

Dokumentacja projektu – File Manager

Wstęp.....	2
Opis aplikacji File Manager.....	2
Ważne linki.....	2
Odbiorcy docelowi.....	2
Aktorzy.....	3
Wymagania funkcjonalne(use cases).....	3
Wymagania funkcjonalne(user stories).....	4
Wymagania pozafunkcjonalne.....	5
Stos technologiczny.....	5
Architektura.....	6
Schemat relacyjnej bazy danych.....	7

Wstęp

Ponizszy dokument został przygotowany w ramach projektu zaliczeniowego z przedmiotu Java na kierunku Informatyka w Wyższej Szkole Bankowej w Poznaniu. Opisuje on aspekty techniczne, funkcjonalne oraz pozafunkcjonalne aplikacji File Manager.

Opis aplikacji File Manager

Stworzona aplikacja jest serwerem plików, który pozwala użytkownikowi zarządzać własnymi plikami, przechowywać je oraz ściągać. Użytkownik ma także możliwość uzyskania podstawowych informacji o plikach tj. nazwa, rozszerzenie, rozmiar. W systemie istnieje także rola administratora, który może określić dozwolone rozszerzenia plików.

Ważne linki

Aplikacja:

<http://filemanager-app.s3-website.eu-central-1.amazonaws.com/>

Repozytorium github:

<https://github.com/KrystianKatafoni/file-manager>

Dokumentacja REST API - Swagger:

<http://filemanager-env-1.eba-ps372jjf.us-east-2.elasticbeanstalk.com/swagger-ui.html#/>

Odbiorcy docelowi

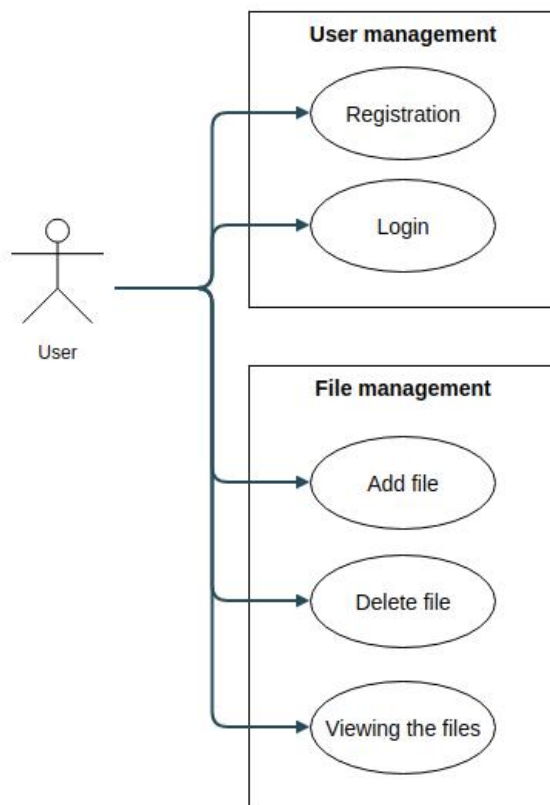
Odbiorcami aplikacji są użytkownicy, którzy mają potrzebę zarządzania plikami oraz przechowywaniem ich w bezpiecznym środowisku.

Aktorzy

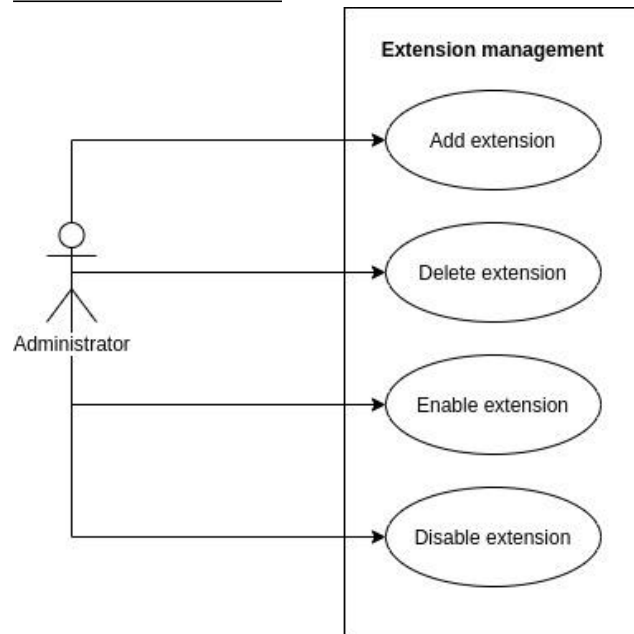
- Użytkownik – osoba korzystająca z głównej funkcjonalności aplikacji tj. zarządzanie plikami.
- Administrator – osoba, która administruje rozszerzeniami tzn. określa dopuszczalne rozszerzenia plików, które mogą być uploadowane do aplikacji przez użytkownika

Wymagania funkcjonalne (use cases)

User:



Administrator:



Wymagania funkcjonalne(user stories)

1. Ja, jako użytkownik chciałbym mieć możliwość rejestrowania się w systemie.
2. Ja, jako użytkownik chciałbym móc zalogować się do aplikacji.
3. Ja, jako użytkownik chciałbym móc dodać plik do systemu, w taki sposób aby był trwale przechowywany.
4. Ja, jako użytkownik chciałbym móc usunąć plik z systemu, co oznacza, że plik nie będzie już przechowywany w aplikacji.
5. Ja, jako użytkownik chciałbym móc uzyskać informacje o plikach tj. nazwa, rozmiar, data stworzenia, login twórcy.
6. Ja, jako użytkownik chciałbym móc ściągnąć na swój dysk wybrany przeze mnie plik.
7. Ja, jako administrator chciałbym móc zarządzać dozwolonymi rozszerzeniami plików.

