

E-Mail applications manager

**Requirements**

# Purpose & scope

The main aim is to deliver a system, capable to facilitate tracking, monitoring and processing customer loan applications coming to the bank via e-mail.

Design and implement the application using ASP.NET Core 2.2 MVC.

The application should have:

* public part (accessible without authentication)
* private part (available for registered users)
* administrative part (available for administrators only)

# Simplified application proce

# Functional requirements

## User authentication

## User must be able to login in the system but could not register. An admin must first create the user and set him an initial password, which the user must change upon first login.

## User Rights & Roles

### Role “Operator” – contain only a single right – “Operator”

### Role “Manager” – contain only a single right – “Manager”

For the purpose of the project we can consider roles and rights as one and the same thing.

## Loan Application format

## Currently there is a well-defined format of a customer’s loan applications send by merchants’ trough existing common mailbox. Unfortunately, application format differ depends on a merchant in a large scale and it is usually manually fulfilled within the email body by the remote operators. Considering the above, any standardization or automatic parsing of the application shall not be performed within the current scope.

## Email – format

## No specific email format was defined so far – mail can be in any encoding, HTML or pure text, with or without attachments.

## Email / Application

## Each email may contain a single loan application. There may be cases where specific email does not contain any application, in such a case a field indicating such a mail shall be considered.

## Email registration

## As soon as mail is received it should be recorded in the system DB.

## All incoming emails shall be registered with unique ID automatically within the system.

## Email - application statuses

## **Not reviewed** - default status – all new emails shall be classified as such.

## **Invalid application** – all email which does not contain loan applications will be marked by operator in such status. Only employees with “Manager” rights can return the email in “Not reviewed” status.

## **New** – All emails shall be reviewed and only the one contains valid loan application will be set by employee with “New” status.

## **Open** – status marked for under processing – in order to set to “Open” status, the system should request Customer ID/EGN and customer phone information to be fulfilled by the bank employee. Respective formal system control of the provided data shall be applied.

## **Closed**

## Statuses flow

## Possible statuses transition depends on granted user/operator rights. They should be as follow:

## Operators (users/employees with Operator right)

## Not reviewed -> Invalid Application

## Not reviewed -> New

## New -> Open

## Open -> Closed (allowed only if the application was set in Open status by the same operator)

## Managers (users/employees with Manager right)

## Not reviewed <-> Invalid Application

## Not reviewed <-> New

## New -> Open (regardless who put the application in Open status)

## Open -> New (regardless who put the application in Open status)

## Open -> Closed (regardless who put the application in Open status)

## Closed -> New

## Email/Application record

## Content of emails under “Not Reviewed” status may be omitted and not stored within the system DB.

## As soon as email is set with status New – the body of the email shall be recorded in DB as well.

## Minimal set of application record

## Sender (email and name / From: field)

## Date received (date of email receiving within the mail server)

## Subject

## Body (HTML body is less than 40 bytes, text body should be shown or and persisted in DB)

## Count of attachments and total (sum) size of attachments (in kb/Mb)

## Customer unique ID / EGN – filled by bank employee / agent when change from New->Open status

## Customer contact phone – filled by bank employee / agent when change from New->Open status

## Date/Time of initial registration within the system (when the email was extracted from MS Exchange server and registered within the DB)

## Date/Time when the application was set in current status

## Date/Time when the application was set in terminal status (Closed or Invalid Application)

## Personal Data subject of GDPR

A following fields may contain customer private data therefore shall be encrypted within the DB:

#### Sender

#### Body

#### Customer unique ID / EGN

## Working screens

### Email preview

#### It should preview the contents of the Email.

#### Minimal set of attachments of data required

##### All attachments shall be listed – (file name, file size). No download or preview of attachment is needed.

##### Mail body preview

### List all emails

#### The screen should list all incoming emails

#### Minimal set of information for any email:

##### DB ID

##### Email DateTime received

##### Sender

##### Subject

##### Status

##### Icon indicating if there any attachments

##### List new emails

##### Email preview button – showing email preview onClick

#### Actions

##### All mails: option to mark specific email as not valid

##### Mails marked as “not valid”: option to remove the flag “not valid” (only for Managers role)

#### Possible filters: show only emails marked as not valid

#### Specific behavior

##### Mails marked as not valid shall be grayed out

##### Mail shall list shall be obtained from mail server in every page load but only if the last load was more than 1 minute(configurable) ago;

### List New applications/emails

#### The screen should list “verified” emails waiting for processing

#### Minimal set of information for any email:

##### DB ID

##### Email DateTime received

##### Sender

##### Subject

##### Time since is in current status (order by this field desc)

##### Email preview button – showing email preview onClick

#### Actions

##### Option to assign application for work (change status to Open)

##### New fields for fulfillment - Customer ID/EGN and customer phone information

##### In order assign application for work, Customer ID/EGN and customer phone shall be fulfilled and formally validated.

