Operations expansion of the united electronic health information system

Interface Specification and Application Guide

**Electronic Recipe Information System**

NVD.VVIS.REC.PAK.PPS.SIS

Version 6.00

**2017**

Document Change History

| Date | Change description | Version | Author | Comment |
| --- | --- | --- | --- | --- |
| 28.03.2012 | Document structure draft | 0.01 | M. Trušelis |  |
| 02.05.2012 | Updated document header and footer | 0.02 | M. Trušelis |  |
| 08.06.2012 | Document draft | 0.03 | J. Džeriņš |  |
| 02.07.2012 | Document changes according to notes in document NVD.EREC.Analize.NNV.1.00 and changes in functionality of control and supervisor institution. | 0.04 | J. Džeriņš |  |
| 04.07.2012 | Approved document version | 1.00 | J. Džeriņš |  |
| 20.07.2012 | Document changes according to project design phase and HL7 standard. | 1.01 | J. Džeriņš |  |
| 06.08.2012 | Document changes according to notes in document NVD.EREC.SIS.NNV.1.00.  Fixed service “BookMedicationOrders” input parameters. | 1.02 | J. Džeriņš |  |
| 30.08.2012 | Added CAN.ALL value to table 79 (Element “retrieve” suffix values), renamed field cancellationRequest to cancelMedicationOrderRequest in table table 20 (Payload PORX\_MT000025UV01\_LV01 elements), corrected data structure CancelMedicationOrderRequest in chapter 5.8. (PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest). | 1.03 | J. Džeriņš |  |
| 04.09.2012 | Removed methods “GetDiagnosisList” and “GetMedicineList” input parameter “retrieve” in chapter 7 (Service usage explanation). | 1.04 | J. Džeriņš |  |
| 11.10.2012 | Spell error fix:  - potentiallyFullfillableInd -> potentiallyFulfillableInd;  - compensedInd -> compensatedInd;  - physicianSpecialityCode -> physicianSpecialtyCode.  Supplemented data example in chapter 7.1.3.3. Get recipe data.  Removed namespace “hl7” from XML schema examples.  Added error list in appendix.  Added chapter 5.13 HL7 standard data structures usage examples.  Changes in data structure “PORX\_MT020070UV01\_LV01. CombinedMedicationDispense” (chapter 5.7):  - EHIC.certificateId - removed (EHIC.id will be used);  - EHIC.certificateIssueTime -> EHIC.issueTime;  - EHIC.certificateEffectiveTime -> EHIC.effectiveTime.  Added missed OIDs. | 1.05 | J. Džeriņš |  |
| 22.11.2012 | Fixed errors, corrected all the XML examples. | 1.06 | J. Džeriņš |  |
| 07.12.2012 | Fixed data structures: GetCompensationConditions input (payload PORX\_MT000003UV01\_LV01), COCT\_MT050000UV01.Person, COCT\_MT230100UV.Medicine, COCT\_MT090000UV01.AssignedEntity, PORX\_MT010120UV01\_LV01.DispenseRequest, PORX\_MT020070UV01\_LV01.CombinedMedicationDispense, PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest, PORX\_MT000007UV01\_LV01.QueryByParameterPayload.  Fixed OID values, rights spelling mistakes, request and response examples.  Supplemented chapter "5.13 HL7 standard data structures usage examples" with explanation of usage of HL7 structures "name", "address" according to HL7 standard. | 1.07 | J. Džeriņš |  |
| 15.02.2013 | Fixed pharmacy OID.  Supplemented XML data examples.  GetPatientcontactList rights are fixed. | 1.08 | J. Džeriņš |  |
| 28.10.2014 | Added continuation services, which are used also in www.latvija.lv. | 1.09 | J. Džeriņš |  |
| 06.11.2014 | Changes with person address its validation and phone. | 1.10 | J. Džeriņš |  |
| 19.11.2014 | Updated error list. | 1.11 | M. Trušelis |  |
| 06.05.2015 | Document updated according to E-Health project 2nd stage requirements (Concluded framework agreements No. VMNVD 2014/3 ERAF and 27 february 2015 contract Nr. VMNVD 2014/3 ERAF 5).  Updated sections: 4. Services subsections, 5.1. PORX\_MT010120UV01\_LV01LV02.CombinedMedicationRequest, 5.2. COCT\_MT050000UV01.Person, 5.3. COCT\_MT230100UV.Medicine, 5.4. COCT\_MT090000UV01.AssignedEntity, 5.5. PORX\_MT010120UV01\_LV01LV02.SubstanceAdministrationRequest, 5.6. PORX\_MT010120UV01\_LV01LV02.DispenseRequest, 5.7. PORX\_MT020070UV01\_LV01LV02.CombinedMedicationDispense, 5.8. PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest, 5.12. QueryByParameterPayload Header, 5.13. PORX\_MT000007UV01\_LV01LV02.QueryByParameterPayload, 5.14. HL7 standard data structures usage examples, 5.14.1. Medication usage examples, 5.14.3. Quantity usage examples, 7.1.1.5. Get compensation conditions, 7.1.1.6. Book recipe number, 7.1.3.3. Get recipe data, 7.2.1.1. Book medication dispense, 7.2.1.3. Register medication dispense, 7.4.2.3. Get patients contact information by medication, 8.1. Error list  Added sections: 4.22. ValidateMedicationDispense, 7.2.1.2. Validate medication dispense. | 1.12 | J. Džeriņš |  |
| 28.05.2015 | Document updated according to Customer comments and checklist document “VVIS 1.kārtas dokumentu pārveidošana atbilstoši Projekta prasībām”.  Changed document identifier, number, fonts, updated sections/pages: 1st, 2nd, 3rd pages; 1. Definitions, symbols and abbreviations; 2. Introduction; 5.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload – parameter QueryByParameterPayload. parameterList. dispensedMedicine type, ORD.DIS description. | 1.13 | J. Džeriņš |  |
| 27.07.2015 | Document reviewed. Grammar and formatting corrections done. | 1.14 | V. Rubene | Quality control |
| 06.08.2015 | Document updated according to retrospection. | 1.15 | J. Džeriņš | Updates after retrospection. |
| 18.08.2015 | Document approved version | 2.00 | J. Džeriņš |  |
| 07.10.2015 | Document updated according to E-Health project 2nd stage requirements (Concluded framework agreements No. VMNVD 2014/3 ERAF and 31 july 2015 contract Nr. VMNVD 2014/3 ERAF 9).  Updated sections (data structures):  6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest,  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense.  Added sections (usage examples):  6.14.1. Potential compensation payer (sponsor) and compensation payer (sponsor) examples,  6.14.2. Person identification and documents examples. | 2.01 | J. Džeriņš |  |
| 09.10.2015 | Document reviewed | 2.01 | A.Spāģe | Quality control |
| 19.10.2015 | Document updated according to CKS comments. | 2.02 | J. Džeriņš | Update after CKS |
| 22.10.2015 | Document approved version | 3.00 | J. Džeriņš |  |
| 03.11.2015 | Document updated according to E-Health project 2nd stage requirements (Concluded framework agreements No. VMNVD 2014/3 ERAF and 22 October 2015 contract Nr. VMNVD 2014/3 ERAF 10).  Updated sections (data structures):  6.2. COCT\_MT050000UV01.Person,  6.6. PORX\_MT010120UV01\_LV02.DispenseRequest,  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense. | 3.01 | M. Trušelis |  |
| 10.11.2015 | Document updated according to CKS comments.  Updated sections:  6.2. COCT\_MT050000UV01.Person,  6.6. PORX\_MT010120UV01\_LV02.DispenseRequest,  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense,  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload.  Added sections: 6.14.3. Foreign citizen identification and documents examples. | 3.02 | M. Trušelis | Update after CKS |
| 27.11.2015 | Document updated according to CKS comments.  Updated sections:  6.2. COCT\_MT050000UV01.Person,  6.6. PORX\_MT010120UV01\_LV02.DispenseRequest,  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense,  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload. | 3.03 | M. Trušelis | Update after CKS |
| 07.12.2015 | Document approved version | 4.00 | M. Trušelis |  |
| 07.12.2015 | Described functions that support permission delegation.  Updated sections:  5.4. GetMedicationOrderData, 5.6. GetMedicationOrderList, 5.15. GetProfile  5.16. SetProfile, 6.9. PORX\_MT000019UV01\_LV01.ProfileSetupRequest.  Added section:  7. Permission delegation | 4.01 | M. Trušelis |  |
| 07.04.2016 | Document updated according to March 18 2016 contract Nr. VMNVD 2014/3 ERAF 11.  Updated section:  10.1. Error list. | 5.00 | M. Trušelis |  |
| 14.04.2016 | Document reviewed | 5.00 | A.Spāģe | Quality control |
| 04.10.2016 | Updated section:  10.1. Error list. | 5.01 | M. Trušelis |  |
| 05.10.2016 | Updated secitions:  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense;  6.14.1. Potential compensation payer (sponsor) and compensation payer (sponsor) examples. | 5.02 | M. Trušelis | Minor fixes. |
| 08.12.2016 | Updated section:  10.1. Error list. | 5.03 | M. Trušelis |  |
| 09.03.2017 | Updated sections:  6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest,  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload,  9.1.3.3. Get recipe data,  9.2.1.1. Book medication dispense,  9.2.1.3. Register medication dispense. | 6.00 | M. Trušelis |  |

Table of Contents

1. No table of figures entries found.Definitions, symbols and abbreviations 16

2. Introduction 16

2.1. Purpose 16

2.1.1. Document purpose 16

2.1.2. Document audience 16

2.2. Scope 16

2.3. References 16

2.4. Document structure 17

3. Interface description 17

4. Interface dependencies 19

5. Services 20

5.1. BookMedicationOrders 20

5.1.1. PORX\_IN000001UV01\_LV01 interaction 20

5.1.1.1. Payload 20

5.1.2. PORX\_IN000002UV01\_LV02 interaction 21

5.1.2.1. Payload 21

5.2. GetCompensationConditionList 21

5.2.1. PORX\_IN000003UV01\_LV01 interaction 22

5.2.1.1. Payload 23

5.2.2. PORX\_IN000004UV01\_LV01 interaction 24

5.2.2.1. Payload 25

5.3. RegisterMedicationOrder 25

5.3.1. PORX\_IN010380UV01\_LV02 interaction 26

5.3.1.1. Payload 26

5.3.2. PORX\_IN000002UV01\_LV02 interaction 26

5.3.2.1. Payload 27

5.4. GetMedicationOrderData 27

5.4.1. PORX\_IN000005UV01\_LV01 interaction 27

5.4.1.1. Payload 28

5.4.2. PORX\_IN000006UV01\_LV02 interaction 28

5.4.2.1. Payload 28

5.5. CancelMedicationOrder 28

5.5.1. PORX\_IN000025UV01\_LV01 interaction 29

5.5.1.1. Payload 29

5.5.2. MCCI\_IN000006UV01\_LV01 interaction 29

5.6. GetMedicationOrderList 29

5.6.1. PORX\_IN000007UV01\_LV02 interaction 30

5.6.1.1. Payload 30

5.6.2. PORX\_IN000006UV01\_LV02 interaction 31

5.6.2.1. Payload 31

5.7. GetMedicationOrderListContinuation 31

5.7.1. QUQI\_IN000003UV01\_LV01 interaction 31

5.7.1.1. Payload 32

5.7.2. PORX\_IN000006UV01\_LV02 interaction 32

5.7.2.1. Payload 32

5.8. GetMedicineList 32

5.8.1. PORX\_IN000008UV01\_LV02 interaction 33

5.8.1.1. Payload 33

5.8.2. PORX\_IN000009UV01\_LV01 interaction 33

5.8.2.1. Payload 33

5.9. GetDiagnosisList 34

5.9.1. PORX\_IN000010UV01\_LV02 interaction 34

5.9.1.1. Payload 34

5.9.2. PORX\_IN000011UV01\_LV01 interaction 35

5.9.2.1. Payload (data) 35

5.10. BookMedicationDispense 35

5.10.1. PORX\_IN000012UV01\_LV01 interaction 36

5.10.1.1. Payload 36

5.10.2. PORX\_IN000013UV01\_LV02 interaction 37

5.10.2.1. Payload 37

5.11. RegisterMedicationDispense 37

5.11.1. PORX\_IN020170UV01\_LV02 interaction 38

5.11.1.1. Payload 38

5.11.2. PORX\_IN000013UV01\_LV02 interaction 38

5.11.2.1. Payload 39

5.12. CancelMedicationDispense 39

5.12.1. PORX\_IN000014UV01\_LV01 interaction 39

5.12.1.1. Payload 40

5.12.2. MCCI\_IN000006UV01\_LV01 interaction 40

5.13. GetMedicationDispenseList 40

5.13.1. PORX\_IN000015UV01\_LV02 interaction 41

5.13.1.1. Payload 41

5.13.2. PORX\_IN000016UV01\_LV02 interaction 41

5.13.2.1. Payload 42

5.14. GetMedicationDispenseListContinuation 42

5.14.1. QUQI\_IN000003UV01\_LV01 interaction 42

5.14.1.1. Payload 43

5.14.2. PORX\_IN000016UV01\_LV02 interaction 43

5.14.2.1. Payload 43

5.15. GetProfile 43

5.15.1. PORX\_IN000017UV01\_LV01 interaction 44

5.15.1.1. Payload 44

5.15.2. PORX\_IN000018UV01\_LV01 interaction 45

5.15.2.1. Payload 45

5.16. SetProfile 45

5.16.1. PORX\_IN000019UV01\_LV01 interaction 46

5.16.1.1. Payload 46

5.16.2. MCCI\_IN000006UV01\_LV01 interaction 46

5.17. GetPatientContactList 46

5.17.1. PORX\_IN000020UV01\_LV02 interaction 47

5.17.1.1. Payload 47

5.17.2. PORX\_IN000021UV01\_LV01 interaction 47

5.17.2.1. Payload (data) 48

5.18. GetPatientContactListContinuation 48

5.18.1. QUQI\_IN000003UV01\_LV01 interaction 48

5.18.1.1. Payload 48

5.18.2. PORX\_IN000021UV01\_LV01 interaction 48

5.18.2.1. Payload (data) 49

5.19. RegisterMedicationWarning 49

5.19.1. PORX\_IN000022UV01\_LV01 interaction 49

5.19.1.1. Payload 50

5.19.2. MCCI\_IN000006UV01\_LV01 interaction 50

5.20. GetMedicationWarningList 50

5.20.1. PORX\_IN000023UV01\_LV01 interaction 50

5.20.1.1. Payload 51

5.20.2. PORX\_IN000024UV01\_LV01 interaction 51

5.20.2.1. Payload 51

5.21. GetMedicationWarningListContinuation 51

5.21.1. QUQI\_IN000003UV01\_LV01 interaction 52

5.21.1.1. Payload 52

5.21.2. PORX\_IN000024UV01\_LV01 interaction 52

5.21.2.1. Payload 52

5.22. ValidateMedicationDispense 53

5.22.1. PORX\_IN020170UV01\_LV02 interaction 53

5.22.1.1. Payload 53

5.22.2. MCCI\_IN000006UV01\_LV01 interaction 53

6. Data structures 54

6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest 54

6.2. COCT\_MT050000UV01.Person 60

6.3. COCT\_MT230100UV.Medicine 65

6.4. COCT\_MT090000UV01.AssignedEntity 67

6.5. PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest 70

6.6. PORX\_MT010120UV01\_LV02.DispenseRequest 73

6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense 76

6.8. PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest 81

6.9. PORX\_MT000019UV01\_LV01.ProfileSetupRequest 82

6.10. PORX\_MT000022UV01\_LV01.MedicationWarning 84

6.11. QueyContinuation Header 85

6.12. QueryByParameterPayload Header 86

6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload 88

6.14. HL7 standard data structures usage examples 93

6.14.1. Potential compensation payer (sponsor) and compensation payer (sponsor) examples 94

6.14.2. Person identification and documents examples 95

6.14.3. Foreign citizen identification and documents examples 95

6.14.4. Medication usage examples 96

6.14.5. Period usage examples 96

6.14.6. Quantity usage examples 96

6.14.7. Name usage examples 97

6.14.8. Address usage examples 97

6.14.8.1. Foreign country address 97

6.14.8.2. Latvian address 97

7. Permission delegation 101

8. Traceability matrix 101

9. Services usage explanation 105

9.1. Physician processes 105

9.1.1. Recipe prescription 105

9.1.1.1. Get patient potentially used medicines 105

9.1.1.2. Get frequently used physician diagnoses 106

9.1.1.3. Get medicines frequently prescribed to patient 106

9.1.1.4. Get medicines frequently prescribed by physician 106

9.1.1.5. Get compensation conditions 107

9.1.1.6. Book recipe number 108

9.1.1.7. Prescribe recipe 108

9.1.2. Recipe reservation 108

9.1.2.1. Recipe reservation 108

9.1.3. Get list of recipes 109

9.1.3.1. Get recipes prescribed by physician 109

9.1.3.2. Get patient recipes 109

9.1.3.3. Get recipe data 110

9.1.4. Cancel recipe 112

9.1.4.1. Get recipe 112

9.1.4.2. Cancel recipe 112

9.2. Pharmacist services 112

9.2.1. Medications dispense and dispense cancellation 112

9.2.1.1. Book medication dispense 113

9.2.1.2. Validate medication dispense 115

9.2.1.3. Register medication dispense 115

9.2.1.4. Cancel medication dispense 116

9.2.2. Paper recipe importing 117

9.2.2.1. Get paper recipe data and number 117

9.2.2.2. Import or edit paper recipe data 117

9.2.3. Pharmacy medication 117

9.2.3.1. Get medications prescribed to pharmacy 117

9.2.3.2. Get medications dispensed from pharmacy 118

9.2.4. Patient recipes 118

9.2.4.1. Patient recipes by person code (demo) 118

9.3. Patient services 119

9.3.1. Patients recipes 119

9.3.1.1. Get patient recipes 119

9.3.1.2. Get recipe data 119

9.3.2. Patient pharmacy 119

9.3.2.1. Get pharmacy 119

9.3.2.2. Set pharmacy 120

9.4. Supervisor institution services 120

9.4.1. Recipe management 120

9.4.1.1. Get recipes 120

9.4.1.2. Get recipe data 121

9.4.1.3. Cancel recipe 121

9.4.2. Medication management 121

9.4.2.1. Get medication warnings 121

9.4.2.2. Register medication warning 122

9.4.2.3. Get patients contact information by medication 122

10. Appendixes 123

10.1. Error list 123

List of Images

[Picture 1 Interface architecture 18](#_Toc476835862)

[Picture 2 Messaging concepts 18](#_Toc476835863)

[Picture 3 Messaging standard 19](#_Toc476835864)

[Picture 4 BookMedicationOrders service 20](#_Toc476835865)

[Picture 5 PORX\_IN000001UV01\_LV01 interaction payload 20](#_Toc476835866)

[Picture 6 GetCompensationConditionList service 22](#_Toc476835867)

[Picture 7 PORX\_IN000003UV01\_LV01 interaction payload 23](#_Toc476835868)

[Picture 8 PORX\_IN000004UV01\_LV01 interaction payload 25](#_Toc476835869)

[Picture 9 RegisterMedicationOrder service 26](#_Toc476835870)

[Picture 10 GetMedicationOrderData service 27](#_Toc476835871)

[Picture 11 PORX\_IN000005UV01\_LV01 interaction payload 28](#_Toc476835872)

[Picture 12 CancelMedicationOrder service 29](#_Toc476835873)

[Picture 13 GetMedicationOrderList service 30](#_Toc476835874)

[Picture 14 GetMedicationOrderListContinuation service 31](#_Toc476835875)

[Picture 15 GetMedicineList service 33](#_Toc476835876)

[Picture 16 GetDiagnosisList service 34](#_Toc476835877)

[Picture 17 PORX\_IN000011UV01\_LV01 interaction payload (data) 35](#_Toc476835878)

[Picture 18 BookMedicationDispense service 36](#_Toc476835879)

[Picture 19 PORX\_IN000012UV01\_LV01 interaction payload 36](#_Toc476835880)

[Picture 20 RegisterMedicationDispense service 38](#_Toc476835881)

[Picture 21 CancelMedicationDispense service 39](#_Toc476835882)

[Picture 22 PORX\_IN000014UV01\_LV01 interaction payload 40](#_Toc476835883)

[Picture 23 GetMedicationDispenseList service 41](#_Toc476835884)

[Picture 24 GetMedicationDispenseListContinuation service 42](#_Toc476835885)

[Picture 25 GetProfile service 44](#_Toc476835886)

[Picture 26 PORX\_IN000017UV01\_LV01 interaction payload 44](#_Toc476835887)

[Picture 27 SetProfile service 46](#_Toc476835888)

[Picture 28 GetPatientContactList service 47](#_Toc476835889)

[Picture 29 GetPatientContactListContinuation service 48](#_Toc476835890)

[Picture 30 RegisterMedicationWarning service 49](#_Toc476835891)

[Picture 31 GetMedicationWarningList service 50](#_Toc476835892)

[Picture 32 PORX\_IN000023UV01\_LV01 interaction payload 51](#_Toc476835893)

[Picture 33 GetMedicationWarningListContinuation service 52](#_Toc476835894)

[Picture 34 ValidateMedicationDispense service 53](#_Toc476835895)

[Picture 35 PORX\_MT010120UV01\_LV02.CombinedMedicationRequest data structure (part 1) 54](#_Toc476835896)

[Picture 36 PORX\_MT010120UV01\_LV02.CombinedMedicationRequest data structure (part 2) 55](#_Toc476835897)

[Picture 37 COCT\_MT050000UV01.Person data structure (part 1) 60](#_Toc476835898)

[Picture 38 COCT\_MT050000UV01.Person data structure (part 2) 61](#_Toc476835899)

[Picture 39 COCT\_MT230100UV.Medicine data structure 65](#_Toc476835900)

[Picture 40 COCT\_MT090000UV01.AssignedEntity data structure 67](#_Toc476835901)

[Picture 41 PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest data structure (part 1) 70](#_Toc476835902)

[Picture 42 PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest data structure (part 2) 71](#_Toc476835903)

[Picture 43 PORX\_MT010120UV01\_LV02.DispenseRequest data structure 73](#_Toc476835904)

[Picture 44 PORX\_MT020070UV01\_LV02.CombinedMedicationDispense data structure 76](#_Toc476835905)

[Picture 45 PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest data structure 81](#_Toc476835906)

[Picture 46 PORX\_MT000019UV01\_LV01.ProfileSetupRequest data structure 82](#_Toc476835907)

[Picture 47 PORX\_MT000022UV01\_LV01.MedicationWarning data structure 84](#_Toc476835908)

[Picture 48 QueyContinuation Header data structure 85](#_Toc476835909)

[Picture 49 QueryByParameterPayload Header data structure 86](#_Toc476835910)

[Picture 50 PORX\_MT000007UV01\_LV02.QueryByParameterPayload data structure 88](#_Toc476835911)

List of Tables

# No table of figures entries found.Definitions, symbols and abbreviations

Table 1 Definitions, acronyms and abbreviations

|  |  |
| --- | --- |
| Acronym, abbreviation | Definition |
| System | Electronic Recipe Information System |
| XML | Extensible Markup Language |
| SOAP | Simple Object Access Protocol |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| HL7 | Health Level Seven International  (<http://www.hl7.org>) |

# Introduction

## Purpose

### Document purpose

Document describes functional and non-functional requirements to provide the software interface between Electronic Recipe Information System (System) and other central or local information systems involved in healthcare processes. Document includes description of data transfer protocols, data exchange message formats and semantic description.