Wymagania pozafunkcjonalne

1. Architektura systemu powinna opierać się na architekturze REST API.
2. System powinien realizować uwierzytelnianie oraz autoryzację na bazie tokenów JWT.
3. Warstwa kliencka systemu powinna zostać napisana pod popularne przeglądarki internetowe tj. chrome, firefox.
4. Architektura przechowywania plików powinna opierać się o rozwiązanie chmurowe np. AWS S3.
5. Aplikacja powinna zostać uruchomiona w środowisku chmurowym AWS.

Stos technologiczny

1. Backend

- a) Spring Boot
- b) Hibernate
- c) Spring Data JPA
- d) Swagger
- e) Junit5
- f) Lombok
- g) Mapstruct

2. Bazy danych

- a) PostgreSQL - aplikacja
- b) H2 - testy

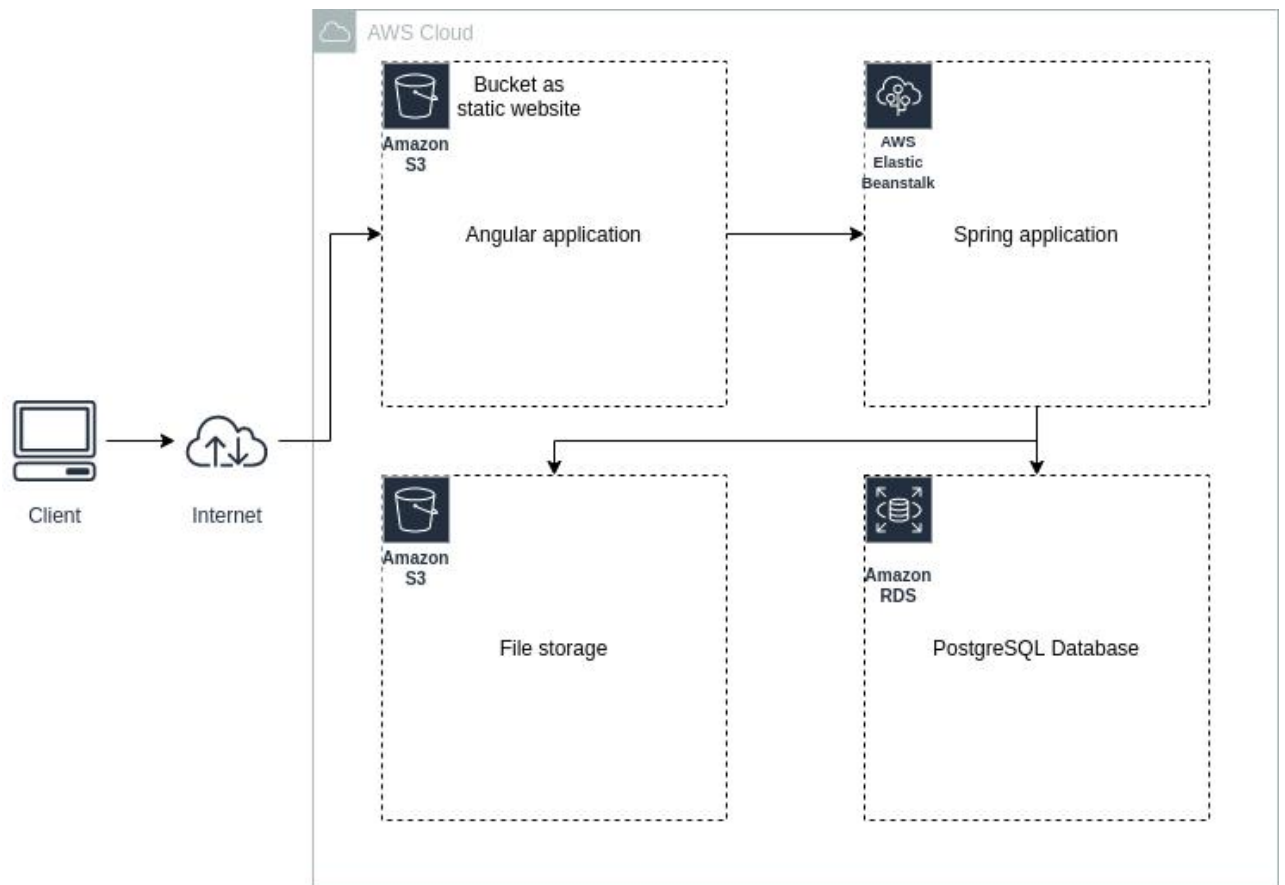
3. Frontend

- a) Angular
- b) Angular Material
- c) TypeScript

4. Development

- a) IDE Backend - IntelliJ Ultimate
- b) IDE Frontend - Webstorm
- c) Docker/Docker compose - uruchamianie bazy
- d) Postman - testowanie API

Architektura



Schemat relacyjnej bazy danych

