Check if user can be deleted

Overview

A requirement exits to delete users. Before deleting a user the system needs to check if the user has any
- open loans,
- open requests,
- unpaid fees/fines,
- unexpired proxy, or
- manual blocks (automated block should be covered by open loans or fees/fines).

If any of the defined checks is true, the user cannot be deleted. In this case the response shall include information about the checks, so that the user can be processed by library staff members.

That means, no loans, requests, fees/fines etc. are deleted automatically by this requirement.

The functionality shall be usable via the ui.

In addition, there exists the requirement to have an API endpoint do the checks only without deletion. It needs to be checked (e.g. via the campus management system) if there is an open fee/fine that needs to be paid before leaving. The user may be deleted only one year later (to create reports for example).

Related Jira Issues

The tickets we are working on are the following:

- **UXPROD-2728** - Delete inactive user through the API with dependency check
  - OPEN
- **UXPROD-2388** - Delete user through the UI
  - OPEN
- **UIU-1971** - Delete inactive user through the UI with dependency check
  - OPEN

Proposed approach

As the requirements are not only to trigger the checks from the frontend, but also to perform references-checks from different systems, the checks cannot be performed in the frontend only. Hence, implementing the checks into the front end is considered not as a viable approach. Instead, the checks will be implemented in mod-users-bl so that the front end and different systems may use them.

- First of all we need to implement functionality to check if the user can be deleted:
  - As this checks span multiple modules (e.g. checking open requests via /request-storage endpoint of mod-circulation-storage) the checks need to be done in a bl-module, namely mod-users-bl.
  - In order to check if a user can be deleted, add the endpoint /bl-users/by-id/{id}/deletable to mod-bl-users. (There is no convention or standard way to do this. However, the _deletable_ endpoint was discussed in)
    - CIRC-179 in-use loan policies may be deleted  DRAFT
  - The proposed endpoint performs defined checks for a given user by querying the corresponding modules:
    1. Fetch open loans from mod-circulation-storage via user's id: /loan-storage/loans?query=(userId==c926be9c-a8ce-4399-a9b3-11ec0fc8d6c9 AND status.name=="Open")
    2. Fetch open requests from mod-circulation-storage via user's id: /request-storage/requests?query=(requesterId=="c926be9c-a8ce-4399-a9b3-11ec0fc8d6c9" AND status==Open)
    3. Fetch open fees/fines from mod-feesfines via user's id:  /accounts?query=(userId=="54f65a75-f35b-4f56-86a6-fa4a3d957e57" AND status.name=="Open")
    4. Fetch proxy info from mod-users via user's id: /proxiesfor?query=(userId=="54f65a75-f35b-4f56-86a6-fa4a3d957e57" OR proxyUserId=="54f65a75-f35b-4f56-86a6-fa4a3d957e57")
    5. Fetch manual blocks from mod-feesfines via user's id: /manualblocks?query=(userId=="54f65a75-f35b-4f56-86a6-fa4a3d957e57")
  - Hence, we need to query three modules: `mod-circulation-storage`, `mod-feesfines` and `mod-users`.
  - The endpoint /bl-users/by-id/{id}/deletable returns if the user can be deleted or not. It can be deleted if totalRecords==0 for all responses of above queries.
The response may look like:

```json
{
    "userID": "uuid-1234",
    "message": "not deletable",
    "deletable": false,
    "loans": 1,
    "requests": 0,
    "fees/fines": 2,
    "proxies": 0,
    "blocks": 0
}
```

Based on this we can conclude that the user has one open loan and 2 open fees/fines.

- In order to delete a user, add a DELETE endpoint to `/bl-users/by-id/{id}`
  - This endpoint first calls `/bl-users/by-id/{id}/deletable` (or its implementation) to check if the user can be deleted. If this is the case, it calls DELETE on `/users/{id}`). If the user cannot be deleted, as some above mentioned resources are pointing to this user, the response shall clarify this.
- When deleting a user from the ui, the frontend must call `/bl-users/by-id/{id}`.

Another approach might be that DELETE `/users/{id}` calls `/bl-users/by-id/{id}/deletable` however this would assume mod-users needs mod-users-bl which would end in a circular dependency.

Open questions

- Is it the correct approach to add a DELETE endpoint to `/bl-users/by-id/{id}`, and calling DELETE on `/users/{id}` from there?
- Is it okay to call DELETE on `/bl-users/by-id/{id}` from ui-users?
- Does this approach work when doing batch deletes of users?