

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

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: База данных статистики github репозитория

Студент гр. 9381

Гурин С.Н.

Студент гр. 9381

Кравченко П.В.

Студент гр. 9381

Щеглов Д.А.

Преподаватель

Заславский М.М.

Санкт-Петербург

2022

ЗАДАНИЕ НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студент Гурин С.Н.

Студент Кравченко П.В.

Студент Щеглов Д.А.

Группа 9381

Тема работы: База данных статистики github репозиторийев

Содержание пояснительной записки:

- Аннотация
- Содержание
- Введение
- Качественные требования к решению
- Сценарии использования
- Модель данных
- Разработанное приложение
- Заключение
- Список использованных источников

Предполагаемый объем пояснительной записки:

Не менее 20 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 14.03.2023

Дата защиты реферата: 14.03.2023

Студент

Гурин С.Н.

Студент

Кравченко П.В.

Студент

Щеглов Д.А.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В данной работе был построен макет UI для сайта фитнес-клуба и составлены сценарии использования приложения, также была построена модель данных и сформированы базы данных. В результате был разработан веб-сайт для просмотра статистики github репозиторий, выбранные пользователем.

SUMMARY

In this work, a user interface layout was created for a fitness club website and application use cases were compiled, a data model was created and data was generated. As a result, a website was developed to view the statistics of the github repositories selected by users.

СОДЕРЖАНИЕ

	Введение	5
1.	Качественные требования к решению	6
2.	Сценарии использования	7
3.	Модель данных	9
4.	Разработанное приложение	19
	Вывод	22

ВВЕДЕНИЕ

Цель данной работы является создание веб-приложения для просмотра статистики github репозиториев.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

1.1. Текущие требования

Текущие требования к решению:

1. Веб-сайт с возможностью авторизации и регистрации.
2. Хранение баз данных статистики репозиторий в СУБД MongoDB.
3. Массовый импорт/экспорт данных в/из СУБД в формате json.
4. Сборка и развертывание приложения через docker-compose.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЯ

Сценарий “Аутентификации”

Основной сценарий

- Пользователь вводит логин и пароль
- Нажимает кнопку войти
- Вход пользователя в систему

Альтернативный сценарий

- Пользователь вводит логин и пароль
- Нажимает кнопку войти
- Данные введены не верно
- Повтор сценария

Сценарий “Регистрация”

Основной сценарий

- Переход в сценарий происходит в сценрии “Аутентификации”
- Пользователь вводит логин, токен и пароль
- Нажимает на кнопку “Зарегистрироваться”
- Переход в сценрий “Аутентификации”

Альтернативный сценарий

- Переход в сценарий происходит в сценрии “Аутентификации”
- Пользователь вводит логин, токен и пароль
- Нажимает на кнопку “Зарегистрироваться”
- Данные введены неверно

Сценарий “Выход”

- Пользователь нажимает на соответствующую кнопку
- Пользователь переходит на страницу аутентификации

Сценарий “Добавить репозиторий”

Основной сценарий

- Пользователь нажимает на кнопку “Добавить репозиторий”
- Пользователь вводит название репозитория
- Репозиторий добавлен в список репозиториев

Альтернативный сценарий

- Пользователь нажимает на кнопку “Добавить репозиторий”
- Пользователь вводит название репозитория
- Данные введены не верно
- Повтор сценария