### Document audience

System interface specification (SIS) refers to The National Health Service Latvia (Customer) ordered project “Development of Electronic Recipe Information System” (procurement identification No.VEC 2010/5/ERAF, 1st phase), which is developed by person group Ltd. „In-volv Latvia” & Ltd. „ABC Software” with subcontractor „INDRA SISTEMAS” S.A. (Executor) and project “Expansion of Single Health Information System” (procurement identification No. VM NVD 2014/3 ERAF, 2nd phase) and its contract 27 february 2015 contract Nr. VMNVD 2014/3 ERAF 5 and March 18, 2016 contract Nr. VMNVD 2014/3 ERAF 11, which is developed by Ltd “Lattelecom”.

Document may be made available to third parties by Customer.

## Scope

Document refers to compliance with IEEE Std 1016-1998 standard [1] and IEEE Std 1471-2000 standard [10].

## References

Table 2 Reference documents

|  |  |
| --- | --- |
| Nr. | Document name (identification, version) |
| [1] | IEEE Recommended Practice for  Software Design Descriptions (IEEE Std 1016-1998) |
| [2] | Tehniskās arhitektūras apraksts (NVD.VVIS.TAA.07) |
| [3] | Projekta Integrācijas platformas informācijas sistēmas izstrāde arhitektūras risinājuma vīzijas documents (NVD.IP.REQ.VIZ.1.07) |
| [4] | HL7 Version 3 Standard: Transport Specification - Web Services Profile, Release 2  (<http://www.hl7.org/v3ballot/html/infrastructure/transport/transport-wsprofiles.html>) |
| [5] | HL7 3.versijas lietotāja ceļvedis (HL7 V3 Guide) |
| [6] | Programmatūras prasību specifikācija (NVD.VVIS.REC.PAK.PPS.5.00) |
| [7] | E-veselības Integrācijas Platformas saskarņu vadlīnijas (VEC.STD.WS.1.04) |
| [8] | E-veselības ziņojumapmaiņā izmantojamo datu struktūras (VEC.STD.HL7.1.02) |
| [10] | IEEE Recommended Practice for Architectural Description for Software-Intensive Systems (IEEE Std 1471-2000) |

## Document structure

This document will be structured in the following sections:

1. **Definitions, symbols and abbreviations** – This chapter contains the table of definitions, symbols and abbreviations used in this document.
2. **Introduction** – This chapter contains brief description of the document purpose and content.
3. **Interface description** – This chapter contains a description of the interface architecture and common requirements to data exchange messages and protocols.
4. **Interface dependencies** – This chapter contains a description of the interface dependencies.
5. **Services** – This chapter specifies services published by System, including the detailed description of input and output data structures.
6. **Data structures** – This chapter specifies business data structures used in data exchange messages.
7. **Permission delegation** – This chapter specifies permission delegation.
8. **Traceability matrix** – This chapter contains traceability matrix between specified services and system requirements.
9. **Services usage explanation** – This chapter contains service usage explanation, by providing examples of services and its parameters usage for common processes.
10. **Appendixes.**

# Interface description

System provides software interfaces for external service consumers, which are external information systems (i.e. Hospital Information Systems, Pharmacy Information Systems, e‑Health Portal), as web services. These web services are published in the e‑Health Integration Platform – according to e‑Health Architecture Guidelines (see [2]) and e‑Health Integration Platform requirements (see [3]).

Next picture illustrates the overall architecture of System interfaces.



Picture 1 Interface architecture

Published web services, as well as XML schemas are available in e‑Health Integration Platform catalogue UDDI based catalogue. In terms of messaging, e‑Health Integration Platform available web services will use SOAP over HTTP protocol with HL7 v3 extensions for SOAP. Next picture illustrates messaging concepts:



Picture 2 Messaging concepts

For detailed information about Service Catalogue, messaging concepts in e‑Health Integration Platform and its message standard see e‑Health Integration Platform architectural documentation [3].

System software interface, according to e‑Health Integration Platform requirements, supports HL7 message transport standard and uses HL7 message envelope for all messages (see [3], [4], [5], [7] and [8]).

For messages containing clinic information HL7 requirements may also apply to message content itself. Next picture illustrates high level concepts for used message standard in Integration Platform.



Picture 3 Messaging standard

Each service consists of 2 interactions as input and output services. Each interaction consists of 3 layers:

* Transmission wrapper;
* Trigger Event Control Act;
* Payload.

Transmission wrappers used in e‑Health are of 2 kinds:

* MCCI\_MT000100UV01\_LV01 – for input interaction;
* MCCI\_MT000200UV01\_LV01 – for output interaction.

Trigger Event Control Act used in e‑Health is the same for every interaction:

* MCAI\_MT700201UV01\_LV01.

Payload contains data information. For some of interactions, there can be no data (for example, output iteration with no data); in such cases there have been used interaction:

* MCCI\_IN000006UV01\_LV01,

In output interactions, there can be returned some messages. For detailed information on services and its interactions see [7].

For every interaction payload and data structure, there is a table of elements, which describes the entire used elements from appropriate schema. Elements, that are not included to table are not used. Additionally, table can contain element, which rows background color is gray – it means that such element is complex type and its further description should be found in appropriate data structure. For every element in table, there is element name, type, cardinality and description.

In service usage explanation, there are main processes and traceability to specific requirement described in the Software requirements specification document [6].

# Interface dependencies

Elements of data exchange messages related to patient or healthcare provider identification (name, surname, identification code, etc) depend on the relevant elements definition in the e‑Health Integration Platform system analysis and design documents.

Elements – references to common classifiers or central registers depend on the classifier value element definition in the e‑Health Integration Platform system analysis and design documents.

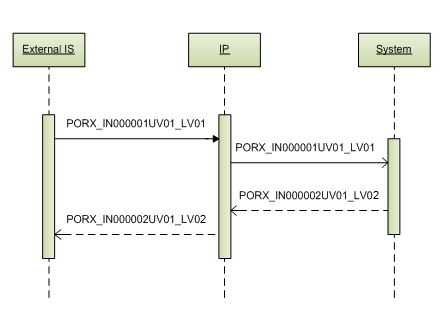
# Services

## BookMedicationOrders

Allows external IS (e‑Health Portal or Hospital IS) to book up to X medication orders (X – System parameter) for Y days (Y – System parameter).

Service roles: physician.

Service rights: RegisterMedicationOrder.



Picture 4 BookMedicationOrders service

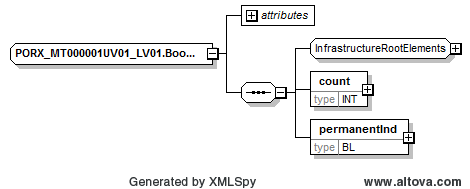
### PORX\_IN000001UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 3 Interaction PORX\_IN000001UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000001UV01\_LV01 | Payload |

#### Payload



Picture 5 PORX\_IN000001UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000001UV01\_LV01.BookMedicationOrderRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="count" type="INT" maxOccurs="1"/>

<xs:element name="permanentInd" type="BL"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMood" use="optional" fixed="RQO"/>

</xs:complexType>

Table 4 Payload PORX\_MT000001UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| bookMedicationOrderRequest | PORX\_MT000001UV01\_LV01.BookMedicationOrderRequest | 1..1 |  |
| bookMedicationOrderRequest. count | INT | 1..1 | Number of medication orders to book |
| bookMedicationOrderRequest. permanentInd | BLrmanentIndeters.rss usage for s.izmantota dažādos nolūkos un metodēs, katrai no tām atgriežot/padantots savs attiecīgais termi | 1..1 | Indicator: book permanently medication orders.  Value “true” must be used for recipe reservation (permanent record will be made). Value “false” must be used for recipe prescription (temporary record will be made with its validity date). |

### PORX\_IN000002UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 5 Interaction PORX\_IN000002UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

Table 6 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..unbounded | See data structure in chapter 6.1. |

## GetCompensationConditionList

Allows external IS (e‑Health Portal, Hospital IS) to get compensation conditions depending on such data: patient, age, gender, diagnosis, medicine, physician specialty.

Service roles: physician.

Service rights: RegisterMedicationOrder.



Picture 6 GetCompensationConditionList service

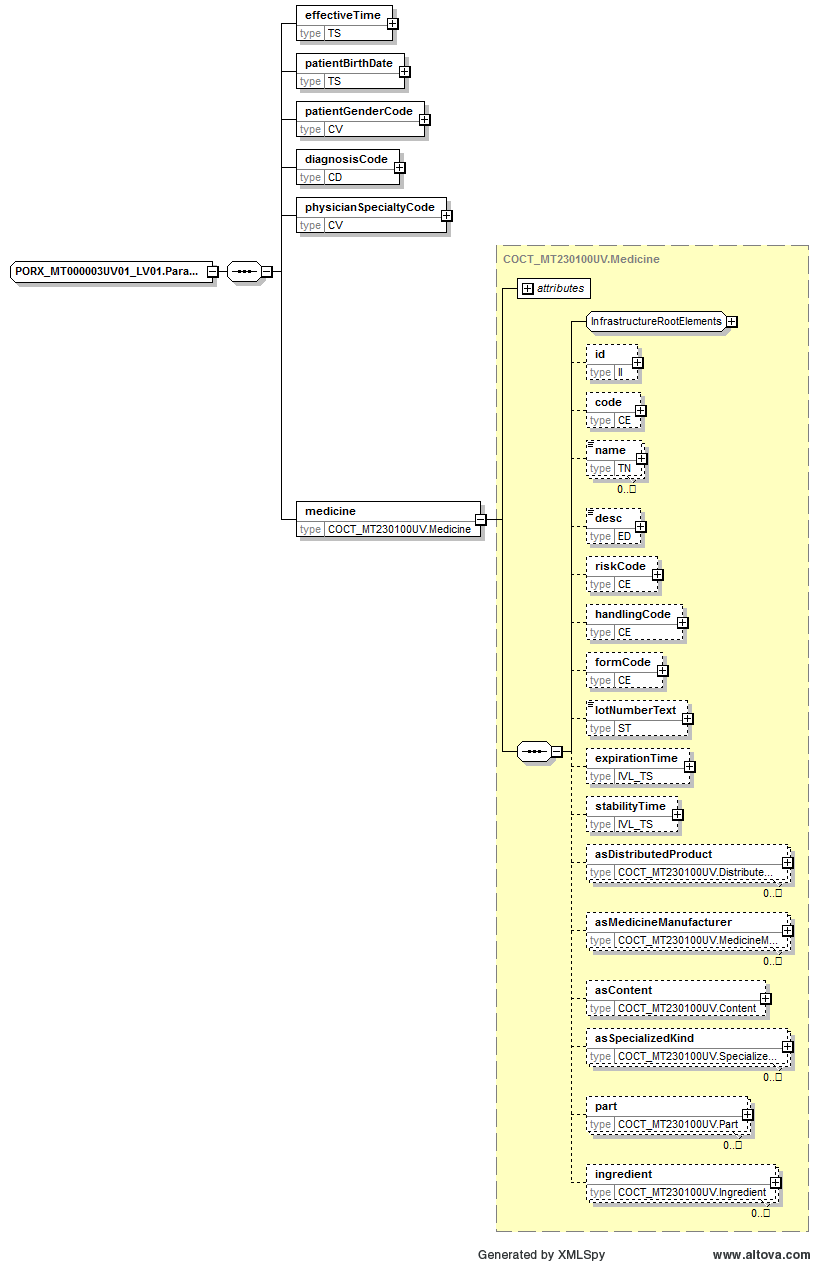
### PORX\_IN000003UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 7 Interaction PORX\_IN000003UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000003UV01\_LV01 | Payload |

#### Payload



Picture 7 PORX\_IN000003UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000003UV01\_LV01.ParameterList">

<xs:sequence>

<xs:element name="effectiveTime" type="TS"/>

<xs:element name="patientId" type="II" minOccurs="0"/>

<xs:element name="patientBirthDate" type="TS"/>

<xs:element name="patientGenderCode" type="CV"/>

<xs:element name="diagnosisCode" type="CD"/>

<xs:element name="physicianSpecialtyCode" type="CV"/>

<xs:element name="medicine" type="COCT\_MT230100UV.Medicine"/>

</xs:sequence>

</xs:complexType>

Table 8 Payload PORX\_MT000003UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| QueryByParameterPayload | PORX\_MT000003UV01\_LV01.QueryByParameterPayload | 1..1 | See data structure in chapter 6.12. |
| QueryByParameterPayload. parameterList | PORX\_MT000003UV01\_LV01.ParameterList | 1..1 |  |
| QueryByParameterPayload. parameterList. effectiveTime | TS | 1..1 | Date for which is requested compensation conditions |
| QueryByParameterPayload. parameterList. patientId | II | 0..1 | Person identifier.  Personal code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1). |
| QueryByParameterPayload. parameterList. patientBirthDate | TS | 1..1 | Patient birth date for compensation validation |
| QueryByParameterPayload. parameterList. patientGenderCode | CV | 1..1 | Patient gender (OID: 1.3.6.1.4.1.38760.2.111) |
| QueryByParameterPayload. parameterList. diagnosisCode | CD | 1..1 | Diagnosis and additional diagnosis ICD-10 code (OID: 1.3.6.1.4.1.38760.2.159) |
| QueryByParameterPayload. parameterList. physicianSpecialtyCode | CV | 1..1 | Physician specialty code (OID: 1.3.6.1.4.1.38760.2.38) |
| QueryByParameterPayload. parameterList. medicine | COCT\_MT230100UV.Medicine | 1..1 |  |
| QueryByParameterPayload. parameterList. medicine. code | CE | 0..1 | Medical registration number (OID: 1.3.6.1.4.1.38760.2.136) or compensation group (OID: 1.3.6.1.4.1.38760.2.177) |
| QueryByParameterPayload. parameterList. medicine. formCode | CE | 0..1 | Medicine form code (OID: 1.3.6.1.4.1.38760.2.137) |

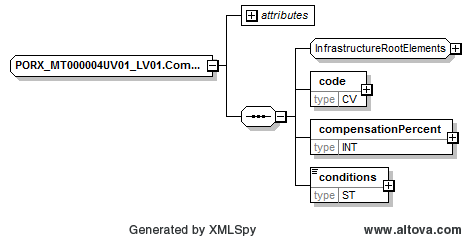
### PORX\_IN000004UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 9 Interaction PORX\_IN000004UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000004UV01\_LV01 | Payload |

#### Payload



Picture 8 PORX\_IN000004UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000004UV01\_LV01.CompensationCondition">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="code" type="CV"/>

<xs:element name="compensationPercent" type="INT"/>

<xs:element name="conditions" type="ST"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMoodEventCriterion" use="optional" fixed="EVN.CRT"/>

</xs:complexType>

Table 10 Payload PORX\_MT000004UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| compensationCondition | PORX\_MT000004UV01\_LV01.CompensationCondition | 0..unbounded |  |
| compensationCondition. code | CV | 1..1 | Compensation code (OID: 1.3.6.1.4.1.38760.2.152) |
| compensationCondition. compensationPercent | INT | 1..1 | Amount of compensation in percentage |
| compensationCondition. conditions | ST | 1..1 | Conditions of compensation |

## RegisterMedicationOrder

Allows external IS (e‑Health Portal, Hospital IS) to register medication order – prescribe and save it to System by making required validations.

Before this, service “BookMedicationOrders” (see “5.1 BookMedicationOrders”, page 20) must be called to get medication order identifier.

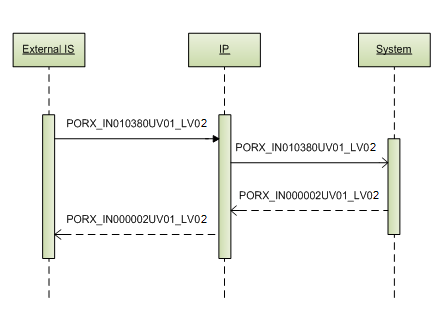
Service role: physician.

Allows external IS (e‑Health Portal, Pharmacy IS) to import and edit paper medication order data including all its medication dispenses to System.

Valid paper medication orders which have at least one medication dispense and are dispensed completely can be imported only.

Service roles: pharmacist.

Service rights: RegisterMedicationOrder, ImportMedicationOrder.



Picture 9 RegisterMedicationOrder service

### PORX\_IN010380UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 11 Interaction PORX\_IN010380UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

Table 12 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..1 | See data structure in chapter 6.1. |

### PORX\_IN000002UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 13 Interaction PORX\_IN000002UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

Table 14 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..unbounded | See data structure in chapter 6.1. |

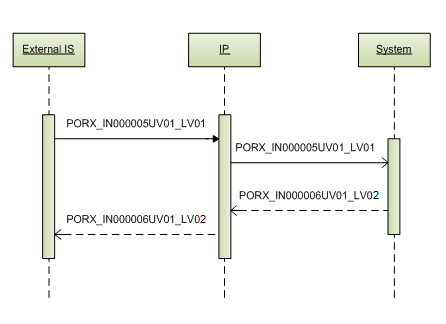
## GetMedicationOrderData

Allows external IS (e‑Health Portal, Hospital IS, Pharmacy IS) to retrieve all the information about medication order by its identifier. This function supports permission delegation (see chapter 7 Permission delegation).

Service roles: physician, pharmacist, patient, supervising institution.

Service rights: QueryMedicationOrders.

* QueryPatientActiveMedicationOrders – to get specific patient active medication orders;
* QueryPatientAllMedicationOrders – to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders;
* QueryAllMedicationOrders – to get all medication orders;
* QueryMedicationDispenses – to get medication dispenses;
* QueryCancellationMessages – to get cancelation messages.



Picture 10 GetMedicationOrderData service

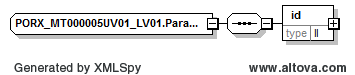
### PORX\_IN000005UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 15 Interaction PORX\_IN000005UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000005UV01\_LV01 | Payload |

#### Payload



Picture 11 PORX\_IN000005UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000005UV01\_LV01.ParameterList">

<xs:sequence>

<xs:element name="id" type="II"/>

</xs:sequence>

</xs:complexType>

Table 16 Payload PORX\_MT000005UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| QueryByParameterPayload | PORX\_MT000005UV01\_LV01.QueryByParameterPayload | 1..1 | See data structure in chapter 6.12. |
| QueryByParameterPayload. parameterList | PORX\_MT000005UV01\_LV01.ParameterList | 1..1 |  |
| QueryByParameterPayload. parameterList. id | II | 1..1 | Medication order identifier (OID: 1.3.6.1.4.1.38760.3.4.11.1) |

### PORX\_IN000006UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 17 Interaction PORX\_IN000006UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

Table 18 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..unbounded | See data structure in chapter 6.1. |

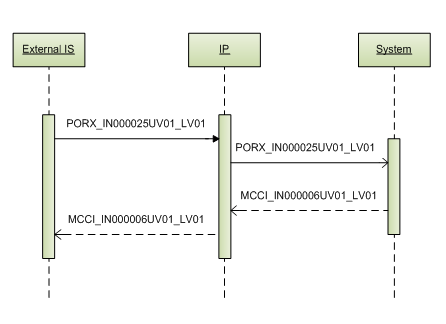
## CancelMedicationOrder

Allows external IS (e‑Health Portal, Hospital IS) to cancel medication order by its identifier, requiring passing cancellation reason and date.

Service roles: physician, supervising institution.

Service rights: CancelMedicationOrder.

* CancelAllMedicationOrders – to cancel medication orders.



