Here's a plain English pseudocode representation of the provided code:

UserManagerApplication:

- Import necessary libraries/packages

- Declare a class called "UserManagerApplication"

- Define the main method

- Run the Spring application

AppErrorController:

- Declare a class called "AppErrorController" that implements ErrorController

- Define a method called "handleError"

- Perform error handling actions (e.g., logging)

- Return the string "error"

- Override the "getErrorPath" method

- Return null

MainController:

- Declare a class called "MainController"

- Autowire the UserService

- Define a Logger and currID variable

- Define a method called "showHomePage"

- Accept a ModelMap and a name parameter

- Add the "name" parameter to the model

- Return the string "home"

- Define a method called "showIndexPage"

- Accept namelogin, passwordlogin, and modelMap parameters

- Try to retrieve a User by the given namelogin

- If the retrieved User's name and password match the given namelogin and passwordlogin

- Return the string "index"

- Else

- Return the string "home"

- Define a method called "isNumber"

- Accept a string parameter

- Check if the string is not null and can be parsed as a double

- Return a boolean indicating the result

- Define a method called "saveDetails"

- Accept id and modelMap parameters

- Try to retrieve a User by the given id

- If the User exists

- Create an ArrayList containing the User

- Set the model's "user" attribute to the ArrayList

- Set the currID variable to the given id

- Else

- Return the string "nouser"

- Set the model's "ID" attribute to the given id

- Return the string "update"

- Define a method called "updateDetails"

- Accept nameedit, emailedit, passwordedit, and modelMap parameters

- Create an ArrayList

- Try to retrieve a User by the currID value

- Update the User's name, email, and password with the given values

- Add the User to the ArrayList

- Set the model's "user" attribute to the ArrayList

- Set the model's "IDedit" attribute to the currID value

- Return the string "update2"

UserController:

- Declare a class called "UserController"

- Autowire the UserService

- Define a Logger variable

- Define a method called "showUsers"

- Accept a ModelMap parameter

- Retrieve all Users from the UserService

- Add the Users to the model

- Return the string "users"

UserExceptionController:

- Declare a class called "UserExceptionController"

- Define a method called "exception" that handles UserNotFoundException

- Return a ResponseEntity with the message "User not found" and status NOT\_FOUND

User:

- Declare a class called "User"

- Define private variables for id, name, email, and password

- Include getter and setter methods for the variables

- Override the toString method

UserNotFoundException:

- Declare a class called "UserNotFoundException" that extends RuntimeException

UserRepository:

- Declare an interface called "UserRepository" that extends CrudRepository<User, Integer>

- Include a method to find a User by name

UserService:

- Declare a class called "UserService"

- Autowire the UserRepository

- Define methods to get all Users, get a User by name, get a User by id, update a User, and set User details

Application.properties:

- Configure application properties

.