

Insurance Management Information System (IMIS) Application
Functional Design Specifications
Functional Area: Insurees and Policies, Administration of registers,
Management of claims and general management

Created by:	Exact Software Ltd
Deliverable Owner:	Swiss TPH
Deliverable Approvers:	
Last Revision Date:	16 July 2018
Version & Status:	1.2.0

Contents

1. OVERVIEW.....	6
1.1. INTRODUCTION	6
1.2. SCOPE	6
1.3. AUDIENCE	6
1.4. ASSUMPTIONS	6
1.5. DOCUMENT HISTORY	7
2. SYSTEM ARCHITECTURE.....	8
2.1.1. IMIS online web Application	8
2.1.2. IMIS offline web Application	8
2.1.3. Window services and Web services.....	9
IMIS has four different applications:	9
2.1.4. Android phone applications.....	9
2.1.5. IMIS Data warehouse	9
3. PROGRAMMING APPROACH, STANDARDS AND METHODS	10
3.1. MULTI TIER DEVELOPMENT.....	10
3.2. CODING STANDARDS AND CONVENTIONS.....	15
3.2.1. Entities.....	16
3.2.2. Methods.....	16
3.2.3. Controls	16
3.2.4. Objects	16
3.2.5. Resources	16
3.3. PHOTO STORAGE	17
4. IMIS MENU STRUCTURE AND INTERFACE FLOW	18
4.1. MAIN MENU	18
4.2. MENU INSUREES AND POLICIES	18
4.3. MENU CLAIMS	19
4.4. MENU ADMINISTRATION	19
4.5. MENU PRICE LISTS.....	20
4.6. MY PROFILE	20
4.7. MENU TOOLS	20
5. FUNCTIONAL AREA DESCRIPTION.....	21
5.1. GENERIC INTERFACE FEATURES	21
5.1.1. Add, Edit, Delete, Save and buttons	21
5.1.2. User Security	22
5.1.3. 'Historical' records	22
5.1.4. Data grids, hyperlinks and pages	23
5.1.5. Multi Language aspects	24
5.1.6. Session expiry	26
5.1.7. Mandatory fields	26
5.1.8. Status bar and Popup	27
5.2. WEB PAGES	27
5.2.1. Default.aspx	29

5.2.2.	ForgotPassword.aspx	30
5.2.3.	ChangePassword.aspx	31
5.2.4.	Home.aspx	32
5.2.5.	Logout.aspx	33
5.2.6.	Family.aspx	34
5.2.7.	FindFamily.aspx	37
5.2.8.	ChangeFamily.aspx	38
5.2.9.	FindInsuree.aspx.....	40
5.2.10.	Insuree.aspx.....	41
5.2.11.	FindPolicy.aspx	43
5.2.12.	Policy.aspx	45
5.2.13.	FindPremium.aspx	46
5.2.14.	Premium.aspx	47
5.2.15.	OverviewFamily.aspx	49
5.2.16.	FindClaims.aspx.....	51
5.2.17.	Claim.aspx.....	53
5.2.18.	ClaimOverview.aspx.....	55
5.2.19.	ClaimReview.aspx.....	59
5.2.20.	ClaimFeedback.aspx.....	61
5.2.21.	BatchRun.aspx	62
5.2.22.	ClaimAdministrator.aspx	64
5.3.	CHECKING OF CLAIMS PROCESS	65
5.4.	BATCH VALUATION PROCESS	67
5.5.	CALCULATION OF 'RELATIVE PRICES INDEX' PROCESS	69
5.6.	GRAPHICAL PAGE ROUTING OVERVIEW CLAIM INTERFACES	71
5.7.	OVERVIEW OF STATUS FIELDS	72
5.7.1.	FindHealthFacility.aspx	73
5.7.2.	HealthFacility.aspx	74
5.7.3.	FindProduct.aspx	75
5.7.4.	Product.aspx	76
5.7.5.	FindMedicallItem.aspx	79
5.7.6.	MedicallItem.aspx	80
5.7.7.	FindMedicalService.aspx	81
5.7.8.	MedicalService.aspx	82
5.7.9.	FindOfficer.aspx	83
5.7.10.	Officer.aspx	84
5.7.11.	FindPayer.aspx	86
5.7.12.	Payer.aspx	87
5.7.13.	FindPriceListMI.aspx	88
5.7.14.	PriceListMI.aspx	89
5.7.15.	FindPriceListMS.aspx	90
5.7.16.	PriceListMS.aspx.....	92
5.7.17.	FindUser.aspx	93
5.7.18.	User.aspx	94

5.7.19.	Locations.aspx	96
5.7.20.	MoveLocation.aspx	97
5.7.21.	EmailSettings.aspx.....	98
5.7.22.	Registers.aspx.....	99
5.7.23.	PolicyRenewals.aspx	101
5.7.24.	IMIS Extracts	102
5.7.25.	Reports.aspx	113
5.7.26.	Utilities.aspx	115
5.7.27.	IMIS.MASTER (used for Menu and Quick Inquiry)	116
5.8.	GRAPHICAL PAGE ROUTING OVERVIEW.....	117
6.	DATABASE DESIGN	118
6.1.	DATABASE INTERCONNECTIVITY	118
6.2.	TABLES WITH MAIN PROPERTIES.....	119
6.1.1.	tblRegion (represented in the database as view).....	120
6.1.2.	tblDistricts (represented in the database as view)	122
6.1.3.	TblWard (Municipality): (represented in the database as view)	124
6.1.4.	tblVillages (represented in the database as view)	126
6.1.5.	TblUsers	127
6.1.6.	tblUsersDistricts	128
6.1.7.	tblOfficer	130
6.1.8.	tblOfficerVillages	132
6.1.9.	tblICDCodes	133
6.1.10.	tblItems.....	134
6.1.11.	tblPLItems	135
6.1.12.	tblPLItemsDetail	135
6.1.13.	tblServices	137
6.1.14.	tblPLServices	138
6.1.15.	tblPLServicesDetail	138
6.1.16.	tblHF	140
6.1.17.	tblFamilies	141
6.1.18.	tblInsuree	143
6.1.19.	tblInsureePolicy	145
6.1.20.	tblHealthStatus	146
6.1.21.	tblPhotos	146
6.1.22.	tblPolicy	147
6.1.23.	tblProduct	149
6.1.24.	tblProductItems	151
6.1.25.	tblProductServices	152
6.1.26.	tblRelDistr.....	153
6.1.27.	tblPremium	154
6.1.28.	tblPayer	155
6.1.29.	tblClaim	156
6.1.30.	tblClaimItems.....	158

6.1.31.	tblClaimServices.....	160
6.1.32.	tblClaimDedRem	162
6.1.33.	tblFeedback.....	163
6.1.34.	tblPolicyRenewals	164
6.1.35.	tblPolicyRenewalsDetails	164
6.1.36.	tblRellIndex	166
6.1.37.	tblBatchRun.....	167
6.1.38.	tblIMISDefaults	168
6.1.39.	tblReporting	169
6.1.40.	tblControl	169
6.1.41.	tblEducations.....	170
6.1.42.	tblProfessions.....	170
6.1.43.	tblIdentificationTypes	170
6.1.44.	tblLegalForms.....	171
6.1.45.	tblRelations.....	171
6.1.46.	tblFamilyTypes	172
6.1.47.	tblConfirmationTypes	172
6.1.48.	tblCeilingInterpretation	172
6.1.49.	tblLanguages.....	173
6.1.50.	tblEmailSettings.....	173
6.1.51.	tblHFSublevel	173
6.1.52.	tblPayerType	174
6.1.53.	tblFromPhone.....	174
6.1.54.	tblSubmittedPhotos	175
6.1.55.	tblLogins	175
6.1.56.	tblGender.....	175
6.2.	GRAPHICAL RELATIONSHIP OVERVIEWS.....	177
6.3.	IMIS ON-LINE AND OFF-LINE	184

1. Overview

1.1. Introduction

This document cover the design of IMIS Master Version Phase 3 to 5 version 17.3.11 to 18.12 for Insurees and policies , administration of registers,Management of Claims and general management, based on the specification of the Insurance Management Information System for the new Community Health Fund structure.

We will elaborate on the technologies used and will highlight certain design aspects of the application.

1.2. Scope

This document will cover the overall design of the Insurance Management Information System for the new IMIS Master version Phase 3 to Phase 5: Insurees and policies, administration of registers, management of claims and general management.

The document will cover the following main areas of the development in the following order:

- Overall system architecture
- Online and Offline Web Application
- Programming approach, standards and methods
- User Interface design
- Database design

The application is developed with a multi-tier approach. A separate section will cover the overall ‘skeleton’ of the IMIS application by elaborating on the internal structure of the Visual Studio solution. Separate paragraphs will cover the coding standards and user security aspects.

The user interface design chapters will start with the menu-structure of the application. Thereafter we will provide the design of each interface separately and will cover the functional area by providing a brief overview of the interface, the menu-link, reference to use cases and the main programming methods and properties used within the interface (programming model).

Based on the data model, a full database design documented with its main properties and constraints. Also a graphical representation included.

This functional design document in conjunction with the draft specification of the Insurance Management Information System version 18.12 should remain the final authority for developers.

1.3. Audience

This document used to develop and implement components for the IMIS application. They translate the functionality described in this document directly into the actual functionality of the Administration of registers and Management of Insurees and policies.

This document is also used by testers to define test scenarios for the functional areas and to test application security.

1.4. Assumptions

Readers will be familiar with terminology used within the IMIS project.

1.5. Document History

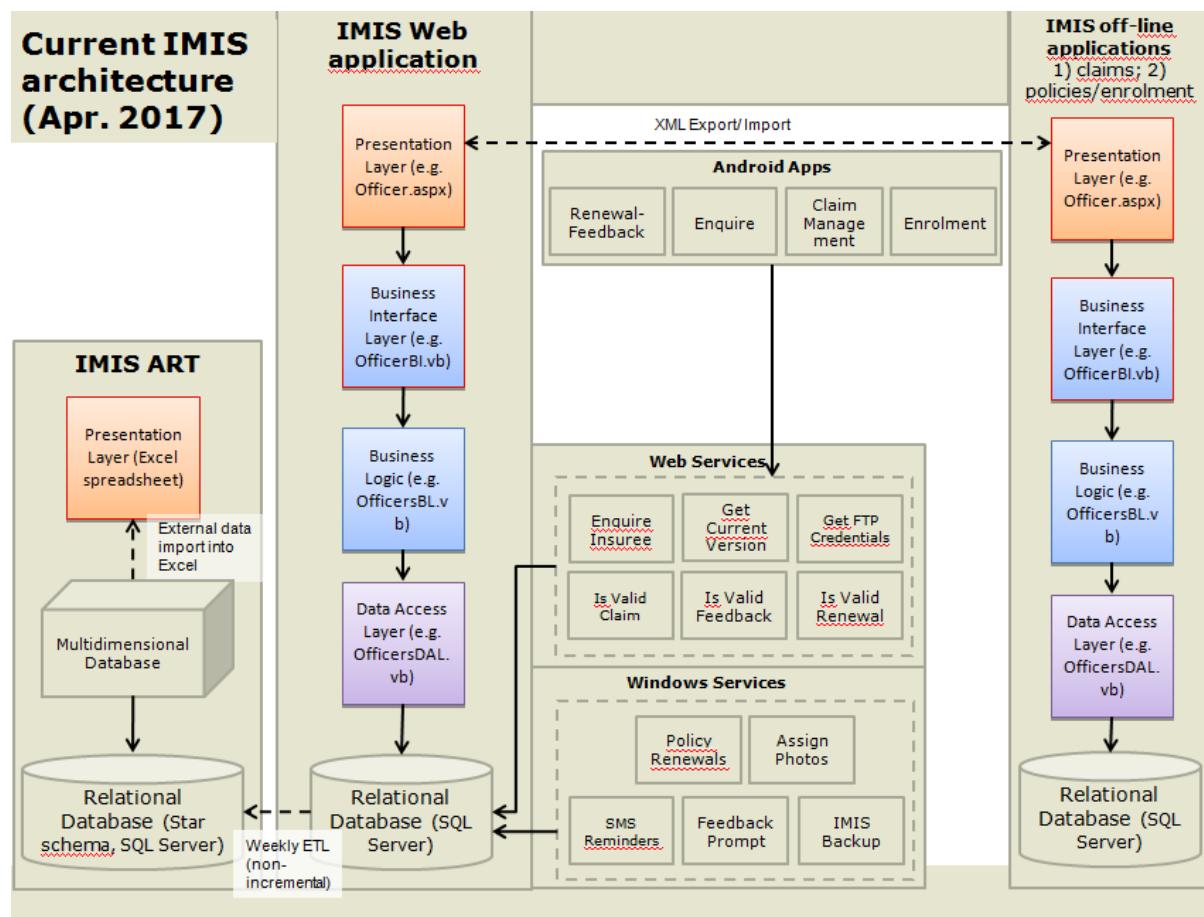
Ver.	Description	Author	Date
16.1.1	IMIS Master phase 1	Hans van Hoppe	24-10-16
16.2.0	IMIS Master phase 2	Hans van Hoppe	15-11-16
17.3.11	IMIS Master Phase 3	Rogers Obed	24-04-17
18.00	IMIS Master Phase 4	Salumu Mapande	05-04-18
18.1.2	IMIS Master Phase 5	Salumu Mapande	16-07-18

2. System Architecture

This chapter will cover the overall architectural design of the IMIS system. The design can be divided in several components:

- IMIS online web application
- IMIS offline web application
- Android phone applications
- Windows services and Web services
- IMIS data warehouse

See below a graphical overview.



2.1.1. IMIS online web Application

The IMIS online web application with its 4-Tier architecture is covered in Chapters 3,4,5 and 6.

2.1.2. IMIS offline web Application

IMIS offers, besides the online application, also the so-called 'offline' applications. IMIS offers 2 offline applications:

- Offline 'enrollment' office (CHF-offline)
- Offline Health Facility (HF-offline)

The offline applications are used in case connectivity is non-existing or relatively poor. The online and offline components are exactly the same in terms of coding and architecture. The IMIS offline web application with its 4-Tier architecture is covered in Chapters 3,4,5 and 6.

2.1.3. Window services and Web services

IMIS has four different applications:

- Renew-Feedback application - used by the enrollment officer to re-new insuree policies. The application is also used to provide health facility feedback from insurees.
- Enquire application - used for checking insuree details.
- Claim management application - used by the claim administrator to register claim details from the insurees.
- Enrollment application- used for enrolling insuree by taking their pictures and provides the insurance number.

For more details on Android phone applications, please refer to the document on ***IMIS Windows and Web Services***.

2.1.4. Android phone applications

Windows services are used for automatic database backup, update insurance policy expiry details and to send SMS to insurees; while Web services are used for connecting between the android application and the core IMIS application.

Web services are also used to execute background activities such as in-background extracts. For more details, please refer to the document on ***IMIS Android Mobile Phone***.

2.1.5. IMIS Data warehouse

Data warehouse is a system used for data analysis and reporting. For more information on the IMIS Data warehouse, please refer to the document on ***IMIS Data warehouse Deployment***.

3. Programming approach, standards and methods

This chapter will further elaborate the architecture of the IMIS solution, coding standards and the methodology of storing images.

3.1. Multi Tier development

IMIS is developed with a multi tier approach. Each Tier is represented in the Visual Studio solution as a separate project.

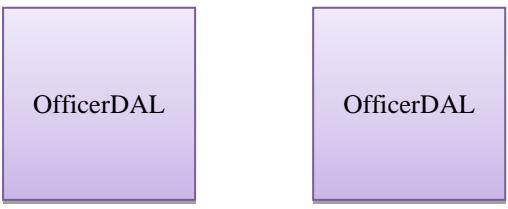
The following projects (tiers) can be found in the IMIS Solution:

Presentation Layer	→ IMIS
Business Interface Layer	→ IMIS_BI
Business Logic Layer	→ IMIS_BL
Data Access Components Layer	→ IMIS_DAL

At the base of the application is the Data Access Layer (DAL) which communicates via a standard library with the SQL Server. Above the DAL is the Business Logic Layer (BLL) which references the DAL and is used to store all methods used in the application. The next layer is the Business Interface Layer (BIL) which acts purely as an organization layer, here only the method calls are stored. Finally the web pages are stored in the Presentation Layer (PL).

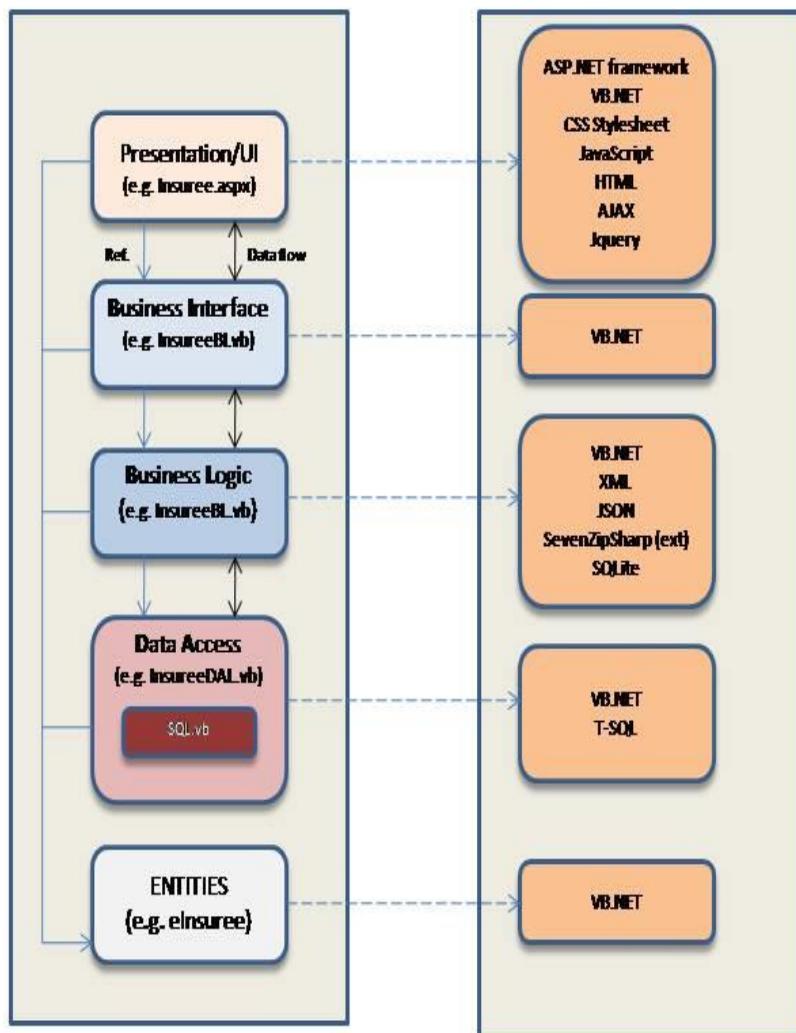
The example below shows how the application references each layer with the Officer Entity

<p>Presentation Layer</p> 	<p>The page Officer.aspx is used to enter and edit officer (Enrollment Assistant) data and use the Officer Entity. While Validation of the page is handled client side within the aspx page, all server side actions are handled by referencing methods found in the Business Interface Layer.</p>
<p>Business Interface Layer</p> 	<p>In the Business Interface Layer, each page has a separate class which stores all the methods available for the page. No programming code is found in these classes the layer acts purely as a container to store specific methods for specific forms. Each method references methods in the Business Logic layer</p>
<p>Business Logic Layer</p>  	<p>The Business Logic Layer is the part of the program that contains nearly all the logic for the application. Events triggered from the presentation layer are sent via the Business Interface Layer to the Business Logic Layer where most of the programmatic decision making methods are found. In most cases the methods will reference the Data Access Layer before returning the results back to the presentation layer.</p>

<p>Data Access Layer</p> 	<p>The Data Access layer is used for storing all methods used to communicate with the database via a Standard Library Class . In the case of IMIS, each Entity has a separate class for calls to the SQL Server. Methods are referenced from the Business Logic Layer and results are sent back to the Presentation Layer.</p>
<p>SQL Server Class</p> 	<p>The SQL Server contains standard methods native to SQL Server. All class within the DAL SQL namespace reference the Exact.DLL which communicates directly with the SQL Server.</p>

See below the graphical illustration of Multi Tier Development Tools and Technology used

Multi Tier Development tools / Technologies used



Presentation layer or User Interface (UI): HTML and CSS languages are used for interfaces development. Technologies such as JavaScript, AJAX and Jquery are used for creation of dynamic webpages.

Business Interface: This layer is purely designed in VB.NET

Business Logic: This layer is developed in VB.NET. XML language is used for XML files creation. Json is used for sending data to the phone in json string format. SevenZipSharp is used for compressing data to zipped folder. SQLite is used for data storage in Phone.

Data Access: This layer is developed in VB.NET. TSQL is used to write SQL statements and queries to send and retrieve data from and to the database.

Enrolment Officers Details

Code	O001
Other Names	Alex
Last Name	Joseph
Date of Birth	09/06/1964
Phone Number	255768108131
Email	alex@gmail.com
District	Dodoma
Substitution	100003 - Mgulo Alber
Works To	07/01/2017

Municipality

MUNICIPALITY
<input type="checkbox"/> Chahwa
<input checked="" type="checkbox"/> Hazina
<input type="checkbox"/> Chihanga

Village

VILLAGE
<input checked="" type="checkbox"/> Hazina mtaa
<input checked="" type="checkbox"/> Mlezi

Feedback collector Details

Code	C001
Last Name	Ismail
Other Names	Rage
Phone Number	255768108131
Date of Birth	01/01/1979

Save Cancel

When the form loads, the page load event is fired, references to the entity eOfficer and the Business Interface class IMIS_BI.OfficerBI

```
Dim eOfficer As New IMIS_EN.tblOfficer
```

```
Dim Officer As New IMIS_BI.OfficerBI
```

3 methods are called, getDistricts, to load the districts, GetOfficers, to load the list of substitution officers in the selected District and LoadOfficer to load the details of the officer if editing.

```
Officer.GetDistricts(imisgen.getUserId(session("user")), True)
```

imisgen.getUserId("user") indicates the User loged in , True indicates that the first row is a user prompt

```
Officer.GetOfficers(1)
```

1 indicates the district selected

```
Officer.LoadOfficer(eOfficer)
```

eOfficer is the entity of the officer

IMIS Business Interface Layer – Class Officer

Public Class OfficerBI

```
Public Function SaveOfficer(ByRef eOfficer As IMIS_EN.tblOfficer) As Integer
```

```
    Dim saveData As New IMIS_BL.OfficersBL
```

```
    Return saveData.SaveOfficers(eOfficer)
```

```
End Function
```

```
Public Function GetDistricts(ByVal userId As Integer, Optional ByVal showSelect As Boolean = False) As DataTable
```

```
    Dim Districts As New IMIS_BL.LocationsBL
```

```
    Return Districts.GetDistricts(userId, showSelect)
```

```
End Function
```

```
Public Function GetOfficers(ByVal DistrictId As Integer, ByVal showSelect As Boolean) As DataTable
```

```
    Dim getDataTable As New IMIS_BL.OfficersBL
```

```
    Return getDataTable.GetOfficers(DistrictId, showselect)
```

```
End Function
```

```
Public Sub LoadOfficer(ByRef eOfficers As IMIS_EN.tblOfficer)
```

```
    Dim loadEntity As New IMIS_BL.OfficersBL
```

```
    loadEntity.LoadOfficer(eOfficers)
```

```
End Sub
```

End Class

Each method now references the specific logic class from the Business Logic class

GetDistricts calls the method from the class LocationsBL, while GetOfficers and LoadOfficer calls methods from the OfficerBL class.

```

Public Class LocationsBL
    Private imisgen As New GeneralBL
    Public Function GetDistricts(ByVal userID As Integer, ByVal showSelect As Boolean, Optional IncludeNational As Boolean = False) As DataTable
        Dim Districts As New IMIS_DAL.LocationsDAL
        Dim dt As DataTable = Districts.GetDistricts(userID)
        If IncludeNational Then
            Dim dr As DataRow = dt.NewRow
            dr("DistrictId") = -1
            dr("DistrictName") = imisgen.getMessage("M_NATIONAL")
            dt.Rows.InsertAt(dr, 0)
        End If

        If dt.Rows.Count > 1 Then
            If showSelect = True Then
                Dim dr As DataRow = dt.NewRow
                dr("DistrictId") = 0
                dr("DistrictName") = imisgen.getMessage("T_SELECTDISTRICT")
                dt.Rows.InsertAt(dr, 0)
            End If
        End If
        Return dt
    End Function

```

Public Class OfficersBL

```

Public Function GetOfficers(ByVal DistrictId As Integer, ByVal showselect As Boolean) As DataTable
    Dim getDataTable As New IMIS_DAL.OfficersDAL
    Dim dtOfficer As DataTable = getDataTable.GetOfficers(DistrictId)

    If showselect = True Then
        Dim dr As DataRow = dtOfficer.NewRow
        dr("OfficerID") = 0
        dr("Code") = imisgen.getMessage("T_SELECTOFFICER")
        dtOfficer.Rows.InsertAt(dr, 0)
    End If
    Return dtOfficer
End Function

Public Sub LoadOfficer(ByRef eOfficers As IMIS_EN.tblOfficer)
    Dim load As New IMIS_DAL.OfficersDAL
    load.LoadOfficer(eOfficers)
End Sub

```

Each reference from the BLL, references a similar method from the specific DAL class.

```

Public Function GetDistricts(ByVal UserID As Integer) As DataTable
    Dim data As New EXACT.ExactSQL
    data.setSQLCommand("select * from tblDistricts order by DistrictName", CommandType.Text)
    data.params("@UserID", SqlDbType.Int, UserID)

    Return data.FillData
End Function

Public Class OfficersDAL
Public Function GetOfficers(ByVal DistrictID As Integer) As DataTable
    Dim data As New EXACT.ExactSQL
    data.setSQLCommand("select tblOfficer.*,Districtname from tblOfficer inner join tblDistricts on tblOfficer.DistrictID ="
    & "tblDistricts.DistrictID where tblOfficer.LegacyId is Null AND tblOfficer.DistrictID = @DistrictID ORDER BY LastName",
    CommandType.Text)
    data.params("@DistrictID", SqlDbType.Int, DistrictID)
    Return data.FillData
End Function

Public Sub LoadOfficer(ByRef eOfficers As IMIS_EN.tblOfficer)
    Dim data As New ExactSQL
    Dim dr As DataRow
    data.setSQLCommand("select * from tblOfficer where OfficerID=@OfficerId", CommandType.Text)
    data.params("@OfficerId", SqlDbType.Int, eOfficers.OfficerID)
    dr = data.FillData()(0)
    If Not dr Is Nothing Then
        Dim eDistricts As New IMIS_EN.tblDistricts
        eDistricts.DistrictID = dr("DistrictID")
        eOfficers.tblDistricts = eDistricts
        eOfficers.Code = dr("Code")
        eOfficers.LastName = dr("LastName")
        eOfficers.OtherNames = dr("OtherNames")
        eOfficers.DOB = IIf(dr("DOB") Is DBNull.Value, Nothing, dr("DOB"))
        eOfficers.Phone = IIf(dr("Phone") Is DBNull.Value, Nothing, dr("Phone"))
        eOfficers.WorksTo = IIf(dr("WorksTo") Is DBNull.Value, Nothing, dr("WorksTo"))
        Dim eofficer2 As New IMIS_EN.tblOfficer
        eofficer2.OfficerID = IIf(dr("OfficerIDSubst") Is DBNull.Value, 0, dr("OfficerIDSubst"))
        eOfficers.tblOfficer2 = eofficer2
        eOfficers.VEOCode = IIf(dr("VEOCode") Is DBNull.Value, Nothing, dr("VEOCode"))
        eOfficers.VEOPhone = IIf(dr("VEOPhone") Is DBNull.Value, Nothing, dr("VEOPhone"))
        eOfficers.VEOLastName = IIf(dr("VEOLastName") Is DBNull.Value, Nothing, dr("VEOLastName"))
        eOfficers.VEOOtherNames = IIf(dr("VEOOtherNames") Is DBNull.Value, Nothing, dr("VEOOtherNames"))
        eOfficers.VEODOB = IIf(dr("VEODOB") Is DBNull.Value, Nothing, dr("VEODOB"))
        If Not dr("ValidityTo") Is DBNull.Value Then
            eOfficers.ValidityTo = dr("ValidityTo").ToString
        End If
        eOfficers.EmailId = dr("EmailId").ToString
        eOfficers.PhoneCommunication = If(dr("PhoneCommunication") Is DBNull.Value, False, dr("PhoneCommunication"))
    End If
End Sub

```

The Data Access Layer returns data from the SQL Server and loads it into the entity, where it is returned to the presentation layer to load the controls within the page. Errors are handled only within the presentation level, if an exception is triggered, handled in the presentation layer, using the Try and Catch. Errors are reported back to the user, by means of the information panel at the bottom of each page.

3.2. Coding Standards and Conventions

In all classes, the following standards and or prefixes could be found. Hereunder we elaborate on the prefixes and their meaning.

3.2.1.Entities

Prefix	Description	Example
E	The Entity for the current data object	eOfficer

3.2.2.Methods

Prefix	Description	Example
Load	Loads data into the entity	LoadOfficer(eOfficer)
Get	Returns data into a datatable	GetOfficer(DistrictId)
Save	Saves data back to the Database	Save(eOfficer)
Insert	Insert of data into the database	Insert(eOfficer)
Update	Update of data into the database	Update

3.2.3.Controls

Prefix	Description	Example
Txt	TextBox	txtLastName
b_	button	btnSave
Ddl	Drop Down List	ddlDistricts
Chk	Check box	chkLegacy
Pnl	Panel	pnlTop
Grv	Gridview	grvOfficers

3.2.4.Objects

Prefix	Description	Example
Dt	Datatable	dtDistricts
Dr	DataRow	dr*
Data	SQL Instance	data(Filldata)
I	Integer	iCount
Str	String	strLocation
B	Boolean	bIsPoverty
Dbl	Double	dblTotal
Dec	Decimal	decPremiumTotal
A	Array	aArray
Dte	Date	dteExpiry

3.2.5.Resources

Prefix	Description	Example
L_	Label test	L_LASTNAME
V_	Validation text	V_LASTNAME
B_	Button text	B_SAVE
M_	Message text	M_NOOFFICERS

*In the case of the data row, normally if only one entity is being referenced a simple dr, dr represents the data row, however in the case of a procedure using multiple tables; the prefix will be followed by the name of the entity.

3.3. Photo Storage

Photos of insurees are captured by the mobile phones and are to be transmitted to the central server by the mobile phone application. In case the mobile network coverage is sufficient, the sending of photographs is done automatically by the mobile phone application (Android). In case there is no connectivity, the photographs could be loaded onto a flash disk or any other media manually. The IMIS application has a folder structure for photo submissions and a separate folder structure for 'consumed' associated photographs.

e.g. C:\XX\YY\SubmittedPhotos
C:\XX\YY\UpdatedPhotos

At the moment of 'updated' photographs with the insure, the photograph will be transferred from the folder 'SubmittedPhotos' toward the folder 'UpdatedPhotos'. The association process is based upon the actual filename of the photo-image. The process of association takes place when we enroll insurees for the first time or when we recall the insuree page and a positive relationship could be established between the Insuree and a photograph found in the submission folder. This process will be covered in more detail later.

The following convention for the filename:

<Insurance NUMBER>_<ENROLLCODE>_<DDMMYYYY>.XXX

< Insurance NUMBER> = QR Code assigned to the insure
<DDMMYYYY> = Date of photograph taken
<ENROLLCODE> = Enrollment officers code
<XXX> = File extension of the photo image e.g. JPG, PNG etc...

Although any file size is allowed to be copied to the Central server, we should set a maximum size of 200Kb. The file format we suggest is JPEG.

4. IMIS Menu structure and Interface flow

The IMIS application consists of many individual ASPX pages which are initiated via the following controls:

- Menu buttons
- Action buttons
- Hyperlinks

Hereunder we will cover the IMIS menu structure with its submenus and actions.

4.1. Main menu



The options '**Home**' and '**Logout**' are not having sub-menus and will navigate directly toward the following pages:

- | | |
|----------------------------|---|
| [Home] | → Home.aspx |
| [Logout] | → Logout.aspx |
| [Insurance Number enquiry] | → IMIS.MASTER embedded JQuery/AJAX code |

As seen above, the main (top level) menu will also have a few sub-menus:

- Insurees and Policies
- Claims
- Administration
- Tools

4.2. Menu Insurees and Policies



The above menu will have four options which will open web pages:

- | | |
|-------------------|--------------------|
| [Families/Groups] | → FindFamily.aspx |
| [Insurees] | → FindInsuree.aspx |
| [Policies] | → FindPolicy.aspx |
| [Contributions] | → FindPremium.aspx |

4.3. Menu Claims



The above menu has three options which will open web pages:

- | | |
|--------------------------|-----------------------|
| [Health Facility Claims] | → FindClaims.aspx |
| [Review] | → ClaimOverview.aspx |
| [Batch Run] | → ProcessBatches.aspx |

4.4. Menu Administration



This menu has only one submenu as shown above:

- Price Lists

The other options open up webpages as shown below:

- | | |
|------------------------|---------------------------|
| [Products] | → FindProduct.aspx |
| [Health Facilities] | → FindHealthfacility.aspx |
| [Medical Items] | → FindMedicalItem.aspx |
| [Medical Services] | → FindMedicalService.aspx |
| [Users] | → FindUser.aspx |
| [Enrolment Assistants] | → FindOfficer.aspx |
| [Claim Administrator] | → ClaimAdministrator.aspx |
| [Payers] | → FindPayer.aspx |
| [Locations] | → Locations.aspx |

4.5. Menu Price Lists



The options Medical Items and medical Services open up the following pages:

- [Medical Items] →FindPriceListMI.aspx
- [Medical Services] →FindPriceListMS.aspx

4.6. My Profile



My profile has only one sub menu which is change password, the Change Password sub menu open up ChangePassword.aspx page

4.7. Menu Tools



This menu has no further submenu and would open pages directly.

- [Registers] →Registers.aspx
- [Policy Renewals] →PolicyRenewals.aspx
- [Feedback Prompt] →FeedbackPrompt.aspx
- [Extracts] → IMISExtracts.aspx
- [Reports] → Reports.aspx
- [Utilities] → Utilities.aspx
- [Funding] → AddFunding.aspx
- [Email Settings] →EmailSettings.aspx

5. Functional Area Description

This chapter will cover the actual screen designs and screen flow of the IMIS application. Each interface will be presented with the following:

- Actual user interface design
- Interface object name
- Menu-Link , redirections and /or Action Link(s)
- Reference to Use Cases
- Brief description of interface
- Class entities (top level)

Some features are to be found on several interfaces. These mechanisms will be described generally rather than for each interface separately.

5.1. Generic interface features

The following features could be found on several interfaces:

- Add, Edit, Delete and Save buttons
- Cancel button
- ‘Historical’ records
- Pages tab
- Data grids
- Hyperlinks
- Multi Language aspects
- Session expiry
- Mandatory fields
- General messages and current menu path

5.1.1. Add, Edit, Save and buttons

The cancel button is always found in the lower section of the screen on the right corner side. The cancel button will cancel any entry on a screen (if any) and will subsequently navigate back to the actual caller (menu or screen).

All interfaces activated under the **administration menu** (except Locations) will have the ‘add’, ‘edit’ and ‘delete’ buttons in the lower section of the screen as shown below:



The add button will generally open up a new page for the entry of a new record.

All entries are empty but the system will automatically populate entries in case only one option is available (e.g. District or product. Etc)

Similar wise the ‘edit’ button will open up a new page for but all fields are loaded with the information of the selected record.

The delete button will display a message prompting the operator to confirm the delete action. On confirmation, the record will be ‘logically’ deleted from the system by flagging the ValidityTo field with a timestamp. The ‘datagrid’ will thereafter refresh.

The interfaces for ‘family/group overview’ and ‘locations’ have the ‘add’, ‘edit’ and ‘delete’ buttons elsewhere as shown below:



- | | | |
|-------------|---|--------------------|
| Add-button | → | green '+' (plus) |
| Edit-button | → | yellow pencil |
| Delete | → | red crossed symbol |

The actions on clicking these buttons are similar as previously described for the buttons on the lower section (administration menu).

The save button most of the time is found in the lower section in the left corner as shown below:



By clicking the save button, the record will be inserted or updated after data validation passed (as peruse cases). The save button will bring you back to the calling interface.

5.1.2. User Security

User security in IMIS is built around pre-defined roles:

- Enrolment Assistant
- Manager
- Accountant
- Clerk
- Medical Officer
- Scheme Administrator
- IMIS Administrator
- Receptionist
- Claim Administrator
- Claim Contributor

At the moment of logging in, IMIS it fetch security value that will relate the operator to one or more security roles. Depending on the role(s) of the operator, menu-options and screen functionality will be enabled for use or disabled if no rights were assigned.

All actions in the system are audited by storing the UserID with the information. An audit report is available in the reports section.

5.1.3. 'Historical' records

IMIS register all its records with timestamps for the validity of the record. All records that are flagged with a ValidityTo date have been updated or deleted via the IMIS interfaces or services.

To be able to view the different versions of a record (e.g. an Insuree that had overtime several changes in its information) one would be able to click the 'Historical' checkbox. By clicking this box, the datagrid will prevail all versions of the records and will strike-out the records that are not current. In this 'historic' mode one cannot change information but is able to view the information by clicking the view button in the lower section of the screen. The full record contents of the 'historic' record will be shown.

See below an example of a datagrid including historical records.

NAME	DATE	DISTRICT	VALID FROM	VALID TO
Dispensaries	01/01/2016		01/01/2015	
Dispensaries Msalala	06/11/2016	Msalala	07/11/2016	
Dispensary	06/11/2016	Msalala	07/11/2016	07/11/2016
Dispensary Msalala	06/11/2016	Msalala	07/11/2016	07/11/2016
Health Centers	01/01/2016		01/01/2015	
Health Centers Msalala	06/11/2016	Msalala	07/11/2016	
Hospital above 25 beds	01/01/2016	-	26/10/2016	26/10/2016
Hospital above 25 beds	01/01/2016	-	26/10/2016	26/10/2016
Hospital above 25 beds	01/01/2016	-	26/10/2016	26/10/2016
Hospital above 25 beds	01/01/2016	-	26/10/2016	26/10/2016
Hospital above 25 beds	01/01/2016		01/01/2015	
Hospitals above 100 beds	01/01/2016		10/11/2016	
Hospitals below 25 beds	01/01/2016		01/01/2015	
Hospitals Msalala	06/11/2016	Msalala	07/11/2016	

5.1.4.Data grids, hyperlinks and pages

As seen in the previous chapter (menus), most of the menu options in IMIS will open up search interfaces. Each search interface will display the result of the search action (after clicking the search button) within a data grid.

Depending on the expected data load, data is already populated in the grids whilst other search screens will only provide information in the grid after setting the criteria and clicking 'search'. For example, the health facilities search screen will automatically display all health facilities available applicable to the current user. The Insuree search page would initially not show any data in the grid as too many records would be unnecessarily loaded.

In most of the search screens one can find in the bottom of the screen a navigation bar for pages as shown below:

HF CODE	NAME	LEGAL FORM	LEVEL	TYPE	PHONE NUMBER	DISTRICT	ID	VALID FROM	VALID TO
00997	Wiyenzele	Government	Dispensary	Out-Patient		Mpwapwa	245	17/05/2013	
00998	Seluka	Government	Dispensary	Out-Patient		Mpwapwa	246	17/05/2013	
00999	Chinyanguku	Government	Dispensary	Out-Patient		Mpwapwa	247	17/05/2013	
03606	Mkoya	Government	Dispensary	Out-Patient	0754033584	Morogoro MC	1086	10/11/2015	
03607	Kibwe	Government	Dispensary	Out-Patient	0624206329	Morogoro MC	2473	23/04/2016	
03608	Legezamwendo	Government	Dispensary	Out-Patient	0624206547	Morogoro MC	2474	23/04/2016	
03609	Morogoro Regional Hospital	Government	Hospital	Both	0624206359	Morogoro MC	1070	23/04/2016	
03610	Vituli	Government	Dispensary	Out-Patient	0624206572	Morogoro MC	2472	23/04/2016	
03611	Luhungo	Government	Dispensary	Out-Patient	0624206716	Morogoro MC	2479	23/04/2016	
03612	Sabasaba	Government	Health Centre	Both	0787141460/ 0718842076	Morogoro MC	1072	03/11/2015	
03613	Uhuru	Government	Hospital	Both	0719666664	Morogoro MC	1071	03/11/2015	
03614	Mafiga	Government	Health Centre	Both	0714619411/0753893177	Morogoro MC	1073	03/11/2015	
03615	Kingolwira	Government	Health Centre	Both	0624205557	Morogoro MC	1074	23/04/2016	
03616	Konga	Government	Dispensary	Out-Patient	0717833500/0754755010	Morogoro MC	1085	03/11/2015	
04139	Iringa Mvumi	Government	Dispensary	Out-Patient		Chamwino	188	07/09/2016	

In the example above, 15 facilities (rows) are shown in the datagrid. However, there are many pages available with health facilities that match the search criteria. To see more facilities one could click on any of the pages available or navigating to the last or first page. After clicking on another page number, the grid will reload a new set of entries that satisfied the search criteria.

827 Facilities Found									
HF CODE	NAME	LEGAL FORM	LEVEL	TYPE	PHONE NUMBER	DISTRICT	ID	VALID FROM	VALID TO
00997	Wiyenzele	Government	Dispensary	Out-Patient		Mpwapwa	245	17/05/2013	
00998	Seluka	Government	Dispensary	Out-Patient		Mpwapwa	246	17/05/2013	
00999	Chinyanguku	Government	Dispensary	Out-Patient		Mpwapwa	247	17/05/2013	
03606	Mkoya	Government	Dispensary	Out-Patient	0754033584	Morogoro MC	1086	10/11/2015	
03607	Kibwe	Government	Dispensary	Out-Patient	0624206329	Morogoro MC	2473	23/04/2016	
03608	Legezamwendo	Government	Dispensary	Out-Patient	0624206547	Morogoro MC	2474	23/04/2016	
03609	Morogoro Regional Hospital	Government	Hospital	Both	0624206359	Morogoro MC	1070	23/04/2016	
03610	Vituli	Government	Dispensary	Out-Patient	0624206572	Morogoro MC	2472	23/04/2016	
03611	Luhungo	Government	Dispensary	Out-Patient	0624206716	Morogoro MC	2479	23/04/2016	
03612	Sabasaba	Government	Health Centre	Both	0787141460/ 0718842076	Morogoro MC	1072	03/11/2015	
03613	Uhuru	Government	Hospital	Both	0719666664	Morogoro MC	1071	03/11/2015	
03614	Mafiga	Government	Health Centre	Both	0714619411/0753893177	Morogoro MC	1073	03/11/2015	
03615	Kingolwira	Government	Health Centre	Both	0624205557	Morogoro MC	1074	23/04/2016	

All the data grids with a yellow (highlighted) row for indicating your current mouse position in the grid. By moving the mouse up and down the yellow row will follow the mouse pointer. By clicking a row, the row gets selected. The selected row is always indicated by the color blue. In case we need to modify/view or delete a record, one should first select the row by clicking on it (marking it blue) and thereafter clicking the desired action (Edit, Delete, and View)

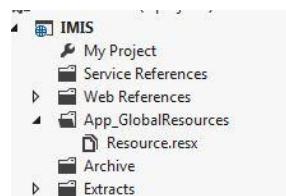
Most grids have the first column as a Hyperlink. Instead of selecting the record (marking blue) and clicking the desired action button, one could alternatively click directly on the hyperlink to 'Edit' the ('clicked on') record.

Some columns, such as expiry date, valid from and to, will have sort order mechanisms built in to allow the operator to sort the grid contents ascending or descending. The sort order change will be performed after clicking the column header.

5.1.5. Multi Language aspects

IMIS developed in multi-language: English and the other language which will be provided with IMIS Administration. The manner of changing from one language to another is depending on the user profile. The default language will be English. English will be an Interface Culture used for loading labels (menu and pages), messaging, grid headers etc...

The actual language elements are kept in so called resource files. Each individual language element can be found in these resource files.



As seen above, we currently only have one resource file included in the project: Resource.resx (default English). Additional languages could be added by simply adding an extra resource file with the correct translations and allowing the user to select this new language via the interface of the user definition.

Each resource file has a similar structure and holds all translated elements. See below 2 separate snippets of the resource file for Swahili.

L_OTHERNAMES	Majina mengine
L_OUTPATIENT	Mgonjwa aliyetoka
L_PASSPORT	Namba ya pasipoti
L_PASSWORD	Namba ya kuingilia
L_PAYER	Walipaji
L_PHONE	Namba ya simu

The above section is an example of labels, starting with 'L_'.

The section below shows some translation for validation messages starting with 'V_'

V_GENDER	Naomba uchague jinsia
V_LASTNAME	Naomba uingie kwenye jina la mwisho
V_MARITAL	Naomba uchague aina ya jinsia
V_OTHERNAMES	Naomba uingie majina mengine

Also for data driven some of drop down has been added with the multi language scenario example of the drop down is Relation, the figure below show the table structure where the data stored. On selection the system will select according to user language

RelationId	Relation	SortOrder	AltLanguage
1	Father	1	Baba
2	Brother	2	Kaka
3	Sister	3	Dada
4	Cousin	4	Binamu
5	Uncle	5	Mjomba

The process of obtaining the correct translation from resource is at HTML code level in each page as shown below for the label and validation text for the field 'other names'.

```

<td class="FormLabel">
<asp:Label
ID="L_OTHERNAMES"
runat="server"
Text='<%$ Resources:Resource,L_OTHERNAMES %>'>
</asp:Label>
</td>
<td class ="DataEntry">
<asp:TextBox ID="txtOtherNames" runat="server" Width="150px">
</asp:TextBox>
</td>
<td>
<asp:RequiredFieldValidator
ID="RequiredFieldOtherNames" runat="server"
ControlToValidate="txtOtherNames"
SetFocusOnError="True"
ValidationGroup="check"
Text='<%$ Resources:Resource,V_OTHERNAMES %>'>
</asp:RequiredFieldValidator>
<asp:RegularExpressionValidator ID="RegularExpressionValidator2" runat="server"
ControlToValidate="txtOtherNames" ErrorMessage="Only alphabets are allowed."
ValidationExpression="^([a-zA-Z]+)$" ValidationGroup="check"></asp:RegularExpressionValidator>
</td>

```

5.1.6.Session expiry

The session expiry has been set at 20 minutes as a default. However, this can be changed to another time interval as required via the web.config file and IIS configuration.