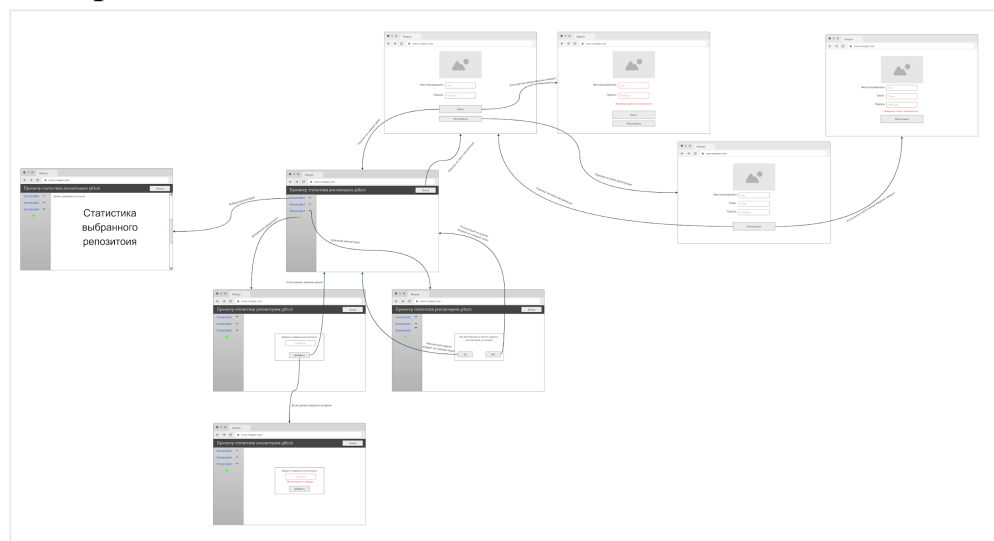
Сценарий “Удаление репозитория”

- Пользователь нажимает на кнопку “Удалить репозиторий” рядом с выбранным репозиторием
- Пользователю выводится диалоговое окно “Удалить репозиторий или нет”
- Репозиторий удален из списка репозитория (Если пользователь согласился)
- Репозиторий не удален из списка репозитория (Если пользователь отказался)

Сценарий “Просмотр статистики”

- Пользователь нажимает на уже добавленный репозиторий
- Пользователю предоставляется статистика выбранного репозитория

Use Case диаграмма



3. МОДЕЛЬ ДАННЫХ

Схема БД и список сущностей

База данных содержит информацию о репозиториях, коммитах, пуллреквестах и Issue. Ниже приведен список полей. ### User - Username - Password - Token - Repos

Репо

- Name
- Issues
- Commits
- Pull_Requests

Commit

- Id
- Author_Name
- Author_login
- Author_email
- Data_and_time
- Changed_files
- Message

Pull_Request

- Id
- Title
- State
- Commit_into
- Creator_name
- Creation_date
- Creator_login
- Creator_email
- Changed_files
- Comments
- Merger_login
- Merger_email

Issue

- Id
- Number
- Title
- State
- Task

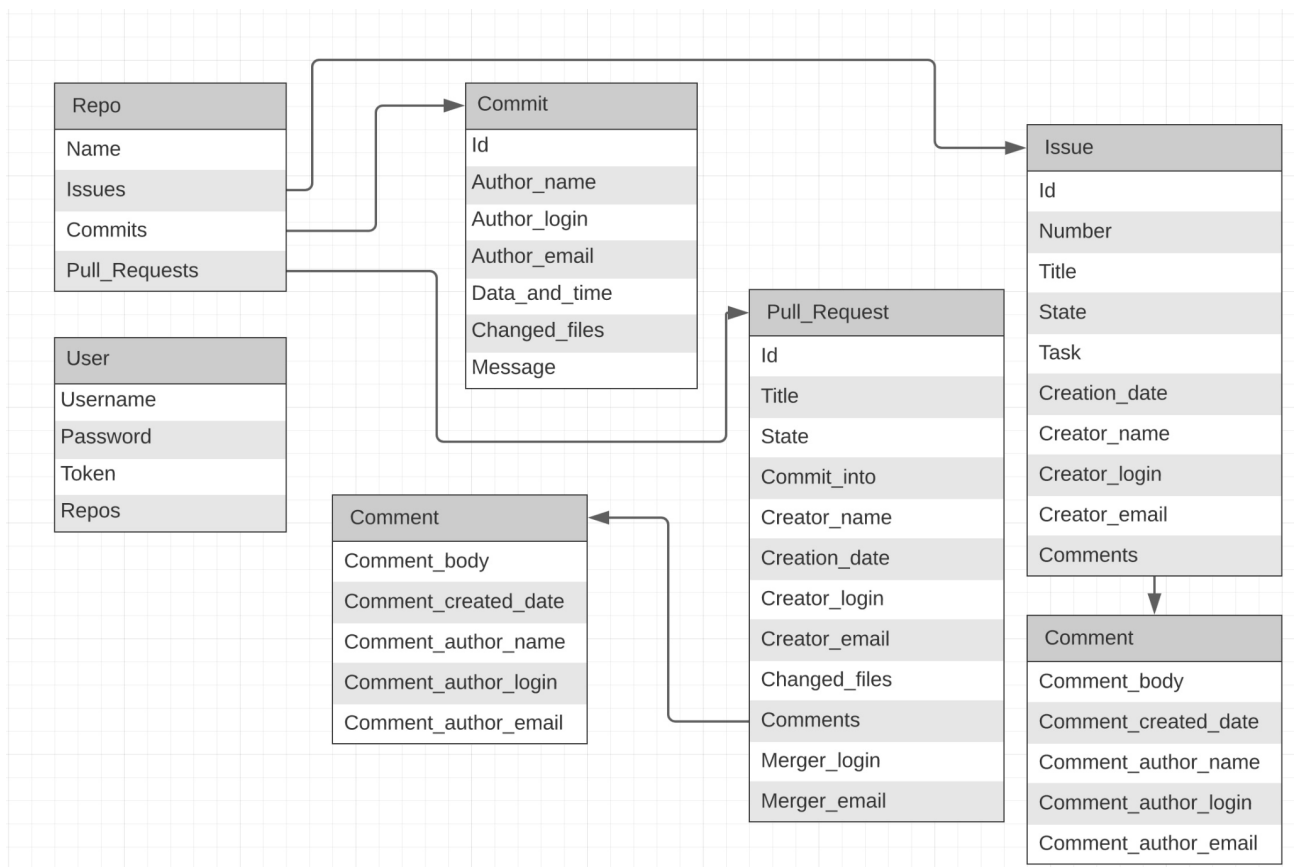
- Creation_date
- Creator_name
- Creator_login
- Creator_email
- Changed_files
- Comments

