Method createNewAccount(username: string, password: string):

Take the input of a password and check it with the criteria

Create a new user account with the provided username and password

Method deleteUser(userID: int):

Confirm account deletion

Delete the user with the given ID from the database

Method getUser(userID: int):

Retrieve the user with the given ID from the database

Method updateUser(userID: int, newUser: User):

Input new username

Input new passowrd

Update the user with the given ID with the provided user data

Method addAdmin(username: string, password: string):

Confirm admin creation

Take the input of a password and check it with the criteria

Add a new admin with the provided username and password (and checked password to meet criteria)

Method deleteAdmin(adminID: int):

Confirm account deletion

Delete the admin with the given ID if it exists from the database

Method getAdmin(adminID: int):

Take adminID

Retrieve the admin with the given ID from the database

Method updateAdmin(adminID: int, newAdmin: Admin):

Input new username

Input new password and check it to the criteria

You chooseAccountType

Update the admin with the given ID with the provided admin data

Method createRestaurant(name: string, location: string):

Make sure User is an admin

Create a new restaurant with the provided name and location

Method deleteRestaurant(restaurantID: int):

Take restaurantID and confirm it exists

Confirm restaurant deletion

Delete the restaurant with the given ID from the database

Method getRestaurant(restaurantID: int):

Retrieve the restaurant with the given ID from the database and return it

Method updateRestaurant(restaurantID: int, newRestaurant: Restaurant):

Take restaurantID

Take new restaurant name

Take new restaurant location

Update the restaurant with the given ID with the provided restaurant data

Method createTipOutMethod(name: string, description: string, type: TipOutMethodType):

Create a new tip out method with the provided name, description, and type

Method deleteTipOutMethod(methodID: int):

Take the methodID

Delete the tip out method with the given ID from the database, if it exists

Method getTipOutMethod(methodID: int):

Take the methodID

Retrieve the tip out method with the given ID from the database

Method updateTipOutMethod(methodID: int, newMethod: tipOutMethod):

Get methodID

Update the tip out method with the given ID with the provided method data

Method sendMessage(senderID: int, receiverID: int, content: string):

Get the user ID of the sender

Get the user ID of the reciever

Send a message from the sender with the given ID to the receiver with the given ID containing the provided content

Method readMessage(messageID: int):

Take messageID

Get the message linked to the ID

Read the message with the given ID from the database

Method deleteMessage(messageID: int):

Confirm you want to remove the Message

Delete the message with the given ID from the database if it exists

Method addInformation(userID: int, info: string):

Get the User based on their userID

Add information to instance

Add the provided information to the user with the given ID

Method chooseAccountType(userID: int, type: AccountType):

Decide if you want to be an admin or not

Decide if you want to be an Employee or not

Leave blank to be stand-alone

Choose the account type for the user with the given ID to the provided type

Method setNewAccount(users: User[]):

Take the user

Set up new account for the user

Method getNewAccount():

Retrieve the details of the newly set account

Method setAccountInformation(username: string, password: string):

Take the username

Take the password

Set the account information with the provided username and password

Method getAccountInformation():

Retrieve and display the current account's username and password

Method createAccountSettings(userID: int, settings: Settings[]):

Take the userID

Create new settings

Create account settings for the specified user ID with the given settings array

Method setAccountType(accountType: User[]):

Set the type of account based on the user data provided

Method getAccountType():

Retrieve and display the type of the current account