Picture 12 CancelMedicationOrder service

### PORX\_IN000025UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 19 Interaction PORX\_IN000025UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000025UV01\_LV01 | Payload |

#### Payload

Table 20 Payload PORX\_MT000025UV01\_LV01 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| cancelMedicationOrderRequest | PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest | 1..1 | See data structure in chapter 6.8. |

### MCCI\_IN000006UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 21 Interaction MCCI\_IN000006UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

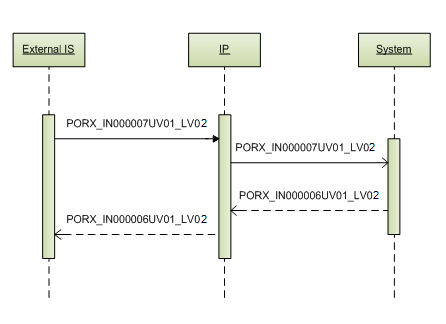
## GetMedicationOrderList

Allows external IS (e‑Health Portal, Hospital IS) to retrieve list of medication orders, allowing filtering list by different parameters. This function supports permission delegation (see chapter 7 Permission delegation).

Service roles: physician, pharmacist, patient, supervisor institution.

Service rights: QueryMedicationOrders.

* QueryPatientActiveMedicationOrders – to get specific patient medication orders;
* QueryPatientAllMedicationOrders – to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders;
* QueryAllMedicationOrders – to get all medication orders;
* QueryMedicationDispenses – to get medication dispenses;
* QueryCancellationMessages – to get cancelation messages.



Picture 13 GetMedicationOrderList service

### PORX\_IN000007UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 22 Interaction PORX\_IN000007UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000007UV01\_LV02 | Payload |

#### Payload

Table 23 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.13. |

### PORX\_IN000006UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 24 Interaction PORX\_IN000006UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

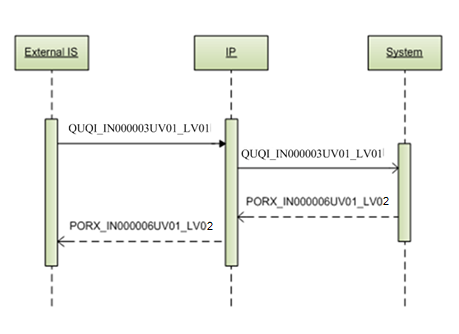
Table 25 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..unbounded | See data structure in chapter 6.1. |

## GetMedicationOrderListContinuation

Allows external IS (e‑Health Portal, Hospital IS) to continue retrieving data by pages, started with service “GetMedicationOrderList” request.

Service roles and rights are the same as for service “GetMedicationOrderList”.



Picture 14 GetMedicationOrderListContinuation service

### QUQI\_IN000003UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 26 Interaction QUQI\_IN000003UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| QUQI\_MT000001UV01 | Trigger Event Control Act |
| QUQI\_MT000001UV01 | Payload |

#### Payload

Table 27 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryContinuation | QUQI\_MT000001UV01.QueryContinuation | 1..1 | See data structure in chapter 6.11. |

### PORX\_IN000006UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 28 Interaction PORX\_IN000006UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT010120UV01\_LV02 | Payload |

#### Payload

Table 29 Payload PORX\_MT010120UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 0..unbounded | See data structure in chapter 6.1. |

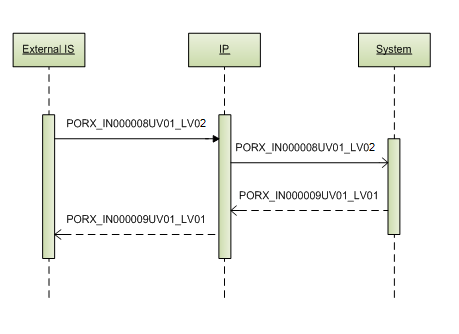
## GetMedicineList

Allows external IS (e‑Health Portal, Hospital IS) to retrieve list of medicines frequently prescribed by physician or to patient during X month ago (X – System parameter).

Service roles: physician.

Service rights: QueryMedicationOrders.

* QueryPatientActiveMedicationOrders – to get specific patient active medication orders;
* QueryPatientAllMedicationOrders – to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders;
* QueryAllMedicationOrders – to get all medication orders.



Picture 15 GetMedicineList service

### PORX\_IN000008UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 30 Interaction PORX\_IN000008UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000007UV01\_LV02 | Payload |

#### Payload

Table 31 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.13. |

### PORX\_IN000009UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 32 Interaction PORX\_IN000009UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| COCT\_MT230100UV | Payload |

#### Payload

Table 33 Payload COCT\_MT230100UV elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| medicine | COCT\_MT230100UV.Medicine | 0..unbounded | See data structure in chapter 6.3. |

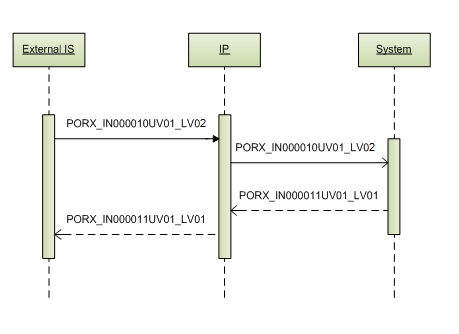
## GetDiagnosisList

Allows external IS (e‑Health Portal, Hospital IS) to retrieve list of diagnoses frequently used by physician during X month ago (X – System parameter).

Service roles: physician.

Service rights: QueryMedicationOrders.

* QueryPatientActiveMedicationOrders – to get specific patient medication orders;
* QueryPatientAllMedicationOrders to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders.
* QueryAllMedicationOrders – to get all medication orders.



Picture 16 GetDiagnosisList service

### PORX\_IN000010UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 34 Interaction PORX\_IN000010UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000007UV01\_LV02 | Payload |

#### Payload

Table 35 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.13. |

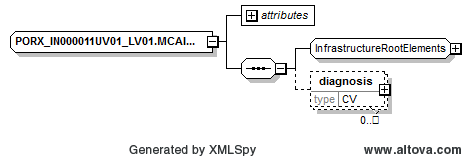
### PORX\_IN000011UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 36 Interaction PORX\_IN000011UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

#### Payload (data)



Picture 17 PORX\_IN000011UV01\_LV01 interaction payload (data)

Payload XML schema:

<xs:complexType name="PORX\_IN000011UV01\_LV01.MCAI\_MT700201UV01\_LV01.Subject">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="diagnosis" type="CV" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="typeCode" type="ActRelationshipHasSubject" use="required" fixed="SUBJ"/>

<xs:attribute name="contextConductionInd" type="bl" use="optional" default="false"/>

</xs:complexType>

Table 37 Payload (data) elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| diagnosis | CV | 0..unbounded | Diagnosis ICD-10 code (OID: 1.3.6.1.4.1.38760.2.159) |

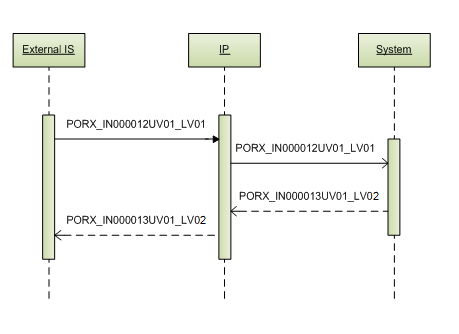
## BookMedicationDispense

Allows external IS (e‑Health Portal, Pharmacy IS) to retrieve medication order data for medication dispensing (including new medication dispense identifier, which is valid for Y hours, where Y – System parameter), in addition, blocking the medication order for submitting or canceling medication dispense, if necessary pass the real medication dispense date to validate medication dispensing correctly.

Service roles: pharmacist.

Service rights: QueryMedicationOrders, RegisterMedicationDispense.

* QueryPatientActiveMedicationOrders – to get specific patient active medication orders;
* QueryPatientAllMedicationOrders – to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders;
* QueryAllMedicationOrders – to get all medication orders.



Picture 18 BookMedicationDispense service

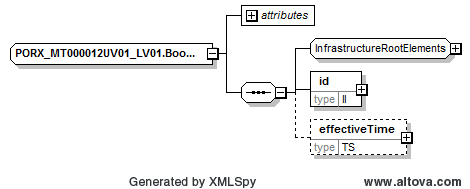
### PORX\_IN000012UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 38 Interaction PORX\_IN000012UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000012UV01\_LV01 | Payload |

#### Payload



Picture 19 PORX\_IN000012UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000012UV01\_LV01.BookMedicationDispenseRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="1" maxOccurs="1"/>

<xs:element name="effectiveTime" type="TS" minOccurs="0"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMood" use="optional" fixed="RQO"/>

</xs:complexType>

Table 39 Payload PORX\_MT000012UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| bookMedicationDispenseRequest | PORX\_MT000012UV01\_LV01.BookMedicationDispenseRequest | 1..1 |  |
| bookMedicationDispenseRequest. id | II | 1..1 | Medication order id (OID: 1.3.6.1.4.1.38760.3.4.11.1) |
| bookMedicationDispenseRequest. effectiveTime | TS | 0..1 | Medication dispense time, required only if fulfillment of medication dispense to System is being made after a time |

### PORX\_IN000013UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 40 Interaction PORX\_IN000013UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

Table 41 Payload PORX\_MT020070UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..1 | See data structure in chapter 6.7. |

## RegisterMedicationDispense

Allows external IS (e‑Health Portal, Pharmacy IS) to add or edit medication dispense data to medication order. For editing medication dispense, medication dispense identifier must be passed.

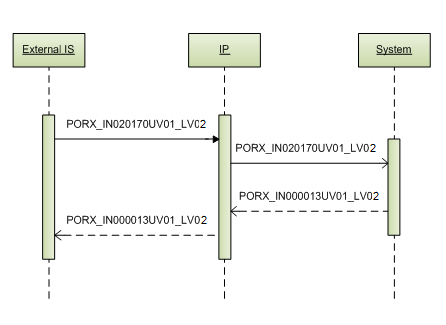
For new medication dispense submitting, before this, service “BookMedicationDispense” (see “5.10 BookMedicationDispense”, page 35) must be called to block medication order for data modifications and to get appropriate medication dispense identifier.

For existing medication dispense editing, before this, service “GetMedicationOrderData” (see “5.4 GetMedicationOrderData”, page 27) must be called to get appropriate medication dispense identifier.

Service roles: pharmacist.

Service rights: RegisterMedicationDispense.

* EditMedicationDispense –to edit medication dispense.



Picture 20 RegisterMedicationDispense service

### PORX\_IN020170UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 42 Interaction PORX\_IN020170UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

Table 43 Payload PORX\_MT020070UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..1 | See data structure in chapter 6.7. |

### PORX\_IN000013UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 44 Interaction PORX\_IN000013UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

Table 45 Payload PORX\_MT020070UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..1 | See data structure in chapter 6.7. |

## CancelMedicationDispense

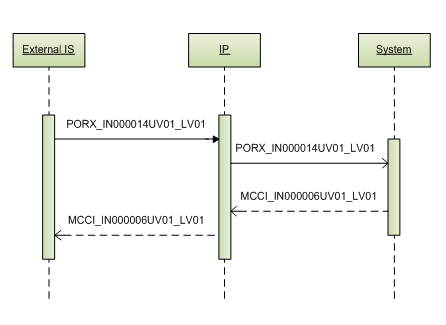
Allows external IS (e‑Health Portal, Pharmacy IS) to cancel medication dispense adding or editing. Medication dispense identifier must be passed.

Before this, service ““BookMedicationDispense” (see “5.10 BookMedicationDispense”, page 35) must be called to block medication order for data modifications and to get appropriate medication dispense identifier.

Service roles: pharmacist.

Service rights: RegisterMedicationDispense.

* CancelMedicationDispense – to cancel medication dispense.



Picture 21 CancelMedicationDispense service

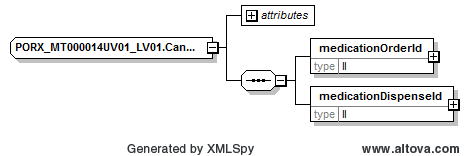
### PORX\_IN000014UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 46 Interaction PORX\_IN000014UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000014UV01\_LV01 | Payload |

#### Payload



Picture 22 PORX\_IN000014UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000014UV01\_LV01.CancelMedicationDispenseRequest">

<xs:sequence>

<xs:element name="medicationOrderId" type="II"/>

<xs:element name="medicationDispenseId" type="II"/>

</xs:sequence>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required" fixed="RQO"/>

</xs:complexType>

Table 47 Payload PORX\_MT000014UV01\_LV01 elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| cancelMedicationDispenseRequest | PORX\_MT000014UV01\_LV01.CancelMedicationDispenseRequest | 1..1 |  |
| cancelMedicationDispenseRequest. medicationOrderId | II | 1..1 | Medication order id (OID: 1.3.6.1.4.1.38760.3.4.11.1) |
| cancelMedicationDispenseRequest. medicationDispenseId | II | 1..1 | Medication dispense id (OID: 1.3.6.1.4.1.38760.3.4.11.3) |

### MCCI\_IN000006UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 48 Interaction MCCI\_IN000006UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

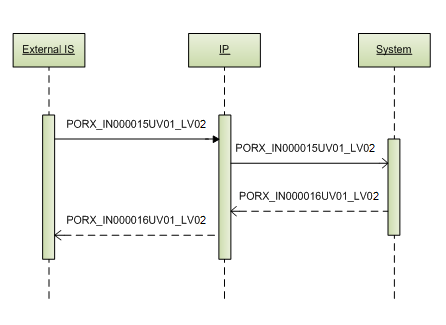
## GetMedicationDispenseList

Allows external IS (e‑Health Portal, Pharmacy IS) to retrieve list of medication dispenses, which was made in pharmacy, allowing filtering list by date, compensated/all medicines.

Service roles: pharmacist.

Service rights: QueryMedicationOrders, QueryMedicationDispenses.

* QueryPatientActiveMedicationOrders – to get specific patient active medication orders;
* QueryPatientAllMedicationOrders – to get specific patient all medication orders;
* QueryOrganizationMedicationOrders – to get specific pharmacy medication orders;
* QueryAllMedicationOrders – to get all medication orders.



Picture 23 GetMedicationDispenseList service

### PORX\_IN000015UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 49 Interaction PORX\_IN000015UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000007UV01\_LV02 | Payload |

#### Payload

Table 50 Payload PORX\_MT000007UV01\_LV02 elements

| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.13. |

### PORX\_IN000016UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 51 Interaction PORX\_IN000016UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

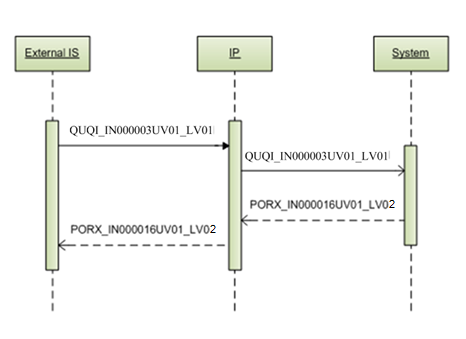
Table 52 Payload PORX\_MT020070UV01\_LV02 elements

| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..unbounded | See data structure in chapter 6.7. |

## GetMedicationDispenseListContinuation

Allows external IS (e‑Health Portal, Pharmacy IS) to continue retrieving data by pages, started with service “GetMedicationDispenseList” request.

Service roles and rights are the same as for service “GetMedicationDispenseList”.



Picture 24 GetMedicationDispenseListContinuation service

### QUQI\_IN000003UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 53 Interaction QUQI\_IN000003UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| QUQI\_MT000001UV01 | Trigger Event Control Act |
| QUQI\_MT000001UV01 | Payload |

#### Payload

Table 54 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryContinuation | QUQI\_MT000001UV01.QueryContinuation | 1..1 | See data structure in chapter 6.11. |

### PORX\_IN000016UV01\_LV02 interaction

This is service response interaction from System; its specification is explained in next table.

Table 55 Interaction PORX\_IN000016UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

Table 56 Payload PORX\_MT020070UV01\_LV02 elements

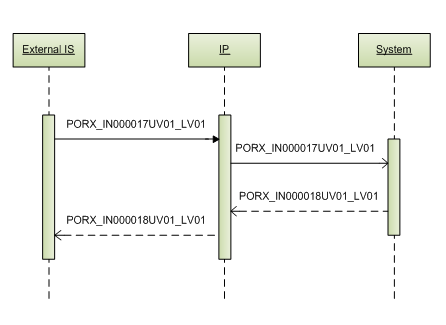
| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..unbounded | See data structure in chapter 6.7. |

## GetProfile

Allows external IS (e‑Health Portal) to retrieve patient’s pharmacy from its profile information. This function supports permission delegation (see chapter 7 Permission delegation).

Service roles: patient.

Service rights: GetProfile.



Picture 25 GetProfile service

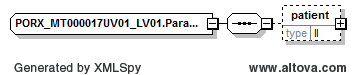
### PORX\_IN000017UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 57 Interaction PORX\_IN000017UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000017UV01\_LV01 | Payload |

#### Payload



Picture 26 PORX\_IN000017UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000017UV01\_LV01.ParameterList">

<xs:sequence>

<xs:element name="patient" type="II" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

Table 58 Payload PORX\_MT000017UV01\_LV01 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000017UV01\_LV01.QueryByParameterPayload | 1..1 | See data structure in chapter 6.12. |
| QueryByParameterPayload. parameterList | PORX\_MT000017UV01\_LV01.ParameterList | 1..1 |  |
| QueryByParameterPayload. parameterList. patient | II | 0..1 | Profile holder’s person identifier.  Personal code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1). |

### PORX\_IN000018UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 59 Interaction PORX\_IN000018UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000019UV01\_LV01 | Payload |

#### Payload

Table 60 Payload PORX\_MT000019UV01\_LV01 elements

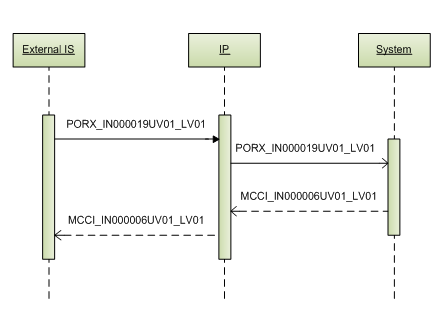
| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| profileSetupRequest | PORX\_MT000019UV01\_LV01.ProfileSetupRequest | 0..1 | See data structure in chapter 6.9. |

## SetProfile

Allows external IS (e‑Health Portal) to save patients pharmacy to its profile information. This function supports permission delegation (see chapter 7 Permission delegation).

Service roles: patient.

Service rights: SetProfile.



Picture 27 SetProfile service

### PORX\_IN000019UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 61 Interaction PORX\_IN000019UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000019UV01\_LV01 | Payload |

#### Payload

Table 62 Payload PORX\_MT000019UV01\_LV01 elements

| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| profileSetupRequest | PORX\_MT000019UV01\_LV01.ProfileSetupRequest | 1..1 | See data structure in chapter 6.9. |

### MCCI\_IN000006UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 63 Interaction MCCI\_IN000006UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

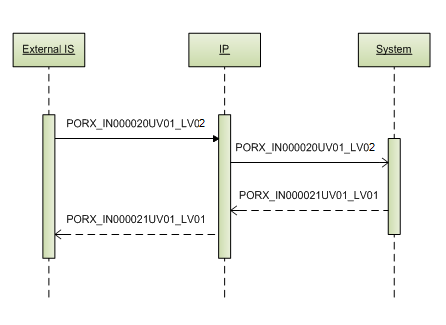
## GetPatientContactList

Allows external IS (e‑Health Portal) to retrieve list of patient contact, for whom there were prescribed medication orders with recalled medicine, by passing recalled medicine.

Service roles: supervisor institution.

Service rights: QueryMedicationOrders.

* QueryAllMedicationOrders – to get all medication orders.



Picture 28 GetPatientContactList service

### PORX\_IN000020UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 64 Interaction PORX\_IN000020UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000007UV01\_LV02 | Payload |

#### Payload

Table 65 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.13. |

### PORX\_IN000021UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 66 Interaction PORX\_IN000021UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

#### Payload (data)

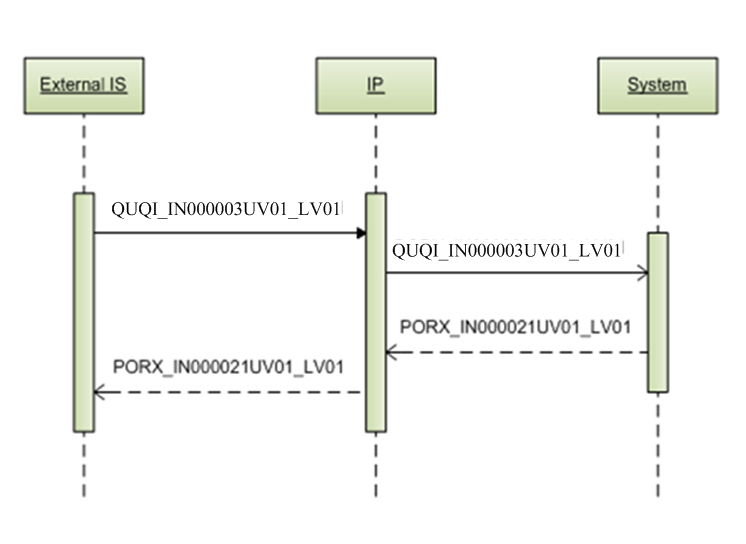
Table 67 Payload (data) elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| person | COCT\_MT050000UV01.Person | 0..unbounded | See data structure in chapter 6.2. |

## GetPatientContactListContinuation

Allows external IS (e‑Health Portal) to continue retrieving data by pages, started with service “GetPatientContactList” request.

Service roles and rights are the same as for service “GetPatientContactList”.