If a session expires the operator will have to login once again.

5.1.7.Mandatory fields

While working with an entry (adding/editing) screens, the operator will see directly which data is incomplete or not correctly entered. The message is shown on the screen on the right side of the actual control in 'red' marked text. This text will automatically disappear at the moment the text is entered.

This first level validation takes place in the presentation tier on the local machine (client side). However more validations will take place in the Business Logic Layer in private functions.

An example of client side validation is shown below:

The screenshot shows a web application interface with a navigation bar at the top containing links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance' is also present. The main content area is divided into two sections: 'Family/Group Details' and 'Policy Holder'. The 'Policy Holder' section contains various input fields for personal information, many of which are marked with a red asterisk (*) indicating they are mandatory. These fields include Insurance Number, Other Names, Last Name, Birth Date, Gender, Marital Status, Beneficiary Card, Current District, Current Municipality, Current Village, Current Address Details, and Profession. At the bottom of the form are 'Save' and 'Cancel' buttons.

5.1.8. Status bar and Popup

The ‘purple’ status bar on the bottom of the screen will act as a message bar that will always show the current menu-route. For example if one would have clicked the option “Facilities” under the Administration menu and thereafter selected a health facility for editing, the status bar will show: Administration-Health facilities-Modify Health facility.

The status bar will also be used for popup messages such as ‘Saved successfully’ these messages will disappear within 10 seconds. The menu path will again, thereafter, be shown in this status bar. In case of more important messages, IMIS will provide popup messages.

5.2. Web pages

The presentation Tier of IMIS will consist of several web pages (aspx pages) that will contain different programming techniques such as:

- HTML
- AJAX
- JQUERY
- JSCRIPT

The web pages are based upon Style sheets. This will enforce uniformity throughout the interfaces and will simplify design changes in future amendments. The web.config file will host the connection string for connectivity to the database and will also be used to host certain application defaults such as folder locations for photographs and any interfacing with external applications such as the mobile phone (file transfers) and EPICOR.

In the following paragraphs we will cover the interfaces:

- AddFunding.aspx
- ChangeFamily.aspx
- ChangePassword.aspx
- Claim.aspx
- ClaimAdministrator.aspx
- claimfeedback.aspx
- ClaimOverView.aspx
- ClaimReview.aspx
- Default.aspx
- Download.aspx
- EmailSettings.aspx
- Error.html
- FaindClaim.aspx
- Family.aspx
- FeedbackPrompt.aspx
- FindClaimAdministrator.aspx
- FindFamily.aspx
- FindHealthFacility.aspx
- FindInsuree.aspx
- FindMedicalItem.aspx
- FindMedicalService.aspx
- FindOfficer.aspx
- FindPayer.aspx
- FindPolicy.aspx
- FindPremium.aspx
- FindPriceListMI.aspx
- FindPriceListMS.aspx
- FindProduct.aspx
- FindUser.aspx

- ForgotPassword.aspx
- General.vb
- Global.asax
- HealthFacility.aspx
- Home.aspx
- IMIS.MASTER (used for Menu and Quick Inquiry)
- Imis_Gen.vb
- IMISExtracts.aspx
- Insuree.aspx
- Locations.aspx
- Logout.aspx
- MedicalItem.aspx
- MedicalService.aspx
- MoveLocations.aspx
- Nojs.html
- Officer.aspx
- OverviewFamily.aspx
- Payer.aspx
- Policy.aspx
- PolicyRenewals.aspx
- Premium.aspx
- PremiumCollection.aspx
- pREMIUMdistribution.aspx
- PriceListMI.aspx
- PriceListMS.aspx
- ProcessBatches.aspx
- Product.aspx
- Redirect.ASPX
- Redirect.htm
- Report.aspx
- Reports.aspx
- Registers.aspx
- User.aspx
- Utilities.aspx
- WebConfig

5.2.1.Default.aspx

Interface object name:	<i>Default.aspx</i>
Menu path:	<i>Logout</i>
Hyperlinks/Redirections:	<i>Opens when IMIS Starts</i>
Action Button:	
Use case Reference	<i>N/A</i>
Class Diagram	<pre> classDiagram class LoginBI { <<Class>> <<Methods>> +getControlsSettings() +GetDefaults() +getHFCCodeAndName() +GetLogin() +InsertLogTime() +SaveOfflineHFID() } </pre>

Brief description:

This interface opens up when the user starts IMIS. A similar design will be used for the Login on mobile phones in case we use web-pages within the phone App. After successful login procedure the user is taken to the home page. Further to this, at this moment we will switch the language to Swahili or English using the so called Interface Cultures within the application and making use of the required resource file (English/Swahili). For the mobile phone only English will be used.

5.2.2.ForgotPassword.aspx

Enter email id to retrieve password

Email:

[Go to Login](#)

Submit

Interface object name:	<i>ForgotPassword.aspx</i>
Menu path:	N/A
Hyperlinks/Redirections:	<i>Open after click Forgot password</i>
Action Button:	<i>Submit Email address</i>
Use case Reference	N/A
Class Diagram	 <pre> classDiagram class ForgotPasswordBI { <<Class>> <<Methods>> <<SendPassword>> } </pre>

Brief description:

This interface opens up when the user click Forgot password button in default.aspx page. The interface used to recover user login password, user has to submit the email address then the system will verify the email and send the correct password.

5.2.3.ChangePassword.aspx

The screenshot shows the 'Change Password' interface. The top navigation bar includes links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar is also present. The main content area is titled 'Change Password' and contains three text input fields: 'Current Password', 'New Password', and 'Confirm Password'. At the bottom of the form are two buttons: 'Save' and 'Cancel'.

Interface object name:	ChangePasswors.aspx
Menu path:	Change Password (Under My Profile Menu)
Hyperlinks/Redirections:	N/A
Action Button:	N/A
Use case Reference	N/A
Class Diagram	<pre> classDiagram class ChangePasswordBI { <<Methods>> ChangePassword LoadUsers } </pre>

Brief description:

This interface opens up when the user click Change Password. The interface used to change user password by enter the current password and the new password.

5.2.4. Home.aspx

Good Afternoon Exact Software v17.3.7

Roles

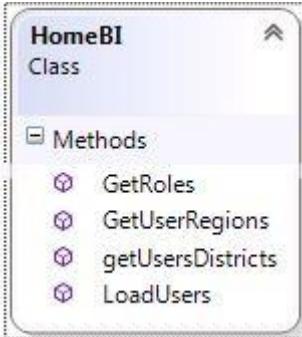
- Enrolment Officer
- Manager
- Accountant
- Clerk
- Medical Officer
- Scheme Administrator
- IMIS Administrator
- Receptionist
- Claim Administrator
- Claim Contributor

Region

- North West
- Region
- South West
- District
- Dummy
- Kongwa
- Bafili
- Dodoma
- Mpapwa
- Basel
- Kilosa
- Kailali
- Msalala
- Bamenda
- Bambui
- Kumbo
- Buea
- Mamfe
- Morogoro

© Swiss Agency for Development and Cooperation
distributed under a royalty-free license
by courtesy of the copyright owner

IMIS

Interface object name:	<i>Home.aspx</i>
Menu path:	<i>Home</i>
Hyperlinks/Redirections:	<i>Opens after logging in successfully</i>
Action Button:	<i>Initiated via all cancel buttons on find screens(called via top menu calls)</i>
Use case Reference	N/A
Class Diagram	 <pre> classDiagram class HomeBI { <<Methods>> GetRoles GetUserRegions getUsersDistricts LoadUsers } </pre>

Brief description:

This interface opens up when the user has successfully logged in. The menu will show in the language of the user.

5.2.5.Logout.aspx

NO DESIGN APPLICABLE

Interface object name:	<i>Logout.aspx</i>
Menu path:	<i>Logout</i>
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	N/A
Class Diagram	N/A

Brief description:

This interface opens up when the user has clicked the Logout option on the top menu. The open sessions will be terminated and this page redirects automatically to the Login page: Defaults.aspx

5.2.6.Family.aspx

The screenshot shows the 'Family.aspx' page with the following details:

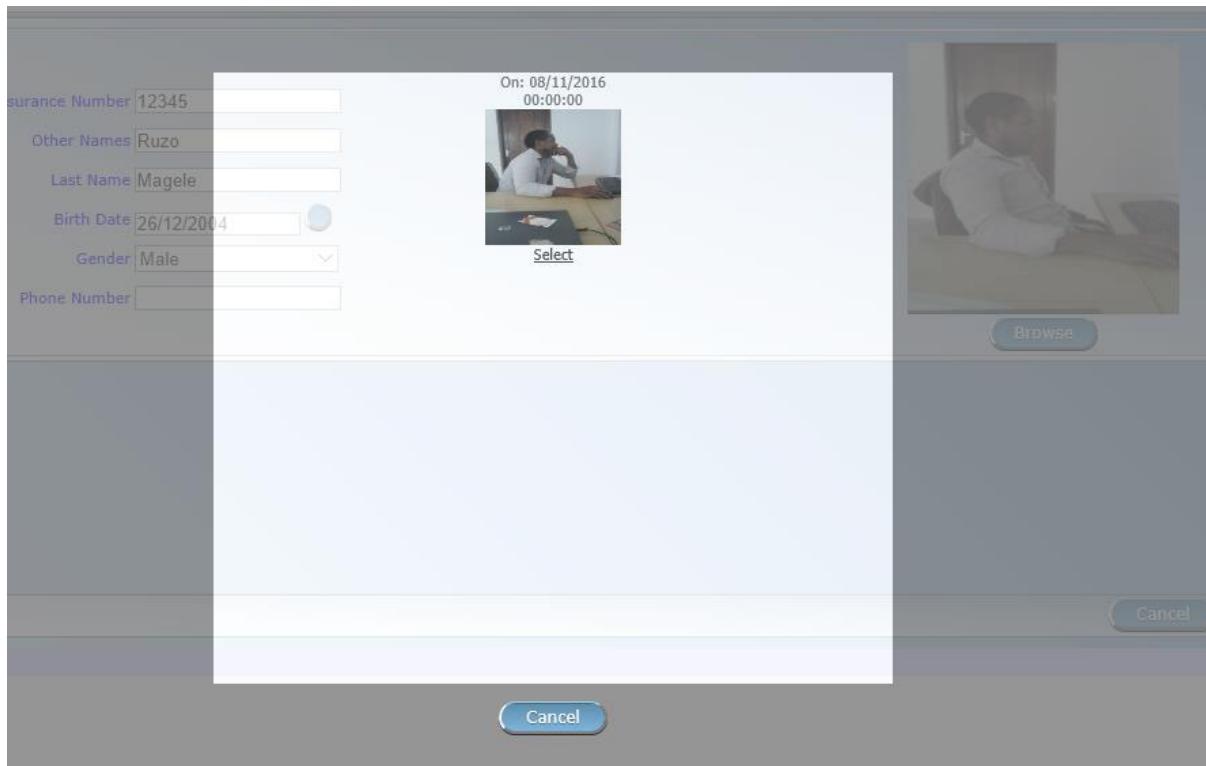
- Family/Group Details:**
 - District: Bahi
 - Municipality: Chali
 - Village: -- Select a Village --
 - Poverty Status: -- Select Yes/No --
 - Confirmation Type: -- Select Type --
 - Group Type: -- Select Type --
 - Permanent Address Details: (Empty input field)
 - Confirmation No.: (Empty input field)
- Policy Holder:**
 - Insurance Number: (Input field)
 - Other Names: (Input field)
 - Last Name: (Input field)
 - Birth Date: (Input field)
 - Gender: -- Select Gender --
 - Marital Status: -- Select Status --
 - Beneficiary Card: -- Select Yes/No --
 - Current District: -- Select a District --
 - Current Municipality: (Input field)
 - Current Village: (Input field)
 - Current Address Details: (Input field)
 - Profession: --Select Profession--
 - Browse: (File upload button)
- Action Buttons:**
 - Save: (Blue button)
 - Cancel: (Blue button)

Interface object name:	Family.aspx
Menu path:	Add Family/Group
Hyperlinks/Redirections:	
Action Button:	OverviewFamily.aspx (Add button in family/group section)
Use case Reference	5.2.1

Class Diagram	<p>The screenshot shows a class diagram for the 'FamilyBI' class. The class is categorized under 'Class'. It has a section for 'Methods' which lists the following methods:</p> <ul style="list-style-type: none">CheckCHFIDExtractCHFIDExtractDateExtractLatitudeExtractLongitudeExtractOfficerCodeFamilyExistsGetCurDistrictsGetCurVillagesGetCurWardsGetDistricts (+ 1 overload)GetDistrictsAllGetEducationGetEthnicityGetFSPHFGetGenderGetHFLevelGetMaritalStatusgetOfficerIDGetProfessionGetRegionsGetRegionsAllGetSubsidyGetTypeOfIdentityGetTypesGetVillagesGetWardsGetYesNOLoadFamilySaveFamily
----------------------	--

Brief description:

This interface opens up when the user has clicked the Add family/group button option on the top menu (Insurees and Policies). This interface is used to add new families into IMIS with the Head of the family/group. When entering the Insurance Number automatically the system queries the folder for submitted photos and will try to associate the photo by using the photo naming convention (discussed later). In case one photo is available the photo will be automatically presented as shown in the interface above. In case there would be more than one photograph, a browser window will open (shown below) and the operator could choose the correct photo.



When clicking the 'save' button first the family/group record is inserted and subsequently the insuree record (as head of family/group). Furthermore, the photograph will be transferred to a new folder for uploaded photos and a record will be inserted in the photo table (if association was successful).

After all saving processes are completed, the user will be directed to the page for the family/group overview as shown below for further adding dependants and/or policies/contributions:

INSURANCE NUMBER	LAST NAME	OTHER NAMES	GENDER	BIRTH DATE	BENEFICIARY CARD
77777770	Foster	James	M	26/12/1988	<input checked="" type="checkbox"/>
77777771	Foster	Jane	F	29/01/1980	<input checked="" type="checkbox"/>
77777772	Foster	Bill	M	28/01/2014	<input type="checkbox"/>
77777773	Foster	Elly	F	10/06/2014	<input checked="" type="checkbox"/>

ENROL DATE	EFFECTIVE DATE	START DATE	EXPIRY DATE	PRODUCT	ENROLMENT OFFICER	POLICY STATUS	POLICY VALUE	VALID FROM	VALID TO
11/12/2016	01/01/2017	01/01/2017	31/12/2017	SFWS001	Matonya David	Active	34,000.00	11/12/2016	
18/11/2015	01/01/2016	01/01/2016	31/12/2016	SFWS001	Matonya David	Active	46,000.00	11/12/2016	

PAYMENT DATE	PAYER	AMOUNT	PAYMENT TYPE	RECEIPT NO.	CONTRIBUTION CATEGORY
09/10/2016		34,000.00	Cash	rc2	Contribution

5.2.7.FindFamily.aspx

The screenshot shows the 'FindFamily.aspx' page with the following interface:

- Header:** Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, Search Insurance No, a play button, and a help icon.
- Select Criteria:** A form with fields for Last Name, Other Names, Region, Insurance Number, Birth Date From, District, Phone Number, Birth Date To, Municipality, Gender, Poverty Status, Village, Email, Confirmation No., and a 'Search' button. There is also a 'Historical' checkbox.
- Results:** A table titled '61 Families/Groups Found' with columns: INSURANCE NUMBER, LAST NAME, OTHER NAMES, REGION, DISTRICT, MUNICIPALITY, VILLAGE, POVERTY, VALID FROM, and VALID TO. The table lists 61 rows of family/group data.
- Pagination:** A navigation bar at the bottom of the results table showing pages 1 through 5.
- Buttons:** A 'Cancel' button at the bottom right of the page.

Interface object name:	FindFamily.aspx
Menu path:	Families
Hyperlinks/Redirections:	
Action Button:	OverviewFamily.aspx (Cancel button),
Use case Reference	5.2.5
Class Diagram	<pre> classDiagram class FindFamilyBI { <<Methods>> checkRoles() GetDistricts() GetFamily() GetGender() GetMaritalStatus() GetPoverty() GetRegions() GetSubsidy() GetVillages() GetWards() } </pre>

Brief description:

This interface opens up when the user has clicked the Families button option on the top menu (Insurees and Policies). This interface is used to search for families. The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. The Insurance Number hyperlink (Head of Family/Group) in the grid will open up the family/group overview page. Further operations will take action from this overview page.

5.2.8.ChangeFamily.aspx

The screenshot shows the 'ChangeFamily.aspx' page with three main tabs:

- Change Family/Group:** Contains fields for District (Kongwa), Municipality (Sagara), Village (Moleti), Poverty Status (Yes), Confirmation Type (dropdown), and Confirmation No. (text input).
- Change Head of Family/Group:** Contains a field for Enter the new Head of Family/Group Insurance Number and a Check button.
- Move Insurees:** Contains a field for Enter the Insurance Number of Insuree to move and a Check button.

Interface object name:	<i>ChangeFamily.aspx</i>
Menu path:	
Hyperlinks/Redirections:	
Action Button:	<i>OverviewFamily.aspx (Edit button in family/group section)</i>
Use case Reference	5.2.9 (<i>Modify Family/Group</i>) 5.2.13 (<i>Change Head of Family/Group</i>) 5.2.14 (<i>Move Insuree</i>)
Class Diagram	<pre> classDiagram class ChangeFamilyBI { <<Methods>> ChangeHead GetDistricts GetEthnicity GetFamilyHeadInfo GetInsureesByCHFID GetMaxMemberCount GetRegions GetSubsidy GetTypes GetVillages GetWards GetYesNO LoadFamily MoveInsuree SaveFamily UpdateChangeFamily } </pre>

Brief description:

This interface opens up when the user has clicked the Edit button (family/group section) in the family/group overview page.

The operator can use this interface for updating the Family/Group information, changing the head of Family/Group and moving an Insuree toward the current family/group selected.

The Insurance Number to be set as the new head of family/group or to be moved toward the current selected family/group will be entered into the Insurance Number fields. By clicking the 'Change' or 'Move' button, the insuree will subsequently, after validating as per 'use-cases', be inserted in the insuree table (archiving) then updated (current record). Also the family table (change head of family/group) will be updated. (For audit record purposes only).

5.2.9.FindInsuree.aspx

The screenshot shows the 'FindInsuree.aspx' page. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, and a search bar labeled 'Search Insurance No' with a dropdown arrow and a magnifying glass icon. Below the navigation is a 'Select Criteria' section with fields for Last Name, Other Names, Region, Insurance Number, Birth Date From, District, Phone Number, Birth Date To, Municipality, Email, Gender, Marital Status, Photo Assigned (with an 'All' option), and Village. A checkbox labeled 'Historical' is also present. A 'Search' button is located at the bottom right of this section. Below this is a header '110 Insurees Found'. A table displays 110 rows of insuree data, each with a unique ID, last name, other names, marital status, gender, phone number, date of birth, region, district, valid from, and valid to. The last row, which has a yellow background, corresponds to the highlighted row in the table above. At the bottom of the table is a navigation bar with numbers 1 through 8. A 'Cancel' button is located at the bottom right of the page.

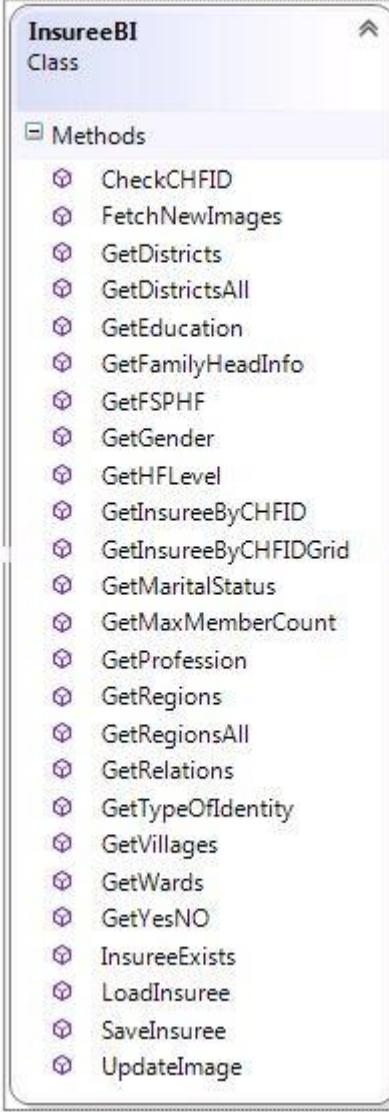
INSURANCE NUMBER	LAST NAME	OTHER NAMES	MARITAL STATUS	GENDER	PHONE NUMBER	DATE OF BIRTH	REGION	DISTRICT	VALID FROM	VALID TO
111111111	Gatsby	John	M	M		01/01/1979	TestRegion	TestDistrict1	09/05/2017	
111111112	Gatsby	Elis	M	F	234567890	30/01/1990	TestRegion	TestDistrict1	09/05/2017	
111111121	Bock	John	M	M	+420234567567	12/06/1984	TestRegion	TestDistrict1	10/05/2017	
111111122	Bock	Elly	M	F	+4358990000	12/06/1984	TestRegion	TestDistrict1	10/05/2017	
111111123	Bock	James	S	M		11/06/2014	TestRegion	TestDistrict1	10/05/2017	
111111124	Bock	Julia	S	F		29/12/2008	TestRegion	TestDistrict1	10/05/2017	
111111125	Bockwurst	Peter	S	M		01/01/1999	TestRegion	TestDistrict1	10/05/2017	
00001	Soni	Hiren		M		27/01/1983	TestRegion	TestDistrict1	15/05/2017	
000011	Soni	Nisha		F		07/08/1984	TestRegion	TestDistrict1	15/05/2017	
000012	Soni	Pooja		F		23/07/1985	TestRegion	TestDistrict1	15/05/2017	
000013	Hindocha	Dhyey		F		01/06/2006	TestRegion	TestDistrict1	15/05/2017	
00002	Obed	Rogers		M		13/05/1991	TestRegion	TestDistrict1	17/05/2017	
111111131	Glenn	Gilmour	S	M		30/01/1990	TestRegion	TestDistrict1	17/05/2017	
111111132	Glenn	Julia	M	F	123456789	30/01/1990	TestRegion	TestDistrict1	17/05/2017	
111111133	Glenn	John	S	M		28/12/2008	TestRegion	TestDistrict1	17/05/2017	

Interface object name:	FindInsuree.aspx
Menu path:	Insurees
Hyperlinks/Redirections:	
Action Button:	OverviewFamily.aspx (Cancel button)
Use case Reference	5.2.6
Class Diagram	<pre> classDiagram class FindInsureeBI { <<Methods>> checkRoles FindInsuree GetDistricts GetGender GetMaritalStatus GetPhotoAssig ... GetRegions GetVillages GetWards } </pre>

Brief description:

This interface opens up when the user has clicked the Insurees button option on the top menu (Insurees and Policies). This interface is used to search for Insurees. The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. The Insurance Number hyperlink (Insuree) in the grid will open up the family/group overview page. Further operations on the insuree will take action from this overview page.

5.2.10. Insuree.aspx

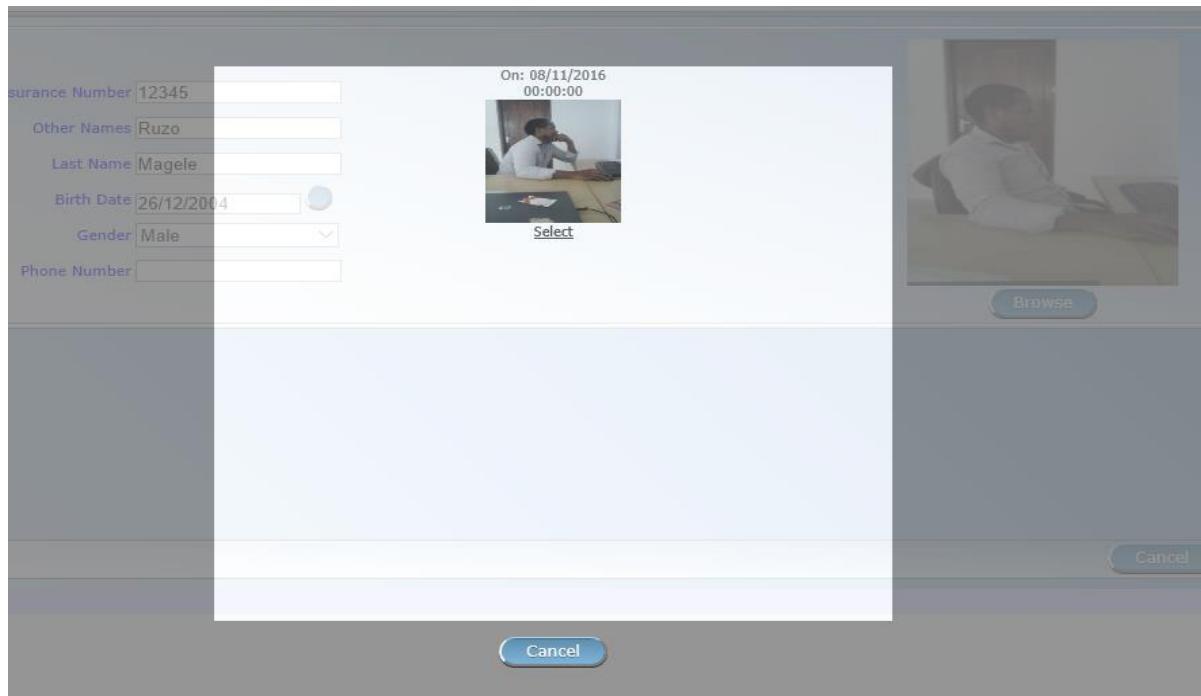
Interface object name:	<i>Insuree.aspx</i>
Menu path:	
Hyperlinks/Redirections:	<i>Insurance Number column in Insurees section in OverviewFamily.aspx</i>
Action Button:	<i>OverviewFamily.aspx (Add and Edit button in Insuree section)</i>
Use case Reference	5.2.2 (Add insure) 5.2.10 (Modify Insuree)
Class Diagram	 <pre> classDiagram class InsureeBI { <<Methods>> CheckCHFID FetchNewImages GetDistricts GetDistrictsAll GetEducation GetFamilyHeadInfo GetFSPHF GetGender GetHFLevel GetInsureeByCHFID GetInsureeByCHFIDGrid GetMaritalStatus GetMaxMemberCount GetProfession GetRegions GetRegionsAll GetRelations GetTypeOfIdentity GetVillages GetWards GetYesNO InsureeExists LoadInsuree SaveInsuree UpdateImage } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the family/group overview (Insuree section) . The insuree information can be changed as per defined ‘use-cases’ for adding and editing an Insuree.

When adding a new Insuree, after entering Insurance Number automatically the system queries the folder for submitted photos and will try to associate the photo by using the photo naming convention (discussed later). In case one photo is available the photo will be automatically presented as shown in the interface above. In case there would be more than one photograph, a browser window will open (shown below) and the operator could choose the correct photo.

The browse button allows viewing all submitted photographs. The following screen will appear:



One could select a new Photo by clicking the 'select' button under the photograph. After selecting the desired photo, it will be reloaded in the page.

On saving, the photo will be associated with the selected Insuree. The cancel button will bring you back without any changes to the photograph holder.

On saving an insuree, he/she will be added /refreshed on the current insurees section within the family/group overview page.

5.2.11. FindPolicy.aspx

The screenshot shows a web-based application for managing policies. At the top, there is a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. There is also a search bar labeled "Search Insurance No" with a dropdown arrow and a magnifying glass icon.

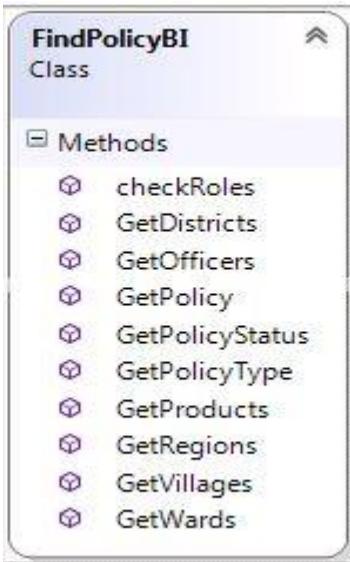
The main area is titled "Select Criteria" and contains several input fields for filtering policies:

- Enrolment Date From: [text box] [calendar icon]
- Effective Date From: [text box] [calendar icon]
- Region: [dropdown menu] --Select Region--
- Enrolment Date To: [text box] [calendar icon]
- Effective Date To: [text box] [calendar icon]
- District: [dropdown menu]
- Start Date From: [text box] [calendar icon]
- Expiry Date From: [text box] [calendar icon]
- Type: [dropdown menu] -- Select Type -- [checkbox] Historical
- Start Date To: [text box] [calendar icon]
- Expiry Date To: [text box] [calendar icon]
- Inactive Insurees: [checkbox]
- Enrolment Officer: [dropdown menu] -- Select Enrolment C
- Product: [dropdown menu] --Select Product-
- Balance: [text box]
- Policy Status: [dropdown menu] -- Select Status --
- Search: [button]

Below the search criteria, a message "68 Policies Found" is displayed above a table. The table has columns for ENROL DATE, NAME, EFFECTIVE DATE, START DATE, EXPIRY DATE, PRODUCT, ENROLMENT OFFICER, POLICY STATUS, POLICY VALUE, BALANCE, TYPE, VALID FROM, and VALID TO. The data in the table is as follows:

ENROL DATE	NAME	EFFECTIVE DATE	START DATE	EXPIRY DATE	PRODUCT	ENROLMENT OFFICER	POLICY STATUS	POLICY VALUE	BALANCE	TYPE	VALID FROM	VALID TO
17/07/2017	Mcharo Prisca		17/07/2017	16/07/2018	RF01	Shark John	Idle	11,000.00	8,000.00	N	17/07/2017	
15/07/2017	Tusk Jean	15/07/2017	15/07/2017	14/07/2018	RX001	Enrolment1 John	Active	17,500.00	0.00	N	15/07/2017	
15/07/2017	Pappen Jane	15/07/2017	15/07/2017	14/07/2018	RX001	Enrolment1 John	Active	18,000.00	0.00	N	15/07/2017	
15/07/2017	Test One	15/07/2017	15/07/2017	14/07/2018	RX001	Black James	Active	17,000.00	0.00	N	18/07/2017	
15/07/2017	Test One	15/07/2017	15/07/2017	14/07/2018	RX001	Enrolment2 Jack	Active	17,000.00	0.00	N	19/07/2017	
15/07/2017	John Timoth	15/07/2017	15/07/2017	14/07/2018	RX001	Enrolment1 John	Active	17,500.00	0.00	N	01/08/2017	
13/07/2017	Ngaiza Gracious	13/07/2017	01/07/2017	30/06/2018	DX01	Park Jack	Active	5,000.00	5,000.00	N	13/07/2017	
13/07/2017	Kiswaga Ayubu		01/10/2017	30/09/2018	RX01	Beck John	Idle	10,000.00	8,000.00	N	13/07/2017	
12/07/2017	Gonzalez Quido	12/07/2017	12/07/2017	11/07/2018	RF01	Black James	Active	10,000.00	0.00	R	12/07/2017	
11/07/2017	D A	11/07/2017	11/07/2017		RF01	Fox James	Idle	11,000.00	11,000.00	N	11/07/2017	
11/07/2017	D A	11/07/2017	11/07/2017		RF01	Fox James	Idle	11,000.00	11,000.00	N	11/07/2017	
10/07/2017	Galan Dick	10/07/2017	10/07/2017		RF01	Fox James	Active	11,000.00	0.00	N	10/07/2017	
10/07/2017	Mattel Dave	10/07/2017	10/07/2017	10/07/2018	DF01	Fox James	Active	45,000.00	0.00	N	10/07/2017	
08/07/2017	Bock John		08/07/2017	07/07/2018	TT01	Fox James	Idle	8,000.00	8,000.00	N	08/07/2017	
05/07/2017	Jo Jo		05/07/2017	04/07/2018	111	Black James	Idle	0.00	0.00	N	05/07/2017	

At the bottom of the table, there is a navigation bar with pages 1, 2, 3, 4, 5, and a "Cancel" button.

Interface object name:	<i>FindPolicy.aspx</i>
Menu path:	<i>Policies</i>
Hyperlinks/Redirections:	
Action Button:	<i>OverviewFamily.aspx (Cancel button)</i>
Use case Reference	5.2.7
Class Diagram	 <pre> classDiagram class FindPolicyBI { <<Methods>> checkRoles GetDistricts GetOfficers GetPolicy GetPolicyStatus GetPolicyType GetProducts GetRegions GetVillages GetWards } </pre>

Brief description:

This interface opens up when the user has clicked the Policies button option on the top menu (Insurees and Policies). This interface is used to search for Policies. The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Enrollment Date) in the grid will open up the family/group overview page. Further operations on the policy will take action from this overview page.

5.2.12. Policy.aspx

The screenshot shows the 'Policy.aspx' page with the following details:

- Family/Group Details:**
 - Insurance Number: 00100109
 - Last Name: Mcharo
 - Other Names: Prisca
 - Poverty Status: Not specified
 - Region: TestRegion
 - District: TestDistrict1
 - Municipality: TestMunicipality11
 - Village: TestVillage11
 - Confirmation Type: Not specified
 - Confirmation No.: Not specified
 - Permanent Address Details: Not specified
- Policy Details:**
 - Enrolment Date: 17/07/2017
 - Product: RF01
 - Effective Date: Not specified
 - Start Date: 17/07/2017
 - Expiry Date: 16/07/2018
 - Enrolment Officer: E002 - Shark John
 - Policy Status: Idle
- Financial Summary:**

Policy Value	Contribution Paid	Balance
11000.00	3000.00	8000.00

General	In-Patient	Out-Patient
Deductible: 0.00	0.00	0.00
Remunerated Health Care: 0.00	0.00	0.00

At the bottom, there are 'Save' and 'Cancel' buttons.

Interface object name:	Policy.aspx
Menu path:	
Hyperlinks/Redirections:	<i>Enroll date column in Policy section in OverviewFamily.aspx</i>
Action Button:	OverviewFamily.aspx (Add and Edit button in Policy section)
Use case Reference	5.2.3 (Add Policy) 5.2.11 (Modify Policy)
Class Diagram	<pre> classDiagram class PolicyBI { <<Methods>> +getEnrollmentOfficerMoved() +GetFamilyHeadInfo() +GetOfficers() +GetPeriodForPolicy() +getPolicyValue() +GetProductAdministrationPeriod() +getProductDetailMin() +GetProductDetails() +GetProductForRenewal() +GetProducts() +isExceededAdherents() +IsRenewalLate() +LoadPolicy() +LoadPolicyDedRem() +ReturnPolicyStatus() +SavePolicy() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the family/group overview (Policy section). The policy information can be changed as per defined 'use-cases' for adding and editing a Policy. On saving a Policy it will be added /refreshed on the current policies section within the family/group overview page.

5.2.13. FindPremium.aspx

The screenshot shows the 'FindPremium.aspx' page. At the top, there is a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. There is also a search bar for 'Search Insurance No' and a help icon. Below the navigation is a 'Select Criteria' section with fields for Payer, Payment Date From, Payment Date To, Region, District, and Contribution Paid. A 'Search' button is located at the bottom right of this section. The main area displays a grid titled '77 Contributions Found' with columns for PAYMENT DATE, CONTRIBUTION PAID, PAYER, PAYMENT TYPE, RECEIPT NO., VALID FROM, and VALID TO. The grid contains 77 rows of data. At the bottom of the grid, there is a navigation bar with links for 1 through 6. In the bottom right corner of the main area, there is a 'Cancel' button.

Interface object name:	FindPremium.aspx
Menu path:	Contributions
Hyperlinks/Redirections:	
Action Button:	OverviewFamily.aspx (Cancel button)
Use case Reference	5.2.8
Class Diagram	<pre> classDiagram class FindPremiumBI { <<Methods>> checkRoles() GetDistricts() GetPayers() GetPremium() GetRegions() GetTypeOfPayment() } </pre>

Brief description:

This interface opens up when the user has clicked the Contributions button option on the top menu (Insurees and Policies). This interface is used to search for Contributions. The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Payer) in the grid will open up the family/group overview page. Further operations on the Contribution will take action from this overview page.

5.2.14. Premium.aspx

The screenshot shows the 'Premium.aspx' page. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar is a search bar labeled 'Search Insurance No' with a dropdown arrow and a magnifying glass icon. Below the navigation bar, there are two main sections: 'Family/Group Details' and 'Contribution'.

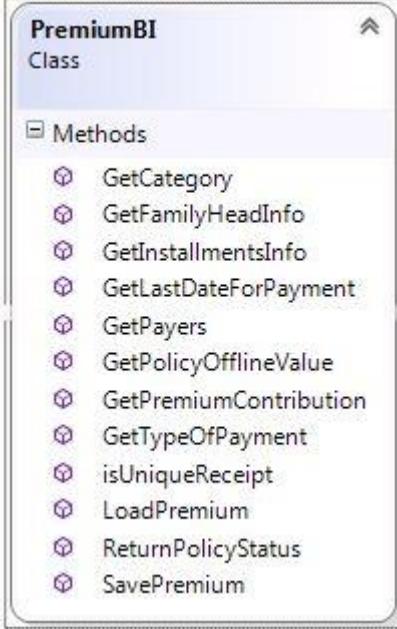
Family/Group Details:

Insurance Number	9003003	Region	TestRegion	Confirmation Type	Other
Last Name	Letara	District	TestDistrict1	Confirmation No.	Rgc
Other Names	Felister	Municipality	TestMunicipality11	Permanent Address Details	
Poverty Status	Yes	Village	TestVillage111		

Contribution:

Payer	-- Select a Payer --	Policy Details			
Contribution Category	Contribution and Other	Policy Value	11,000.00	Balance	0.00
Contribution Paid	11,000.00	Contribution Paid	11,000.00	Policy Status	Active
Receipt No.	Fdsdjjbv1				
Payment Date	21/10/2017				
Payment Type	Mobile Phone				

At the bottom of the page are two buttons: 'Save' and 'Cancel'.

Interface object name:	Premium.aspx
Menu path:	
Hyperlinks/Redirections:	Pay Date column in Contributions section in OverviewFamily.aspx
Action Button:	OverviewFamily.aspx (Add and Edit button in Policy section)
Use case Reference	5.2.4 (Add Contributions) 5.2.12 (Modify Contributions)
Class Diagram	 <pre> classDiagram class PremiumBI { <<Methods>> GetCategory() GetFamilyHeadInfo() GetInstallmentsInfo() GetLastDateForPayment() GetPayers() GetPolicyOfflineValue() GetPremiumContribution() GetTypeOfPayment() isUniqueReceipt() LoadPremium() ReturnPolicyStatus() SavePremium() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the family/group overview (Contributions section). The Contribution information can be changed as per defined 'use-cases' for adding and editing a Contributions.

On saving a Contribution it will be added /refreshed on the current Contributions section within the family/group overview page.

The Load Contribution call will also fetch the amount to be paid according the product(s) selected and member counts (member count, quantity of adults and children) of the family/group (refer to use cases). In case a Contribution paid is lower than the amount to be paid, a message will appear alerting the operator of such case. In the latter case the operator has the option to activate the policy regardless the covered amount. Each product has a grace period. In case of 'under' payments, the system will automatically 'suspend' the policy after expiration of the grace period. This action is performed by a daily service that runs in the background.

A policy will become active only when the sum of Contributions paid is equal or higher than the amount to be paid.

5.2.15. OverviewFamily.aspx

The screenshot shows the 'OverviewFamily.aspx' page with four main sections:

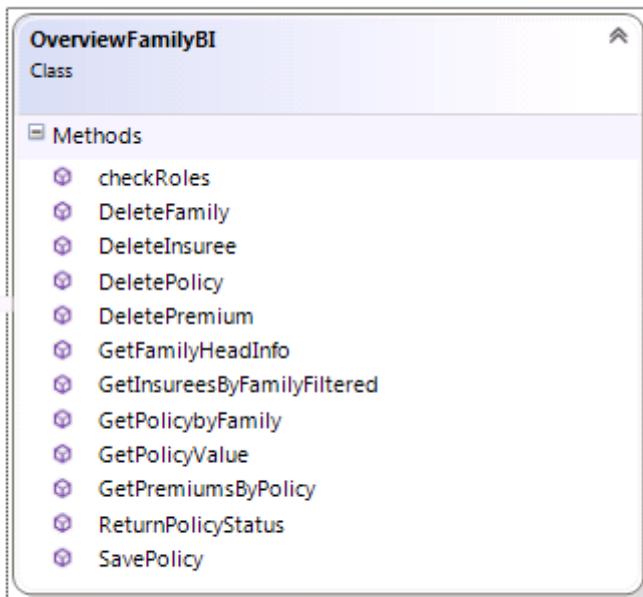
- Family/Group:** Displays basic information: Insurance Number (777888991), Last Name (Tusk), Other Names (Jean), Region (TestRegionX), District (TestDistrictX1), Municipality (TestMunicipality11), Village (TestVillageX111), Poverty Status (Yes), Confirmation Type (Municipality), and Address Details (str. Majdalena).
- Insurees:** A table listing three individuals:

INSURANCE NUMBER	LAST NAME	OTHER NAMES	GENDER	BIRTH DATE	BENEFICIARY CARD
777888991	Tusk	Jean	M	09/05/1993	<input checked="" type="checkbox"/>
777888992	Tusk	Jane	F	30/11/1989	<input checked="" type="checkbox"/>
777888993	Tusk	Loran	F	09/05/2016	<input checked="" type="checkbox"/>
- Policies:** A table listing one policy:

ENROL DATE	EFFECTIVE DATE	START DATE	EXPIRY DATE	PRODUCT	ENROLMENT OFFICER	POLICY STATUS	POLICY VALUE	VALID FROM	VALID TO
15/07/2017	15/07/2017	15/07/2017	14/07/2018	RX001	Enrolment1 John	Active	17,500.00	15/07/2017	
- Contributions:** A table listing two contributions:

PAYMENT DATE	PAYER	AMOUNT	PAYMENT TYPE	RECEIPT NO.	CONTRIBUTION CATEGORY
15/07/2017		16,000.00	Bank Transfer	rx1	Contribution
15/07/2017		1,500.00	Bank Transfer	rx2	Contribution

Interface object name:	OverviewFamily.aspx
Menu path:	
Hyperlinks/Redirections:	FindFamily.aspx (Insurance Number) FindInsuree.aspx (Insurance Number) FindPolicy.aspx (Enrollment date) FindPremium.aspx (Payer) Product.aspx (product.aspx)
Action Button:	Premium.aspx (cancel button) Policy.aspx (cancel button) Insuree.aspx (cancel button) ChangeFamily.aspx (cancel button)
Use case Reference	5.2.15 (Delete Family/Group) 5.2.16 (Delete Insuree) N/A (Delete Policy) N/A (delete Contribution) <i>Basically this interface is the result of several searches and will be used to activate several other use cases (adding, editing, deleting)</i>

Class Diagram***Brief description:***

This page will be the main page for adding, editing and deleting of the following entities related to a family/group:

- Insurees
- Policies
- Contributions

All insurees related to the family/group (active only) will be shown in the section for Insurees. Adding of new Insurees under the same family/group will be done by clicking the add button in the Insuree section. Similarly we can add policies and Contributions that way.

Editing of any of the three entities will be performed by clicking the edit button or by clicking the hyperlink (first column) in the sections. The functionality of 'Modifying a family/group', 'change of head of family/group' or 'move insuree' is found under the edit function in the family/group section. The actual functionality has been described earlier in the chapter as for several other functionalities activated in this user interface.

By selecting the Policy, the Contribution section will be refreshed, showing the Contributions paid on the selected policy. By default the first policy found is selected automatically.

A separate hyperlink will be positioned on the 'Product' column in the policies section to show the actual product in view mode via a popup window.

Similarly a hyperlink will be available on the 'Payer' column in the Contributions section to provide, via a read-only popup, additional information on the payer.

By clicking the delete button in the sections, a validity check will take place on the deletions. In case a record should not be deleted the user will be informed. If the validation has passed, IMIS produces a confirmation box where the operator has to confirm the operation.

Please find below the confirmation message for this deletion.

The screenshot shows the 'Family/Group' section of the openIMIS application. At the top, there are tabs for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance' is also present. Below the tabs, there are sections for 'Insurance Number' (321321331), 'District Dodoma', 'Poverty Status Yes', and 'Address Details'. Under 'Insurees', there is a table with columns for INSURANCE NUMBER and LAST NAME, containing entries like 321321331 (Gau) and 321321332 (Gau). A modal dialog box titled 'IMIS' is centered over the page, asking 'Are you sure you want to delete the Family/Group ?' with 'Yes' and 'No' buttons. In the background, there are sections for 'Polices' and 'Contributions'.

After confirmation, the entity will be deleted by flagging the ValidityTo field in the record. No additional (audit) record needs to be inserted as the record will be logically deleted from the system. Special developed 'nested' SQL Triggers will update all related records by setting the ValidityTo field.

The design has also added 'deletion' possibilities for Products and Contributions as there might be situations in which the operator mistakenly entered policies or Contributions. These mistakes can in this case be rectified by deletions. No use cases support this scenario.

5.2.16. FindClaims.aspx

The screenshot shows the 'FindClaims.aspx' page. At the top, there are tabs for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance No' is also present. Below the tabs, there is a 'Select Criteria' section with fields for Region, HF Name, Visit Date From, District, Review Status, Claim Date From, HF Code, Feedback Status, Main Dg., Claim Administrator, Claim Status, Batch Run, Insurance Number, Claim No., Visit Type, and a 'Search' button. Below this, a message says '8 Claims Found' and a 'Select To Submit' button. A table displays 8 claims with columns: CLAIM NO., HF NAME, DATE CLAIMED, FEEDBACK STATUS, REVIEW STATUS, CLAIMED, APPROVED, and CLAIM STATUS. The claims listed are:

CLAIM NO.	HF NAME	DATE CLAIMED	FEEDBACK STATUS	REVIEW STATUS	CLAIMED	APPROVED	CLAIM STATUS
cl00122	District hospital 2	16/07/2017	Idle	Idle	14,600.00		Entered
clv002	Regional hospital	14/07/2017	Idle	Idle	1,200.00		Entered
CMP20170	Regional hospital	14/07/2017	Idle	Idle	1,700.00		Entered
clp00w	Regional hospital	13/07/2017	Idle	Idle	200.00		Entered
CLP0001	Regional hospital	10/07/2017	Idle	Idle	200.00		Entered
CLP001	Regional hospital	10/07/2017	Idle	Idle	2,200.00		Entered
xxx4	Regional hospital	27/06/2017	Idle	Idle	200.00		Entered
xxx1	Regional hospital	27/06/2017	Idle	Idle	1,000.00		Entered

At the bottom, there are buttons for Add, Load, Delete, Submit, and Cancel.

