



Wednesday, May 17, 2023

# Individual Requests System (MIRS)

Software Requirement Specification

## Table of Contents

Overview.....	2
Introduction.....	2
Card filter attributes.....	2
Screens .....	3
Login .....	3
Main .....	3
Jobs.....	4
Job .....	4
System Configuration .....	5
Users.....	5
Dashboard.....	6
Account.....	6
Audit Log.....	7
Other functional requirements.....	7
Assumptions .....	7
Technical requirements and guidelines .....	8
Logging .....	8
Contract .....	9

## Overview

The purpose of this document is to describe the high-level specifications of the Individual Requests System (MIRS) which is a simple web-based solution that enables the operation team in a cards perso center to list all the received card requests, filter and select requests they want to process, and export them to text-based files. The solution should also provide a simple dashboard that summarizes the daily and weekly activities.

## Introduction

The main purpose for developing MIRS is to allow the operation team to export the production data stored in a database to a text-based file. The database is regularly populated with data using an external service.

Each export operation is recorded in the system as a job. So each job represents a set of card requests that were exported to a production file.

In order for a user to export data, he needs first to identify the attributes of the cards to be exported based on a specific set of filters.

## Card filter attributes

### Account Number

- Free Text

### Date

- Date Range

### Type

- Predefined list (Card.EmbossingFileId-> EmbossingFile. TypeLookupId-> Lookup. LookupId)

### Maintenance

- Predefined list (Card.MaintenanceLookupId -> Lookup. LookupId)
- Multi-items can be selected

### Site

- Mapped list (Card.SiteId -> SiteMapping. SiteId -> SiteMapping. ZoneLookupId -> Lookup. LookupId)
- Multi-items can be selected

### Product Code

- Predefined list (Card. ProductCodeId-> ProductCode. ProductCodeId)
- Multi-items can be selected

### Issue group code

- Predefined list (Card. IssueCodeId-> IssueCode. IssueCodeId)
- Multi-items can be selected


Image Id  
Free text

## Screens

### Login

Instant Issuance Individual Requests

http://



Username

Password

☐ Remember Me [Forgot My password](#)

Clicking on "Forgot My password" will send an email to the admin email which is stored in the systemConfiguration table

**Login**

### Main

Instant Issuance Individual Requests

http://

**Sidebar Menu**

- Pending Files
- Jobs
- Administration
  - o Users
  - o Configuration
  - o Audit Log
- Analytics
  - o Dashboard
  - o Account
  - o Logout

Account Number

Type

From  To  Today

Image Id

Site

Maintenance

Product Code

Issue Group Code

**Refresh**

Date	Account	Name	Site	Site	Maintena	Product C	Issue Group	Filename
<input checked="" type="checkbox"/> 1-1-2023 12	123654	Osama Sh	1001	Baghc	New	oc 285	ic 12	R06_CL0327_IQ_EmbossingFile_0010_20250804_
<input type="checkbox"/> 1-1-2023 12	123654	Ali Shibly	1001	Baghc	New	oc 285	ic 12	R06_CL0327_IQ_EmbossingFile_0010_20250804_
<input checked="" type="checkbox"/> 1-1-2023 12	123654	Mazen Fou	1001	Baghc	Replace	oc 285	ic 12	R06_CL0327_IQ_EmbossingFile_0010_20250804_
<input type="checkbox"/> 1-1-2023 12	123654	Ali Shibly	1001	Baghc	Renew	oc 285	ic 12	R06_CL0327_IQ_EmbossingFile_0010_20250804_