Picture 29 GetPatientContactListContinuation service

### QUQI\_IN000003UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 68 Interaction QUQI\_IN000003UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| QUQI\_MT000001UV01 | Trigger Event Control Act |
| QUQI\_MT000001UV01 | Payload |

#### Payload

Table 69 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryContinuation | QUQI\_MT000001UV01.QueryContinuation | 1..1 | See data structure in chapter 6.11. |

### PORX\_IN000021UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 70 Interaction PORX\_IN000021UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

#### Payload (data)

Table 71 Payload (data) elements

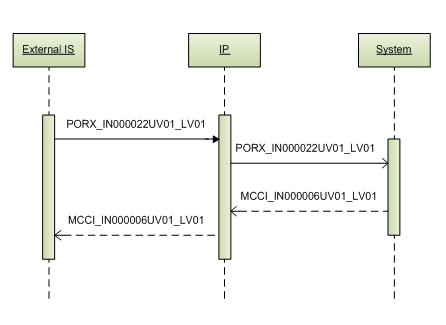
| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| person | COCT\_MT050000UV01.Person | 0..unbounded | See data structure in chapter 6.2. |

## RegisterMedicationWarning

Allows external IS (e‑Health Portal) to register medication warning, if it is dangerous or because of any other reasonable reason, requiring passing the medicine, warning message and show mode (none, show for physician only, show for physician and pharmacist).

Service roles: supervisor institution.

Service rights: RegisterMedicationWarning.



Picture 30 RegisterMedicationWarning service

### PORX\_IN000022UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 72 Interaction PORX\_IN000022UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000022UV01\_LV01 | Payload |

#### Payload

Table 73 Payload PORX\_MT000022UV01\_LV01 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| medicationWarning | PORX\_MT000022UV01\_LV01.MedicationWarning | 1..1 | See data structure in chapter 6.10. |

### MCCI\_IN000006UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 74 Interaction MCCI\_IN000006UV01\_LV01 interaction specification

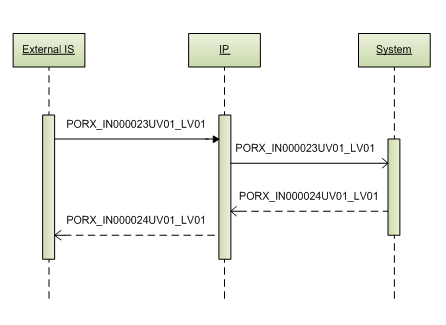
|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

## GetMedicationWarningList

Allows external IS (e‑Health Portal) to retrieve list of medication warnings for supervisor institution, allowing filtering by date active/all medicine warnings.

Service roles: supervisor institution.

Service rights: QueryMedicationWarnings.



Picture 31 GetMedicationWarningList service

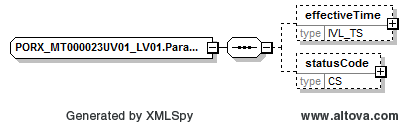
### PORX\_IN000023UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 75 Interaction PORX\_IN000023UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000023UV01\_LV01 | Payload |

#### Payload



Picture 32 PORX\_IN000023UV01\_LV01 interaction payload

Payload XML schema:

<xs:complexType name="PORX\_MT000023UV01\_LV01.ParameterList">

<xs:sequence>

<xs:element name="effectiveTime" type="IVL\_TS" minOccurs="0"/>

<xs:element name="statusCode" type="CS" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

Table 76 Payload PORX\_MT000023UV01\_LV01 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryByParameterPayload | PORX\_MT000023UV01\_LV01.QueryByParameterPayload | 1..1 | See data structure in chapter 6.12. |
| QueryByParameterPayload. parameterList | PORX\_MT000023UV01\_LV01.ParameterList | 1..1 |  |
| QueryByParameterPayload. parameterList. effectiveTime | IVL\_TS | 0..1 | Date from-to |
| QueryByParameterPayload. parameterList. statusCode | CS | 0..1 | Medication warning status (active – for active, obsolete – for obsolete) |

### PORX\_IN000024UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 77 Interaction PORX\_IN000024UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000022UV01\_LV01 | Payload |

#### Payload

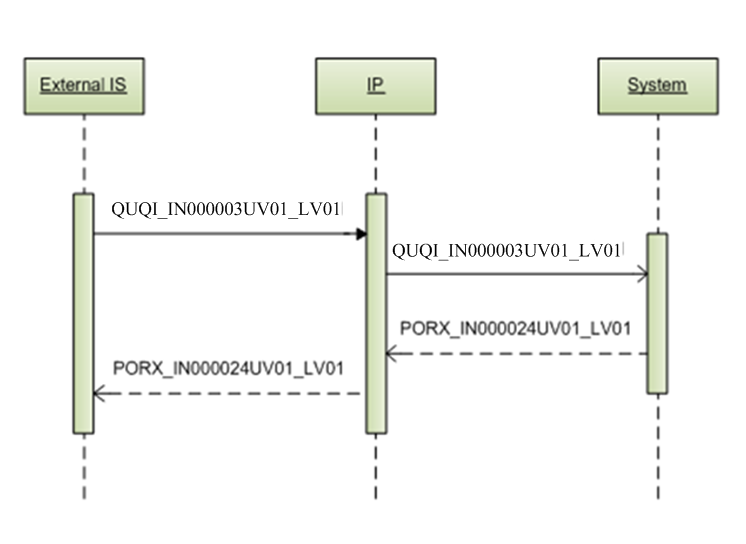
Table 78 Payload PORX\_MT000022UV01\_LV01 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| medicationWarning | PORX\_MT000022UV01\_LV01.MedicationWarning | 0..unbounded | See data structure in chapter 6.10. |

## GetMedicationWarningListContinuation

Allows external IS (e‑Health Portal) to continue retrieving data by pages, started with service “GetMedicationWarningList” request.

Service roles and rights are the same as for service “GetMedicationWarningList”.



Picture 33 GetMedicationWarningListContinuation service

### QUQI\_IN000003UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

Table 79 Interaction QUQI\_IN000003UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| QUQI\_MT000001UV01 | Trigger Event Control Act |
| QUQI\_MT000001UV01 | Payload |

#### Payload

Table 80 Payload PORX\_MT000007UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| QueryContinuation | QUQI\_MT000001UV01.QueryContinuation | 1..1 | See data structure in chapter 6.11. |

### PORX\_IN000024UV01\_LV01 interaction

This is service response interaction from System; its specification is explained in next table.

Table 81 Interaction PORX\_IN000024UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| **Identification** | **Description** |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT000022UV01\_LV01 | Payload |

#### Payload

Table 82 Payload PORX\_MT000022UV01\_LV01 elements

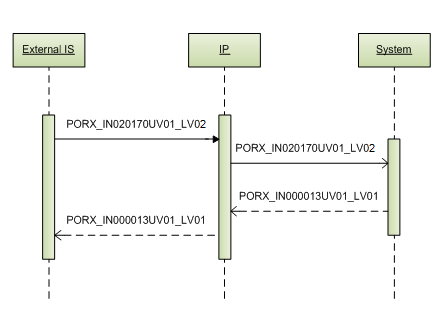
| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| medicationWarning | PORX\_MT000022UV01\_LV01.MedicationWarning | 0..unbounded | See data structure in chapter 6.10. |

## ValidateMedicationDispense

Allow external IS (e-Health Portal, Pharmacy IS) to validate medication dispense data before its registration.

Service roles: pharmacist.

Service rights: ValidateMedicationDispense.



Picture 34 ValidateMedicationDispense service

### PORX\_IN020170UV01\_LV02 interaction

This is service request interaction to System; its specification is explained in next table.

Table 83 Interaction PORX\_IN020170UV01\_LV02 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000100UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |
| PORX\_MT020070UV01\_LV02 | Payload |

#### Payload

Table 84 Payload PORX\_MT020070UV01\_LV02 elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 0..1 | See data structure in chapter 6.7. |

### MCCI\_IN000006UV01\_LV01 interaction

This is service request interaction to System; its specification is explained in next table.

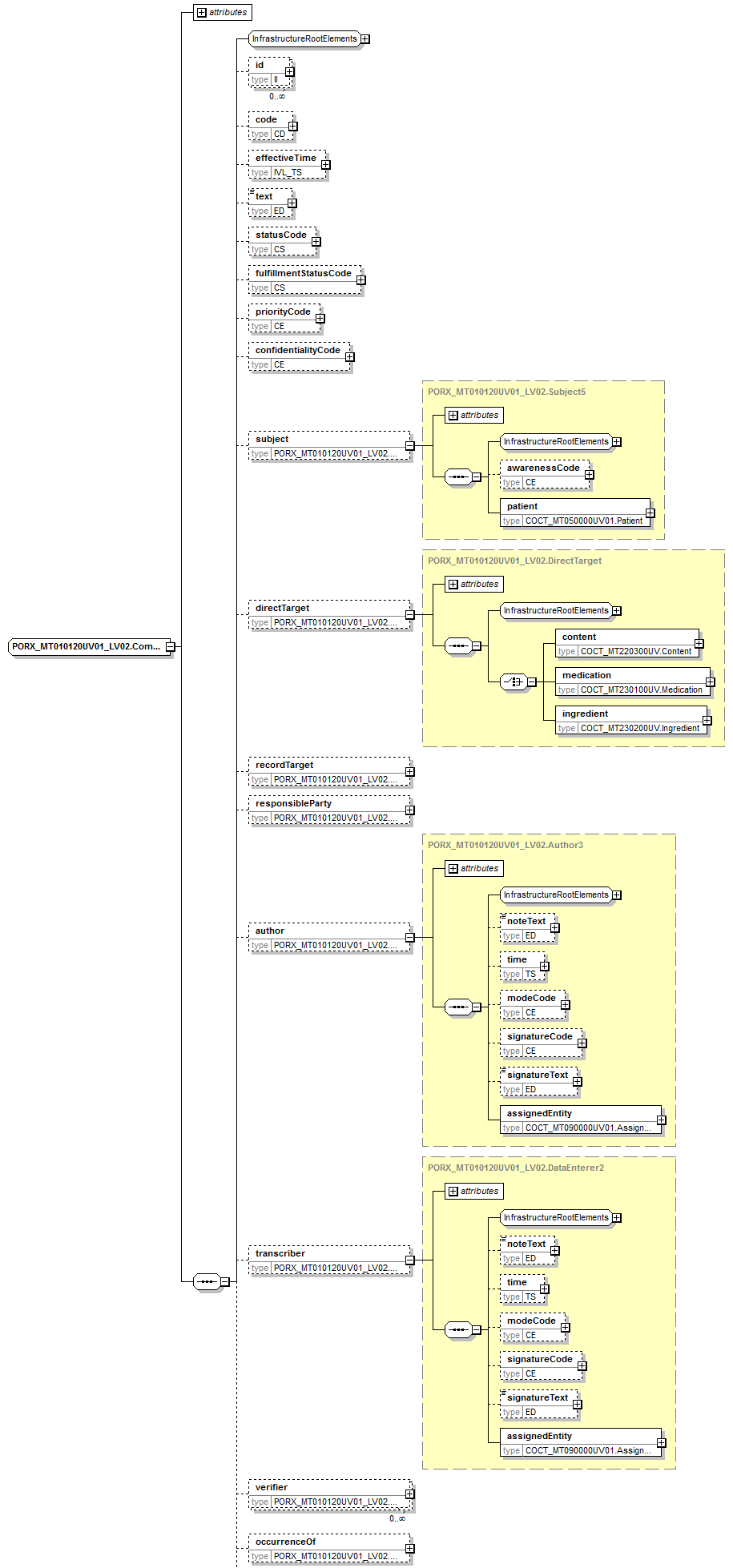
Table 85 Interaction MCCI\_IN000006UV01\_LV01 interaction specification

|  |  |
| --- | --- |
| Identification | Description |
| MCCI\_MT000200UV01\_LV01 | Transmission wrapper |
| MCAI\_MT700201UV01\_LV01 | Trigger Event Control Act |

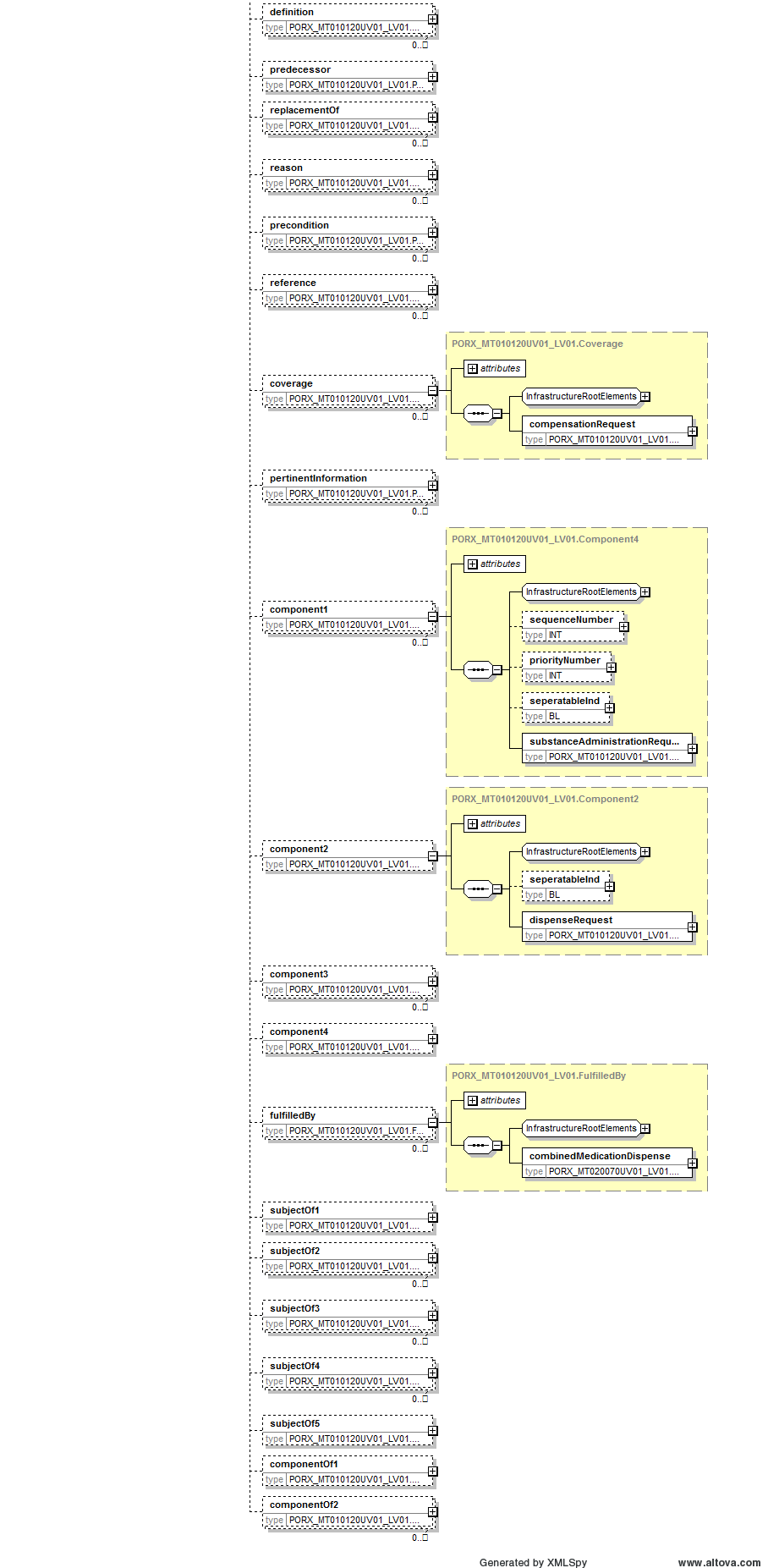
# Data structures

## PORX\_MT010120UV01\_LV02.CombinedMedicationRequest

This is complex data structure based on HL7 standard; it describes structure of medication order.



Picture 35 PORX\_MT010120UV01\_LV02.CombinedMedicationRequest data structure (part 1)



Picture 36 PORX\_MT010120UV01\_LV02.CombinedMedicationRequest data structure (part 2)

Data structure XML schema:

<xs:complexType name="PORX\_MT010120UV01\_LV02.CombinedMedicationRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="code" type="CD" minOccurs="0" maxOccurs="1"/>

<xs:element name="effectiveTime" type="IVL\_TS" minOccurs="0"/>

<xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="statusCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="fulfillmentStatusCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="priorityCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="confidentialityCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="subject" type="PORX\_MT010120UV01\_LV02.Subject5" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="directTarget" type="PORX\_MT010120UV01\_LV02.DirectTarget" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="recordTarget" type="PORX\_MT010120UV01\_LV02.RecordTarget" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="responsibleParty" type="PORX\_MT010120UV01\_LV02.ResponsibleParty" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="author" type="PORX\_MT010120UV01\_LV02.Author3" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="transcriber" type="PORX\_MT010120UV01\_LV02.DataEnterer2" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="verifier" type="PORX\_MT010120UV01\_LV02.Verifier" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="occurrenceOf" type="PORX\_MT010120UV01\_LV02.OcurrenceOf" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="definition" type="PORX\_MT010120UV01\_LV02.Definition" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="predecessor" type="PORX\_MT010120UV01\_LV02.Predecessor" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="replacementOf" type="PORX\_MT010120UV01\_LV02.ReplacementOf" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="reason" type="PORX\_MT010120UV01\_LV02.Reason" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="precondition" type="PORX\_MT010120UV01\_LV02.Precondition" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="reference" type="PORX\_MT010120UV01\_LV02.Reference" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="coverage" type="PORX\_MT010120UV01\_LV02.Coverage" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="pertinentInformation" type="PORX\_MT010120UV01\_LV02.PertinentInformation" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component1" type="PORX\_MT010120UV01\_LV02.Component4" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component2" type="PORX\_MT010120UV01\_LV02.Component2" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="component3" type="PORX\_MT010120UV01\_LV02.SourceOf" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component4" type="PORX\_MT010120UV01\_LV02.Component8" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="fulfilledBy" type="PORX\_MT010120UV01\_LV02.FulfilledBy" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf1" type="PORX\_MT010120UV01\_LV02.Subject4" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="subjectOf2" type="PORX\_MT010120UV01\_LV02.Subject2" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf3" type="PORX\_MT010120UV01\_LV02.Subject3" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf4" type="PORX\_MT010120UV01\_LV02.Subject" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf5" type="PORX\_MT010120UV01\_LV02.Subject7" nillable="true" minOccurs="0"/>

<xs:element name="componentOf1" type="PORX\_MT010120UV01\_LV02.Component5" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="componentOf2" type="PORX\_MT010120UV01\_LV02.Component6" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required"/>

