГГ) with a calendar icon
- Application Number: Text input
- Application Date: Date input (ДД.ММ.ГГГГ) with a calendar icon
- Authors: Text input
- Authors Count: Text input
- Right Holders: Text input
- Contact To Third Parties: Text input
- Program Name: Text input
- Creation Year: Text input
- Registration Publish Date: Date input (ДД.ММ.ГГГГ) with a calendar icon
- Registration Publish Number: Text input
- Actual: Checkbox
- Publication Uri: Text input

At the bottom of the form are two buttons: "Add" (green) and "Cancel" (white).

Рисунок 6. Страница добавления патента.

Использованные технологии

БД: MongoDB

Бэкенд: Python, Flask, mongoengine

Фронтенд: HTML, JavaScript, JQuery, DataTables, Bootstrap 5.0, CSS

Ссылка на приложение

<https://github.com/moevm/nosql2h22-rospatent>

5. ВЫВОДЫ

Результаты

Разработано веб-приложение каталога базы данных Роспатента, были реализованы основные сценарии по импорту/экспорту, фильтрации данных.

Недостатки и пути для улучшения полученного решения

Расположение элементов интерфейса не совсем интуитивное, следует предоставить пользователю больше удобств. Также следует реализовать дополнительные сценарии, которые добавят пользователю возможностей.

Будущее развитие решения

Помимо каталога реестра программ для ЭВМ, реализовать каталог для остальных открытых данных роспатента <https://rospatent.gov.ru/opendata>.

6. ПРИЛОЖЕНИЯ

Документация по сборке приложения указана в файле README.md по ссылке <https://github.com/moevm/nosql2h22-rospatent#readme>.

7. ЛИТЕРАТУРА

- [1] СУБД MongoDB <https://www.mongodb.com/docs/manual/>
- [2] Flask <https://flask.palletsprojects.com/en/2.2.x/>
- [3] MongoEngine <https://docs.mongoengine.org/guide/index.html>
- [4] DataTables <https://datatables.net/manual/>