Moneyware

Implementation Approach

Contents

[1. Overview of Changes 3](#_Toc118805689)

[Tranche 1 3](#_Toc118805690)

[Tranche 2 3](#_Toc118805691)

[2. Implementation Approach 3](#_Toc118805692)

[3. Process Flow 4](#_Toc118805693)

[4. Schema 5](#_Toc118805694)

[5. Execution Screenshots 5](#_Toc118805695)

# Overview of Changes

## Tranche 1

Moneyware Ltd is a financial institution, who wants to digitalize the indexing and archiving process of the scanned documents from its client’s customer related documents such as Identity card, Proof of Address.

1. Login of staff.
2. 2) Upload of scanned documents and classified as Document Type (Personal, Address,..). Below details need to be kept as well: File, File size, Filename, Unique ID, Timestamp, customer ID, Document Type, Status
3. 3) The file details are saved with a status ‘Completed’ and we need to allow reading of the document database every day, each one hour based on this status. Once the job reads the db record, it should generate the file on a specific folder. That folder path can change over time.
4. 4) To be able to identify each document, an index file is needed. For each one document generation, an index file with a unique name (DS+Timestamp) will be generated equally and have below format. This file shall contain a generic comment, Timestamp, customer\_id, document\_type, filename.

## Tranche 2

Once the file and the corresponding index file has been generated, you need to upload them to a FTP server without a manual upload process.

# Implementation Approach

1. Moneyware Web Portal (Front end)
2. User logins to Moneyware Web Portal and is routed to document upload page where he selects the document type and browses to select the file and then clicks on Upload button.
3. Upload button initiates a Rest API call to backend Document Upload Service and sends the selected file along with Username and Document Type as Form Data.

2. Moneyware Document Upload API (Back end)

a. The Document Upload Service receives the file as MultipartFile and will save its Base64 encoded byte array with other file details to Document Details table.

b. The status of document will be set to Completed marking upload is complete.

3. Moneyware Scheduler (Back end)

a. The scheduler runs every hour and picks all files having status as Completed. It will perform Base64 decode on the byte array and save it as file to the file location as per application.properties.

b. It will generate index file as text file with the Document Id in its file name along with timestamp and save to the same location.

c. It will update the document status in document\_details table as Archived and insert corresponding new entry in archival\_details table with the filepaths of the archived file and the uploaded file.