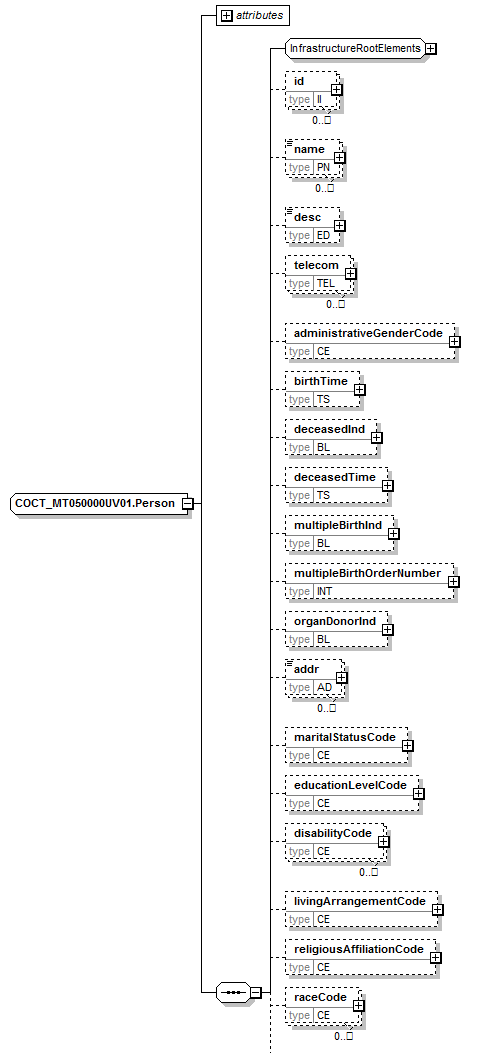
</xs:complexType>

Table 86 Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest elements

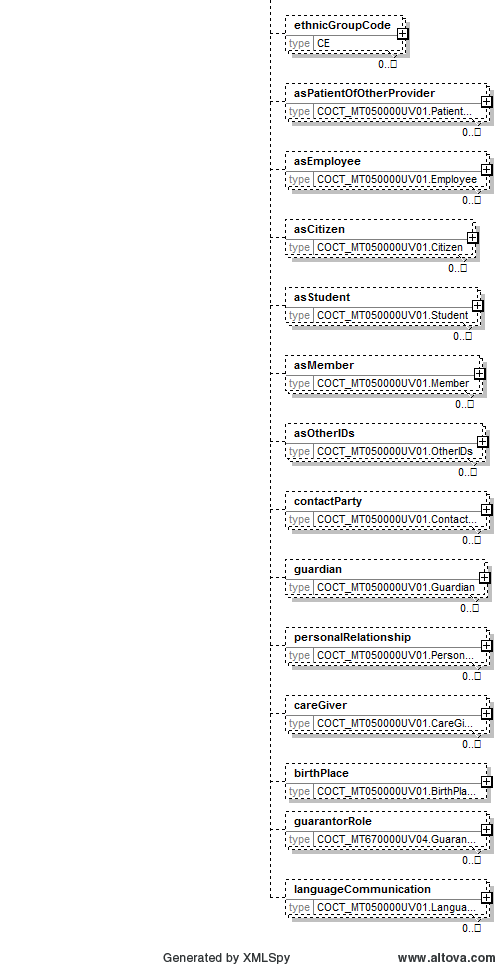
| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| Id | II | 0..unbounded | Medication order id (OID: 1.3.6.1.4.1.38760.3.4.11.1) |
| effectiveTime | IVL\_TS | 0..1 | Medication order document effective time.  For statusCode ‘new’ it represents reservation date, otherwise it represents planned date of archiving. |
| statusCode | CS | 0..1 | Medication order document status code (new – for reserved, active – for active, cancelled – for cancelled before activation,  aborted – for cancelled after activation, complete – for expired or fully dispensed) |
| fulfillmentStatusCode | CS | 0..1 | Medication otder fulfilment status code (unfulfilled, partial – partially fulfilled, fulfilled) |
| Subject | PORX\_MT010120UV01\_LV02.Subject5 | 0..1 |  |
| subject. Patient | COCT\_MT050000UV01.Patient | 1..1 |  |
| subject. patient. patientPerson | COCT\_MT050000UV01.Person | 0..1 | Patient as person.  See data structure in chapter 6.2. |
| directTarget | PORX\_MT010120UV01\_LV02.DirectTarget | 0..1 |  |
| directTarget. medication | COCT\_MT230100UV.Medication | 1..1 |  |
| directTarget. medication. administrableMedicine | COCT\_MT230100UV.Medicine | 1..1 | Prescribed medication.  See data structure in chapter 6.3. |
| author | PORX\_MT010120UV01\_LV02.Author3 | 0..1 |  |
| author. assignedEntity | COCT\_MT090000UV01.AssignedEntity | 0..1 | Physician, who prescribes medication order.  See data structure in chapter 6.4. |
| transcriber | PORX\_MT010120UV01\_LV02.DataEnterer2 | 0..1 |  |
| transcriber. noteText | ED | 0..1 | Notes or additional information. Can be used by pharmacist during import of paper medication order to System. |
| coverage | PORX\_MT010120UV01\_LV02.Coverage | 0..unbounded |  |
| coverage. compensationRequest | PORX\_MT010120UV01\_LV02.CompensationRequest | 1..1 |  |
| coverage. compensationRequest. code | CV | 0..1 | Compensation code (OID: 1.3.6.1.4.1.38760.2.152) |
| coverage. compensationRequest. percentageAmount | INT | 0..1 | Compensation amount in percentage |
| coverage. compensationRequest. substitutionReason | CV | 0..1 | For non-classified reason, attribute required:  - nullFlavor="UNC". |
| coverage. compensationRequest. substitutionReason. originalText | ED | 0..1 | Reason for substitution of INN to compensable medication. |
| coverage. compensationRequest. payer | CD | 0..1 | Potential compensation payer (sponsor). (OID: 1.3.6.1.4.1.38760.2.93).  For non-classified value, atribute required:  - nullFlavor=”UNC”. |
| component1 | PORX\_MT010120UV01\_LV02.Component4 | 0..unbounded |  |
| component1. substanceAdministrationRequest | PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest | 1..1 | Information about substance administration.  See data structure in chapter 6.5. |
| component2 | PORX\_MT010120UV01\_LV02.Component2 | 0..1 |  |
| component2. dispenseRequest | PORX\_MT010120UV01\_LV02.DispenseRequest | 1..1 | Information about dispense request.  See data structure in chapter 6.6. |
| fulfilledBy | PORX\_MT010120UV01\_LV02.FulfilledBy | 0..unbounded |  |
| fulfilledBy. combinedMedicationDispense | PORX\_MT020070UV01\_LV02.CombinedMedicationDispense | 1..1 | Medication dispenses.  See data structure in chapter 6.7. |
| subjectOf4 | PORX\_MT010120UV01\_LV02.Subject | 0..unbounded |  |
| subjectOf4. substitutionPermission | PORX\_MT010120UV01\_LV02.SubstitutionPermission | 1..1 |  |
| subjectOf4. substitutionPermission. code | CE | 1..1 | Medication substitution type code according to HL7:  - N - No substitution occurred or is permitted;  - E - equivalent (Substitution occurred or is permitted with another bioequivalent and therapeutically equivalent product). |
| subjectOf4. substitutionPermission. reasonCode | CE | 0..unbounded | For non-classified reason, attribute required:  - nullFlavor="UNC". |
| subjectOf4. substitutionPermission. reasonCode. originalText | ED | 0..1 | Substitution prohibition reason grounds. |
| subjectOf5 | PORX\_MT010120UV01\_LV02.Subject7 | 0..1 |  |
| subjectOf5. cancelMedicationOrderRequest | PORX\_MT000025UV01\_LV02.CancelMedicationOrderRequest | 1..1 | Cancellation information. Fulfilled on cancellation of medication order.  See data structure in chapter 6.8. |

## COCT\_MT050000UV01.Person

This is complex data structure based on HL7 standard; it describes structure of person.



Picture 37 COCT\_MT050000UV01.Person data structure (part 1)



Picture 38 COCT\_MT050000UV01.Person data structure (part 2)

Data structure XML schema:

<xs:complexType name="COCT\_MT050000UV01.Person">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="name" type="PN" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="desc" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="administrativeGenderCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="birthTime" type="TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="deceasedInd" type="BL" minOccurs="0" maxOccurs="1"/>

<xs:element name="deceasedTime" type="TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="multipleBirthInd" type="BL" minOccurs="0" maxOccurs="1"/>

<xs:element name="multipleBirthOrderNumber" type="INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="organDonorInd" type="BL" minOccurs="0" maxOccurs="1"/>

<xs:element name="addr" type="AD" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="maritalStatusCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="educationLevelCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="disabilityCode" type="CE" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="livingArrangementCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="religiousAffiliationCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="raceCode" type="CE" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ethnicGroupCode" type="CE" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asPatientOfOtherProvider" type="COCT\_MT050000UV01.PatientOfOtherProvider" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asEmployee" type="COCT\_MT050000UV01.Employee" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asCitizen" type="COCT\_MT050000UV01.Citizen" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asStudent" type="COCT\_MT050000UV01.Student" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asMember" type="COCT\_MT050000UV01.Member" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asOtherIDs" type="COCT\_MT050000UV01.OtherIDs" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="contactParty" type="COCT\_MT050000UV01.ContactParty" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="guardian" type="COCT\_MT050000UV01.Guardian" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="personalRelationship" type="COCT\_MT050000UV01.PersonalRelationship" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="careGiver" type="COCT\_MT050000UV01.CareGiver" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="birthPlace" type="COCT\_MT050000UV01.BirthPlace" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="guarantorRole" type="COCT\_MT670000UV04.GuarantorRole" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="languageCommunication" type="COCT\_MT050000UV01.LanguageCommunication" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="EntityClass" use="optional" fixed="PSN"/>

<xs:attribute name="determinerCode" type="EntityDeterminer" use="optional" fixed="INSTANCE"/>

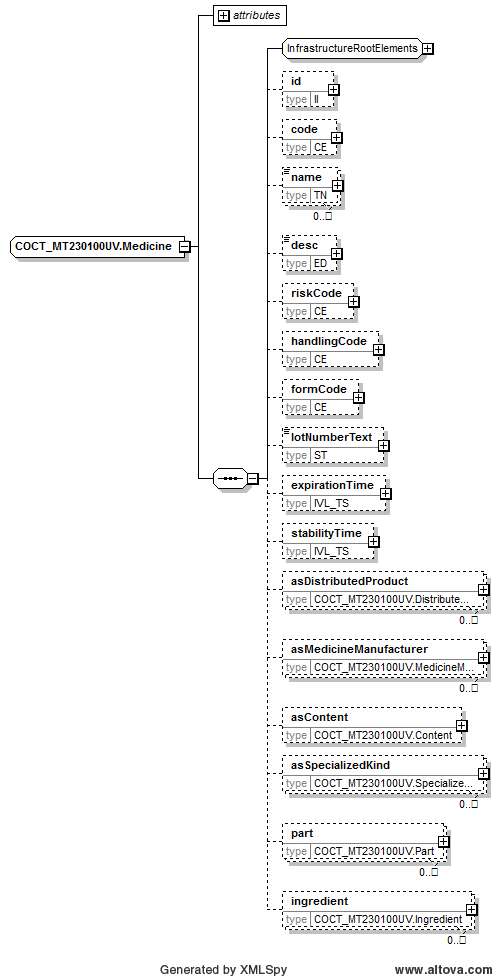
</xs:complexType>

Table 87 Data structure COCT\_MT050000UV01.Person elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| id | II | 0..unbounded | Person identifier.  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1).  Alternative identifier for newborns – consists of mother’s personal code and child birth time (OID: 1.3.6.1.4.1.38760.3.1.3).  Identification number for foreign country citizens (any child node in OID namespace 1.3.6.1.4.1.38760.3.1.8 except 1.3.6.1.4.1.38760.3.1.8.60).  EHIC card number can be added also (OID: 1.3.6.1.4.1.38760.3.3.1.3). |
| name | PN | 0..unbounded | Person name as text with added logical mark-up. The approach is similar to HTML or XML markup of text. |
| name. family | en.family | 0..unbouned | Logical mark-up of family name |
| name. given | en.given | 0..unbouned | Logical mark-up of given name |
| administrativeGenderCode | CE | 0..1 | Gender (OID: 1.3.6.1.4.1.38760.2.111) |
| birthTime | TS | 0..1 | Birth date and time |
| addr | AD | 0..unbounded | Address as text with added logical mark-up. The approach is similar to HTML or XML markup of text.  It is not mandatory, but very preferable because address data will be used in further data analysis. |
| addr. delimiter | adxp.delimiter | 0..unbounded | Logical mark-up of delimiter. |
| addr. country | adxp.country | 0..unbounded | Logical mark-up of country code (Text value can be taken from classifier with OID: 1.3.6.1.4.1.38760.2.7) |
| addr. postalCode | adxp.postalCode | 0..unbounded | Logical mark-up of postal code. |
| addr. county | adxp.county | 0..unbounded | Logical mark-up of county, parish and rural area (Text values can be taken from classifier with OID: 1.3.6.1.4.1.38760.2.113 for county; 1.3.6.1.4.1.38760.2.114 for parish and rural area) |
| addr. city | adxp.city | 0..unbounded | Logical mark-up of city or village (Text value can be taken from classifier with OID: 1.3.6.1.4.1.38760.2.116 for any city, 1.3.6.1.4.1.38760.2.114 for county city, 1.3.6.1.4.1.38760.2.8 for republic cities) |
| addr. streetName | adxp.streetName | 0..unbounded | Logical mark-up of street name (Text value can be taken from classifier with OID: 1.3.6.1.4.1.38760.2.9) |
| addr. houseNumber | adxp.houseNumber | 0..unbounded | Logical mark-up of house number. |
| addr. additionalLocator | adxp.additionalLocator | 0..unbounded | Logical mark-up of house name. |
| addr. unitID | adxp.unitID | 0..unbounded | Logical mark-up of flat number |
| addr. censusTract | adxp.censusTract | 0..unbounded | Logical mark-up of ATU code.  Must be one of these codes:  ATU Code 1 (OID: 1.3.6.1.4.1.38760.2.8).  ATU Code 2 (OID: 1.3.6.1.4.1.38760.2.113).  ATU Code 3 (OID: 1.3.6.1.4.1.38760.2.114) |
| telecom | TEL | 0.. unbounded | Phone number.  According to HL7, value must be set in URL format:  - tel:<phone number>;  - mailto:<e-mail address>. |

## COCT\_MT230100UV.Medicine

This is complex data structure based on HL7 standard; it describes structure of medicine.



Picture 39 COCT\_MT230100UV.Medicine data structure

Data structure XML schema:

<xs:complexType name="COCT\_MT230100UV.Medicine">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="code" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="name" type="TN" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="desc" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="riskCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="handlingCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="formCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="lotNumberText" type="ST" minOccurs="0" maxOccurs="1"/>

<xs:element name="expirationTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="stabilityTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="asDistributedProduct" type="COCT\_MT230100UV.DistributedProduct" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asMedicineManufacturer" type="COCT\_MT230100UV.MedicineManufacturer" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="asContent" type="COCT\_MT230100UV.Content" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="asSpecializedKind" type="COCT\_MT230100UV.SpecializedKind" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="part" type="COCT\_MT230100UV.Part" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ingredient" type="COCT\_MT230100UV.Ingredient" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="EntityClassManufacturedMaterial" use="required"/>

<xs:attribute name="determinerCode" type="EntityDeterminer" use="required"/>

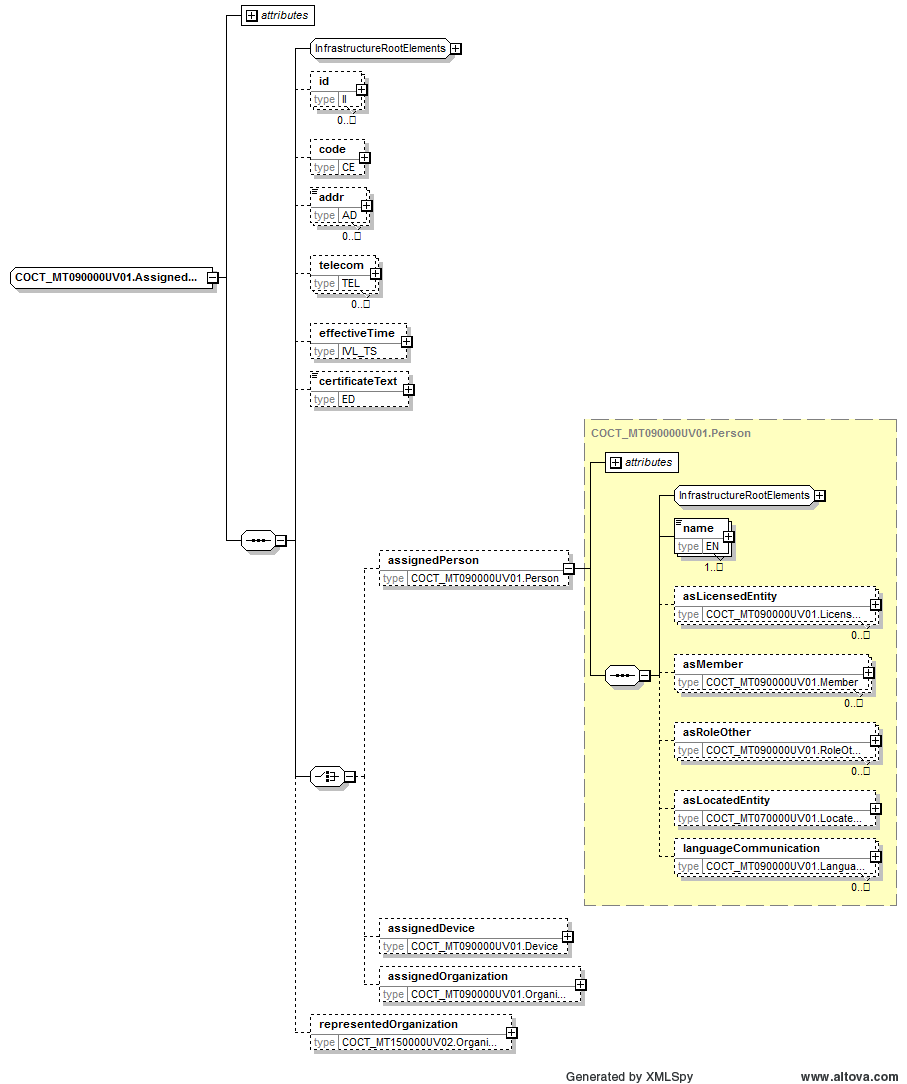
</xs:complexType>

Table 88 Data structure COCT\_MT230100UV.Medicine elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| code | CE | 0..1 | Medical registration number (OID: 1.3.6.1.4.1.38760.2.136), compensation group (OID: 1.3.6.1.4.1.38760.2.177) or compensable medicine (OID: 1.3.6.1.4.1.38760.2.151) |
| name | TN | 0..unbounded | Medicine name.  For compensation group and medicine in free text format, medicine name must be supplemented with medicine strength. |
| desc | ED | 0..1 | Full combined medicine name, including strength, form etc. |
| formCode | CE | 0..1 | Medicine form code (OID: 1.3.6.1.4.1.38760.2.137) |
| ingredient | COCT\_MT230100UV.Ingredient | 0..unbounded |  |
| ingredient. ingredientSubstance | COCT\_MT230100UV.Substance | 0..1 |  |
| ingredient. ingredientSubstance. code | CE | 0..1 | Active substance code (OID: 1.3.6.1.4.1.38760.2.140) |
| ingredient. quantity | RTO | 0..1 | Amount of active substance.  For non-classified value, attribute required:  - nullFlavor="UNC". |
| ingredient. quantity. numerator | PQ | 1..1 | For non-classified value, attribute required:  - nullFlavor="UNC". Use sub-element translation.originalText to fill the value. For sub-element translation, for non-classifiend value, attribute required:  - nullFlavor="UNC". |
| ingredient. quantity. denominator | PQ | 1..1 | Value = 1. |

## COCT\_MT090000UV01.AssignedEntity

This is complex data structure based on HL7 standard; it describes structure of author as assigned entity.



Picture 40 COCT\_MT090000UV01.AssignedEntity data structure

Data structure XML schema:

<xs:complexType name="COCT\_MT090000UV01.AssignedEntity">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="code" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="addr" type="AD" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="effectiveTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="certificateText" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:choice>

<xs:element name="assignedPerson" type="COCT\_MT090000UV01.Person" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="assignedDevice" type="COCT\_MT090000UV01.Device" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="assignedOrganization" type="COCT\_MT090000UV01.Organization" nillable="true" minOccurs="0" maxOccurs="1"/>

</xs:choice>

<xs:element name="representedOrganization" type="COCT\_MT150000UV02.Organization" nillable="true" minOccurs="0" maxOccurs="1"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="RoleClassAssignedEntity" use="required"/>

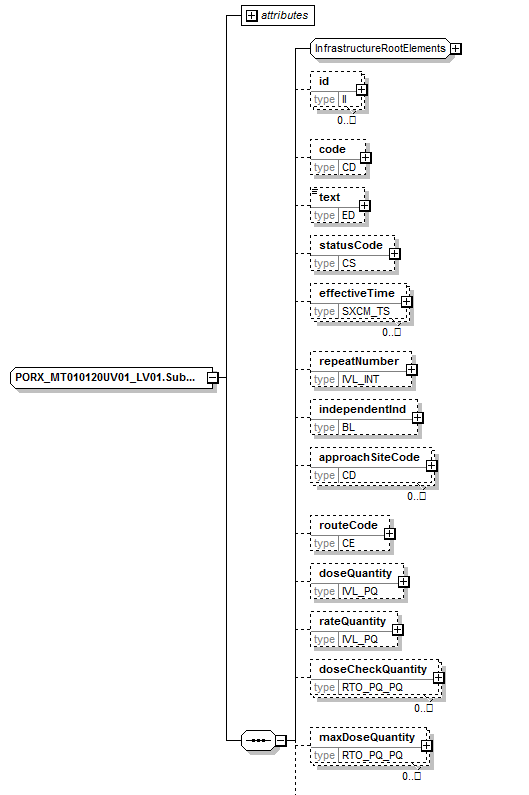
</xs:complexType>

Table 89 Data structure COCT\_MT090000UV01.AssignedEntity elements

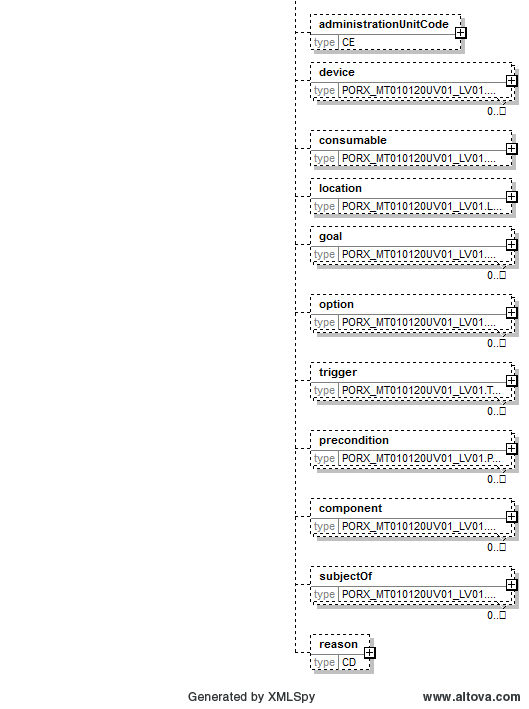
| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| id | II | 0..unbounded | Person code or other identifying code (for example, physician code)  Physician (OID: 1.3.6.1.4.1.38760.2.1).  Physician code (OID: 1.3.6.1.4.1.38760.3.1.4).  Pharmacist (OID: 1.3.6.1.4.1.38760.2.46).  Pharmacist code (OID: 1.3.6.1.4.1.38760.3.1.5).  Person code for Latvian citizen – for both physician and pharmacist (OID: 1.3.6.1.4.1.38760.3.1.1) |
| assignedPerson | COCT\_MT090000UV01.Person | 0..1 |  |
| assignedPerson. name | EN | 1..unbounded | Person name as text with added logical mark-up. The approach is similar to HTML or XML markup of text. |
| assignedPerson. name. family | en.family | 0..unbounded | Logical mark-up of family name |
| assignedPerson. name. given | en.given | 0..unbounded | Logical mark-up of given name |
| assignedPerson. asLicensedEntity | COCT\_MT090000UV01.LicensedEntity | 0..unbounded |  |
| assignedPerson. asLicensedEntity. code | CE | 0..1 | Specialty code and name.  Physician specialty (OID: 1.3.6.1.4.1.38760.2.38).  Pharmacist specialty (OID: 1.3.6.1.4.1.38760.2.47). |
| representedOrganization. id | II | 1..unbounded | Physician or pharmacy organization code or branch code.  Medical institution code (OID: 1.3.6.1.4.1.38760.2.23).  Medical institution branch code (OID: 1.3.6.1.4.1.38760.2.28).  Pharmacy code (OID: 1.3.6.1.4.1.38760.2.134). |
| representedOrganization. name | ON | 0..unbounded | Organization name |
| representedOrganization. addr | AD | 0..1 | Organization address as text with added logical mark-up, if necessary. The approach is similar to HTML or XML markup of text. |
| representedOrganization. telecom | TEL | 0..unbounded | Organization phone number.  For medication order physician only.  According to HL7, value must be set in URL format:  - tel:<phone number>;  - mailto:<e-mail address>. |

## PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest

This is complex data structure based on HL7 standard; it describes structure of substance administration request.



