Medios Connect

Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| ver | date | author | description |
| 0.1 | 13.10.2015. | Marija Vucetic | First draft of the template (Sections: 1 – 4) |
| 0.2 | 18.11.2015. | Nemanja Zigic | First draft of the template (Sections: 5 - 6) |
| 1.1 | 03.05.2017 | Goran Ugrincic | Updates after OCT functionality |

Contents

[1 Installation guide 5](#_Toc482620003)

[1.1 Create account via Example App 5](#_Toc482620004)

[1.1.1 Prerequisites 5](#_Toc482620005)

[1.1.2 Web.config file 5](#_Toc482620006)

[1.1 Medios Connect Maintenance Service 7](#_Toc482620007)

[1.1.1 Installation 7](#_Toc482620008)

[1.1.2 Service Tasks 8](#_Toc482620009)

[1.1.3 Installation service window 8](#_Toc482620010)

[1.1.4 Order summary 8](#_Toc482620011)

[2 Dataimporter 9](#_Toc482620012)

[2.1 Initial setup for new tenant 9](#_Toc482620013)

[2.2 Data import from excel files 9](#_Toc482620014)

[2.3 Data rebuild on the elastic server 10](#_Toc482620015)

[2.4 Data corrections in the relational database 10](#_Toc482620016)

[2.4.1 Clean up cases on the elastic server – not to be used 10](#_Toc482620017)

[2.4.2 Update doctors and patients – add contract assignment – not to be used 10](#_Toc482620018)

[2.4.3 Recalculate all GPOS-es on tenant – not to be used 10](#_Toc482620019)

[2.4.4 Delete last used aftercares 10](#_Toc482620020)

[2.4.5 Fix doctors db table – not to be used 10](#_Toc482620021)

[2.4.6 Fix gpos in the db – not to be used 10](#_Toc482620022)

[2.4.7 Fix aftercare FS status – not to be used 11](#_Toc482620023)

[2.4.8 Add company settings for tenant 11](#_Toc482620024)

[2.4.9 Fix open aftercares bill positions – not to be used 11](#_Toc482620025)

[2.4.10 Fix double aftercares on old cases 11](#_Toc482620026)

[2.4.11 Fix contract roles – not to be used 11](#_Toc482620027)

[2.4.12 Fix multiple patient participation consents 11](#_Toc482620028)

[2.4.13 Fix multiple aftercare FS statuses 11](#_Toc482620029)

[2.4.14 Fix multiple doctor consents 11](#_Toc482620030)

[2.4.15 Fix missing GPOS-es 11](#_Toc482620031)

[2.4.16 Update bill position numbers 11](#_Toc482620032)

[2.4.17 Fix bill positions for open aftercares 11](#_Toc482620033)

[2.4.18 Fix cases that are twice in the report 11](#_Toc482620034)

[2.4.19 Fix aftercares which are performed/paid twice 11](#_Toc482620035)

[2.4.20 Fix aftercares FS statuses deleted by mistake 12](#_Toc482620036)

[2.4.21 Fix missing aftercare GPOS-es caused by new contract 12](#_Toc482620037)

[2.4.22 Set all cases with case number below 10000 to have had their submit.pdf report downloaded 12](#_Toc482620038)

[2.4.23 Fix Invoice to practice case property 12](#_Toc482620039)

[2.4.24 Fix cases submitted without patient participation consent 12](#_Toc482620040)

[2.4.25 Convert pre-examinations into OCTs 12](#_Toc482620041)

[2.4.26 Fix double oct planned action type 12](#_Toc482620042)

[2.4.27 Add missing open OCTs 12](#_Toc482620043)

[2.4.28 Fix relevant planned action creation timestamps 12](#_Toc482620044)

[2.4.29 Fix patient insurances creation timestamps 12](#_Toc482620045)

[2.4.30 Mark all cancelled orders as exported 12](#_Toc482620046)

[2.4.31 Fix FS8 cases wrongfully transferred to FS7 12](#_Toc482620047)

[2.4.32 Fix patients with HIP with no contract that have consents 12](#_Toc482620048)

[2.4.33 Fix documentation cases without GPOS 13](#_Toc482620049)

[2.4.34 Fix submitted cases that were cancelled 13](#_Toc482620050)

[2.4.35 Add default pharmacy to old orders 13](#_Toc482620051)

[2.4.36 Fix missing billing information for reassigned aftercares 13](#_Toc482620052)

[2.4.37 Fix open aftercares (cases where both error correction and open aftercare exist) 13](#_Toc482620053)

[2.4.38 Check if there are cases with multiple billed aftercares without aftercare cancellation 13](#_Toc482620054)

[2.4.39 Fix performed OCTs without FS status 13](#_Toc482620055)

[2.4.40 Fix missing open aftercares 13](#_Toc482620056)

[2.4.41 Fix missing gpos for open OCTs 13](#_Toc482620057)

[2.4.42 Remove OCTs attached to cases with missing Ozurdex GPOS 13](#_Toc482620058)

[2.5 Set password to all users 13](#_Toc482620059)

[3 Process 13](#_Toc482620060)

[4 Database 18](#_Toc482620061)

[4.1 Practice data 18](#_Toc482620062)

[4.2 Doctor data 19](#_Toc482620063)

[4.2.1 Contract data 19](#_Toc482620064)

[4.2.2 GPOS data 19](#_Toc482620065)

[4.2.3 Drugs/Diagnoses data 20](#_Toc482620066)

[4.2.4 Add doctor to contract 20](#_Toc482620067)

[4.2.5 Add GPOS to contract 20](#_Toc482620068)

[4.3 Users 20](#_Toc482620069)

[4.4 Archive 21](#_Toc482620070)

[4.5 Case 21](#_Toc482620071)

[4.6 Submit case 23](#_Toc482620072)

[5 Special processes: 23](#_Toc482620073)

[5.1 DOC APP 23](#_Toc482620074)

[5.1.1 New case: 23](#_Toc482620075)

[5.1.2 Submit case: 24](#_Toc482620076)

[5.1.3 Cancel case: 25](#_Toc482620077)

[5.1.4 Multi edit/submit: 25](#_Toc482620078)

[5.1.5 Patients manipulation: 25](#_Toc482620079)

[5.1.6 Notifications and expired session: 26](#_Toc482620080)

[5.1.7 Submit order 26](#_Toc482620081)

[5.2 MM APP 26](#_Toc482620082)

[5.2.1 Dashboard actions: 26](#_Toc482620083)

[5.2.2 Manual case status/management fee status change: 27](#_Toc482620084)

[5.2.3 Drug order rejection: 27](#_Toc482620085)

[5.2.4 New practice/doctor: 27](#_Toc482620086)

[5.2.5 Temporary doctor: 27](#_Toc482620087)

[5.2.6 Doctor contract consents: 28](#_Toc482620088)

[5.2.7 Contracts: 28](#_Toc482620089)

[5.2.8 Notifications and expired session: 28](#_Toc482620090)

# Installation guide

Framework:

* Target Framework: Microsoft.NET Framework 4.5
* Target Databases: MySQL and Elastic

## Create account via Example App

To be able to MMDocConnection applications first you need to create a master account. For this, at this moment, we are using the Example App.

To register a new account, click on the “Register Tenant” button.



Then just fill the data and click “Register Tenant”.

