

Sistema CRUD de imagens

A aplicação deve fazer o controle de autenticação dos usuários e gerenciar imagens de cada usuário.

Criar uma aplicação web dividida em dois *containers*: sendo uma aplicação backend que forneça uma API para gerenciar autenticação de usuários, gerenciar imagens de um usuário e outra aplicação frontend para integrar todas as funcionalidades de UI (*User Interface*) de forma que consuma as APIs de backend e gerencie os fluxos de entrada, principais eventos de cada funcionalidade da aplicação e controle os fluxos de saída.

1. Aplicação backend

A aplicação backend (criada em DjangoRest) deve ter seu próprio repositório de código hospedado no Github.

A aplicação backend deve prover pelo menos os seguintes serviços:

- registrar novo usuário (usuario, nome completo, email, senha)
- listar todos os usuários registrados
- dado um id de um usuário, mostrar as informações de um usuário específico
- autenticação de usuário usando e-mail e senha
- dado um nome de um arquivo de imagem (jpg ou png), referência do arquivo e id de usuário fazer o upload (tamanho máximo de 10 MB) deste arquivo para o servidor da aplicação de backend
- dado um id de usuário e id de imagem retornar o arquivo da imagem
- dado um id de usuário retorna a quantidade de imagens que o usuário tem
- dado um id de usuário retornar todas as imagens do usuário
- dado um id de usuário e id de imagem retornar as informações (nome do arquivo, path onde o arquivo foi armazenado) da imagem
- dado um id de usuário e id de imagem deletar a imagem

2. Aplicação frontend

A aplicação frontend (baseada no React) deve ter seu próprio repositório de código hospedado no Github.

A aplicação frontend deve gerenciar as seguintes telas:

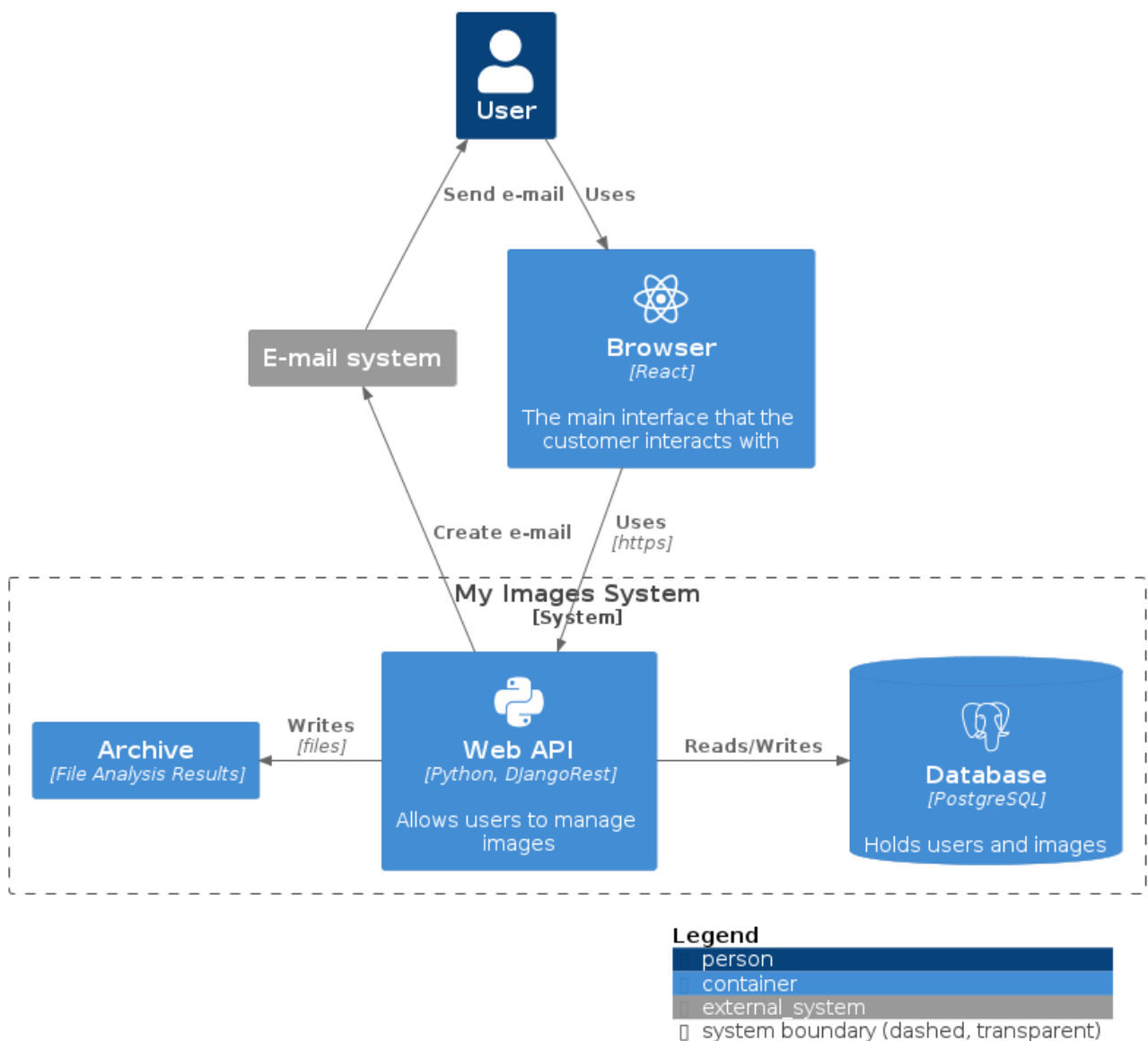
- Tela de registro do usuário (usuario, nome completo, email, senha)
- Tela de login (email, senha)
- Tela de dashboard do usuário
- Tela de upload de imagem
- Tela de listar imagens do usuário
- Tela para exibir imagem selecionada