Interface object name:	<i>FindClaims.aspx</i>
Menu path:	<i>Health Facility Claims(under Claims menu)</i>
Hyperlinks/Redirections:	
Action Button:	<i>Claim.aspx (Save and Cancel buttons)</i>
Use case Reference	3.2.8(<i>Finding Claim Direct</i>)
Class Diagram	<pre> classDiagram class FindClaimsBI { <<Fields>> Claims_BL <<Methods>> checkRoles() DeleteClaim() GetBatchRun() GetClaims() GetClaimsCount() GetClaimStatus() GetDistricts() GetFeedbackStatus() GetHFClaimAdminCodes() GetHFCodes() GetICDCodes() GetRegions() GetReviewStatus() GetVisitTypes() IsClaimStatusChanged() SubmitClaims() WriteToXml() ZipXMLs() } </pre>

Brief description:

This interface opens up when the user has clicked the 'Claims' option under the top menu Claims. In case an operator is 'linked' to a specific Health facility, the District, Health Facility Code and Health Facility Name boxes are already entered and locked. Furthermore, in that case, all claims with the status 'Entered' are automatically displayed. Alternatively, the operator is able to search for the desired claims by using the various filters as seen in the screenshot. By clicking the search button, all records satisfying the criteria will appear. A click on the hyperlink on the first column (Claim Code) in the grid will open up the Claim page to show all related information to the claim.

The following buttons are available on the button bar (depending on statuses and operators role):

Button	Role	Claim Status
Add	Claim Administrator, Clerk	'Entered', 'Checked', 'Processed', 'Valuated'
Delete	Claim Administrator, Clerk	'Entered'
View	Medical Officer	'Entered', 'Checked', 'Processed', 'Valuated'
Load	Claim Administrator, Clerk	'Checked', 'Processed', 'Valuated'
Submit	Claim Administrator, Clerk	'Entered', 'Checked', 'Processed', 'Valuated'

Depending on the role of the operator and the status of the claim (see above), by clicking the 'Add', or 'View' buttons, the Claim.aspx interface will be opened.

The Claim interface will be used to add, edit or view a claim.

By clicking the hyperlink of the claim (first column), the claim interface will be opened in read-only mode or edit mode depending on the role of the user and the status of the claim.

As long as claims are in the status 'entered' they could be modified or deleted. To finally submit a claim to IMIS Administrator, Admin would need to select the claim on the right side of the data grid and click the 'Submit' button. The claim status will now automatically be changed to 'Checked' after the system has performed the standard verification.

One could select one or multiple claims for submission. By clicking the 'All' button above the column 'Submit' in the grid, all claims with status 'Entered' will be candidate for submission by having its Submit field checked. By clicking the 'Submit' button with having multiple claims selected, all claims will be verified and set to the status 'Checked'.

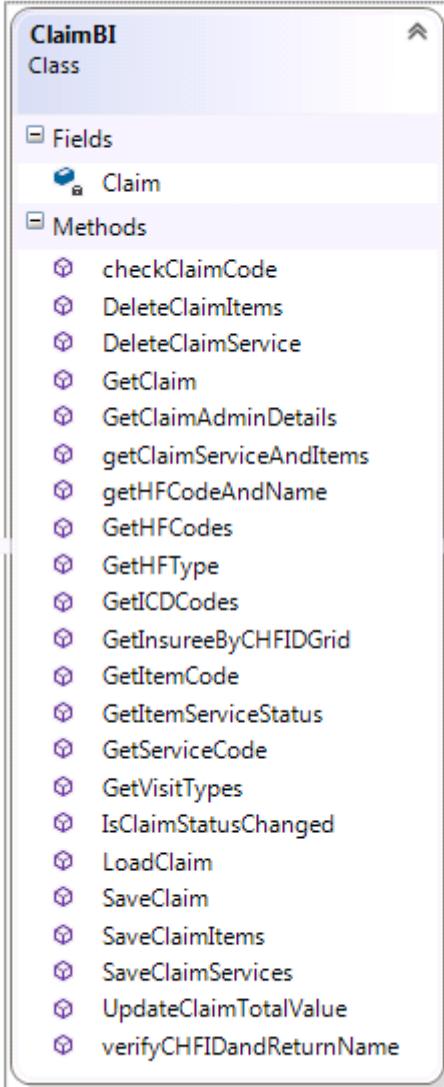
In case we are operating in an 'Offline' mode, on clicking the 'Submit' button, the system will generate an XML file containing the claims. At the same time, the claims are updated to 'Checked'. The 'Checked' status in the offline mode obviously should be interpreted differently as if we would have the checked status on Central level. In case we are Offline, 'Checked' means that the HF has submitted the claim.

The XML file needs to be provided (uploaded) to the server in order to be inserted in the Claims table. This process of inserting the XML based claims is done automatically by a windows service running in the background at the server. The upload procedure of the XML data from mobile phones and offline IMIS clients is exactly the same.

5.2.17. Claim.aspx

The screenshot displays the 'Claim.aspx' interface. At the top, there's a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance' is also present. The main content area is divided into sections:

- Claim Details:** Contains fields for HF Code (P10001), HF Name (Primary Health Centre), Visit Date From/To, Main Dg. (Select Dg.), Sec Dg1 (Select Dg.), Sec Dg2 (Select Dg.), Sec Dg3 (Select Dg.), Sec Dg4 (Select Dg.), Claim No., Date Claimed, Claimed amount (0.00), Claim Administrator (X101), and Guarantee No.
- Services:** A table with columns for SERVICE CODE, QUANTITY, PRICE, and EXPLANATION. It shows 5 rows of data.
- Items:** A table with columns for ITEM CODE, QUANTITY, PRICE, and EXPLANATION. It shows 5 rows of data.
- Buttons:** At the bottom are three buttons: Save, Add, and Cancel.

Interface object name:	Claim.aspx
Menu path:	
Hyperlinks/Redirections:	FindClaims.aspx (Claim code column)
Action Button:	FindClaims.aspx (Add and Edit button)
Use case Reference	3.2.2
Class Diagram	 <pre> classDiagram class ClaimBI { <<Fields>> Claim <<Methods>> checkClaimCode() DeleteClaimItems() DeleteClaimService() GetClaim() GetClaimAdminDetails() getClaimServiceAndItems() getHFCodeAndName() GetHFCodes() GetHFType() GetICDCodes() GetInsureeByCHFIDGrid() GetItemCode() GetItemServiceStatus() GetServiceCode() GetVisitTypes() IsClaimStatusChanged() LoadClaim() SaveClaim() SaveClaimItems() SaveClaimServices() UpdateClaimTotalValue() verifyCHFIDandReturnName() } </pre>

Brief description:

This interface opens up when the user has clicked the hyperlink on the Find Claim interface or the 'Add', or 'View' button on the Find Claim interface.

This interface is used to Add, Edit and View the selected claim.

The HF Code is mandatory and needs to be selected in case the operator has no HF code linked to its user profile.

The claim interface holds in the top section of the screen the header information on the claim. The operator has to enter all relevant information on the claim as per use cases. By clicking the save button, data will be

validated and stored in the database. Data is editable only in case the claim is in the status ‘entered’ and the operator role is ‘Clerk’ or ‘Claim Administrator’.

In all other cases, data is only available in read-only mode.

The claim total is dynamically calculated (client side) at loading of the claim and each time an item or service is added, changed or deleted.

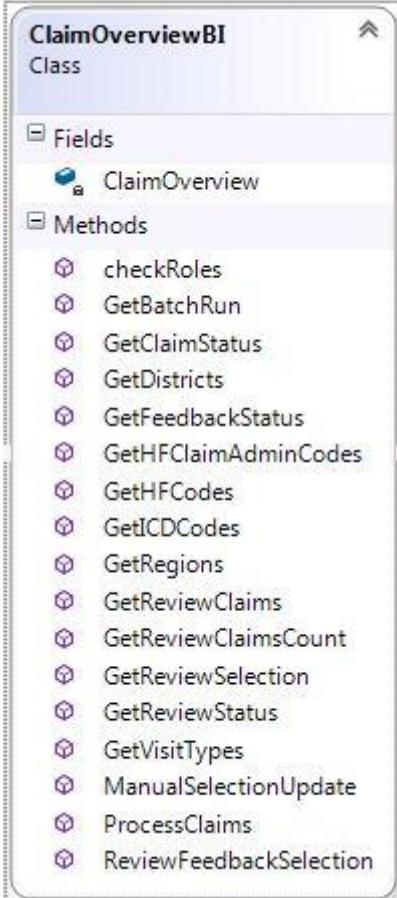
The operator is able to enter optionally any explanation to elaborate on the claim contents or value. Further to this, the operator is also able to enter an explanation on claim item or service level.

5.2.18. ClaimOverview.aspx

The screenshot shows the 'ClaimOverview.aspx' page. At the top, there is a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance No.' is also present. Below the navigation bar is a section titled 'Select Criteria' containing various dropdown menus and input fields for filtering claims based on Region, HF Name, Visit Date, District, Review Status, Claim Date, HF Code, Feedback Status, Main Dg., Claim Administrator, Claim Status, Batch Run, Insurance Number, Claim No., and Visit Type. A 'Search' button is located on the right of this section. Below this is a section titled 'Claim Selection Update' with a 'Criteria Details' panel featuring dropdowns for 'Select', 'Random', 'Value', and 'Variance', along with a 'Update' button. The main content area displays a table titled '10 Claims Found' with the following data:

CLAIM NO.	HF NAME	DATE CLAIMED	FEEDBACK	REVIEW	CLAIMED	APPROVED	CLAIM STATUS
CLM003	Regional hospital X	18/07/2017	Idle	Idle	12,000.00	12,000.00	Checked
CLM002	Regional hospital X	16/07/2017	Idle	Idle	16,000.00	16,000.00	Checked
CLM001	Health Centre 2	16/07/2017	Idle	Idle	28,000.00	8,000.00	Checked
clx12	Regional hospital	13/07/2017	Selected for Feedback	Selected for Review	5,900.00	1,900.00	Checked
xc01	Regional hospital	05/07/2017	Selected for Feedback	Selected for Review	8,500.00	6,000.00	Checked
xix1	Regional hospital	01/07/2017	Selected for Feedback	Selected for Review	9,000.00	6,000.00	Checked
xxx9	Regional hospital	05/07/2017	Delivered	Selected for Review	5,800.00	1,500.00	Checked
xxx8	Regional hospital	05/07/2017	Delivered	Selected for Review	5,200.00	2,200.00	Checked
C1500	Regional hospital	03/07/2017	Selected for Feedback	Selected for Review	1,200.00	200.00	Checked
xxx7	Regional hospital	28/06/2017	Delivered	Not Selected	2,000.00	2,000.00	Checked

At the bottom of the page, there are several buttons: 'Review', 'Feedback', 'Update', 'Process' (highlighted in red), and 'Cancel'.

Interface object name:	<i>ClaimOverview.aspx</i>
Menu path:	<i>Review (under Claims main menu)</i>
Hyperlinks/Redirections:	
Action Button:	<i>ClaimFeedBack.aspx (Save & Cancel buttons) ClaimReview.aspx (Save & Cancel buttons)</i>
Use case Reference	<i>3.2.9(Review selection and Review batch of Claims) 3.2.10(Feedback selection and prompting for feedback)</i>
Class Diagram	 <pre> classDiagram class ClaimOverviewBI { <<Fields>> ClaimOverview <<Methods>> checkRoles() GetBatchRun() GetClaimStatus() GetDistricts() GetFeedbackStatus() GetHFClaimAdminCodes() GetHFCodes() GetICDCodes() GetRegions() GetReviewClaims() GetReviewClaimsCount() GetReviewSelection() GetReviewStatus() GetVisitTypes() ManualSelectionUpdate() ProcessClaims() ReviewFeedbackSelection() } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Review’ menu option under the Claims menu. This interface is used to get information about the claim(s) and to select the claim(s) for further review or feedback.

If the claim has the status ‘Processed’ (=8) or Valuated (=16) the interface will only allow viewing of information.

The operator is able to filter the claims by using the various filters as seen in the screenshot. The following filters are available for filtering the claims:

- District
- HF Code
- HF Name
- Claim Diagnosis
- Insurance Number
- Claim From Date

- Claim To Date
- Visit From Date
- Visit To Date
- Review Status
- Feedback status
- Claim Status
- Claim Code
- Claim Administrator
- Visit Type
- Batch Run

The HF Code combo box will be only containing the HFs of the district (s) available for the current operator. The default Claim status will be set to 'Checked'.

The system will automatically show all claims with status 'Checked' And (Review Status OR Feedback Status) = 'Idle'. These claims will be the claims that need to be selected for Review or Feedback.

The 'claim date from' and 'claim date to' will be blank initially.

The operator can, after the automatic data retrieval, further filter the data by using the optional filters and clicking the 'Search' button.

Each Claim within the grid has a combo box that shows the current Review status of the claim. The following statuses could appear:

- I → Idle (=0)
- N → Not selected for review (=1)
- S → Selected for review (=2)
- R → Reviewed (=4)
- P → (By-) Passed (=8)

The operator can use 4 methods for the 'review' selection process

- Automatic: randomly select claims based on a certain percentage (default = 5%)
- Automatic: Select claims that are having a claim total over a certain value (default = 40,000)
- Automatic: Select claims that have a variance of over a certain percentage compared with the average of claims related to a similar ICD code over 1 years information (currently suggested) within the same district. (or country wide?) (Default =50%)
- Manually: Operator judges and changes the review status of claims in the grid manually and clicks the update button in the bottom bar of the interface.

The automatic selection can only be applied to the claims with a review status 'Idle'.

In case the operator chooses one of the automatic selection methods, all claims satisfying the criteria will be changed from 'I' →'N' or 'S'. It is possible to manually 'select' or 'un-select' a claim for review. In case the review status of a claim has been changed to 'R' (via interface ClaimReview.aspx), no further changes are allowed to the claim review status.

A click on the 'Review' button will open up the Claim Review page for the selected claim with the status 'S' or 'R' only.

Besides the review status, each claim within the grid has a combo box that shows the current Feedback status of the claim. The following statuses could appear:

- I → Idle (=0)
- N → Not selected for feedback (=1)
- S → Selected for feedback (=2)
- D → Delivered (=4)
- P → (By-) Passed (=8)

The operator can use 4 methods for the 'feedback' selection process:

- Automatic: randomly select claims based on a certain percentage (default = 5%)
- Automatic: Select claims that are having a claim total over a certain value (default = 40,000)

- Automatic: Select claims that have a variance of over a certain percentage compared with the average of claims related to a similar ICD code over 1 years information (currently suggested) within the same district. (Or country wide?) (Default =50%)
- Manually: Operator judges and changes the feedback status of claims in the grid manually and clicks the 'update' button in the bottom bar of the interface..

In case the operator chooses one of the automatic selection methods, all claims satisfying the criteria will be changed from 'I' →'N' to 'S'. It is possible to manually 'select' or 'un-select' a claim for feedback. In case the feedback status of a claim has been changed to 'D' via interface ClaimFeedback.aspx or the automatic upload of Feedback records via the XML upload procedure, no further changes are allowed to the claim feedback status.

A click on the 'Feedback' button will open up the Claim Feedback page for claims with the status 'S' or 'D' only.

The claims selected for feedback will be handles by a separate process that runs each day automatically (via a separate windows service) from the server. This process will loop through all claims that were selected for feedback and will register in a separate table 'TblFeedbackPrompts' all needed details to be sent via the SMS Message.

A separate interface will be available (like the policy Renewals in Phase 1) to allow reporting on the feedback SMS journal and running a report separate from the SMS service, in case the VEOs do not have mobile connectivity or the phone number is missing. The report will contain all information as mentioned in use case 3.2.10.

In case the medical officer considers all currently shown claims in the grid as 'reviewed' and would like to process these claims, he/she could change the status of these to 'Processed' (=8) by clicking the 'Process' button on the right side of the screen.

On clicking the 'process' button, the system will only consider claims in the grid with claim status 'Checked' and Review Status <> 'Idle' And Feedback Status <> 'Idle'.

In case one or more claims considered in the process, were having the status 'selected for review' at the moment of processing, a warning message will be displayed alerting the operator that some claims have not been reviewed yet. The same procedure will be applied to the feedback status. Any claims that are still in the 'Checked' status and are selected for review will be set to review status 'ByPassed' (=8). The same applies to claims that were selected for feedback but feedback was not delivered yet. In the latter case, the feedback status will be set to 'ByPassed' (=8).

After the operator confirms the update, all considered claims are set to 'processed' and the 'processed date' is set to 'now'.

The system will now attempt to 'valuate' the 'processed claims'. The 'valuation' process, is described later in paragraph 3.3.

5.2.19. ClaimReview.aspx

The screenshot shows the openIMIS ClaimReview.aspx page. At the top, there are tabs for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance' is also present.

Claim Details:

- Hospital: HF H10001 - Dodoma regional Hospital
- Insurance Number: 777777771
- Partient Name: Jane Foster
- Claimed: 8700.00
- Main Dg: A03
- Claim No.: vc4
- Date Claimed: 18/09/2016
- Approved: 300.00
- Sec Dg1: A03
- Sec Dg2
- Sec Dg3
- Sec Dg4
- Date Processed: 11/12/2016
- Visit Date From: 05/09/2016
- Visit Date To: 08/12/2016
- Adjusted: 240.00
- Claim Administrator: X100
- Visit Type: Referrals
- Guarantee No:

Services:

SERVICE CODE	QTY	PRICE	EXPLANATION	APP. QTY	APP. PRICE	JUSTIFICATION	STATUS	VALUATED R
DIFB12 Delivery-normal	1	8000.00		0.00			Rejected	16

Items:

ITEM CODE	QTY	PRICE	EXPLANATION	APP. QTY	APP. PRICE	JUSTIFICATION	STATUS	VALUATED R
GBBB09 Acetylsalicylic Acid	1	300.00					Passed	240.00 0
GBBC01 Ampicillin PDR	1	400.00		0.00			Rejected	4

Explanation: [Text Box] Claim Status: Valuated

Adjustment: [Text Box]

Cancel

Interface object name:	ClaimReview.aspx
Menu path:	
Hyperlinks/Redirections:	
Action Button:	ClaimOverview.aspx (Review button)
Use case Reference	3.2.9 (review a claim)
Class Diagram	<p>ClaimReviewBI Class</p> <p>Methods</p> <ul style="list-style-type: none"> ⌚ getClaimServiceAndItems ⌚ GetClaimStatus ⌚ GetItemServiceStatus ⌚ GetVisitTypeText ⌚ IsClaimReviewStatusChanged ⌚ IsClaimStatusChecked ⌚ IsItSystemRejectedItem ⌚ IsItSystemRejectedService ⌚ LoadClaim ⌚ ReturnClaimStatus ⌚ SaveClaimItemsforReview ⌚ SaveClaimReview ⌚ SaveClaimServicesforReview ⌚ UpdateClaimApprovedValue

Brief description:

This interface opens up when the user has clicked the 'Review' button on the Review selection interface (claim review status 'S' or 'R'). This interface is used for displaying claim information and entering review data by the medical officer.

The medical officer can adjust the following statuses manually:

- Claim status ([Checked] ,Processed, Rejected)
- Claim Item Status (Passed , Rejected)
- Claim Service Status (Passed, Rejected)

If a claim has not yet been set as processed, the operator can still change rejected items and services as passed as long as the rejection did not come from the automated check at the moment of submitting the batch. A separate field will be kept internally for the rejection reasons (automatic or manual)

A claim could also be manually adjusted by entering the Qty Approved and Value approved fields in the sections for services and items. An optional justification comment could be provided as well. Further to this, the medical officer could simply reject the claim item or service.

Additional justification text can be amended to each claim item and service and also on general claim level. The date released will be set by the system as soon as the claim has been set to the status valuated (automated via a system procedure).

This interface is only available to medical officers.

The interface will show three dynamically calculated totals for values (locked):

- Original Claim value
- Adjusted Claim value
- Valuated Claim value

A claim with status 'Checked' and feedback status <>'Idle' and review status <> 'Idle' could be set to processed by changing the Claim status and clicking the 'Save' button.

At the moment of saving a Claim with the new status 'Processed' , the status for Review will automatically change to 'Reviewed' (=4). In case the Feedback Status is 'S' but feedback was not delivered yet, the Feedback status will be set to 'ByPassed' (=8). Furthermore, the date processed will be set to 'now'.

5.2.20. ClaimFeedback.aspx

The screenshot shows the 'Claim' section of the application. At the top, there are tabs for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance' is also present. The main area is titled 'Claim' and contains 'Claim Details' and 'Feedback' sections.

Claim Details:

- Claim No. t055
- Insurance Number: 11111111
- Last Name: Thadei
- Other Names: Gideon Saidi
- HF Code: H10001
- HF Name: Dodoma regional Hospital
- Visit Date From: 13/11/2016
- Claim Status: Rejected
- Claim Date: 14/11/2016
- Visit Date To: 13/11/2016
- Review Status: Reviewed
- Claim Administrator: X100
- Feedback Selected for Status Feedback

Feedback:

Enrolment Officer: -- Select Enrolment Officer --

Care Rendered: -- Select Yes/No --

Payment Asked: -- Select Yes/No --

Drugs Prescribed: -- Select Yes/No --

Drugs Received: -- Select Yes/No --

Overall Assessment: 0 1 2 3 4 5

Feedback Date: [Text input field]

Buttons at the bottom: Save and Cancel.

Interface object name:	ClaimFeedback.aspx
Menu path:	
Hyperlinks/Redirections:	
Action Button:	ClaimOverview.aspx (Feedback button)
Use case Reference	3.2.11 (Feedback of a claim)
Class Diagram	<pre> classDiagram class ClaimFeedbackBI { <<Methods>> +GetOfficers() +GetYesNo() +LoadClaim() +ReturnClaimStatus() +ReturnFeedbackStatus() +ReturnReviewStatus() +SaveFeedback() } </pre>

Brief description:

This interface opens up when the user has clicked the 'Feedback' button on Claim Review interface (claim feedback status 'S' or 'D') this interface is used for displaying feedback information on a claim for the medical officer and entering feedback data by the clerk.

The clerk will enter the required feedback and will click the save button to insert the information in the database. At the time of saving, the Feedback status of the claim changes to 'D' (delivered = 4). Feedback records could be manually entered or will be uploaded automatically by a service running in the backgrounds that uploads XML files originating from the mobile phones (via mobile feedback application).

5.2.21. BatchRun.aspx

The screenshot shows the 'Batch Processing' page of the openIMIS web application. The top navigation bar includes links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Search Insurance No.' is also present.

Batch Processing

Filter

Type	Year	Period	Region	District	Product	Category
Monthly	--Year--	--Period--	--Select Region--	--District--	--Product--	--Select category--

Display

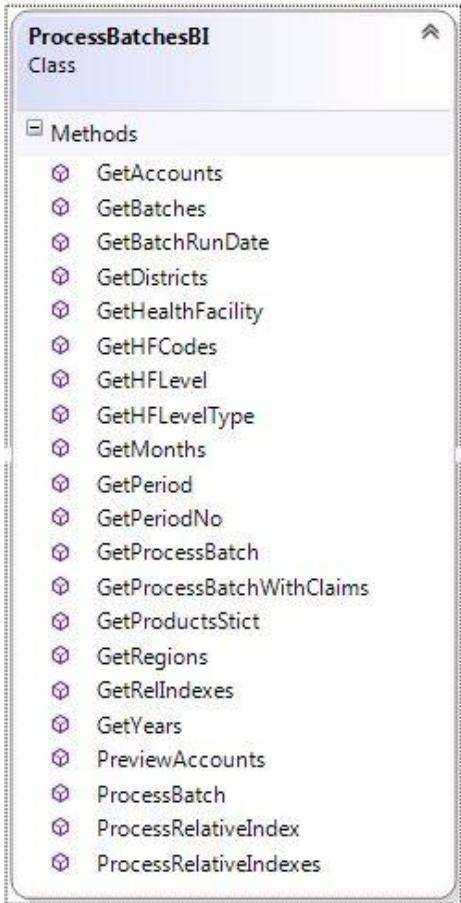
YEAR	PERIOD	PRODUCT	CATEGORY	CALCULATED DATE	INDEX
2017	May	District fixed enrolment	Hospital	12/07/2017	100.00
2017	May	District fixed enrolment	Non Hospital	12/07/2017	100.00
2017	June	District fixed enrolment	Hospital	12/07/2017	100.00
2017	June	District fixed enrolment	Non Hospital	12/07/2017	100.00
2017	July	Regional X product	Hospital	16/07/2017	0.81
2017	July	Regional X product	Non Hospital	16/07/2017	3.40
2017	July	Regional X product	Hospital	19/07/2017	1.60
2017	July	Regional X product	Non Hospital	19/07/2017	3.66

Filter for Accounts

Group By: HF Product Show Claims

Region	District	Product	HF	Level	Date From	Date To	Preview
--Select Region--	--District--	--Product--	--Select HF Code--	--Select Level--			

Result filtered successfully

Interface object name:	<i>ProcessBatches.aspx</i>
Menu path:	<i>Batch Run (under Claims menu)</i>
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	<i>3.2.15 (Calculating the relative indexes) and 'Export to Epicor'</i>
Class Diagram	 <pre> classDiagram class ProcessBatchesBI { <<Methods>> GetAccounts GetBatches GetBatchRunDate GetDistricts GetHealthFacility GetHFCodes GetHFLevel GetHFLevelType GetMonths GetPeriod GetPeriodNo GetProcessBatch GetProcessBatchWithClaims GetProductsStict GetRegions GetRelIndexes GetYears PreviewAccounts ProcessBatch ProcessRelativeIndex ProcessRelativeIndexes } </pre>

Brief description:

This interface opens up when the user has clicked the 'Batch Run' menu option in the Claims menu. This function is used to generate the monthly, quarterly and yearly indexes used for the calculation of relative prices for items and services that have been defined under the product with Price origin 'R' (Relative). This function should be executed each month. In normal cases this process is executed one month after the calendar month has been passed. E.g. we would execute this process for the month of June by the end of July to allow health facilities and IMIS Scheme administrator to(enter, submit, review etc..) all batches for the period of June.

By clicking the process button, the system will calculate the index for the month selected. Furthermore, in case we are processing month 3, 6, 9 or 12, we will also run the calculation process for the quarterly index. In case we are in month 12 we will besides the monthly and quarterly index also calculate and insert a yearly index record in table *tblRelIndex*. The actual process that will run is elaborated in a separate flowchart. The system will not allow duplicate indexes for periods. In case an operator attempts to insert an already existing index, a user warning will be displayed and the process will be cancelled.

After running the calculation of relative indexes as described above, the system will go through all claims that are in the status 'Processed' and will attempt to set the claims to the status 'Valuated' by going through all claims which are in the status 'processed' and applying the relative index were needed. If all items and services in a claim are valuated, the claim is valuated.

After all claims are being handled and possibly be set to 'valuated', we will run through all claims in the state 'valuated' that have not yet been assigned a so called 'Run ID'.

This ID will 'bind' all claims belonging to the same run together for reporting to the accounting system. All unassigned 'valuated' claims are now updated with the 'RunID'.

All batches that could not be 'valuated' are kept in the status 'Processed' and might be included in the next run. Yearly, Quarterly and Monthly runs are kept separate.

The search button will allow the operator to filter existing indexes for products etc. The calculated indexes that satisfy the criteria will be displayed in the grid.

The lower section will allow filtering records from the BatchRun table. The following filters will be available:

- Period from (will be filtering Batch run date)
- Period to (will be filtering Batch run date)
- Month of run
- Year of run
- District
- Product
- Health facility
- Care type (Inpatient , Outpatient, both)

After setting the filters the system will prevail which batch RUNS (full or partial contents) satisfy the criteria. The preview button will generate the report based on the criteria. This report could be a subset of what was sent as an instruction to the accounting system or the full accounting instruction. The report will be grouped by District, Health facility and Product.

5.2.22. ClaimAdministrator.aspx

The screenshot shows a web application interface for managing claim administrators. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar are icons for search, help, and version information (v16.3.0).

The main content area is titled "Claim Administrator Details". It contains the following fields:

- Code: X102
- Other Names: Peter
- Last Name: Fulton
- Date of Birth: 12/06/1984
- Phone Number: (empty field)
- Email: (empty field)
- HF Code: H10011 - Kongwa Di: (dropdown menu)

At the bottom of the form are two buttons: "Save" and "Cancel".

Interface object name:	<i>ClaimAdministrator.aspx</i>
Menu path:	<i>Claim Administrators (under Administration menu)</i>
Hyperlinks/Redirections:	<i>FindClaimAdministrator.aspx (Claim Administrator Code)</i>
Action Button:	<i>Add, Edit Button</i>
Use case Reference	<i>(Adding, Editing, Deletion of Claim Administrators)</i>
Class Diagram	<pre> classDiagram class ClaimAdministratorBI { <<Methods>> CheckRoles() GetHFCodes() LoadClaimAdmin() RunPageSecurity() SaveClaimAdmin() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the claim administrator search page (or clicked the claim administrator code hyperlink). The claim administrator information can be changed as per defined ‘use-cases’ for adding and editing claim administrator.

On saving, claim administrator will be added /refreshed on the data grid within the claim administrator search page as described before.

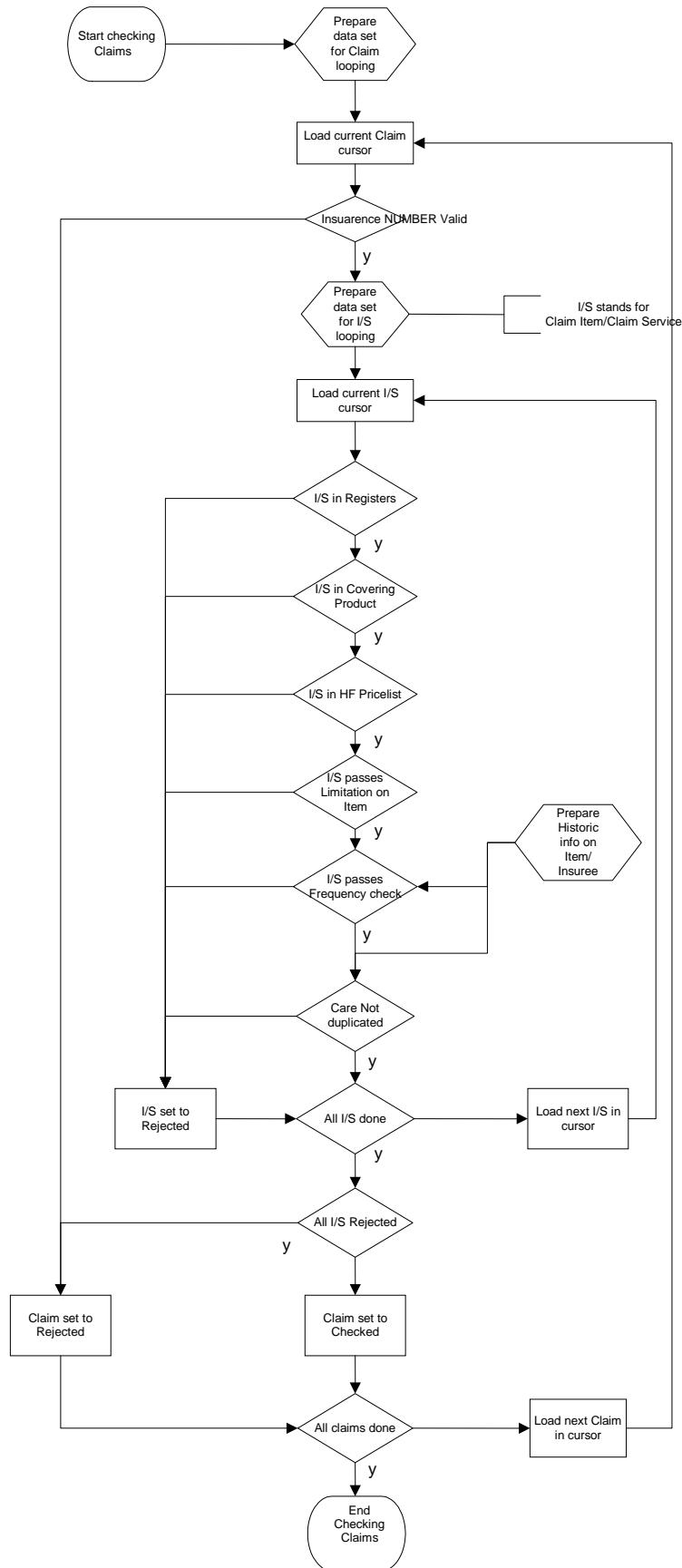
5.3. Checking of claims process

The checking of claims process will be automatically executed at the moment the Claim administrator or Clerk submits the claims (via the Submit button). The actual checking procedure will run purely on T-SQL level at the server side. The procedure will loop through all claims one by one. Each claim will have again a looping mechanism for each item and service found under the claim. The items and claims are checked on validity and could have either the status ‘Passed’ or ‘Rejected’.

Claims will be updated by the process from the status ‘entered’ (default) towards status ‘checked’ or ‘rejected’.

The batch will be updated from ‘Open’ towards ‘Checked’ or ‘Rejected’.

The following flowchart will graphically present the checking process of submitted Claims



5.4. Batch Valuation process

The claim valuation process will start at the moment of clicking the Process button on the ClaimOverview page (possibly multiple claims) or on the individual save action on the Claim review interface e (ClaimReview.aspx)

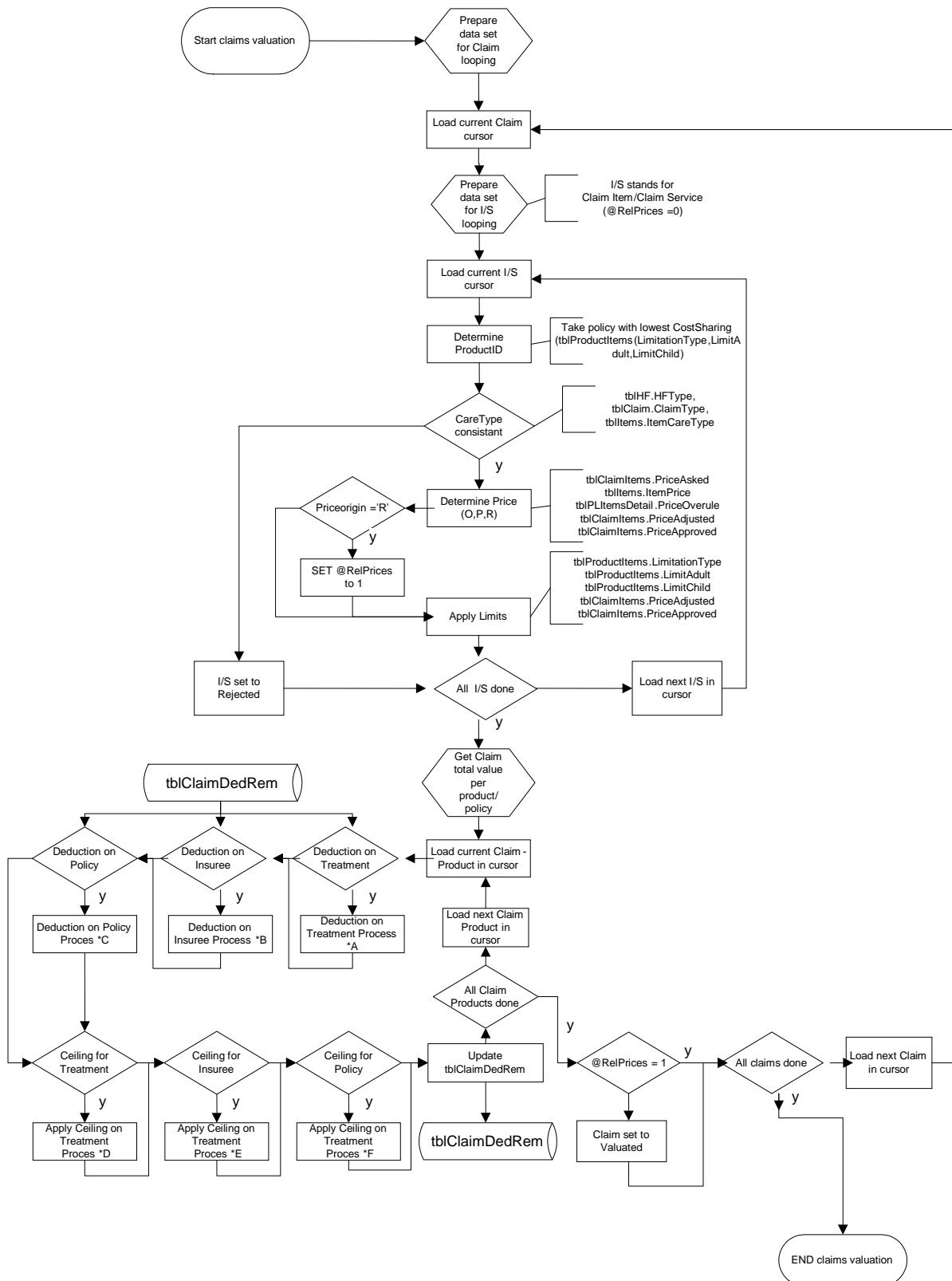
Automatically the system will run through a process that will loop through all claims and will update the claims with the amount to be remunerated towards the health facilities.

Most of the validations are done on item and service level lined up with the actual covered product /policy for the insuree. First we will validate and provide information on prices of services and items. Thereafter we will loop through all products and adjust (if needed) the amounts to be renumerated depending on deductions (Treatment, Insuree, Policy) and Ceilings (Treatment, Insuree, Policy).

In case a product item or service involved in the claim is having the 'price origin' set to 'R' (relative prices), then the final valuation of the claim can only be done after the relative prices index has been calculated for the period in question.

The process of relative price calculation will be performed manually on monthly basis. In case we are generating the monthly index for March we will, at the same time calculate the relative price index for the first quarter of the year. In case we calculate the relative price index for the month of December, we will at the same time also calculate the index for the last quarter of the year and the yearly index. The process of index calculation is covered in a separate paragraph.

The claim valuation process will also insert records in table 'tblClaimDedRem'. This table will hold all transactions on deductibles and remunerations per policy and per insuree. This table will be used for output and input in the valuation process.



5.5. Calculation of 'relative prices index' process

The process of calculation of the index of relative prices is needed to determine the actual price to be paid to the health facility for items and services that were covered with an insurance product with the price origin 'R' (relative prices) for its items and services.

The calculation of relative prices is done once a month on manual basis for the menu function 'Batch Run' under the claims menu. This function is available for restricted use, e.g. only the IMIS Scheme administrator.

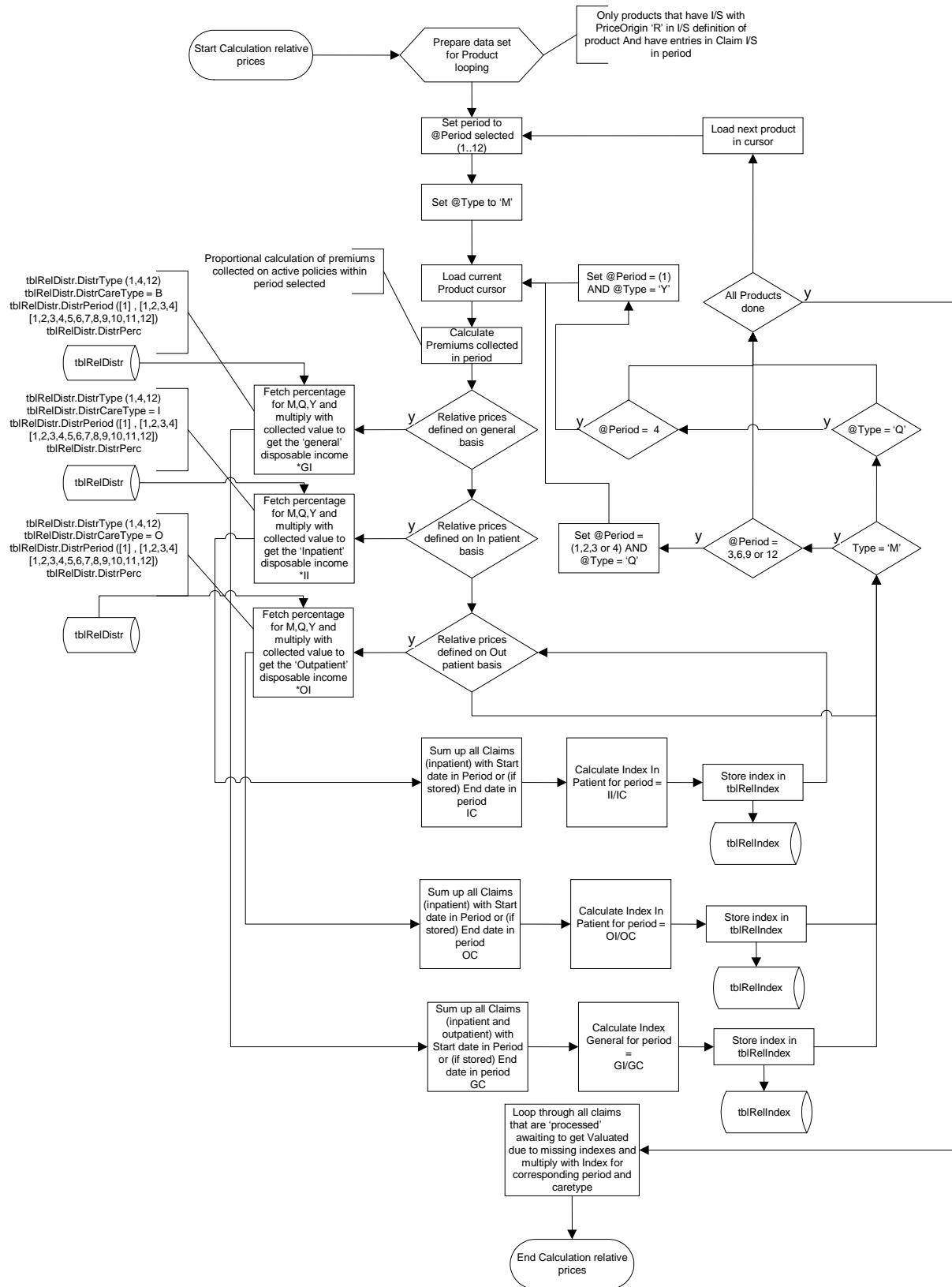
The function will first gather all Contributions collected for the period selected, e.g. March 2012. This value will be proportionally calculated as Contributions are normally prepaid for a year (or other period defined). In this calculation we will ignore the 'grace' period and late payments on Contributions. We will only use the policy value spread out over the period (default = 12) in question and use the 'slice' that will be valid for March 2012.

Now we will need to calculate the total amount of claims over the calculation period (e.g. March or Quarter 1 2012). This is done by the use of the table `tblClaimItems` and `tblClaimServices`. We will check how much is claimed within the period in question by looking at the start date of health care provided or the end date of the health care in case it was inpatient care.

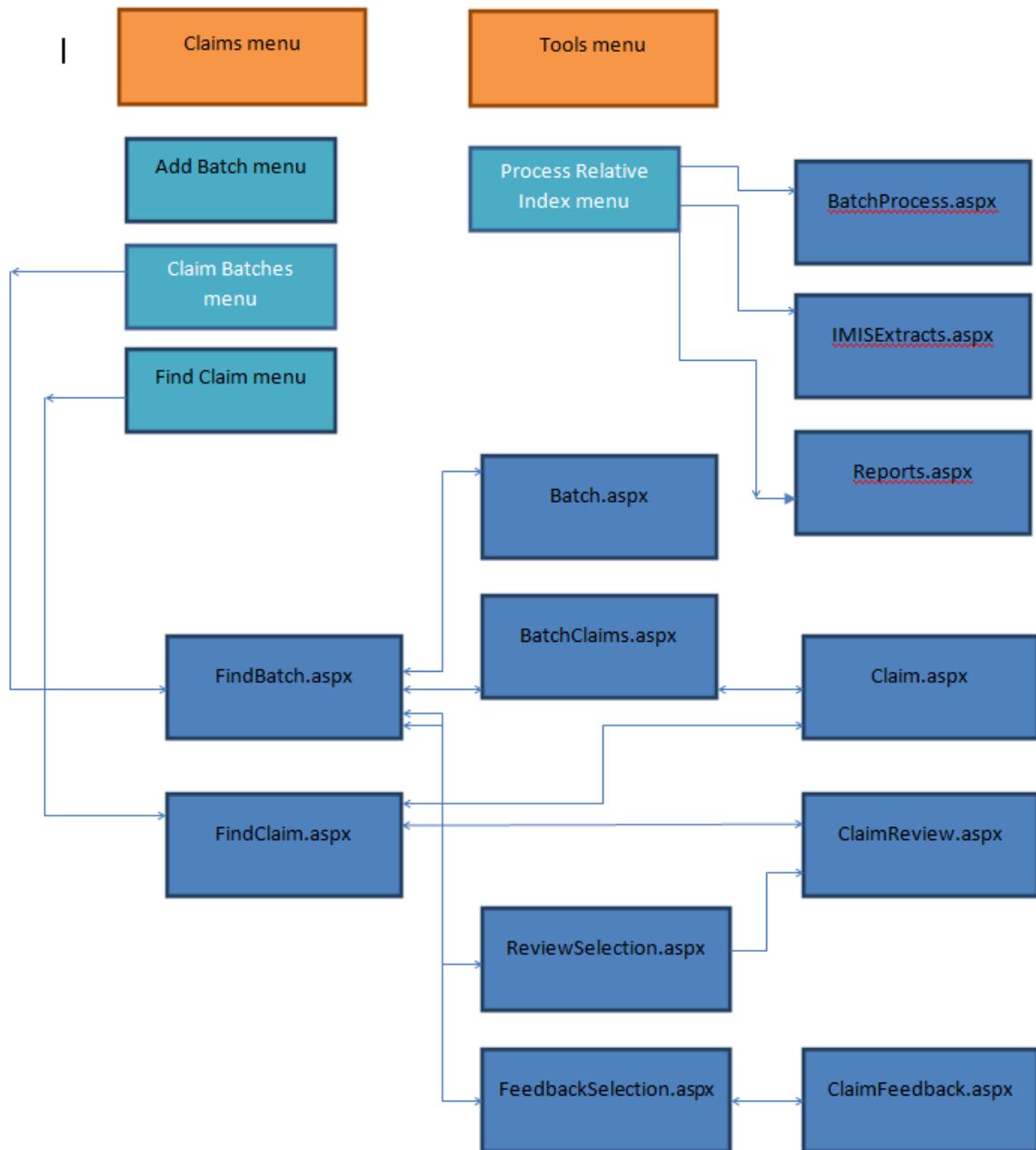
Separate indexes are calculated:

- General (all claims are used in the calculation)
- In patient care (Only claims from health facilities of HFLevel 'H' are considered)
- Out-patient care (Only claims from health facilities with HFLevel 'C' and 'D' are considered)

Please find below the flowchart of this process.