Comment

- Comment_body
- Comment_created_date
- Comment_author_name
- Comment_author_login
- Comment_author_email

Нереляционная модель данных

Графическое представление



Оценка удельного объема информации, хранимой в модели Пусть в базе данных N1 пользователей N2 репозиторий N3 пулреквестов N4 коммитов и N5 Issue .
 Максимальные размеры полей документов: ### User - Username: 128 bytes - Password: 128 bytes - Token: 93 bytes - Repos: 760 bytes

Чистый объём документа user: 1109 bytes

Фактической объём документа user: 1109 bytes

Repo

- Name: 256 bytes
- Issues: 760 bytes
- Commits: 760 bytes
- Pull_Requests: 760 bytes

Чистый объём документа Repo: 2536 bytes

Фактической объём документа Repo: 2536 bytes

Commit

- Id: 8 bytes
- Author_Name :128 bytes
- Author_login: 128 bytes
- Author_email: 128 bytes
- Data_and_time: 20 bytes
- Changed_files: 760 bytes
- Message: 408 bytes

Чистый объём документа commit: 1572 bytes

Фактической объём документа commit: 1580 bytes

Pull_Request

- Id: 8 bytes
- Title: 256 bytes
- State: 100 bytes
- Commit_info: 100 bytes
- Creator_name: 128 bytes
- Creation_date: 20 bytes
- Creator_login: 128 bytes
- Creator_email: 128 bytes
- Changed_files: 760 bytes
- Comments: 1404 bytes
- Merger_login: 128 bytes
- Merger_email: 128 bytes

Чистый объём документа Pull_Request: 4684 bytes

Фактической объём документа Pull_Request: 4692 bytes

Issue

- Id: 8 bytes
- Number: 8 bytes
- Title: 256 bytes
- State: 100 bytes
- Task: 1000 bytes
- Creation_date: 20 bytes
- Creator_name: 128 bytes
- Creator_login: 128 bytes
- Creator_email: 128 bytes
- Changed_files: 760 bytes
- Comments: 1404 bytes

Чистый объём документа Issue: 3932 bytes

Фактической объём документа Issue: 3940 bytes

Comment

- Comment_body: 1000 bytes
- Comment_created_date: 20 bytes
- Comment_author_name: 128 bytes
- Comment_author_login: 128 bytes
- Comment_author_email: 128 bytes

Чистый объём документа Comment: 1404 bytes

Фактической объём документа Comment: 1404 bytes

Избыточность модели

Пусть в базе данных $N_1=5$ пользователей, $N_2=10$ репозиторий, $N_3=100$ пулреквестов, $N_4=120$ коммитов, $N_5=10$ Issue и $N_6=200$ комментариев

