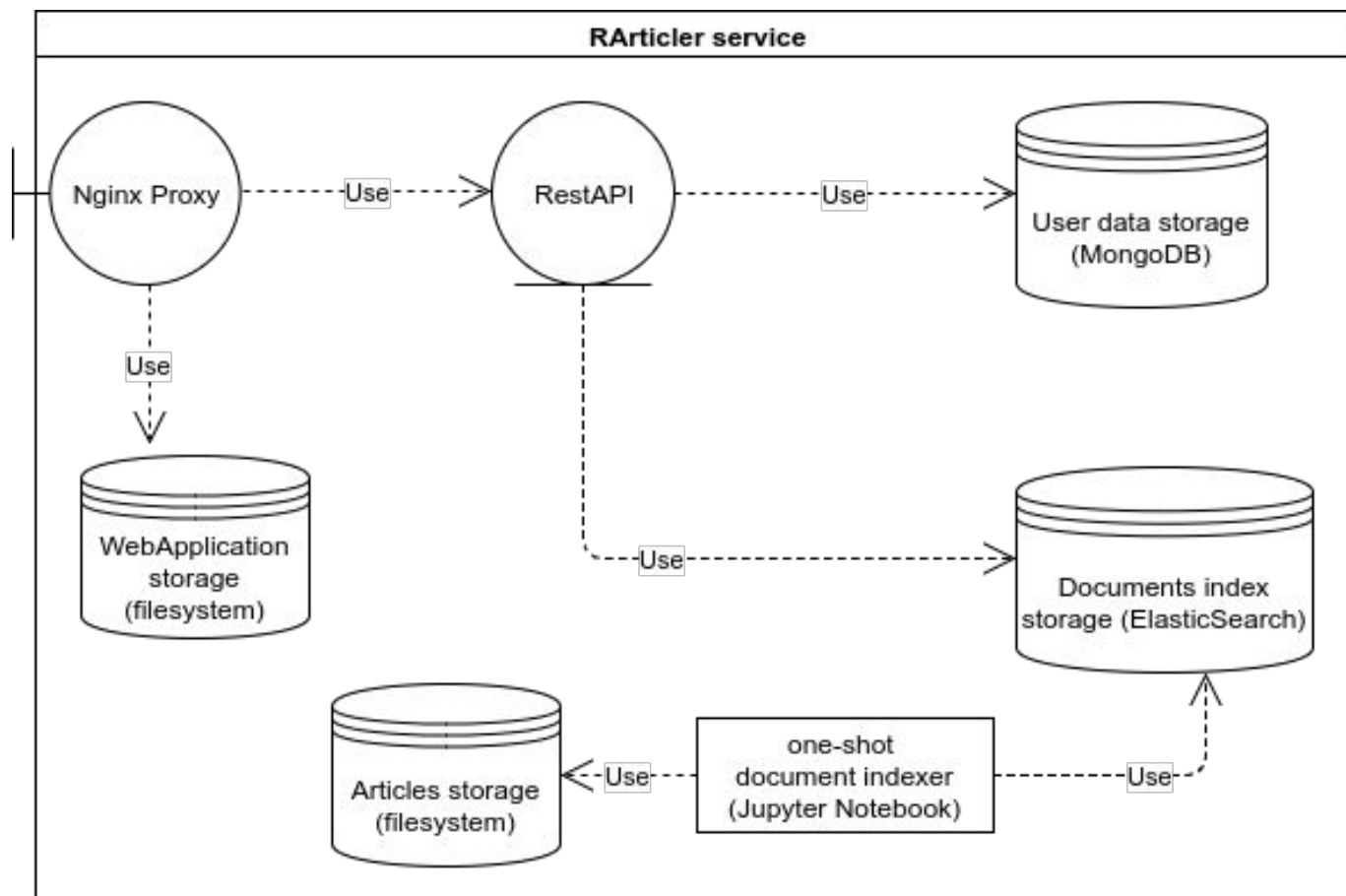


Поисковая система статей

- Основной функциональностью сервиса является поиск статей релевантных запросу
- Сервис должен предоставлять пользователю функцию создания аккаунта
- Сервис должен предоставлять зарегистрированному пользователю функцию личной библиотеки

Архитектура



Технологии

- ❑ Фронтенд (графический интерфейс)
 - ❑ Vue.js - фреймворк для создания одностраничных приложений на JavaScript
 - ❑ Vuetify.js - material design для vue.js
 - ❑ Axios.js - библиотека, упрощающая выполнение асинхронных http запросов.
 - ❑ Vuex - дополнение к vue.js, позволяющее хранить глобальные значения не нарушая принципов ООП
- ❑ Бэкенд (конструирование запроса к бд, выполнение запроса, отдача результата)
 - ❑ Python 3
 - ❑ Sanic - простой веб сервер на Python 3 на основе asyncio
 - ❑ Uvloop - альтернативный цикл событий для asyncio
 - ❑ Sanic-jwt - аутентификация и авторизация с использованием токенов
 - ❑ Elasticsearch - система индексирования, хранения и поиска документов
 - ❑ MongoDB - документо-ориентированная СУБД
- ❑ Шлюз (взаимодействие приложения с пользователем)
 - ❑ Nginx - высокопроизводительный веб-сервер, балансировщик нагрузки и прокси.