# Process Flow

Open Moneyware Portal

Login with User credentials

Select document type and upload the document

Document exists in system

Insert new record to DB with the document details with status= Completed

No

End

Scheduler runs every hour and scans document table

Check if any records in Completed status

Save document from DB to archival location

Generate index file with document details and place to same path

Update document status =Archived in document table

Yes

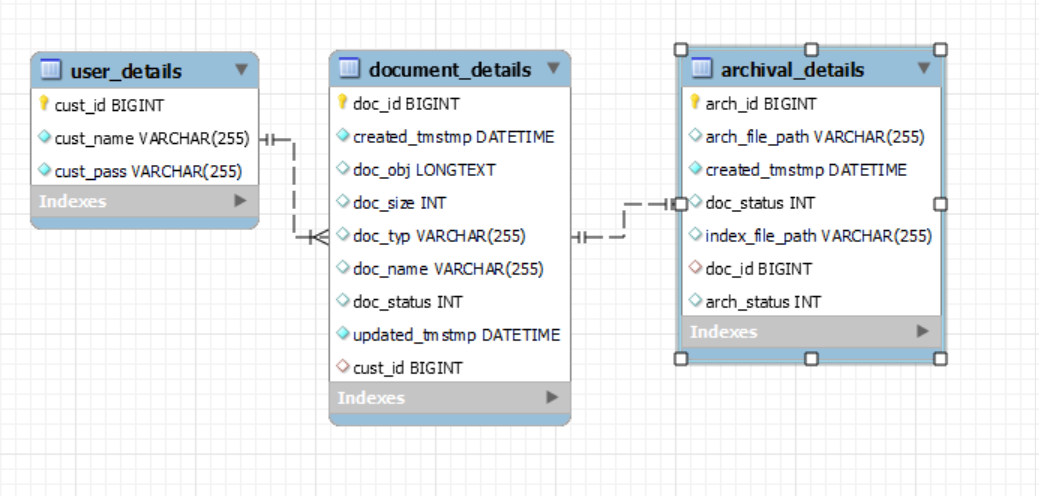
Save archival details to DB

End

Web Application Flow

Scheduler Flow

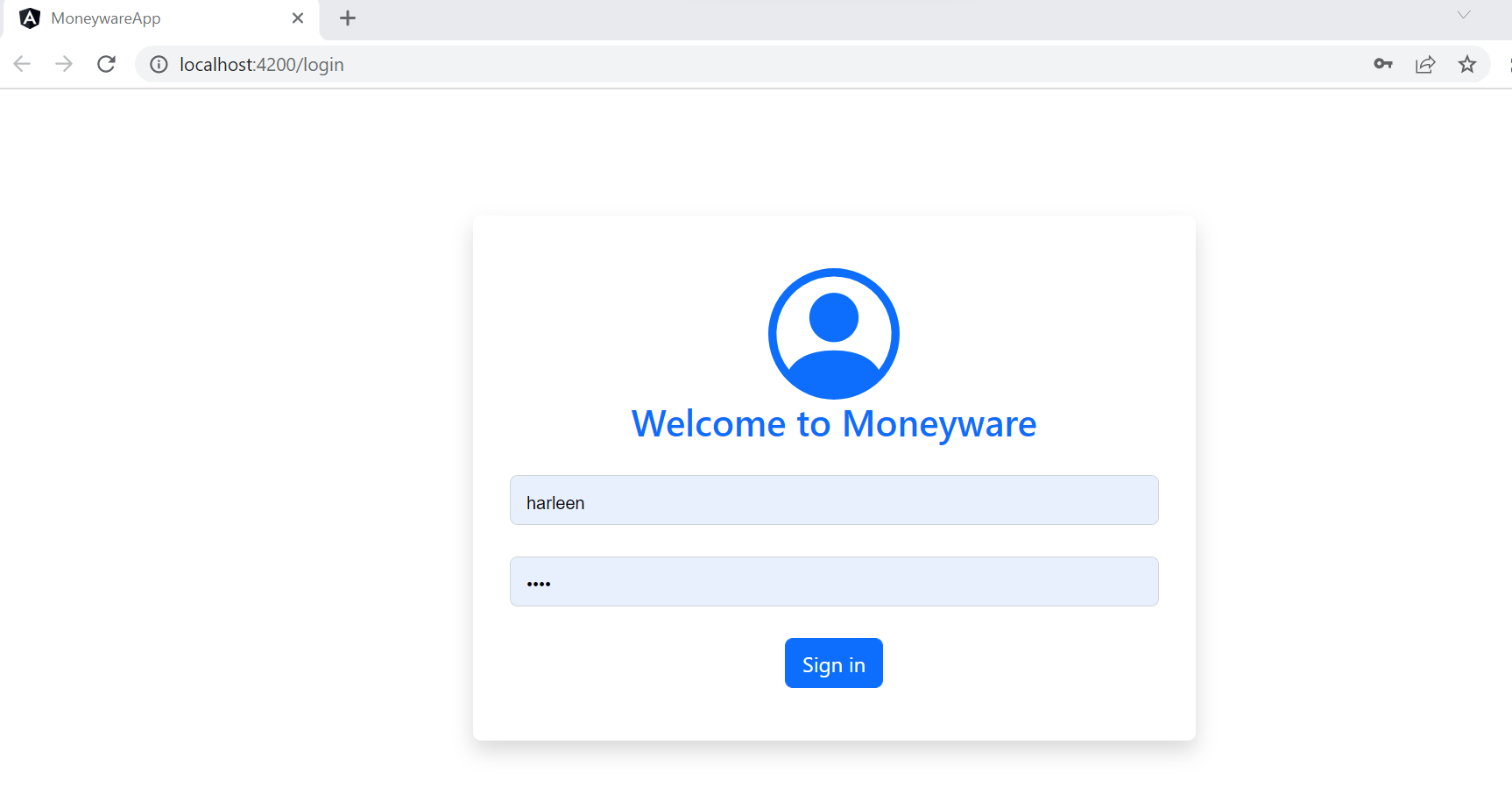
# Schema



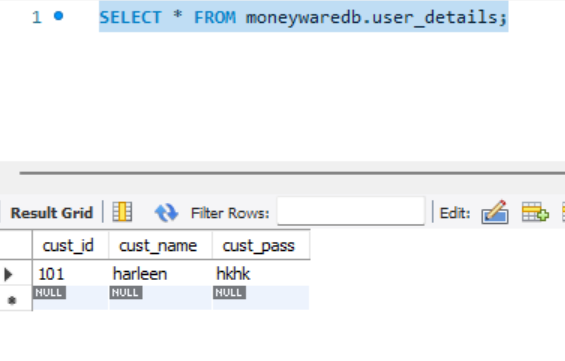
Schema name: moneywaredb

## Execution Screenshots

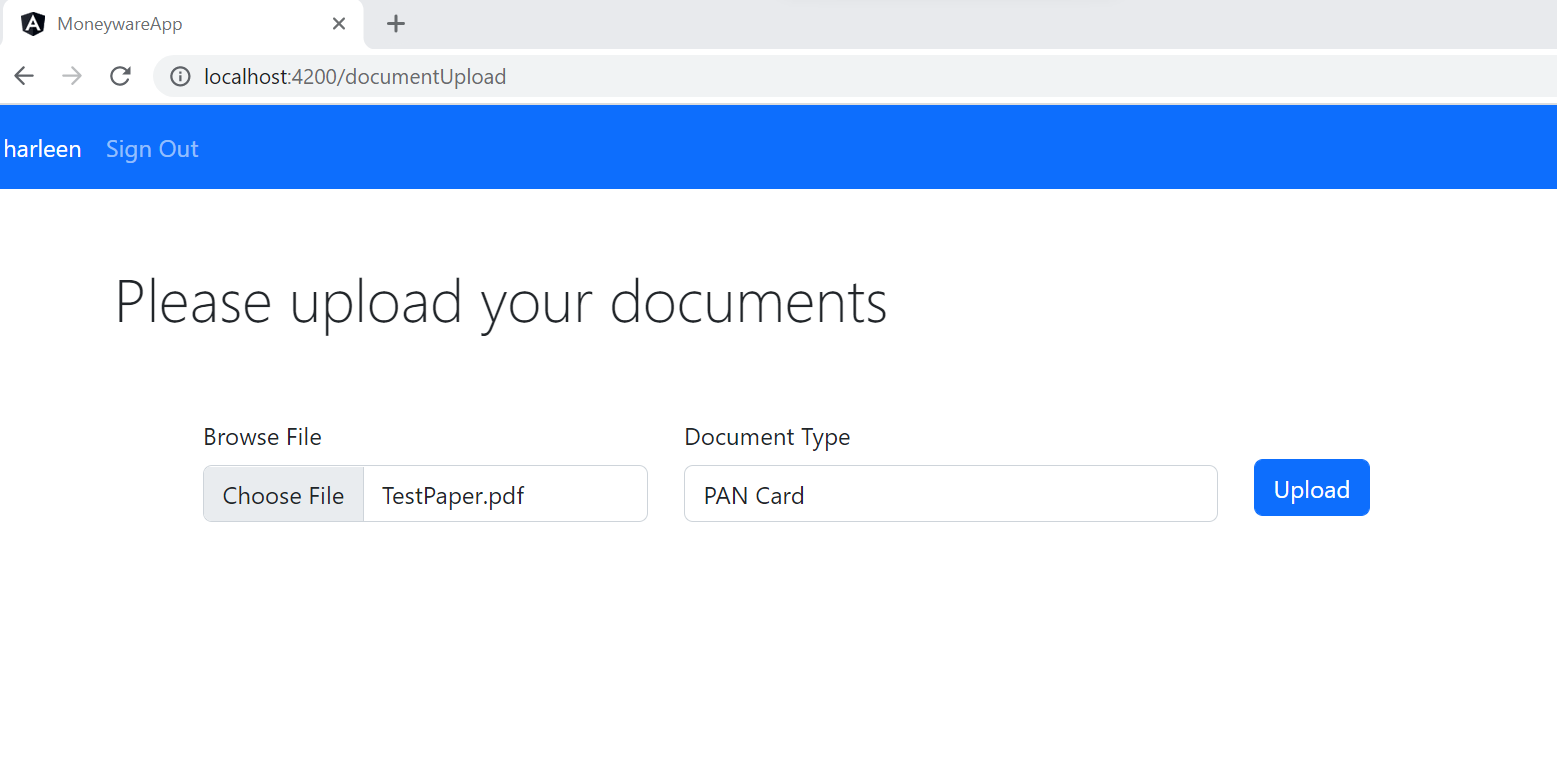