$V_{clear} = N_1 * 1109 + N_2 * 2536 + N_3 * 4684 + N_4 * 1164 + N_5 * 3932 + N_6 * 1404 =$
 $5 * 1109 + 10 * 2536 + 100 * 4684 + 120 * 1164 + 10 * 3932 + 200 * 1404 = 959105$ bytes

$V_{real} = N_1 * 1109 + N_2 * 2536 + N_3 * 4692 + N_4 * 1172 + N_5 * 3940 + N_6 * 1404 = 5 * 1109$
 $+ 10 * 2536 + 100 * 4692 + 120 * 1172 + 10 * 3940 + 200 * 1404 = 960945$ bytes

Избыточность: $V_{real} / V_{clear} = 960945 / 959105 = 1,00191846$

Направление роста модели при увеличении количества объектов каждой сущности

Размер базы данных растёт линейно по каждому параметру.

Запросы к модели, с помощью которых реализуются сценарии использования

Добавление

```
collection.new_commit({ "Id":1, "Author_Name":"Pavel",  
"Author_login":"cirilohysP", "Author_email":"realmathsmc@gmail.com",  
"Data_and_time":"2022-12-12 19:58:48", "Changed_files":",Новая папка  
(2)/README.md" })
```

Удаление

```
collection.delete_commit('id' : '1')
```

Обновление

```
collection.update_commit({ "_id" : "1"}, { "$set" : { "Author_Name" : "Pavel  
Kravchenko" } })
```

Поиск по полю

```
collection.find_commit({ "Author_login" : "Dunkel" })
```

Сортировка

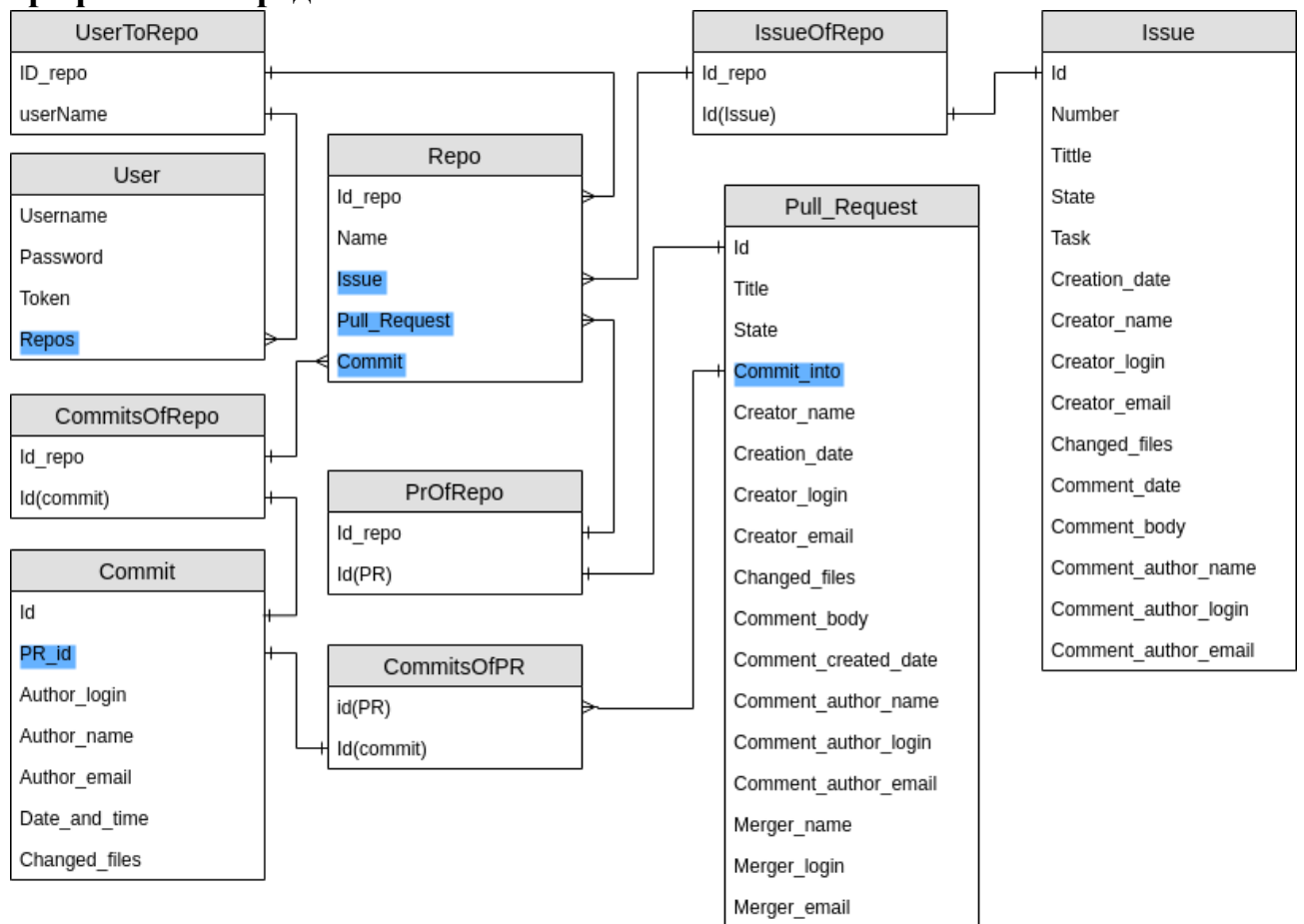
```
collection.find().sort("Author_Name")
```