Picture 41 PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest data structure (part 1)



Picture 42 PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest data structure (part 2)

Data structure XML schema:

<xs:complexType name="PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="code" type="CD" minOccurs="0" maxOccurs="1"/>

<xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="statusCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="effectiveTime" type="SXCM\_TS" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="repeatNumber" type="IVL\_INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="independentInd" type="BL" minOccurs="0" maxOccurs="1"/>

<xs:element name="approachSiteCode" type="CD" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="routeCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="doseQuantity" type="IVL\_PQ" minOccurs="0" maxOccurs="1"/>

<xs:element name="rateQuantity" type="IVL\_PQ" minOccurs="0" maxOccurs="1"/>

<xs:element name="doseCheckQuantity" type="RTO\_PQ\_PQ" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="maxDoseQuantity" type="RTO\_PQ\_PQ" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="administrationUnitCode" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="device" type="PORX\_MT010120UV01\_LV02.Device" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="consumable" type="PORX\_MT010120UV01\_LV02.Consumable" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="location" type="PORX\_MT010120UV01\_LV02.Location1" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="goal" type="PORX\_MT010120UV01\_LV02.Goal" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="option" type="PORX\_MT010120UV01\_LV02.Option" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="trigger" type="PORX\_MT010120UV01\_LV02.Trigger" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="precondition" type="PORX\_MT010120UV01\_LV02.Precondition2" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component" type="PORX\_MT010120UV01\_LV02.Component7" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf" type="PORX\_MT010120UV01\_LV02.Subject1" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="reason" type="CD" minOccurs="0"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClassSubstanceAdministration" use="required"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required"/>

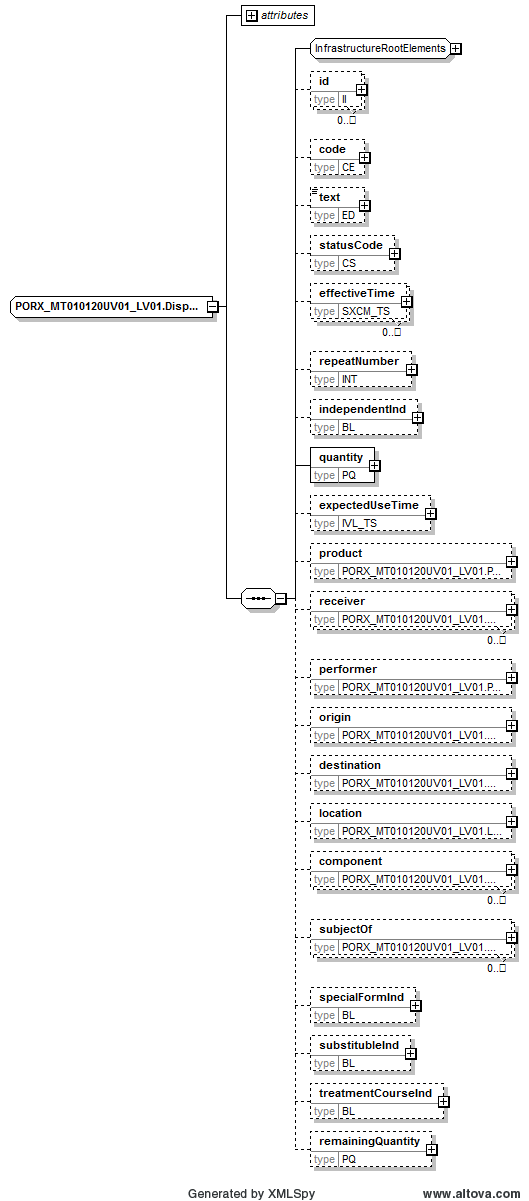
</xs:complexType>

Table 90 Data structure PORX\_MT010120UV01\_LV02.SubstanceAdministrationRequest elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| text | ED | 0..1 | Administration recommendation (*signatura*) |
| effectiveTime | SXCM\_TS | 0..unbounded | Length of treatment course |
| reason | CD | 0..1 | Diagnosis and additional diagnosis ICD-10 code (OID: 1.3.6.1.4.1.38760.2.159) |

## PORX\_MT010120UV01\_LV02.DispenseRequest

This is complex data structure based on HL7 standard; it describes structure of medication order dispense request.



Picture 43 PORX\_MT010120UV01\_LV02.DispenseRequest data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT010120UV01\_LV02.DispenseRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="code" type="CE" minOccurs="0" maxOccurs="1"/>

<xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="statusCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="effectiveTime" type="SXCM\_TS" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="repeatNumber" type="INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="independentInd" type="BL" minOccurs="0" maxOccurs="1"/>

<xs:element name="quantity" type="PQ" minOccurs="1" maxOccurs="1"/>

<xs:element name="expectedUseTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="product" type="PORX\_MT010120UV01\_LV02.Product" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="receiver" type="PORX\_MT010120UV01\_LV02.Receiver" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="performer" type="PORX\_MT010120UV01\_LV02.Performer" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="origin" type="PORX\_MT010120UV01\_LV02.Origin" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="destination" type="PORX\_MT010120UV01\_LV02.Destination" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="location" type="PORX\_MT010120UV01\_LV02.Location2" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="component" type="PORX\_MT010120UV01\_LV02.Component3" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf" type="PORX\_MT010120UV01\_LV02.Subject6" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="specialFormInd" type="BL" minOccurs="0"/>

<xs:element name="treatmentCourseInd" type="BL" minOccurs="0"/>

<xs:element name="remainingQuantity" type="PQ" minOccurs="0"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClassSupply" use="required"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required"/>

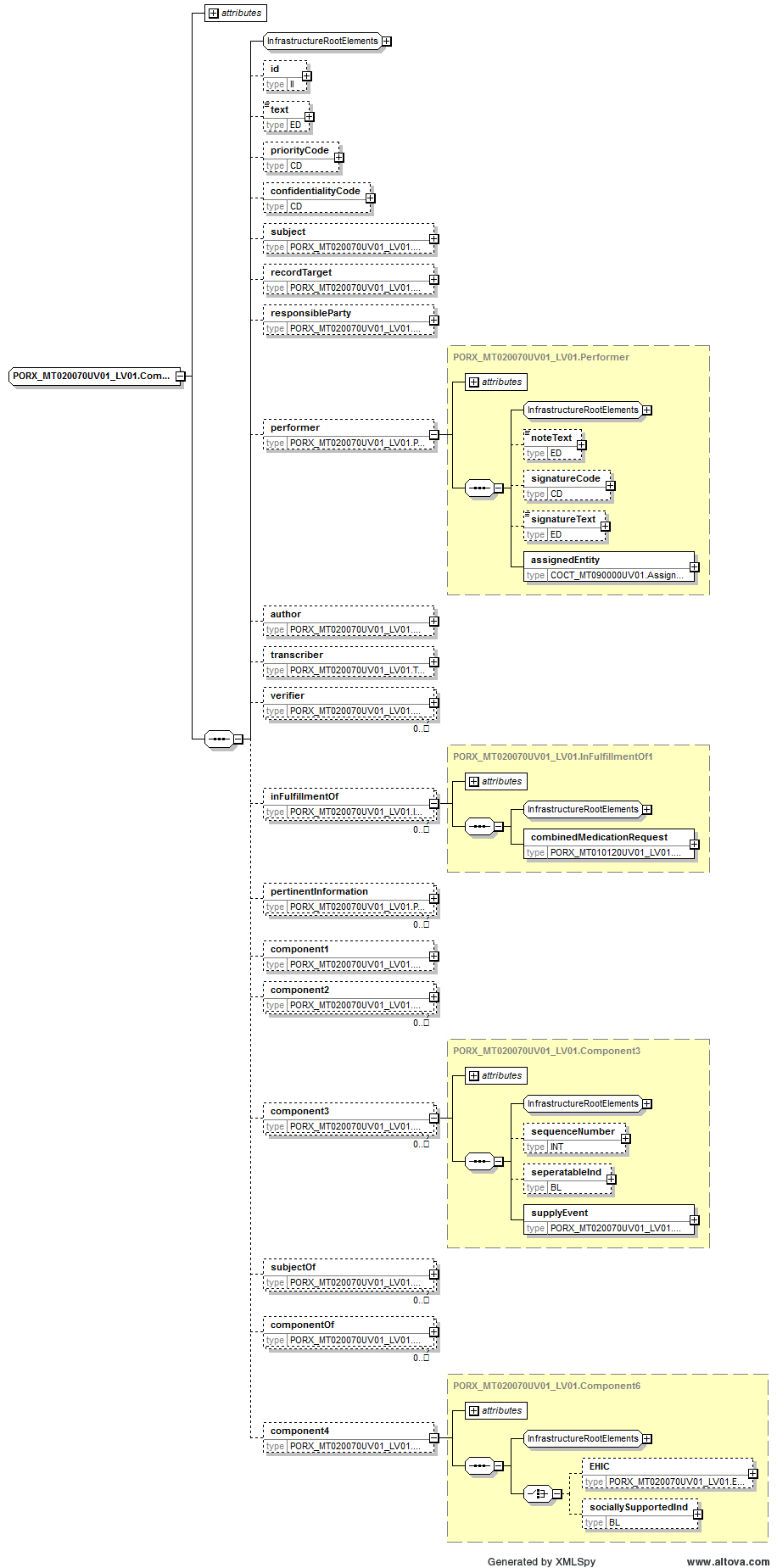
</xs:complexType>

Table 91 Data structure PORX\_MT010120UV01\_LV02.DispenseRequest elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| id | II | 0..unbounded | Medication order paper identifier (form number) (OID: 1.3.6.1.4.1.38760.3.4.11.2) |
| text | ED | 0..1 | Special dispense requirements |
| effectiveTime | SXCM\_TS | 0..unbounded | Medication order prescription time and validity time. System expects single IVL\_TS type element. |
| quantity | PQ | 1..1 | Amount of prescribed medication. |
| receiver | PORX\_MT010120UV01\_LV02.Receiver | 0..unbounded |  |
| receiver. assignedPerson | COCT\_MT090100UV01.AssignedPerson | 1..1 |  |
| receiver. assignedPerson. id | II | 0..unbounded | Person identifier for delegated.  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1).  Identification number for foreign country citizens (any child node in OID namespace 1.3.6.1.4.1.38760.3.1.8 except 1.3.6.1.4.1.38760.3.1.8.60). |
| receiver. assignedPerson. assignedPerson | COCT\_MT090000UV01.Person | 0..1 |  |
| receiver. assignedPerson. assignedPerson. name | EN | 1..unbounded | Person name as text with added logical mark-up. The approach is similar to HTML or XML markup of text. |
| receiver. assignedPerson. assignedPerson. name. family | en.family | 0..unbounded | Logical mark-up of family name. |
| receiver. assignedPerson. assignedPerson. name. given | en.given | 0..unbounded | Logical mark-up of given name. |
| specialFormInd | BL | 0..1 | Indicator: special paper form (narcotic drugs and psychotropic, compensated medication) |
| treatmentCourseInd | BL | 0..1 | Indicator: medication order is for treatment course, which is longer than 3 months |
| remainingQuantity | PQ | 0..1 | Remaining amount of medication.  Calculated based on prescribed and dispensed amount of medication. |

## PORX\_MT020070UV01\_LV02.CombinedMedicationDispense

This is complex data structure based on HL7 standard; it describes structure of medication dispense.



Picture 44 PORX\_MT020070UV01\_LV02.CombinedMedicationDispense data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT020070UV01\_LV02.CombinedMedicationDispense">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>

<xs:element name="priorityCode" type="CD" minOccurs="0" maxOccurs="1"/>

<xs:element name="confidentialityCode" type="CD" minOccurs="0" maxOccurs="1"/>

<xs:element name="subject" type="PORX\_MT020070UV01\_LV02.Subject4" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="recordTarget" type="PORX\_MT020070UV01\_LV02.RecordTarget" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="responsibleParty" type="PORX\_MT020070UV01\_LV02.ResponsibleParty2" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="performer" type="PORX\_MT020070UV01\_LV02.Performer" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="author" type="PORX\_MT020070UV01\_LV02.Author2" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="transcriber" type="PORX\_MT020070UV01\_LV02.Transcriber" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="verifier" type="PORX\_MT020070UV01\_LV02.Verifier" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="inFulfillmentOf" type="PORX\_MT020070UV01\_LV02.InFulfillmentOf1" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="pertinentInformation" type="PORX\_MT020070UV01\_LV02.PertinentInformation" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component1" type="PORX\_MT020070UV01\_LV02.Component1" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="component2" type="PORX\_MT020070UV01\_LV02.Component5" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component3" type="PORX\_MT020070UV01\_LV02.Component3" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="subjectOf" type="PORX\_MT020070UV01\_LV02.Subject3" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="componentOf" type="PORX\_MT020070UV01\_LV02.Component2" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="component4" type="PORX\_MT020070UV01\_LV02.Component6" nillable="true" minOccurs="0"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMood" use="optional" fixed="EVN"/>

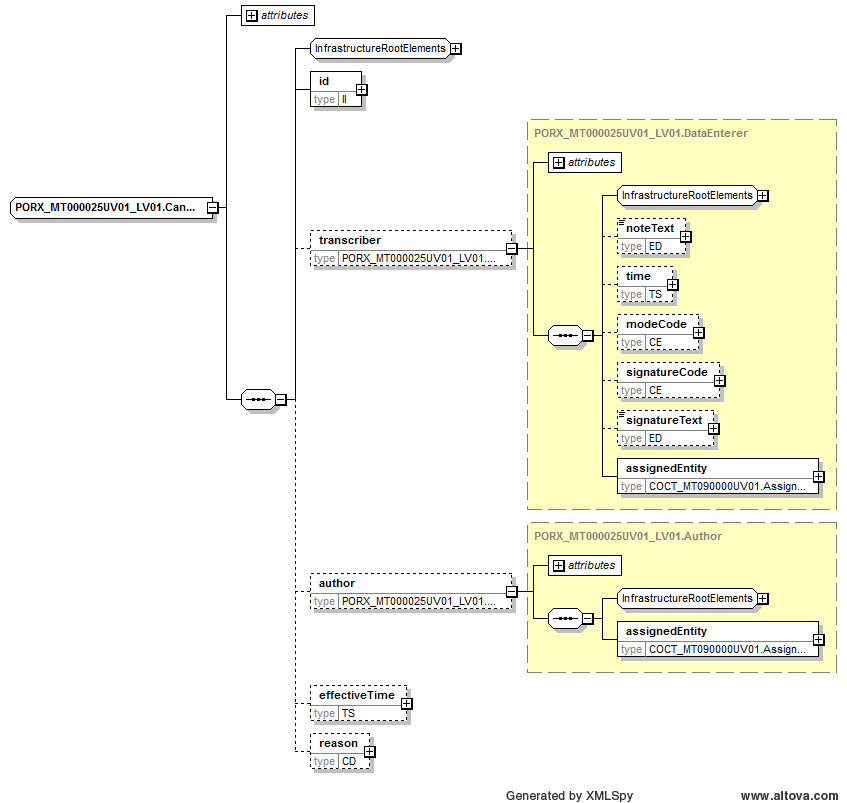
</xs:complexType>

Table 92 Data structure PORX\_MT020070UV01\_LV02.CombinedMedicationDispense elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| id | II | 0..1 | Medication dispense identifier (OID: 1.3.6.1.4.1.38760.3.4.11.3) |
| performer | PORX\_MT020070UV01\_LV02.Performer | 0..1 |  |
| performer. noteText | ED | 0..1 | Pharmacist notes |
| performer. assignedEntity | COCT\_MT090000UV01.AssignedEntity | 1..1 | See data structure in chapter 6.4. |
| inFulfillmentOf | PORX\_MT020070UV01\_LV02.InFulfillmentOf1 | 0..unbounded |  |
| inFulfillmentOf. combinedMedicationRequest | PORX\_MT010120UV01\_LV02.CombinedMedicationRequest | 1..1 |  |
| inFulfillmentOf. combinedMedicationRequest. id | II | 0..unbounded | Medication order identifier (OID: 1.3.6.1.4.1.38760.3.4.11.1) |
| inFulfillmentOf. combinedMedicationRequest. statusCode | CD | 0..1 | Medication order status code (new – for reserved, active – for active, cancelled – for cancelled, complete – for fully dispensed). |
| component1 | PORX\_MT020070UV01\_LV02.Component1 | 0..1 |  |
| component1. substitutionMade | PORX\_MT020070UV01\_LV02.SubstitutionMade | 1..1 |  |
| component1. substitutionMade. code | CD | 1..1 | Medication substitution type code according to HL7:  - N - No substitution occurred or is permitted;  - E - equivalent (Substitution occurred or is permitted with another bioequivalent and therapeutically equivalent product). |
| component1. substitutionMade. reasonCode | CD | 0..unbounded | For non-classified reason, attribute required:  - nullFlavor="UNC". |
| component1. substitutionMade. reasonCode. originalText | ED | 0..1 | Substitution reason grounds. |
| component3 | PORX\_MT020070UV01\_LV02.Component3 | 0..unbounded |  |
| component3. supplyEvent | PORX\_MT020070UV01\_LV02.SupplyEvent | 1..1 |  |
| component3. supplyEvent. effectiveTime | IVL\_TS | 0..1 | Dispense time |
| component3. supplyEvent. quantity | PQ | 1..1 | Dispensed amount of medication, including amount of packages. |
| component3. supplyEvent. consumable | PORX\_MT020070UV01\_LV02.Consumable1 | 0..unbounded |  |
| component3. supplyEvent. consumable. content | COCT\_MT220300UV.Content | 1..1 |  |
| component3. supplyEvent. consumable. content. containedMedicine | COCT\_MT220300UV.Medicine | 1..1 |  |
| component3. supplyEvent. consumable. content. containedMedicine. Code | CE | 0..1 | Dispensed medication code (OID: 1.3.6.1.4.1.38760.2.144) |
| component3. supplyEvent. consumable. content. containedMedicine. Name | TN | 0..unbounded | Dispensed medication name |
| component3. supplyEvent. receiver | PORX\_MT010120UV01\_LV02.Receiver | 0..unbounded |  |
| component3. supplyEvent. receiver. assignedPerson | COCT\_MT090100UV01.AssignedPerson | 1..1 |  |
| component3. supplyEvent. receiver. assignedPerson. id | II | 0..unbounded | Person identifier for delegated.  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1).  Identification number for foreign country citizens (any child node in OID namespace 1.3.6.1.4.1.38760.3.1.8 except 1.3.6.1.4.1.38760.3.1.8.60).  Identification number for person document type (OID):  - 1.3.6.1.4.1.38760.3.3.1 – passport;  1.3.6.1.4.1.38760.3.3.5 – ID card. |
| component3. supplyEvent. receiver. assignedPerson. assignedPerson | COCT\_MT090000UV01.Person | 0..1 |  |
| component3. supplyEvent. receiver. assignedPerson. assignedPerson. name | EN | 1..unbounded | Person name as text with added logical mark-up. The approach is similar to HTML or XML markup of text. |
| component3. supplyEvent. receiver. assignedPerson. assignedPerson. name. family | en.family | 0..unbounded | Logical mark-up of family name. |
| component3. supplyEvent. receiver. assignedPerson. assignedPerson. name. given | en.given | 0..unbounded | Logical mark-up of given name. |
| component3. supplyEvent. price | MO | 0..1 | Price of medication package or unit |
| component3. supplyEvent. totalAmount | MO | 0..1 | Total amount of money for medication to pay |
| component3. supplyEvent. paymentAmount | MO | 0..1 | Amount of money for patient to pay |
| component3. supplyEvent. compensatedAmount | MO | 0..1 | Compensated amount of money |
| component3. supplyEvent. payer | CD | 0..1 | Compensation payer sponsor (OID: 1.3.6.1.4.1.38760.2.93).  For non-classified value, atribute required:  - nullFlavor=”UNC”. |
| Component3. supplyEvent. compensationPercent | INT | 0..1 | Amount of compensation in percentage |
| component4 | PORX\_MT020070UV01\_LV02.Component6 | 0..1 |  |
| component4. sociallySupportedInd | BL | 0..1 | Indicator: patient is socially supported |
| component4. EHIC | PORX\_MT020070UV01\_LV02.EHIC | 0..1 |  |
| component4. EHIC. id | II | 1..1 | EHIC identifier (OID: 1.3.6.1.4.1.38760.3.3.1.3)  Replacement certificate identifier (OID: 1.3.6.1.4.1.38760.3.3.1.29) |
| component4. EHIC. issueTime | TS | 0..1 | Replacement certificate issue time |
| component4. EHIC. effectiveTime | IVL\_TS | 0..1 | Replacement certificate validity (from - to) |

## PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest

This is complex data structure based on HL7 standard; it describes structure of medication order cancellation request.



