System Architecture

Chemical Registry

2020

Table of Contents

1 Abbreviation and Acronyms

UMM	User Management Module
CDM	Common Data Module
SB	Service Bus / Integration Module
NCDC	National Center of Disease Control

1. System architecture Document Purpose

System Architecture document for Chemical Registry describes technical architecture of the system components, specification of their interrelations and used technologies. Target group of the document are system administrator and developers, but can contain relevant information for the other parties who has access to the system and/or the document. The document contains all the needed and necessary information for the system administration, support and future development.

Current document does not cover the detailed manual for the administration interface. For this purpose, an important document is Chemical Registry user Manual for users and Authorities.

2. User Management Module

2.1.1 General Information about UM Module

User Management module is part of Chemical Registry, purpose of which is to manage users and access rights for the central module (portal). More specifically, it defines users, right and access levels, which also supports that by validation mechanism for prevention of unauthorized access to the central module of the system. UM Module contains unified directory of the users, defined roles, access rights for the roles and limitations. Simplified (one-step) authorization is realized in this module and consists of the interfaces that are enabled after single authorization, without need in re-authorization after changing the interfaces between various modules. The access rights are unified across the modules.

2.1.1 Functions of UM Module

The enterprise functions of the module are legalized by the following business processes:

- Management of and control of the unified user directory. This directory represents the list of the persons and/or organizations who have access rights for the central module (portal).
- When needed, based on the current or future necessities of the central module, possibility of creation of roles and user groups. The task of role creation encompasses strict definition of the resources for the role. The resource is later can be managed (created, edited or deleted) by the defined user based on the specific needs.
- Access right definition for the user groups according to the resources. Creation of the additional attributes and specific values for the attributes.
- Registration of various kinds of informative messages for the Chemical Registry various modules.

3. Service Bus

Data transport between Chemical Registry system is realized using internal Service Bus. Part the system is communication module (HMIS.messaging.web), using of which all the module of Chemical Registry sends messages and requests to the other modules, not with direct connection. Every sent message should contain the following data:

- Data of that method which have to be executed in the destination module
    - Format: module_name.class_name.method_name

- Method attributes
- Serialized mthod attributes *(using BinaryFormatter)*

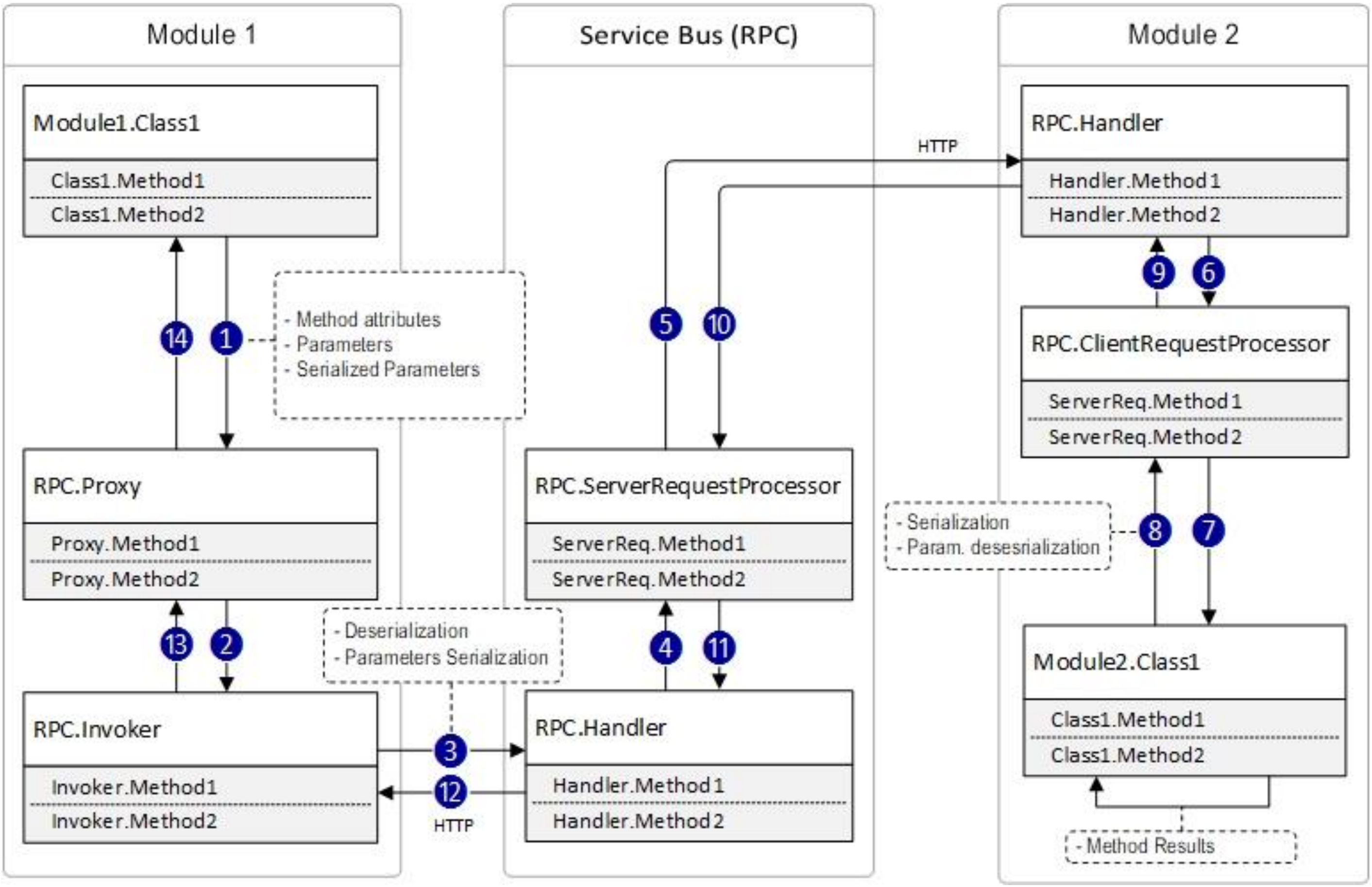

Figure 1. Process-Concept diagram of Service Bus communication loop

Service Bus reads module name provided with the request and send the request to the corresponding module. When the target module receives request from Service Bus, it tries to find the class and that method in the class, which was specified in the message and farther fries to execute it. After that the results of the execution is returned to the requesting module. For the data transfer details please refer to the **Figure 1.** Process-Concept diagram of Service Bus communication loop

.

For this purpose, to exchange data between modules so called proxy-classes are used. These classes are responsible form exchanging requests and results between modules. The messages, which are exchanged contain binary data, serialized using *BinaryFormatter.* No other standard (eg.: SOAP) is used in this process.