Поиск по нескольким полям

```
collection.find_commit({ "Author_Name":"Pavel", "Changed_files":",Новая папка  
(2)/README.md" })
```

Аналог модели данных для SQL СУБД

Графическое представление



ModelData

User

- UserName: 128 bytes
- Password: 128 bytes
- Token: 93 bytes

Чистый объём документа user: 349 bytes

Фактической объём документа user: 349 bytes

Repo

- Name: 256 bytes
- Id: 8 bytes Чистый объём документа Repo: 256 bytes

Фактической объём документа Repo: 264 bytes

Commit

- Id: 8 bytes
- Author_Name: 128 bytes
- Author_login: 128 bytes

- Author_email: 128 bytes
- Data_and_time: 20 bytes
- Changed_files: 760bytes

Чистый объём документа commit: 1164 bytes

Фактической объём документа commit: 1172 bytes

Pull_Request

- Id:8 bytes
- Title:256 bytes
- State: 100 bytes
- Creator_name: 128 bytes
- Creation_date: 20 bytes
- Creator_login: 128 bytes
- Creator_email: 128 bytes
- Changed_files: 760bytes
- Comment_body: 1000 bytes
- Comment_created_date: 20 bytes
- Comment_author_name: 128 bytes
- Comment_author_login: 128 bytes
- Comment_author_email: 128 bytes
- Merger_name: 128 bytes
- Merger_login: 128 bytes
- Merger_email: 128 bytes

Чистый объём документа Pull_Request: 3308 bytes

Фактической объём документа Pull_Request: 3316 bytes

Issue

- Id:8 bytes
- Number:8 bytes
- Title:256 bytes
- State:100 bytes
- Task: 1000 bytes
- Creation_date:20 bytes
- Creator_name:128 bytes
- Creator_login:128 bytes
- Creator_email:128 bytes
- Changed_files:760 bytes
- Comment_date:20 bytes
- Comment_body:1000 bytes

- Comment_author_name:128 bytes
- Comment_author_login:128 bytes
- Comment_author_email:128 bytes

Чистый объём документа Issue: 3932 bytes

Фактической объём документа Issue: 3940 bytes

IssueOfRepo

- Id_repo: 8 bytes
- Id(Issue):8 bytes

Чистый объём документа IssueOfRepo: 0 bytes

Фактической объём документа IssueOfRepo: 16 bytes

UserToRepo

- ID_repo:8 bytes
- userName:8 bytes

Чистый объём документа UserToRepo: 0 bytes

Фактической объём документа UserToRepo: 16 bytes

CommitsOfRepo

- Id_repo:8 bytes
- Id(commit):8 bytes

Чистый объём документа CommitsOfRepo: 0 bytes

Фактической объём документа CommitsOfRepo: 16 bytes

PrOfRepo

- Id_repo:8 bytes
- Id(PR):8 bytes

Чистый объём документа PrOfRepo: 0 bytes

Фактической объём документа PrOfRepo: 16 bytes

CommitsOfPR

- id(PR):8 bytes
- Id(commit):8 bytes

Чистый объём документа PrOfRepo: 0 bytes

Фактической объём документа PrOfRepo: 16 bytes

Оценка удельного объема информации, хранимой в модели

Пусть у нас есть $N1=5$ пользователей каждый работает в $N2=10$ репозиториях (все во всех) Имеется по $N4=120$ коммитов, $N3=100$ PR и $N5=10$ Issue