### Prerequisites

* Installed Elastic
* Installed MySQL DB
* Created account via Example app

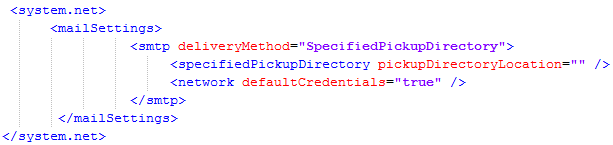
### Web.config file

#### App Settings

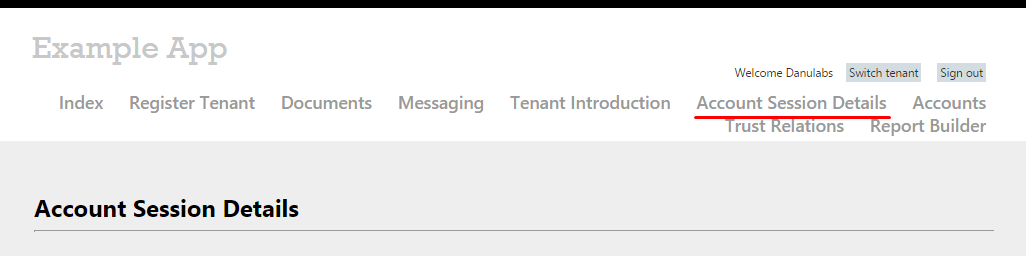


* **ElasticConnection** - specifies the elastic URL
* **ElasticPort** - specifies the elastic port
* **mailInfo -** specifies the mail where messages for orders are sent
* **mailFrom** - specifies the mail from which the message is sent
* **activeConnection -** specifies the MySQL Connection (specify the connection in the <connectionStrings> </connectionStrings> section above the <appSettings>)
* **mediosmanagementFullAddress** – specifies Medios Connect address from PDF report
* **mediosmanagement –** specifies Company name in PDF report (Medios Connect GmbH)
* **mediosmanagementAddress** – specifies short format of address in PDF Report
* **mediosmanagementTelefon** – specifies Company phone number in PDF Report
* **mediosmanagementFax** - specifies Company Fax in PDF Report
* **mediosmanagementEmail** – specifies Company Email in PDF Report
* **mediosmanagementWeb** – specifies Company web address in PDF Report
* **mediosmanagementCEO** – specifies Company CEO in PDF Report
* **mediosmanagementHRB –** specifies Company HRB in PDF Report
* **mediosmanagementAddress2 –** specifies second Company address in PDF Report
* **mediosmanagementBank –** specifies Company Bank name in PDF Report
* **mediosmanagementIBAN –** specifies Company IBAN in PDF Report
* **mediosmanagementBIC –** specifies Company BIC Code in PDF Report
* **mediosmanagementUST –** specifies Company UST in PDF Report
* **bop.config.provider.tms –** specifies URL to tms service on Linux server
* **bop.config.provider.doc –** specifies URL to doc service on Linux server
* **bop.config.provider.rms –** specifies URL to rms service on Linux server
* **bop.config.provider.rps –** specifies URL to rps service on Linux server
* **bop.config.provider.trust –** specifies URL to trust service on Linux server
* **mmAppUrl –** specifies default name of MM application
* **docAppUrl -**  specifies default name of DOC application
* **notificationTimeoutInMiliseconds –** specifies period for notification update in miliseconds
* **mailInfo -**  specifies default email for information
* **mailFrom -**  specifies default email for notifications
* **edifact-generate-test-data –** specifies whether edifact be generated using test data
* **generateReceipPdf –** specifies does pdf report should be generated after change status of cases
* **test-hip-for-case -**  specifies ID of case for generate report
* **test-hip-for-case-file-path –** specifies path to file where all analysis from complete report will be saved
* **gracePeriodDuration –** specifies duration of grace period
* **autoRenewalWithQuickOrder –** specifies does quick order auto renewal grace period

#### Mail setting



* **pickupDirectoryLocation** - Mail pickup directory (for example: “C:\inetpub\mailroot\Pickup”)



## Medios Connect Maintenance Service

Before installing the service, we need to setup the web.config.

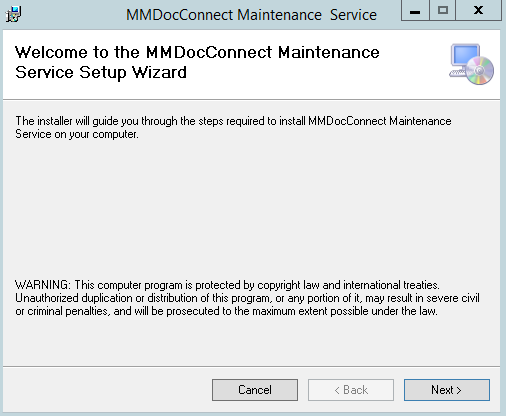
1. In the <connectionStrings> tag add the connection string to your relation data base.

2. In the <appSettings> tag there are several “keys” that have to be filled:

1. mmdocconnect.elastic.connection - specifies the elastic URL
2. mmdocconnect.elastic.port - specifies the elastic port
3. mmdocconnect.recipient.mail - specifies the mail which will receive daily status report, this is just a fallback mail if the company setting in the MM app have not been defined
4. mmdocconnect.send.mail - specifies the mail from which the message is sent
5. mmdocconnect.order.notification.interval - specifies order notification interval (how often mail is sent) just a fallback setting if it is not specified in the MM app
6. mmdocconnect.order.notification.login.url.mmapp - specifies the URL to the MM app (example: https://lynxv3.b-op.com/mmdocconnect-mm/)
7. mmdocconnect.tenantid – specifies the Tenant ID of the application
8. activeConnection – specifies the connection string name you have specified in the <connectionStrings> tag

### Installation

Click on the setup.exe file. A new window will appear. Just click next and the service will be installed.

****

***Installation service window***

When installed the service is Running by default.

### Service Tasks

Maintenance service is responsible for couple of different tasks:

#### Order summary

A summary of all new orders is sent to the MM employee in a specified cycle.

* Service runs every x minutes and generates a summary email
* X = “Order Notification Interval - minutes” setting
* Whenever this setting is changed in the settings page, the service trigger is changed
* Email is sent to “Standard-Benachrichtigungsadresse” setting
* If this setting does not exist, use the one from a config files

### Installation service window

When installed the service is Running by default.

Maintenance service is responsible for couple of different tasks:

### Order summary

A summary of all new orders is sent to the MM employee in a specified cycle.

* Service runs every x minutes and generates a summary email
* X = “Order Notification Interval - minutes” setting
* Whenever this setting is changed in the settings page, the service trigger is changed
* Email is sent to “Standard-Benachrichtigungsadresse” setting
* If this setting does not exist, use the one from a config files

#### Set up status to overdue

* A service runs every day shortly after midnight and sets certain treatments, aftercares and orders to overdue.

#### Settlement timer

A MM employee gets an email notification when the maximum time to get HIP feedback has passed and they haven’t sent a response, so that he can contact them to see what’s the holdup.

* Service triggers the email sending every day around 8 am when people come in for work
* The email should arrive to the standard notification email for mm
* The service should go through all active contracts, pick up the “Max time for HIP to send feedback for fee settlement” if available
* Checks how many days have passed since each of the cases currently in status FS2 have been put into that status
* If the number is Max time or higher, send an email with temporary content: “Dear MM Team, xx days have passed since you submitted some of your settlement cases, but no response was imported. Maybe you should check with the health insurance from yyyyy contract.”

# Dataimporter

Dataimporter is console app with Program.cs and Main method. This app is used for creating help methods and methods used for maintenance services, such as all rebuild methods. Main method is divided into 4 semantic sections:

1. Initial setup for new tenant.
2. Data import from excel files.
3. Data rebuild on the elastic server
4. Data corrections in the relational database

## Initial setup for new tenant

By running this command, the user creates initial data necessary for docApp/mmApp work. IVI-Vertrag and mm House Contract are created, along with the 9 drugs and 3 diagnoses, connected to IVI-Vertrag, including GPOS-es to which these drugs and diagnoses are bound.