Picture 45 PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="id" type="II" maxOccurs="unbounded"/>

<xs:element name="transcriber" type="PORX\_MT000025UV01\_LV01.DataEnterer" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="author" type="PORX\_MT000025UV01\_LV01.Author" nillable="true" minOccurs="0"/>

<xs:element name="effectiveTime" type="TS" minOccurs="0"/>

<xs:element name="reason" type="CD" minOccurs="0"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMood" use="optional" fixed="RQO"/>

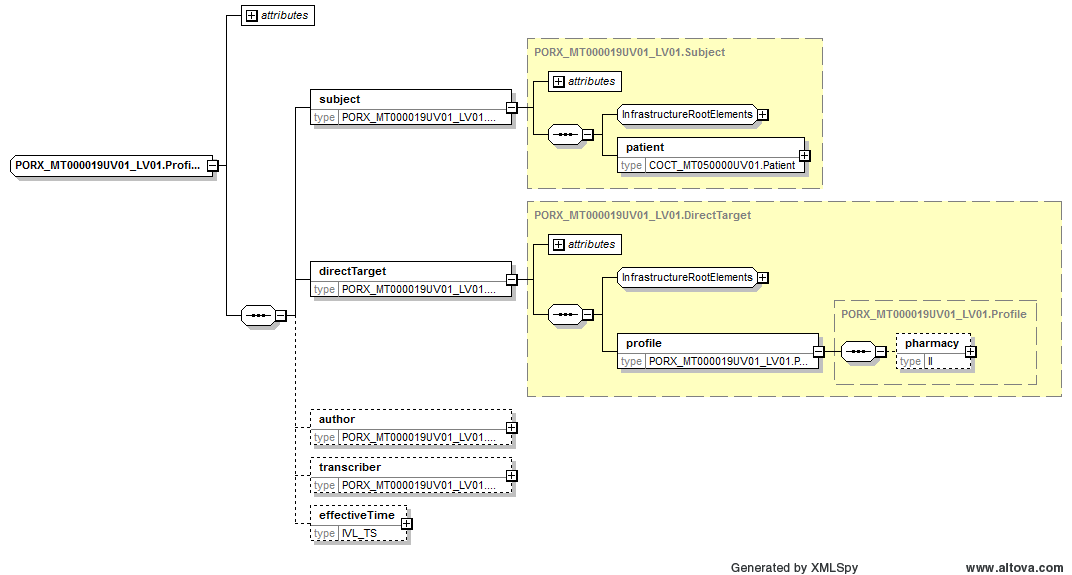
</xs:complexType>

Table 93 Data structure PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest elements

| **Element** | **Type** | **Cardinality** | **Description** |
| --- | --- | --- | --- |
| id | II | 0..unbounded |  |
| author | PORX\_MT000025UV01\_LV01.Author | 0..1 |  |
| author. assignedEntity | COCT\_MT090000UV01.AssignedEntity | 1..1 |  |
| author. assignedEntity. id | II | 0..unbounded | Person code of person, who cancelled medication order.  Physician (OID: 1.3.6.1.4.1.38760.2.1).  Physician code (OID: 1.3.6.1.4.1.38760.3.1.4).  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1) |
| effectiveTime | TS | 1..1 | Cancellation time |
| reason | CD | 1..1 | Cancelation reason (OID: 1.3.6.1.4.1.38760.2.300) |

## PORX\_MT000019UV01\_LV01.ProfileSetupRequest

This is complex data structure based on HL7 standard; it describes structure of user profile.



Picture 46 PORX\_MT000019UV01\_LV01.ProfileSetupRequest data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT000019UV01\_LV01.ProfileSetupRequest">

<xs:sequence>

<xs:element name="subject" type="PORX\_MT000019UV01\_LV01.Subject"/>

<xs:element name="directTarget" type="PORX\_MT000019UV01\_LV01.DirectTarget"/>

<xs:element name="author" type="PORX\_MT000019UV01\_LV01.Author" minOccurs="0"/>

<xs:element name="transcriber" type="PORX\_MT000019UV01\_LV01.DataEnterer" minOccurs="0"/>

<xs:element name="effectiveTime" type="IVL\_TS" minOccurs="0"/>

</xs:sequence>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required" fixed="RQO"/>

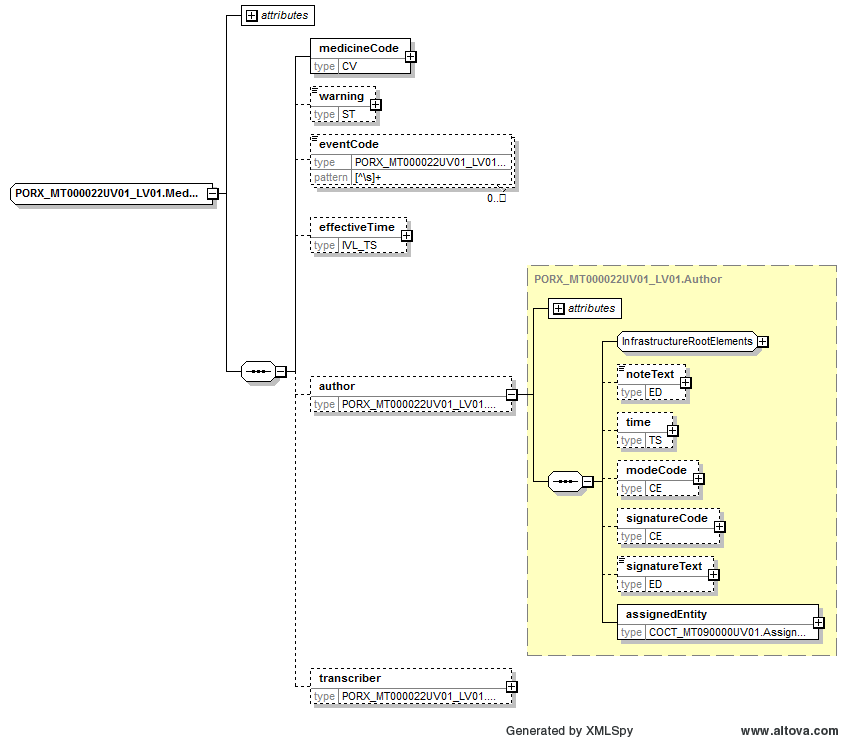
</xs:complexType>

Table 94 Data structure PORX\_MT000019UV01\_LV01.ProfileSetupRequest elements

| Element | Type | Cardinality | Description |
| --- | --- | --- | --- |
| subject | PORX\_MT000019UV01\_LV01.Subject | 1..1 |  |
| subject. patient | COCT\_MT050000UV01.Patient | 1..1 |  |
| subject. patient. id | II | 0..unbounded | Profile holder’s person identifier.  Personal code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1). |
| directTarget | PORX\_MT000019UV01\_LV01.DirectTarget | 1..1 |  |
| directTarget. profile | PORX\_MT000019UV01\_LV01.Profile | 1..1 |  |
| directTarget. profile. pharmacy | II | 0..1 | Patient pharmacy code in user profile (OID: 1.3.6.1.4.1.38760.2.134) |

## PORX\_MT000022UV01\_LV01.MedicationWarning

This is complex data structure based on HL7 standard; it describes structure of medication warning.



Picture 47 PORX\_MT000022UV01\_LV01.MedicationWarning data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT000022UV01\_LV01.MedicationWarning">

<xs:sequence>

<xs:element name="medicineCode" type="CV"/>

<xs:element name="warning" type="ST" minOccurs="0"/>

<xs:element name="eventCode" type="PORX\_MT000022UV01\_LV01.MedicationWarningEventType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="effectiveTime" type="IVL\_TS" minOccurs="0"/>

<xs:element name="author" type="PORX\_MT000022UV01\_LV01.Author" nillable="true" minOccurs="0" maxOccurs="1"/>

<xs:element name="transcriber" type="PORX\_MT000022UV01\_LV01.DataEnterer" nillable="true" minOccurs="0" maxOccurs="1"/>

</xs:sequence>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

<xs:attribute name="classCode" type="ActClass" use="optional" fixed="ACT"/>

<xs:attribute name="moodCode" type="ActMoodRequest" use="required" fixed="RQO"/>

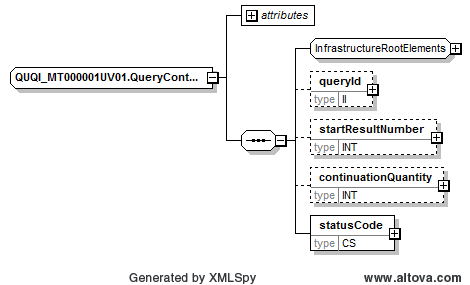
</xs:complexType>

Table 95 Data structure PORX\_MT000022UV01\_LV01.MedicationWarning elements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element | Type | Cardinality | | Description |
| medicineCode | CV | | 1..1 | Medication code (OID: 1.3.6.1.4.1.38760.2.136) |
| warning | ST | | 0..1 | Warning text |
| eventCode | PORX\_MT000022UV01\_LV01.MedicationWarningEventType | | 0..unbounded | Event code declaring when to show appropriate warning |
| effectiveTime | IVL\_TS | | 0..1 | Medication warning registration time |
| author | PORX\_MT000022UV01\_LV01.Author | | 0..1 |  |
| author. assignedEntity | COCT\_MT090000UV01.AssignedEntity | | 1..1 |  |
| author. assignedEntity. id | II | | 0..unbounded | Person, who registered medication warning, identifier.  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1) |

## QueyContinuation Header

This is complex data structure based on HL7 standard; it describes structure of query by parameter payload header, which is used in several services for querying appropriate information.



Picture 48 QueyContinuation Header data structure

Data structure XML schema:

<xs:complexType name="QUQI\_MT000001UV01.QueryContinuation">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="queryId" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="startResultNumber" type="INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="continuationQuantity" type="INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="statusCode" type="CS" minOccurs="1" maxOccurs="1"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

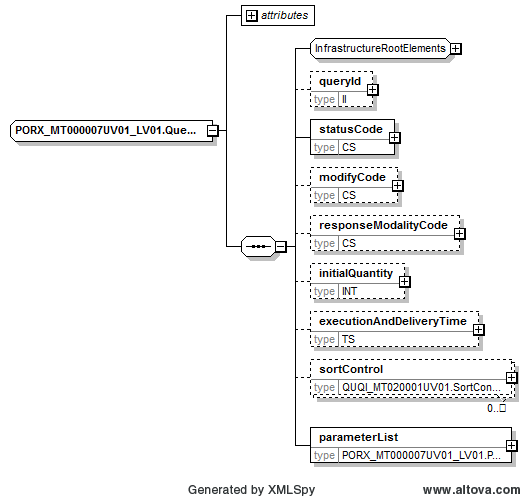
</xs:complexType>

Table 96 Data structure QueyContinuation Header elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Cardinality** | **Description** |
| queryId | II | 0..1 | See document [7] |
| startResultNumber | INT | 0..1 | See document [7] |
| continuationQuantity | INT | 0..1 | See document [7] |
| statusCode | CS | 1..1 | See document [7] |

## QueryByParameterPayload Header

This is complex data structure based on HL7 standard; it describes structure of query by parameter payload header, which is used in several services for querying appropriate information.



Picture 49 QueryByParameterPayload Header data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT000007UV01\_LV02.QueryByParameterPayload">

<xs:sequence>

<xs:group ref="InfrastructureRootElements"/>

<xs:element name="queryId" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="statusCode" type="CS" minOccurs="1" maxOccurs="1"/>

<xs:element name="modifyCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="responseModalityCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="initialQuantity" type="INT" minOccurs="0" maxOccurs="1"/>

<xs:element name="executionAndDeliveryTime" type="TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="sortControl" type="QUQI\_MT020001UV01.SortControl" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="parameterList" type="PORX\_MT000007UV01\_LV02.ParameterList" minOccurs="1" maxOccurs="1"/>

</xs:sequence>

<xs:attributeGroup ref="InfrastructureRootAttributes"/>

<xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>

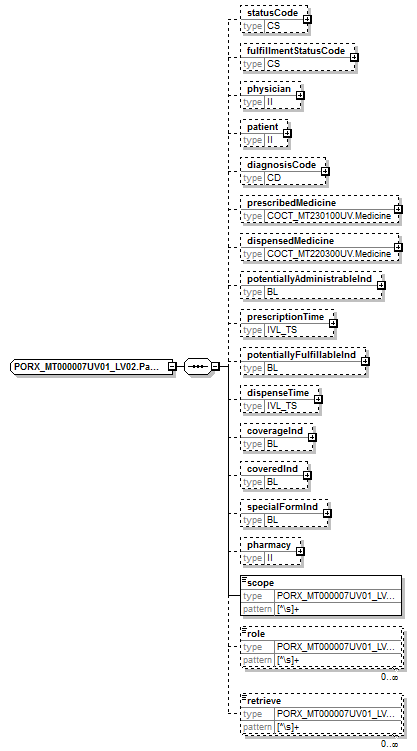
</xs:complexType>

Table 97 Data structure QueryByParameterPayload Header elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| queryId | II | 0..1 | See document [7] |
| statusCode | CS | 1..1 | See document [7] |
| modifyCode | CS | 0..1 | See document [7] |
| responseModalityCode | CS | 0..1 | See document [7] |
| initialQuantity | INT | 0..1 | See document [7] |
| executionAndDeliveryTime | TS | 0..1 | See document [7] |
| sortControl | QUQI\_MT020001UV01.SortControl | 0..unbouned | See document [7] |

## PORX\_MT000007UV01\_LV02.QueryByParameterPayload

This is complex data structure based on HL7 standard; it describes structure of specific query by parameter payload body, which is used in several services for querying appropriate information.



Picture 50 PORX\_MT000007UV01\_LV02.QueryByParameterPayload data structure

Data structure XML schema:

<xs:complexType name="PORX\_MT000007UV01\_LV02.ParameterList">

<xs:sequence>

<xs:element name="statusCode" type="CS" minOccurs="0" maxOccurs="1"/>

<xs:element name="fulfillmentStatusCode" type="CS" minOccurs="0" maxOccurs="1"/> <xs:element name="physician" type="II" minOccurs="0"/>

<xs:element name="patient" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="diagnosisCode" type="CD" minOccurs="0"/>

<xs:element name="prescribedMedicine" type="COCT\_MT230100UV.Medicine" minOccurs="0" maxOccurs="1"/>

<xs:element name="dispensedMedicine" type="COCT\_MT220300UV.Medicine" minOccurs="0" maxOccurs="1"/>

<xs:element name="potentiallyAdministrableInd" type="BL" minOccurs="0"/>

<xs:element name="prescriptionTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="potentiallyFulfillableInd" type="BL" minOccurs="0"/>

<xs:element name="dispenseTime" type="IVL\_TS" minOccurs="0" maxOccurs="1"/>

<xs:element name="coverageInd" type="BL" minOccurs="0"/>

<xs:element name="coveredInd" type="BL" minOccurs="0"/>

<xs:element name="specialFormInd" type="BL" minOccurs="0"/>

<xs:element name="pharmacy" type="II" minOccurs="0" maxOccurs="1"/>

<xs:element name="scope" type="PORX\_MT000007UV01\_LV02.ScopeType"/>

<xs:element name="role" type="PORX\_MT000007UV01\_LV02.RoleType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="retrieve" type="PORX\_MT000007UV01\_LV02.RetrieveType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

Table 98 Data structure PORX\_MT000007UV01\_LV02.QueryByParameterPayload elements

|  |  |  |  |
| --- | --- | --- | --- |
| Element | Type | Cardinality | Description |
| QueryByParameterPayload | PORX\_MT000007UV01\_LV02.QueryByParameterPayload | 1..1 | See data structure in chapter 6.12. |
| QueryByParameterPayload. parameterList | PORX\_MT000007UV01\_LV02.ParameterList | 1..1 |  |
| QueryByParameterPayload. parameterList. statusCode | CS | 0..1 | Medication order status code (new – for reserved, active – for active, cancelled – for cancelled before activation, aborted – for cancelled after activation, complete – for expired or fully dispensed) |
| QueryByParameterPayload. parameterList. fulfillmentStatusCode | CS | 0..1 | Medication otder fulfilment status code (unfulfilled, partial – partially fulfilled, fulfilled) |
| QueryByParameterPayload. parameterList. physician | II | 0..1 | Person code or other identifying code (for example, physician code)  Physician (OID: 1.3.6.1.4.1.38760.2.1).  Physician code (OID: 1.3.6.1.4.1.38760.3.1.4).  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1) |
| QueryByParameterPayload. parameterList. patient | II | 0..1 | Person identifier.  Person code for Latvian citizen (OID: 1.3.6.1.4.1.38760.3.1.1).  Alternative identifier for newborns – consists of mother’s personal code and child birth time (OID: 1.3.6.1.4.1.38760.3.1.3).  Identification number for foreign country citizens (any child node in OID namespace 1.3.6.1.4.1.38760.3.1.8 except 1.3.6.1.4.1.38760.3.1.8.60). |
| QueryByParameterPayload. parameterList. diagnosisCode | CD | 0..1 | Diagnosis ICD-10 code (OID: 1.3.6.1.4.1.38760.2.159) |
| QueryByParameterPayload. parameterList. prescribedMedicine | COCT\_MT230100UV. Medicine | 0..1 |  |
| QueryByParameterPayload. parameterList. prescribedMedicine. code | CE | 0..1 | Prescribed medicine – medical registration number (OID: 1.3.6.1.4.1.38760.2.136), compensable medicine (OID: 1.3.6.1.4.1.38760.2.151) or compensation group (OID: 1.3.6.1.4.1.38760.2.177). |
| QueryByParameterPayload. parameterList. prescribedMedicine. name | TN | 0..unbound | Prescribed medicine – medicine name or part of it. |
| QueryByParameterPayload. parameterList. prescribedMedicine. ingredient | COCT\_MT230100UV.Ingredient | 0..unbounded |  |
| QueryByParameterPayload. parameterList. prescribedMedicine. ingredient. ingredientSubstance | COCT\_MT230100UV.Substance | 0..1 |  |
| QueryByParameterPayload. parameterList. prescribedMedicine. ingredient. ingredientSubstance. code | CE | 0..1 | Prescribed medicine – active substance code (OID: 1.3.6.1.4.1.38760.2.140) |
| QueryByParameterPayload. parameterList. dispensedMedicine | COCT\_MT220300UV.Medicine | 0..1 |  |
| QueryByParameterPayload. parameterList. dispensedMedicine. code | CE | 0..1 | Dispensed medicine – medical registration number (OID: 1.3.6.1.4.1.38760.2.136) or medication code (OID: 1.3.6.1.4.1.38760.2.144). |
| QueryByParameterPayload. parameterList. dispensedMedicine. name | TN | 0..unbound | Dispensed medicine – name or part of it. |
| QueryByParameterPayload. parameterList. potentiallyAdministrableInd | BL | 0..1 | Indicator: get potentially administrable medicines only |
| QueryByParameterPayload. parameterList. prescriptionTime | IVL\_TS | 0..1 | Medication order prescription time (from – to) |
| QueryByParameterPayload. parameterList. potentiallyFulfillableInd | BL | 0..1 | Indicator: get potentially fulfillable medication orders only  (for pharmacists – to get medication prescribed to pharmacy; for patient – to get medication orders, which are still dispensable).  Based on medication orders, which are not expired and not dispensed or partly dispensed. |
| QueryByParameterPayload. parameterList. dispenseTime | IVL\_TS | 0..1 | Medication dispense time (from – to) |
| QueryByParameterPayload. parameterList. coverageInd | BL | 0..1 | Indicator: get prescribed compensable medicine only. |
| QueryByParameterPayload. parameterList. coveredInd | BL | 0..1 | Indicator: get dispensed compensable medicine only. |
| QueryByParameterPayload. parameterList. specialFormInd | BL | 0..1 | Indicator: special paper form (narcotic drugs and psychotropic, compensated medication) |
| QueryByParameterPayload. parameterList. pharmacy | II | 0..1 | Pharmacy (OID: 1.3.6.1.4.1.38760.2.134) |
| QueryByParameterPayload. parameterList. scope | PORX\_MT000007UV01\_LV02.ScopeType | 1..1 | \* Allow to set scope of data to retrieve.  One of values:  - USR – get medication orders where user is in any of passed roles;  - PTN – get medication orders where passed patient is in any of passed roles;  - ORG – get data from passed organization pharmacy;  - ALL – get all the medication orders.  Uses with “role” element. |
| QueryByParameterPayload. parameterList. role | PORX\_MT000007UV01\_LV02.RoleType | 0..unbounded | \* Allow to set role for scope.  One or more of values:  - SBJ – medication orders prescribed for user or passed patient;  - AUT – user prescribed medication orders;  - TRN – user transcribed medication orders;  - DLG – medication orders delegated for user or passed patient.  Uses with “scope” element. |
| QueryByParameterPayload. parameterList. retrieve | PORX\_MT000007UV01\_LV02.RetrieveType | 0..unbounded | \*\* Allow to set retrievable data structure parts.  One or more of values:  - ORD.ADM;  - ORD.DGN;  - ORD.DIS;  - ORD.REC;  - ORD.PTN;  - ORD.AUT;  - ORD.TRN;  - ORD.MED;  - ORD.CMP;  - ORD.ALL;  - DIS.SUP;  - DIS.REC;  - DIS.ALL;  - CAN.ALL. |

\* Specification for elements “scope” and “role” usage:

* for “scope” value “USR” there must be passed “role” value, it can be any of values “SBJ”, “AUT”, “TRN”, “DLG”;
* for “scope” value “PTN” there must be passed “role” value, it can be any of values “SBJ”, “DLG”;
* for “scope” value “ALL” there is no “role” value needed, “role” value will be ignored;
* for “scope” value “ORG” there is no “role” value needed, “role” value will be ignored.

\*\* Specification for element “retrieve” usage, explaining each of value.

Value consists of prefix and suffix.

Prefix explains which data structure to fulfill into retrieve data structure.

Suffix explains which data structure part to fulfill into retrievable data structure.