As telas deverão usar *Material UI*, os componentes devem ser criados usando *functional components* e a biblioteca de consumo de API deve ser o *Axios*.

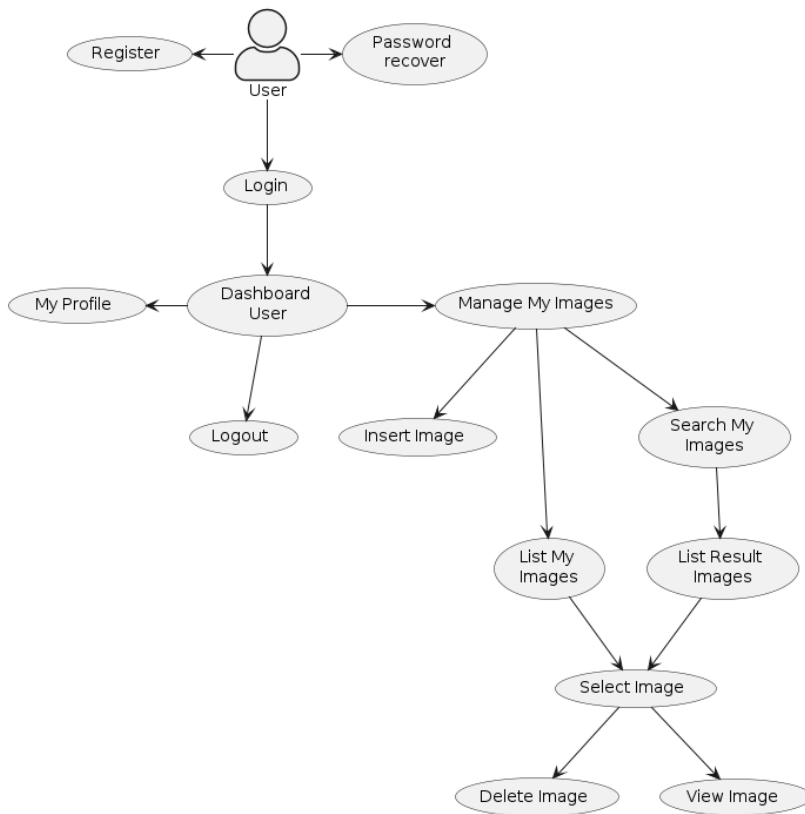
3. Banco de Dados

O banco de dados usado para registrar os dados de usuários e imagens deve ser o PostgreSQL. Deve ser criado um banco para gerenciar a aplicação CRUD de imagens contendo pelo menos uma tabela para gerenciar usuários e outra tabela para gerenciar imagens.

4. Arquitetura sugerida



5. Casos de Uso



6. Funcionalidades

A aplicação deve ter as seguintes funcionalidades:

F1. Login de usuário.

1. Página de login do usuário

Entradas: nome do usuário, senha do usuário

As entradas devem ser validadas.