## Data import from excel files

By running this command, the user is redirected to new menu which contains 5 new commands:

1. Practices, doctors and patients import. (excel file names: doctors.xlsx, practices.xlsx, patients.xlsx)
2. IBAN/BIC codes import. (excel file name: cases.xlsx)
3. Cases import. (excel file name: cases.xlsx)
4. Diagnoses import. (excel file name: ICD10.xlsx)
5. HIPs import. (excel file name: HIPs.xlsx)

The corresponding files (written next to command names) must be in the Excel folder in the root of the application.

By running any of the commands, appropriate excel file is loaded and then validated. After validation, a new excel file is generated. If the validations fail, no data will be imported. If the validations are passed, the data is imported into the relational database and elastic server.

## Data rebuild on the elastic server

By running this command, the user is redirected to new menu which contains 10 new commands. Each of these commands rebuilds certain section of the elastic server by using the data from the relational db. If there are any errors or if no data is retrieved from the relational database, the data on the elastic server remains as it was before rebuilding. If the data is retrieved from the relational database and no errors occur, data section on the elastic server is deleted and new data is imported, along with new mapping (if there have been any adjustments to the elastic mapping).

## Data corrections in the relational database

By running this command, the user is redirected to new menu which contains 43 new commands, all of which have been used to correct the data during the development:

### Clean up cases on the elastic server – not to be used

By running this command, all of the data related to cases on elastic server is deleted and replace with the data from the excel file (cases.xlsx in the Excel folder). Since there is an option to rebuild elastic server from the relational database, this command should never be used.

### Update doctors and patients – add contract assignment – not to be used

By running this command, all of the doctors and patients are updated with consents on IVI-Vertrag. Doctors are updated with one consent that starts 15.06.2013, while the patients are updated with 3 consents, (15.06.2013, 15.06.2014, 15.06.2015). Since these same consents are created during the data import from the excel files, this command should never be used.

### Recalculate all GPOS-es on tenant – not to be used

By running this command, all cases GPOS positions, belonging to IVI-Vertrag are recalculated. Since this command uses old GPOS calculation logic, this command should never be used.

### Delete last used aftercares

By running this command, all of the last used aftercare doctors on the elastic server will be deleted. These doctors appear in the autocomplete for assigning the aftercare doctor to a case (The first three in the list).

### Fix doctors db table – not to be used

By running this command, doctors which aren’t temporary have their Account\_RefID field in the HEC\_Doctors table updated with a reference to the actual USR\_Accounts account id. Since the doctors have this field updated during import, this command should never be used.

### Fix gpos in the db – not to be used

By running this command, the 36620062 gpos position has it’s diagnose reference fixed. Since this is fixed in the setup for now tenant command, this command should never be used.

### Fix aftercare FS status – not to be used

By running this command, aftercare bill positions which didn’t have an FS status assigned to them are updated with one. Since this command is using old GPOS calculation logic to do this, it should never be used.

### Add company settings for tenant

By running this command, default company settings are added to a new tenant. These settings include standard e-mail for sending notifications, immediate order interval and order interval.

### Fix open aftercares bill positions – not to be used

By running this command, non-submitted aftercares which don’t have bill positions assigned are updated. Since this command is using old GPOS calculation logic to do this, it should never be used.

### Fix double aftercares on old cases

By running this command, old cases (which had their treatments and aftercares billed together) have their double aftercare bill positions removed, since the aftercares were billed together with the treatments.

### Fix contract roles – not to be used

By running this command, contract roles in the relational database are fixed by deleting duplicates of the same role. Since this has been fixed in the setup for new tenant, this command should never be used.

### Fix multiple patient participation consents

By running this command, duplicated patient participation consents on the same contract are removed.

### Fix multiple aftercare FS statuses

By running this command, aftercare bill positions that have multiple active FS statuses are updated – duplicates are removed.

### Fix multiple doctor consents

By running this command, duplicated doctor participation consent on the same contract are removed.

### Fix missing GPOS-es

By running this command, all of the cases which are missing special GPOS-es (Wartenzeitenmanagement and Zusatzposition Bevacizumab) are updated with them. This command is tested only on IVI-Vertrag and there are no guaranties that it will work with multiple contracts.

### Update bill position numbers

By running this command, any bill position that has an FS status (has been at least submitted to MM) and doesn’t have a bill position number is assigned one.

### Fix bill positions for open aftercares

By running this command, all deleted bill positions and deleted bill codes for aftercares without them will be changed into non-deleted.

### Fix cases that are twice in the report

By running this command, bill codes and bill positions for all non-submitted cases are deleted.

### Fix aftercares which are performed/paid twice

By running this command, duplicated bill codes and bill positions will be deleted.

### Fix aftercares FS statuses deleted by mistake

By running this command, all accidentally deleted FS statuses will be changed into non-deleted

### Fix missing aftercare GPOS-es caused by new contract

By running this command, bill position and bill code will be created and performed if needed for all aftercare on new contract.

### Set all cases with case number below 10000 to have had their submit.pdf report downloaded

By running this command, case universal property will be created for all cases where global property is missing also all old settlements will be edited and flag “is\_report\_downloaded” will be set to true.

### Fix Invoice to practice case property

By running this command,

### Fix cases submitted without patient participation consent

By running this command, submitted treatments without patient participation consent will be deleted, created aftercare for that treatment will be deleted, elastic will be updated and GPOS will be calculated again.

### Convert pre-examinations into OCTs

By running this command, OCT will be created for each pre-examination and pre-examination will be deleted.

### Fix double oct planned action type

By running this command, doubled action types for planned OCTs will be deleted.

### Add missing open OCTs

By running this command, for all submitted and non-canceled cases without OCT, open OCT will be created.

### Fix relevant planned action creation timestamps

By running this command, relevant planned action creation timestamp for cases will be updated with planned action creation timestamp.

### Fix patient insurances creation timestamps

By running this command, all insurances created on 27.12.2015 will be changed, creation timestamp of first insurance will be set as current value.

### Mark all cancelled orders as exported

By running this command, cancelled orders will be marked as exported.

### Fix FS8 cases wrongfully transferred to FS7

By running this command, all cases which mistakenly transferred to FS8 status will be transferred to FS7 status.

### Fix patients with HIP with no contract that have consents

By running this command, all insurance to broker contract for patient with HIP with not contract that have consents will be deleted.

### Fix documentation cases without GPOS

By running this command, create bill position for all documented cases without bill position.

### Fix submitted cases that were cancelled

By running this command, all submitted treatments and aftercares for cancelled case will be deleted.

### Add default pharmacy to old orders

By running this command, for orders without pharmacy, default pharmacy will be set.

### Fix missing billing information for reassigned aftercares

By running this command, for all submitted aftercares without fs status, fs status will be created.

### Fix open aftercares (cases where both error correction and open aftercare exist)

By running this command, all open aftercares will be deleted if there is aftercare in error correction.

### Check if there are cases with multiple billed aftercares without aftercare cancellation

By running this command, user will get list of all multiple submitted aftercares.

### Fix performed OCTs without FS status

By running this command, new open OCT will be created for all cases with all submitted OCTs.

### Fix missing open aftercares

By running this command, new bill position status will be created and GPOS will be recalculated for all open aftercare without bill position status.

### Fix missing gpos for open OCTs

By running this command, new bill position status will be created and GPOS will be recalculated for all open OCTs without bill position status.

### Remove OCTs attached to cases with missing Ozurdex GPOS

By running this command, all OCTs attached to cases with missing Ozurdex GPOS will be deleted.