5.6. Graphical page routing overview Claim interfaces



5.7. Overview of status fields

Please find below an overview of the different statuses found in several tables and the actual interface/action that would set the status in question.

Table	Field	Status	Description
<i>TblClaim</i>	ClaimStatus	1	Rejected
		2	Entered
		4	Checked
		8	Processed
		16	Valuated
	ReviewStatus	0	Idle
		1	Not selected
		2	Selected for Review
		4	Reviewed
		8	ByPassed
	FeedbackStatus	0	Idle
		1	Not selected
		2	Selected for Feedback
		4	Delivered
		8	ByPassed
	ApprovalStatus	1	N/U
		2	N/U
		4	N/U
<i>TblClaimItem</i>	ClaimItemStatus	1	Passed
		2	Rejected
<i>TblClaimService</i>	ClaimServiceStatus	1	Passed
		2	Rejected

5.7.1.FindHealthFacility.aspx

The screenshot shows the 'FindHealthFacility.aspx' page. At the top, there's a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, and a search bar for 'Search Insurance No'. Below the navigation is a 'Select Criteria' section with fields for HF Code, Name, Level, Phone Number, Email, Legal Form, Region, District, Care Type, and Fax. A 'Search' button is also present. The main area displays a table titled '7 Facilities Found' with columns: HF CODE, NAME, LEGAL FORM, LEVEL, CARE TYPE, PHONE NUMBER, REGION, DISTRICT, ID, VALID FROM, and VALID TO. The table contains 14 rows of data. At the bottom of the page are buttons for Add, Edit, Delete, and Cancel.

HF CODE	NAME	LEGAL FORM	LEVEL	CARE TYPE	PHONE NUMBER	REGION	DISTRICT	ID	VALID FROM	VALID TO
HF01	Regional hospital	Government	Hospital	Both		TestRegion	TestDistrict1	1	28/09/2017	
HF02	District1 health Center	District organization	Health Centre	Out-Patient	+42011111111111	TestRegion	TestDistrict1	3	28/09/2017	
HF03	Health centre District2	District organization	Health Centre	Out-Patient		TestRegion	TestDistrict2	6	28/09/2017	
HF04	Regional hospital X	District organization	Hospital	Both		TestRegionX	TestDistrictX1	8	15/07/2017	
HF05	Regional Health Centre	District organization	Health Centre	Out-Patient		TestRegionX	TestDistrictX1	9	16/07/2017	
HF06	Health Centre 2	Government	Health Centre	Out-Patient		TestRegionX	TestDistrictX2	11	16/07/2017	
HF07	District hospital 2	Private organization	Hospital	Both		TestRegionX	TestDistrictX2	12	16/07/2017	
HF08										
HF09										
HF10										
HF11										
HF12										
HF13										
HF14										

Interface object name:	FindHealthFacility.aspx
Menu path:	Health Facilities
Hyperlinks/Redirections:	
Action Button:	HealthFacility.aspx (Save and Cancel button)
Use case Reference	6.2.17(Finding Health Facility) 6.2.19 (Deleting Health Facility)
Class Diagram	<pre> classDiagram class FindHealthFacilityBI { <<Methods>> checkRoles() DeleteHealthFacility() GetDistricts() GetHealthFacility() GetHFLegal() GetHFLevel() GetHType() GetRegions() } </pre>

Brief description:

This interface opens up when the user has clicked the 'Health Facilities' button option on the top menu (Administration). This interface is used to search for Health Facilities and acts as the selector for additions, modifications and deletions on the entity Health facility.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Name) in the grid will open up the health facility page. Any further operations on Health Facilities will take action from this page.

5.7.2. HealthFacility.aspx

The screenshot shows the 'Health Facility Details' form. It includes fields for Legal Form (Government), Level (Hospital), Sub Level (Select SubLevel), Code (HF01), Name (Regional hospital), Address (str. Colombo), Region (TestRegion), District (TestDistrict1), Phone Number, Fax, Email, Care Type (Both), Price Lists (Medical Services) (Hospital service pricel), Price Lists (Medical Items) (Hospital items pricelis), and Account Code. Below the form are three dropdown grids:

- Region:** Options include Region, North West, South West, Central, Capital, TestRegion, Singida, Maritime, Plateaux, Central, Kara, Savanes, Togo Délégation Générale (Nationale), Sud Kivu, TestRegion2, East West, Hans, TestRegionX, and Region.
- District:** Options include TestDistrict1, TestDistrict2, TestDistrict3, TestDistrict4, TestDistrict5, and TestDistrict6.
- Municipality:** Options include TestMunicipality11, TestMunicipality12, TestMunicipality13, TestMunicipality21, TestMunicipality31, TestMunicipality32, TestMunicipality33, TestMunicipality11, TestMunicipality12, TestMunicipality13, TestMunicipality31, TestMunicipality32, TestMunicipality33, TestMunicipality34, TestMunicipality14, and TestMunicipality15.

Below the grids is a 'Village' section with columns for VILLAGE and CATCHMENT, containing entries like TestVillage111, TestVillage112, etc., each with a value of 100.

At the bottom are 'Save' and 'Cancel' buttons.

Interface object name:	HealthFacility.aspx
Menu path:	
Hyperlinks/Redirections:	FindHealthFacility.aspx (name column)
Action Button:	FindHealthFacility.aspx (Add and Edit button)
Use case Reference	6.2.16 (Add Health Facility) 6.2.18 (Modify health Facility)

Class Diagram

The screenshot shows a 'Class Diagram' interface. On the left, there is a vertical sidebar with the title 'Class Diagram'. To the right of this is a main panel containing a class definition for 'HealthFacilityBI'. The class has the following details:

- Name:** HealthFacilityBI
- Type:** Class
- Methods:**
 - GetAllVillage
 - GetAllWards
 - GetDistricts
 - GetHFLegal
 - GetHFLevel
 - GetHFType
 - GetPLItems
 - GetPLServices
 - GetRegions
 - GetSublevel
 - GetVillages
 - GetWards
 - LoadCatchmentDistricts
 - LoadCatchmentVillage
 - LoadCatchmentWard
 - LoadCathmentRegions
 - LoadHF
 - SaveHealthFacilities

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the health facility search page (or clicked the hyperlink). The health facility information can be changed as per defined 'use-cases' for adding and editing a health facility.

On saving a health facility, it will be added /refreshed on the data grid on the health facilities search page as described before.

5.7.3.FindProduct.aspx

The screenshot shows the 'FindProduct.aspx' interface. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, and a search bar labeled 'Search Insurance' with a dropdown menu. Below the navigation bar is a 'Select Criteria' section with fields for Product Code, Name, District (selected to 'Select a District'), Date From, Date To, and a checkbox for 'Historical' data. A 'Search' button is also present. Below this section is a header '37 Products Found' followed by a table displaying product information. The table has columns: CODE, NAME, DISTRICT, DATE FROM, DATE TO, MAXIMUM LUMP MEMBERS SUM, CONTRIBUTION ADULT, CONTRIBUTION CHILD, INSURANCE PERIOD, GRACE PERIOD ENROLMENT, VALID FROM, and VALID TO. The table lists various products from different districts like Dodoma, Bafut, and Bamenda. At the bottom of the table are navigation buttons for page 1, 2, and 3, and action buttons for Add, Edit, Duplicate, and Cancel.

CODE	NAME	DISTRICT	DATE FROM	DATE TO	MAXIMUM LUMP MEMBERS SUM	CONTRIBUTION ADULT	CONTRIBUTION CHILD	INSURANCE PERIOD	GRACE PERIOD ENROLMENT	VALID FROM	VALID TO
5555PS	Primary and Secondary		01/11/2016	31/12/2020	6 10,000.00	4,000.00	2,000.00	12	0 11/11/2016		
BDACOL	Bafut Colleges		01/01/2014	31/12/2030	5,000 0.00	4,500.00	4,500.00	12	6 15/11/2016		
BDAH/H	Bafut Household		01/01/2014	31/12/2030	10 0.00	4,000.00	4,000.00	12	3 14/11/2016		
BSL1	LAMAL		01/11/2016	31/12/2020	6 10,000.00	4,000.00	2,000.00	12	0 18/11/2016		
DBAORG	Bamenda Org/Institutions		01/01/2014	31/12/2030	5,000 0.00	4,500.00	4,500.00	12	6 11/11/2016		
DFB002	Basic free enrolment Dodoma	Dodoma	01/01/2015	02/01/2025	5 20,000.00	1,000.00	500.00	12	0 02/12/2016		
DFC002	Ceilings free enrolment Dodoma	Dodoma	01/01/2015	01/01/2030	100 10,000.00	2,000.00	1,000.00	12	0 01/01/2015		
DFW001	Waiting free enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	0 01/01/2015		
DXC001	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/01/2015		
DXC002	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/11/2016		
DXC003	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/11/2016		
DXC004	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/11/2016		
DXC005	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/11/2016		
DXC006	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 01/11/2016		
DXC007	Ceilings fixed enrolment Dodoma	Dodoma	01/01/2015	01/01/2031	7 30,000.00	5,000.00	3,000.00	12	1 05/11/2016		

Interface object name:	<i>FindProduct.aspx</i>
Menu path:	<i>Products</i>
Hyperlinks/Redirections:	
Action Button:	<i>Product.aspx (Save and Cancel button)</i>
Use case Reference	<i>6.2.14(Finding Insurance Product) N/A (Deleting Insurance Product)</i>
Class Diagram	<pre> classDiagram class FindProductsBI { <<Methods>> checkRoles DeleteProduct GetDistricts GetProducts GetRegions } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Products’ button option on the top menu (Administration). This interface is used to search for Products and acts as the selector for additions, modifications and deletions on the entity Product.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Product Code) in the grid will open up the product page. Any further operations on products will take action from this page. The delete button is not covered by use cases but we foresee data entry errors and the need for deletions on products. Validations will take place before a product can be deleted.

An extra button is added for duplicating a Product into a new Product. This will be handy as perhaps a new product is more or less similar as another product with just a few changes on perhaps the item/services or pricing parameters.

To duplicate a product:

Search and select the product to be duplicated in the grid. Click the ‘Duplicate’ button. Automatically the system will now open up the product page and insert (in memory) the new product with its details (including items and services) as a duplicate of the original. The operator can make further modifications to the information and could decide to save the ‘duplicated’ product. In this case the normal procedure for saving will take place. If the operator cancels the entry, nothing will be saved.

5.7.4.Product.aspx

Home Insurees and Policies Claims Administration Tools My profile Logout Search Insurance v16.3.0

Insurance Product details

Code	DFB002
Name	Basic free enrolment Doc
District	Dodoma
Date From	01/01/2015
Date To	02/01/2025
Conversion	Click button to load
Lump Sum	20,000.00
Threshold Members	3
Maximum Members	5
Contribution Adult	1,000.00
Contribution Child	500.00
Insurance Period	12
Administration Period	1
Max Installments	3
Grace Period Payment	0
Grace Period Enrolment	0
Grace Period Renewal	0
Renewal Disc. %	
Renewal Disc. Period	
Enrolment Disc. %	

Check All

Medical Services

CODE	NAME	TYPE	HF LEVEL	PRICE	LIMIT O	LIMIT R	LIMIT E	ORIGIN
<input checked="" type="checkbox"/> AOFB01	Antenatal Examination	Preventive	Simple Service	800.00	C	C	C	P
<input checked="" type="checkbox"/> COBB01	Consultation GP	Curative	Visit	200.00	C	C	C	P
<input checked="" type="checkbox"/> DIFB12	Delivery-normal	Curative	Simple Service	8000.00	C	C	C	P
<input checked="" type="checkbox"/> GBBB01	Urine Analysis	Curative	Simple Service	500.00	C	C	C	P

Check All

Medical Items

CODE	NAME	TYPE PACKAGE	PRICE	LIMIT O	LIMIT R	LIMIT E	ORIGIN	ADULT O	ADULT R
<input checked="" type="checkbox"/> GBBB09	Acetylsalicylic Acid	Drug Tabs	300.00	C	C	C	P	100.00	1
<input checked="" type="checkbox"/> GBBC01	Ampicillin PDR	Drug Tabs	400.00	C	C	C	P	100.00	1
<input type="checkbox"/> GBC003	Albendazole Tab.	Drug 100 tablets	500.00	C	C	C	P	100.00	1
<input type="checkbox"/> GBC004	Bendrofluazide	Drug 100 tablets	400.00	C	C	C	P	100.00	1

Account Code(Remuneration) Account Code(Contribution)
 Registration Lump Sum Assembly Lump Sum
 Registration Fee Assembly Fee
 Start Cycle 1 Day Month Start Cycle 3 Day Month
 Start Cycle 2 Day Month Start Cycle 4 Day Month
 Ceiling Interpretation Hospital

Buttons: Save Cancel

[P] = Scheme price [O] = Providers Own price [R] = Relative price

Interface object name:	<i>Product.aspx</i>
Menu path:	
Hyperlinks/Redirections:	<i>FindProduct.aspx (Product Code)</i>
Action Button:	<i>FindProduct.aspx (Add and Edit button)</i>
Use case Reference	<i>6.2.13 (Add Insurance Product) 6.2.15 (Modify Insurance Product)</i>
Class Diagram	<pre> classDiagram class ProductBI { <<Methods>> ChangeProductItems ChangeProductServices GetAllDistributionData GetCeilingInterpritation GetDays GetDistribution GetDistricts getLevels GetMedicalServices GetMonths GetPeriod GetProductItems GetProducts GetRegions getSubLevels LoadProduct SaveProduct SaveRelDistributionPercentageChanges UpdateProductItems UpdateProductServices } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the product search page (or clicked the hyperlink). The product information can be changed as per defined 'use-cases' for adding and editing a product. The product page will have 2 separate sections for the items and services covered in the product. The check all checkbox will allow quick selection of all items or services.

Product Deductibles and Product Ceilings will be defined as per Treatment or Insuree or Policy. (Not a combination)

Furthermore we could define the ceilings and deductions on general level. In that case we are not able to define these parameters also separately for 'in' and -'out' patient level.

The product distribution information is used for the calculation of relative indexes.

The distribution will be set at Monthly, Quarterly or Yearly level (independent for general, in-patient and out-patient). At the moment of changing the setting, new records will be populated (1 for yearly, 4 for quarterly and 12 for monthly). The operator will now use the separate grids (period-percentage) with scroll bar to view and/or edit the distribution percentages.

On saving a product it will be added /refreshed on the data grid within the product search page as described before.

5.7.5.FindMedicalItem.aspx

The screenshot shows a web application interface for finding medical items. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar are search fields for 'Search Insurance' and a version indicator 'v16.3.0'. Below the navigation bar is a section titled 'Select Criteria' with fields for 'Code', 'Name', 'Type' (a dropdown menu), 'Package', and a checkbox for 'Historical' data. A 'Search' button is located at the bottom right of this section. Below this is a header '4 Items Found' followed by a table displaying four medical items. The table columns are CODE, NAME, TYPE, PACKAGE, PRICE, VALID FROM, and VALID TO. The items listed are:

CODE	NAME	TYPE	PACKAGE	PRICE	VALID FROM	VALID TO
GBBB09	Acetyl/Isalicylic Acid	Drug	Tabs	300.00	01/01/2015	
GBBC01	Ampicillin PDR	Drug	Tabs	400.00	01/01/2015	
GBC003	Albendazole Tab.	Drug	100 tablets	500.00	12/12/2016	
GBC004	Bendrofluazide	Drug	100 tablets	400.00	12/12/2016	

At the bottom of the page are buttons for 'Add', 'Edit', 'Delete', and 'Cancel'.

Interface object name:	FindMedicalItem.aspx
Menu path:	Medical Items
Hyperlinks/Redirections:	
Action Button:	MedicalItem.aspx (Save and Cancel button)
Use case Reference	6.2.2(Finding Medical Item) 6.2.4 (Deleting Medical Item)
Class Diagram	<pre> classDiagram class FindMedicalItemsBI { <<Methods>> checkRoles() DeleteItem() GetItemType() GetMedicalItems() GetPatient() } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Medical Items’ button option on the top menu (Administration). This interface is used to search for medical items and acts as the selector for additions, modifications and deletions on the entity Medical Item.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Code) in the grid will open up the medical item page. Any further operations on medical items will take action from this page.

5.7.6.MedicalItem.aspx

Medical Item Details

Code: GBC004
Name: Bendrofluazide
Type: Drug Medical Prostheses
Package: 100 tablets
Price: 400
Care Type: Out-Patient In-Patient Both
Frequency: 0
Patient: Man Woman Adult Child

Save **Cancel**

Interface object name:	MedicalItem.aspx
Menu path:	
Hyperlinks/Redirections:	FindMedicalItem.aspx (Code column)
Action Button:	FindMedicalItem.aspx (Add and Edit button)
Use case Reference	6.2.5 (Add Medical Item) 6.2.7 (Modify Medical Item)
Class Diagram	<pre> classDiagram class MedicalItemBI { <<Class>> <<Methods>> +GetItemType() +GetMedicalItems() +GetMedicalServices() +LoadItem() +SaveMedicalItem() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the Medical Items search page (or clicked the hyperlink). The medical item information can be changed as per defined 'use-cases' for adding and editing a medical item.

On saving a medical item, it will be added /refreshed on the data grid on the medical items search page as described before.

5.7.7.FindMedicalService.aspx

The screenshot shows a web application interface for managing medical services. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar are search fields for 'Search Insurance' and 'v16.3.0'. Below the navigation bar is a section titled 'Select Criteria' with fields for 'Code' and 'Name', a dropdown for 'Type' (with an option for 'Historical'), and a 'Search' button. A message '13 Services Found' is displayed above a grid table. The table has columns: CODE, NAME, TYPE, HF LEVEL, PRICE, VALID FROM, and VALID TO. The data in the table is as follows:

CODE	NAME	TYPE	HF LEVEL	PRICE	VALID FROM	VALID TO
AOFB01	Antenatal Examination	Preventive	Simple Service	800.00	01/01/2015	
CORB01	Consultation GP	Curative	Visit	200.00	01/01/2015	
DEL	Delivery	Curative	Hospital Case	15,000.00	08/11/2016	
DIFB12	Delivery-normal	Curative	Simple Service	8,000.00	01/01/2015	
GBBB01	Urine Analysis	Curative	Simple Service	500.00	01/01/2015	
GBBB02	Gastronomy	Curative	Simple Service	4,000.00	01/01/2015	
GBBX	Tracheostomy	Curative	Simple Service	20,000.00	29/11/2016	
GOMA01	Burst Abdomen	Curative	Simple Service	1,000.00	01/01/2015	
HOSP	Inpatient Hospitalization	Curative	Hospital Case	25,000.00	08/11/2016	
OPD	Outpatient Consultation	Curative	Simple Service	15,000.00	08/11/2016	
SIBB51	Colostomy	Curative	Simple Service	5,000.00	01/01/2015	
SIFA01	Mastectomy	Curative	Simple Service	2,000.00	01/01/2015	
SURG	Surgery	Curative	Hospital Case	70,000.00	08/11/2016	

At the bottom of the grid, there are four buttons: 'Add', 'Edit', 'Delete', and 'Cancel'.

Interface object name:	<i>FindMedicalService.aspx</i>
Menu path:	<i>Medical Services</i>
Hyperlinks/Redirections:	
Action Button:	<i>MedicalService.aspx (Save and Cancel button)</i>
Use case Reference	6.2.2(Finding Medical Service) 6.2.4 (Deleting Medical Service)
Class Diagram	<pre> classDiagram class FindMedicalServicesBI { <<Methods>> checkRoles DeleteService GetMedicalService GetPatient GetServiceLevel GetServiceType } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Medical Services’ button option on the top menu (Administration). This interface is used to search for medical services and acts as the selector for additions, modifications and deletions on the entity Medical Service.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Code) in the grid will open up the medical service page. Any further operations on medical services will take action from this page.

5.7.8.MedicalService.aspx

Medical Service Details:

- Code: DEL
- Name: Delivery
- Type: Curative
- Category: Delivery
- HF Level: Hospital case
- Price: 15,000
- Care Type: Both
- Frequency: 0
- Patient: Man, Woman, Adult, Child

Buttons: Save, Cancel

Interface object name:	MedicalService.aspx
Menu path:	
Hyperlinks/Redirections:	FindMedicalService.aspx (Code column)
Action Button:	FindMedicalService.aspx (Add and Edit button)
Use case Reference	6.2.5 (Add Medical Service) 6.2.7 (Modify Medical Service)
Class Diagram	<pre> classDiagram class MedicalServiceBI { <<Methods>> +GetServiceCategory() +GetServiceLevel() +GetServiceType() +LoadMedicalServices() +SaveMedicalService() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the medical services search page (or clicked the hyperlink). The medical service information can be changed as per defined ‘use-cases’ for adding and editing a medical service.

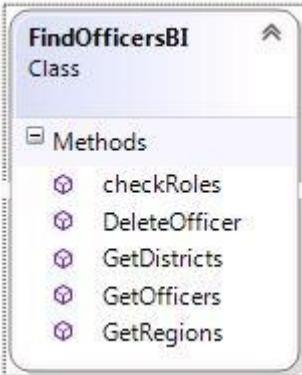
On saving a medical service it will be added /refreshed on the data grid on the medical services search page as described before.

5.7.9.FindOfficer.aspx

The screenshot shows the 'FindOfficer.aspx' page with the following components:

- Header:** Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, Search Insurance No, and Help (?)
- Select Criteria:** A form with fields for Last Name, Other Names, Region, Code, Birth Date From, Birth Date To, District, Email, Phone Number, and a Search button. There is also a checkbox for Historical.
- Results:** A table titled '10 Enrolment Officers Found' displaying 10 rows of data. The columns are: CODE, LAST NAME, OTHER NAMES, BIRTH DATE, PHONE NUMBER, REGION, DISTRICT, VALID FROM, and VALID TO.
- Buttons:** At the bottom are buttons for Add, Edit, Delete, and Cancel.

CODE	LAST NAME	OTHER NAMES	BIRTH DATE	PHONE NUMBER	REGION	DISTRICT	VALID FROM	VALID TO
E001	Fox	James	13/06/1984	+420111111111	TestRegion	TestDistrict1	03/07/2017	
E002	Shark	John	12/06/1984	+420111111111	TestRegion	TestDistrict1	09/05/2017	
E003	James	Ellis	26/12/1988		TestRegion	TestDistrict2	09/05/2017	
E004	Park	Jack	14/06/1994		TestRegion	TestDistrict1	09/05/2017	
E005	Shakespeare	Magi	26/12/1988		TestRegion	TestDistrict3	09/05/2017	
E009	Beck	John	15/06/2004		TestRegion	TestDistrict1	06/07/2017	
E011	Enrolment1	John	01/01/2009	+420899477757	TestRegionX	TestDistrictX1	15/07/2017	
E012	Enrolment2	Jack	14/06/1994		TestRegionX	TestDistrictX2	15/07/2017	
E1001	Ngaiza	Allen	13/05/1985	0768108131	Region	Dodoma	17/05/2017	

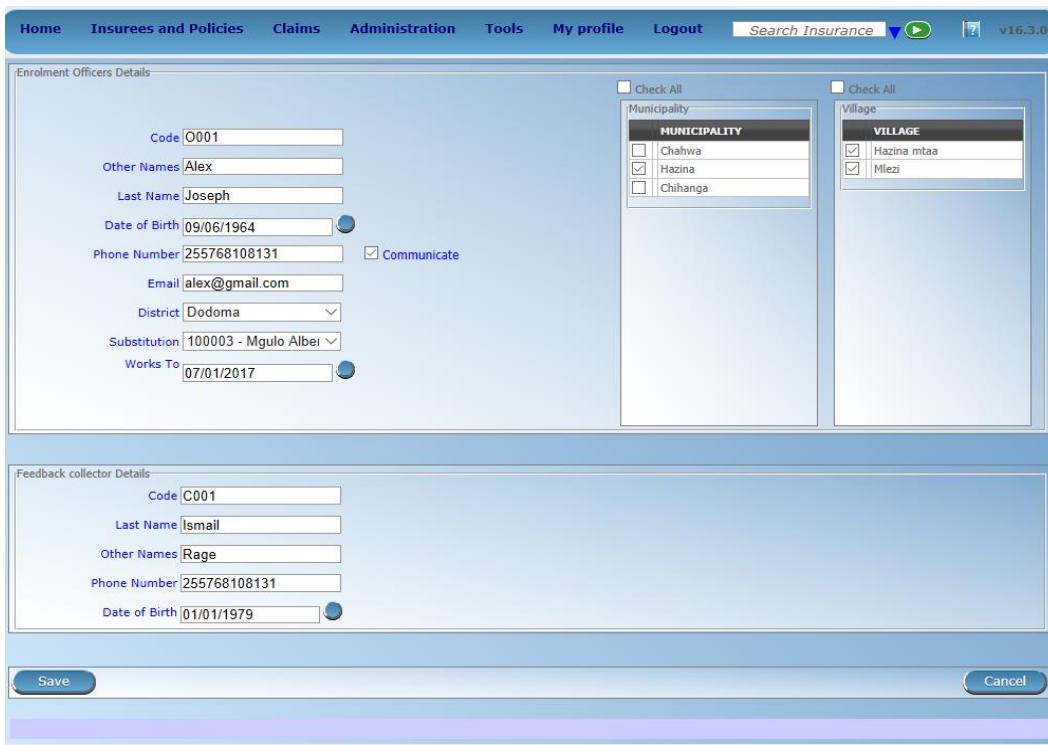
Interface object name:	FindOfficer.aspx
Menu path:	Enrollment Assistant
Hyperlinks/Redirections:	
Action Button:	Officer.aspx (Save and Cancel button)
Use case Reference	6.2.10(Finding Officers) 6.2.12 (Deleting Officers)
Class Diagram	 <pre> classDiagram class FindOfficersBI { <<Methods>> checkRoles() DeleteOfficer() GetDistricts() GetOfficers() GetRegions() } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Officers’ button option on the top menu (Administration). This interface is used to search for officers and acts as the selector for additions, modifications and deletions on the entity Enrolment Officer.

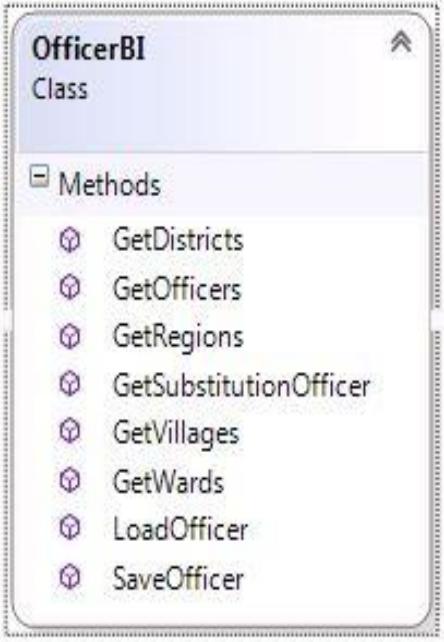
The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Code) in the grid will open up the officer’s page. Any further operations on officers will take action from this page.

5.7.10. Officer.aspx



The screenshot shows the 'Officer.aspx' page with the following details:

- Enrolment Officers Details:**
 - Code: 0001
 - Other Names: Alex
 - Last Name: Joseph
 - Date of Birth: 09/06/1964
 - Phone Number: 255768108131
 - Email: alex@gmail.com
 - District: Dodoma
 - Substitution: 100003 - Mgulu Alberi
 - Works To: 07/01/2017
- Municipality:** Chalwa, Hazina, Chihanga
- Village:** Hazina mtaa, Mlezi
- Feedback collector Details:**
 - Code: C001
 - Last Name: Ismail
 - Other Names: Rage
 - Phone Number: 255768108131
 - Date of Birth: 01/01/1979
- Buttons:** Save, Cancel

Interface object name:	Officer.aspx
Menu path:	
Hyperlinks/Redirections:	FindOfficer.aspx (Code column)
Action Button:	FindOfficer.aspx (Add and Edit button)
Use case Reference	6.2.9 (Add Officer) 6.2.11 (Modify officer)
Class Diagram	 <pre> classDiagram class OfficerBI { <<Methods>> +GetDistricts() +GetOfficers() +GetRegions() +GetSubstitutionOfficer() +GetVillages() +GetWards() +LoadOfficer() +SaveOfficer() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the officer search page (or clicked the hyperlink). The officer information can be changed as per defined 'use-cases' for adding and editing an officer.

On saving an officer, he/she will be added /refreshed on the data grid on the officer search page as described before.

5.7.11. FindPayer.aspx

The screenshot shows the 'FindPayer.aspx' page. At the top, there's a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, and a search bar for 'Search Insurance No'. Below the navigation is a 'Select Criteria' section with fields for Name, Region, District, Email, Type, Phone Number, and a checkbox for 'Historical'. A 'Search' button is also present. The main area displays a table titled '4 Payers Found' with columns: NAME, TYPE, ADDRESS DETAILS, REGION, DISTRICT, PHONE NUMBER, VALID FROM, and VALID TO. The data in the table is as follows:

NAME	TYPE	ADDRESS DETAILS	REGION	DISTRICT	PHONE NUMBER	VALID FROM	VALID TO
PORALG	G	Dodoma, str. Liberty	National		+312456345	08/05/2017	
Provincial office Region	L	Dodoma, street Victory	National		+2345516666	08/05/2017	
RichDonor	D	str. Stars	TestRegionX		+420345678899	15/07/2017	
District office Dodoma	L	Dodoma, str. Station	Region	Dodoma	+324456789	08/05/2017	

At the bottom of the page are buttons for Add, Edit, Delete, and Cancel.

Interface object name:	FindPayer.aspx
Menu path:	Payers
Hyperlinks/Redirections:	
Action Button:	Payer.aspx (Save and Cancel button)
Use case Reference	6.2.6(Finding Payers) 6.2.8 (Deleting Payers)
Class Diagram	<pre> classDiagram class FindPayersBI { <<Methods>> checkRoles() DeletePayer() GetDistricts() GetPayers() GetPayerType() GetRegions() } </pre>

Brief description:

This interface opens up when the user has clicked the 'Payers' button option on the top menu (Administration). This interface is used to search for payers and acts as the selector for additions, modifications and deletions on the entity Payer.

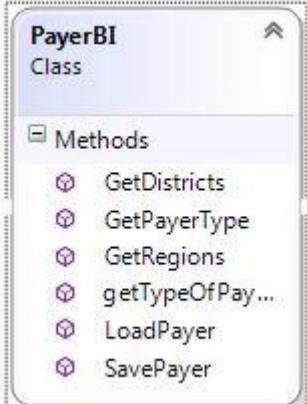
The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Name... **Note**. to be changed in the field order) in the grid will open up the payers page. Any further operations on payers will take action from this page.

5.7.12. Payer.aspx

The screenshot shows a web application interface for managing payers. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar are search and filter icons, and the text "v1.6.3.0". Below the navigation bar is a header titled "Payer Details". The form contains the following fields:

- Type: Government (dropdown)
- Name: PORALG (text input)
- Address: Dodoma, Victory street (text input)
- Address: (empty text input)
- District: National (dropdown)
- Phone Number: +255 (0) 4566 5667 (text input)
- Fax: (empty text input)
- Email: poralg@gov.org.tz (text input)

At the bottom of the form are two buttons: "Save" and "Cancel".

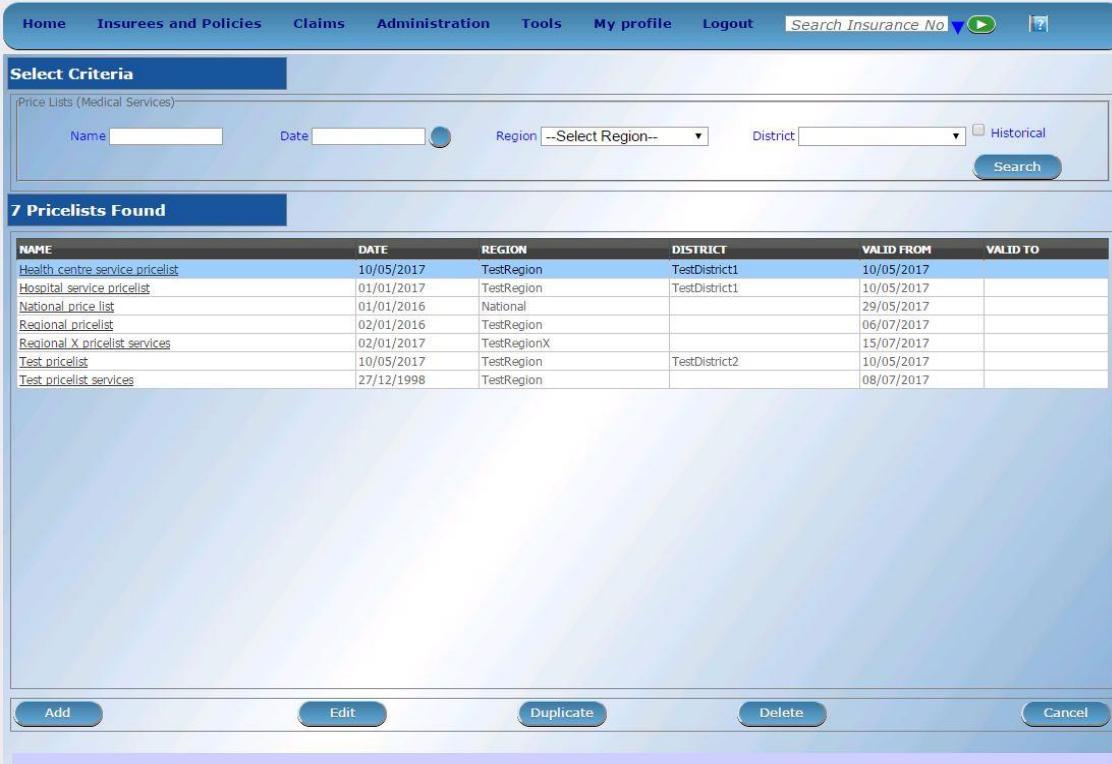
Interface object name:	Payer.aspx
Menu path:	
Hyperlinks/Redirections:	FindPayer.aspx (Name column)
Action Button:	Findpayer.aspx (Add and Edit button)
Use case Reference	6.2.5 (Add payer) 6.2.7 (Modify payer)
Class Diagram	 <pre> classDiagram class PayerBI { <<Methods>> GetDistricts() GetPayerType() GetRegions() getTypeOfPay... LoadPayer() SavePayer() } </pre>

Brief description:

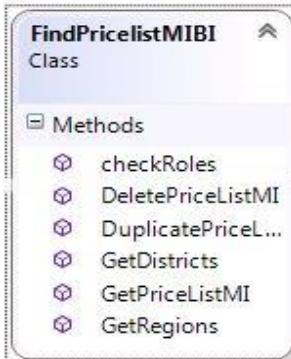
This interface opens up when the user has clicked the Add or Edit button on the payer search page (or clicked the hyperlink). The payer information can be changed as per defined 'use-cases' for adding and editing a Payer.

On saving a payer, the new entry will be added /refreshed on the data grid on the payer's search page as described before.

5.7.13. FindPriceListMI.aspx



NAME	DATE	REGION	DISTRICT	VALID FROM	VALID TO
Health centre service pricelist	10/05/2017	TestRegion	TestDistrict1	10/05/2017	
Hospital service pricelist	01/01/2017	TestRegion	TestDistrict1	10/05/2017	
National price list	01/01/2016	National		29/05/2017	
Regional pricelist	02/01/2016	TestRegion		06/07/2017	
Regional X pricelist services	02/01/2017	TestRegionX		15/07/2017	
Test pricelist	10/05/2017	TestRegion	TestDistrict2	10/05/2017	
Test pricelist services	27/12/1998	TestRegion		08/07/2017	

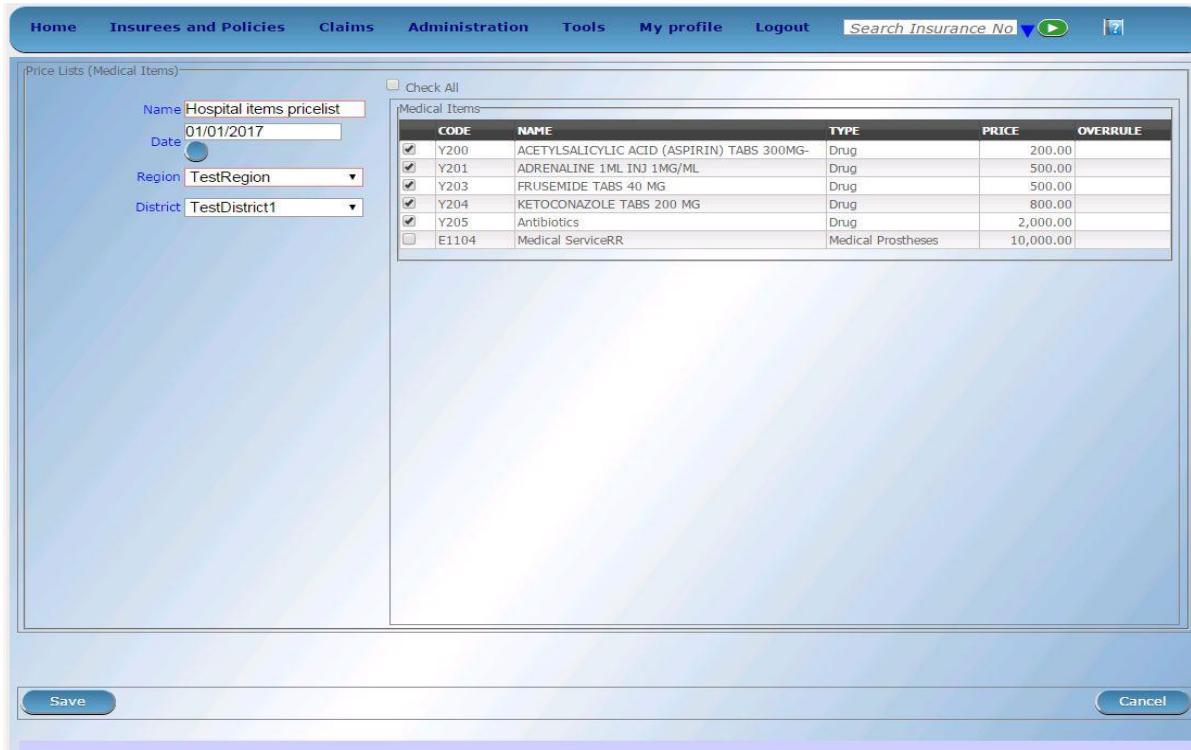
Interface object name:	FindPricelistMI.aspx
Menu path:	Pricelists - Medical Items
Hyperlinks/Redirections:	
Action Button:	PriceListMI.aspx (Save and Cancel button)
Use case Reference	6.2.21(Finding Pricelist medical Items) 6.2.23 (Deleting Pricelist medical items)
Class Diagram	 <pre> classDiagram class FindPricelistMIBI { <<Methods>> checkRoles() DeletePriceListMI() DuplicatePriceL... GetDistricts() GetPriceListMI() GetRegions() } </pre>

Brief description:

This interface opens up when the user has clicked the ‘Medical Items’ button option on the top menu Administration, sub menu Pricelists. This interface is used to search for Pricelists of medical items and acts as the selector for additions, modifications and deletions on the entity Pricelist of Items.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Name) in the grid will open up the pricelist medical items page. Any further operations on pricelist of medical items will take action from this page.

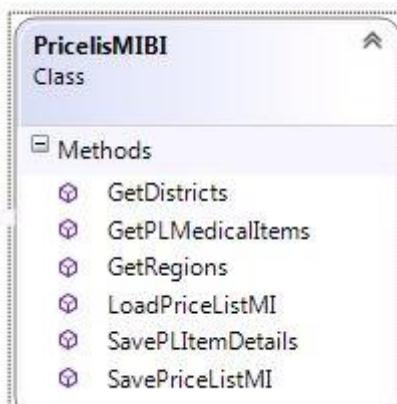
5.7.14. PriceListMI.aspx



The screenshot shows the 'Price Lists (Medical Items)' page. At the top, there are search filters: Name (Hospital items pricelist), Date (01/01/2017), Region (TestRegion), and District (TestDistrict1). Below the filters is a 'Check All' checkbox and a table titled 'Medical Items'. The table has columns: CODE, NAME, TYPE, PRICE, and OVERRULE. The data in the table is as follows:

CODE	NAME	TYPE	PRICE	OVERRULE
Y200	ACETYL SALICYLIC ACID (ASPIRIN) TABS 300MG-	Drug	200.00	
Y201	ADRENALINE 1ML INJ 1MG/ML	Drug	500.00	
Y203	FRUSEMIDE TABS 40 MG	Drug	500.00	
Y204	KETOCONAZOLE TABS 200 MG	Drug	800.00	
Y205	Antibiotics	Drug	2,000.00	
E1104	Medical ServiceRR	Medical Prostheses	10,000.00	

At the bottom of the page are 'Save' and 'Cancel' buttons.

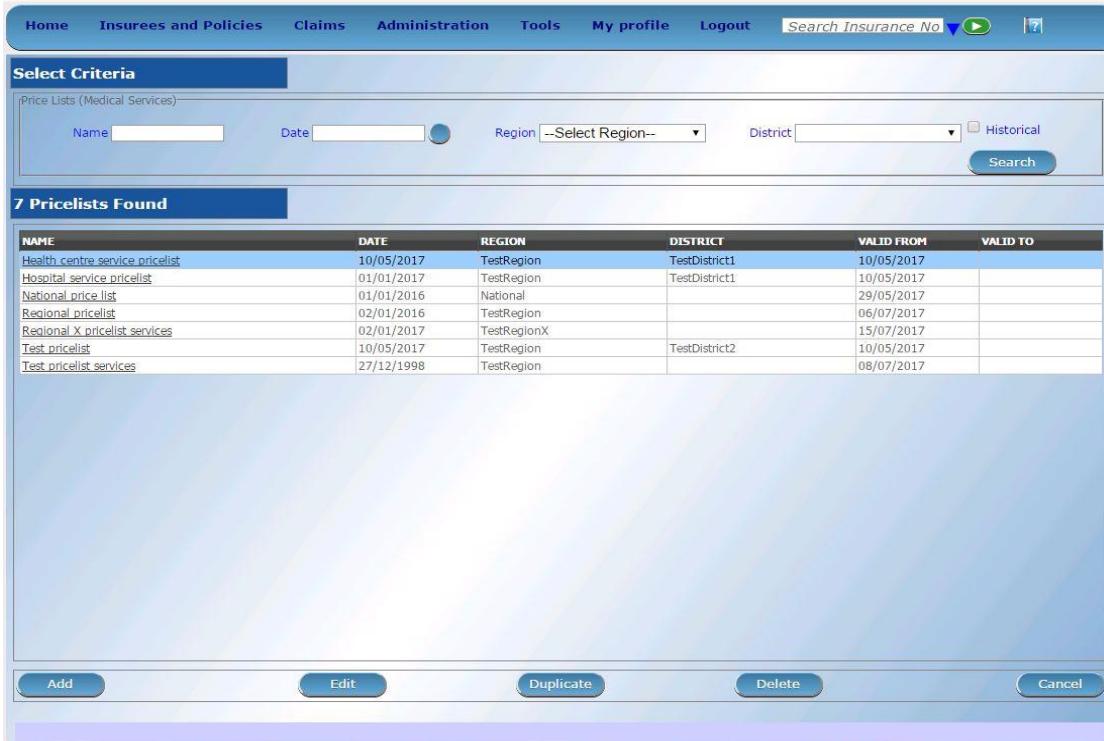
Interface object name:	PriceListMI.aspx
Menu path:	
Hyperlinks/Redirections:	FindPricelistMI.aspx (Name column)
Action Button:	FindPriceListMI.aspx (Add and Edit button)
Use case Reference	6.2.20 (Add Pricelist Medical Items) 6.2.22 (Modify Pricelist Medical Items)
Class Diagram	 <pre> classDiagram class PricelisMIBI { <<Methods>> GetDistricts() GetPLMedicalItems() GetRegions() LoadPriceListMI() SavePLItemDetails() SavePriceListMI() } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the Pricelist medical items search page (or clicked the hyperlink). The pricelist (of medical items) information can be changed as per defined 'use-cases' for adding and editing a pricelist of medical items.

On saving a pricelist of medical items, the new entry will be added /refreshed on the data grid on the pricelist of medical items search page as described before.

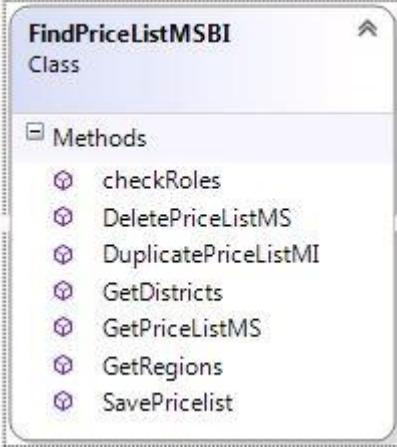
5.7.15. FindPriceListMS.aspx



The screenshot shows the 'FindPriceListMS.aspx' page. At the top, there is a navigation bar with links: Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation bar is a search bar labeled 'Search Insurance No.' with a dropdown arrow and a magnifying glass icon. Below the navigation bar is a 'Select Criteria' section with a blue header. It contains fields for 'Name' (text input), 'Date' (date input), 'Region' (dropdown menu with placeholder '--Select Region--'), 'District' (dropdown menu with placeholder 'Historical'), and a 'Search' button. Below this section is a table titled '7 Pricelists Found' with a blue header. The table has columns: NAME, DATE, REGION, DISTRICT, VALID FROM, and VALID TO. The data in the table is as follows:

NAME	DATE	REGION	DISTRICT	VALID FROM	VALID TO
Health centre service pricelist	10/05/2017	TestRegion	TestDistrict1	10/05/2017	
Hospital service pricelist	01/01/2017	TestRegion	TestDistrict1	10/05/2017	
National price list	01/01/2016	National		29/05/2017	
Regional pricelist	02/01/2016	TestRegion		06/07/2017	
Regional X pricelist services	02/01/2017	TestRegionX		15/07/2017	
Test pricelist	10/05/2017	TestRegion	TestDistrict2	10/05/2017	
Test pricelist services	27/12/1998	TestRegion		08/07/2017	

At the bottom of the page are five buttons: Add, Edit, Duplicate, Delete, and Cancel.