3.1 Common Data Module

Common Data Module (CDM, HMIS.commondata.web) is a combination of universally used electronic services, which contains types of services and data, that are common throughout Chemical Registry modules. Such data can be various directories: phone index, location names, texts used for Multilanguage support and others.

Common Data Module has interface, using of which is possible to translate the pages into various languages and add and modify them in the database. In case of need, CDM enables logging of accesses to all the Chemical Registry modules (Which use accessed which page and how much time took server to return the reply. In CDM specific interface exists where it is possible to overview the information above.

4. Platform and used Technologies

For all the System modules architecture types and technologies are the same and are based on the following platforms and technologies:

Microsoft .NET Framework 4.5.1 and .NET Core 2.2 [1]

- For Chemical Registry, Microsoft .NET represents software development environment, which contains large libraries of the classes (Framework Class Library - FCL) and enables interoperability. The code, written for .NET Framework, is executed in software environment (different from hardware environment) in Common Language Runtime (CLR), which is virtual machine for the application and supports such services as security, memory management and exceptions processing. FCL together with CLR represents Microsoft .NET Framework. Chemical Registry is built on one of the latest stable version - .NET Core 2.2 and .NET Framework 4.5.1

CITI.EVO.TwoWayModelBinding – MVC[2] analogue for ASP.NET Web Forms

- Web-application environment that is based on Model–view–controller methodology.

NHibernate [3]

- NHibernate is an object-relation connector (Object-Relational Mapper - ORM), which means to “translate” Chemical Registry data from .NET environment to almost any relational database format (Oracle DB, IBM DB2, mysql, postgre SQL and etc.). As a result, Chemical Registry system is not depending on the specific type of database and any relational database can be used in the solution.

Microsoft SQL [4]

[1] https://www.microsoft.com/en-us/download/details.aspx?id=40779
[2] https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx
[3] http://nhibernate.info
[4] https://www.microsoft.com/en-us/sql-server/sql-server-2017

- Microsoft SQL acts as a storage for Chemical Registry metadata, that means the information about data structure, but not the data itself.

5. User Management Module

User Management module is realized using web technologies. The dialed information is available in Chapter 5 - General Information about UM Module.

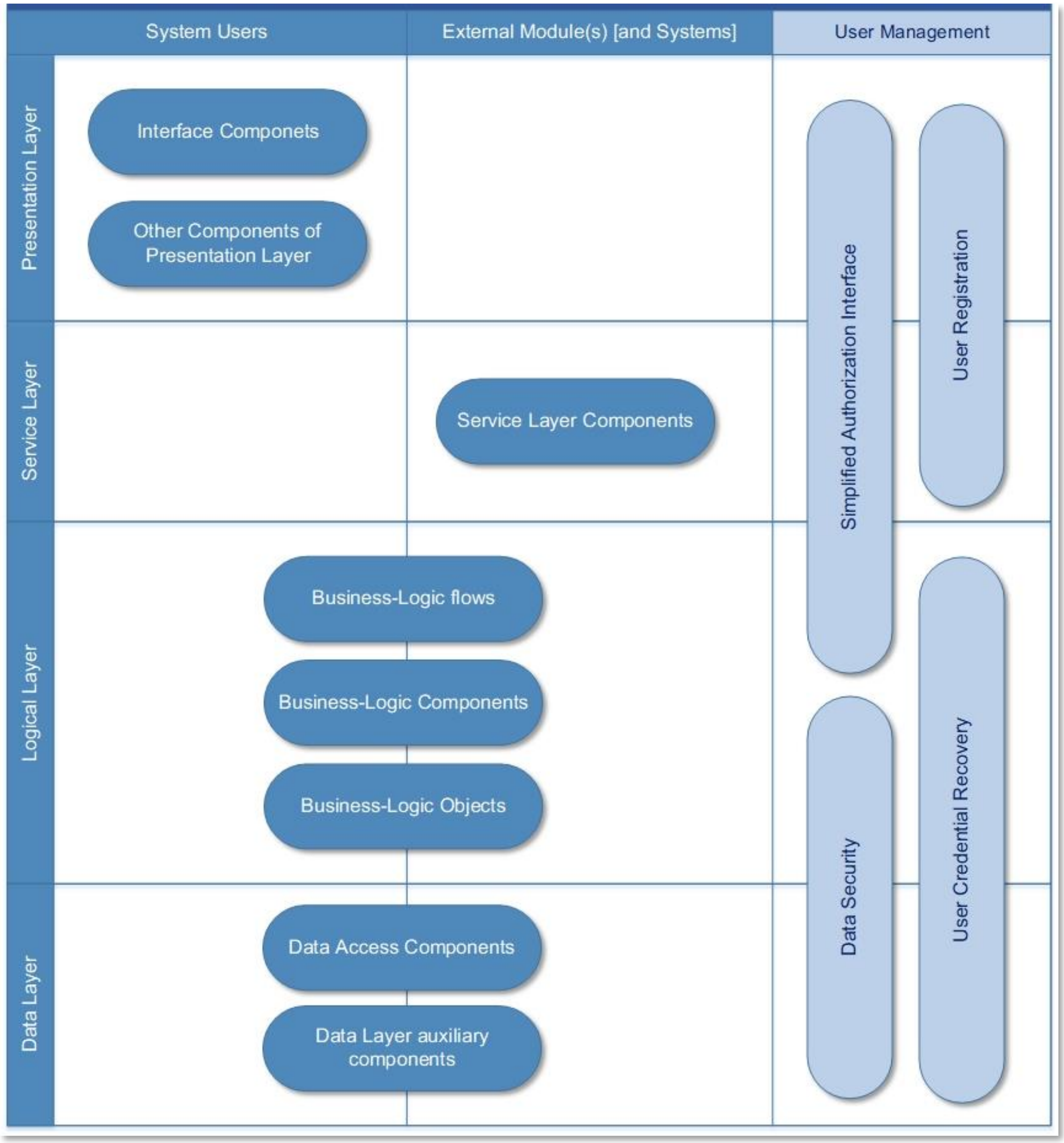

Figure 2. Various layers of User Management Module

5.1.1 General Architecture

User Management Module general architecture encompasses the following layers:

- Presentation Layer
- Business Logic Layer
- Service Layer
- Data Layer

Each of the layer has specific role and function under User management module unified architecture. Same time, these layers are tightly connected to each other are actively exchange necessary information streams (see **Figure 2.** Various layers of User Management Module

).

5.1.1.1 Presentation Layer

Presentation layer is represented by the collection of those interfaces and working web-forms, using of which enables administrators to work in the system. This layer is such an interface, via which realized scenarios on the business level are executed and being operated on.

5.1.1.2 Business-Logic Layer

Business-Logic Level is an important component in the general architecture of UM Module. All those scenarios, validations and control mechanisms are collected here, that defines module characteristics and purpose. This layer is connected with the other layers under general architecture. It is business-logic layer, through which presentation and service layers are connected to the database layer.