## Set password to all users

By running this command, password for all doctor accounts will be changed with password specified in web.config or written using console if password from web.config is empty.

# Process

Application structure

In example bellow is shown structure of MMApp, DocApp structure is the same.

* .nuget
* After adding new nuget package, change .csproj of project in manner:
* <Reference Include="PdfSharp.Charting">
* <HintPath>$(SolutionDir)\packages\PDFsharp-MigraDoc-GDI.1.32.4334.0\lib\net20\PdfSharp.Charting.dll</HintPath>
* </Reference>
* External Projects
* Internal Projects

1. BopSDK
2. CSV2Core
3. CSV2Core\_MySQL
4. DLCore\_DBCommons
5. DLCore\_TokenVerificationV3
6. Edifact API
7. LogUtils

* Libraries

1. ORMMethods
2. CL2\_Language

* Solution Items
* MMDocConnectDBMethods

Methods are divided into folders; each folder contains items from one scope and name of a scope. Every folder has two subfolders: Atomic and Complex, every subfolder has folders Manipulation and Retrieval. Manipulation folders has items who are performing Save/Edit operations, Retrieval folder contains items which are getting data from DB. Every Manipulation and Retrieval folder has subfolders \_Support which contains XML files that are used for generating methods via Danulabs code generator. After running code generator SQL folder has been generated and method cls + xml file name + .cs. DB methods Class Library is shared among MM app and Doc App

Translated static data is stored at Properties – Resources.de.resx (for German language) and Resources.resx (for English language).

1. Properties
2. References
3. Service References
4. Archive
5. Case
6. Doctor
7. ElasticRefresh
8. MainData
9. Medication
10. Order
11. Patient
12. Pharmacy
13. App.config
14. Packages.config

* MMDocConnectElasticSearchMethods

Elastic Methods are divided into folders, each folder contains items from one scope and name of a scope. Every folder has two subfolders: Manipulation and Retrieval. Manipulation folders contains data for Save/Edit data in these methods are stored elastic mappings. Retrieval folders contains methods for getting data from elastic db. Elastic Search methods Class Library is shared among MM app and Doc App

1. Properties
2. References
3. Archive
4. Case
5. Diagnose
6. Doctor
7. ElasticUtils
8. HIP
9. Models
10. Oct
11. Order
12. Patient
13. Receipt
14. Settlement
15. app.config
16. Packages.config

* MMDocConnectMMapp

MMapp is divided into folders by function items from one folder are responsible for. Content folder is divided into three folders css, fonts and images. css folder contains .css files used in MM app.

Controllers folder contains Web Api Controllers, every controller belongs to one scope.

Scripts folder contains all JavaScript files used for front end application. It is divided into subfolders: src, test and vendor. Src folder contains subfolders app, common and resource\_files. App folder is divided into folders, each folder contains items from one scope and name of a scope. Every folder with scope name has subfolders: controllers, directives, services and view. Common folder contains subfolders: controllers, directives, factories, services, view and .js file application-configuration. In Common folders are stored general items such as custom directives and reusable items which are used through entire application. Resource\_files contains .js files with static data on German and English language. Vendor folder contains imported .js files such as angular, jquery etc.

Main.js file is used for register .js files controllers, directives, services etc.

1. Properties
2. References
3. App\_Data
4. App\_Start
5. Content
6. Controllers
7. Models
8. scripts
9. Global.asax
10. Index.html
11. Login.html
12. Main.js
13. Packages.config
14. Web.Config

* MMDocConnectDocApp

DocApp is divided into folders by function items from one folder are responsible for. Content folder is divided into three folders css, fonts and images. css folder contains .css files used in Doc app. Controllers folder contains Web Api Controllers, every controller belongs to one scope. EmailTemplates folder contains all needed html templates for email notifications. Filters folder contains .cs file for authentication logged user.

Scripts folder contains all JavaScript files used for front end application. It is divided into subfolders: src and vendor. Src folder contains subfolders app, common and resource\_files. App folder is divided into folders, each folder contains items from one scope and name of a scope. Every folder with scope name has subfolders: controllers, directives, services and view. Common folder contains subfolders: controllers, directives, factories, services, view and .js file application-configuration. In Common folders are stored general items such as custom directives and reusable items which are used through entire application. Resource\_files contains .js files with static data on German and English language. Vendor folder contains imported .js files such as angular, jquery etc.

1. Properties
2. References
3. ServiceReferences
4. App\_data
5. App\_Start
6. Content
7. Controllers
8. EmailTemplates
9. Filters
10. Models
11. ReportContent
12. scripts
13. Global.asax
14. gulpfile.js
15. index.html
16. log4net.xml
17. login.html
18. package.json
19. packages.config
20. SessionHub.cs
21. Startup.cs
22. Web.config

* MMDocConnectAppInterfaces

Interfaces Class Library contains interfaces from one scope and name of a scope.

1. Properties
2. References
3. App.config
4. IArchiveDataServices.cs
5. IContractDataServices.cs
6. IDashboardDataServices.cs
7. IDoctorDataServices.cs
8. IMainData.cs
9. IMedicationDataServices.cs
10. IOrderDataServices.cs
11. IPatientDataServices.cs
12. IPharmacyDataServices.cs
13. ISettingsService.cs
14. ITreatmentDataService.cs

* MMDocConnectMMAppModels

MMapp Models Class Library contains models. Each model has name of a scope or part of a scope depends on items from Model.

* MMDocConnecMMAppServices

MMappServices Class Library contains items from one scope and name of a scope. These classes are used for communication between Web Api Controllers and DB – by calling a DB methods from DBMethods Class Library or calling a method from Elastic Search Methods Class Library

* MMDocConnectUtils

MMDocConnectUtils contains imported items which are used in both app this Class Library is shared between MMapp and Doc App

1. Properties
2. References
3. ExcelTemplates
   1. CaseForReportModel.cs
   2. GenerateReportCases.cs
   3. GenerateReportOrders.cs
   4. OrdersForReportModel.cs
4. AuthenticationMethods.cs
5. Documents.cs
6. EmailNotificationSenderUtil.cs
7. Enums.cs
8. ExcelUtils.cs
9. GenericUtils.cs
10. GlobalProperties.cs
11. IElasticMapper.cs
12. SessionSecurityTicket.cs
13. SignalRNotifier.cs
14. SynchronizedSequentialNumberGenerator.cs
15. SynchronizedSequentialNumberGeneratorCase.cs
16. TileInfo.cs
17. TileInfoModel.cs
18. TransactionalInformation.cs
19. UtilMethods.cs
20. ZipFilesUtils.cs

* MMDocConnectUtils

MMDocConnectUtils Class Library contains ElasticRebuild utils for rebuild all section on elastic server.

# Database

This section has list of all used tables along with fields. Every table has additional fields such as Tenant\_RefID, Primary Key, IsDeleted, Creation and ModificationTimeStamp.