$$V_{clear} = N1*349 + N2*256 + N3*3308 + N4*1164 + N5*3932 = 5*349 + 10*256 + 100*3308 + 120*1164 + 10*3932 = 514105 \text{ bytes}$$

$$\begin{aligned} V_{real} = & N1*349 + N2*264 + N3*3316 + \\ & N4*1172 + N5*3940 + N1*N2*16 + N3*N2*16 + N4*N2*16 + N5*N2*16 + N4*16 = \\ & 5*349 + 10*264 + 100*3316 + 120*1172 + 10*3940 \\ & + 5*10*16 + 100*10*16 + 120*10*16 + 10*10*16 + 120*16 = 555545 \text{ bytes} \end{aligned}$$

Тогда по сравнению с нереализованной мы задействуем на $N2*(N1+N3+N4+N5)*(8+8) + N4*(8+8) = 39520$ bytes на связующие таблицы, но сэкономим на Полях перечисления $N1*(1109-349) + N2*(2544-264) + N3*(3416-3316) + N4*(1172-1172) + N5*(3940-3940) = 5*760 + 10*2280 + 100*100 + 120*0 + 10*0 = 36600$ bytes.

Разница фактического веса моделей при заданных данных равна: $555545 - 552625 = 2920$ bytes

Избыточность модели

Избыточность: $V_{real} / V_{clear} = 555545 / 514105 = 1,08060610187$

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

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

Удаление

```
DELETE FROM Commit WHERE id == 1;
```

Обновление

```
UPDATE commit SET Author_Name = Pavel Kravchenko WHERE id == 1;
```

Поиск по полю

```
SELECT * FROM commit WHERE Author_login = Dunkel;
```

Сортировка

```
SELECT * FROM commit ORDERED BY Author_Name ASC;
```

Поиск по нескольким полям

```
SELECT * FROM commit WHERE Author_Name = Pavel AND  
Changed_files="Новая папка (2)/README.md"
```

Пример представления данных в базе данных

```
_id: ObjectId('640fb41dab235c9acbf71376')  
Name: "test"  
▸ Issues: Array  
▸ Commits: Array  
▸ Pull_Requests: Array
```

```
▸ _id: ObjectId('640fb3fdab235c9acbf71375')  
  UserName: "SimonGyrin"  
  Password: "123"  
  Token: "████████████████████████████████████████████████████████████████████████████████"  
  ▸ Repos: Array
```

