

<b>startUserManagement(loginData, userChangeType)</b>
<p><b>Description:</b> Starts user management action, with the options to create and change user data and delete users.</p> <p><b>Precondition:</b> loginData must be right</p> <p><b>Postcondition:</b> User management is possible or the authorization must have failed</p>
<p><b>Returns:</b> Error, if there was a problem by invoking this function</p> <p><b>Exceptions:</b> Throws exception if is unauthorized</p> <p><b>Result:</b> If user is authorized userManagement will be started (user gets acces to user management operations), otherwise he will be informed that he is unauthorized</p>
<p><b>Type:</b> System - operation</p> <p><b>References:</b> startUserManagement(loginData, userChangeType), createUser(nu: newUserData), changeUserData(oldUserID), changedUser(changedUserData), deleteUser(userID)</p> <p><b>Comment:</b> -</p>

<b>CreateUser(nu: newUserData)</b>
<p><b>Description:</b> Creates new user if data is valid. Otherwise he will be informed that the access was denied</p> <p><b>Precondition:</b> User management has started, user is authorized, userChangeType == newUser,</p> <p><b>Postcondition:</b> New user created or user informed about the failure. (User index may be updated)</p>
<p><b>Returns:</b> Nothing</p> <p><b>Exceptions:</b> If user already exists, or if the creation of a new user failed.</p> <p><b>Result:</b> Confirmation that a new user was created successfully</p>
<p><b>Type:</b> System - operation</p> <p><b>References:</b> startUserManagement(loginData, userChangeType), createUser(nu: newUserData), changeUserData(oldUserID), changedUser(changedUserData), deleteUser(userID)</p> <p><b>Comment:</b> -</p>

<b>changeUserData(oldUserID)</b>
<b>Description:</b> Changes user data if userID is valid <b>Precondition:</b> Login was valid and user management is running <b>Postcondition:</b> User can now enter new user data, otherwise the operation has failed
<b>Returns:</b> Nothing <b>Exceptions:</b> Throw exception if userID is invalid <b>Result:</b> User can now enter new user data, otherwise the operation has failed
<b>Type:</b> System - operation <b>References:</b> startUserManagement(loginData, userChangeType), createUser(nu: newUserData), changeUserData(oldUserID), changedUser(changedUserData), deleteUser(userID)  <b>Comment:</b>

<b>changedUser(changedUserData)</b>
<b>Description:</b> Changes the user data of the previous entered user if user data is valid <b>Precondition:</b> changeUserData was executed successfully <b>Postcondition:</b> Userdata are changed or the operation failed
<b>Returns:</b> Nothing <b>Exceptions:</b> Throws exception if userdata is invalid <b>Result:</b> Userdata are changed or user gets notified about the failure
<b>Type:</b> System - operation <b>References:</b> startUserManagement(loginData, userChangeType), createUser(nu: newUserData), changeUserData(oldUserID), changedUser(changedUserData), deleteUser(userID) <b>Comment:</b>

<b>deleteUser(userID)</b>
<b>Description:</b> Deletes user and all references of him, with the entered userID if he exists <b>Precondition:</b> User management action is started and user is authorized

**Postcondition:**

User with ID userID was deleted with every single record he had.  
Otherwise user gets notification that it was not possible

**Returns:**

Nothing

**Exceptions:**

Throws exception if user with requested ID didn't exist.

**Result:**

User with ID userID was deleted with every single record he had.  
Otherwise user gets notification that it was not possible

**Type:**

System - operation

**References:**

startUserManagement(loginData, userChangeType), createUser(newUserData), changeUserData(oldUserID),  
changedUser(changedUserData), deleteUser(userID)

**Comment:**