## Practice data

* USR\_Accounts (BusinessParticipant\_RefID, DisplayName)
* USR\_Account\_FunctionLevelRights\_Group (Label, GlobalPropertyMatchingID, USR\_Account\_FunctionLevelRights\_GroupID,)
* USR\_Account\_FunctionLevelRights (FunctionLevelRights\_Group\_RefID, RightName: UI-IsSurgeryPractice)
* ORM\_USR\_Account\_2\_FunctionLevelRight (Account\_RefID, USR\_Account\_FunctionLevelRightID, AssignmentID)
* CMN\_BPT\_CTM\_Customers (Ext\_BusinessParticipant\_RefID, IsMedicalPractice, IfMedicalPractise\_HEC\_MedicalPractice\_RefID)
* CMN\_BPT\_CTM\_OrganizationalUnits(Customer\_RefID)
* CMN\_BPT\_BusinessParticipants (DisplayName: UI- PracticeName, IsCompany)
* CMN\_BPT\_CTM\_OrganizationalUnits(OrganizationalUnit\_SimpleName)
* CMN\_COM\_CompanyInfo (CompanyInfo\_EstablishmentNumber: UI- BSNR)
* CMN\_COM\_CompanyInfo\_Addresses (IsDefault = true; IsShipping = true; IsBilling = true; Address\_Description = “Standard address for billing, shipping”, IsDefault = true; IsContact = true; Address\_Description = “Standard contact person data”)
* CMN\_UniversalContactDetails (CompanyName\_Line1: UI- Practice name, IsCompany, Street\_Name: UI- Street, Street\_Number – UI- Street Number, ZIP: UI-ZIP, Town: UI-City, Contact\_Email: UI- Main Email, Email; Contact\_Telephone: UI-Main Phone, Phone; Contact\_Fax: UI- Fax; First\_Name: UI- Contact Person First Name; Last\_Name: UI- Contact Person Last Name)
* CMN\_BPT\_BusinessParticipant\_2\_BankAccount (CMN\_BPT\_BusinessParticipant\_RefID, ACC\_BNK\_BankAccount\_RefID)
* ACC\_BNK\_BankAccounts (OwnerText: UI- Account Holder; Bank\_RefID, IBAN: UI- IBAN)
* ACC\_BNK\_Banks (BICCode: UI- Bic, BankName: UI- Bank)
* HEC\_MedicalPractises
* HEC\_MedicalPractice\_UniversalProperties (PropertyName: UI- “Surgery Practice”, "Order Drugs", “Default Shipping Date Offset", "Only Label Required", "Waive Service Fee"; IsValue\_Boolean)
* HEC\_MedicalPractice\_2\_UniversalProperty (HEC\_MedicalPractice\_RefID, HEC\_MedicalPractice\_UniversalProperty\_RefID, Value\_Boolean)

## Doctor data

* USR\_Accounts (BusinessParticipant\_RefID)
* USR\_Account\_FunctionLevelRights\_Group (Label, GlobalPropertyMatchingID, USR\_Account\_FunctionLevelRights\_GroupID, RightName)
* ORM\_USR\_Account\_2\_FunctionLevelRight (Account\_RefID, USR\_Account\_FunctionLevelRightID, AssignmentID)
* CMN\_BPT\_BusinessParticipants(DisplayName: UI- FirstName + LastName; IsNaturalPerson, IfNaturalPerson\_CMN\_PER\_PersonInfo\_RefID)
* CMN\_BPT\_CTM\_OrganizationalUnit\_Staff (OrganizationalUnit\_RefID = practice, BusinessParticipant\_RefID = doctor)
* CMN\_PER\_PersonInfo (Salutation\_General: UI-Salutation, Title: UI-Title, FirstName: UI-First Name, LastName: UI- Last Name)
* CMN\_PER\_CommunicationContacts (Content: UI-Email,Phone; Contact\_Type, PersonInfo\_RefID)
* CMN\_PER\_CommunicationContact\_Type (Type)
* HEC\_Doctors (DoctorIDNumber: UI- LANR; BusinessParticipant\_RefID, Account\_RefID)
* CMN\_BPT\_BusinessParticipant\_2\_BankAccount (CMN\_BPT\_BusinessParticipant\_RefID, CC\_BNK\_BankAccount\_RefID)
* ACC\_BNK\_BankAccounts (OwnerText: UI- Account Holder; Bank\_RefID, IBAN: UI- IBAN)
* ACC\_BNK\_Banks (BICCode: UI- Bic, BankName: UI- Bank)

### Contract data

* CMN\_CTR\_Contracts (Valid from, ValidThrough, ContractName: UI-Contract Name)
* CMN\_CTR\_Contract\_Party (Undersigning\_BusinessParticipant\_RefID, Contract\_RefID, Party\_ContractRole\_RefID)
* CMN\_CTR\_Contract\_Role (RoleName)
* ORM\_CMN\_CTR\_Contract\_Parameter (ParameterName, IfNumericValue\_Value, IsNumericValue, IfStringValue\_Value, IsStringValue)
* CMN\_PRO\_Products (Product\_Name\_DictID, IsProducable\_Internally, Product\_Number)
* HEC\_CRT\_InsuranceToBrokerContract (Ext\_CMN\_CTR\_Contract\_RefID)
* HEC\_CTR\_InsuranceToBrokerContracts\_CoveredHealthcareProducts
* HEC\_CTR\_InsuranceToBrokerContracts\_CoveredPotentialDiagnoses
* HEC\_CTR\_InsuranceToBrokerContracts\_CoveredPotentialBillCodes
* HEC\_CRT\_InsuranceToBrokerContract\_ParticipatingDoctors
* HEC\_CRT\_InsuranceToBrokerContract\_ParticipatingPatient

### GPOS data

* HEC\_BIL\_PotentialCodes (CodeName\_DictID: UI- Gpos name; BillingCode: UI- Gpos nr.)
* HEC\_BIL\_PotentialCode\_Catalogs (GlobalPropertyMatchingID: UI- Case type)
* HEC\_CTR\_I2BC\_CoveredPotentialBillCodes\_UniversalProperties (PropertyName; IsValue\_Number; IsValue\_String)
* HEC\_CTR\_I2BC\_CoveredPotentialBillCodes\_2\_UniversalProperty (Value\_String, Value\_Number: UI- From injections nr, Fee in Eur, Service fee in Eur)
* HEC\_BIL\_PotentialCode\_2\_HealthCareProduct
* HEC\_BIL\_PotentialCode\_2\_PotentialDiagnosis

### Drugs/Diagnoses data

* HEC\_Product[[1]](#footnote-1)
* CMN\_PRO\_Product (Product\_Number, Product\_Name, IsProducable\_Internally)
* HEC\_HIS\_HealthInsurance\_Company (CMN\_BPT\_BusinessParticipant\_RefID, HealthInsurance\_IKNumber)
* HEC\_CTR\_InsuranceToBrokerContracts\_CoveredPotentialDiagnosis (InsuranceToBrokerContract\_RefID, PotentialDiagnosis\_RefID)
* HEC\_DIA\_PotentialDiagnosis
* HEC\_DIA\_PotentialDiagnosis\_CatalogCode(PotentialDiagnosis\_RefID, PotentialDiagnosis\_Catalog\_RefID, Code)
* HEC\_DIA\_PotentialDiagnosis\_Catalog (Catalog\_DisplayName, Catalog\_Name)

### Add doctor to contract

* HEC\_CRT\_InsuranceToBrokerContract (HEC\_CRT\_InsuranceToBrokerContractID, Ext\_CMN\_CTR\_Contract\_RefID)
* HEC\_CRT\_InsuranceToBrokerContract\_ParticipatingDoctor (HEC\_CRT\_InsuranceToBrokerContract\_ParticipatingDoctorID, ValidFrom: UI-Valid From, ValidThrough: UI-Valid To, Doctor\_RefID)

### Add GPOS to contract

* HEC\_BIL\_PotentialCode (CodeName, BillingCode, Price\_RefID, PotentialCode\_Catalog\_RefID)
* HEC\_BIL\_PotentialCode\_Catalog (GlobalPropertyMatchingID)
* HEC\_CTR\_InsuranceToBrokerContracts\_CoveredPotentialBillCode (InsuranceToBrokerContract\_RefID, PotentialBillCode\_RefID)
* HEC\_CTR\_I2BC\_CoveredPotentialBillCodes\_2\_UniversalProperty (Value\_String, PropertyName)
* ORM\_CMN\_Currency
* CMN\_Price
* CMN\_Price\_Value (Price\_RefID, PriceValue\_Amount, PriceValue\_Currency\_RefID)
* HEC\_BIL\_PotentialCode\_2\_HealthcareProduct (HEC\_BIL\_PotentialCode\_RefID, HEC\_Product\_RefID, AssignmentID)
* HEC\_BIL\_PotentialCode\_2\_PotentialDiagnosis (HEC\_BIL\_PotentialCode\_RefID, HEC\_DIA\_PotentialDiagnosis\_RefID)