Сравнение моделей

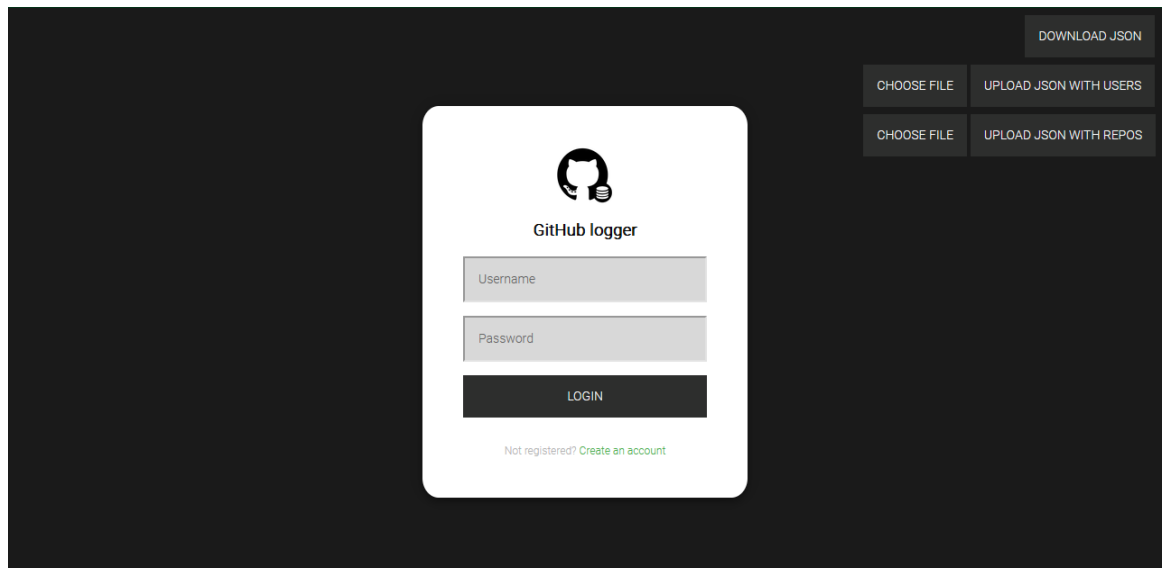
В данной модели обнаружено преимущество нереляционной модели все связи с отсутствием накладных данных на таблицы связи, которые покрываются списками в одной сущности.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

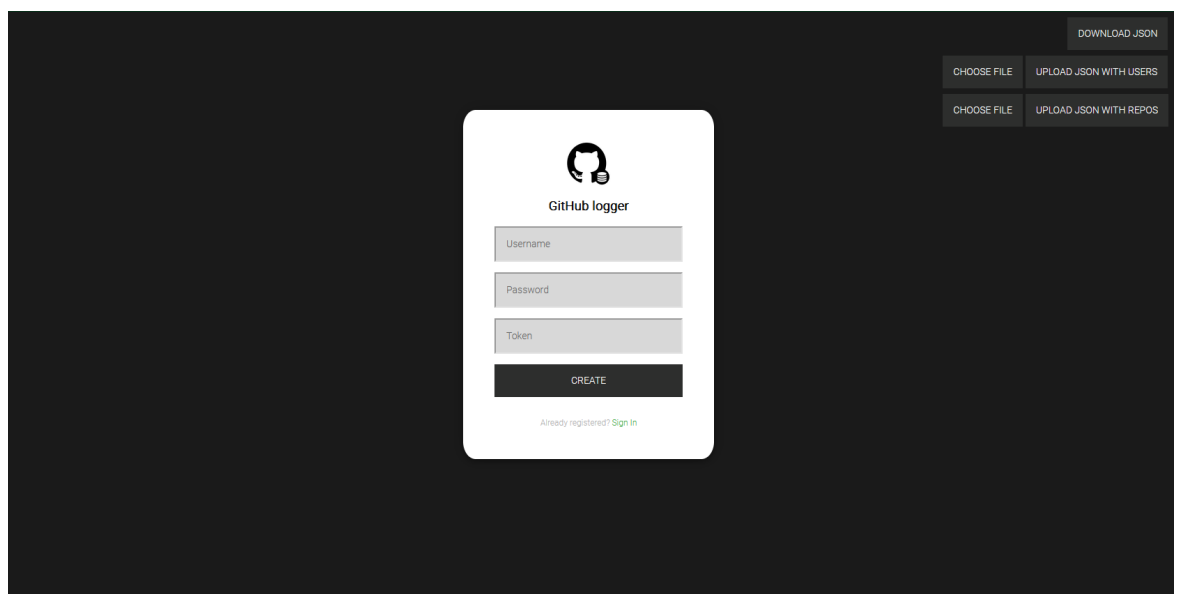
Краткое описание

Приложение реализовано с использованием Python библиотеки Flask, для работы с MongoDB была использована библиотека pymongo.

