

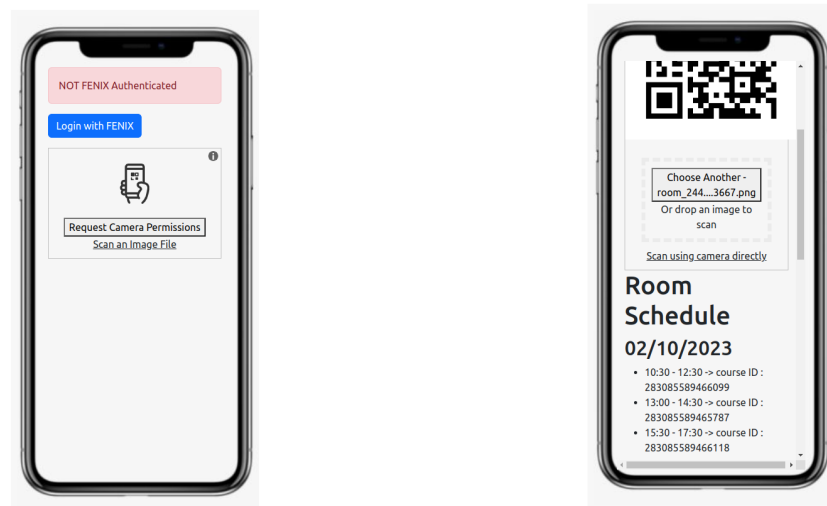
Entrega Final: José Pedro Lopes 100001, Tiago Costa 100094, Miguel Lourenço 100044

Neste relatório relativo à entrega final do projeto, será apenas exposto as alterações e novas implementações para esta submissão bem como alguns detalhes de implementação da nossa aplicação.

Overview

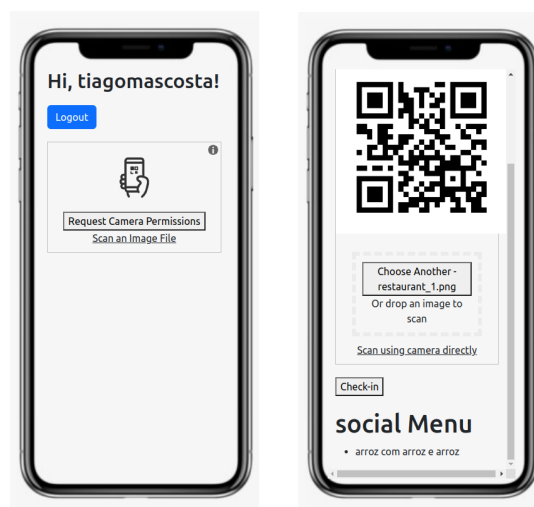
Login screen:

À direita informação pública de uma sala, depois de dar scan num QRCode.

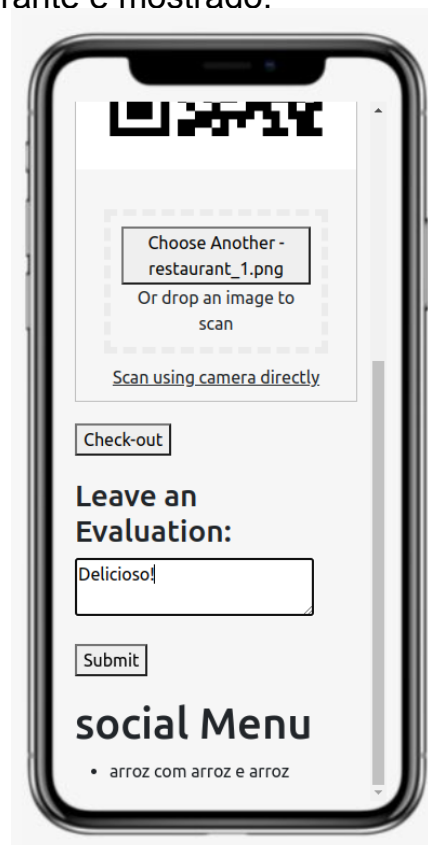


Main screen:

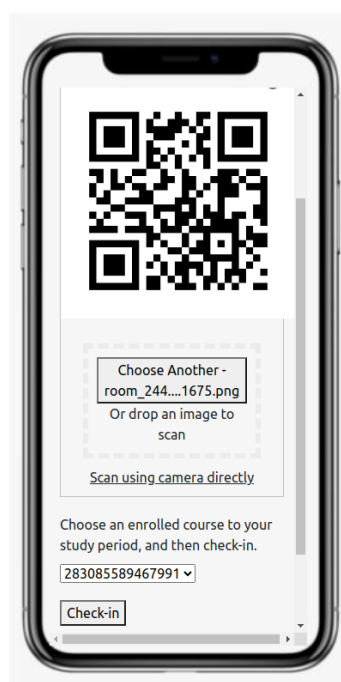
À esquerda logo após dar login e à direita depois de dar scan num QRCode de um restaurante, apresentando o menu e a opção de Check-In.