## Users

* USR\_Accounts
* CMN\_BPT\_BusinessParticipant(DisplayName: UI- First Name, Last Name; IfNaturalPerson\_CMN\_PER\_PersonInfo\_RefID, IsNaturalPerson)
* CMN\_PER\_PersonInfo (FirstName: UI- First Name; LastName: UI-Last Name; Salutation\_General: UI- Salutation; Title: UI-Title)
* CMN\_PER\_CommunicationContact (PersonInfo\_RefID)
* CMN\_PER\_CommunicationContact\_Type (CMN\_PER\_CommunicationContact\_TypeID, Type: UI-Email, Phone;)
* USR\_Account\_ApplicationSetting (Account\_RefID, ItemValue: UI- Receive Notification, ApplicationSetting\_Definition\_RefID)
* USR\_Account\_ApplicationSetting\_Definition(ItemKey)
* Patients
* HEC\_Patient(CMN\_BPT\_BusinessParticipant\_RefID)
* CMN\_BPT\_BusinessParticipant (IsNaturalPerson, IfNaturalPerson\_CMN\_PER\_PersonInfo\_RefID)
* CMN\_PER\_PersonInfo (FirstName: UI- First Name; LastName: UI- Last Name; BirthDate: UI- Birthdate; Gender: UI- Sex)
* HEC\_Patient\_MedicalPractice (HEC\_Patient\_RefID, HEC\_MedicalPractices\_RefID)
* HEC\_Patient\_HealthInsurance (Patient\_RefID, HealthInsurance\_Number: UI- Insurance Number; InsuranceStateCode: UI- Insurance status; HIS\_HealthInsurance\_Company\_RefID)
* HEC\_CRT\_InsuranceToBrokerContract\_ParticipatingPatient (InsuranceToBrokerContract\_RefID, ValidFrom: UI- Valid From; ValidThrough: UI- Valid To; Patient\_RefID)

## Archive

* DOC\_Document (Alias, PrimaryType, GlobalPropertyMatchingID)
* DOC\_DocumentRevision (File\_MIMEType, Document\_RefID, UploadedByAccount, File\_ServerLocation, File\_Name, File\_Description)
* DOC\_Structure (Label)
* DOC\_Document\_2\_Structure (Document\_RefID, Structure\_RefID)

## Case

* HEC\_CAS\_Case (CreatedBy\_BusinessParticipant\_RefID, Patient\_RefID, Patient\_FirstName, Patient\_LastName, Patient\_Gender, Patient\_BirthDate, CaseNumber, Patient\_Age)
* HEC\_ACT\_ActionType (GlobalPropertyMatchingID, ActionType\_Name)
* HEC\_ACT\_PerformedAction (IfPerfomed\_DateOfAction, IfPerformed\_DateOfAction\_Day, IfPerformed\_DateOfAction\_Month, IfPerformed\_DateOfAction\_Year, IsPerformed\_Internally, IsPerformed\_MedicalPractice\_RefID, Patient\_RefID, IfPerformedInternaly\_ResponsibleBusinessParticipant\_RefID)
* HEC\_CAS\_Case\_RelevantPerformedAction (Case\_RefID, PerformedAction\_RefID)
* HEC\_ACT\_PerformedAction\_2\_ActionType (HEC\_ACT\_ActionType\_RefID, HEC\_ACT\_PlannedAction\_RefID)
* HEC\_TRE\_PotentialProcedure\_Package (GlobalPropertyMatchingID)
* HEC\_TRE\_PotentialProcedure\_2\_ProcedurePackage(HEC\_TRE\_PotentialProcedure\_Package\_RefID)
* ORD\_PRC\_ProcurementOrder\_Header (CreatedBy\_BusinessParticipant\_RefID, ProcurementOrder\_Date, Current\_ProcurementOrderStatus\_RefID, ProcurementOrder\_Number)
* ORD\_PRC\_ProcurementOrder\_Status (GlobalPropertyMatchingID, Status\_Code, Status\_Name)
* ORD\_PRC\_ProcurementOrder\_StatusHistory (ProcurementOrder\_Header\_RefID, ProcurementOrder\_Status\_RefID, IsStatus\_Created, TriggeredAt\_Date, TriggeredBy\_BusinessParticipant\_RefID)
* ORD\_PRC\_ProcurementOrder\_Position (CMN\_PRO\_Product\_RefID, Position\_RequestedDateOfDelivery: *UI – Delivery Date*, ProcurementOrder\_Header\_RefID, RequestedDateOfDelivery\_TimeFrame\_From: *UI- Delivery Time From*, RequestedDateOfDelivery\_TimeFrame\_To: *UI- Delivery Time To*, IsProFormaOrderPosition: *UI- Is label only,* BillTo\_BusinessParticipant\_RefID)
* ORD\_PRC\_ProcurementOrder\_Position\_History(IsCreated, TriggeredBy\_BusinessParticipant\_RefID, ProcurementOrder\_Position\_RefID)
* HEC\_PRC\_ProcurementOrder\_Position (Ext\_ORD\_PRC\_ProcurementOrder\_Position\_RefID, OrderedFor\_Patient\_RefID, IfProFormaOrderPosition\_PrintLabelOnly*: UI- Is Label Only*, IsOrderForPatient\_PatientFeeWaived: *UI- Is Patient Fee Waived*, OrderedFor\_Doctor\_RefID, OrderedForDoctor\_DisplayName, OrderedForPatient\_DisplayName)
* HEC\_CAS\_Case\_UniversalProperty (PropertyName*,* IsValue\_Boolean, GlobalPropertyMatchingID, HEC\_CAS\_Case\_UniversalPropertyID)
* HEC\_CAS\_Case\_UniversalPropertyValue (HEC\_CAS\_Case\_RefID, HEC\_CAS\_Case\_UniversalProperty\_RefID, Value\_Boolean: *UI- Send invoice to practice*)
* HEC\_ACT\_PlannedAction (IsPlannedFollowup, IfPlannedFollowup\_PreviousAction\_RefID, IsPerformed, MedicalPractice\_RefID, Patient\_RefID, PlannedFor\_Date, ToBePerformedBy\_BusinessParticipant\_RefID)
* HEC\_ACT\_PlannedAction\_PotentialProcedure (PlannedAction\_RefID, PotentialProcedure\_RefID)
* HEC\_ACT\_PlannedAction\_PotentialProcedure\_RequiredProduct (BoundTo\_HealthcareProcurementOrderPosition\_RefID, PlannedAction\_PotentialProcedure\_RefID, HealthcareProduct\_RefID)
* HEC\_Patient\_Diagnosis (Patient\_RefID, R\_IsConfirmed, R\_PotentialDiagnosis\_RefID)
* HEC\_DIA\_Diagnosis\_Localization (Diagnosis\_RefID, HEC\_DIA\_Diagnosis\_LocalizationID, LocalizationCode: *UI- Localization*)
* HEC\_ACT\_PerformedAction\_DiagnosisUpdate (HEC\_ACT\_PerformedAction\_RefID, IsDiagnosisConfirmed, PotentialDiagnosis\_RefID, HEC\_Patient\_Diagnosis\_RefID, IM\_PotentialDiagnosis\_Code, IM\_PotentialDiagnosis\_Name, IM\_PotentialDiagnosisCatalog\_Name)
* HEC\_ACT\_PerformedAction\_DiagnosisUpdate\_Localization (HEX\_EXC\_Action\_DiagnosisUpdate\_RefID, HEC\_DIA\_Diagnosis\_Localization\_RefID, IM\_PotentialDiagnosisLocalization\_Code)
* HEC\_Patient\_Diagnosis\_Localization (DIA\_Diagnosis\_Localization\_RefID, Patient\_Diagnosis\_RefID)
* HEC\_CAS\_Case\_RelevantPlannedAction (Case\_RefID, PlannedAction\_RefID, GlobalPropertyMatchingID)
* HEC\_ACT\_PlannedAction\_2\_ActionType (HEC\_ACT\_ActionType\_RefID, HEC\_ACT\_PlannedAction\_RefID)
* BIL\_BillPosition (BIL\_BilHeader\_RefID, PositionValue\_IncludingTax)
* HEC\_BIL\_BillPosition (Ext\_BIL\_BillPosition\_RefID, PositionFor\_Patient\_RefID)
* HEC\_BIL\_BillPosition\_BillCode (BillPosition\_RefID, IM\_BillingCode, PotentialCode\_RefID)
* HEC\_CAS\_Case\_BillCode (HEC\_BIL\_BillPosition\_BillCode\_RefID, HEC\_CAS\_Case\_RefID)