**Create Job and Download Embossing File**

Sidebar menu is auto collapse

- Upon clicking Create job button, a confirmation message should be popup



## System Configuration

System Configuraiton

← → × ↗

http://

Q

Home -> System Configuraiton

Header

Code

Name

Value

Note

Add

Edit

Delete

Set Password

Q search

Code	Name	Value
1228	setting 1	1544
1225	Server host	http://asasdada.com

## Users

Users

← → × ↗

http://

Q

Home -> Users

Header

Full name

username

☒ Active

Role

Operator

Admin

Email

Note

Add

Edit

Delete

Set Password

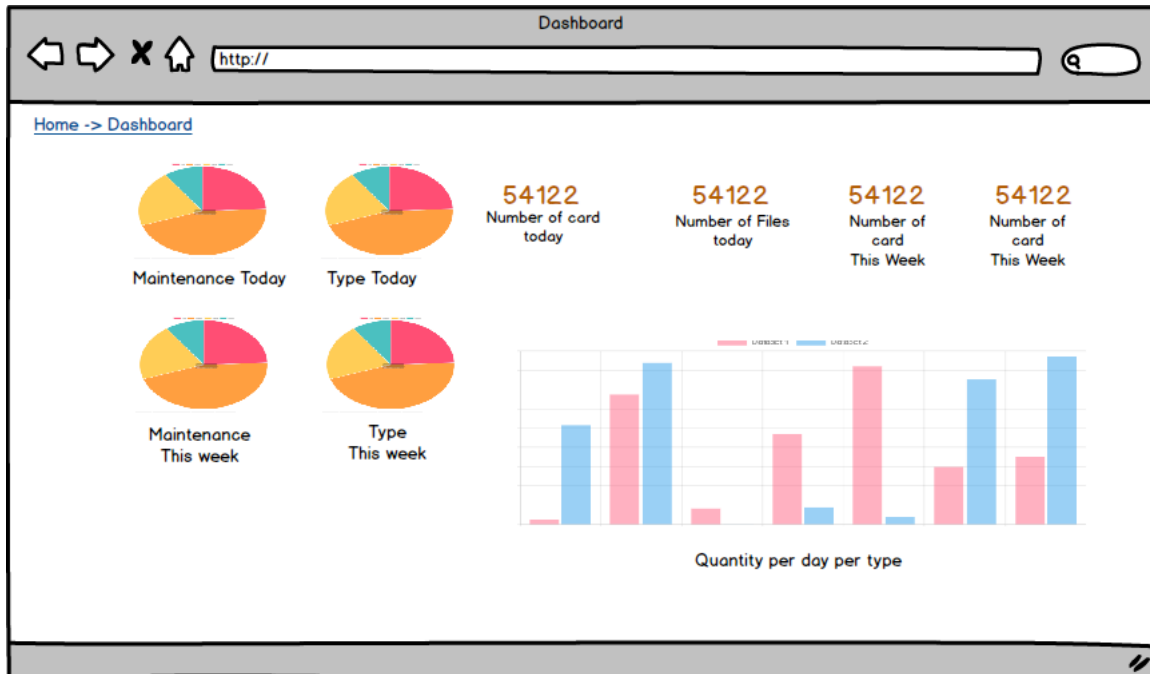
Details

Q search

Username	full name	email	Role	Active
oshaat	osama shaat	oshaat@gmail.com	Operator	Active
oshaat1	osama1 shaat	oshaat1@gmail.com	Admin	Not Active

- Username is mandatory and unique
- Fullname is mandatory
- Email format validation should be applied
- Role is mandatory
- Set password will open a popup to enter the password and password confirmation

## Dashboard



## Account

The Account UI features a browser-like header with navigation icons and a search bar. Below the header, a breadcrumb trail reads "Home -> Account". The main content area contains a registration form with the following fields:

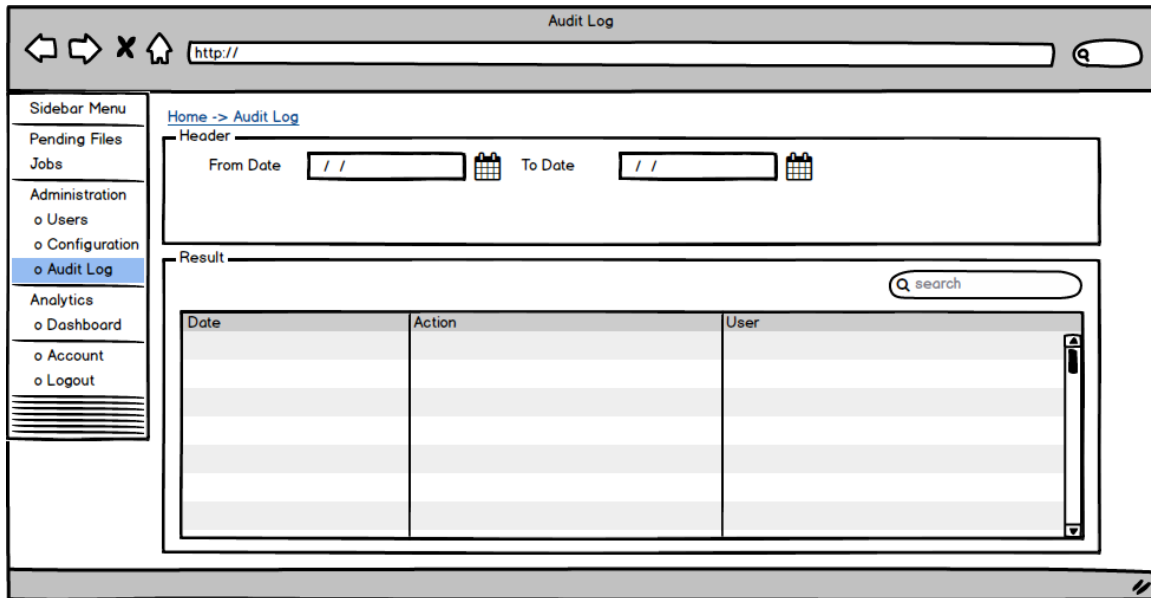
- Username:** A text input field that is disabled (indicated by a light gray background).
- Fullname:** A text input field.
- Email:** A text input field.