### List open applications/emails

#### The screen should list only emails/application with Open status

#### Minimal set of information for any email:

##### DB ID

##### Email DateTime received

##### Sender

##### Subject

##### Time since is in current status (order by this field desc)

##### Email preview button – showing email preview onClick

##### User who put application in status “Open” status (only Manager role)

#### Actions

##### Close application with Approve or Reject status

##### Return application status to “New” (only Manager role)

#### Specific behavior

##### “Operator” role – should filter applications only opened by currently logged user

##### If logged user has Manager role – all application in Open status

##### Possible filter by User who put application in “Open” status – only if logged user has Manager role

### List closed applications screen

#### The screen should list only emails/application with Closed status

#### Minimal set of information for any email:

##### DB ID

##### Email DateTime received

##### Sender

##### Subject

##### Status and clear indication if the application was “approved” or “rejected”

##### Email preview button – showing email preview onClick

##### User who put application in status “closed” status (only Manager role)

#### Actions

##### Return application status to “New” (only Manager role)

#### Specific behavior

##### “Operator” role – should filter applications only put in status “closed” by currently logged user

##### If logged user has Manager role – all applications “closed” status shall be shown

##### Possible filter by User who put application in “closed” status – only if logged user has Manager role

##### Possible filter for approved or rejected

## Audit log

All actions (status changes) shall be logged with minimum information set

### TimeStamp

### Action performed (status change, etc.)

### User performed the action

### Last status

### New status

# Non-functional requirements

## Architecture

MVC APP

## Logging & operational

### Each component shall support centralized logging capability (preferred in Azure Log Analytics)

### Operational status reporting supporting centralized health monitoring

## Technology Reference model

### Client - Cross browser - Web based application supporting

#### Firefox v. 50 or newer

#### Chrome 70 or newer

#### IE 10 or newer

#### Edge

### HTML5, CSS3, JavaScript

### Client (JavaScript) framework may be used (single one is preferred), without CDN. Mixture of multiple (more than one or two) client-side frameworks is not recommended.

### Client Working Resolution optimized for:

#### 1920px x 1080px @ 24 bit color

### Database – PostgreSQL - Azure Database for PostgreSQL preferred (or running @Azure Container Instances)

### Backend

#### .Net or .Net Core

#### Entity Framework (EF) - Recommended

#### Language - C# recommended

#### Capability to run as Azure Web App service or alternatively hosted on IIS

### Logging – usage of standard logging library (log4net preferred)

## Security requirements

### Secure data communication between all “layers” (for HTTPs –TLS 1.2)

#### Client <-> backend

#### Backend <-> database

#### Intra modules communication (if any)

#### Backend <-> Mail server

### Transparent data encryption (protecting data at rest) or column level data encryption (preferred) for confidential customer data.

### The system must be resistant to attacks like cross-site scripting, SQL injection, buffer overflow, session hijacking etc. Protection from attacks including (but not limited):

##### Input validation performed on server and client side (adhere to validation rules like length arrays, content of text arrays, mandatory arrays etc.);

##### HTTP session management prevents session hijacking (e.g. change of session ID after logon, regular session changes, additional security mechanisms as tickets, etc.);

##### Consistent use of "escape sequences" for non-standard character outputs;

##### Consistent use of “Prepared Statement” for database queries and restriction of SQL commands from input data;

##### Error messages displayed to user, customer or third parties do not provide sensitive information on data or system (e.g. for "wrong password", do not display stack traces, only HTTP 200 and 300 as response);

## Performance and SLA

### Maximum concurrent users – 30

### Maximal click to action response – 0.5s

### Maximal click to result action – 1.0 s.