## Submit case

* BIL\_BillHeader (BillRecipient\_BuisnessParticipant\_RefID, CreatedBy\_BusinessParticipant\_RefID, TotalValue\_IncludingTax)
* BIL\_BillHeader\_History (BillHeader\_RefID, Comment, TriggeredBy\_BusinessParticipant\_RefID)
* HEC\_BIL\_BillHeader (Ext\_BIL\_BillHeader\_RefID, Patient\_RefID)
* BIL\_BillPosition\_TransmitionStatus (BillPosition\_RefID, IsActive, TransmitionCode, TransmittedOnDate, TransmitionStatusKey)
* BIL\_BillPosition\_PropertyValue (BIL\_BillPosition\_RefID, PropertyKey, PropertyValue)

# Special processes:

## DOC APP

### New case:

#### Validation:

During creation of a new case, several validations are triggered. First, the system checks whether or not a contract in which patient’s Health Insurance Provider (HIP) participates exists. If there are one or more contracts found, the system checks whether any of the doctors in the current logged in doctor’s practice have a valid participation consent on any of such contracts. If yes, the system checks for valid patient’s participation consent on such contracts. If there is one or more valid patient’s participation consent, the system checks whether selected drugs have GPOS on any of such contracts. If yes, the doctor can schedule a treatment for the selected patient with the selected drug. If treatment for selected date exist, case will not be saved and popup with message will be displayed.

#### Case creation:

If all validations are passed, a new case is created in the system. At this point case GPOS-es are calculated. If the order button was set to ‘Yes’, a new drug order is also created. Case can be order only, treatment only or both order and treatment. Treatment can be with or without OCT. If the case treatment date is set to a date in the past, the order button will be disabled, since no order can be placed in the past.

#### Case edit:

If the case has a drug ordered, and the drug is in MO1 (ordered) or MO2 (accepted) status, any change to patient, treatment date and the drug itself causes the current order to be cancelled and a new one to be placed in its place. If the drug order is in MO3 (shipped) status, the order cannot be edited and won’t be affected by changes to treatment date or patient. If the drug order is in MO4 (rejected) status, if the order button is set to ‘Yes’, a new order will be created. If not, the MO4 order will be cancelled. If the order is in MO6 (cancelled) status, if the order button is set to ‘Yes’, a new order will be created. If the treatment date is changed to be in the past, if the drug order is in any status other that MO3, it will be cancelled and no new order will be placed.

After saving the edited case, case GPOS-es will be recalculated, in case drug, diagnose or localization was changed.

#### Error correction edit:

During error correction edit, if the case edited is an aftercare, only date of aftercare and the aftercare doctor can be updated. If the case edited is a treatment, treatment date, diagnosis and treatment doctor can be updated. If the diagnose is changed, GPOS-es are re-calculated. If the case edited is oct, oct date , localization and oct doctor can be updated.

#### GPOS calculation:

GPOS-es are calculated based on selected drug, diagnose and localization combination. Every GPOS has drug, diagnose and from which treatment it’s considered attached to itself. The system then finds the most suitable GPOS based on the drug, diagnose, localization and number of previous treatments combination.

### Submit case:

#### Validation:

During case submission, several validations are triggered. If the case being submitted is treatment, the system checks whether the OP doctor has a valid participation consent on a contract on which drug/diagnose combination can be found for the given treatment date and check if oct doctor selected.

If the submitted case is an aftercare, the checks whether the AC doctor has a valid participation consent on a contract on which drug/diagnose combination can be found for the given date when the aftercare was performed. If yes, system then checks whether the patient has a valid participation consent on the same contract. After that, system checks if aftercare date after treatment date. If yes, system then checks how many days has between treatment and aftercare.

If the submitted case is an OCT, the system checks whether the OCT doctor has a valid participation consent on a contract on which drug/diagnose combination can be found for the given date when the OCT was performed. If yes, system then checks whether the patient has a valid participation consent on the same contract. After that, system checks if OCT date is in the past or present. Then system checks how many OCTs are already submitted, how much OCTs are submitter before treatment, how much days between OCT and treatment and how much days has between OCT and last treatment date, does practice have an OCT device.

If all validations pass, case can be submitted after providing credentials of the doctor assigned as the OP doctor to the case if case being submitted is a treatment, or AC doctor if the case being submitted is an aftercare or OCT doctor if the case being submitted is and OCT.

#### Submission

If the submitted case is a treatment, after the case is submitted, it is transferred to FS1 (Submitted to MM) status and is no longer visible in the planning section of the DocApp and a new aftercare is created in AC1 (open) status. If the case submitted is an aftercare, it is transferred to FS1 (Submitted to MM) status and is no longer visible in the aftercare section of the DocApp. If the case submitted is and OCT, it is transferred to FS1 (Submitted to MM) status and is no longer visible in the oct section in DocApp and a new OCT is created in OCT1 (open) status.

### Cancel case:

#### Cancellation:

If the case being cancelled is order only, the order will be cancelled and transferred into MO6 (order cancelled) status. If the case being cancelled is a treatment without order, the case will be cancelled and transferred to OP4 (case deleted) status. If the case being cancelled is a treatment with an order, the user is prompted to confirm whether he/she wants to cancel only the order or both order and treatment. If the user selects order only, the order is cancelled and transferred into MO6 status, while the case status will remain the same. If the user selects to cancel both order and treatment, the order will be cancelled and transferred to MO6 status, while the treatment will be cancelled and transferred to OP4 status. If the case being cancelled is OCT, the OCT will be cancelled and transferred into OCT4 (withdrawn) status.

### Multi edit/submit:

#### Multi edit:

Selecting any of the checkboxes will trigger multi edit/submit mode.

On the planning and aftercare sections of the DocApp, if the user selects OP or AC doctor, upon clicking on the save button, the selected cases will be updated with selected AC or OP doctor or both. Cases on which selected AC or OP doctors are already assigned will be ignored.

On the settlement section of the DocApp, if the user selects status to which he/she wishes to transfer selected cases, upon clicking on the save button, selected cases will be updated and transferred to selected status. If the cases are already in the selected status, they will be ignored.

On the OCT section of the DocApp, if the user selects OCT doctor, upon clicking on the save button, the selected cases will be updated with selected OCT doctor. Cases on which selected OCT doctors are already assigned will be ignored.