1. Login Page



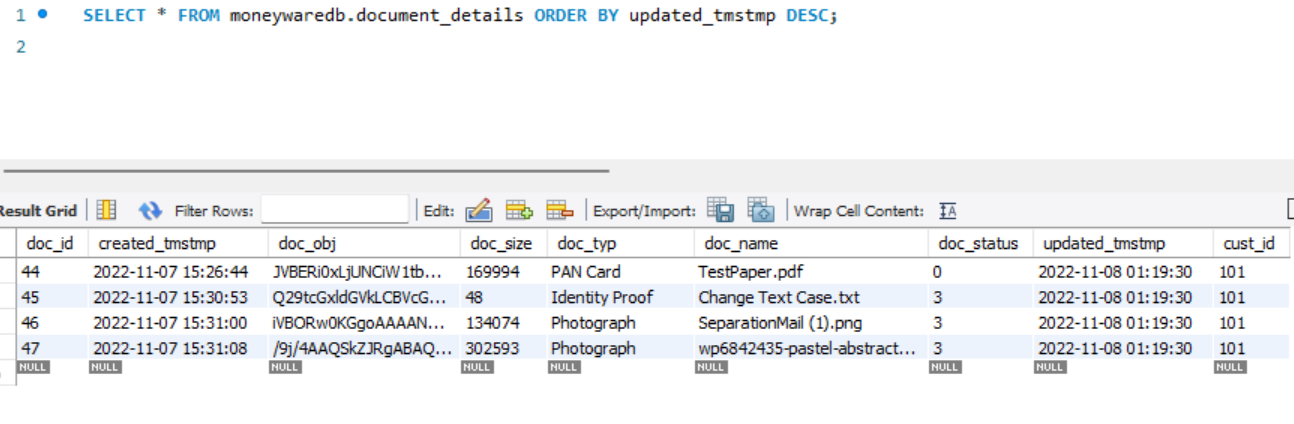
1. User\_details table



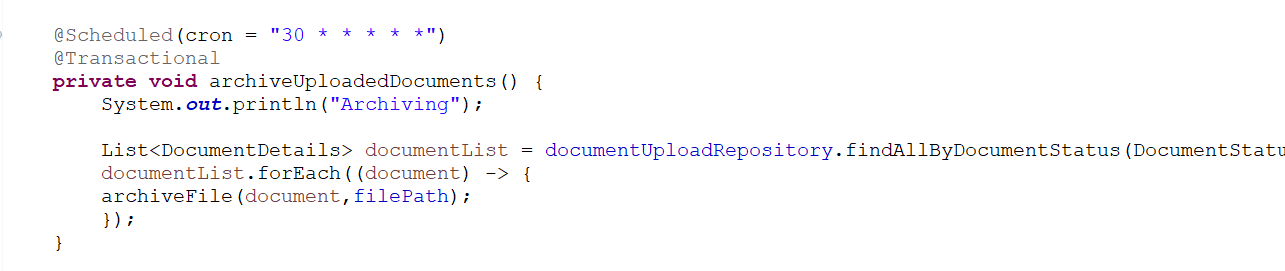
1. Document Upload Page



1. Document\_details table

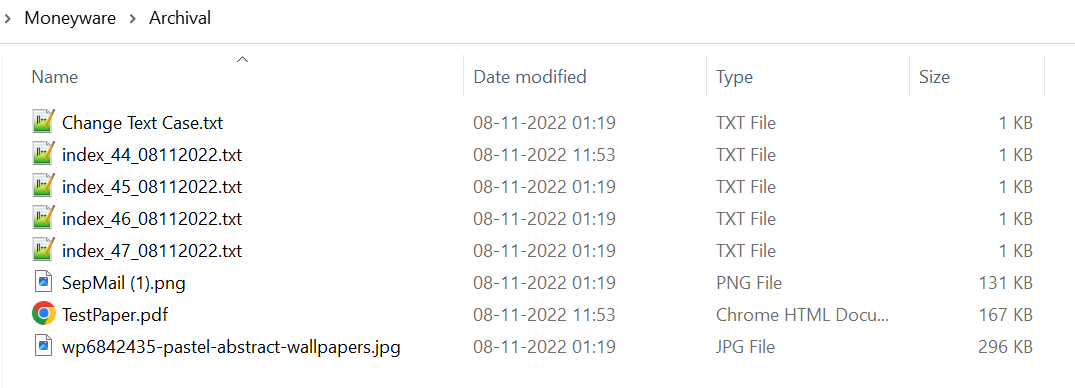


1. Cron Scheduler



Currently set to 30 seconds for development environment.

1. Archived Files along with Index Files



1. Index File (doc\_id + timestamp)