Fluxo principal

1. Usuário fornece um nome de usuário válido.
2. Usuário fornece um password válido.
3. O sistema loga com as informações do usuário.
4. O sistema carrega a página de dashboad do usuário.

Fluxos alternativos (não é obrigatório neste primeiro momento)

1. Lembrar usuário (*password recover*)
2. O usuário fornece o e-mail do usuário
3. O sistema envia um e-mail de lembrete

Fluxos de exceção

1. Usuário fornece dados inválidos
2. O sistema exibe mensagem "usuário inválido"

F2. Logout de usuário.

Fluxo Principal

1. O usuário deve estar na tela de dashboard
2. O usuário clica em logout
3. O sistema encerra a sessão do usuário
4. O sistema carrega a página de login

F3. Registro de novo usuário.

1. Página de registro de novo usuário

Entradas: nome do usuário, nome completo do usuário, e-mail do usuário, senha do usuário, confirma senha do usuário

As entradas devem ser validadas.

Fluxo principal

1. Usuário fornece um nome de usuário válido.
2. Usuário fornece seu nome completo.
3. Usuário fornece um e-mail válido de usuário.
4. Usuário fornece um password.
5. Usuário confirma o password.
6. O sistema loga com as informações do usuário.
7. O sistema carrega a página de dashboard do usuário.

Fluxos de exceção

1. Usuário fornece nome de usuário já existente
2. O sistema exibe mensagem "usuário já existente"
3. Usuário fornece nome completo de usuário já existente
4. O sistema exibe mensagem "usuário já existente"
5. Usuário fornece e-mail de usuário já existente
6. O sistema exibe mensagem "e-mail já existente"

F4. Dashboard do usuário.

1. Página de dashbord

A página deve ter um cabeçalho (nome do sistema, menu de opções, nome do usuário logado)

A página deve ter um menu de opções (navbar)

A página deve ter um corpo de conteúdo (container)

A página deve ter um rodapé (nome do sistema, data corrente dd/mm/yyyy)

O menu de opções deve ter as seguintes opções:

- Listar minhas imagens
- Fazer o upload de uma imagem
- Buscar uma imagem (não é obrigatório neste primeiro momento)
- Logout

2. A página de dashboard deve ter uma área (corpo de conteúdo) para mostrar cards de informações.

Na área de card de informações, deve existir um card para mostrar a quantidade de imagens do usuário corrente logado.

F5. Listar imagens.

1. Página listar imagens do usuário

Dados para exibição: id da imagem, nome da imagem, thumbnail da imagem, opções (Exibir, Deletar)

Fluxo Principal

1. O sistema deve exibir uma lista de imagens cadastradas pelo usuário.

Fluxos Alternativos

1. Exibir imagem selecionada
2. Deletar Imagem selecionada

Fluxo de exceção

O sistema exibe uma mensagem "Nenhuma imagem cadastrada".

F6. Exibir imagem.

1. Página exibir imagem

Fluxo Principal

1. O usuário seleciona a imagem
2. O sistema exibe a imagem

F7. Upload de imagem.

1. Página de upload de imagem

Dados de entrada: nome da imagem, escolher arquivo
Tipos de imagens válidas: jpg, png

Fluxo Principal

1. O usuário informa um nome de imagem válido
 2. O usuário informa um arquivo válido
 3. O sistema faz o upload da imagem
- O sistema mostra a mensagem "Upload realizado com sucesso!".

Fluxo de exceção

1. O sistema não conclui o upload
2. O sistema deve exibir a mensagem "Upload inválido".

F8. Deletar imagem.

Fluxo principal

1. O sistema lista das imagens do usuário
2. O usuário seleciona uma imagem para deletar
3. O sistema deleta a imagem
4. O sistema deve exibir a lista de imagens atualizadas

Fluxo de exceção

1. O sistema não deleta imagem
2. O sistema deve exibir uma mensagem de erro

F9. Buscar imagem (não é obrigatório neste primeiro momento)

1. Página de buscar imagem

dados de entrada: nome da imagem

Fluxo Principal

1. O usuário informa nome do arquivo
2. O usuário clica no botão procurar
3. O sistema exibe a imagem ou lista de imagens correspondentes ao nome procurado.

Fluxo de Exceção

1. Imagem não encontrada
2. O sistema exibe a mensagem "Nenhuma imagem encontrada!"