Interface object name:	<i>FindPricelistMS.aspx</i>
Menu path:	<i>Pricelists - Medical Services</i>
Hyperlinks/Redirections:	
Action Button:	<i>PriceListMS.aspx (Save and Cancel button)</i>
Use case Reference	<i>6.2.21(Finding Pricelist medical services) 6.2.23 (Deleting Pricelist medical services)</i>
Class Diagram	 <pre> classDiagram class FindPriceListMSBI { <<Methods>> checkRoles DeletePriceListMS DuplicatePriceListMI GetDistricts GetPriceListMS GetRegions SavePricelist } </pre>

Brief description:

This interface opens up when the user has clicked the 'Medical Services' button option on the top menu Administration, sub menu PriceLists. This interface is used to search for PriceLists of medical services and acts as the selector for additions, modifications and deletions on the entity Pricelist of Services.

The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Name) in the grid will open up the pricelist medical services page. Any further operations on pricelist of medical services will take action from this page.

5.7.16. PriceListMS.aspx

The screenshot shows a web application interface for managing medical service price lists. At the top, there's a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for 'Insurance No' is also present. The main content area is titled 'Price Lists (Medical Services)'. It includes a form with fields for Name ('Health centre service price'), Date ('10/05/2017'), Region ('TestRegion'), and District ('TestDistrict1'). There's a checkbox labeled 'Check All' and a table titled 'Medical Services' with the following data:

CODE	NAME	TYPE	PRICE	OVERRULE
X106	Urinary lab test	Curative	200.00	
X108	Antenatal examination	Preventive	100.00	
X109	GP visit	Curative	100.00	
X100	Burst Abdomen	Curative	1,000.00	
X101	Cholecystectomy	Curative	2,000.00	
X102	Caesarean Section Delivery	Curative	500.00	
X103	Delivery-Normal	Curative	300.00	
X105	Mastectomy	Curative	2,000.00	
X107	Tubectomy	Curative	300.00	
X999	Testé	Curative	1,000.00	

At the bottom, there are 'Save' and 'Cancel' buttons.

Interface object name:	PriceListMS.aspx
Menu path:	
Hyperlinks/Redirections:	FindPricelistMS.aspx (Name column)
Action Button:	FindPriceListMS.aspx (Add and Edit button)
Use case Reference	6.2.20 (Add Pricelist Medical Services) 6.2.22 (Modify Pricelist Medical Services)
Class Diagram	<pre> classDiagram class PricelistMSBI { <<Methods>> GetDistricts GetMedicalServices (+ 1 overload) GetRegions LoadPriceListMS SavePLServicesDetail SavePriceListMS } </pre>

Brief description:

This interface opens up when the user has clicked the Add or Edit button on the Pricelist medical services search page (or clicked the hyperlink). The pricelist (of medical services) information can be changed as per defined 'use-cases' for adding and editing a pricelist of medical services.

On saving a pricelist of medical services, the new entry will be added /refreshed on the data grid on the pricelist of medical services search page as described before.

5.7.17. FindUser.aspx

The screenshot shows the 'FindUser.aspx' page with a 'Select Criteria' section at the top containing fields for Last Name, Other Names, Region, Login Name, Roles, District, Phone Number, HF Name, Language, and Email. A 'Search' button is also present. Below this, a table titled '38 Users Found' displays user information across six columns: LOGIN NAME, LAST NAME, OTHER NAMES, PHONE NUMBER, VALID FROM, and VALID TO. The table lists users such as Admin, Alex, Damien, Revault, etc., with their respective details. At the bottom of the table are navigation links (1, 2, 3) and buttons for Add, Edit, Delete, and Cancel.

LOGIN NAME	LAST NAME	OTHER NAMES	PHONE NUMBER	VALID FROM	VALID TO
Admin	Admin	Admin		21/04/2017	
Alex	Alex	Alex		05/07/2017	
damien	Revault	Damien		20/12/2016	
erwezaura	Rwezaura	Elizeus		07/11/2016	
exact	Software	Exact	0768108131	17/07/2017	
george	George	Atohmboom Yuh		04/11/2016	
georgebda	Yuh George	Atohmboon		11/11/2016	
gideon	Christopher	Gideon		23/02/2017	
gsimon	Simon	Gidion		08/11/2016	
hans	hgfsch	Hoppe		04/07/2017	
help	soni	Whitesonč	075661661	31/07/2017	
hiren	Soni	Hiren		06/06/2017	
jean	Charles	Jean		21/04/2017	
jiri	Nemec	Jiri		25/07/2017	
jriAccountant	Nemec	Jiri		15/07/2017	

Interface object name:	FindUser.aspx
Menu path:	Users
Hyperlinks/Redirections:	
Action Button:	User.aspx (Save and Cancel button)
Use case Reference	6.2.2(Finding Users) 6.2.4 (Deleting Users)
Class Diagram	<pre> classDiagram class FindUsersBI { <<Methods>> checkRoles DeleteUser GetDistricts GetHFCodes GetLanguage GetRegions GetUserRoles GetUsers } </pre>

Brief description:

This interface opens up when the user has clicked the 'Users' button option on the top menu Administration. This interface is used to search for users and acts as the selector for additions, modifications and deletions on the entity User.

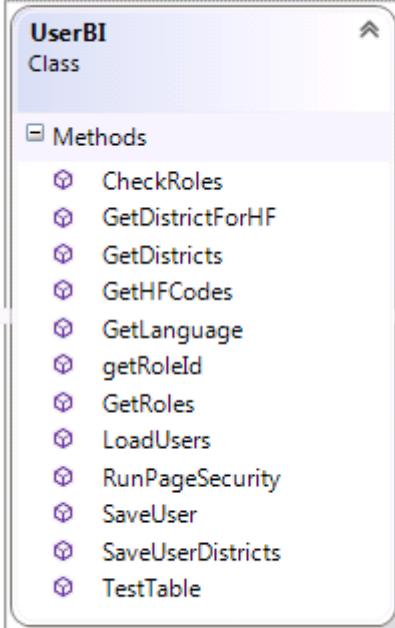
The operator can enter the search criteria and click the search button. All records satisfying the criteria will appear. A click on the hyperlink on the first column (Login Name) in the grid will open up 'user' page. Any further operations on users will take action from this page.

5.7.18. User.aspx

The screenshot shows the 'User.aspx' page with the following components:

- User Details:** A form containing fields for Language (English), Other Names (Jiri), Last Name (Nemec), Phone Number, Email (jirneme@volny.cz), Login Name (jiri), Password, Confirm Password, and HF Name (--- Select HF Code ---).
- Role Selection:** A list of checkboxes for various roles: Enrolment Officer, Manager, Accountant, Clerk, Medical Officer, Scheme Administrator, IMIS Administrator, Receptionist, Claim Administrator, and Claim Contributor. The 'Check All' checkbox is checked.
- Region Selection:** A list of checkboxes for regions: Region, North West, South West, and Central. The 'Check All' checkbox is checked.
- District Selection:** A list of checkboxes for districts: Dummy, Kongwa, Bahi, Dodoma, Basel, Kilosa, Mpapwa, Kailali, Msalala, Bamenda, Bambui, Kumbo, Buea, Mamfe, and Morogoro. The 'Check All' checkbox is checked.
- Action Buttons:** At the bottom are 'Save' and 'Cancel' buttons.

Interface object name:	User.aspx
Menu path:	
Hyperlinks/Redirections:	FindUser.aspx (Name column)
Action Button:	FindUser.aspx (Add and Edit button)
Use case Reference	6.2.1 (Add User) 6.2.3 (Modify User)

Class Diagram**Brief description:**

This interface opens up when the user has clicked the Add or Edit button on the users search page (or clicked the hyperlink). The user information can be changed as per defined 'use-cases' for adding and editing a user.

On saving a user, the new entry will be added /refreshed on the data grid on the users search page as described before.

5.7.19. Locations.aspx

The screenshot shows the 'Locations.aspx' page with four main sections:

- 7 Region(s)**: A table with rows for Region (100), North West, South West, Central, Capital, TestRegion, and Singida.
- 11 District(s)**: A table with rows for Bahi, Bamenda, Basel, Dodoma, Dummy, Kailali, Kilosa, Kongwa, Morogoro, Mpapwa, and Msalala.
- 2 Municipality(s)**: A table with rows for 10011, Babayu, and Chali.
- 2 Village(s)**: A table with rows for Asanje and Babayu.

At the bottom, there are buttons for Add, Edit, Delete, Move, and Cancel.

Interface object name:	Locations.aspx
Menu path:	Locations
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	6.2.24 (Uploading code lists)
Class Diagram	<pre> classDiagram class LocationsBI { <<Methods>> checkRoles DeleteLocation GetDistricts GetRegions GetVillages GetWards IsLocCodeUnique SaveLocation UploadLocations } </pre>

Brief description:

This interface opens up when the user has clicked the menu option 'Locations' in the Administration menu. Although no real use-case has been involved with this screen, we feel it is better to maintain the hierarchy of Districts, Municipality and Villages manually after the first upload. This first upload will be initiated by the developer on the basis of a template provided in excel. Hereafter, any change will be performed by the use of this interface. By selecting the district, the Municipality will automatically refresh and displayed in the Municipality section. The focus will always be set to the first ward. By now selecting any ward, automatically all defined villages will appear in the villages section.

One could add, edit or (logically) delete districts, Municipality and villages. For adding and editing, the user can directly enter codes in the grids. The normal rules for archiving records will apply. Deletion will only be performed by flagging the record with the validity to field (date stamp).

This screen is only available to the IMIS administrator.

5.7.20. MoveLocation.aspx

The screenshot shows the 'MoveLocation.aspx' page with the following interface elements:

- Top Navigation Bar:** Home, Insurees and Policies, Claims, Administration, Tools, My profile, Logout, Search Insurance No, and Help (?)
- Region Selection:** A panel on the left with dropdown menus for Region (selected: Region), District (selected: Bamenda), and Municipality (selected: Azire). Below this is a table for VILLAGE CODE and VILLAGE, listing Bali Park and Church Center.
- District Selection:** A panel showing Region North West and District Kumbo. Below this is a table for MUNICIPALITY CODE and MUNICIPALITY, listing Bui and Donga-Mantung.
- Municipality Selection:** A panel showing Region South West. Below this is a table for DISTRICT CODE and DISTRICT, listing Duea and Mamfe.
- Region List:** A panel on the right showing a table for REGION CODE and REGION, listing 100 (Region, North West, South West, Central, Capital), 20 (TestRegion), and 150 (Singida).
- Buttons:** A 'Cancel' button at the bottom right.

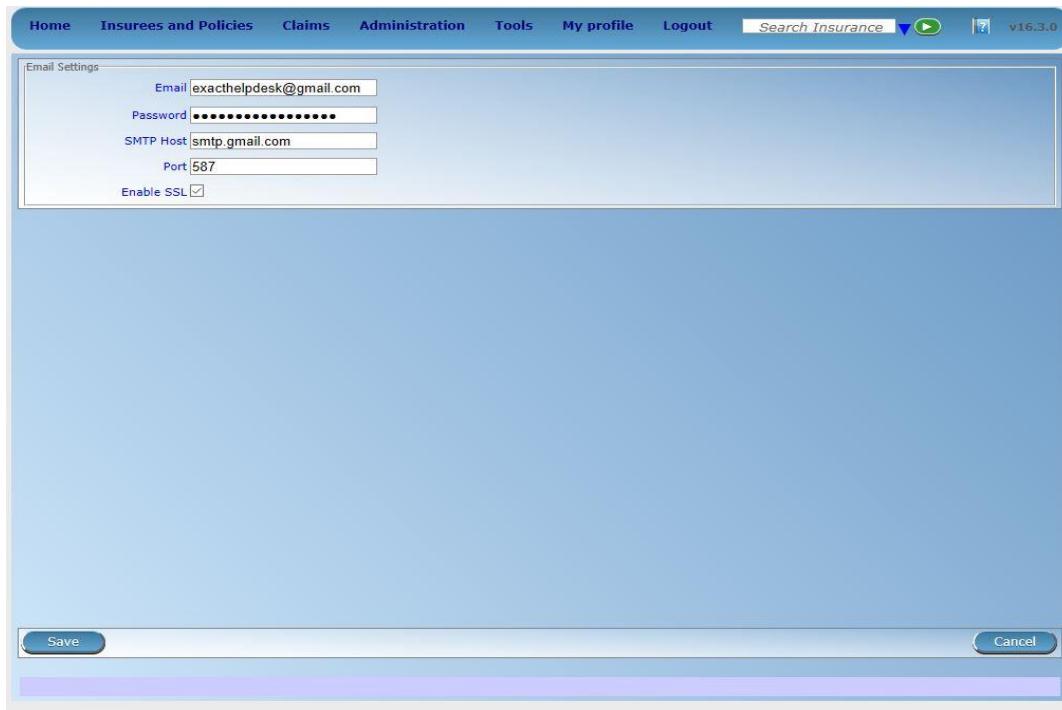
Interface object name:	MoveLocation.aspx
Menu path:	Move Location
Hyperlinks/Redirections:	N/A
Action Button:	Move click on Locations page
Use case Reference	N/A
Class Diagram	 <pre> classDiagram class MoveLocationBI { <<Methods>> GetDistricts() GetRegions() GetVillages() GetWards() MoveLocation() } </pre>

Brief description:

This interface opens up when the user has clicked the move button on Locations page. The arrow button in between will be used move from one location to another location. User must select the source and the destination Location.

Village can be moved from one Municipality to another Municipality, Musicality can be moved from one district to another district and district can be moved from one region to another region.

5.7.21. EmailSettings.aspx



Email Settings

Email: exacthelpdesk@gmail.com

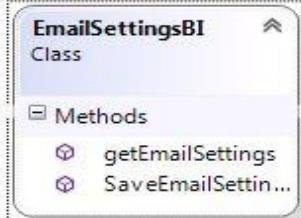
Password: [REDACTED]

SMTP Host: smtp.gmail.com

Port: 587

Enable SSL:

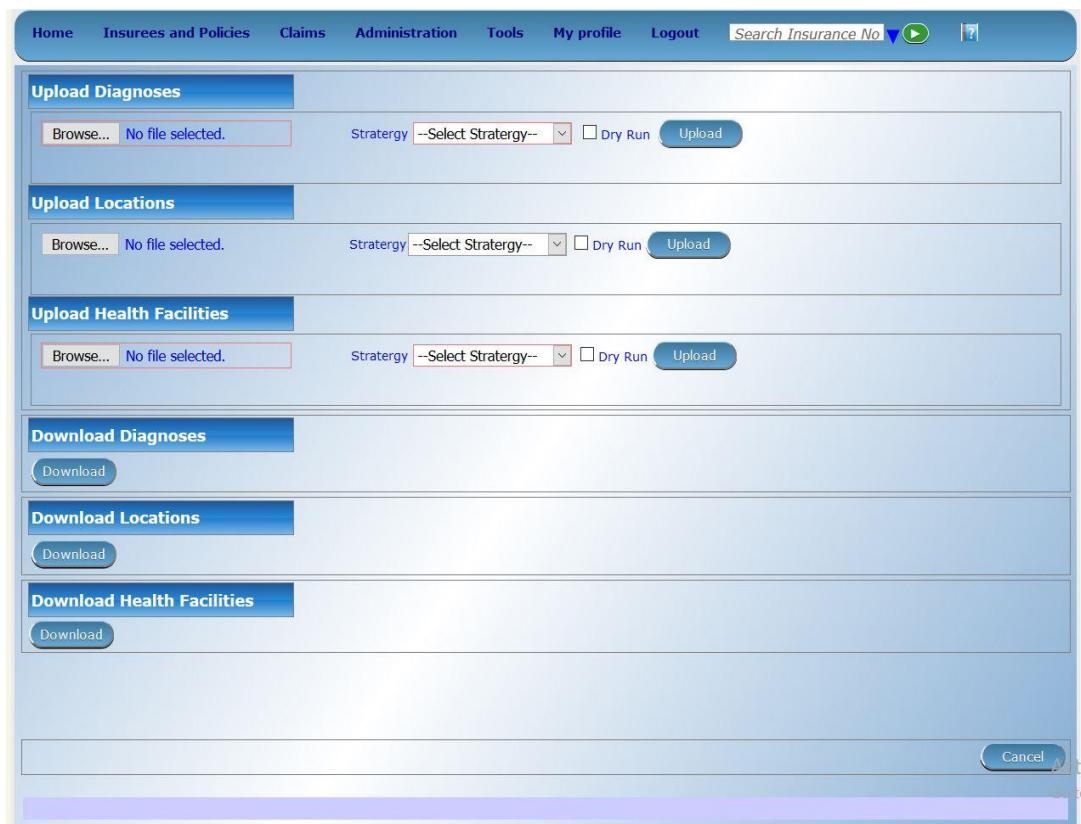
Save Cancel

Interface object name:	EmailSettings.aspx
Menu path:	Email Setting
Hyperlinks/Redirections:	N/A
Action Button:	N/A
Use case Reference	N/A
Class Diagram	 <pre> classDiagram class EmailSettingsBI { <<Methods>> getEmailSettings() SaveEmailSetting() } </pre>

Brief description:

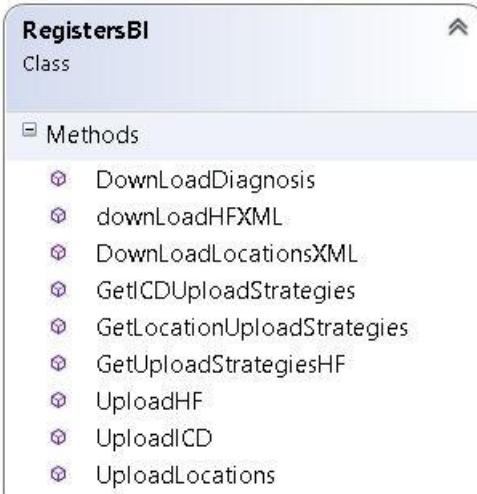
This interface opens up when the user has clicked the menu option 'EmailSetting' in the 'Administration' menu. The save button will change default email setup this include email address (will be sender email), Password this is the email password.

5.7.22. Registers.aspx



The screenshot shows the 'Registers.aspx' page with the following sections:

- Upload Diagnoses:** Includes a 'Browse...' button, a message 'No file selected.', a 'Strategy' dropdown set to '--Select Strategy--', a 'Dry Run' checkbox, and an 'Upload' button.
- Upload Locations:** Similar to the Diagnoses section, with a 'Browse...' button, a message 'No file selected.', a 'Strategy' dropdown, a 'Dry Run' checkbox, and an 'Upload' button.
- Upload Health Facilities:** Similar to the Diagnoses section, with a 'Browse...' button, a message 'No file selected.', a 'Strategy' dropdown, a 'Dry Run' checkbox, and an 'Upload' button.
- Download Diagnoses:** Includes a 'Download' button.
- Download Locations:** Includes a 'Download' button.
- Download Health Facilities:** Includes a 'Download' button.

Interface object name:	Registers.aspx
Menu path:	Upload Main DG List
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	6.2.24 (Uploading code lists)
Class Diagram	 <pre> classDiagram class RegistersBI { <<Methods>> DownLoadDiagnosis downLoadHFXML DownLoadLocationsXML GetICDUploadStrategies GetLocationUploadStrategies GetUploadStrategiesHF UploadHF UploadICD UploadLocations } </pre>

Brief description:

This interface opens up when the user has clicked the menu option ‘Upload Main Dg List’ in the ‘Tools’ menu. The first upload will be initiated by the developer on the basis of an Excel file layout to be provided. After the initial upload, IMIS can be uploaded with a new (updated) list via this interface. The operator will have to select the location of the ICD file via the button Select file. By clicking the Upload button, the system will first verify if the file layout is correct. In case there is an issue with the actual Dg file, an operator message will be displayed. In case the file is valid to be uploaded, the file will be ‘consumed’ by IMIS and new ICD codes will be amended.

The upload will work in such a way that new codes will be added to the system, changes in descriptions of codes will be updated with an audit record automatically generated, or deleted if the code is in the current list but not included in the list to be uploaded. Deletions are performed by setting the validity to field.

The layout of the excel file will be provided by the developer and will have an additional column for codes that would still exist but would need an update in the description.

This feature is only available to the IMIS administrator

5.7.23. PolicyRenewals.aspx

The screenshot shows the 'PolicyRenewals.aspx' page. At the top, there's a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of the navigation is a search bar labeled 'Search Insurance' with a dropdown arrow, a magnifying glass icon, and a help icon. Next to it is the text 'v16.3.0'. Below the navigation is a form titled 'Select Criteria' with the following fields:

- Preview Journal
- Policy Status: Idle (dropdown menu)
- District: -- Select a District -- (dropdown menu)
- Municipality: (dropdown menu)
- Village: (dropdown menu)
- Enrolment Officer: -- Select Enrolment C (dropdown menu)
- Date From: (text input field)
- Date To: (text input field)
- Send SMS: (button)

At the bottom of the page are three buttons: 'Update', 'Preview', and 'Cancel'.

Interface object name:	PolicyRenewals.aspx
Menu path:	Policy Renewals
Hyperlinks/Redirections:	Automatic renewal prompts (SMS via web-service)
Action Button:	
Use case Reference	5.2.17 (prompting for policy renewal)
Class Diagram	<pre> classDiagram class PolicyRenewalBI { <<Methods>> GetJournalOn() GetPolicyPromptJournal() GetPolicyStatus() GetSMSStatus() sendSMS() UpdatPolicyRenewal() } </pre>

Brief description:

This interface opens up when the user has clicked the menu option 'Policy Renewals' in the Tools menu. The filter options allow the operator to filter the district, village, enrollment officer and a specific period for the policy renewal report or renewal 'journal'. The Policy renewal report (via preview button) will contain all policies that will expire between the 'date from' and 'date to' for all officers or a specific officer. The layout for this report is shown below and is grouped by officer.

The screenshot shows the 'Policy Status Overview' page. At the top, there's a search bar with 'Search Insurance' and a date range 'Date From 01/01/2015 To 31/12/2020, Policy Status: Idle'. Below the search bar, there are several sections for different locations:

- Dodoma**: Shows no policies.
- Agent: -**: Shows no policies.
- Chahwa**: Shows no policies.
- Chahwa mtaa**: This section contains a table of policy renewals. The columns are: Insurance Number, Last Name, Other Names, Code, Name, Renewal Date, and Product Value. The data includes entries for Sawmeli, Asma Said, and various policy codes like DFB002, SFB003, etc., with renewal dates ranging from 01/11/2017 to 01/02/2018 and product values from 16,0 to 22,0.
- Ward (Chahwa mtaa)**: Shows a total value of 631,0.
- VDC/Municipality (Chahwa)**: Shows a total value of 631,0.
- Chihanga**: Shows no policies.
- Nzasa**: Shows no policies.
- Chahwa mtaa**: This section contains another table of policy renewals. The columns are: Insurance Number, Last Name, Other Names, Code, Name, Renewal Date, and Product Value. It shows entries for Thadei, Gideon Saidi, and policy codes DXC001, DXC002, etc., with renewal dates on 01/01/2018 and product values of 60,0 and 38,0.

The Journal button will provide a report on all renewal prompts processed by the system. The report will be based on information kept in the tables: tblPolicyRenewals and tblPolicyRenewalsDetails

These tables are populated with data from a background running service that on daily basis will insert policies that expire within @Prompt Days (variable set at in web.config). The service will generate an entry for each expired policy and will also maintain information on photographs that need to be renewed considering the rules.

Another service will sent SMS messages to the phones of enrollment officers as described in use case 5.2.17. An extra status for SMS (not in current design) will be added to the design of the report selection screen to be able to filter out:

- Renewals not sent to enrollment officers phone (no phone number in registration record of the officer)
- Renewals unsuccessfully sent (problem with phone number?)
- Renewals successfully sent

5.7.24. IMIS Extracts

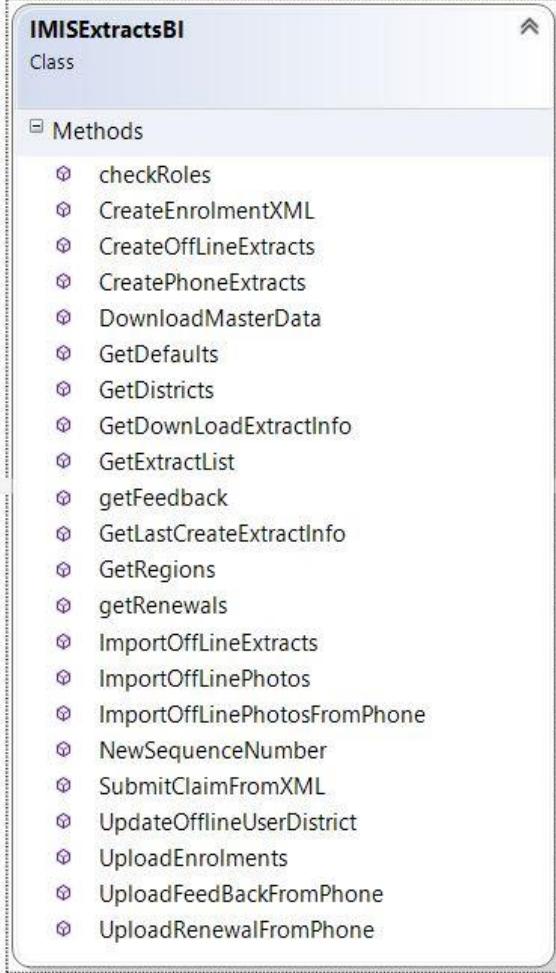
This page opens in two different modes depending on the type of IMIS installation: IMIS Central (live server) or IMIS Offline (installed on local network in a health facility or local network in IMIS).

IMIS Extracts (ONLINE MODE)

The screenshot displays the openIMIS WEB APPLICATION interface version 1.2.0. The top navigation bar includes links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. A search bar for Insurance No is also present. Below the navigation, there are four main sections:

- Create Phone Extract:** Includes dropdowns for Region and District, checkboxes for "With Insuree" and "In background", and a "Create" button.
- Create Offline Extract:** Similar to the Phone Extract section, with additional download buttons for photos and extracts labeled D, E, and F.
- Upload Claims XML:** A file upload form with a "Choose File" input and a "No file chosen" message, followed by a "Upload" button.
- Upload Enrolments:** Another file upload form with a "Choose File" input and a "No file chosen" message, followed by a "Upload" button.

A "Cancel" button is located at the bottom right of the main content area.

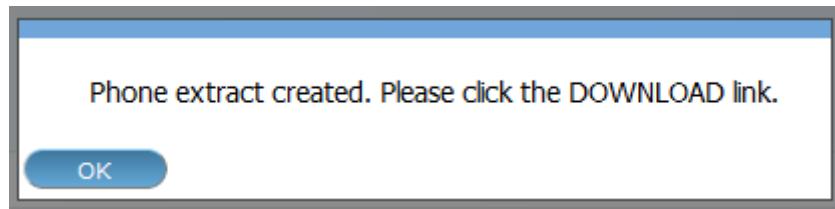
Interface object name:	<i>IMISEtracts.aspx</i>
Menu path:	<i>IMIS extracts(Tools menu)</i>
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	4.2.2 (<i>Downloading data to an off-line client</i>)
Class Diagram	 <pre> classDiagram class IMISEtractsBI { <<Methods>> checkRoles CreateEnrolmentXML CreateOffLineExtracts CreatePhoneExtracts DownloadMasterData GetDefaults GetDistricts GetDownLoadExtractInfo GetExtractList getFeedback GetLastCreateExtractInfo GetRegions getRenewals ImportOffLineExtracts ImportOffLinePhotos ImportOffLinePhotosFromPhone NewSequenceNumber SubmitClaimFromXML UpdateOfflineUserDistrict UploadEnrolments UploadFeedBackFromPhone UploadRenewalFromPhone } </pre>

A-Phone Extract panel

The Phone extract panel is used for the generation of so called SQLite database files for the mobile phone applications. Each district will have its own Phone extract file that needs to be distributed to the mobile phones within the district. To generate a phone extract file, the operator has to select a district from the list of available districts. In case the user is having access to its own district only, the district will be automatically selected and shown on the display.

By clicking the 'create' button in panel A, a phone extract will be created. This process might take a while. As long as the hour glass (as a cursor) is shown, if you check in-background the service will download file in background while user continue working, after finish service will give the user message to download the package. The file size depends on the amount of photographs included in the extract. The file size could range into hundreds of MBs.

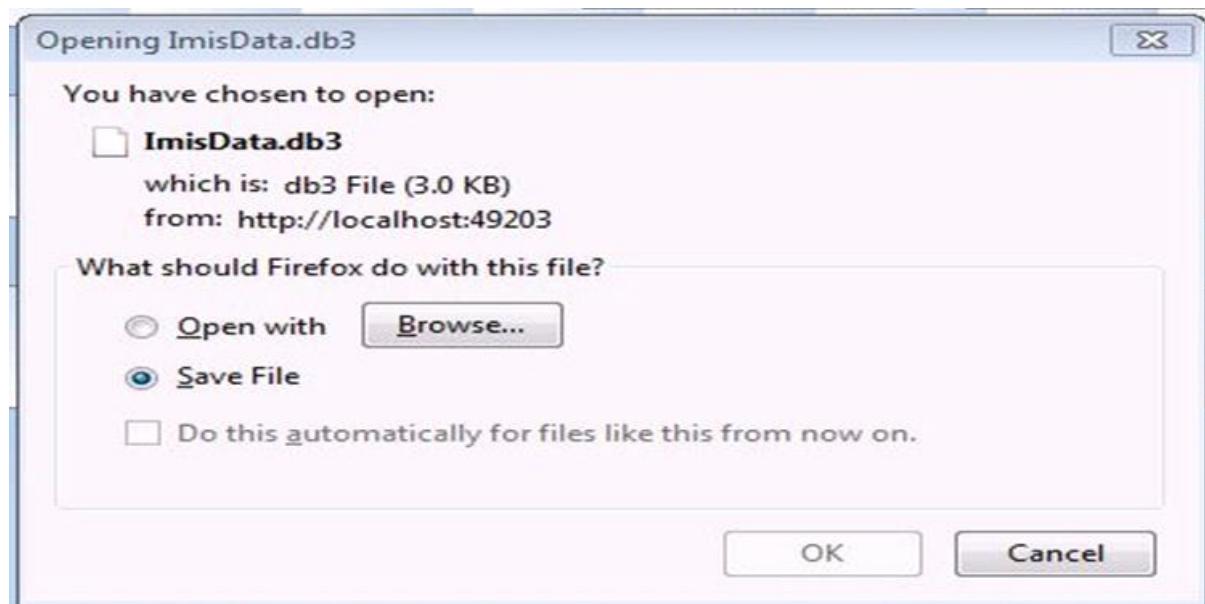
After the file has been created successfully the following message appears as shown below.



The extract will be downloaded to your local computer by clicking the ‘download’ link that will appear after the creation of the extract, as shown below.



The extract file is called ‘IMISDATA.DB3’ and needs first to be copied (downloaded) to the local machine. After clicking the Download button, the operator is able to select the destination folder (locally) for the file to download as shown below.



The extract is now ready to be transferred/copied to the mobile phones. This process is performed manually by connecting the mobile phone to the computer with the provided USB cable. The user needs to copy, manually, the file from the local machine into the ‘IMIS’ Folder on the mobile phone.

B-Offline Extract panel

The Offline extract panel is used to generate the IMIS ‘Offline’ extract files for the health facilities that run IMIS offline. When an operator belongs to one specific district, the district box is already selected with the district of the user. To create a new extract, the operator needs to click the ‘Create’ button (in panel B).

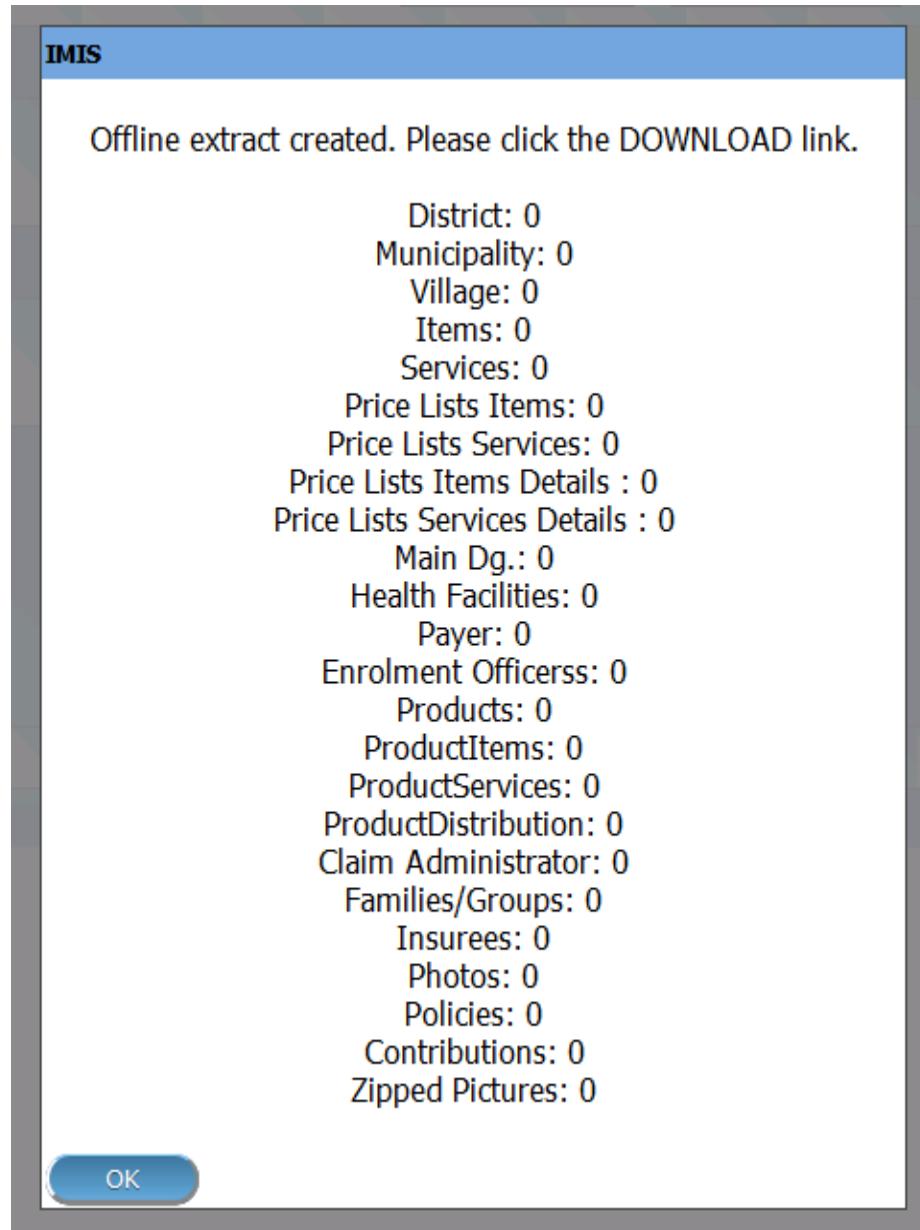
Two types of extracts could be generated:

- Differential extract
Differential extracts will only contain the differences in data compared with the previous extract. The first differential extract (sequence 000001) will contain all data as it will be the first extract. Thereafter, this type of extract, will only contain any differences after the previous extract. This will result in smaller files to sent to the Health Facilities in off-line mode. When we click the create button, the differential extract is ALWAYS generated and will be assigned the next sequence number. A separate Photo extract will be created containing only photographs linked to changes compared with the previous extract.
- Full extract ('Download F')

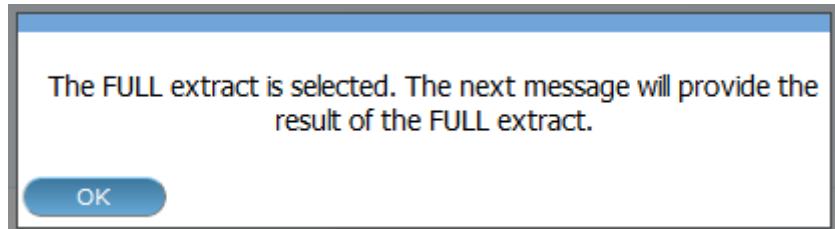
The Full extract will always contain ALL information in the database. These extracts are only generated in case the 'Full Extract' checkbox is selected as shown below.

By clicking the Create button, in case of 'Full extract' selected, two extracts will be generated, one differential extract and one FULL extract. Both extracts will have the same sequence number. This implies that FULL extracts are not always needed/generated. A separate Photo extract will be created containing ALL photographs.

After clicking the 'Create' button, the system will create the extract file and will on completion display the following message:



The message is only shown to provide some details on how much information is exported to the extract file. Depending on the 'Full extract' option, we will be redirected to the extract page and will see the newly generated extract sequence in the list OR will get a new message as shown below:



After clicking OK the statistics of the FULL extract will be shown:

IMIS

Offline extract created. Please click the DOWNLOAD link.

District: 3
Municipality: 4
Village: 8
Items: 2
Services: 2
Price Lists Items: 2
Price Lists Services: 2
Price Lists Items Details : 4
Price Lists Services Details : 4
Main Dg.: 3
Health Facilities: 2
Payer: 0
Enrolment Officerss: 2
Products: 6
ProductItems: 4
ProductServices: 5
ProductDistribution: 153
Claim Administrator: 1
Families/Groups: 2
Insurees: 4
Photos: 4
Policies: 10
Contributions: 8
Zipped Pictures: 0

OK

We are now ready to download the extract to our computer.

The combo box next to the district selector contains information on all generated extracts with the sequence number and date. (E.g. Sequence 000007 – Date 06-09-2012). If the extract selector does not show any entries (blank) it means that no previous extracts were created. At least one FULL extract needs to be generated. This is needed to initialise a new offline IMIS installation.

To download the actual extracts, the operator needs to select the desired extract sequence from the list of available extracts.

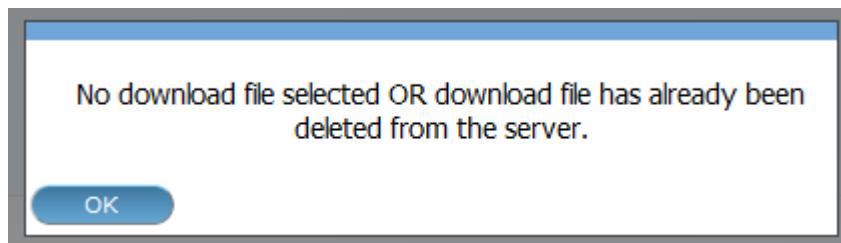
Four different types of extracts could be downloaded by clicking one of the following buttons:

- 'Download D' (Differential extract)
 - Will download the selected differential extract with the following filename
Filename: OE_D_<DistrictID>_<Sequence>.RAR (e.g. *OE_D_1_8.RAR*)
- 'Download F' (Full extract)
 - Will download the latest FULL extract with the following filename
Filename: OE_F_<DistrictID>_<Sequence>.RAR (e.g. *OE_F_1_8.RAR*)
- 'Download Photos D' (Differential Photo extract)
 - Will download the selected differential photo extract with filename:
Filename: OE_D_<DistrictID>_<Sequence>.RAR (e.g. *OE_D_1_8_Photos.RAR*)
- 'Download Photos F' (Full Photo extract)
 - Will download the latest FULL photo extract with the following filename
Filename: OE_D_<DistrictID>_<Sequence>.RAR (e.g. *OE_F_1_8_Photos.RAR*)

After clicking the desired extract download button, the file download dialog box appears to select the destination folder for the extract file as shown below:

The screenshot shows the 'Create Offline Extract' interface. At the top, there are dropdown menus for 'Region' and 'District', and several checkboxes: 'With Insuree', 'In background', 'Full extract', and a 'Create' button. Below these are two sections: 'Upload Claims XML' and 'Upload Enrolments', each with a 'Choose File' button and an 'Upload' button. The 'Choose File' buttons have the message 'No file chosen' displayed next to them.

In case the extract file is not available (anymore) on the server, the following dialog box might appear:



The reason for this box to appear could be that the file to be downloaded has been removed from the server or that you have attempted the download a 'FULL' extract but no 'FULL' extract was generated (only the differential extracts exist). It is also possible that you have attempted to download a Photograph extract but no photos were added since the last extract.

C- Import Extract panel

This panel will be disabled in the IMIS Online mode. (Only available for IMIS Offline)

D- Import Photos panel

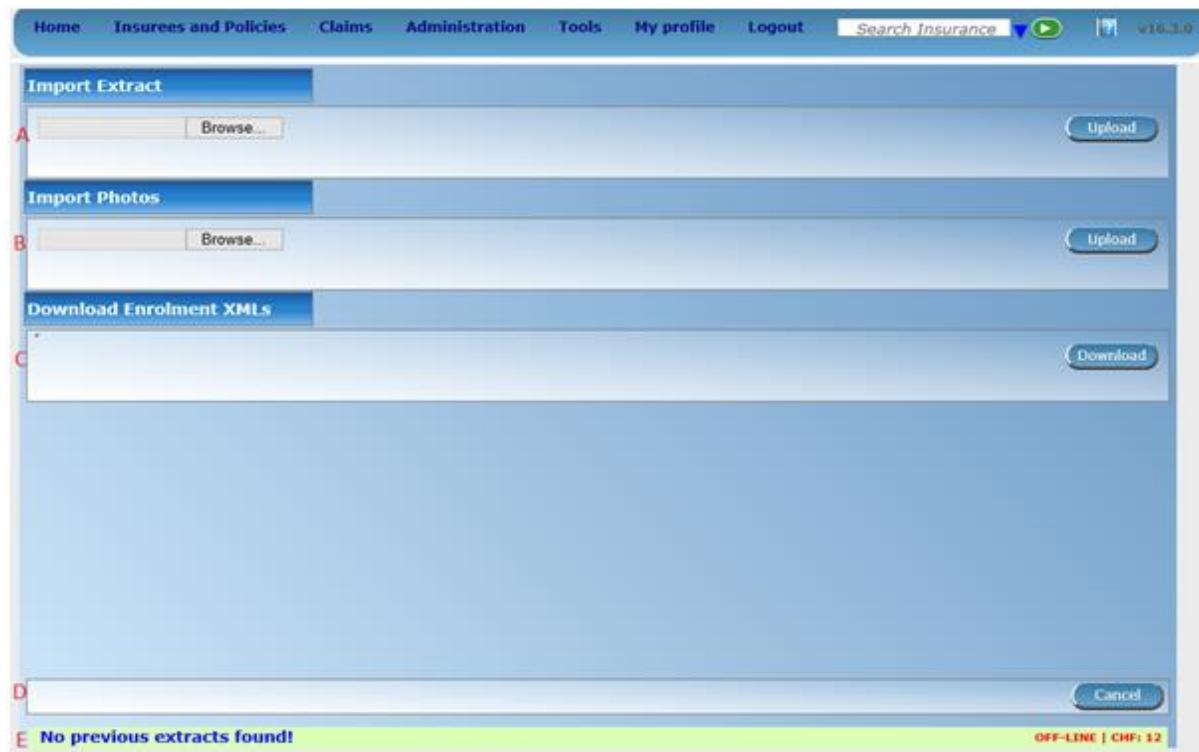
This panel will be disabled in the IMIS Online mode. (Only available for IMIS Offline)

E- Button panel

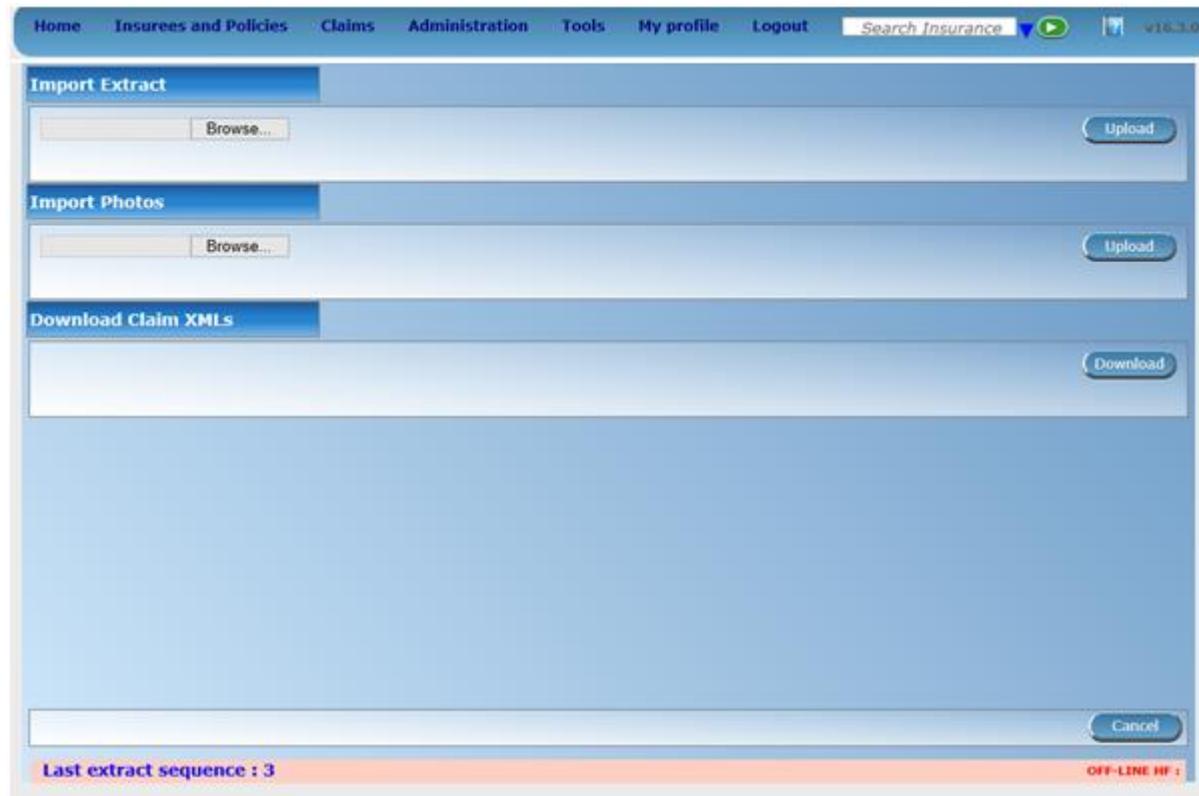
The 'Cancel' button brings the operator back to the main page of IMIS.