The list of validations is realized in User Management Module, that are necessary for the system to fulfil the designed aims, to gather the correct information about users and distribute it to the other parts of the system. Due to the characteristics of the system, validations enable functions of UM module, depending Central module (Portal) and all the other modules in future. Below there is a list of the validations (with definitions) which are critically important for valid operation of UM Module:

- **Validation of necessary fields** - For example it is necessary for the user to have username and password
- **Prerequisite for a group creation** – user must necessarily be a member of a desired module (at this stage only Portal is necessary) (administrator and/or regular user).
- **Duplicity validation** – Validation on the duplicity according to username.
- **Various additional validations** – Due to the specific characteristics of the interfaces, there are validations which support seamless operation of the module (for instance, which execution of the specific commands, there are necessary attributes which must be defined in the method). Also validations based on the field format, so that correct format value is sent to the needed service.

Figure 3. User Management Administration - Business Process shows validations and logical information flows between the modules, according to the specific tasks.

5.1.1.3 Service Layer

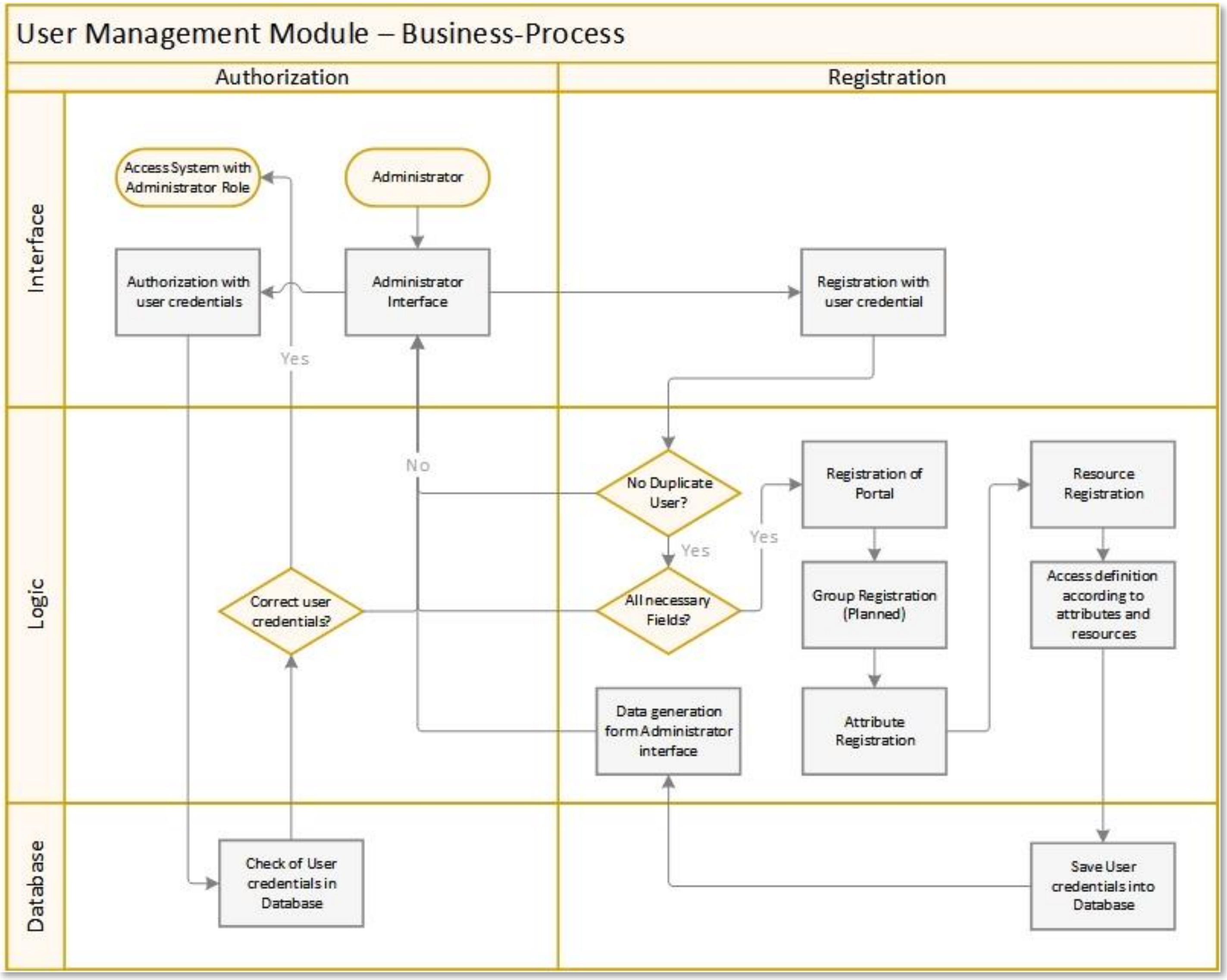
User Management Module – Business-Process
Authorization
Registration
Interface
Logic
Database
Access System with Administrator Role
Administrator
Authorization with user credentials
Administrator Interface
Registration with user credential
Yes
No
No Duplicate User?
Yes
Yes
Registration of Portal
Resource Registration
Correct user credentials?
All necessary Fields?
Group Registration (Planned)
Access definition according to attributes and resources
Data generation form Administrator interface
Attribute Registration
Check of User credentials in Database
Save User credentials into Database

Figure 3. User Management Administration - Business Process

Service layer is one of the most important components of the UM module, however on this stage the usage of its full functions is limiter, because the system is comprised by only two modules, therefore the need in information exchange between them is low. The main purpose of this layer is to organize communication between the system modules and external environment (in case if corresponding access is there). All external and internal modules exchange information using this service layer.

5.1.1.4 Data Layer (**DB**)

The main purpose of Data layer is to collect and organize data from the other layers of UM module in an optimal way, which enables the module to fulfil the tasks: fast search, indexing, data generation and data output. Performance of this layer is one of the most critical and defines overall performance of the module. No business-process is realized on this layer not to duplicate scenarios with business-logic layer.

5.1.2 Informational Streams

With help of various methods and channels, User Management Module receives and sends various information. Therefore, these information streams are divided into two groups "input/received information" and "output information". In addition, the methods of receiving or sending, also methods of saving the data into the database differ. The figure below (**Figure 4.** User Management Module - Information Streams) shows graphical diagram of information streams, where green color defines input/received information streams and red color – output information streams.