#### Multi submit:

Multi submit is triggered in the same way as multi edit. If the user selects AC or OP doctor, selected cases will first be updated, and then, upon clicking on the submit button and after providing OP doctor’s credentials, submitted. If there are multiple OP doctors assigned to selected cases, password confirmation popups will appear for every doctor individually. If any of the popups is cancelled, the system will move on to the next doctor.

-This behavior is applied to the aftercare page as well.

On the OCT section of the DocApp, if the user selects OCT doctor, selected cases will first be updated and then, upon clicking on the submit button submitted.

### Patients manipulation:

#### Validation:

Upon creation of a new patient, several validations are triggered. First, patient birthdate must be in the past. Next, patient insurance number has two validations. First validates old insurance numbers, and the second one validates new ones. Same is applied to the insurance status. If user enters old insurance number and an old insurance status, user will be prompted with a popup notifying them that the entered insurance number and status pass the old validation and whether or not they wish to proceed with saving. If the user selects yes, the patient data will be saved.

#### Participation consent:

Upon creation of patient’s participation consent, system validates the consent date. The date is validated so that it fits into selected contracts validity timeframe.

### Notifications and expired session:

Notifications are sent whenever the data in the database is updated for the currently logged in doctors practice. Upon session expiration, a popup appears which informs the doctor that the session has expired and upon clicking on the OK button, the doctor is redirected to the login page.

### Submit order

#### Submission

During order submission, popup will be displayed with information about that order. Popup contains information about delivery date, delivery time and pharmacy. If default pharmacy doesn’t exist, user can add a new pharmacy in order popup. In this case, the user should add pharmacy information, such as: address, number, city and postal code. After that order will be transferred to MM in MO1 status and pharmacy will be created if already doesn’t exist.

## MM APP

### Dashboard actions:

#### Case export to edifact (FS1 -> FS2):

During case export to edifact, only FS1 cases (treatments, aftercares and OCTs) are taken into account and transferred to edifact files. At this point, an excel report is also generated. One edifact file is created per HIP of the patients for whom the cases were created. If any of the cases has already been submitted, but was returned as a negative response from the HIP and then resubmitted to MM again, case is marked with an increase of the negative tries counter (RGI section of the edifact file).

#### HIP feedback import (FS2 -> FS4/FS5):

During HIP feedback import, if the file uploaded is an excel file, system will consider it to be a positive feedback. All of the bill positions whose numbers are contained in the file will have their FS status set to FS4. If the file uploaded is an edifact or a text file, the system will consider it to be a negative feedback. All of the bill positions whose numbers are contained in the file will have their FS status set to FS5.

#### Payment (FS4 -> FS7):

During payment, popup will be displayed with 2 options, status change and status changed on date. If user select status change, all FS4 cases will be transferred to FS7 status and report will be created. If user unselect status change, only report will be generated. Date of status change will be date added in popup.

#### Show all cases in error correction:

Redirect logged user to treatment page with preselected appropriate filter (show FS5 treatments).

#### Show all on hold cases:

Redirect logged user to treatment page with preselected appropriate filter (show FS3 treatments).

#### New orders export (MO1 -> MO2):

During the export of the new orders, all of the MO1 orders will be exported into an excel file which will be uploaded to the archive. All of the procurement order positions will have their MO status set to MO2.

#### Shipped orders import (MO2 -> MO3):

During import of the shipped orders, all of the procurement order positions whose numbers are in the excel file uploaded will have their MO status set to MO3.

#### Complete report:

Complete report contains all of the treatments in the system (both treatments on planned and submitted cases), only those aftercares (both open and submitted to MM) whose treatments are submitted to the MM and all submitted OCTs.

#### Complete order report:

Complete order report contains all of the orders in the system submitted to MM.

#### Show temporary doctors:

Redirect logged user to doctors page with preselected appropriate filter (show all temporary doctors).

### Manual case status/management fee status change:

This action is invoked by selecting any of the cases in the list on the treatments page by checking the checkbox in its row and clicking on the Status/Management fee update button in the table header. At this time, a popup will appear on which the user can select whether he/she wants to change only management fee status, case status or both. Before confirming the action, username and password of an admin user must be provided. If any of the selected cases is already in the case status the user wishes to transfer them to, warning message will appear informing the user that those cases will not be affected since they are already in that status.

### Drug order rejection:

This action is invoked by clicking on Reject order button in the row of the order the user wishes to reject. At this time, a popup will appear asking the user to enter credentials. Before confirming the action, username and password of an admin user must be provided.

### New practice/doctor:

During new practice/doctor creation, several validations are triggered. Practice BSNR must be unique in the system, as well as the login e-mail address used. LANR must be unique in the selected practice, but multiple doctors with the same LANR in different practices can exist. Doctor login e-mail must be unique in the system.

### Temporary doctor:

Temporary doctors can be saved as new doctors in the system or merged with the existing doctors in the system. After the user enters LANR and selects a practice, the system will try to find an existing doctor in the selected practice with the same LANR. If the system finds such doctor, the user is offered an option to merge all of the data of the temporary doctor (all of the open aftercares) to the existing doctor. If a new doctor is created from the temporary doctor, all of the temporary doctor’s data is merged to this new doctor.

### Doctor contract consents:

Doctor can have only one consent on a contract per time period. No overlaps on the same contract are allowed.

### Contracts:

During creation or edit of a contract, all of the entered data, such as participating HIPs, covered drugs and diagnoses, GPOS-es etc. are saved only when the user clicks on the save button. If the page is refreshed or if the users navigates away from it, the data is not saved.

During creation or edit of a GPOS, GPOS-es which don’t have any drugs and diagnoses attached to them are automatically added to the corresponding case type (operation or aftercare). If a GPOS has a drugs attached to it, but no diagnoses, it is added automatically to the corresponding case type (operation or aftercare) when any of those drugs is used for an operation. These special GPOS-es must have their injections from field left empty, otherwise they won’t be taken into consideration.

If any of the due dates checkbox is left unchecked, the value of that due date is infinite.

During addition of a doctor to a contract, the user must specify doctors consent start date, which must be greater or equal to the contract’s start date and less or equal to the contract’s end date. During removal of a doctor from a contract, the user must specify doctors consent end date, which must be greater or equal to doctor’s consent start date and less or equal to date on which the action is taking place.

Each contract has additional settings, contract constraints:

1. Consent duration – represents the number of months that consent is valid for
2. Time between surgery and aftercare – represents the number of days between surgery and aftercare. Aftercare can be submitted only in that range.
3. Time between surgery and billing – represent the number of days between surgery and billing. Billing is impossible only in that range.
4. Submission replay – represents the number of days between billing and feedback from health insurance
5. Response payment – represents the number of days between feedback and disbursement.
6. Response rejection – represents the number of days between feedback and rejection by health insurance.
7. Max OCT number – represents the number of OCT which can be submitted per treatment year/settlement year. Treatment/settlement year is also parameter expressed in days.
8. OP renews consent – represents automatically renew consent after surgery within the validity period.
9. Use settlement year – represents using settlement year instead of treatment year. That is specified fir AOK contracts.
10. Doctor needs certification – represents if doctor has certification for submit OCT. It will be checked only if “Use settlement year” is set to true.

### Notifications and expired session:

Notifications are sent whenever the data in the database is updated. If the user is on the dashboard page, the dashboard data is refreshed as well. Upon session expiration, a popup appears which informs the user that the session has expired and upon clicking on the OK button, the user is redirected to the login page.

1. HEC\_Product. HEC\_ProductID has been used through application as Drug ID. [↑](#footnote-ref-1)