F- Information panel

The information panel is used to display messages back to the user. Messages will occur once an action has completed or if there was an error at any time during the process of these actions.

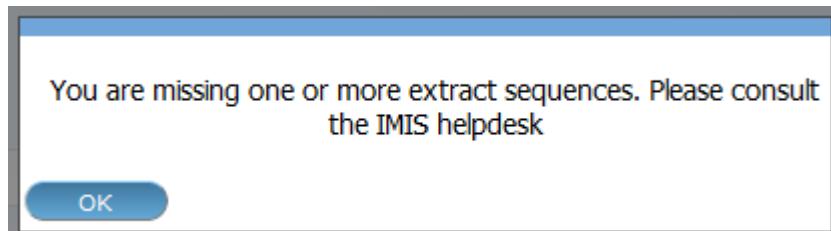
IMIS Extracts (OFFLINE MODE) HEALTH FACILITY

On clicking the Choose file button, the file selector dialog appears as shown below:

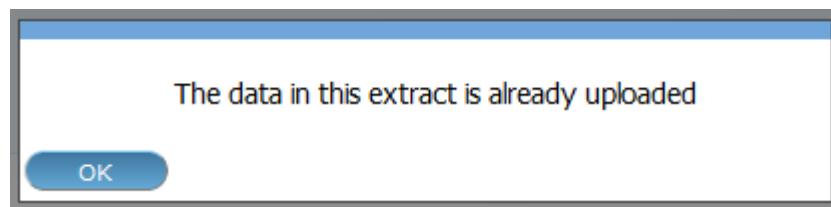


With the import/upload of an extract it is important to understand that each extract has its sequence number. This sequence number is found in the filename of the extract. We would in case of differential imports/uploads have to follow the sequence. In the example screen above, it shows in the status bar, that the last import was number 6. Therefore we should select in this case the differential extract number 7 as highlighted in the file selection dialog.

Alternatively the operator could select any 'FULL' extract with a sequence number higher than 6. In case a wrong extract is selected, warning messages will appear as shown below:



OR



In case you are missing extract sequences, additional extracts are needed to be uploaded before the extract selected. The extract selected, in this case, does not directly follow the last sequence as indicated in the status bar of the screen. The additional extracts are to be provided by IMIS scheme administrator. In case the extract file selected is valid, the system will import the data. New data will be added and existing data might be modified. After a successful import of an extract (Differential and FULL), a form is displayed with the statistics of the import as shown below:



The above statistics are provided to give some quick overview of how many records were inserted or updated during the import process. In case we would for example update the phone number of an enrolment officer, it would result in one update and one insert as we always keep historical records. The photos inserts and updates are related to information on the photos, but are not the actual photographs. The actual Photographs (*.jpg) are uploaded separately.

B-Import Photos

The import of photos is optional and will have no further checking on sequence numbers. Scheme Administrator should provide (if available) with each extract the photo extract as well.

E.g. (for Differential extract)

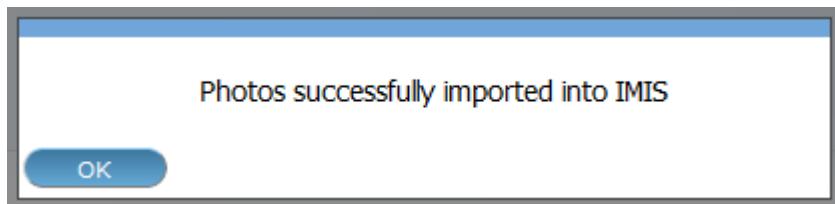
OE_D_1.RAR
 OE_D_1_Photos.RAR

OR (for FULL extract)

- OE_F_1_8.RAR
- OE_F_1_8_Photos.RAR

The photo extract will contain all photographs associated with the actual extract in a zipped format. The Upload procedure will simply unzip the extract and copy the image files to the photo folder of IMIS.

After successful upload of the photographs the following message appears:



D- Button panel

The 'Cancel' button brings the operator back to the main page of IMIS.

E- Information panel

The information panel is used to display messages back to the user. Messages will occur once an action has completed or if there was an error at any time during the process of these actions. If the user opens the IMIS extracts page (in Offline mode only), the status bar will show the last sequence number uploaded.

IMIS Extracts (OFFLINE MODE)

- **Import Extract**

Used to upload extract obtained from online IMIS, refer steps as mentioned on import extract in offline health facility.

- **Import Photos**

Used to upload photos obtained from online IMIS, refer steps as mentioned on import photos in offline health facility

- **Download Enrolment XMLs**

Used to download families, insurees, policies and Contributions created in the offline IMIS HF prior to be sent to online IMIS.

5.7.25. Reports.aspx

Reports

Month Quarter Year Region District Product

--Month-- --Period-- 2018 --Select Region-- --Select Product--

Primary Operational Indicators-polices
Primary Operational Indicators-claims
Derived Operational Indicators
Contribution Collection
Product Sales
Contribution Distribution
User Activity Report
Enrolment Performance Indicators
Status of Registers
Insurees without Photos
Payment Category Overview
Matching Funds
Claim Overview
Percentage of Referrals
Families and Insurees Overview

Preview Cancel

Interface object name:	<i>Reports.aspx</i>
Menu path:	<i>Reports (under Tools menu)</i>
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	(creation of all reports)
Class Diagram	 <pre> classDiagram class ReportsBI { <<Methods>> checkRoles() GetActions() GetAllProducts() getCatchmentArea() GetClaimOverview() GetClaimStatus() GetDerivedIndicators() GetDistricts() GetEnroledFamilies() GetEntities() GetHFCodes() GetInsureesWithoutPhotos() GetMatchingFunds() GetMonths() GetOfficers() GetPayers() GetPaymentCategoryOverview() GetPendingInsurees() GetPercentageReferral() GetPeriodNo() GetPolicySold() GetPolicyStatus() GetPremiumCollection() GetPremiumDistribution() GetPreviousMatchingFundsReportDates() GetPrimaryIndicators1() GetPrimaryIndicators2() GetProductCapitationDetails() GetProductName_Account() GetProducts() GetProductsStict() GetRegions() getRejectedPhoto() GetRenewals() GetReportTypes() GetSorting() GetStatusofRegisters() GetTypeOfPayment() GetUserActivityData() GetUsers() GetVillages() GetWards() GetYears() } </pre>

Brief description:

This interface opens up when the user has clicked the 'Reports' option in the reports menu under the Tools top menu. This function is used to generate statistical reports from IMIS. The report selector has several filters as shown in the screenshot. By selecting any of the report, and clicking the preview button, the desired report will be displayed applying the desired filter setting.

The report will be generated using SQL Server reporting services (SSRS). The report will be exportable to MS Excel and PDF formats.

5.7.26. Utilities.aspx

The screenshot shows the 'Utilities.aspx' page of the openIMIS WEB APPLICATION. At the top, there is a navigation bar with links for Home, Insurees and Policies, Claims, Administration, Tools, My profile, and Logout. To the right of these links are search fields for 'Search Insurance' and 'v16.3.0'. Below the navigation bar, the page title 'Backend Version: 1.2' is displayed. The main content area is divided into three sections: 'BACKUP', 'RESTORE', and 'EXECUTE SCRIPT'. The 'BACKUP' section contains a text input field with the path 'C:\AutoIMIS-DB-Backups\' and a checkbox labeled 'Save path', followed by a 'BACKUP' button. The 'RESTORE' section contains a text input field and a 'RESTORE' button. The 'EXECUTE SCRIPT' section contains a 'Browse...' button and an 'EXECUTE' button. At the bottom of the page, there is a 'Cancel' button.

Interface object name:	Utilities.aspx
Menu path:	Utilities under Tools menu
Hyperlinks/Redirections:	
Action Button:	
Use case Reference	N/A
Class Diagram	None

Brief description:

The interface is only available for the Offline IMIS installation in health facilities.

This interface is to be used for the following purposes:

- Creation of database backup
- Restoring database
- Executing a database script

By clicking the 'backup' button, the full database backup will be created in the backup folder selected. Optionally one could change the backup folder name and save it as a default.

The restore feature will allow a full database to be restored using a previously created SQL Server database backup file. The operator will have to fully type the path and filename of the backup file. On clicking the restore button, the database will be overwritten with the contents of the database in the backup file. Obviously this action should be used with caution and will only be available for the IMIS HF Administrator (Role 512).

The script execution is only needed in case IMIS will have some structural changes on database level or some specific data manipulation is required. The operator should select the file provided and click the 'Execute' button. The system will now automatically apply the database changes. The files to run will be provided by Exact Software, the developers.

5.7.27. IMIS.MASTER (used for Menu and Quick Inquiry)

The screenshot shows a web-based application interface for managing patient data and deductions. At the top, there is a small thumbnail image of a person. Below it, the patient's details are listed:

Last Name	Soni
Other Names	Hiren
Birth Date	01-27-83
Gender	M

Below these details is a red navigation bar with numbers 1 through 8, followed by a red 'X' icon. Underneath the navigation bar is a table showing hospital and non-hospital deduction information:

CODE	NAME	EXPIRE DATE	STATUS	HOSPITAL DEDUCTION	NON-HOSPITAL DEDUCTION	HOSPITAL CEILING	NON-HOSPITAL CEILING
NOCYCLE	No Cycle	09/05/2017	A	1000.00	1000.00	20000.00	20000.00

At the bottom of the interface is another table with various deduction amounts:

ADM LEFT	VIS LEFT	CONS LEFT	SURG LEFT	DEL LEFT	ANT LEFT	CONS AMT/CLAIM	SURG AMT/CLAIM	DEL AMT/CLAIM	HOSP AMT/CLAIM	ANT AMT/CLAIM
N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

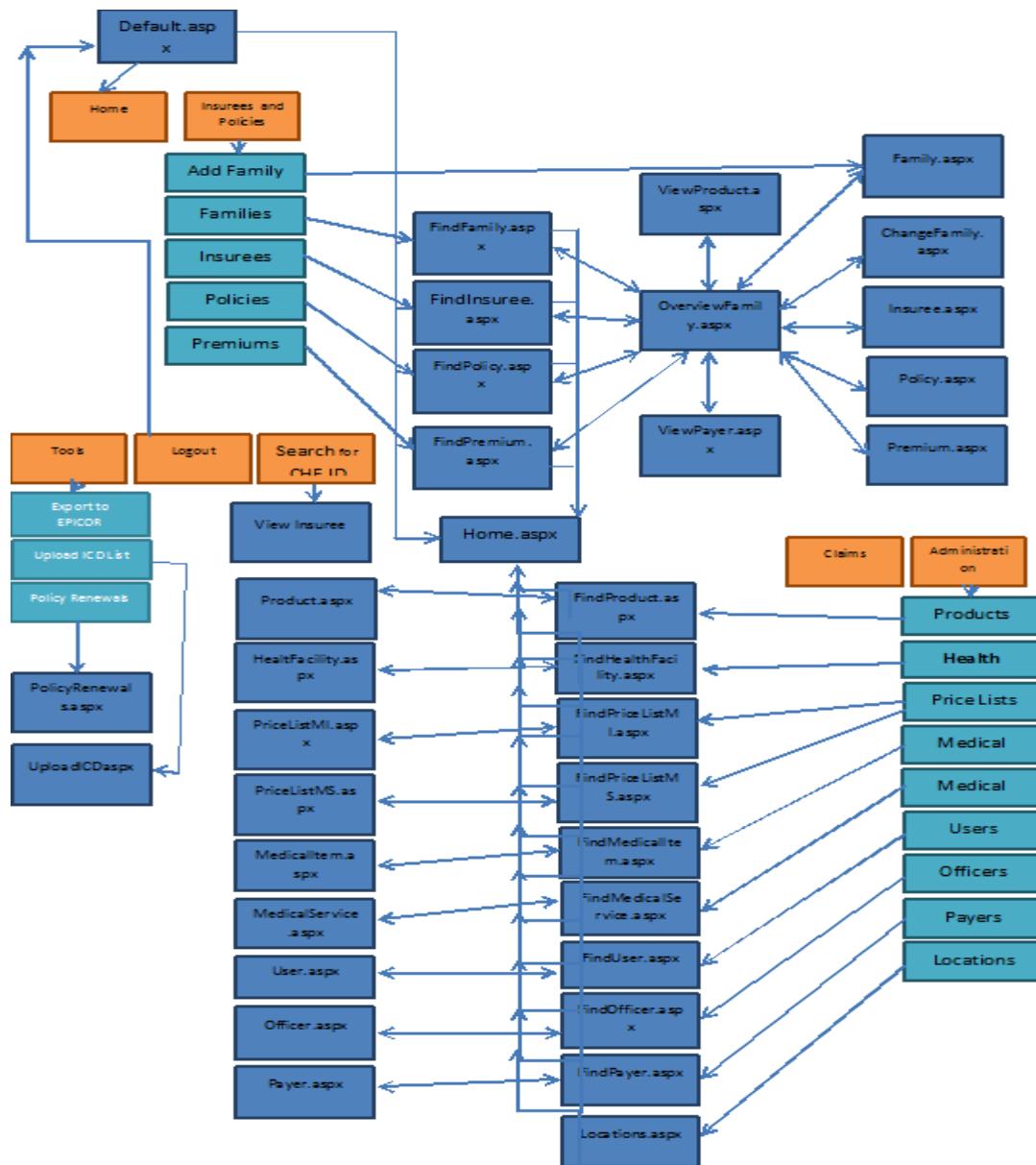
Although this object is not an aspx page as such it is included in this chapter as it is part of the presentation Tier and is included in the use case 5.2.20 scenario in the specification for phase 3.

The above window (policy overview insure) is activated via the Quick Insuree search facility in the top menu bar as shown below:



This page is now developed for the On-line and off - line system. A similar page is suggested to be developed for the quick enquiry screen for mobile phones. The designs for the mobile phone are currently under review by experts from the Swiss tropical health institute.

5.8. Graphical page routing overview



6. Database Design

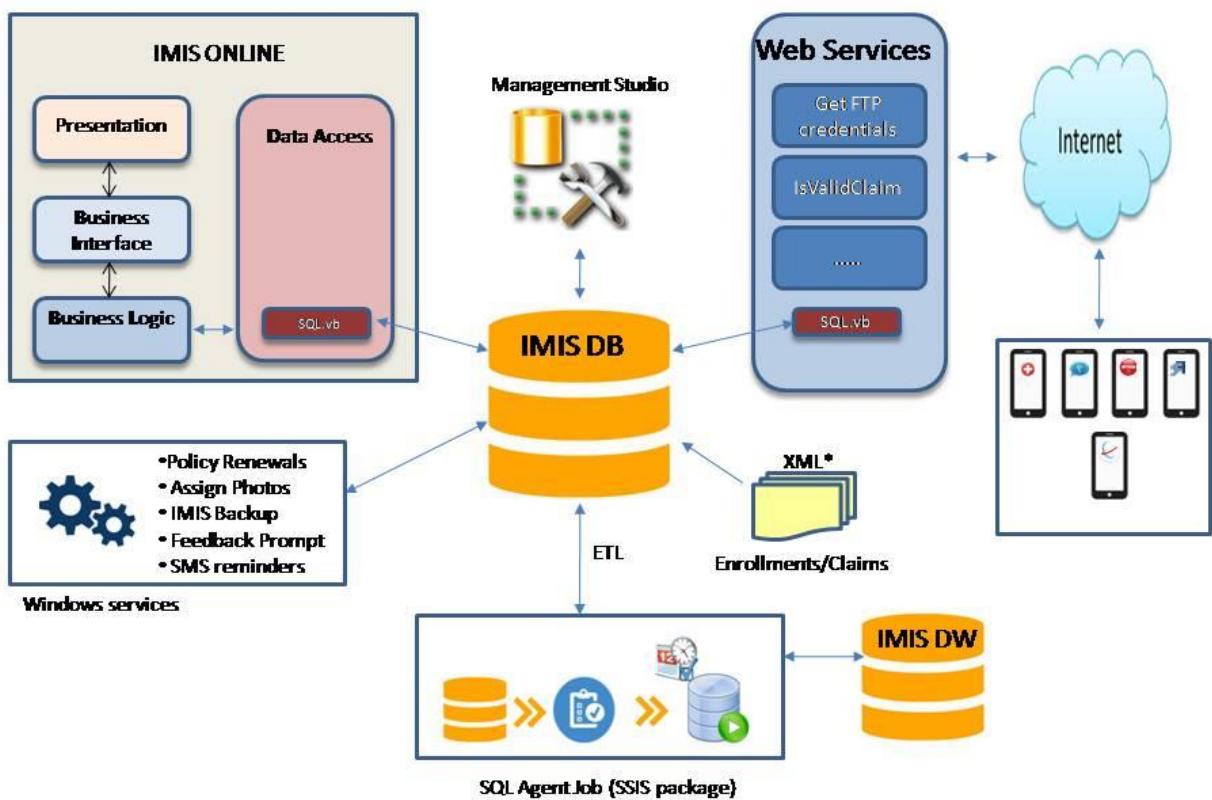
This chapter will cover the database design for the Online IMIS and the Off-line clients that have no connectivity. This chapter will provide illustration of how IMIS Database is interconnected with various components. This chapter will also provide graphical relationship overview of IMIS Database.

Lastly, this chapter, will provide the design of the database 'IMIS'. This database we will find in the on-line and off-line scenario. The off-line client will have an extra database for the Claim management objects in order to separate the 'master data' that originates from the Central database from the Claim management objects belonging to the offline facility.

6.1. Database Interconnectivity

IMIS Database interconnected with IMIS Online, Web Service, Windows Service as well as SQL Server Management Studio,

The graphic below illustrates the IMIS database inter connectivity



IMIS database is connected with IMIS web application. It also connected with Windows services for assigning photo to relevant insurees, policy renewal, IMIS Backup, feedback prompt as well we SMS reminder. Via Web services functions, IMIS database exchanges data from online to offline web application as well as exchanging data with Phone application,

6.2. Tables with main properties

Each record in each table in the database has an internal ID number. This number is automatically generated via the identity seed configuration for these ID fields. The ID fields are also used to enforce the integrity constraints between the tables. The database diagrams discussed later in this chapter will provide an overview on the relationships using these ID fields in the various tables.

Each table in the database has an additional 4 fields:

- ValidityFrom
- ValidityTo
- LegacyID
- AuditUserID

These 4 fields are used for data auditing and data archiving. Any record created in IMIS will have by default today's timestamp in the ValidityFrom field. This is automatically enforced by the default constraint 'GETDATE()' in the ValidityFrom field.

In case any record is about to be changed by the application, e.g. a price of a service, the following actions will occur:

1. Create a new record containing the original data of the record (before the change) with in the ValidityTo field the value of today's timestamp and in the LegacyID field the ID value of the original record.
2. Update the current record with the changed information.

The legacyID will be used to link all 'versions' of a record together with a ValidityFrom and ValidityTo date. The original ID of the record will always remain the unique key of the record and is used to relate to other objects in the database. The ValidityTo field of the original record will only be updated in the original record in case of delete actions.

The AuditUserID will always contain the UserID of the operator that inserted the record or '0' if it was updated by a service.

Certain tables (claim management tables) will have besides the normal audit fields as well similar audit fields for a 'reviewer':

- ValidityFromReview
- ValidityToReview
- AuditUserIDReview

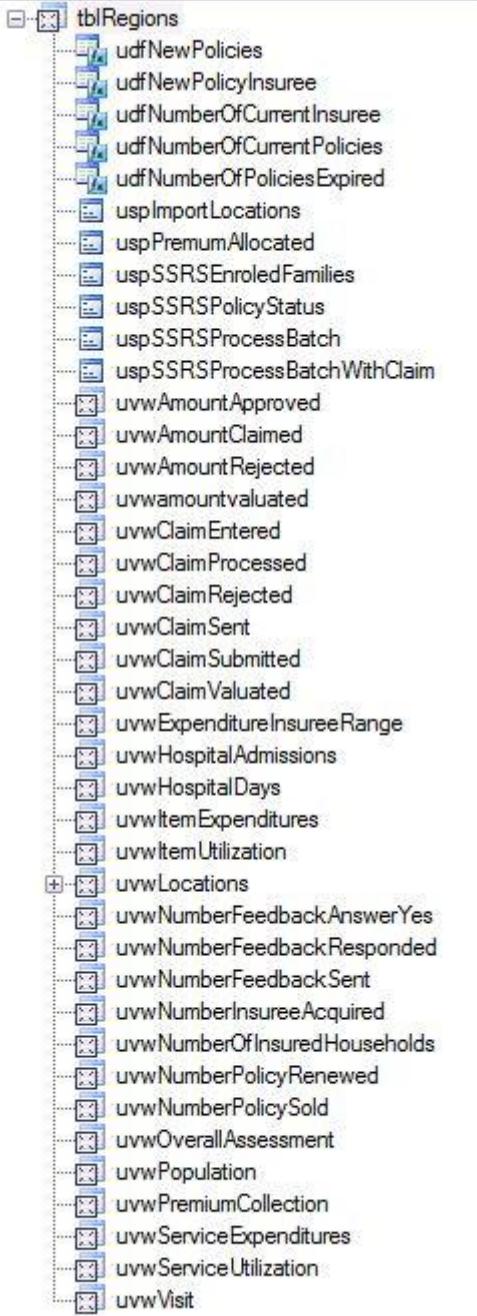
These fields will get values at the moment the reviewer is adding information for the first time. This way we can prevent that always we will have multiple records while only information is added by the reviewer and original data is actually not changed. At the moment a reviewer is changing information (the audit information for the reviewer has already been registered in the record), like in the normal archiving scenario, an additional record will be created to register the change of data and the legacyID will be used for keeping the versioning of records together.

Hereunder, we will further cover all tables and views individually.

6.1.1. tblRegion (represented in the database as view)

Column	Alias	Table	Outp...
LocationId	RegionId	tblLocations	<input checked="" type="checkbox"/>
LocationCode	Region...	tblLocations	<input checked="" type="checkbox"/>
LocationName	Region...	tblLocations	<input checked="" type="checkbox"/>
ValidityFrom		tblLocations	<input checked="" type="checkbox"/>
ValidityTo		tblLocations	<input checked="" type="checkbox"/>
LegacyId		tblLocations	<input checked="" type="checkbox"/>
AuditUserId		tblLocations	<input checked="" type="checkbox"/>
RowId		tblLocations	<input checked="" type="checkbox"/>
LocationType		tblLocations	<input type="checkbox"/>
			<input checked="" type="checkbox"/>

Objects that depend on tblRegion:	TblRegion depends on:
-----------------------------------	-----------------------

 <pre> tblRegions +-- udfNewPolicies +-- udfNewPolicyInsuree +-- udfNumberOfCurrentInsuree +-- udfNumberOfCurrentPolicies +-- udfNumberOfPoliciesExpired +-- uspImportLocations +-- uspPremiumAllocated +-- uspSSRSEnroledFamilies +-- uspSSRSPolicyStatus +-- uspSSRSProcessBatch +-- uspSSRSProcessBatchWithClaim +-- uvwAmountApproved +-- uvwAmountClaimed +-- uvwAmountRejected +-- uvwamountvaluated +-- uvwClaimEntered +-- uvwClaimProcessed +-- uvwClaimRejected +-- uvwClaimSent +-- uvwClaimSubmitted +-- uvwClaimValuated +-- uvwExpenditureInsureeRange +-- uvwHospitalAdmissions +-- uvwHospitalDays +-- uvwItemExpenditures +-- uvwItemUtilization +-- uvwLocations +-- uvwNumberFeedbackAnswerYes +-- uvwNumberFeedbackResponded +-- uvwNumberFeedbackSent +-- uvwNumberInsureeAcquired +-- uvwNumberOfInsuredHouseholds +-- uvwNumberPolicyRenewed +-- uvwNumberPolicySold +-- uvwOverallAssessment +-- uvwPopulation +-- uvwPremiumCollection +-- uvwServiceExpenditures +-- uvwServiceUtilization +-- uvwVisit </pre>	None
Reference to Data Model Entity: Region	
<p>Notes: This view will be initially uploaded from external files. This table will be initially uploaded from external files provided by Administrator</p>	

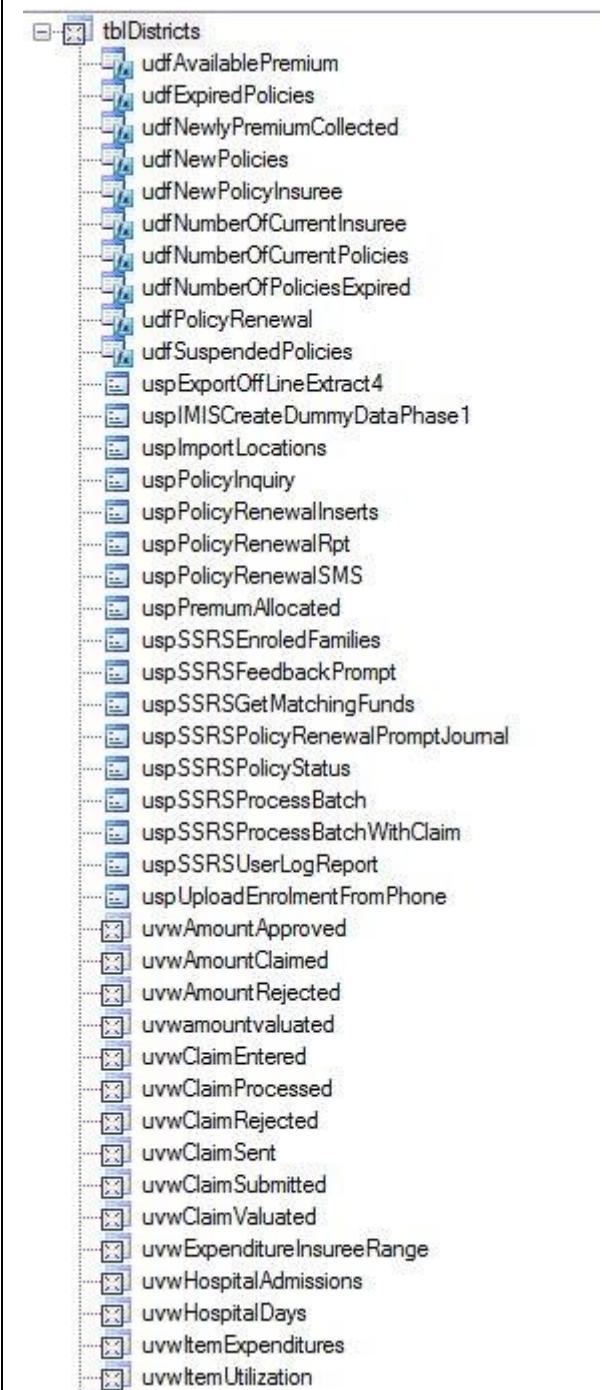
6.1.2.tblDistricts (represented in the database as view)

TblDistricts will contain all districts to be used and referenced in IMIS.

This table will initially be populated from an external file to avoid excessive data entry. Thereafter this table will be maintained via the user interfaces.

Column	Alias	Table	O...
LocationId	DistrictId	tblLocations	<input checked="" type="checkbox"/>
LocationCode	DistrictCode	tblLocations	<input checked="" type="checkbox"/>
LocationName	DistrictName	tblLocations	<input checked="" type="checkbox"/>
ParentLocatio...	Region	tblLocations	<input checked="" type="checkbox"/>
ValidityFrom		tblLocations	<input checked="" type="checkbox"/>
ValidityTo		tblLocations	<input checked="" type="checkbox"/>
LegacyId		tblLocations	<input checked="" type="checkbox"/>
AuditUserId		tblLocations	<input checked="" type="checkbox"/>
RowId		tblLocations	<input checked="" type="checkbox"/>
LocationType		tblLocations	<input type="checkbox"/>
			<input checked="" type="checkbox"/>

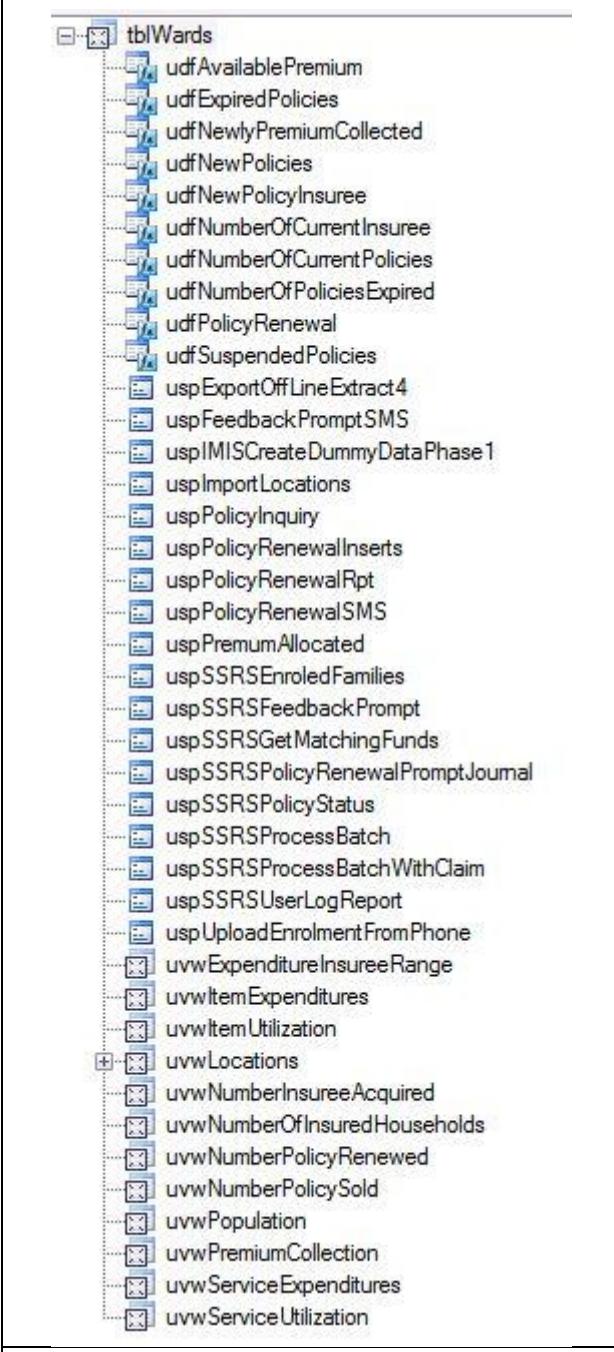
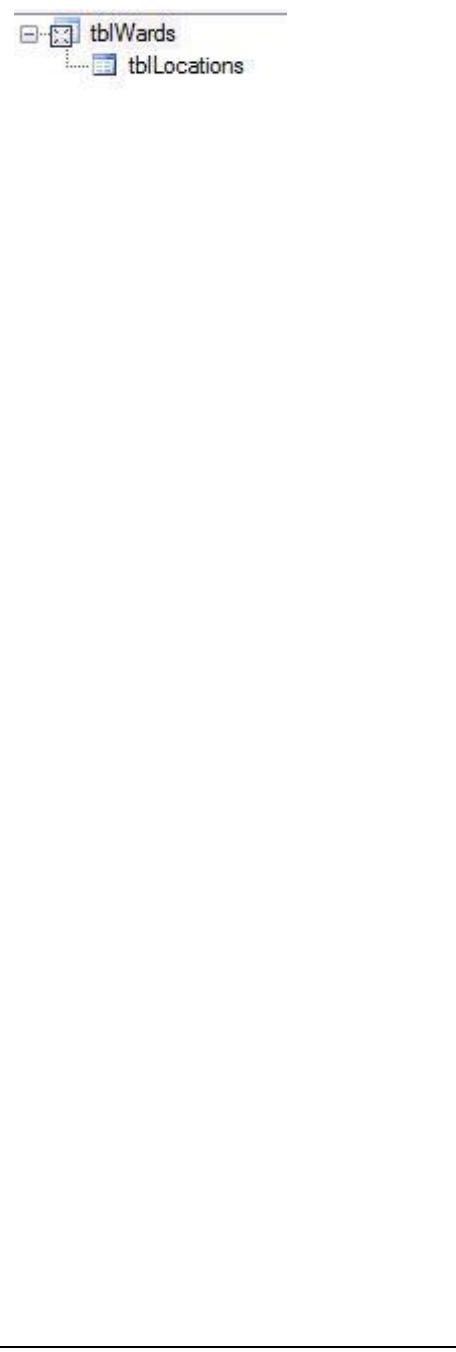
Objects that depend on tblDistricts:	TblDistricts depends on:
--------------------------------------	--------------------------

 <pre> tblDistricts +-----+ udfAvailablePremium udfExpiredPolicies udfNewlyPremiumCollected udfNewPolicies udfNewPolicyInsuree udfNumberOfCurrentInsuree udfNumberOfCurrentPolicies udfNumberOfPoliciesExpired udfPolicyRenewal udfSuspendedPolicies uspExportOffLineExtract4 uspIMISCreateDummyDataPhase1 uspImportLocations uspPolicyInquiry uspPolicyRenewalInserts uspPolicyRenewalRpt uspPolicyRenewalSMS uspPremiumAllocated uspSSRSEnroledFamilies uspSSRSFeedbackPrompt uspSSRSGetMatchingFunds uspSSRSPolicyRenewalPromptJournal uspSSRSPolicyStatus uspSSRSProcessBatch uspSSRSProcessBatchWithClaim uspSSRSUserLogReport uspUploadEnrolmentFromPhone uvwAmountApproved uvwAmountClaimed uvwAmountRejected uvwamountvaluated uvwClaimEntered uvwClaimProcessed uvwClaimRejected uvwClaimSent uvwClaimSubmitted uvwClaimValuated uvwExpenditureInsureeRange uvwHospitalAdmissions uvwHospitalDays uvwItemExpenditures uvwItemUtilization </pre>	 <pre> tblDistricts +-----+ tblLocations </pre>
Reference to Data Model Entity: District	
Notes: This view will be initially uploaded from external files. This table will be initially uploaded from external files provided by Administrator	

6.1.3.TblWard (Municipality): (represented in the database as view)

TblWard will contain all Municipality that fall under the villages. This table will initially be populated from an external file to avoid excessive data entry. Thereafter this table will be maintained via the user interfaces.

Column	Alias	Table	Output
LocationId	WardId	tblLocations	<input checked="" type="checkbox"/>
ParentLocationId	DistrictId	tblLocations	<input checked="" type="checkbox"/>
LocationCode	WardCode	tblLocations	<input checked="" type="checkbox"/>
LocationName	WardName	tblLocations	<input checked="" type="checkbox"/>
ValidityFrom		tblLocations	<input checked="" type="checkbox"/>
ValidityTo		tblLocations	<input checked="" type="checkbox"/>
LegacyId		tblLocations	<input checked="" type="checkbox"/>
AuditUserId		tblLocations	<input checked="" type="checkbox"/>
RowId		tblLocations	<input checked="" type="checkbox"/>
LocationType		tblLocations	<input type="checkbox"/>
			<input checked="" type="checkbox"/>
Objects that depend on TblWard :		TblWard depends on:	

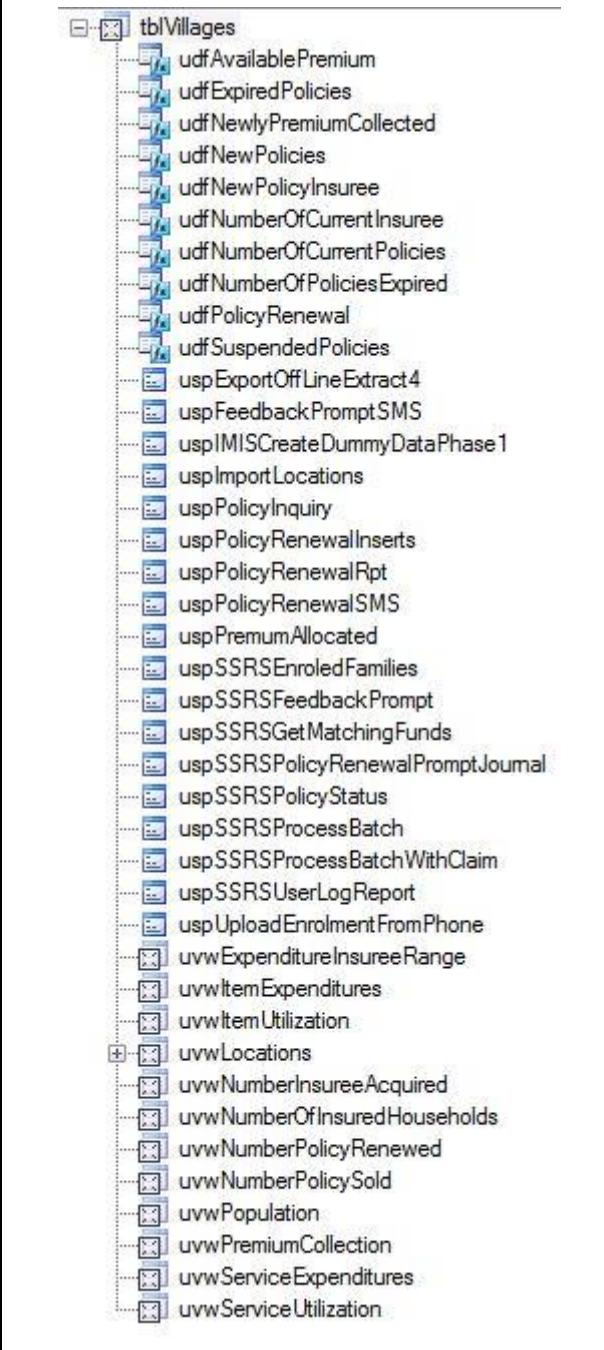
	
<p>Reference to Data Model Code List: Districts-Villages-Municipality</p>	
<p>Notes: This view will be initially uploaded from external files provided by Administrator</p>	

6.1.4. tblVillages (represented in the database as view)

TblVillages will contain all villages that fall under the districts. This table will initially be populated from an external file to avoid excessive data entry. Thereafter this table will be maintained via the user interfaces.

Column	Alias	Table	Outp...
LocationId	VillageId	tblLocations	<input checked="" type="checkbox"/>
ParentLocationId	WardId	tblLocations	<input checked="" type="checkbox"/>
LocationCode	VillageCode	tblLocations	<input checked="" type="checkbox"/>
LocationName	VillageName	tblLocations	<input checked="" type="checkbox"/>
MalePopulation		tblLocations	<input checked="" type="checkbox"/>
FemalePopulation		tblLocations	<input checked="" type="checkbox"/>
OtherPopulation		tblLocations	<input checked="" type="checkbox"/>
Families		tblLocations	<input checked="" type="checkbox"/>
ValidityFrom		tblLocations	<input checked="" type="checkbox"/>
ValidityTo		tblLocations	<input checked="" type="checkbox"/>
LegacyId		tblLocations	<input checked="" type="checkbox"/>
AuditUserId		tblLocations	<input checked="" type="checkbox"/>
RowId		tblLocations	<input checked="" type="checkbox"/>
LocationType		tblLocations	<input type="checkbox"/>
			<input type="checkbox"/>

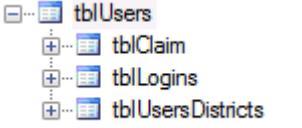
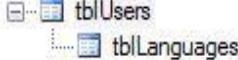
Objects that depend on tblVillages:	TblVillages depends on:
-------------------------------------	-------------------------

 <pre> tblVillages +-- udfAvailablePremium +-- udfExpiredPolicies +-- udfNewlyPremiumCollected +-- udfNewPolicies +-- udfNewPolicyInsuree +-- udfNumberOfCurrentInsuree +-- udfNumberOfCurrentPolicies +-- udfNumberOfPoliciesExpired +-- udfPolicyRenewal +-- udfSuspendedPolicies +-- uspExportOffLineExtract4 +-- uspFeedbackPromptSMS +-- uspIMISCreateDummyDataPhase1 +-- uspImportLocations +-- uspPolicyInquiry +-- uspPolicyRenewalInserts +-- uspPolicyRenewalRpt +-- uspPolicyRenewalSMS +-- uspPremiumAllocated +-- uspSSRSEnroledFamilies +-- uspSSRSFeedbackPrompt +-- uspSSRSGetMatchingFunds +-- uspSSRS Policy Renewal Prompt Journal +-- uspSSRS Policy Status +-- uspSSRSProcessBatch +-- uspSSRSProcessBatchWithClaim +-- uspSSRSUserLogReport +-- uspUploadEnrolmentFromPhone +-- uvwExpenditureInsureeRange +-- uvwItemExpenditures +-- uvwItemUtilization +-- uvwLocations +-- uvwNumberInsureeAcquired +-- uvwNumberOfInsuredHouseholds +-- uvwNumberPolicyRenewed +-- uvwNumberPolicySold +-- uvwPopulation +-- uvwPremiumCollection +-- uvwServiceExpenditures +-- uvwServiceUtilization </pre>	 <pre> tblVillages +-- tblLocations </pre>
Reference to Data Model Code List: Districts-Villages-Municipality	
Notes: This view will be initially uploaded from external files provided by Administrator	

6.1.5.TblUsers

TblUser will contain all IMIS users with credentials and security role.
The security field RoleID has the following

Column Name	Data Type	Allow Nulls
UserID	int	<input type="checkbox"/>
LanguageID	nvarchar(2)	<input type="checkbox"/>
LastName	nvarchar(100)	<input type="checkbox"/>
OtherNames	nvarchar(100)	<input type="checkbox"/>
Phone	nvarchar(50)	<input checked="" type="checkbox"/>
LoginName	nvarchar(25)	<input type="checkbox"/>
RoleID	int	<input type="checkbox"/>
HFID	int	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
password	varbinary(256)	<input checked="" type="checkbox"/>
DummyPwd	nvarchar(25)	<input checked="" type="checkbox"/>
EmailId	nvarchar(200)	<input checked="" type="checkbox"/>

Objects that depend on tblUsers:	TblUsers depends on:
	

Reference to Data Model Entity: User
<p>Notes: RoleID will contain a bitmask for each of the User Roles:</p> <ul style="list-style-type: none"> • 1 = Enrolment Assistant • 2 = Manager • 4 = Accountant • 8 = Clerk • 16 = Medical Officer • 32 = Scheme Administrator • 64 = IMIS Scheme Administrator • 128 = Receptionist • 256 = Claim Administrator • 512 = Claim Contributor • 524288= HF Administrator • 1048576= Scheme Offline Administrator <p>E.g. 48 means Medical Officer + Scheme Administrator</p>

6.1.6. **tblUsersDistricts**

TblUsersDistricts will contain all districts that a user could select/use in IMIS. Within these districts the security role of the user is defined at user level (tblUser).

Column Name	Data Type	Allow Nulls
UserDistrictID	int	<input type="checkbox"/>
UserID	int	<input type="checkbox"/>
LocationId	int	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblUsersDistricts:	tblUsersDistricts depends on:
<ul style="list-style-type: none"> -> tblUsersDistricts <ul style="list-style-type: none"> -> uspCleanTables -> uspIMISCreateDummyDataPhase1 -> uspSSRSSStatusRegister -> uspSSRSUserLogReport 	<ul style="list-style-type: none"> -> tblUsersDistricts <ul style="list-style-type: none"> -> tblLocations -> tblUsers

Reference to Data Model Entity: **N/A**

Notes: All mappings of users towards Districts is hosted in this table to keep a uniform approach.

6.1.7. tblOfficer

tblOfficer contains all enrolment officers.

 OfficerID	int	<input type="checkbox"/>
Code	nvarchar(8)	<input type="checkbox"/>
LastName	nvarchar(100)	<input type="checkbox"/>
OtherNames	nvarchar(100)	<input type="checkbox"/>
DOB	date	<input checked="" type="checkbox"/>
Phone	nvarchar(50)	<input checked="" type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
OfficerIDSubst	int	<input checked="" type="checkbox"/>
WorksTo	smalldatetime	<input checked="" type="checkbox"/>
VEOCode	nvarchar(8)	<input checked="" type="checkbox"/>
VEOLastName	nvarchar(100)	<input checked="" type="checkbox"/>
VEOOtherNames	nvarchar(100)	<input checked="" type="checkbox"/>
VEODOB	date	<input checked="" type="checkbox"/>
VEOPhone	nvarchar(25)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
EmailId	nvarchar(200)	<input checked="" type="checkbox"/>
PhoneCommunication	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>
Objects that depend on tblOfficer:	tblOfficer depends on:	



6.1.8. tblOfficerVillages

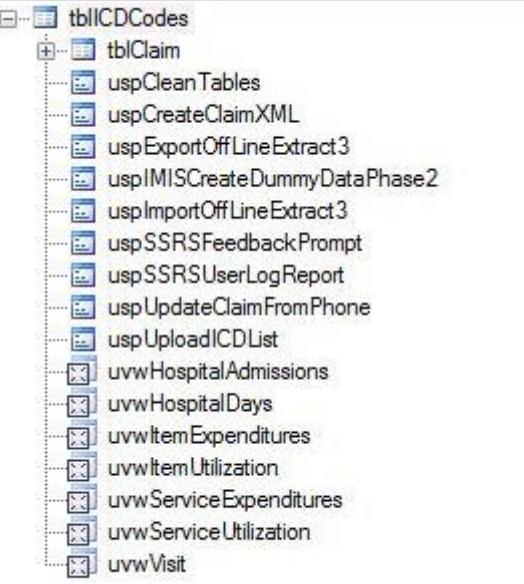
	Column Name	Data Type	Allow Nulls
PK	OfficerVillageId	int	<input type="checkbox"/>
	OfficerId	int	<input checked="" type="checkbox"/>
	LocationId	int	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input checked="" type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyId	int	<input checked="" type="checkbox"/>
	AuditUserId	int	<input checked="" type="checkbox"/>
	RowId	timestamp	<input type="checkbox"/>
			<input type="checkbox"/>

Objects that depend on tblOfficerVillages:	tblOfficerVillages depends on:
<ul style="list-style-type: none">-> tblOfficerVillages-> uspCleanTables	<ul style="list-style-type: none">-> tblOfficerVillages-> tblOfficer+> tblVillages
Reference to Data Model Entity: N/A	

6.1.9.tbICDCodes

This table will contain standard international codes for diseases. These codes are used at the entry of claims.