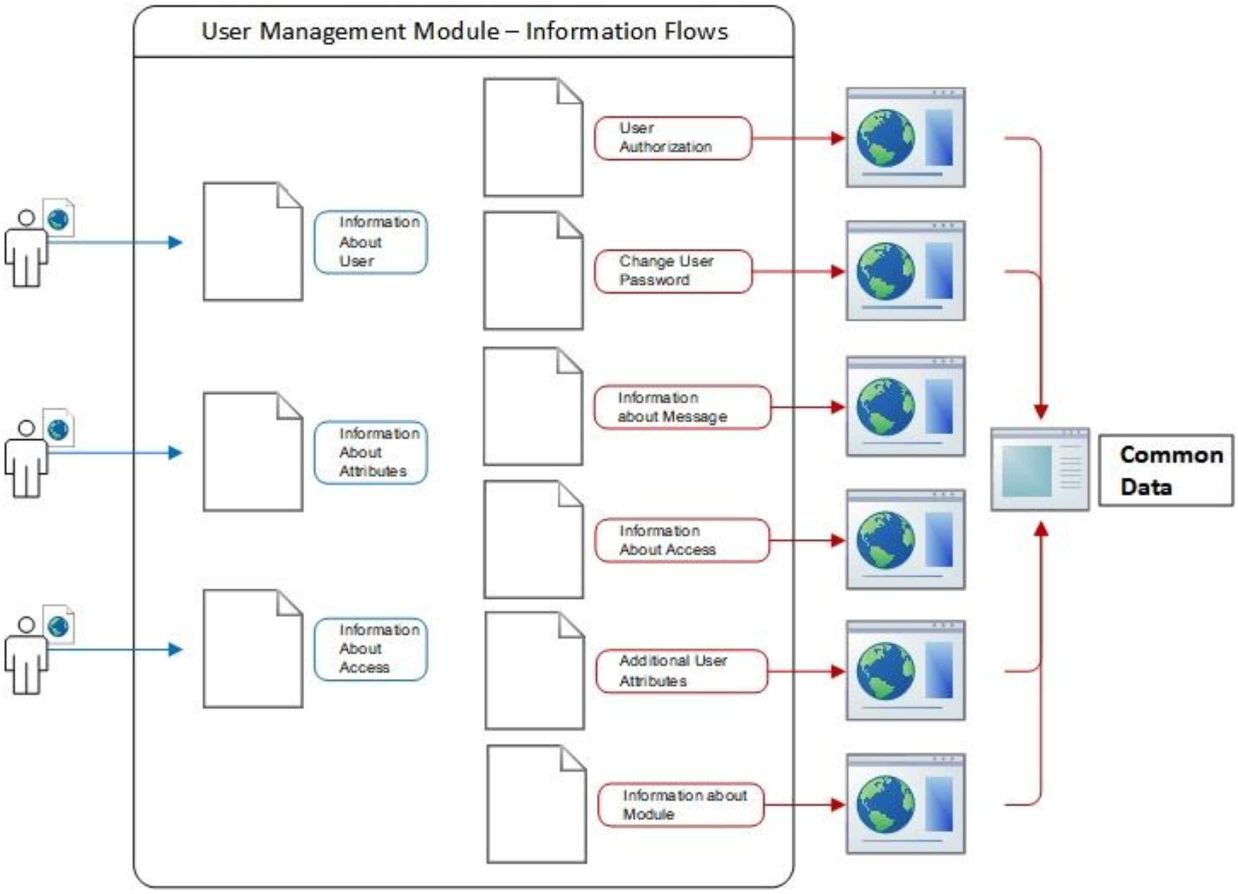

Figure 4. User Management Module - Information Streams

Table 1 and **Table 2** show the details regarding "input/received information" and "output information".

Table 1. Input/Received information streams

Information Stream	Source	Input Form	Period	Stored in Database	Address	Method
Information about user	Module user	Interface	Per request	Yes		N/A

Information Stream	Source	Input Form	Period	Stored in Database	Address	Method
Information about changed password	User Management Module (UMM)	Web-service	Per request and/or after password expiration period		[5]	[6]
Information about attribute schema, attribute schema node and attribute values	User Management Module (UMM)	Interface	Per request			N/A
Information about resources and the access rights for the resources	User Management Module (UMM)	Interface	Per request			N/A

Table 2. Sent information streams

Information Stream	Destination	Input Form	Period	Address	Method
Information about all the users, according to the authorized users	Portal	Web-service	Per request	[7]	[8]
Information about current user					[9]
Information about active/passive status of the authorized user token					[10]
Authorization					[11]
Information about the members of user groups					[12]
Information about user groups					[13]
Information about group attributes (additional information)					[14]

[5] http://chemicals.ncdc.ge/Chemical/Registry/Web/
[6] PasswordChangeResultEnum ChangePassword(Guid token, String newPassword, String oldPassword)
[7] http://chemicals.ncdc.ge/Chemical/UserManagement/
[8] List<UserContract> GetAllUsers(Guid token)
[9] public UserContract GetCurrentUser(Guid token)
[10] bool IsTokenActual(Guid token)
[11] Guid? Login(String loginName, String password, bool encryptedPassword)
[12] List<UserContract> GetGroupUsers(Guid token, Guid groupID)
[13] List<GroupContract> GetUserGroups(Guid token, Guid userID, Guid projectID)
[14] List<GroupAttributeContract> GetGroupAttributes(Guid token, Guid groupID)

Information Stream	Destination	Input Form	Period	Address	Method
Information about user attributes (additional information)					15

5.1.3 Security

Non authorized access and information security to the User Management Module is controlled dynamically by current module. It also supports single authorization functionality using corresponding credentials (username and password), which means getting possibility to access the central and other modules (for future development), using access rights and periods, separately for each of the modules. The competences of User Management Module also contain the possibility to define and control on the usage of its resources by the users and corresponding access rights.

5.2 Common Data Module

The architecture of Common Data Module (CDM) is the same as for User Management Module and encompasses the unity of following layers:

- Presentation Layer
- Business Logic Layer
- Service Layer
- Data Layer

For more details, refer to the Chapter 5.1.1.

5.2.1 Common Data Module – Information streams

Common Data Module is a shared data storage of information and services under Chemical Registry system solution. The information streams, methods of the information exchange and the corresponding sources are gathered into the following two categories:

- Chemical Registry Modules interface components' names in Various languages;
- User The operations list (Logs) under Chemical Registry system;

Module also receives and sends various data using various methods. Therefore, these streams are divided into two groups – "received information" and "Sent information". The methods of information stream input/output, also the methods of storing the data into database are different. The list of information streams, with their direction, input/output methods and periods are given below:

Table 3. Received information

Information Stream	Source	Input Form	Period	Stored in Database	Address
Description of the interface components of	Interface	Interface	Solely for a single word	Yes	16

[15] List<UserAttributeContract> GetUserAttributes(Guid token, Guid userID, Guid projectID)

[16] http://chemicals.ncdc.ge/Chemical/CommonData/

the CHEMICAL REGISTRY modules in various languages					
SetTranslatedText: adds desired translation to the specific control	CHEMICAL REGISTRY Modules	Web-service	Solely for a single Control		

Table 4. Sent information

Information Stream	Source	Input Form	Period	Address
GetLanguages: Returns the list of the language into which CHEMICAL REGISTRY modules are translated	CHEMICAL REGISTRY Modules	Web-service	Per request	17
GetTranslatedText: Returns the translation of the interface control into the desired language	CHEMICAL REGISTRY Modules	Web-service	Per request	

6. Database Structure and Definitions

Data Storage, processing and output tasks in Chemical Registry databases are decentralized into the two various database systems. This solution makes the tasks related to the data more dynamic and so solve them, in various cases two different databased are used per task – the database is chosen according to the characteristics of the task and when using one specific database is more effective and efficient.