On the right side of the form, there are two buttons: "Add" and "Change Password". A yellow sticky note is placed over the form, containing the following text:

Username is always disabled and can't be changed. Change password will open a popup to enter the password and password confirmation.

- Username is always read-only
- Fullname is mandatory
- Email format validation should be applied

## Audit Log



The wireframe illustrates the 'Audit Log' web application. It features a top navigation bar with a title 'Audit Log' and a search input field. A left sidebar menu contains categories like 'Pending Files', 'Jobs', 'Administration' (with sub-items 'Users', 'Configuration', and 'Audit Log'), and 'Analytics' (with sub-items 'Dashboard', 'Account', and 'Logout'). The main content area is titled 'Home -> Audit Log' and includes a 'Header' section with 'From Date' and 'To Date' filters, each with a date picker icon. Below this is a 'Result' section containing a table with columns 'Date', 'Action', and 'User'. A search bar is positioned above the table. The table has a vertical scrollbar on the right side.

Date	Action	User

## Other functional requirements

- There are two roles supported in the system:
  - Administrator: has access to all functionalities
  - Operator: has access to all functionalities except the user management

## Assumptions

- The shared screenshots in this document should be considered as general guidelines of the layout (wireframes) and should not be treated as a detailed design.



## Technical requirements and guidelines

- The application is requested be developed as ASP.Net Core MVC 7.0 using C#
- The frontend should be responsive
- The frontend should be developed using a bootstrap admin template
- The database is SqlServer 15+ or higher
- All data access logic should be implemented using stored procedures
- Inline SQL scripts and SQL concatenation must not be used anywhere in the project
- In case there was a need for an array of values to be passed to a stored procedure, Table-Valued Parameters should be used, not a concatenated string value.
- Dapper is the ORM that must be used to connect to and interact with the database. EF should not be used.
- All data grid should support a search box to allow the user to filter the data quickly
- All data grid should support exporting the data to excel
- It is highly recommended to use datatables.net
- Db connection string should be in a configuration file and not hardcoded in the code
- Password should be strongly hashed and shouldn't be stored clear in the database
- Code should be modular, and no redundant code blocks should be detected.
- Constants should be used wherever possible
- The Database structure (Schema, metadata, objects' names...etc) must not be altered/modified without prior approval from our side
- The Server memory should be treated carefully (example: minimize the usage of global variables or session variables)
- Database connection should be opened only when needed and closed immediately after query execution completion and there is no need for it anymore.
- Views should have breadcrumbs
- The charts of the dashboard are recommended to be developed using ChartJs
- The dropdown lists should be searchable and support multiple selection

## Logging

- All the user actions and events –especially the delete operations- should have error handling with logging
- Logs should be stored in the database and text files
- Text files logs should be rolled out on a daily basis, where each line is tab separated, and contains the date & time of the event, username, and the log message

## Contract

- The payment will be released after 2 working days of delivering the final projects deliverables (like source code, installation files, documentations and other artifacts) and should run successfully on a local development server of the client.
- The source code is an exclusive intellectual property of the client
- Upon the submitting the deliverables:
  - Share a document that explains any workarounds or hacks employed to develop or run the solution
  - Send a sheet of all: added external third party libraries, components, reference
- Only free and open source libraries can be used. Paid, or trial commercial libraries must not be used.