Column Name	Data Type	Allow Null
ICDID	int	<input type="checkbox"/>
ICDCode	nvarchar(6)	<input type="checkbox"/>
ICDName	nvarchar(255)	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>

Objects that depend on tbICDCodes:	tbICDCodes depends on:
	

Reference to Data Model Code List: Diseases
Notes: This table will be initially uploaded from external files provided by Administrator

6.1.10. tblItems

	Column Name	Data Type	Allow Nulls
	ItemID	int	<input type="checkbox"/>
	ItemCode	nvarchar(6)	<input type="checkbox"/>
	ItemName	nvarchar(100)	<input type="checkbox"/>
	ItemType	char(1)	<input type="checkbox"/>
	ItemPackage	nvarchar(255)	<input checked="" type="checkbox"/>
	ItemPrice	decimal(18, 2)	<input type="checkbox"/>
	ItemCareType	char(1)	<input type="checkbox"/>
	ItemFrequency	smallint	<input checked="" type="checkbox"/>
	ItemPatCat	tinyint	<input type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>

Objects that depend on tblItems:	tblItems depends on:
<ul style="list-style-type: none"> -> tblItems <ul style="list-style-type: none"> +--> tblClaimItems +--> tblPLItemsDetail +--> tblProductItems 	<ul style="list-style-type: none"> --> tblItems

Reference to Data Model Entity: Medical Item
<p>Notes:</p> <p>ItemType could contain the following options:</p> <ul style="list-style-type: none"> • 'D' = Drug • 'P' = Prostheses <p>ItemCareType will contain a bitmask for each of the following options:</p> <ul style="list-style-type: none"> • I = In patient • O = Out-patient • B = Both <p>ItemPatCat will contain a bitmask for each of the following options:</p> <ul style="list-style-type: none"> • 1 = Man • 2 = Woman • 4 = Adult • 8 = Child

6.1.11. tbIPLItems

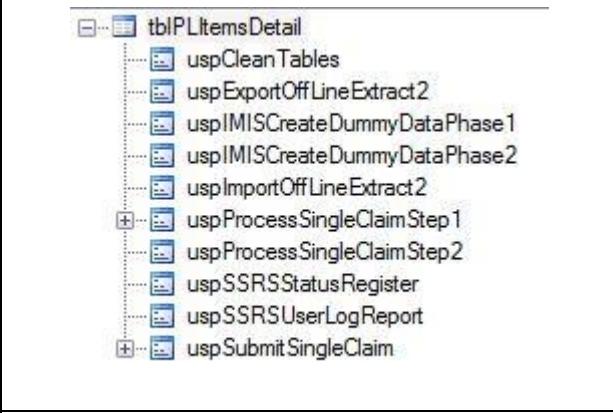
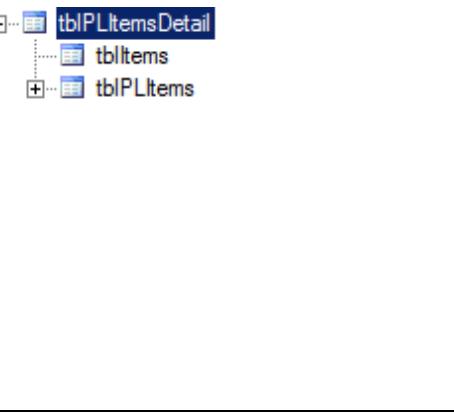
Column Name	Data Type	Allow Nulls
PLItemID	int	<input type="checkbox"/>
PLItemName	nvarchar(100)	<input type="checkbox"/>
DatePL	date	<input type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tbIPLItems:	tbIPLItems depends on:
	
Reference to Data Model Entity: Pricelist of Medical Items	
Notes: DistrictID can contain null. This means that in that case all districts could use this pricelist definition	

6.1.12. tbIPLItemsDetail

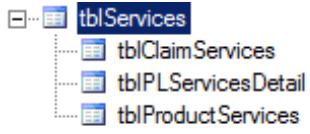
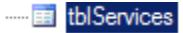
Column Name	Data Type	Allow Nulls
PLItemDetailID	int	<input type="checkbox"/>
PLItemID	int	<input type="checkbox"/>
ItemID	int	<input type="checkbox"/>
PriceOverrule	decimal(18, 2)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>

Objects that depend on tbIPLItemsDetail:	tbIPLItemsDetail depends on:
--	------------------------------

	
<p>Reference to Data Model Entity: Remark5</p>	
<p>Notes: This table maps items with pricelists (PLItemID) with the option to overrule the standard price from tblItems (ItemID)</p>	

6.1.13. tblServices

Column Name	Data Type	Allow Nulls
ServiceID	int	<input type="checkbox"/>
ServCode	nvarchar(6)	<input type="checkbox"/>
ServName	nvarchar(100)	<input type="checkbox"/>
ServType	char(1)	<input type="checkbox"/>
ServLevel	char(1)	<input type="checkbox"/>
ServPrice	decimal(18, 2)	<input type="checkbox"/>
ServCareType	char(1)	<input type="checkbox"/>
ServFrequency	smallint	<input checked="" type="checkbox"/>
ServPatCat	tinyint	<input type="checkbox"/>
ValidityFrom	datetime	<input checked="" type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input checked="" type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
ServCategory	char(1)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblServices:	tblServices depends on:
	

Reference to Data Model Entity: Medical Item

Notes:

ServType could contain the following options:

- 'P' = Preventative
- 'C' = Curative

ServLevel:

- 'S' = Simple service
- 'V' = Visit
- 'D' = Day of stay
- 'H' = Hospital case

ServCareType will contain a bitmask for each of the following options:
--

- I = In patient
- O = Out-patient
- B = Both

ServPatCat will contain a bitmask for each of the following options:
--

- 1 = Man
- 2 = Woman
- 4 = Adult
- 8 = Child

6.1.14. tbIPLServices

	Column Name	Data Type	Allow Nulls
PK	PLServiceID	int	<input type="checkbox"/>
	PLServiceName	nvarchar(100)	<input type="checkbox"/>
	DatePL	date	<input type="checkbox"/>
	LocationId	int	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

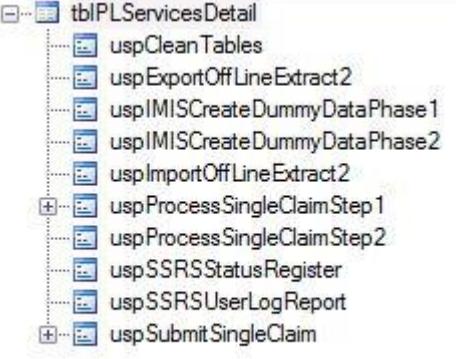
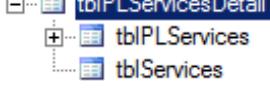
Objects that depend on tbIPLServices:	tbIPLServices depends on:
<ul style="list-style-type: none"> -> tblIPLServices + tblIHF + tblIPLServicesDetail 	<ul style="list-style-type: none"> -> tblIPLServices + tblDistricts

Reference to Data Model Entity: Pricelist of Services	
Notes: DistrictID can contain null. This means that in that case all districts could use this pricelist definition	

6.1.15. tbIPLServicesDetail

	Column Name	Data Type	Allow Nulls
PK	PLServiceDetailID	int	<input type="checkbox"/>
	PLServiceID	int	<input type="checkbox"/>
	ServiceID	int	<input type="checkbox"/>
	PriceOverrule	decimal(18, 2)	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Objects that depend on tbIPLServicesDetail:	tbIPLServicesDetail depends on:
---	---------------------------------

	
<p>Reference to Data Model Entity: Remark4</p>	
<p>Notes: This table maps items with pricelists (PLServiceID) with the option to overrule the standard price from tblServices (ServiceID)</p>	

6.1.16. tblHF

	HfID	int	
	HFCode	nvarchar(8)	<input type="checkbox"/>
	HFName	nvarchar(100)	<input type="checkbox"/>
	LegalForm	char(1)	<input type="checkbox"/>
	HFLevel	char(1)	<input type="checkbox"/>
	HFSublevel	char(1)	<input checked="" type="checkbox"/>
	HFAddress	nvarchar(100)	<input checked="" type="checkbox"/>
	LocationId	int	<input type="checkbox"/>
	Phone	nvarchar(50)	<input checked="" type="checkbox"/>
	Fax	nvarchar(50)	<input checked="" type="checkbox"/>
	eMail	nvarchar(50)	<input checked="" type="checkbox"/>
	HFCareType	char(1)	<input type="checkbox"/>
	PLServiceID	int	<input checked="" type="checkbox"/>
	PLItemID	int	<input checked="" type="checkbox"/>
	AccCode	nvarchar(25)	<input checked="" type="checkbox"/>
	OffLine	bit	<input type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Objects that depend on tblHF:	tblHF depends on:
<ul style="list-style-type: none"> -> tblHF <ul style="list-style-type: none"> -> tblClaimAdmin + tblInsuree 	<ul style="list-style-type: none"> -> tblHF <ul style="list-style-type: none"> -> tblHFSublevel -> tblLegalForms + tblPLItems -> tblPLServices

Reference to Data Model Entity: **Health facility**

Notes: We might want to include in the design an extra field for the facility to indicate if the HF is online OR offline (e.g. isOffline bit). Also it might be useful to add an HFCode (nvarchar(4)) unique to identify the offline database.

LegalForm:

- ‘C’ = Organization/Charity
- ‘D’ = Government/District
- ‘P’ = Organization/Private

HFLevel:

- ‘S’ = Sub health post
- ‘P’ = Heath post
- ‘C’ = Primary Health Centre
- ‘H’ = Hospital

HFCareType:

- 'I' = Inpatient
- 'O' = Outpatient
- 'B' = Both

6.1.17. tblFamilies

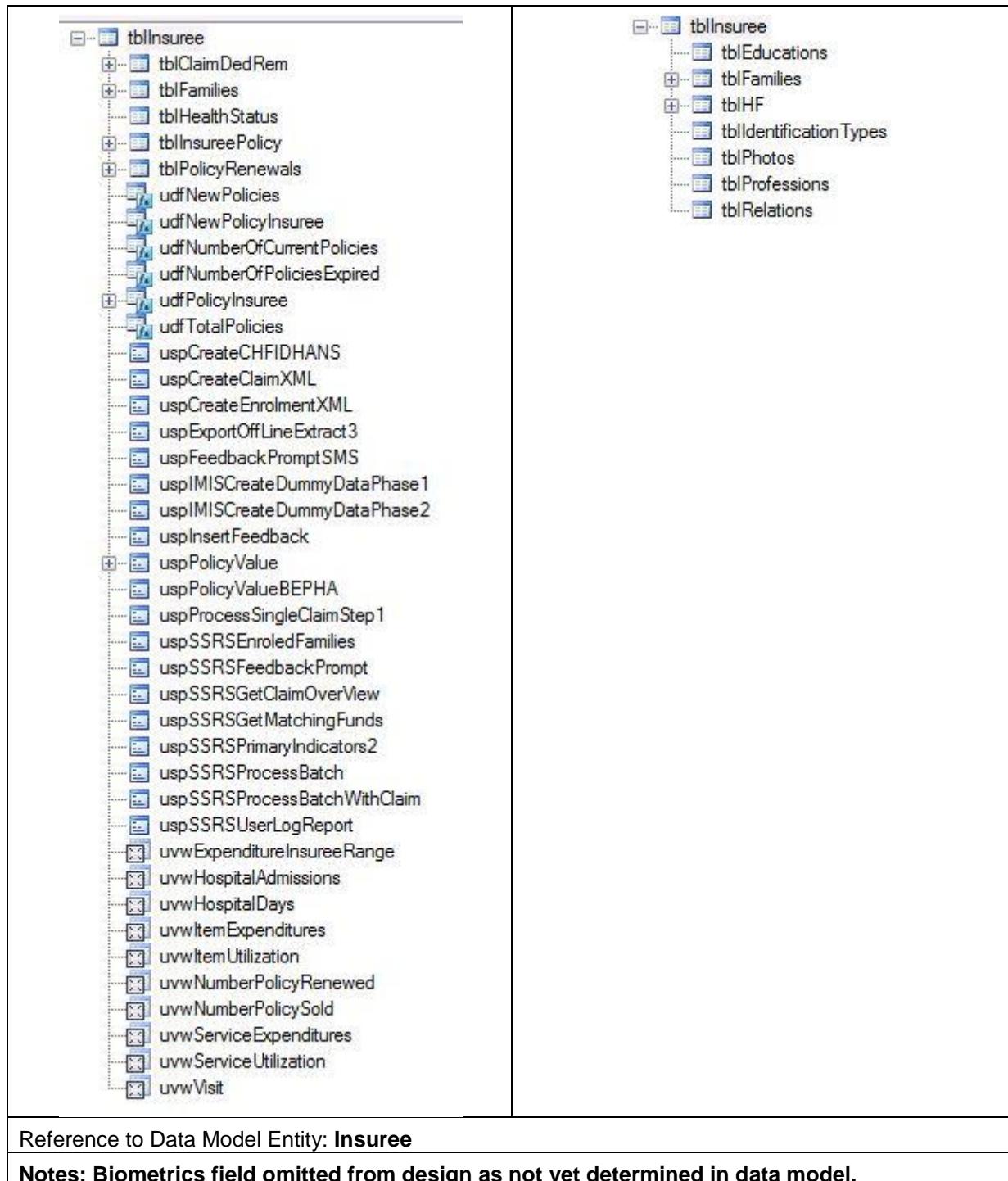
	Column Name	Data Type	Allow Nulls
key	FamilyID	int	<input type="checkbox"/>
	InsureeID	int	<input type="checkbox"/>
	LocationId	int	<input type="checkbox"/>
	Poverty	bit	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>
	FamilyType	nvarchar(2)	<input checked="" type="checkbox"/>
	FamilyAddress	nvarchar(200)	<input checked="" type="checkbox"/>
►	isOffline	bit	<input checked="" type="checkbox"/>
	Ethnicity	nvarchar(1)	<input checked="" type="checkbox"/>
	ConfirmationNo	nvarchar(12)	<input checked="" type="checkbox"/>
	ConfirmationType	nvarchar(3)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Objects that depend on tblFamilies: tblFamilies depends on:

<pre> □...□ tblFamilies □...□ tblInsuree □...□ tblPolicy □ udfAvailablePremium □ udfExpiredPolicies □ udfNewlyPremiumCollected □ udfNewPolicies □ udfNewPolicyInsuree □ udfNumberOfCurrentInsuree □ udfNumberOfCurrentPolicies □ udfNumberOfPoliciesExpired □ udfPolicyInsuree □ udfPolicyRenewal □ udfPolicyRenewal2 □ udfSuspendedPolicies □ udfTotalPolicies □ uspAddFund □ uspCleanTables □ uspCreateEnrolmentXML □ uspExportOffLineExtract3 □ uspExportOffLineExtract4 □ uspFeedbackPromptSMS □ uspIMISCreateDummyDataPhase1 □ uspIMISCreateDummyDataPhase2 □ uspImportOffLineExtract4 □ uspPolicyInquiry □ uspPolicyInquiry2 □ uspPolicyRenewalInserts □ uspPolicyRenewalRpt □ uspPolicyRenewalSMS □ uspPremiumAllocated □ uspProcessSingleClaimStep1 □ uspSSRSEnroledFamilies □ uspSSRSFeedbackPrompt □ uspSSRSGetMatchingFunds □ uspSSRSPaymentCategoryOverview □ uspSSRSPolicyRenewalPromptJournal □ uspSSRSPolicyStatus □ uspSSRSPremiumCollection □ uspSSRSProcessBatch □ uspSSRSProcessBatchWithClaim □ uspSSRSProductSales □ uspSSRSUserLogReport </pre>	<pre> □...□ tblFamilies □...□ tblConfirmationTypes □...□ tblFamilyTypes □...□ tblInsuree □...□ tblLocations </pre>
<p>Reference to Data Model Entity: Insured family/group. Insuree field will determine the head of family/group. This field is included for simplicity of historic reporting as basically the family/group changes by change of head of family/group.</p>	
<p>Notes:</p>	

6.1.18. tbInsuree

Column Name	Data Type	Allow Nulls
InsureeID	int	<input checked="" type="checkbox"/>
FamilyID	int	<input checked="" type="checkbox"/>
CHFID	nvarchar(12)	<input checked="" type="checkbox"/>
LastName	nvarchar(100)	<input checked="" type="checkbox"/>
OtherNames	nvarchar(100)	<input checked="" type="checkbox"/>
DOB	date	<input checked="" type="checkbox"/>
Gender	char(1)	<input checked="" type="checkbox"/>
Marital	char(1)	<input checked="" type="checkbox"/>
IsHead	bit	<input checked="" type="checkbox"/>
passport	nvarchar(25)	<input checked="" type="checkbox"/>
Phone	nvarchar(50)	<input checked="" type="checkbox"/>
PhotoID	int	<input checked="" type="checkbox"/>
PhotoDate	date	<input checked="" type="checkbox"/>
CardIssued	bit	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input checked="" type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input checked="" type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
Relationship	smallint	<input checked="" type="checkbox"/>
Profession	smallint	<input checked="" type="checkbox"/>
Education	smallint	<input checked="" type="checkbox"/>
Email	nvarchar(100)	<input checked="" type="checkbox"/>
isOffline	bit	<input checked="" type="checkbox"/>
TypeOfId	nvarchar(1)	<input checked="" type="checkbox"/>
HFID	int	<input checked="" type="checkbox"/>
CurrentAddress	nvarchar(200)	<input checked="" type="checkbox"/>
GeoLocation	nvarchar(250)	<input checked="" type="checkbox"/>
CurrentVillage	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>
Objects that depend on tbInsuree:	tbInsuree depends on:	



Reference to Data Model Entity: **Insuree**

Notes: Biometrics field omitted from design as not yet determined in data model.

6.1.19. tbllInsureePolicy

	Column Name	Data Type	Allow Nulls
	InsureePolicyId	int	<input type="checkbox"/>
	InsureeId	int	<input checked="" type="checkbox"/>
	PolicyId	int	<input checked="" type="checkbox"/>
	EnrollmentDate	date	<input checked="" type="checkbox"/>
	StartDate	date	<input checked="" type="checkbox"/>
	EffectiveDate	date	<input checked="" type="checkbox"/>
	ExpiryDate	date	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input checked="" type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyId	int	<input checked="" type="checkbox"/>
	AuditUserId	int	<input checked="" type="checkbox"/>
	isOffline	bit	<input checked="" type="checkbox"/>
	RowId	timestamp	<input checked="" type="checkbox"/>

Objects that depend on tbllInsureePolicy:	tbllInsureePolicy depends on:
<ul style="list-style-type: none"> [-] tbllInsureePolicy <ul style="list-style-type: none"> [+/-] udfNumberOfCurrentInsuree [+/-] udfPremiumComposition [+/-] uspAddFund [+/-] uspAddInsureePolicy [+/-] uspCleanTables [+/-] uspExportOffLineExtract4 [+/-] uspImportOffLineExtract4 [+/-] uspIsValidRenewal [+/-] uspPolicyInquiry [+/-] uspProcessSingleClaimStep2 [+/-] uspServiceItemEnquiry [+/-] uspSSRSPaymentCategoryOverview [+/-] uspSubmitSingleClaim [+/-] uvwNumberInsureeAcquired 	<ul style="list-style-type: none"> [-] tbllInsureePolicy <ul style="list-style-type: none"> [+/-] tblPolicy
Reference to Data Model Entity: N/A	
Notes: This table store only active insuree	

6.1.20. tblHealthStatus

	Column Name	Data Type	Allow Nulls
▼	HealthStatusID	int	<input type="checkbox"/>
	InsureeID	int	<input type="checkbox"/>
	Description	nvarchar(255)	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input checked="" type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	AuditUserID	int	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>

Objects that depend on tblHealthStatus:	tblHealthStatus depends on:
.....tblHealthStatustblHealthStatustblInsuree

Reference to Data Model Entity: N/A (will be determined later)

Notes: This table has no further function in the current IMIS design but included for future use.

6.1.21. tblPhotos

	Column Name	Data Type	Allow Nulls
▼	PhotoID	int	<input type="checkbox"/>
	InsureeID	int	<input checked="" type="checkbox"/>
	CHFID	nvarchar(9)	<input type="checkbox"/>
	PhotoFolder	nvarchar(255)	<input type="checkbox"/>
	PhotoFileName	nvarchar(50)	<input type="checkbox"/>
	OfficerID	int	<input type="checkbox"/>
	PhotoDate	date	<input type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	AuditUserID	int	<input checked="" type="checkbox"/>
	RowID	timestamp	<input checked="" type="checkbox"/>

Objects that depend on tblPhotos:	tblPhotos depends on:
.....tblPhotostblInsureeuspCleanTablesuspExportOffLineExtract4uspImportOffLineExtract4uspPolicyInquiryuspPolicyInquiry2uspPolicyRenewalInsertsuspSSRSUserLogReportuspUploadEnrolmentFromPhoneuspUploadEnrolmentstblPhotos

Reference to Data Model Entity: N/A

Notes: This table has references to externally stored photo image files.

6.1.22. tblPolicy

Column Name	Data Type	Allow Nulls
PolicyID	int	<input type="checkbox"/>
FamilyID	int	<input type="checkbox"/>
EnrollDate	date	<input type="checkbox"/>
StartDate	date	<input type="checkbox"/>
EffectiveDate	date	<input checked="" type="checkbox"/>
ExpiryDate	date	<input checked="" type="checkbox"/>
PolicyStatus	tinyint	<input checked="" type="checkbox"/>
PolicyValue	decimal(18, 2)	<input checked="" type="checkbox"/>
ProdID	int	<input type="checkbox"/>
OfficerID	int	<input type="checkbox"/>
PolicyStage	char(1)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
isOffline	bit	<input checked="" type="checkbox"/>

Objects that depend on tblPolicy: tblPolicy depends on:

Reference to Data Model Entity: Policy	
Notes: <p>PolicyStatus:</p> <ul style="list-style-type: none"> • 1 = Entered • 2 = Active • 4 = Suspended • 8 = Expired 	

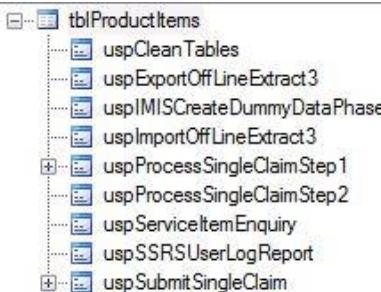
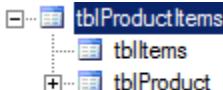
6.1.23. tblProduct

ProdID	int	<input type="checkbox"/>
ProductCode	nvarchar(8)	<input type="checkbox"/>
ProductName	nvarchar(100)	<input type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
InsurancePeriod	tinyint	<input type="checkbox"/>
DateFrom	smalldatetime	<input type="checkbox"/>
DateTo	smalldatetime	<input type="checkbox"/>
ConversionProdID	int	<input checked="" type="checkbox"/>
LumpSum	decimal(18, 2)	<input type="checkbox"/>
MemberCount	smallint	<input type="checkbox"/>
PremiumAdult	decimal(18, 2)	<input checked="" type="checkbox"/>
PremiumChild	decimal(18, 2)	<input checked="" type="checkbox"/>
DedInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
DedOPInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
DedIPInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxOPInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxIPInsuree	decimal(18, 2)	<input checked="" type="checkbox"/>
PeriodRelPrices	char(1)	<input checked="" type="checkbox"/>
PeriodRelPricesOP	char(1)	<input checked="" type="checkbox"/>
PeriodRelPricesIP	char(1)	<input checked="" type="checkbox"/>
AccCodePremiums	nvarchar(25)	<input checked="" type="checkbox"/>
AccCodeRemuneration	nvarchar(25)	<input checked="" type="checkbox"/>
DedTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
► DedOPTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
DedIPTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxOPTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxIPTreatment	decimal(18, 2)	<input checked="" type="checkbox"/>
DedPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
DedOPPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
DedIPPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxOPPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
MaxIPPolicy	decimal(18, 2)	<input checked="" type="checkbox"/>
GracePeriod	int	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>

Objects that depend on tblProduct:	tblProduct depends on:
<ul style="list-style-type: none"> -> <input type="checkbox"/> tblProduct + <input type="checkbox"/> tblClaimItems + <input type="checkbox"/> tblClaimServices + <input type="checkbox"/> tblPolicy + <input type="checkbox"/> tblPolicyRenewals + <input type="checkbox"/> tblProductItems + <input type="checkbox"/> tblProductServices + <input type="checkbox"/> tblRelDistr + <input type="checkbox"/> tblRelIndex + <input type="checkbox"/> udfNumberOfCurrentInsuree + <input type="checkbox"/> udfNumberOfCurrentPolicies + <input type="checkbox"/> udfNumberOfPoliciesExpired + <input type="checkbox"/> udfPolicyInsuree + <input type="checkbox"/> udfPolicyRenewal2 + <input type="checkbox"/> udfPremiumComposition + <input type="checkbox"/> udfTotalPolicies + <input type="checkbox"/> uspAddFund + <input type="checkbox"/> uspGetPolicyPeriod + <input type="checkbox"/> uspLastDateForPayment + <input type="checkbox"/> uspPolicyInquiry + <input type="checkbox"/> uspPolicyInquiry2 + <input type="checkbox"/> uspPolicyValueBEPHA + <input type="checkbox"/> uspPremiumAllocated + <input type="checkbox"/> uspSSRSGetMatchingFunds + <input type="checkbox"/> uspSSRSPaymentCategoryOverview + <input type="checkbox"/> uspSSRSPremiumCollection + <input type="checkbox"/> uspSSRSPremiumDistribution + <input type="checkbox"/> uspSSRSProductSales + <input type="checkbox"/> uspSSRSStatusRegister + <input type="checkbox"/> uvwNumberInsureeAcquired + <input type="checkbox"/> uvwNumberPolicyRenewed + <input type="checkbox"/> uvwNumberPolicySold + <input type="checkbox"/> uvwPremiumCollection 	<ul style="list-style-type: none"> -> <input type="checkbox"/> tblProduct + <input type="checkbox"/> tblCeilingInterpretation + <input type="checkbox"/> tblHFSublevel + <input type="checkbox"/> tblLocations
Reference to Data Model Entity: Insurance product	
Notes: ConversionID refers to a ProductID previously entered (self referenced). ConversionID will be Null if current entry was not referenced.	

6.1.24. tblProductItems

Column Name	Data Type	Allow Nulls
ProdItemID	int	<input type="checkbox"/>
ProdID	int	<input type="checkbox"/>
ItemID	int	<input type="checkbox"/>
LimitationType	char(1)	<input type="checkbox"/>
PriceOrigin	char(1)	<input type="checkbox"/>
LimitAdult	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChild	decimal(18, 2)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
WaitingPeriodAdult	int	<input checked="" type="checkbox"/>
WaitingPeriodChild	int	<input checked="" type="checkbox"/>
LimitNoAdult	int	<input checked="" type="checkbox"/>
LimitNoChild	int	<input checked="" type="checkbox"/>
LimitationTypeR	char(1)	<input checked="" type="checkbox"/>
LimitationTypeE	char(1)	<input checked="" type="checkbox"/>
LimitAdultR	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitAdultE	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChildR	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChildE	decimal(18, 2)	<input checked="" type="checkbox"/>
CeilingExclusionAdult	nvarchar(1)	<input checked="" type="checkbox"/>
CeilingExclusionChild	nvarchar(1)	<input checked="" type="checkbox"/>

Objects that depend on tblProductItems:	tblProductItems depends on:
	

Reference to Data Model Entity: Remark2
--

Notes:

LimitationType:

- 'C' = Co-Insurance
- 'F' = Fixed Limit

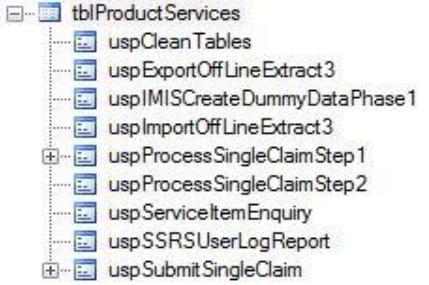
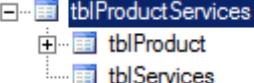
PriceOrigin:

- 'P' = Price List
- 'O' = Own providers price
- 'R' = Relative Price

LimitAdult,LimitChild → will hold percentage covered by scheme Administrator for LimitationType = 'C' or limit to cover by scheme Administrator for Limitationtype = 'F'

6.1.25. tblProductServices

Column Name	Data Type	Allow Nulls
ProdServiceID	int	<input type="checkbox"/>
ProdID	int	<input type="checkbox"/>
ServiceID	int	<input type="checkbox"/>
LimitationType	char(1)	<input type="checkbox"/>
PriceOrigin	char(1)	<input type="checkbox"/>
LimitAdult	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChild	decimal(18, 2)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
WaitingPeriodAdult	int	<input checked="" type="checkbox"/>
WaitingPeriodChild	int	<input checked="" type="checkbox"/>
LimitNoAdult	int	<input checked="" type="checkbox"/>
LimitNoChild	int	<input checked="" type="checkbox"/>
LimitationTypeR	char(1)	<input checked="" type="checkbox"/>
LimitationTypeE	char(1)	<input checked="" type="checkbox"/>
LimitAdultR	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitAdultE	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChildR	decimal(18, 2)	<input checked="" type="checkbox"/>
LimitChildE	decimal(18, 2)	<input checked="" type="checkbox"/>
CeilingExclusionAdult	nvarchar(1)	<input checked="" type="checkbox"/>
CeilingExclusionChild	nvarchar(1)	<input checked="" type="checkbox"/>

Objects that depend on tblProductServices:	tblProductServices depends on:
	

Reference to Data Model Entity: Remark1	
<p>Notes:</p> <p>LimitationType:</p> <ul style="list-style-type: none"> • 'C' = Co-Insurance • 'F' = Fixed Limit <p>PriceOrigin:</p> <ul style="list-style-type: none"> • 'P' = Price List • 'O' = Own providers price • 'R' = Relative Price <p>LimitAdult,LimitChild → will hold percentage covered by IMIS for LimitationType = 'C' or limit to cover by IMIS for Limitationtype = 'F'</p>	

6.1.26. tblRelDistr

Column Name	Data Type	Allow Nulls
DistrID	int	<input type="checkbox"/>
DistrType	tinyint	<input type="checkbox"/>
DistrCareType	char(1)	<input type="checkbox"/>
ProdID	int	<input type="checkbox"/>
Period	tinyint	<input type="checkbox"/>
DistrPerc	decimal(18, 2)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>

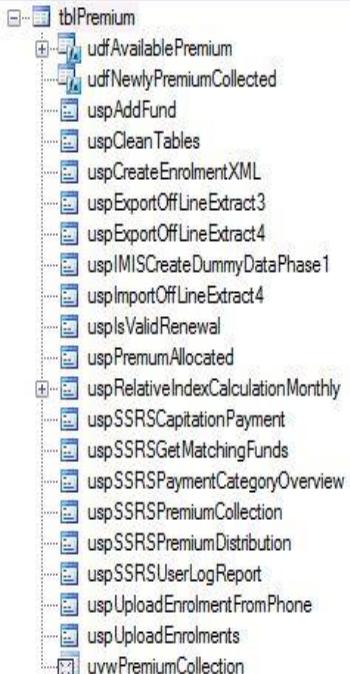
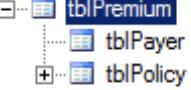
Objects that depend on tblRelDistr:	tblRelDistr depends on:
... tblRelDistr	... tblRelDistr ... tblProduct

Reference to Data Model table: Indexes of relative prices
--

Notes:
DistrType could have three values
<ul style="list-style-type: none"> • 1 = Yearly record • 4 = Quarterly record • 12 = Monthly record
DistrCareType:
<ul style="list-style-type: none"> • 'I' = In patient • 'O' = Out patient • 'B' = Both

6.1.27. tblPremium

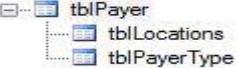
Column Name	Data Type	Allow Nulls
PremiumId	int	<input type="checkbox"/>
PolicyID	int	<input type="checkbox"/>
PayerID	int	<input checked="" type="checkbox"/>
Amount	decimal(18, 2)	<input type="checkbox"/>
Receipt	nvarchar(50)	<input type="checkbox"/>
PayDate	date	<input type="checkbox"/>
PayType	char(1)	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
isPhotoFee	bit	<input checked="" type="checkbox"/>
isOffline	bit	<input checked="" type="checkbox"/>
ReportingId	int	<input checked="" type="checkbox"/>

Objects that depend on tblPremium:	tblPremium depends on:
 <ul style="list-style-type: none"> tblPremium udfAvailablePremium udfNewlyPremiumCollected uspAddFund uspCleanTables uspCreateEnrolmentXML uspExportOffLineExtract3 uspExportOffLineExtract4 uspIMISCreateDummyDataPhase1 uspImportOffLineExtract4 uspIsValidRenewal uspPremiumAllocated uspRelativeIndexCalculationMonthly uspSSRSCapitationPayment uspSSRSGetMatchingFunds uspSSRSPaymentCategoryOverview uspSSRSPremiumCollection uspSSRSPremiumDistribution uspSSRSUserLogReport uspUploadEnrolmentFromPhone uspUploadEnrolments uvwPremiumCollection 	 <ul style="list-style-type: none"> tblPremium tblPayer tblPolicy

Reference to Data Model Entity: Premium
Notes: PayerID has a value if payment NOT made by Insuree/Family/Group otherwise NULL

6.1.28. tblPayer

Column Name	Data Type	Allow Nulls
PayerID	int	<input type="checkbox"/>
PayerType	char(1)	<input type="checkbox"/>
PayerName	nvarchar(100)	<input type="checkbox"/>
PayerAddress	nvarchar(100)	<input checked="" type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
Phone	nvarchar(50)	<input checked="" type="checkbox"/>
Fax	nvarchar(50)	<input checked="" type="checkbox"/>
eMail	nvarchar(50)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblPayer:	tblPayer depends on:
	
Reference to Data Model Entity: Payer	
Notes: DistrictID will be null if Payer is available for more than one district.	

6.1.29. tblClaim

ClaimID	int	<input type="checkbox"/>
InsureeID	int	<input type="checkbox"/>
ClaimCode	nvarchar(8)	<input type="checkbox"/>
DateFrom	smalldatetime	<input type="checkbox"/>
DateTo	smalldatetime	<input checked="" type="checkbox"/>
ICDID	int	<input type="checkbox"/>
ClaimStatus	tinyint	<input type="checkbox"/>
Adjuster	int	<input checked="" type="checkbox"/>
Adjustment	ntext	<input checked="" type="checkbox"/>
Claimed	decimal(18, 2)	<input checked="" type="checkbox"/>
Approved	decimal(18, 2)	<input checked="" type="checkbox"/>
Reinsured	decimal(18, 2)	<input checked="" type="checkbox"/>
Valuated	decimal(18, 2)	<input checked="" type="checkbox"/>
DateClaimed	date	<input type="checkbox"/>
DateProcessed	smalldatetime	<input checked="" type="checkbox"/>
Feedback	bit	<input type="checkbox"/>
FeedbackID	int	<input checked="" type="checkbox"/>
Explanation	ntext	<input checked="" type="checkbox"/>
FeedbackStatus	tinyint	<input checked="" type="checkbox"/>
ReviewStatus	tinyint	<input checked="" type="checkbox"/>
ApprovalStatus	tinyint	<input checked="" type="checkbox"/>
RejectionReason	tinyint	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
ValidityFromReview	datetime	<input checked="" type="checkbox"/>
ValidityToReview	datetime	<input checked="" type="checkbox"/>
AuditUserIDReview	int	<input checked="" type="checkbox"/>
RowID	timestamp	<input checked="" type="checkbox"/>
HFID	int	<input type="checkbox"/>
RunID	int	<input checked="" type="checkbox"/>
AuditUserIDSubmit	int	<input checked="" type="checkbox"/>
AuditUserIDProcess	int	<input checked="" type="checkbox"/>
SubmitStamp	datetime	<input checked="" type="checkbox"/>
ProcessStamp	datetime	<input checked="" type="checkbox"/>
Remunerated	decimal(18, 2)	<input checked="" type="checkbox"/>
GuaranteeId	nvarchar(50)	<input checked="" type="checkbox"/>
ClaimAdminId	int	<input checked="" type="checkbox"/>
ICDID1	int	<input checked="" type="checkbox"/>
ICDID2	int	<input checked="" type="checkbox"/>
ICDID3	int	<input checked="" type="checkbox"/>
ICDID4	int	<input checked="" type="checkbox"/>
VisitType	char(1)	<input checked="" type="checkbox"/>

Objects that depend on tblClaim:	tblClaim depends on:
<ul style="list-style-type: none"> -> tblClaim <ul style="list-style-type: none"> -> tblClaimItems -> tblClaimServices -> tblFeedback -> trgFeedback Prompt -> uspClaimSelection -> uspFeedbackPromptSMS -> uspInsertFeedback -> uspSSRSFeedbackPrompt -> uspSSRSUserLogReport -> uvwClaimEntered -> uvwClaimRejected -> uvwClaimSubmitted 	<ul style="list-style-type: none"> -> tblClaim <ul style="list-style-type: none"> -> tblBatchRun -> tblClaimAdmin -> tblFeedbackPrompt -> tblIHF -> tblICDCodes -> tblInsuree -> tblUsers
Reference to Data Model Entity: Claim	
<p>Notes: Feedback indicates if claim expects feedback. FeedbackID relates to a record in the feedback table if present. FeedbackID might be redundant but currently kept in design.</p>	
<p>Field ClaimStatus :</p> <ul style="list-style-type: none"> • 1 = Rejected • 2 = Entered • 4 = Checked • 8 = Processed • 16 = Valuated <p>Field ReviewStatus</p> <ul style="list-style-type: none"> • 1 = Not selected • 2 = Selected for review • 4 = Reviewed <p>Field Feedback Status:</p> <ul style="list-style-type: none"> • 1 = Not selected • 2 = Selected for feedback • 4 = Delivered 	

6.1.30. tblClaimItems

Column Name	Data Type	Allow Nulls
ClaimItemID	int	<input type="checkbox"/>
ClaimID	int	<input type="checkbox"/>
ItemID	int	<input type="checkbox"/>
ProdID	int	<input checked="" type="checkbox"/>
ClaimItemStatus	tinyint	<input type="checkbox"/>
Availability	bit	<input type="checkbox"/>
QtyProvided	decimal(18, 2)	<input type="checkbox"/>
QtyApproved	decimal(18, 2)	<input checked="" type="checkbox"/>
PriceAsked	decimal(18, 2)	<input type="checkbox"/>
PriceAdjusted	decimal(18, 2)	<input checked="" type="checkbox"/>
PriceApproved	decimal(18, 2)	<input checked="" type="checkbox"/>
PriceValuated	decimal(18, 2)	<input checked="" type="checkbox"/>
Explanation	ntext	<input checked="" type="checkbox"/>
Justification	ntext	<input checked="" type="checkbox"/>
RejectionReason	smallint	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
ValidityFromReview	datetime	<input checked="" type="checkbox"/>
ValidityToReview	datetime	<input checked="" type="checkbox"/>
AuditUserIDReview	int	<input checked="" type="checkbox"/>
LimitationValue	decimal(18, 2)	<input checked="" type="checkbox"/>
Limitation	char(1)	<input checked="" type="checkbox"/>
PolicyID	int	<input checked="" type="checkbox"/>
RemuneratedAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
DeductableAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
ExceedCeilingAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
PriceOrigin	char(1)	<input checked="" type="checkbox"/>
ExceedCeilingAmountCa...	decimal(18, 2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblClaimItems:	tblClaimItems depends on:
---------------------------------------	---------------------------

<pre> □...tblClaimItems +...udfRejectedClaims +...udfRemunerated +...udfTotalClaims +...uspCleanTables +...uspCreateClaimXML +...uspGetExpenditureRange +...uspIMISCreateDummyDataPhase2 +...uspProcessSingleClaimStep1 +...uspProcessSingleClaimStep2 +...uspRelativeIndexCalculationMonthly +...uspServiceItemEnquiry +...uspSSRSCapitationPayment +...uspSSRSGetClaimOverView +...uspSSRSProcessBatch +...uspSSRSProcessBatchWithClaim +...uspSubmitSingleClaim +...uvwAmountApproved +...uvwAmountClaimed +...uvwAmountRejected +...uvwamountvaluated +...uvwClaimProcessed +...uvwClaimSent +...uvwClaimValuated +...uvwExpenditureInsureeRange +...uvwHospitalAdmissions +...uvwHospitalDays +...uvwItemExpenditures +...uvwItemUtilization +...uvwNumberFeedbackAnswerYes +...uvwNumberFeedbackResponded +...uvwNumberFeedbackSent +...uvwOverallAssessment +...uvwVisit </pre>	<pre> □...tblClaimItems +...tblClaim -...tblItems +...tblProduct </pre>
Reference to Data Model Entity: Remark7	
Notes:	
<p>Field ClaimItemStatus:</p> <ul style="list-style-type: none"> • 1 = Rejected • 2 = Passed <p>Field RejectionReason is reserved for possible use later</p>	

6.1.31. tblClaimServices

	Column Name	Data Type	Allow Nulls
PK	ClaimServiceID	int	<input type="checkbox"/>
	ClaimID	int	<input type="checkbox"/>
	ServiceID	int	<input type="checkbox"/>
	ProdID	int	<input checked="" type="checkbox"/>
	ClaimServiceStatus	tinyint	<input type="checkbox"/>
	QtyProvided	decimal(18, 2)	<input type="checkbox"/>
	QtyApproved	decimal(18, 2)	<input checked="" type="checkbox"/>
	PriceAsked	decimal(18, 2)	<input type="checkbox"/>
	PriceAdjusted	decimal(18, 2)	<input checked="" type="checkbox"/>
	PriceApproved	decimal(18, 2)	<input checked="" type="checkbox"/>
	PriceValuated	decimal(18, 2)	<input checked="" type="checkbox"/>
	Explanation	ntext	<input checked="" type="checkbox"/>
	Justification	ntext	<input checked="" type="checkbox"/>
	RejectionReason	smallint	<input checked="" type="checkbox"/>
	ValidityFrom	datetime	<input type="checkbox"/>
	ValidityTo	datetime	<input checked="" type="checkbox"/>
	LegacyID	int	<input checked="" type="checkbox"/>
	AuditUserID	int	<input type="checkbox"/>
	ValidityFromReview	datetime	<input checked="" type="checkbox"/>
	ValidityToReview	datetime	<input checked="" type="checkbox"/>
	AuditUserIDReview	int	<input checked="" type="checkbox"/>
	LimitationValue	decimal(18, 2)	<input checked="" type="checkbox"/>
	Limitation	char(1)	<input checked="" type="checkbox"/>
	PolicyID	int	<input checked="" type="checkbox"/>
	RemuneratedAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
	DeductableAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
	ExceedCeilingAmount	decimal(18, 2)	<input checked="" type="checkbox"/>
	PriceOrigin	char(1)	<input checked="" type="checkbox"/>
	ExceedCeilingAmountCa...	decimal(18, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
Objects that depend on tblClaimServices:		tblClaimServices depends on:	

<pre> □...tblClaimServices +□ udfRejectedClaims +□ udfRemunerated +□ udfTotalClaims +□ uspCleanTables +□ uspCreateClaimXML +□ uspGetExpenditureRange +□ uspIMISCreateDummyDataPhase2 +□ uspProcessSingleClaimStep1 +□ uspProcessSingleClaimStep2 +□ uspRelativeIndexCalculationMonthly +□ uspServiceItemEnquiry +□ uspSSRSCapitationPayment +□ uspSSRSGetClaimOverView +□ uspSSRSProcessBatch +□ uspSSRSProcessBatchWithClaim +□ uspSubmitSingleClaim +□ uvwAmountApproved +□ uvwAmountClaimed +□ uvwAmountRejected +□ uvwamountvaluated +□ uvwClaimProcessed +□ uvwClaimSent +□ uvwClaimValuated +□ uvwExpenditureInsureeRange +□ uvwHospitalAdmissions +□ uvwHospitalDays +□ uvwNumberFeedbackAnswerYes +□ uvwNumberFeedbackResponded +□ uvwNumberFeedbackSent +□ uvwOverallAssessment +□ uvwServiceExpenditures +□ uvwServiceUtilization +□ uvwVisit </pre>	<pre> □...tblClaimServices +□ tblClaim +□ tblProduct +□ tblServices </pre>
Reference to Data Model Entity: Remark6	
<p>Notes:</p> <p>Field ClaimServiceStatus:</p> <ul style="list-style-type: none"> • 1 = Rejected • 2 = Passed <p>Field RejectionReason is reserved for possible use later</p>	

6.1.32. tblClaimDedRem

Column Name	Data Type	Allow Nulls
ExpenditureID	int	<input type="checkbox"/>
PolicyID	int	<input type="checkbox"/>
InsureeID	int	<input type="checkbox"/>
ClaimID	int	<input type="checkbox"/>
DedG	decimal(18, 2)	<input checked="" type="checkbox"/>
DedOP	decimal(18, 2)	<input checked="" type="checkbox"/>
DedIP	decimal(18, 2)	<input checked="" type="checkbox"/>
RemG	decimal(18, 2)	<input checked="" type="checkbox"/>
RemIP	decimal(18, 2)	<input checked="" type="checkbox"/>
RemOP	decimal(18, 2)	<input checked="" type="checkbox"/>
RemConsult	decimal(18, 2)	<input checked="" type="checkbox"/>
RemSurgery	decimal(18, 2)	<input checked="" type="checkbox"/>
RemDelivery	decimal(18, 2)	<input checked="" type="checkbox"/>
RemHospitalization	decimal(18, 2)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RemAntenatal	decimal(18, 2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblClaimDedRem:	tblClaimDedRem depends on:
<ul style="list-style-type: none"> -> tblClaimDedRem <ul style="list-style-type: none"> -> tblCleanTables -> tblPolicyInquiry -> tblPolicyInquiry2 +> tblProcessSingleClaimStep2 -> tblServiceItemEnquiry 	<ul style="list-style-type: none"> -> tblClaimDedRem <ul style="list-style-type: none"> +> tblInsuree +> tblPolicy
Reference to Data Model Entity: Expenditures for insuree	
Notes:	

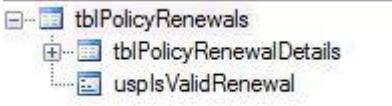
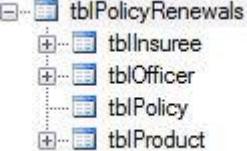
6.1.33. tblFeedback

Column Name	Data Type	Allow Nulls
FeedbackID	int	<input type="checkbox"/>
ClaimID	int	<input type="checkbox"/>
CareRendered	bit	<input checked="" type="checkbox"/>
PaymentAsked	bit	<input checked="" type="checkbox"/>
DrugPrescribed	bit	<input checked="" type="checkbox"/>
DrugReceived	bit	<input checked="" type="checkbox"/>
Assessment	tinyint	<input checked="" type="checkbox"/>
CHFOfficerCode	int	<input checked="" type="checkbox"/>
FeedbackDate	datetime	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>

Objects that depend on tblFeedback:	tblFeedback depends on:
<ul style="list-style-type: none"> - tblFeedback + tblClaim uspCleanTables uspInsertFeedback uspSSRSDerivedIndicators2 uspSSRSUserLogReport uvwNumberFeedbackAnswerYes uvwNumberFeedbackResponded uvwOverallAssessment 	<ul style="list-style-type: none"> - tblFeedback + tblClaim
Reference to Data Model Entity: Feedback	
Notes: ClaimID might be redundant but currently kept in design.	