- Microsoft SQL[18] : See the Chapters **Error! Reference source not found.** and **Error! Reference source not found.**.
- MongoDB[19]

6.1 User Management Module Database

User Management Module Database is realized as a centralized database for a single module, which is located on the database server (For the address and hardware details see Chapter **Error! Reference source not found.**). Logging functionality is partly dealized for those data, data historical analysis of which might be needed in the future (for this purpose the tables contains the following columns set: DateCreated, DateChanged, DateDeleted).

Logging system for modules performance monitoring is also realized in the solutions. Its interface enables accounting and monitoring of the time spent by used on the specific interface operations.

[17] http://chemicals.ncdc.ge/Chemical/CommonData/

[18] https://www.microsoft.com/en-us/sql-server/sql-server-2017

[19] https://www.mongodb.com

6.1.1 Database Structure

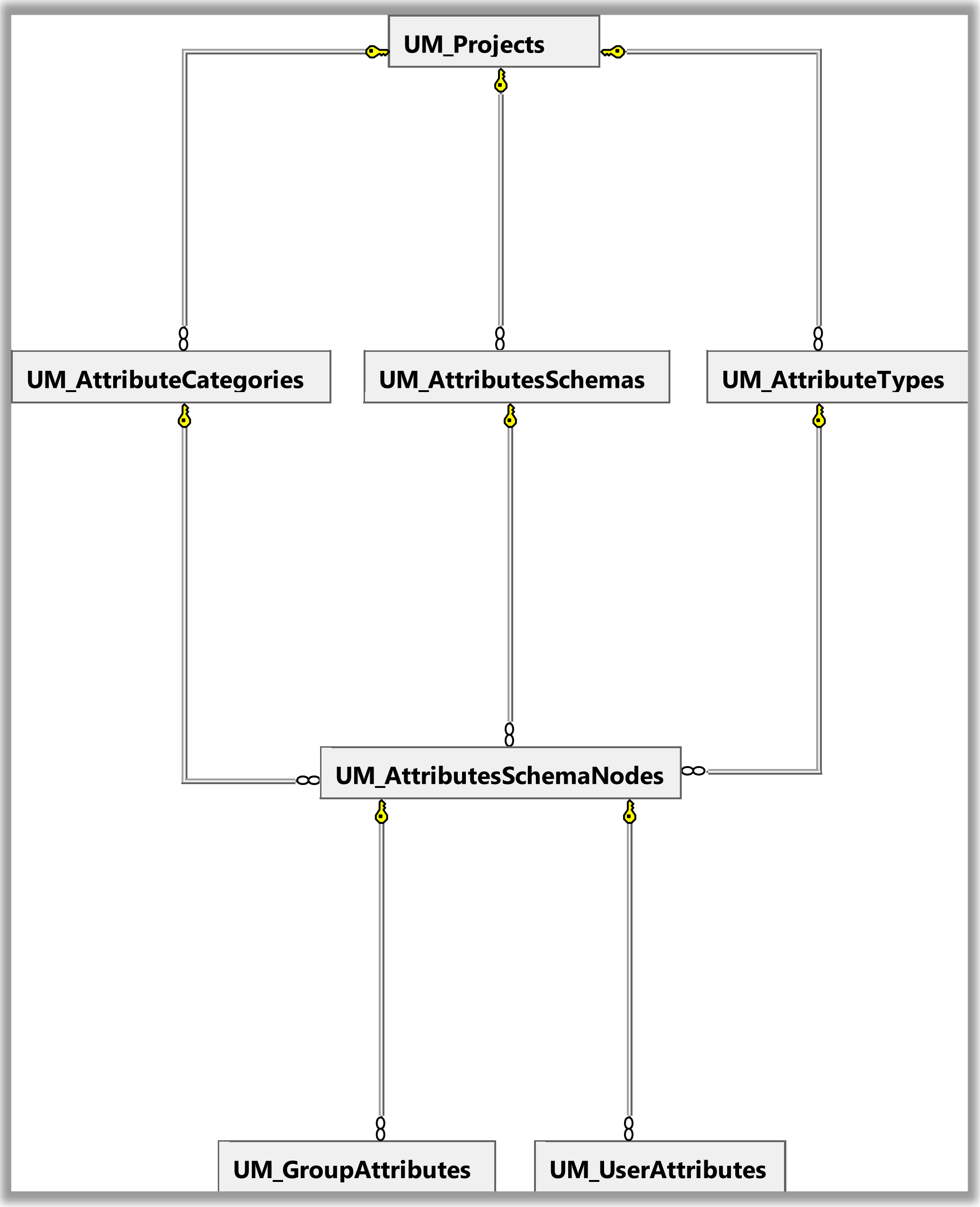

Figure 53. Attributes schema

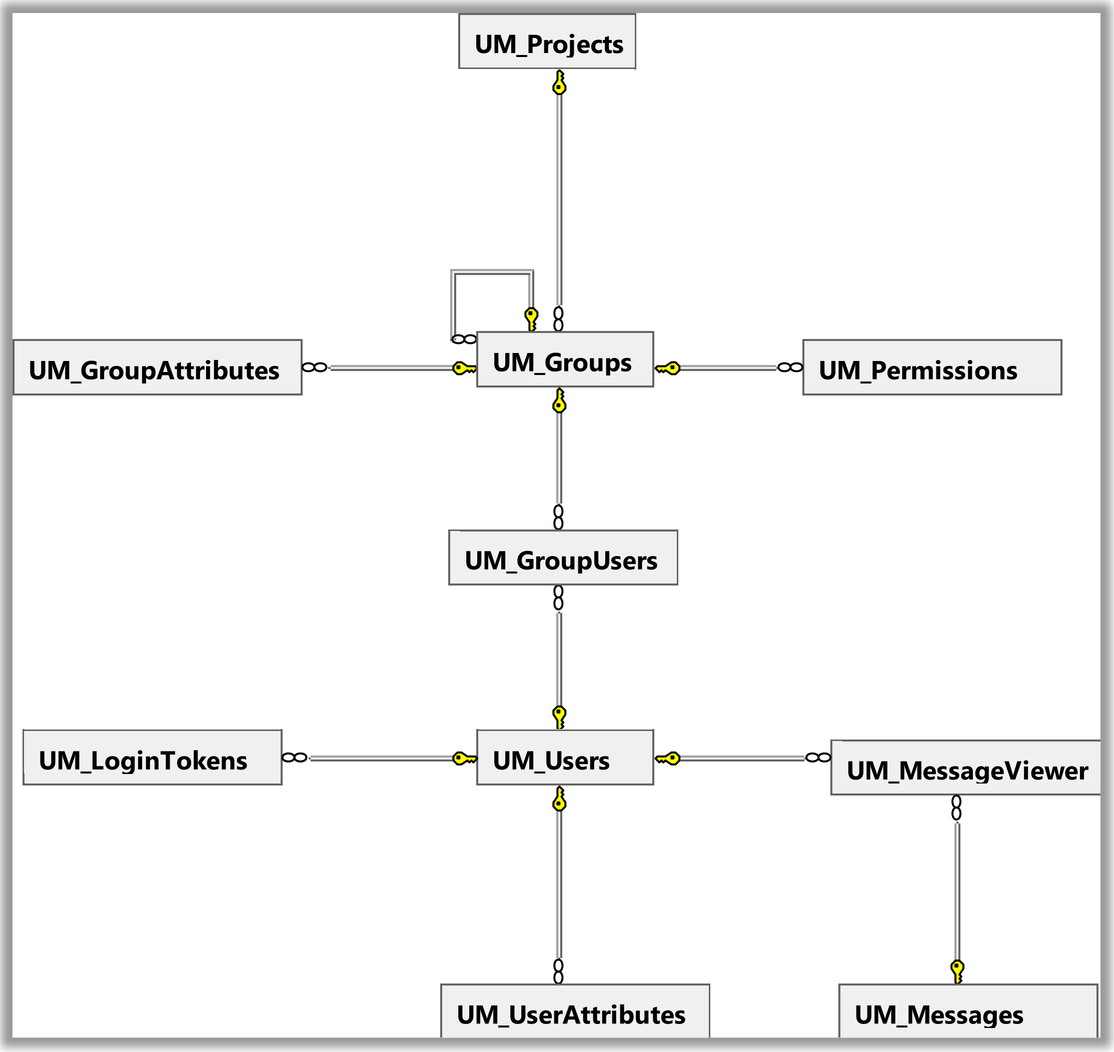

Figure 64. User rights and attribute definitions

6.1.2 User Management Module Database Tables and Columns

Table 5. User Management Module database tables and fields

Table name	Table Description	Columns
UM_AttributesSchemaNodes	According to category and attribute types attribute schema nodes are created. For example “Provider ID”, which itself is included in attribute schema	ID Name AttributeCategoryID AttributeTypeID AttributesSchemaID DateCreated DateChanged DateDeleted Hashcode
		ID

Table name	Table Description	Columns
UM_AttributesSchemas	Attributes schema defined at the module level, which acts as attribute schema node grouper	ProjectID Name DateCreated DateChanged DateDeleted Hashcode
UM_GroupAttributes	Additional information definition for the groups, which is created according to attribute schemas, attribute categories, attribute types and attribute schema nodes	ID GroupID AttributesSchemaNodeID Value DateCreated DateChanged DateDeleted Hashcode
UM_Groups	Groups defined at the module level	ID ParentID ProjectID Name DateCreated DateChanged DateDeleted Hashcode
UM_GroupUsers	Table for users and user groups relation	ID UserID GroupID AccessLevel DateCreated DateChanged DateDeleted Hashcode
UM_LoginTokens	Table for tokens created as a result of authorization in various modules	ID LoginToken UserID ExpireDate LastAccessDate DateCreated DateChanged DateDeleted Hashcode DeleteReason

Table name	Table Description	Columns
UM_Messages	Table of messages at the module, group and user level	ID Subject Text ObjectID Type DateCreated DateChanged DateDeleted
UM_MessageViewer	Table of read/unread messages	ID MessageID UserID DateCreated DateChanged DateDeleted
UM_Permissions	Table for the permissions on View, Add, Edit, Delete according to the groups and the resources of the module	ID GroupID ResourceID RuleValue DateCreated DateChanged DateDeleted Hashcode
UM_Projects	Modules table	ID Name DateCreated DateChanged DateDeleted IsActive Hashcode
