**Design specification**

1. Database Design
2. Background Jobs
3. API Design

Database Designs

**Accounts**

|  |  |  |
| --- | --- | --- |
| Field | Type | Remarks |
| \*AccountNum | Text | Primary Key |
| Name | Text |  |
| AccountType | Enum | Saving, Creditcard, Loan |
| Balance | Decimal |  |
| Outstanding | Decimal |  |

**Customers**

|  |  |  |
| --- | --- | --- |
| Field | Type | Remarks |
| \*CustomerId | Text | Primary Key |
| GivenName | Text |  |
| FamilyName | Text |  |

**Transactions**

|  |  |  |
| --- | --- | --- |
| Field | Type | Remarks |
| \*Id | Int | Primary Key |
| AccountNum | Text | Ref: Accounts.AccountNum |
| CustomerId | Text | Ref: Customers.CustomerId |
| TransactionDate | Date |  |
| TransactionTime | Time |  |
| Description | Text |  |
| Amount | Decimal |  |

All tables should be soft delete enable, and having the fields to track creation, and lastmodify for audit needs.  
1. isDeleted  
2. createdBy  
3. createdTime  
4. lastModifyBy  
5. lastModifyTime

*createdBy* and *lastModifyBy* should be the *name of services*(if the data are last CRUD by service automatically), or the *usernam*e from JWT claim.

Background Jobs

**Folders**

1. /incoming

Entry folder for new interface file

1. /inprogress

Once the file from /incoming is recognized to processing, move it to here as immediate duplicate run prevention, in the situation of restart or etc.

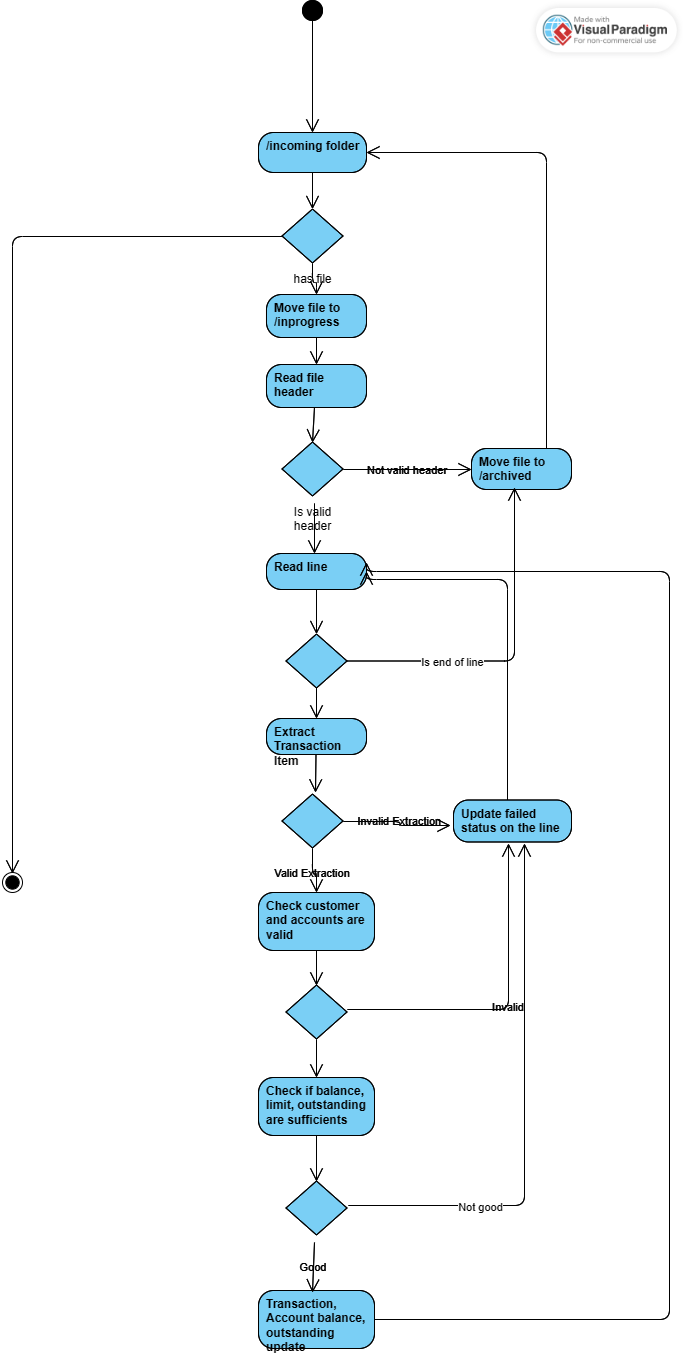
1. /archived

The final destination of the file, once the job run against it.

1. /result

Copy of run of the interface files, file name appended with {yyyy-mm-dd…} to indivate the time of the file being processed, also to avoid the duplication of file name.

**Activity Diagram**



API Design(RESTful)

**Endpoints**

1. **POST** /auth

Generate and response the JWT token in plaintext, *username* and *password* are required

1. **GET** /api/transaction/search

Request/response format should meet needs for pagination: *page*, *limit*, *count*

Always return status 200 for authorized request.

GET /api/transaction/:id  
Retrieve the transaction details by id  
Response status:  
200 for success update

404 for invalid transaction id

1. **PATCH** /api/transaction/:id

Update description field, database transaction are required, once all the update is done then only commit the transaction, otherwise always rollback the transaction as the fallback.  
Response status:

202 for success update

404 for invalid transaction id

409 for the conflict from multiple update

**Security**

1. Except /auth, all other available endpoints should be protected by OAuth2’s token validation
2. Token should be expired after 15 minutes
3. The secret value in configuration should be in valid length on HMACSHA256.