6.1.34. tblPolicyRenewals

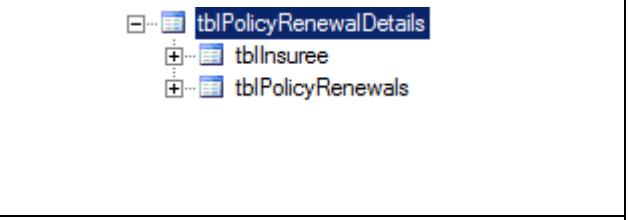
Column Name	Data Type	Allow Nulls
RenewalID	int	<input type="checkbox"/>
RenewalPromptDate	date	<input type="checkbox"/>
RenewalDate	date	<input type="checkbox"/>
NewOfficerID	int	<input checked="" type="checkbox"/>
PhoneNumber	nvarchar(25)	<input checked="" type="checkbox"/>
SMSStatus	tinyint	<input type="checkbox"/>
InsureeID	int	<input type="checkbox"/>
PolicyID	int	<input type="checkbox"/>
NewProdID	int	<input type="checkbox"/>
RenewalWarnings	tinyint	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditCreateUser	int	<input checked="" type="checkbox"/>
ResponseStatus	int	<input checked="" type="checkbox"/>
ResponseDate	datetime	<input checked="" type="checkbox"/>

Objects that depend on PolicyRenewals:	PolicyRenewals depends on:
	
Reference to Data Model Entity: N/A	
Notes: This table is used by a webservice that populates renewal prompts with the (new) OfficerID, (new Product code as per 'use case' 5.17. A separate service will loop through this table and will sent SMS messages to the phones and will update the SMS status.	

6.1.35. tblPolicyRenewalsDetails

Column Name	Data Type	Allow Nulls
RenewalDetailID	int	<input type="checkbox"/>
RenewalID	int	<input type="checkbox"/>
InsureeID	int	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditCreateUser	int	<input type="checkbox"/>

Objects that depend on PolicyRenewals:	PolicyRenewals depends on:
--	----------------------------

 <p>tblPolicyRenewalDetails</p> <ul style="list-style-type: none">- uspCleanTables- uspPolicyRenewalInserts- uspPolicyRenewalRpt- uspPolicyRenewalSMS- uspSSRSPolicyRenewalPromptJournal	 <p>tblPolicyRenewalDetails</p> <ul style="list-style-type: none">- tblInsuree- tblPolicyRenewals
Reference to Data Model Entity: N/A	
Notes: This table is used by a webservice that populates renewal prompts. Every insuree that needs a Photo update (internal rules) will be hosted in this table in order to alert the officer.	

6.1.36. tblRelIndex

Column Name	Data Type	Allow Nulls
RelIndexID	int	<input type="checkbox"/>
ProdID	int	<input type="checkbox"/>
RelType	tinyint	<input type="checkbox"/>
RelCareType	char(1)	<input type="checkbox"/>
RelYear	int	<input type="checkbox"/>
RelPeriod	tinyint	<input type="checkbox"/>
CalcDate	datetime	<input type="checkbox"/>
RelIndex	decimal(18, 4)	<input checked="" type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblClaimDedRem:	tblClaimDedRem depends on:
<ul style="list-style-type: none"> -> tblRelIndex <ul style="list-style-type: none"> -> tblRelIndex -> tblProduct 	
Reference to Data Model Entity: Indexes of relative prices	

Notes:

Field RelCareType:

- I = In patient
- O = OutPatient
- B = Both (general)

RelType:

- 1 = Yearly
- 4 = Quarterly
- 12 = Monthly

6.1.37. tblBatchRun

Column Name	Data Type	Allow Nulls
RunID	int	<input type="checkbox"/>
LocationId	int	<input checked="" type="checkbox"/>
RunDate	datetime	<input type="checkbox"/>
ValidityFrom	datetime	<input type="checkbox"/>
ValidityTo	datetime	<input checked="" type="checkbox"/>
LegacyID	int	<input checked="" type="checkbox"/>
AuditUserID	int	<input type="checkbox"/>
RunYear	int	<input type="checkbox"/>
RunMonth	tinyint	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

Objects that depend on tblBatchRun:	tblBatchRun depends on:
<ul style="list-style-type: none"> -> tblBatchRun +> tblClaim -> uspBatchProcess -> uspCleanTables -> uspSSRSDerivedIndicators1 -> uspSSRSUserLogReport 	<ul style="list-style-type: none"> -> tblBatchRun -> tblLocations
Reference to Data Model Entity: N/A	
Notes: This table is used to hold the batches that were fully 'valuated' together for reporting toward accounting	

6.1.38. tbIMISDefaults

	Column Name	Data Type	Allow Nulls		
	DefaultID	int	<input checked="" type="checkbox"/>		
	PolicyRenewalInterval	int	<input checked="" type="checkbox"/>		
	FTPHost	nvarchar(50)	<input checked="" type="checkbox"/>		
	FTPUser	nvarchar(50)	<input checked="" type="checkbox"/>		
	FTPPassword	nvarchar(20)	<input checked="" type="checkbox"/>		
	FTPPort	int	<input checked="" type="checkbox"/>		
	FTPEnrollmentFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	AssociatedPhotoFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	FTPClaimFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	FTPFeedbackFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	FTPPolicyRenewalFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	FTPPhoneExtractFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	FTPOffLineExtractFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	AppVersionBackEnd	decimal(3, 1)	<input checked="" type="checkbox"/>		
	AppVersionEnquire	decimal(3, 1)	<input checked="" type="checkbox"/>		
	AppVersionEnroll	decimal(3, 1)	<input checked="" type="checkbox"/>		
	AppVersionRenewal	decimal(3, 1)	<input checked="" type="checkbox"/>		
	AppVersionFeedback	decimal(3, 1)	<input checked="" type="checkbox"/>		
	AppVersionClaim	decimal(3, 1)	<input checked="" type="checkbox"/>		
	OffLineHF	int	<input checked="" type="checkbox"/>		
	WinRarFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	DatabaseBackupFolder	nvarchar(255)	<input checked="" type="checkbox"/>		
	OfflineCHF	int	<input checked="" type="checkbox"/>		
	SMSLink	nvarchar(500)	<input checked="" type="checkbox"/>		
	SMSIP	nvarchar(15)	<input checked="" type="checkbox"/>		
	SMSUserName	nvarchar(15)	<input checked="" type="checkbox"/>		
	SMSPassword	nvarchar(50)	<input checked="" type="checkbox"/>		
	SMSSource	nvarchar(15)	<input checked="" type="checkbox"/>		
	SMSSDlr	int	<input checked="" type="checkbox"/>		
	SMSType	int	<input checked="" type="checkbox"/>		
Objects that depend on tbIMISDefaults:		tbIMISDefaults depends on:			
		None			
Reference to Data Model Entity: N/A					
Notes:					

This table is used to hold default information for IMIS to function.
 All FTP settings are required for the mobile phones to connect to IMIS and to transfer the information to the correct folders under the IMIS root folder.
 The OfflineHF field should only contain a value <> 0 in case of an Offline installation. This field should in that case contain the HFID (DB Key of tblHF) of the Health Facility.

6.1.39. tblReporting

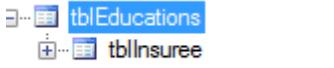
	Column Name	Data Type	Allow Nulls
▶	ReportingId	int	<input type="checkbox"/>
	ReportingDate	datetime	<input type="checkbox"/>
	LocationId	int	<input type="checkbox"/>
	ProdId	int	<input type="checkbox"/>
	PayerId	int	<input checked="" type="checkbox"/>
	StartDate	date	<input type="checkbox"/>
	EndDate	date	<input type="checkbox"/>
	RecordFound	int	<input type="checkbox"/>
			<input type="checkbox"/>
Objects that depend on tblReporting:	tblReporting depends on:		
<ul style="list-style-type: none"> ↳ ... ↳ tblReporting ↳ ... ↳ uspCleanTables ↳ ... ↳ uspSSRSGetMatchingFunds 	None		
Reference to Data Model Entity: N/A			
Notes: This table is used to hold criterias used to generate Matching Funds Reports			

6.1.40. tblControl

	Column Name	Data Type	Allow Nulls
▶	FieldName	nvarchar(50)	<input type="checkbox"/>
	Adjustability	nvarchar(1)	<input type="checkbox"/>
	Usage	nvarchar(200)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>
Objects that depend on tblControl:	tblControl depends on:		
None	None		
Reference to Data Model Entity: N/A			
Notes: This table is used to handle the accessibility of the controls on the user interface for the country specific versions			

6.1.41. tblEducations

Column Name	Data Type	Allow Nulls
EducationId	smallint	<input type="checkbox"/>
Education	nvarchar(50)	<input type="checkbox"/>
SortOrder	int	<input checked="" type="checkbox"/>
AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

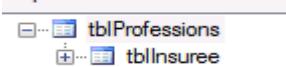
Objects that depend on tblEducations:	tblEducations depends on:
	None

Reference to Data Model Entity: **N/A**

Notes: This table is used to hold records for the list of Education Level

6.1.42. tblProfessions

Column Name	Data Type	Allow Nulls
ProfessionId	smallint	<input type="checkbox"/>
Profession	nvarchar(50)	<input type="checkbox"/>
SortOrder	int	<input checked="" type="checkbox"/>
AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

Objects that depend on tblProfessions:	tblProfessions depends on:
	None

Reference to Data Model Entity: **N/A**

Notes: This table is used to hold records for the list of Professions in the Country.

6.1.43. tbIdentificationTypes

Column Name	Data Type	Allow Nulls
IdentificationCode	nvarchar(1)	<input type="checkbox"/>
IdentificationTypes	nvarchar(50)	<input type="checkbox"/>
AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>
SortOrder	int	<input checked="" type="checkbox"/>

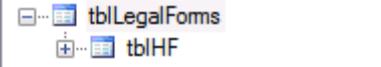
Objects that depend on tbIdentificationTypes:	tbIdentificationTypes depends on:
	None

Reference to Data Model Entity: N/A

Notes: This table is used to hold records for the list of Professions in the Country.

6.1.44. tblLegalForms

	Column Name	Data Type	Allow Nulls
Key	LegalFormCode	char(1)	<input type="checkbox"/>
	LegalForms	nvarchar(50)	<input type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

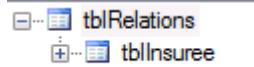
Objects that depend on tblLegalForms:	tblLegalForms depends on:
	None

Reference to Data Model Entity: N/A

Notes: This table is used to hold records for the list of Legal forms.

6.1.45. tblRelations

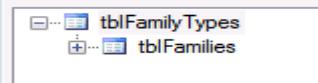
	Column Name	Data Type	Allow Nulls
Key	RelationId	smallint	<input type="checkbox"/>
	Relation	nvarchar(50)	<input type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

Objects that depend on tblRelations:	tblRelations depends on:
	None

Reference to Data Model Entity: N/A
Notes: This table is used to hold records for the list of Relationships.

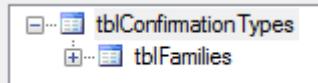
6.1.46. tblFamilyTypes

	Column Name	Data Type	Allow Null
	FamilyTypeCode	nvarchar(2)	<input type="checkbox"/>
	FamilyType	nvarchar(50)	<input type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

Objects that depend on tblFamilyTypes:	tblFamilyTypes depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used to hold records for the list of Family Types.	

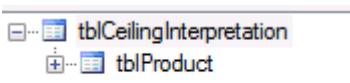
6.1.47. tblConfirmationTypes

	Column Name	Data Type	Allow Nulls
	IdentificationCode	nvarchar(1)	<input type="checkbox"/>
	IdentificationTypes	nvarchar(50)	<input type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>

Objects that depend on tblConfirmationTypes:	tblConfirmationTypes depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used to hold records for the list of Confirmation Type.	

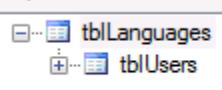
6.1.48. tblCeilingInterpretation

	Column Name	Data Type	Allow Nulls
	CeilingIntCode	char(1)	<input type="checkbox"/>
	CeilingIntDesc	nvarchar(100)	<input type="checkbox"/>
	AltLanguage	nvarchar(100)	<input checked="" type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>

Objects that depend on tblCeilingInterpretation:	tblCeilingInterpretation depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used to hold records for the list of Ceiling Interpretation.	

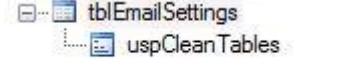
6.1.49. tblLanguages

	Column Name	Data Type	Allow Nulls
▶	LanguageCode	nvarchar(2)	<input type="checkbox"/>
	LanguageName	nvarchar(50)	<input type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>

Objects that depend on tblLanguages:	tblLanguages depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used to hold records for the list of Languages.	

6.1.50. tblEmailSettings

	Column Name	Data Type	Allow Nulls
▶	EmailId	nvarchar(200)	<input type="checkbox"/>
	EmailPassword	nvarchar(200)	<input type="checkbox"/>
	SMTPHost	nvarchar(200)	<input type="checkbox"/>
	Port	int	<input type="checkbox"/>
	EnableSSL	bit	<input type="checkbox"/>

Objects that depend on tblEmailSettings:	tblEmailSettings depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used for Handling email setting.	

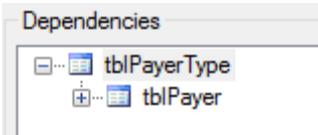
6.1.51. tblHFSublevel

	Column Name	Data Type	Allow Nulls
▶	HFSublevel	char(1)	<input type="checkbox"/>
	HFSublevelDesc	nvarchar(50)	<input checked="" type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>

Objects that depend on tblHFSubLevel:	tblHFSubLevel depends on:
	None
Reference to Data Model Entity: N/A	

6.1.52. tblPayerType

	Column Name	Data Type	Allow Nulls
1	Code	char(1)	<input type="checkbox"/>
	PayerType	nvarchar(50)	<input type="checkbox"/>
	AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>
	SortOrder	int	<input checked="" type="checkbox"/>

Objects that depend on tblPayerType	tblPayerType Depends on:
	None
Reference to Data Model Entity: N/A	

6.1.53. tblFromPhone

	Column Name	Data Type	Allow Nulls
1	FromPhoneId	int	<input type="checkbox"/>
	DocType	nvarchar(3)	<input type="checkbox"/>
	DocName	nvarchar(200)	<input type="checkbox"/>
	DocStatus	nvarchar(3)	<input checked="" type="checkbox"/>
	LandedDate	datetime	<input type="checkbox"/>
	OfficerCode	nvarchar(8)	<input checked="" type="checkbox"/>
	CHFID	nvarchar(9)	<input checked="" type="checkbox"/>
	PhotoSubmittedDate	datetime	<input checked="" type="checkbox"/>
	ClaimId	int	<input checked="" type="checkbox"/>

Objects that depend on tblFromPhone:	tblFromPhone depends on:
	None
Reference to Data Model Entity: N/A	
Notes: This table is used for Handling all records which are uploaded from the phone.	

6.1.54. tblSubmittedPhotos

Column Name	Data Type	Allow Nulls
PhotoId	int	<input type="checkbox"/>
ImageName	nvarchar(50)	<input checked="" type="checkbox"/>
CHFID	nvarchar(9)	<input checked="" type="checkbox"/>
OfficerCode	nvarchar(8)	<input checked="" type="checkbox"/>
PhotoDate	date	<input checked="" type="checkbox"/>
RegisterDate	datetime	<input checked="" type="checkbox"/>

Objects that depend on tblSubmittedPhotos:	tblSubmittedPhotos on:
tblSubmittedPhotos uspCleanTables	None

Reference to Data Model Entity: N/A
Notes: This table is used for Handling photograph records when the service runs.

6.1.55. tblLogins

Column Name	Data Type	Allow Nulls
LoginId	int	<input type="checkbox"/>
UserId	int	<input checked="" type="checkbox"/>
LogTime	datetime	<input checked="" type="checkbox"/>
LogAction	int	<input checked="" type="checkbox"/>

Objects that depend on tblLogins:	tblLogins depends on:
tblLogins uspCleanTables uspSSRSUserLogReport	tblLogins tblUsers

Reference to Data Model Entity: N/A
Notes: This table is used for store user activities when they login

6.1.56. tblGender

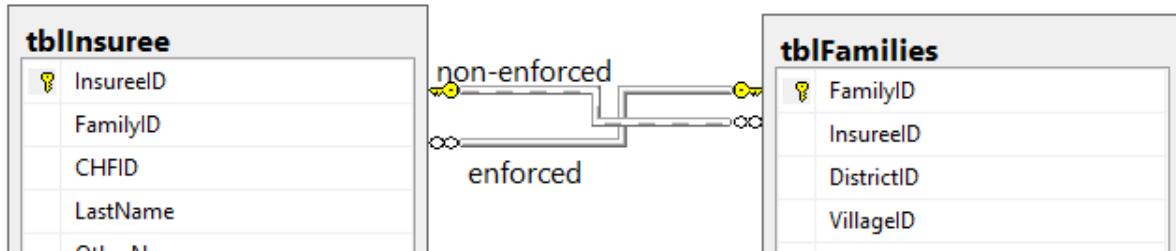
Column Name	Data Type	Allow Nulls
Code	char(1)	<input type="checkbox"/>
Gender	nvarchar(50)	<input checked="" type="checkbox"/>
AltLanguage	nvarchar(50)	<input checked="" type="checkbox"/>
SortOrder	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Object that depend on tblGender:	tblGender depends on:
----------------------------------	-----------------------

<ul style="list-style-type: none">[-] tblGender[+] tblInsuree[-] uspAPIEditFamily[-] uspAPIEditMemberFamily[-] uspAPIEnterFamily[-] uspAPIEnterMemberFamily[-] uspCleanTables[-] uspExportOffLineExtract3[-] uspImportOffLineExtract3	<p>.... tblGender</p>
Reference to Data Model Entity: N/A	
Note: This table is used to store genders for all family members	

6.2. Graphical relationship overviews

The graphical representation of relationships is covered in this paragraph in separate so-called database diagrams. We could have enforced and non-enforced relationships as shown below:



Each diagram contains a limited amount of tables to make the diagrams more readable. Some tables might be included in several diagrams for clarity of the model.

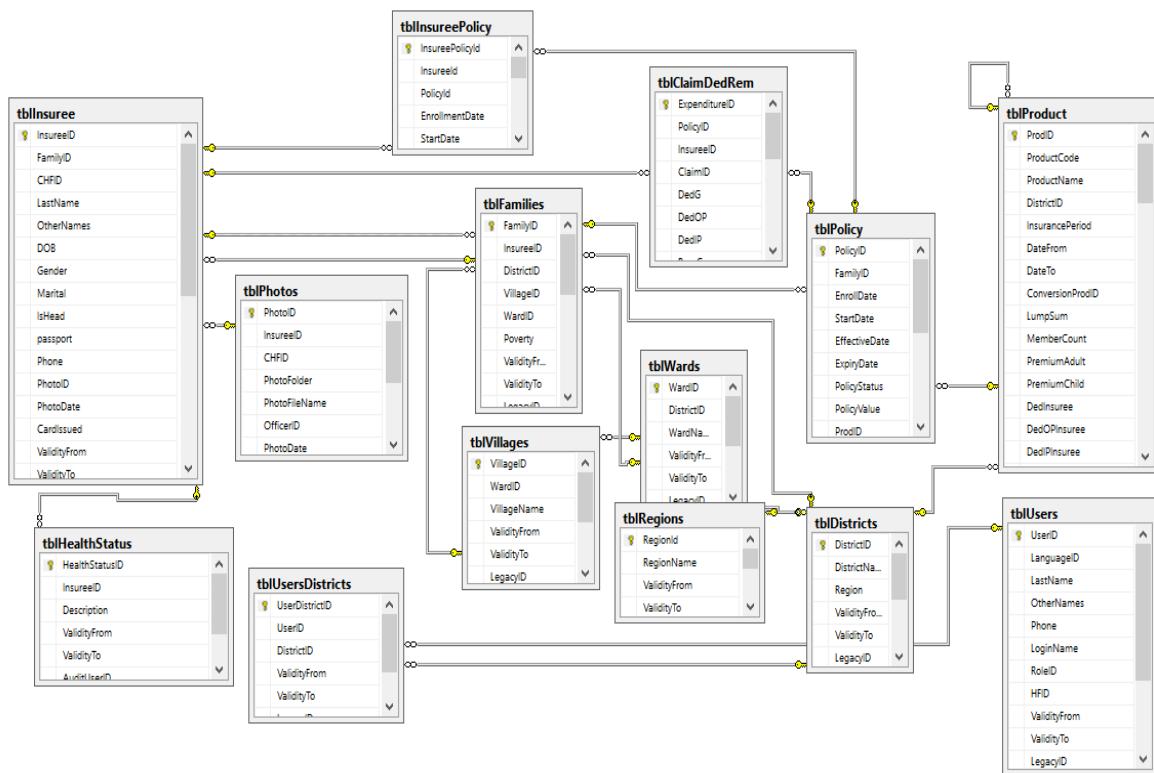


Diagram 1: Family-Districts-Villages-Municipality -Users-Insuree-Policy-Product-Relation-Language Professions

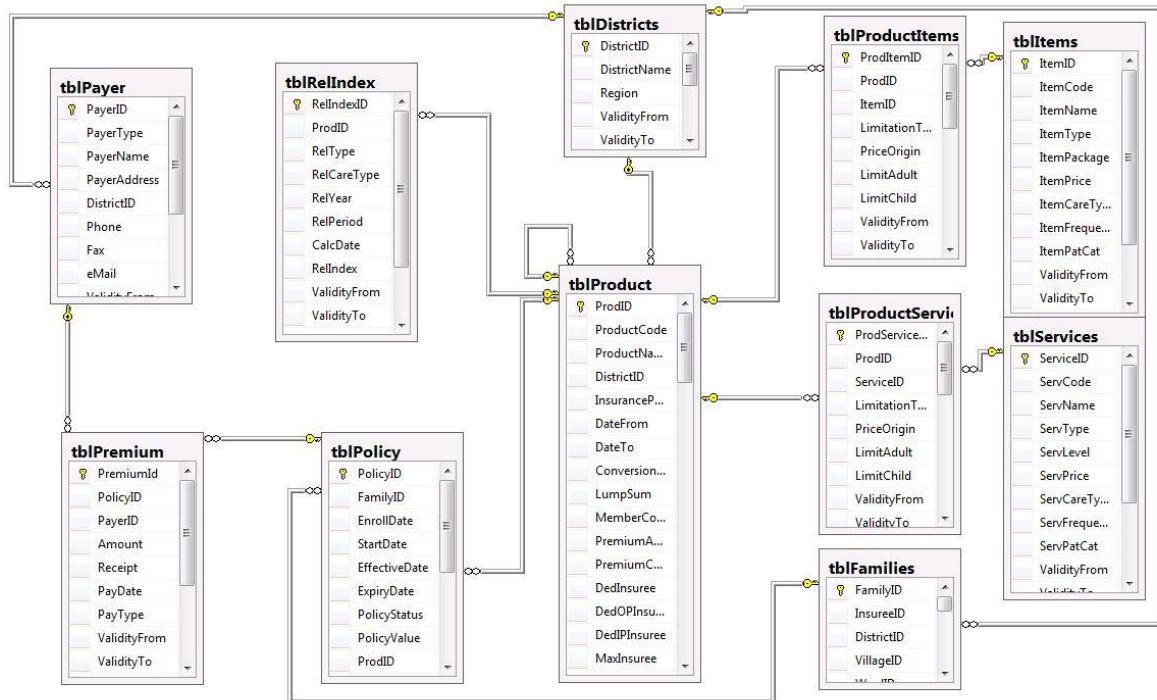


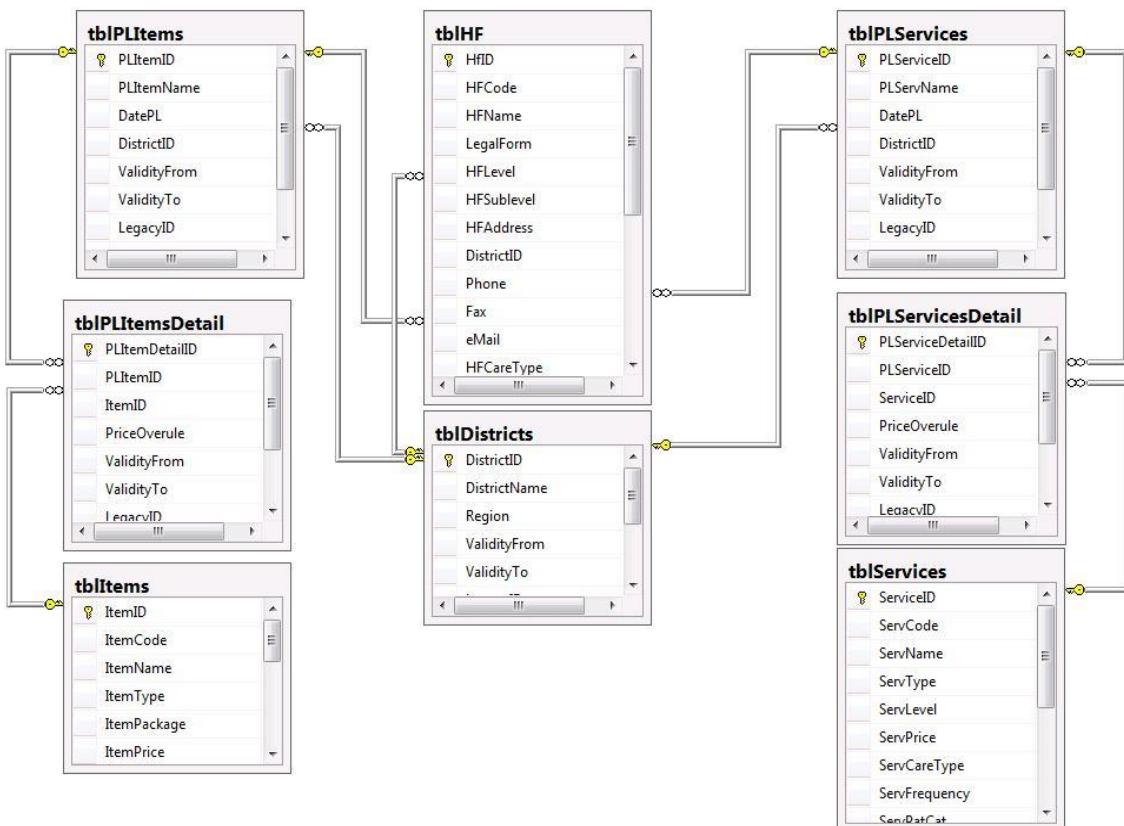
Diagram 2: Payer-Contributions-Policy-Product-Family

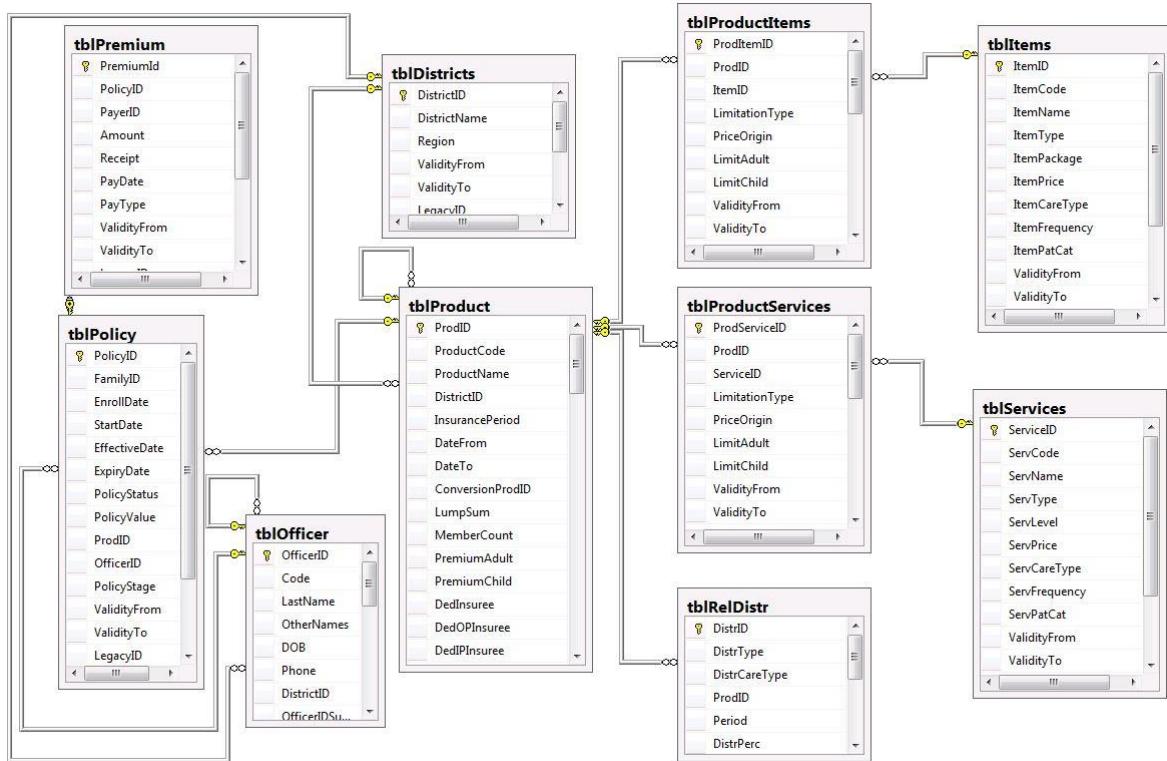
Diagram 3: HF-Pricelists

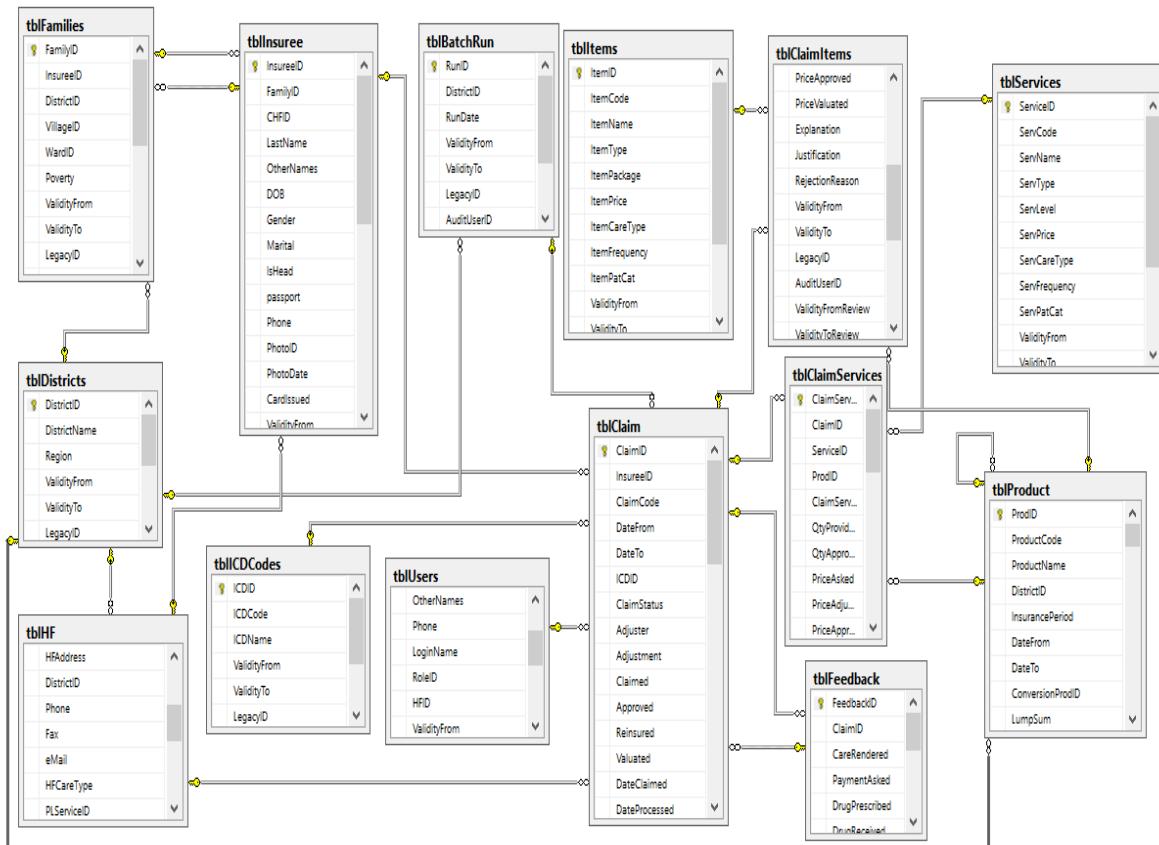
Diagram 4: Policy-Product-Contributions

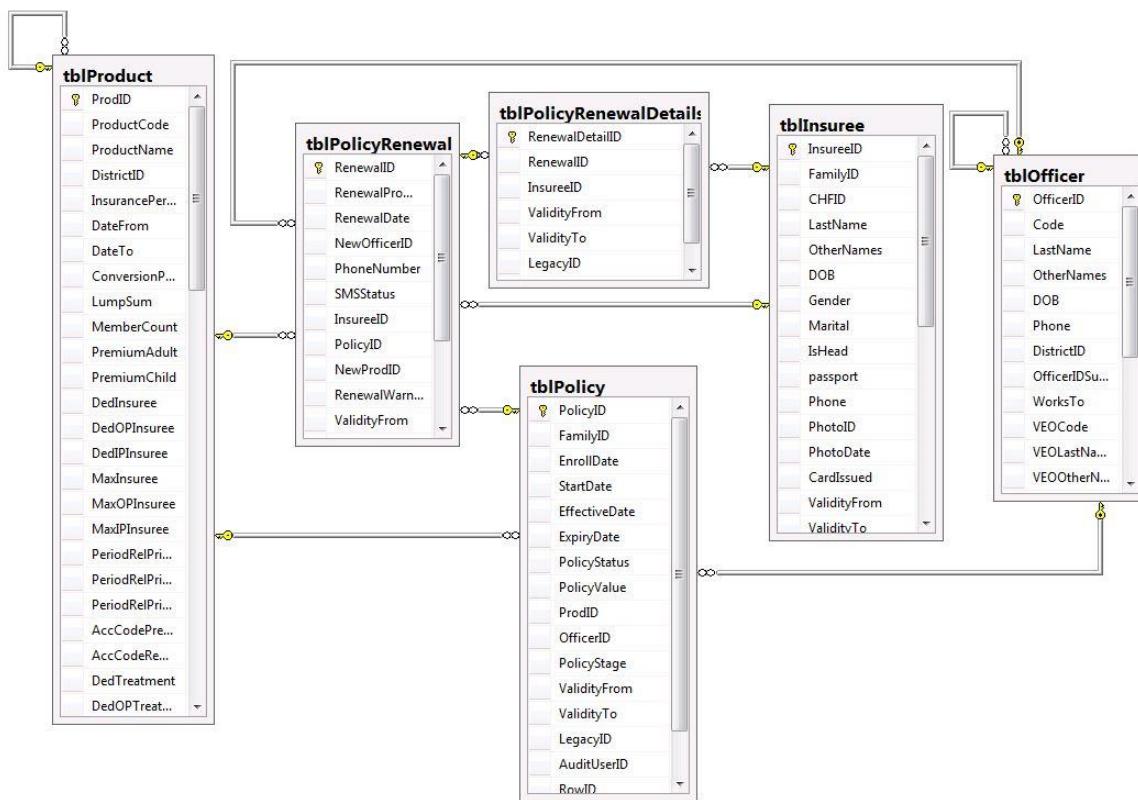
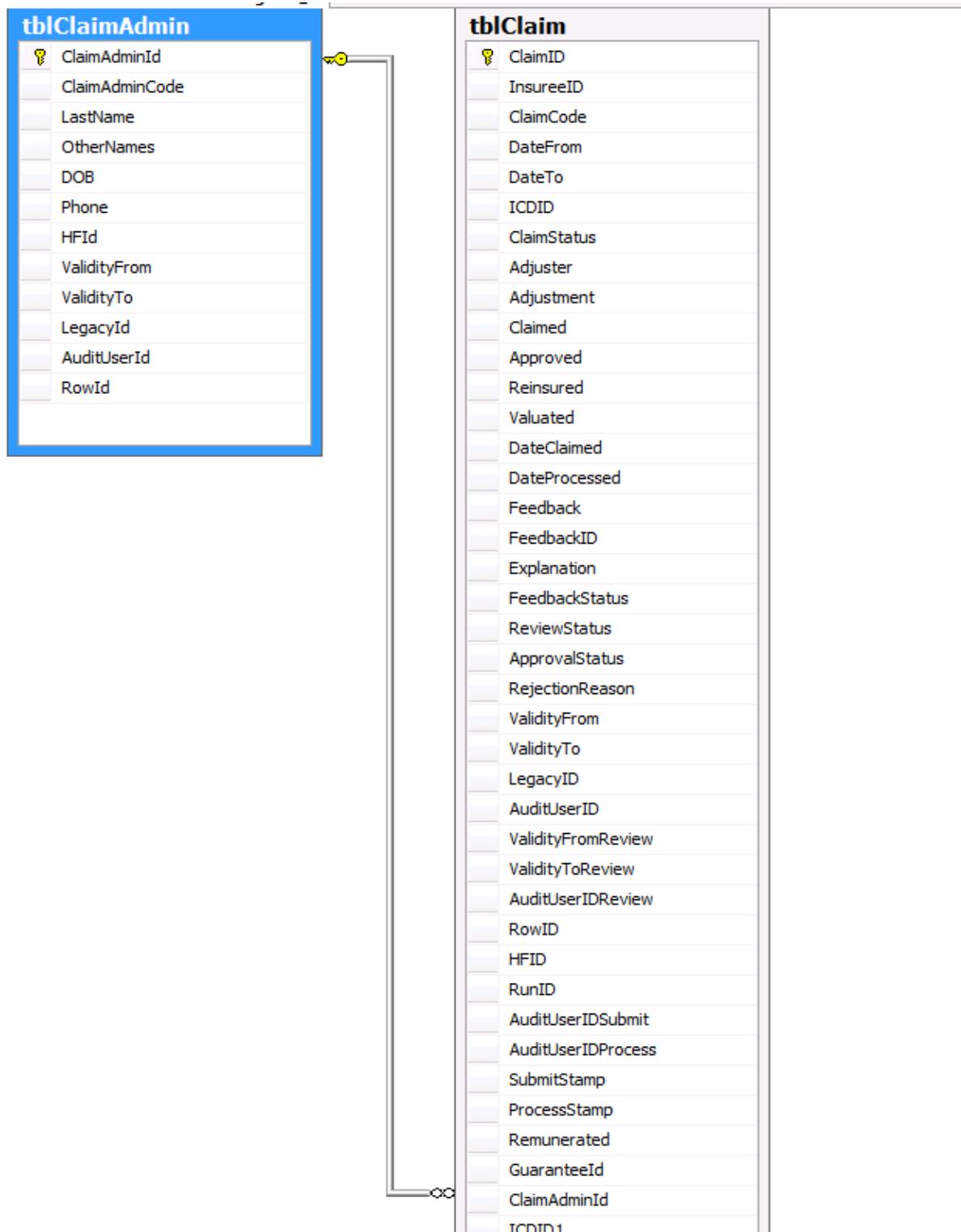
Diagram 5: HF-(Batch)-Claims-Products-family-Insuree

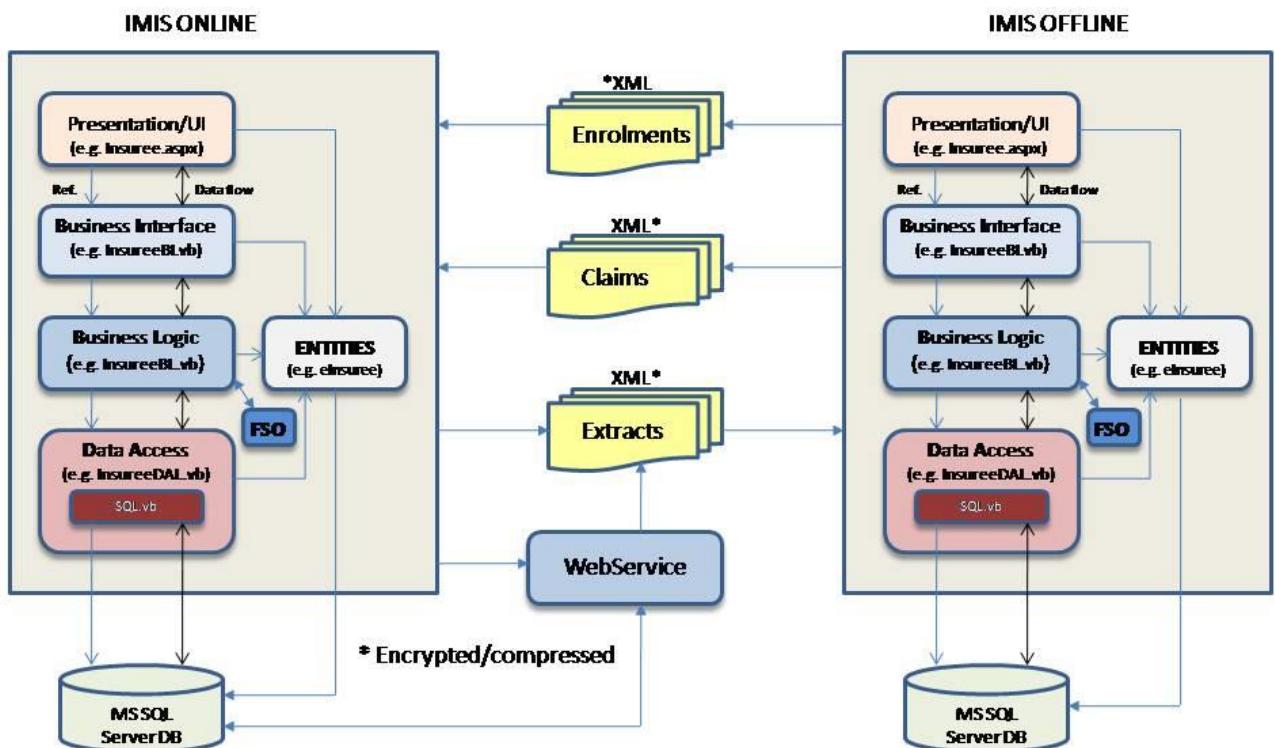
Diagram 6: Policies-Renewals-Officers**Diagram 7:** Claim Administrator

6.3. IMIS On-line and Off-line

The online and offline client will use exactly the same 'web-application' but differ on few areas:

- Off line client has an additional database for keeping claim objects separate
- Off line client will have an additional feature for creating XML batches

The graphic below illustrate how the online /offline systems are interconnected



The files are exchanged between online and offline in XML format. WebService is used for passing data (containing enrolments, Insurees, policies, contribution and claims) from online to offline and vice versa. These files are properly encrypted and zipped for security purposes

The off-line client will operate independent from the Central database and will connect to an instance of a locally (LAN) installed SQL Server. The SQL Server will host 2 separate databases. These databases will include in their names 'XXX_IMIS', where XXXX stands for the unique code for the facility/hospital (TBD).

- 'Copy' of the IMIS: XXX_IMIS
- XXX_IMIS

The 'Copy' of Central will be restored on periodical basis with a special developed tool. All databases will be protected by password security for restore actions (so called SQL Server backup password shipped with the physical backup).

Automatically after restoration, the data not needed for the off-line client, will be flushed from the system. Depending on the rules to be set for the availability of data for the off-line client and security constraints, data will be (partially) flushed from the IMIS database. Data flushing parameters might be for example DistrictID and HFID.

At this stage (in the absence of the design of the Claim management module) we anticipate flushing the following tables from the IMIS. These tables will be hosted in the additional offline database 'XXX_IMIS'

- tblBatch
- tblClaim
- tblClaimItems
- tblClaimServices
- tblFeedback
- tblPolicyRenewals
- tblPolicyRenewalsDetails

The following tables might be candidates for manipulation/partial deletion:

- tblUsers
- tblUsersDistricts
- tblHF
- tblPLItems
- tblPLItemsDetail
- tblPLServices
- tblPLServicesDetail
- tblFamilies
- tblInsuree
- tblHealthStatus
- tblPolicy
- tblPhotos
- tblPremium
- tblPayer
- tblOfficer

The following tables are anticipated to remain as a full copy in the XXXX_IMIS:

- tblDistricts
- tblVillages
- tblWards
- tblItems
- tblServices
- tblProduct
- tblProductItems
- tblProductServices
- tbICDCodes

The OFFLINE_IMIS database will host the following tables:

- tblBatch
- tblClaim
- tblClaimItems

- tblClaimServices
- tblFeedback

These tables will have exactly the same structure as the tables hosted in the IMIS database. The offline application will utilize all other information from the IMIS database for referencing.

- used for creating .db3 Database, used by mobile applications.