

### Retrieve All Users

Use Case: Retrieve All Users

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of admin is to view existing profiles in the lobby service

Multiplicity: Multiple admins can view this information at the same time.

Primary Actor: Admin

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* informs *System* of desire to retrieve active profiles.

2. [\*System\* requests \*User's\* role](#) to *Lobby Service*

Only occurs if the requesting user is admin.

3. *System* requests data from *Lobby Service* and displays it to the user.

Extensions:

2.a. *Lobby Service* informs *System* that *User* is non-admin. Use-case ends in failure.

### Get User Details

Use Case: Get User Details

Scope: Lobby Service Client

Level: Subfunction

Intention in Context: Intention of user is to retrieve the profile details of a user

Multiplicity: Multiple users can request the information of multiple users at a time

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* informs *System* of desire to view user details

2. [\*System\* requests \*User's\* role](#) to *Lobby Service*

3. *System* forwards this request to *Lobby Service* and displays the information to the *User*

Extensions:

This can occur if the requesting user is non-admin and requests data for other users, among others.

3.a. *Lobby Service* fails to validate the request, *System* informs *User*, use case ends in failure.

### Create User Account

Use Case: Create User Account

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of user is to create a new user profile

Multiplicity: Multiple users can create one account at a time.

Primary Actor: User

Secondary Actor: Lobby Service

Main Success Scenario:

1. *User* provides to *System* the requisite data to create a new profile
2. [System requests User's role](#) to *Lobby Service*

Only occurs if user is admin

3. *System* forwards this data and request to *Lobby Service*

Extensions:

- 2.a. *Lobby Service* informs *System* that *User* is non-admin. Use-case ends in failure.
- 3.a. *Lobby Service* fails to validate the request (e.g non-compliant password), *System* informs *User*, use case ends in failure

Delete User Account

Use Case: Delete User Account

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of user is to delete a certain user account

Multiplicity: Multiple users can delete one account at a time

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* provides to *System* required data to delete a user
2. [System requests User's role](#) to *Lobby Service*

Only occurs if user is admin

3. *System* forwards this request to *Lobby Service*.

Update User Password

Use Case: Update User Password

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of user is to update a user password

Multiplicity: Multiple users can update one password at a time.

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* provides to the *System* the requisite data to update the password and makes the request.
2. *System* forwards the data and the request to *Lobby Service* and responds to the *User* indicating success.

Extensions:

- 1a. If *User* is not admin, *User* must provide to *System* extra data, indicating the current password.
- 2a. If *Lobby Service* fails to validate this request (e.g. non-compliant password) the *System* displays failure to the *User*. The use case ends in failure.

### Change Preferred Colour

Use Case: Change Preferred Colour

Scope: Lobby Service Client

Level: User Goal

Intention in Context: User is to change a target user's preferred colour.

Multiplicity: Many users can change one preferred colour at a time.

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* provides to *System* required data to specify new colour, and requests the change.
2. *System* forwards this request to the *Lobby Service*.

### *Get OAuth Role*

Use Case: Get OAuth Role

Scope: Lobby Service Client

Level: Subfunction

Intention in Context: Intention of Lobby Service is to retrieve and send User Role data to System.

Multiplicity: One Lobby Service can handle multiple role requests at a time.

Primary Actor: Lobby Service

Main Success Scenario:

1. *System* forwards role request data and requests role from *Lobby Service*
2. *Lobby Service* responds to the *System* with the role of the requested user.

### *Get OAuth2 Token and Renew Token*

Use Case: Get OAuth2 Token and Renew Token

Scope: Lobby Service Client

Level: Subfunction

Intention in Context: Intention of *Lobby Service* is to generate a new oauth token and send it to the *System*.

Multiplicity: One *Lobby Service* can handle multiple token requests at a time.

Primary Actor: *Lobby Service*

Main Success Scenario:

1. *System* forwards token request data and request to *Lobby Service*
2. *Lobby Service* responds to the *System* by returning the token pair for the target user.

Extensions

- 2.a. If *Lobby Service* is unable to identify a user by the request data it informs *System* and use-case ends in failure.

### *Revoke OAuth2 Token and Renew Token*

Use Case: Revoke OAuth2 Token and Renew Token

Scope: Lobby Service Client

Level: Subfunction

Intention in Context: Intention of *Lobby Service* is to revoke a users oauth tokens.

Multiplicity: One *Lobby Service* can revoke many tokens simultaneously.

Primary Actor: *Lobby Service*

Main Success Scenario:

1. *System* informs *Lobby Service* of delete token and forwards required data
2. *Lobby Service* responds to the System indicating success.

### Log in

Use Case: Log in

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of User is to identify themselves

Multiplicity: Multiple users can log in simultaneously.

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* provides log in data and request to the System.
2. *System* [gets oauth2 token](#) and renew token for provided data and responds to user indicating success by changing to matchmaking screen.

Extensions:

- 2.a. If get token fails, *Lobby Service* informs *User* and the use case continues at step 1.

### Log Out

Use Case: Log Out

Scope: Lobby Service Client

Level: User Goal

Intention in Context: Intention of User is to log out.

Multiplicity: Multiple users can log out simultaneously.

Primary Actor: User

Secondary Actors: Lobby Service

Main Success Scenario:

1. *User* informs *System* of desire to log out.
2. *System* [revokes oauth2 token](#) and responds to user by changing to log in screen.