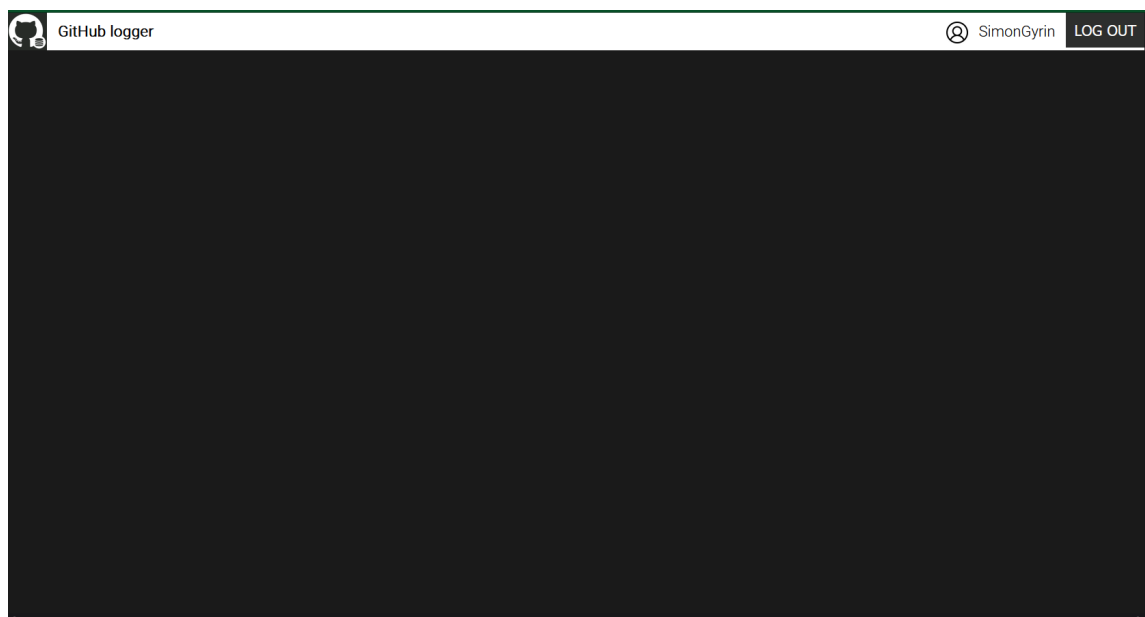
Схема экранов приложения



Экран входа



Экран регистрации



Главный экран



Экран выбора репозитория

GitHub logger

SimonGyrin LOG OUT

Parameter

Name	Id	Author name	Author login	Author email	Data and Time
Add files via upload	1d419f434866f054f1660e87de051c159d7f41ae		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-22T01:24:36
Add files via upload	05cc9072f8a93ea25056d9e034caf53dd8a1d32d		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-22T01:17:12
Delete 3 lab directory	0396d385343f7b5ca3173943bdc7cd7af0a1ba4		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-22T01:17:01
Add files via upload	c19124599d9f1a7e5d53321dc53f6b99dc6659905		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-22T01:16:52
Delete 1 lab directory	6041b0e26007c79c3640f009635ae67f8f3c46		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-22T01:16:28
Add files via upload	e1a257588b81b9f70b8d47c25ce10489caa262c4		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-21T22:52:28
Create README.md	fa0270e4261257db566ccdf514c82ddb4a43ba4bb3		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-21T20:40:41
lab_1	434abc9fcc002c5a418bcd628db0d99bc2bd76f1		SimonGyrin	55016616+SimonGyrin@users.noreply.github.com	2021-12-21T20:36:30

Changed files of Add files via upload
3 lab/Untitled.ipynb

Changed files of Add files via upload
2 lab/Untitled.ipynb
2 lab/record1.wav

Changed files of Delete 3 lab directory
3 lab/Untitled.ipynb

Changed files of Add files via upload
1 lab/03-01-05-01-01-01-03_A3.wav
1 lab/Untitled.ipynb
1 lab/detected.wav
1 lab/distortion.wav
1 lab/echo.wav
1 lab/lowpass.wav

Экран статистики коммитов в выбранном репозитории

GitHub logger

SimonGyrin LOG OUT

Find repo

TEST

MATHPACKAGES_LABS

+

CLOSE

FIND

INPUT REPOSITORY NAME YOU WANT TO ADD

Name of your new repo

ADD

Экран добавления репозитория

Использованные технологии

СУБД: MongoDB

Backend: pymongo, Flask

Frontend: HTML, js, jquery

Ссылка на приложение

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

ВЫВОД

В ходе выполнения задачи было реализовано web-приложение просмотра статистики git репозитория. Были изучены методы обращения с нереляционными СУБД в частности с MongoDB.