UM_Resources	Table of the resources, where it is possible to store virtual addresses, as well as the addresses of the pages in various modules and components of those pages. For example: module/pages/default.aspx	ID ParentID ProjectID Name Description Type Value DateCreated DateChanged DateDeleted Hashcode

Table name	Table Description	Columns
UM_UserAttributes	Additional information definition for users, which is created according to the attribute schemas, attribute categories, attribute types and attribute schema nodes	ID
		UserID
		AttributesSchemaNodeID
		Value
		DateCreated
		DateChanged
		DateDeleted
		Hashcode
UM_Users	Users table	ID
		LoginName
		Password
		FirstName
		LastName
		Email
		Address
		OrganizationName
		Department
		Division
		Position
		Telephone
		IsSuperAdmin
		IsActive
		UserCategoryID
		PasswordExpirationDate
		DateCreated
		DateChanged
		DateDeleted
		Hashcode

6.1.3 Table Relations

Database tables are related to each other by the corresponding columns and using defined logic. Figure **7.** Attribute schemas**7** and **Figure 8**6 show diagrams for those relations and

Table 6 contains detailed information about the relations and relation types. Also, the definitions are available that describe the purpose of those relations.

Table 6. Connections between database tables and their types

Relation name	Table(1)	Table (2)	Relation Type	Description
FK_UM_Groups_UM_GroupsParent	UM_Groups	UM_Groups	One To Many	Parent-child relation between the User management Module groups
FK_UM_Groups_UM_Projects	UM_Groups	UM_Projects	One To Many	Groups relation to the modules
FK_UM_GroupUsers_UM_Groups	UM_GroupUsers	UM_Groups	Many To Many	Users relation to the user groups
FK_UM_GroupUsers_UM_Users	UM_GroupUsers	UM_Users	Many To Many	User groups relation to the users in the group
FK_UM_LoginTokens_UM_Users	UM_LoginTokens	UM_Users	One To Many	User token relation to the authorized users
FK_UM_MessageViewer_UM_MessageViewer	UM_MessageViewer	UM_Messages	One To Many	Read/unread messages relation to the messages. Is used for messages management under CHEMICAL REGISTRY modules
FK_UM_MessageViewer_UM_Users	UM_MessageViewer	UM_Users	One To Many	Read/unread messages relation to the users. Is used for messages management under CHEMICAL REGISTRY modules
FK_UM_Permissions_UM_Groups	UM_Permissions	UM_Groups	One To Many	Permissions relation to the groups. Is used for roles management under CHEMICAL REGISTRY modules
FK_UM_Permissions_UM_Resources	UM_Permissions	UM_Resources	One To Many	Permissions relation to the resources. Is used for roles management under CHEMICAL REGISTRY modules
FK_UM_Resources_UM_Projects	UM_Resources	UM_Projects	One To Many	Resources relation to the modules. Is used for roles management under CHEMICAL REGISTRY modules
FK_UM_UserAttributes_UM_AttributesSchemas	UM_UserAttributes	UM_AttributesSchemaNodes	One To Many	User attributes relation to the attribute schema nodes. Additional information about users.
FK_UM_UserAttributes_UM_Users	UM_UserAttributes	UM_Users	One To Many	Users attributes relation to the users. Additional information about users.