Assim que efetua o check-in nesse restaurante, um formulário para deixar uma avaliação ao restaurante é mostrado.

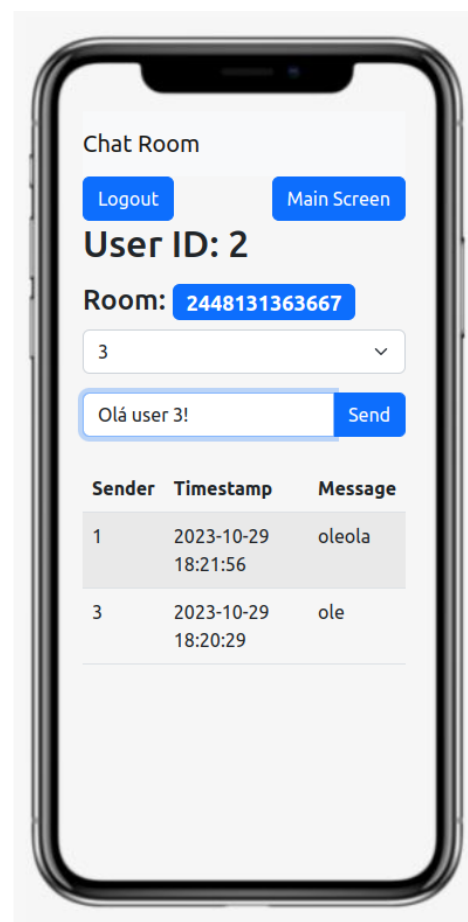
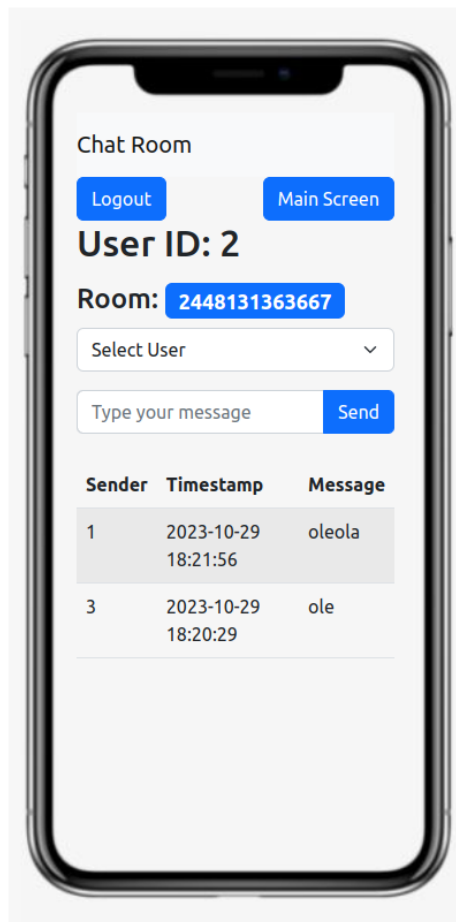


Se o utilizador fizer scan de uma sala aparecerá o horário e a opção de fazer check-in na mesma. Se tiver horário livre, uma opção com as cadeiras em que o aluno está inscrito aparecerá, permitindo-lhe reservar a sala para estudar. À direita temos o ecrã de quando o check-In na sala é efetuado.



Message screen:

Neste screen o utilizador 2 na sala 2448131363667, neste caso, pode consultar as mensagens que utilizadores que estão nessa sala lhe enviaram (imagem da esquerda). Pode ainda seleccionar um dos utilizadores que esteja nessa sala e enviar-lhe uma mensagem (imagem da direita)



QRCodegenerator:

host='0.0.0.0', port=5011

New endpoints:

1. "/API/generateqr": API responsável por gerar e armazenar qr codes de todas as salas e restaurantes criados pelos serviços foodserviceAdmin e roomsservicesAdmin.

FoodService / FoodAdminApp

host='0.0.0.0', port=5003

O FoodAdminAPP é agora capaz de realizar um post no QRCodegenerator sempre que cria um novo restaurante.

New Endpoints:

1. '/API/<int:restaurant_id>/Menu': API endpoint responsável por dar return ao menu de um dado restaurante.
2. '/API/<int:restaurant_id>/Evaluate': API endpoint responsável por aceitar pedidos POST e armazenar uma nova avaliação de um cerco restaurante.

RoomService / RoomAdminApp

host='0.0.0.0', port=5002

O RoomAdminAPP é agora capaz de realizar um post no QRCodegenerator sempre que cria uma nova sala, para além desta nova funcionalidade também é capaz de atualizar qualquer horário através de uma interação com o API do FENIX.

New Endpoints:

1. '/API/<room_id>/schedule': API endpoint responsável por retornar todos os eventos de uma certa sala.