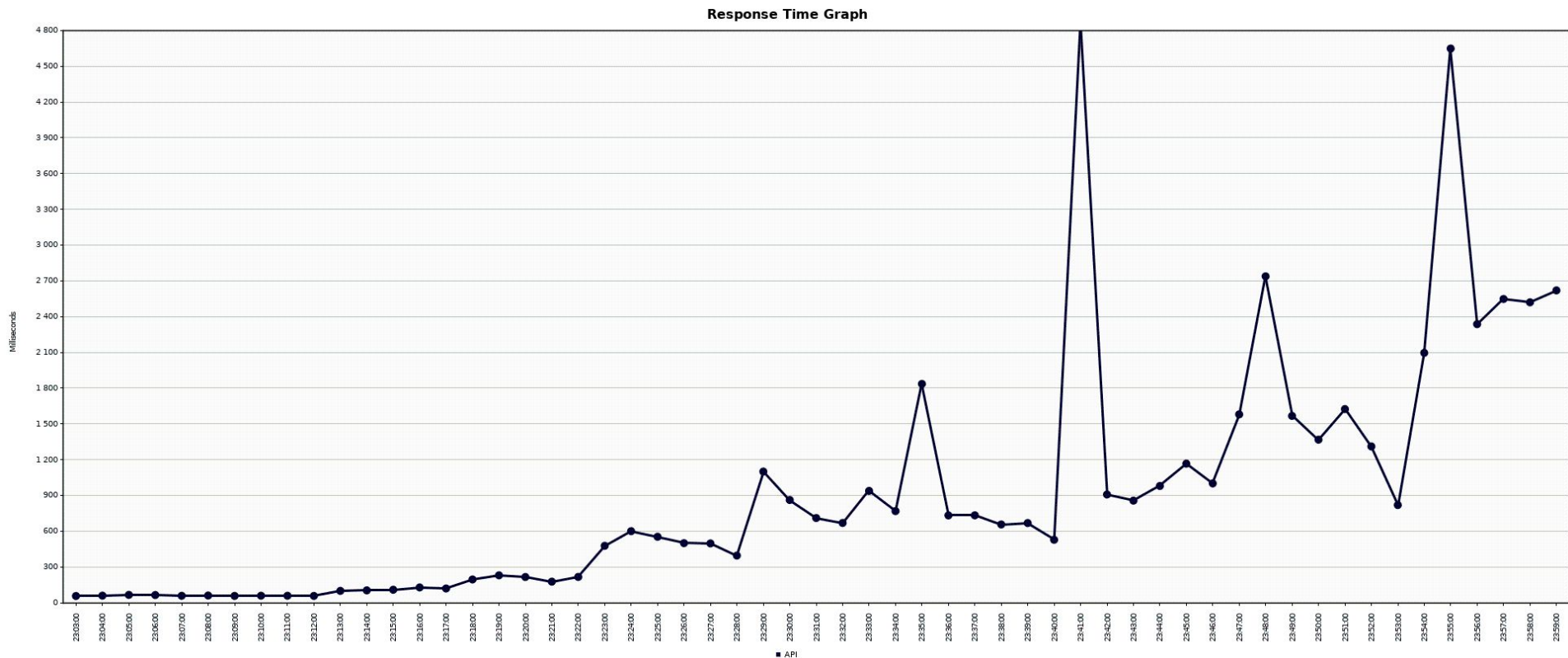
Обзор реализованной функциональности

Фаза	Возможности
Минимальный продукт	Поиск документов релевантных запросам Просмотр результатов поиска
Персонализация	Регистрация в системе Аутентификация в системе
Библиотека	Добавление статьи из поиска в библиотеку Просмотр содержимого библиотеки

Производительность сервиса поиска

Users	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput
1	61	48	57	87	463	35	1171	0,000%	16
10	57	43	65	86	381	33	2182	0,000%	172
50	114	43	159	324	1649	33	11460	0,000%	434
100	212	49	328	897	3289	33	130255	0,001%	457
250	529	77	844	1912	7327	33	130221	0,082%	398
350	812	79	1082	2835	11390	33	157922	0,249%	352
400	716	76	914	2042	9381	33	131379	0,240%	457
500	1000	78	1085	3027	15921	33	168967	0,434%	398
700	1551	81	1394	4013	60091	33	162614	0,946%	326
1000	2646	91	2909	7741	64561	33	157608	2,017%	279

Производительность сервиса поиска



Сложные моменты

- Реализация пагинации
- Комбинация basic auth с jwt
- Проведение нагрузочного тестирования
- Дизайн
- Изучение документации технологий требует времени

Команда

Сивец Денис

- ❖ Фронтенд
- ❖ Бэкенд
- ❖ Нагрузочное тестирование
- ❖ Техническая документация

Кулеш Антон

- ❖ Майнинг статей
- ❖ Документация

Presentation checklist

1. Запустить все изменения по проекту в гит **до 22.12.2017**. Ссылку на репозиторий скинуть в слак.
2. Подготовить презентацию проекта на 10-15 минут. Слайды должны быть следующими:
 - a. **Слайд 1:** Тема проекта, список функциональных требований
 - b. **Слайд 2:** Диаграмма архитектуры (плюс другие диаграммы, если есть)
 - c. **Слайд 3:** Список выбранных технологий, краткое устное обоснование их выбора
 - d. **Слайд 4:** Обзор реализованной функциональности
 - e. **Слайд 5:** Таблица результатов производительности приложения для основного сервиса приложения (результаты на различных объемах данных и/или количестве запросов)
 - f. **Слайд 6:** Сложные моменты в построении архитектуры приложения и в реализации, с которыми пришлось столкнуться
 - g. **Слайды 7,8,9:** Личный вклад каждого участника команды в проект
3. Слайды 1-6 рассказывает один человек из команды, слайды 7-9 - каждый говорит о своей работе.
4. Продемонстрировать работу приложения на своем ноутбуке.