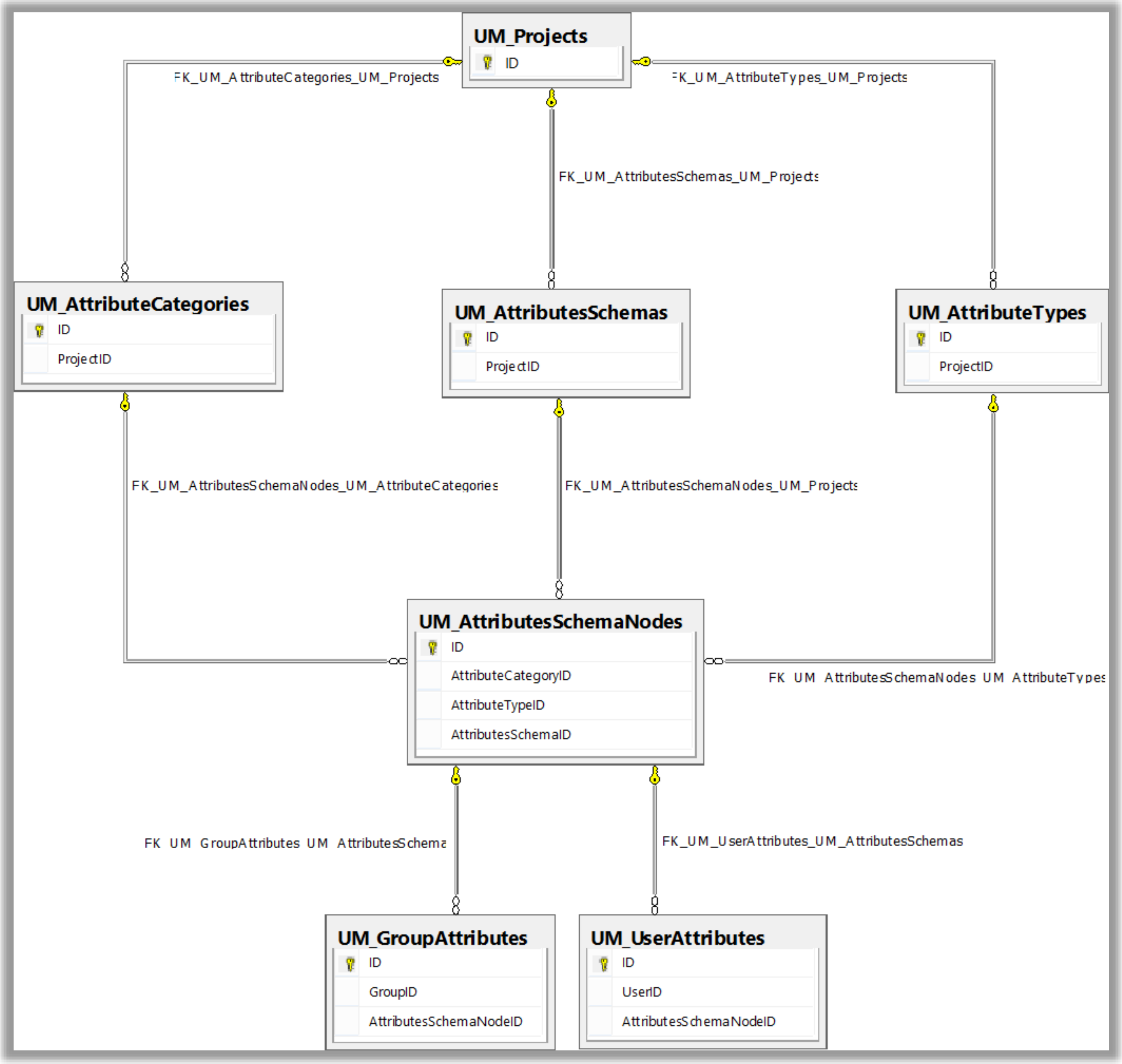
UM_Projects
ID
FK_UM_AttributeCategories_UM_Projects
FK_UM_AttributeTypes_UM_Projects
FK_UM_AttributesSchemas_UM_Projects
UM_AttributeCategories
ID
ProjectID
UM_AttributesSchemas
ID
ProjectID
UM_AttributeTypes
ID
ProjectID
FK_UM_AttributesSchemaNodes_UM_AttributeCategories
FK_UM_AttributesSchemaNodes_UM_Projects
UM_AttributesSchemaNodes
ID
AttributeCategoryID
AttributeTypeID
AttributesSchemaID
FK UM AttributesSchemaNodes UM AttributeTypes
FK UM GroupAttributes UM AttributesSchema
FK_UM_UserAttributes_UM_AttributesSchemas
UM_GroupAttributes
ID
GroupID
AttributesSchemaNodeID
UM_UserAttributes
ID
UserID
AttributesSchemaNodeID