Message:

host='0.0.0.0', port=5005

New Endpoints:

1. "/API/messages/<id>/list": Este API tem como função processar e enviar a lista de mensagens recebidas pelo utilizador com o ID *id*.
2. "/API/sendmessage": Graças a este novo endpoint agora a Message ao receber um pedido POST neste endpoint, guardará uma mensagem enviada de um utilizador para outro na sua base de dados.

Check-In/Check-out:

host='0.0.0.0', port=5004

New Endpoints:

1. “/API/checkIn”: Este API recebe e processa um pedido de Check-In num determinado espaço, atualizando as informações necessárias na sua base de dados.
2. “/API/checkOut”: Este endpoint executa o processo de Check-Out de um determinado utilizador.
3. “/API/users/<roomID>/list”: Este API retribui uma lista dos utilizadores que estão neste momento checked-In na sala *roomID*.

New Database:

A base de dados relativa aos check-ins foi alterada de modo a que guarde dois novos atributos action,: responsável por guardar ações como eating, class ou studying, de modo a ser possível atribuir várias ações aos check-ins realizados; também foi adicionado o atributo course_id de forma a guardar a aula assistida ou a disciplina estudada.

QRContext:

host='0.0.0.0', port= 5000

Introdução:

O QRContext tem uma componente responsável por gerir o HTML + JavaScript, por todos os pedidos realizados aqui pelo user aqui são feitos pedidos HTTP de forma assíncrona para o QRContext Server que realiza a informação do pedido através de pedidos para os APIs dos restantes servers.

Endpoints:

“/” -> página inicial, com apenas o botão de login e o qr code scanner, isto possibilita ao utilizador fazer login com a plataforma do FENIX(F1) ou então ver informações sobre salas ou restaurantes, se scannar um QR correspondente à sala ou ao restaurante(F2 e F6).

“/main_screen”-> Aqui está o ecrã principal, onde necessariamente já se fez o login com a plataforma do FENIX, e sendo assim já possibilita o utilizador de ler um QR code e fazer check-in numa sala ou num restaurante(F3), avaliar a refeição no caso de ser restaurante (F5), no caso de ser uma sala tem a possibilidade de dar check-in numa aula ,no caso de estar a acontecer uma aula das que está matriculado(F7 e F8), caso contrário poderá iniciar um período de estudo e escolher uma cadeira que irá estudar(F10 e F11). Por fim, tanto dentro de uma sala como de um restaurante poderá dar check-out(F4 e F9). Este ecrã também tem a possibilidade de ir para o ecrã das mensagens, se estiver dentro de uma sala.

“/messages”-> Este é o ecrã das mensagens, apenas acessível quando dentro de uma sala, onde se pode enviar mensagens para outros utilizadores dentro da mesma sala(F12), e ver as mensagens que me foram enviadas.

“/logout”-> Este endpoint serve para quando um utilizador faz logout e redireciona para a página inicial

“/authorize/<provider>”/“callback/<provider>”-> Estes endpoints são responsáveis pela autenticação no FENIX, são os que fazem login na aplicação através do FENIX.

“/other”-> Vai buscar os ID's das cadeiras onde o utilizador está inscrito na plataforma do FENIX.

Interação com ROOMS API:

1. '/API/rooms/<room_id>/schedule' -> (F6) Responsável por pedir os dados de um dado horário de uma sala ao respetivo API.

Interação com FOOD API:

1. '/API/restaurant/<restaurant_id>/Menu' -> (F2) Responsável por pedir os dados do menu de certo restaurante ao respetivo API.
2. '/API/restaurant/<restaurant_id>/evaluation' -> (F5) responsável por dar post de uma avaliação feita num certo restaurante. As avaliações são emitidas de forma anônima para que o user não tenha problemas em ser o mais franco e sincero possível.

Interação com Check API:

1. '/API/checkIn' -> (F3 , F8, F10) Responsável por dar post de um Check-in em todas as situações pedidas.
2. '/API/checkOut' -> (F4, F9, F11) Responsável por dar post de um Check-out em todas as situações pedidas.
3. '/API/users/<roomID>/list' -> Responsável por pedir os ids do users que se encontram numa determinada sala.

Interação com Message API:

1. '/API/messages/<myUsername>/list' -> Responsável por pedir a lista de mensagens que o user recebeu.
2. '/API/sendmessage' -> Responsável por dar post de uma mensagem no respetivo API.

Considerações finais:

O programa desenvolvido implementa todas as funcionalidades pedidas. É capaz de realizar tudo isto usando o mínimo de reloads e mudanças de páginas possível. O HTML e o JavaScript realizam as interações com o servidor e este realiza-as com os respectivos serviços.

Quanto às limitações o nosso programa não consegue retornar do message screen para o main screen mantendo as informações que estavam disponibilizadas neste anteriormente, pelo que decidimos que sempre que um user tenta realizar esta tarefa realiza automaticamente um check out da sala onde se encontrava.