Fees/fines refactoring

This document outlines issues with current fees/fines functionality implementation and proposes solutions to these issues.

Move fee/fine actions calculation to BE

Currently, calculations for fee/fine actions are happening on FE without any checks on the BE side. Instead, when a value is entered in the amount field ("Payment amount", "Waive amount" etc.) a call to one of the new endpoints should be made in order to check if this value is valid and if the action is allowed:

- POST /accounts/{accountId}/check-pay
- POST /accounts/{accountId}/check-waive
- POST /accounts/{accountId}/check-transfer
- POST /accounts/{accountId}/check-refund

Body:

```json
{
  "amount": "1.00"
}
```

Response

In case of success:

Status code: 200
Response body:

```json
{
  "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
  "amount": "1.00",
  "allowed": true,
  "remainingAmount": "9.00"
}
```

In case fee/fine was not found:

Status code: 422
Response body:

```json
{
  "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
  "amount": "1.00",
  "allowed": false,
  "errorMessage": "Fee/fine was not found"
}
```

In case fee/fine is already closed:

Status code: 422
Response body:

```json
{
  "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
  "amount": "1.00",
  "allowed": false,
  "errorMessage": "Fee/fine is already closed"
}
```
In case if amount is zero or negative:

Status code: 422  
Response body:

```json
{
    "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
    "amount": "1.00",
    "allowed": false,
    "errorMessage": "Amount must be positive"
}
```

In case if the amount is too high:

Status code: 422  
Response body:

```json
{
    "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
    "amount": "1.00",
    "allowed": false,
    "errorMessage": "Requested amount exceeds remaining amount"
}
```

In case of invalid amount value (e.g. negative or not parsable):

Status code: 422  
Response body:

```json
{
    "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
    "amount": "abcdefg",
    "allowed": false,
    "errorMessage": "Invalid amount entered"
}
```

After receiving a positive response from BE ("allowed": true), when a user proceeds with the fee/fine action, instead of updating Account and creating FeeFineAction objects directly UI needs to call one of the new endpoints with the same body:

- POST /accounts/{accountId}/pay
- POST /accounts/{accountId}/waive
- POST /accounts/{accountId}/transfer
- POST /accounts/{accountId}/refund

**Pay**

Body:

```json
{
    "amount": "1.00",
    "comments": "STAFF : comment for staff \n PATRON : comment for patron",
    "transactionInfo": "check #3456",
    "notify Patron": true,
    "service Point Id": "7c5abc9f-f3d7-4856-b8d7-6712462ca007",
    "userName": "Folio, James",
    "paymentMethod": "Cash"
}
```
BE should take care of updating Account object and creating new FeeFineAction object.

Response

In case of success:

Status code: 201
Response body:

```
{
    "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
    "amount": "1.00"
}
```

FE's responsibility is to (re)load Account and new FeeFineAction objects and update the page accordingly.

In case of a failure (which is still possible - for example, someone else can waive a fine that we're trying to pay between "check" and "perform" calls):

Status code: 422
Response body:

```
{
    "accountId": "e74d50c9-0c69-4f80-9e1b-a819719fc0c9",
    "amount": "1.00",
    "errorMessage": "Requested amount exceeds remaining amount"
}
```