Universidad Rafael Landívar

Programación Web

Sección 1

Documentación proyecto de curso

Walter Alexander Osoy Veliz

1126017

Pantalla de inicio de sesión

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El propósito de esta pantalla es identificar que usuario es el que esta intentando acceder a la pantalla principal, y también diferenciar entre clientes nuevo y clientes existentes que ya tiene una actividad previa en la aplicación.

* Botón de “Sign In”:
  + Se debe pedir hacer un get con el usuario y contraseña escritos y en caso de ingresar los datos bien se retornara un json con un bear token
* Botón de “Forgot password?”:
  + Pagina pendiente de crear
  + Se debe pedir el email y nombre de usuario para validar la existencia del usuario y hacer un update al endpoint /api/users/password
* Botón de “Don’t have account?”:
  + Pagina pendiente de crear
  + Se debe pedir los datos al usuario y enviarlos en formato json y enviarlos al endpoitn /api/Users
    - Ejemplo del body

{

"id": 0,

"userName": "string",

"password": "string",

"email": "string",

"name": "string",

"lastName": "string",

"lastLoginDate": "2023-03-31T23:50:18.756Z",

"creationDate": "2023-03-31T23:50:18.756Z"

}

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El objetivo de esta pantalla es mostrarle a nuestro cliente previamente identificado sus cuentas y los recursos que cuenta en cada una de ellas. También puede borrar una o varias cuentas si así lo desea, podrá crear una nueva cuenta en caso de este así lo desee para representar una nueva cuenta en un nuevo banco. También podrá actualizar la cuenta y registrar su actividad en la cuenta.

* Tabla
  + Se hace un get al inicio de la aplicación para obtener todas las cuentas asociadas con el usuario al endpoint
    - Get /Account/UserName
    - Respuesta

[

{

"id": 0,

"userId": 0,

"accountType": "string",

"creationDate": "2023-04-01T00:03:00.554Z",

"amount": 0,

"description": "string"

}

]

* Botón “Create new account”
  + Pagina pendiente de crear
  + Se debe de pedir al usuario los datos de tipo de cuenta, descripción y un moonton inicial.
  + Se creara con el endpoitn
    - Post /Account
    - Body

{

"id": 0,

"userId": 0,

"accountType": "string",

"creationDate": "2023-04-01T00:04:38.949Z",

"amount": 0,

"description": "string"

}

* Botón de icono de basurero:
  + Se eliminaran las cuentas que hayan sido seleccionadas
  + Se debe hacer un delete al endpoint
    - Delete /Account/{UserName}/{id}

Tablas

CREATE TABLE Users (

id INT NOT NULL AUTO\_INCREMENT,

userName VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL,

name VARCHAR(255) NOT NULL,

lastName VARCHAR(255) NOT NULL,

lastLoginDate TIMESTAMP DEFAULT NULL,

creationDate TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id)

);

CREATE TABLE Account (

id INT NOT NULL AUTO\_INCREMENT,

userId INT NOT NULL,

accountType VARCHAR(255) NOT NULL,

creationDate TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

amount DECIMAL(10,2) NOT NULL DEFAULT 0.00,

description VARCHAR(255) DEFAULT NULL,

PRIMARY KEY (id),

FOREIGN KEY (userId) REFERENCES Users(id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Objects (

id INT NOT NULL AUTO\_INCREMENT,

userId INT NOT NULL,

accountId INT NOT NULL,

creationDate TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

description VARCHAR(255) DEFAULT NULL,

PRIMARY KEY (id),

FOREIGN KEY (accountId) REFERENCES Account(id) ON DELETE CASCADE ON UPDATE CASCADE

);