Figure 7. Attribute schemas

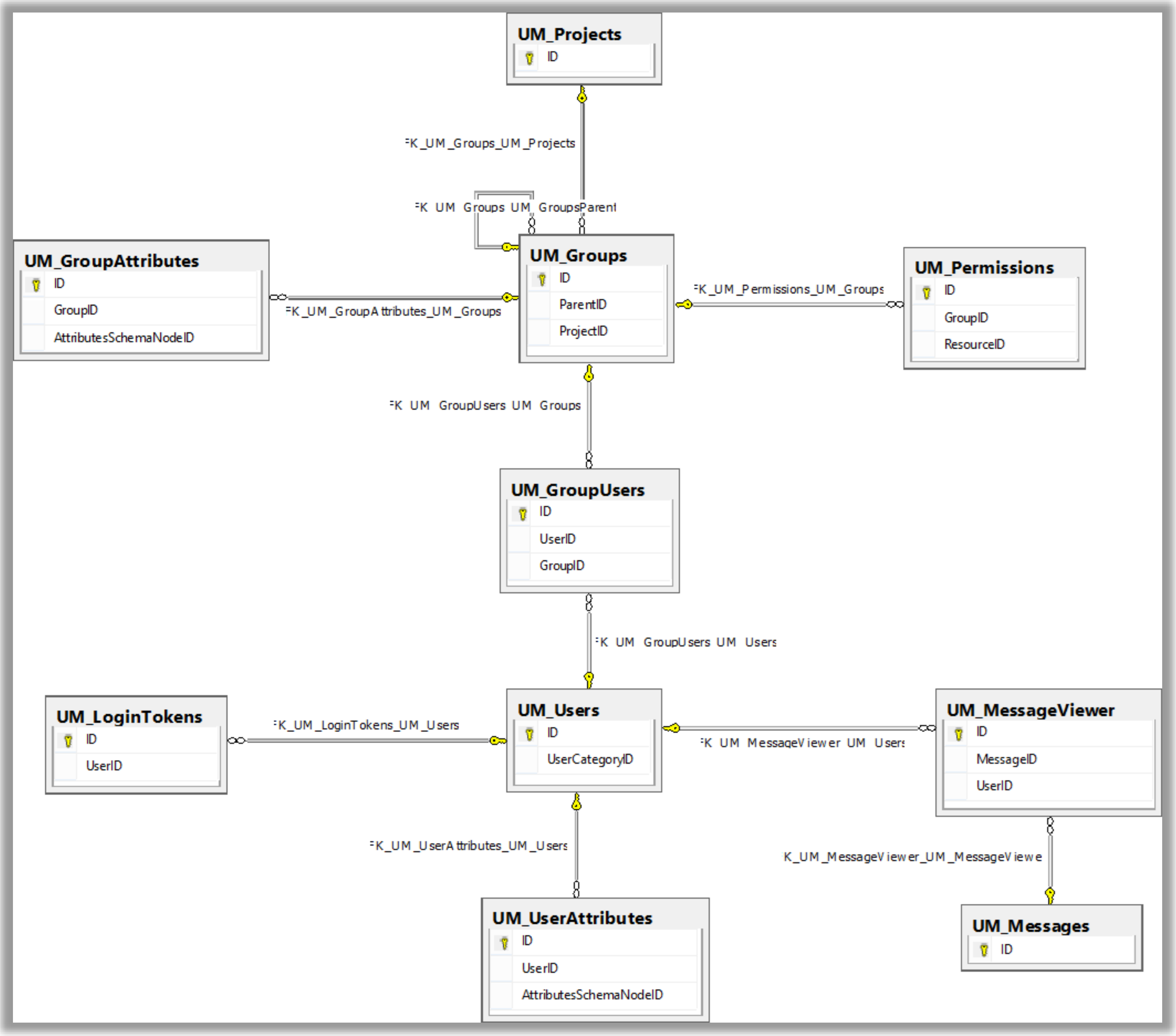

Figure 85. Users and attribute definitions

7. Necessary Libraries

To fully function, the Chemical Registry system requires additional necessary libraries. The following additional libraries are used in the system and are necessary requirements:

7.1 NPOI

This library is .NET version of POI Java Project[20]. POI is and open source project[21] using which it is possible to read and write .xls, .doc, and .ppt. In Chemical Registry system NPOI is used to import and export Excel files, without installation of Microsoft Office on server side. This solution is more effective and efficient, than running Microsoft Excel ActiveX in background mode.

7.2 ASP Chart Control

Chart controls[22,23] supports creation of ASP.NET and Windows Forms applications to create simple, intuitive and visually easily absorbable diagrams for complicated statistical and financial data analytics. Using this library, it is possible to create 35 charts and diagrams of various types.

7.3 DevExpress

DevExpress[24] is a collection of tools and libraries for software development, which using additional software plugins makes software development faster and more effective. In Chemical Registry system DevExpress ASP.NET user interface controls are used.

7.4 NHibernate

NHibernate[25] is an Object-Relational Mapper (ORM), that means translation of Chemical Registry system data from .NET environment into the format compatible with any relational Database (Microsoft SQL, Oracle DB, IBM DB2, mysql, postgre SQL and etc.). As a result, Chemical Registry system is not depending on specific database and any relational database can be used according to the specific task.

7.5 Bootstrap

Bootstrap[26] is tool for creation of user interface controls and represents front-end[27] environment for designing web-page and web-applications. Bootstrap contains HTML- and CSS-based design templates, using of which it is possible to create Chemical Registry user interfaces (forms, buttons, navigation and other components of the interface; it is also possible to create and use Javascript addons).

7.6 Angular 7.2.16

Angular 7.2.16[29] is tool for building features quickly with simple, declarative templates. Extend the template language with your own components and use a wide array of existing components. Get

[20] http://poi.apache.org
[21] https://npoi.codeplex.com
[22] https://code.msdn.microsoft.com/mschart
[23] https://msdn.microsoft.com/en-us/library/dd456632.aspx
[24] https://www.devexpress.com
[25] http://nhibernate.info
[26] http://getbootstrap.com/css
[27] https://www.wikiwand.com/en/Front-end_web_development

immediate Angular-specific help and feedback with nearly every IDE and editor. All this comes together so you can focus on building amazing apps rather than trying to make the code work.

8. Hardware Requirements

8.1 Minimum Architecture

Hardware requirements

CPU: 4x – 2000Mhz
RAM: 16GB
HDD: 200GB

Software environment requirements

Microsoft Windows Server 2008 R2[28]
Microsoft Information Services (IIS)[29]
Microsoft Visual Studio.Net 2010/2012/2013[30]
ASP.NET, .NET Core, DevExpress UI controls, Microsoft Chart controls
NHibernate-compatible relational database (e.g.: Microsoft SQL)

8.2 Recommendations andout Backup Procedure

Due to the average critical level of Chemical Registry system, at this stage implementing additional High Availability solution technology is not necessary. However, to secure system and the data from possible errors, it is recommended to use standard Backup scheme with the following periodicity:

- Full backup
  - Full reservation of system components and data
  - Period: Weekly
- Incremental Backup
  - Reservation only of those data, which were changed after the latest full backup
  - Period: Hourly

9. Installation and Configurations

For installation or backup recovery of Chemical Registry system, the platform mush be prepared according to the following order:

1. Installation and standard configuration of operation system (Microsoft Windows Server 2008/2012 R2)
2. Installation and configuration of Microsoft Internet Information Server

[28] https://technet.microsoft.com/en-us/library/dd349801(v=ws.10).aspx
[29] https://www.iis.net
[30] https://www.visualstudio.com

3. Installation and configuration of Microsoft .NET Framework and .NET Core
4. Installation and configuration of DevExpress tools and libraries
5. Installation and configuration of Database servers
    a. Creation of database and the tables (or recovery from backup package) – Microsoft SQL
6. Execute file: iis.config.cmd.