Table 99 Element “retrieve” suffix values

|  |  |
| --- | --- |
| Value | Data structure part to include |
| ORD.ADM | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “component1.substanceAdministrationRequest”.  *Except element “component1.substanceAdministrationRequest.reason”.* |
| ORD.DGN | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “component1.substanceAdministrationRequest.reason”. |
| ORD.DIS | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “component2.dispenseRequest”.  *Except element “component2.dispenseRequest.receiver”.* |
| ORD.REC | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “component2.dispenseRequest.receiver”. |
| fORD.PTN | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “subject.patient.patientPerson”. |
| ORD.AUT | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “author.assignedEntity”. |
| ORD.TRN | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “transcriber.assignedEntity”. |
| ORD.MED | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “directTarget.medication.administrableMedicine”. |
| ORD.CMP | Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “coverage.compensationRequest”. |
| ORD.ALL | Entire data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest. |
| DIS.SUP | Data structure PORX\_MT020070UV01\_LV02.CombinedMedicationDispense element “component3.supplyEvent”.  *Except element “component3.supplyEvent.receiver”.* |
| DIS.REC | Data structure PORX\_MT020070UV01\_LV02.CombinedMedicationDispense element “component3.supplyEvent.receiver”. |
| DIS.ALL | Entire data structure PORX\_MT020070UV01\_LV02.CombinedMedicationDispense.  or  Data structure PORX\_MT010120UV01\_LV02.CombinedMedicationRequest element “fulfilledBy.combinedMedicationDispense”. |
| CAN.ALL | Data structure PORX\_MT000025UV01\_LV01.CancelMedicationOrderRequest element “subjectOf5.cancelMedicationOrderRequest”. |

## HL7 standard data structures usage examples

This chapter contains examples of HL7 standard data structures usage, including different cases.

According to HL7 standard:

* elements (of type PQ) attribute “unit” must be used for HL7 standard supported UCUM units only. Size of package (UCUM value - orig) must be used in “translation” element as it is e‑Health specific classifier;
* elements “name” and “address” are to be fulfilled as texts with added logical mark-up (The approach is similar to HTML or XML markup of text).

System supports such UCUM units ([*http://unitsofmeasure.org/ucum.html*](http://unitsofmeasure.org/ucum.html)) case insensitive:

* TimeUnits – "s", "min", "h", "d", "wk", "mo", "a";
* MassUnits – "ug", "mg", "g", "kg";
* VolumeUnits – "ml", "l", "cm3", "dm3", "m3";
* Units – "1", "{ORIG}".

### Potential compensation payer (sponsor) and compensation payer (sponsor) examples

Example of potential compensation payer (sponsor) as State – use appropriate classifier (OID: 1.3.6.1.4.1.38760.2.93) value “STATE” – according to compensation condition with code Z25.1-J07BB02-50 in medication order:

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<payer code="STATE" codeSystem="1.3.6.1.4.1.38760.2.93" />

<code code="Z25.1-J07BB02-50" codeSystem="1.3.6.1.4.1.38760.2.152" />

</compensationRequest>

</coverage>

Example of compensation payer (sponsor) as State – use appropriate classifier (OID: 1.3.6.1.4.1.38760.2.93) value “STATE” – according to compensation condition with code Z25.1-J07BB02-50 in medication dispense:

<supplyEvent>

…

<payer code="STATE" codeSystem="1.3.6.1.4.1.38760.2.93" />

…

<supplyEvent>

Example of potential compensation payer (sponsor) as private company with compensation percentage of 50% in medication order:

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<payer code="01" codeSystem="1.3.6.1.4.1.38760.2.93" displayName="A/S Balta"/>

<percentageAmount value="50" />

</compensationRequest>

</coverage>

Example of compensation payer (sponsor) as private company with compensation percentage of 50% in medication dispense:

<supplyEvent>

…

<payer code="01" codeSystem="1.3.6.1.4.1.38760.2.93" displayName="A/S Balta"/>

<compensationPercent value="50" />

…

<supplyEvent>

Example of potential compensation payer (sponsor) as non-classified value with compensation percentage of 50% in medication order:

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<payer nullFlavor="UNC">

<originalText>A/S Balta</originalText>

</payer>

<percentageAmount value="50" />

</compensationRequest>

</coverage>

Example of compensation payer (sponsor) as non-classified value with compensation percentage of 50% in medication dispense:

<supplyEvent>

…

<payer nullFlavor="UNC">

<originalText>A/S Balta</originalText>

</payer>

<compensationPercent value="50" />

…

<supplyEvent>

Example of both potential compensation payers (sponsors) State compensation (according to compensation condition with code Z25.1-J07BB02-50) and private company compensation (with compensation percentage of 50%)in medication order:

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<payer code="STATE" codeSystem="1.3.6.1.4.1.38760.2.93" />

<code code="Z25.1-J07BB02-50" codeSystem="1.3.6.1.4.1.38760.2.152" />

</compensationRequest>

</coverage>

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<payer code="01" codeSystem="1.3.6.1.4.1.38760.2.93" displayName="A/S Balta"/>

<percentageAmount value="50" />

</compensationRequest>

</coverage>

### Person identification and documents examples

Example of receiver person identification by his Latvian citizen person code (OID: 1.3.6.1.4.1.38760.3.1.1; person code = 01018211119) and passport (OID: 1.3.6.1.4.1.38760.3.3.1.1; passport number = LV0000001) in medication dispense:

<supplyEvent>

…

<receiver typeCode="RCV">

<assignedPerson classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119" />

<id root="1.3.6.1.4.1.38760.3.3.1.1" extension="LV0000001" />

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name use="L">

<given>Jānis</given>

<family>Liepiņš</family>

</name>

</assignedPerson>

</assignedPerson>

</receiver>

…

<supplyEvent>

### Foreign citizen identification and documents examples

Example of patient person identification by his foreign citizen identifier code (OID: 1.3.6.1.4.1.38760.3.1.8.77) and EHIC card number (OID: 1.3.6.1.4.1.38760.3.3.1.3) in medication order:

<combinedMedicationRequest moodCode="RQO">

…

<subject typeCode="SBJ">

<patient classCode="PAT">

<patientPerson>

<id root="1.3.6.1.4.1.38760.3.1.8.77" extension="AB123456C" />

<id root="1.3.6.1.4.1.38760.3.3.1.3" extension="123412341234" />

<name use="L">

<given>Mery</given>

<family>Smith</family>

</name>

<administrativeGenderCode code="S" codeSystem="1.3.6.1.4.1.38760.2.111" />

<birthTime value="19811012000000.0000+0300" />

<addr>

Building 407 St John Street, London, <country>GB</country>

</addr>

</patientPerson>

</patient>

</subject>

…

</combinedMedicationRequest>

### Medication usage examples

For medication strength there must be used medication data structure element “name”, which should be like this:

[medication name from classifier] [medication strength with unit]

Example of active substance of compensation group (OID: 1.3.6.1.4.1.38760.2.177) with code = 2 (Maravirocum), strength = 200mg, formCode = 1141:

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="1" codeSystem="1.3.6.1.4.1.38760.2.177" />

<name>Maravirocum 200mg</name>

<formCode code="1141" codeSystem="1.3.6.1.4.1.38760.2.137" />

</administrableMedicine>

Example of medication (OID: 1.3.6.1.4.1.38760.2.137) with code = 05-0604:

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="05-0604" codeSystem="1.3.6.1.4.1.38760.2.136" />

</administrableMedicine>

Example of compensable medicine (OID: 1.3.6.1.4.1.38760.2.151) with code = 10-0149-01:

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="10-0149-01" codeSystem="1.3.6.1.4.1.38760.2.151" />

</administrableMedicine>

Example of medication with free text format with name = Maravirocum, strength = 200mg, formCode = 1141:

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<name>Maravirocum 200mg</name>

<formCode code="1141" codeSystem="1.3.6.1.4.1.38760.2.137" />

</administrableMedicine>

Example of medication consisting of active substances (OID: 1.3.6.1.4.1.38760.2.140) with code = 523, strength = 150mg, formCode = 1141:

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<ingredient classCode="INGR">

<ingredientSubstance classCode="MMAT" determinerCode="KIND">

<code code="523" codeSystem="1.3.6.1.4.1.38760.2.140"/>

</ingredientSubstance>

<quantity nullFlavor="UNC">

<numerator xsi:type="PQ" nullFlavor="UNC">

<translation nullFlavor="UNC">

<originalText>150 mg</originalText>

</translation>

</numerator>

<denominator xsi:type="PQ"/>

</quantity>

</ingredient>

<formCode code="1141" codeSystem="1.3.6.1.4.1.38760.2.137" />

</administrableMedicine>

### Period usage examples

Example of administration period duration 30 days:

<width value="30" unit="d" />

### Quantity usage examples

Example of 10 tablets:

<quantity value="10" />

* using HL7 units:

<quantity value="10" unit="ml" />

Example of 10 mililiters (ml), which is half of package (for medication dispense):

<quantity value="10" unit="ml" >

<translation value="0.5" unit="{ORIG}" />

</quantity>

### Name usage examples

According to HL7 standard, person element “name” is textual with logical mark‑ups. It means that such text may contain some special tags (mark-ups), which are surrounding and therefore identifying appropriate fields (example of mark‑up: <myMarkUp>Some code value</myMarkUp>). Only marked fields will be processed.

For each of names there can be used such mark-ups (other mark-ups will be ignored):

* given – given name (name, firstname) mark-up;
* family – family name (surname, lastname) mark-up;

Example of name with marked given and family names:

<name><given>Jānis</given> <family>Bērziņš</family></name>

Example of name with marked given, family names and prefix:

<name><prefix >Mr.</prefix> <given>Jānis</given> <family>Bērziņš</family></name>

### Address usage examples

According to HL7 standard, element “address” is a textual with logical mark‑ups. It means that such text may contain some special tags (mark-ups), which are surrounding and therefore identifying appropriate fields (example of mark‑up: <myMarkUp>Some code value</myMarkUp>). Only marked fields will be processed.

Any address can be used in 2 different forms:

* Non-structured – element “address” doesn’t contain any of mark-up (should be used in very limited cases, when there is no chance to structure any part of address);
* Structured – element “address” contain at least one mark-up – contry code mark-up “country” (for example: addreses … <country>LV</country>).

Non-structured form is not preferable because address data will be used for furthere data analysis, it means that non-strucuted form should be used only in limited cases!

#### Foreign country address

Foreign country address can be used in non-structured or structured form. For each of foreign country citizen addresses there should be used such mark-ups (other mark-ups will be ignored):

* country – country code mark-up (mandatory).

Non-structured form is not preferable because address data will be used for furthere data analysis, it means that non-structured form should be used only in case, when there is no chance to identify the country!

Example of foreign citizen address in non-structured form (not preferable):

* 3300 Washtenaw Avenue, Suite 227 Ann Arbor, MI 48104 US

<addr>3300 Washtenaw Avenue, Suite 227 Ann Arbor, MI 48104, US</addr>

Example of foreign citizen address:

* 3300 Washtenaw Avenue, Suite 227 Ann Arbor, MI 48104 US

<addr>3300 Washtenaw Avenue, Suite 227 Ann Arbor, MI 48104 <country>US</country></addr>

#### Latvian address

Latvian address can be used in non-structured or structured form. Correct form of Latvian address should be like this:

[street name] [house number | house name]–[flat number], [city | village], [parish | rural area], [county], [postal code], [country] [ATU code]

For each of Latvian addresses there can be used such mark-ups (other mark-ups will be ignored):

* streetName – street name mark-up;
* houseNumber – house number mark-up;
* additionalLocation – house name mark-up;
* unitID – flat number mark-up;
* city – city or village mark-up;
* county – county, parish and rural area mark-up;
* postalCode – postal code mark-up;
* country – country code mark-up;
* censusTract – ATU code mark-up;
* delimiter – any other text between mark-ups can be surrounded by delimiter mark-ups to make all the address text marked.

Depending on Latvian address case, some fields may be omitted, there are such main rules for Latvian addresses mark-ups:

* postalCode – mandatory;
* censusTract – mandatory;
* city | county – at least one of these is mandatory;
* additionalLocation | streetName + houseNumber – at least one of these is mandatory;
* streetName + houseNumber – if there is one of these than other must be.

Non-structured form is not preferable because address data will be used for furthere data analysis, it means that non-structured form should be used only in case, when there is no chance to identify the address structure!

Example of address in non-structurel form (not preferable):

* Bišu iela 3, Gulbene, Gulbenes nov., LV-4401, LV

<addr>Bišu iela 3, Gulbene, Gulbenes nov., LV-4401, LV</addr>

Example of address with marked street name, house number, city, county, postal code, country and ATU code, buy without delimiter markups:

* Bišu iela 3, Gulbene, Gulbenes nov., LV-4401

<addr><streetName>Bišu iela</streetName> <houseNumber>3</houseNumber>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country><censusTract>0500201</censusTract></addr>

Example of address with marked street name, house number, city, county, postal code, country, ATU code and delimiters between them:

* Bišu iela 3, Gulbene, Gulbenes nov., LV-4401

<addr><streetName>Bišu iela</streetName><delimiter> </delimiter><houseNumber>3</houseNumber><delimiter>, </delimiter><city>Gulbene</city><delimiter>, </delimiter><county>Gulbenes nov.</county><delimiter>, </delimiter><postalCode>LV-4401</postalCode>, <country>LV</country><censusTract>0500201</censusTract></addr>

Examples of addresses with correct postal code:

* Dāboltiņi, Bringi, Mežvidu pag., Kārsavas nov., LV-5753

<addr><additionalLocator>Dāboltiņi</additionalLocator>, <city>Bringi</city>, <county>Mežvidu pag.</county>, <county>Kārsavas nov.</county>, <postalCode>LV-5753</postalCode>, <country>LV</country><censusTract>0681070</censusTract></addr>

Examlpes of addresses with correct street names:

* Melnsila iela 6, Rīga, LV-1046

<addr><streetName>Melnsila iela</streetName> <houseNumber>6</houseNumber>, <city>Rīga</city>, <postalCode>LV-1046</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Kārļa Ulmaņa gatve 2A, Rīga, LV-1004

<addr><streetName>Kārļa Ulmaņa gatve</streetName> <houseNumber>2A</houseNumber>, <city>Rīga</city>, <postalCode>LV-1004</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Aspazijas bulvāris 24, Rīga, LV-1050

<addr><streetName>Aspāzijas bulvāris</streetName> <houseNumber>24</houseNumber>, <city>Rīga</city>, <postalCode>LV-1050</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Ūbeļu iela 16-51, Ādaži, Ādažu nov., LV-2164

<addr><streetName>Ūbeļu iela</streetName> <houseNumber>16</houseNumber>-<unitID>16</unitID>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country><censusTract>0804400</censusTract></addr>

Examples of addresses with correct house numbers (with street name; building block must be passed as "k-XXX", not separated of house number with slash.):

* Zaļā iela 3–11, Namiķi, Lutriņu pag., Saldus nov., LV-3861

<addr><streetName>Zāļu iela</streetName> <houseNumber>3</houseNumber>-<unitID>11</unitID>, <city>Namiķi</city>, <county>Lutriņu pag.</county>, <county>Saldus nov.</county>, <postalCode>LV-3861</postalCode>, <country>LV</country><censusTract>0840266</censusTract></addr>

* Vienības gatve 168–50, Rīga, LV-1058

<addr><streetName>Vienības gatve </streetName> <houseNumber>168</houseNumber>-<unitID>50</unitID>, <city>Rīga</city>, <postalCode>LV-1058</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Miera iela 15A, Strūžāni, Stružānu pag., Rēzeknes nov., LV-4643

<addr><streetName>Miera iela</streetName> <houseNumber>15A</houseNumber>, <city>Stružāni</city>, <county>Stružānu pag.</county>, <county>Rēzeknes nov.</county>, <postalCode>LV-4643</postalCode>, <country>LV</country><censusTract>0780294</censusTract></addr>

* Kolkasraga iela 16/18–1, Rīga, LV-1002

<addr><streetName>Kolkasraga iela</streetName> <houseNumber>16/18</houseNumber>-<unitID>1</unitID>, <city>Rīga</city>, <postalCode>LV-1002</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Katrīnas dambis 26 k-4, Rīga, LV-1045

<addr><streetName>Katrīnas dambis </streetName> <houseNumber>26 k-4</houseNumber>, <city>Rīga</city>, <postalCode>LV-1045</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

Examples of addresses with correct house names (without street name):

* Kalna Vēliņi, Raunas pag., Raunas nov., LV-4131

<addr><additionalLocator>Kalna Vēliņi</additionalLocator>, <county>Raunas pag.</county>, <county>Raunas nov.</county>, <postalCode>LV-4131</postalCode>, <country>LV</country><censusTract>0427776</censusTract></addr>

* Ziedkalni–8, Grenči, Zemītes pag., Kandavas nov., LV-3136

<addr><additionalLocator>Ziedkalni-8</additionalLocator>, <city>Grenči</city>, <county>Zemītes pag.</county>, <county>Kandavas nov.</county>, <postalCode>LV-3136</postalCode>, <country>LV</country><censusTract>0901294</censusTract></addr>

* Jaunsaules, Dzērbenes pag., Vecpiebalgas nov., LV-4118

<addr><additionalLocator>Jaunsaules</additionalLocator>, <county>Dzērbenes pag.</county>, <county>Vecpiebalgas nov.</county>, <postalCode>LV-4118</postalCode>, <country>LV</country><censusTract>0429350</censusTract></addr>

* Kalniņi 2, Lejasciema pag., Gulbenes nov., LV-4413

<addr><additionalLocator>Kalniņi 2</additionalLocator>, <county>Lejasciema pag.</county>, <county>Gulbenes nov.</county>, <postalCode>LV-4413</postalCode>, <country>LV</country><censusTract>0500264</censusTract></addr>

* Dzelzceļa ēka 23. km, Olaines pag., Olaines nov., LV-2127

<addr><additionalLocator>Dzelceļa ēka 23. km</additionalLocator>, <county>Olaines pag.</county>, <county>Olaines nov.</county>, <postalCode>LV-2127</postalCode>, <country>LV</country><censusTract>0801080</censusTract></addr>

* Druvienas, Kaspari, Mežvidu pag., Kārsavas nov., LV-5725

<addr><additionalLocator>Kalniņi 2</additionalLocator>, <county>Lejasciema pag.</county>, <county>Gulbenes nov.</county>, <postalCode>LV-5725</postalCode>, <country>LV</country><censusTract>0681070</censusTract></addr>

Examples of addresses with correct flat numbers:

* Brīvības iela 20–4, Ogre, Ogres nov., LV-5001

<addr><streetName>Brīvības iela</streetName> <houseNumber>20</houseNumber>-<unitID>4</unitID>, <city>Ogre</city>, <county>Ogres pag.</county>, <postalCode>LV-5001</postalCode>, <country>LV</country><censusTract>0740201</censusTract></addr>

* Artilērijas iela 6–6, Rīga, LV-1001

<addr><streetName>Artilērijas iela</streetName> <houseNumber>6</houseNumber>-<unitID>6</unitID>, <city>Rīga</city>, <postalCode>LV-1001</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Mārupes iela 6–2A, Rīga, LV-1002

<addr><streetName>Mārupes iela</streetName> <houseNumber>6</houseNumber>-<unitID>2A</unitID>, <city>Rīga</city>, <postalCode>LV-1002</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Sīmaņa iela 6–34/35, Rīga, LV-1005

<addr><streetName>Sīmaņa iela</streetName> <houseNumber>6</houseNumber>-<unitID>34/35</unitID>, <city>Rīga</city>, <postalCode>LV-1005</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

* Ziedkalnes 2–18, Mūrmuiža, Vilces pag., Jelgavas nov., LV-3027

<addr><additionalLocator>Ziedkalnes 2</additionalLocator>-<unitID>18</unitID>, <city>Mūrmuiža</city>, <county>Vilces pag.</county>, <county>Jelgavas nov.</county>, <postalCode>LV-3027</postalCode>, <country>LV</country><censusTract>0540290</censusTract></addr>

Examples of republic cities address:

* Dūņu iela 1, Jelgava, LV-3001

<addr><streetName>Dūņu iela</streetName> <houseNumber>1</houseNumber>, <city>Jelgava</city>, <postalCode>LV-3001</postalCode>, <country>LV</country><censusTract>0090000</censusTract></addr>

* Višķu iela 1-44, Rīga, LV-1063

<addr><streetName>Višķu iela</streetName> <houseNumber>1</houseNumber>-<unitID>44</unitID>, <city>Rīga</city>, <postalCode>LV-1063</postalCode>, <country>LV</country><censusTract>0010000</censusTract></addr>

Examples of county cities address:

* Bērzpils iela 3–4, Balvi, Balvu nov., LV-4501

<addr><streetName>Bērzpils iela</streetName> <houseNumber>3</houseNumber>-<unitID>4</unitID>, <city>Balvi</city>, <county>Balvu no.</county>, <postalCode>LV-4501</postalCode>, <country>LV</country><censusTract>0380201</censusTract></addr>

* Vārpu iela 1-2, Ikšķile, Ikšķiles nov., LV-5052

<addr><streetName>Vārpu iela</streetName> <houseNumber>1</houseNumber>-<unitID>2</unitID>, <city>Ikšķile</city>, <county>Ikšķiles pag.</county>, <postalCode>LV-5052</postalCode>, <country>LV</country><censusTract>0740605</censusTract></addr>

Examples of village address:

* Daugavas prospekts 3-2, Ādamlauks, Ikšķiles l.t., Ikšķiles nov., LV-5052

<addr><streetName>Daugavpils prospekts</streetName> <houseNumber>3</houseNumber>-<unitID>2</unitID>, <city>Ādamlauks</city>, <county>Ikšķiles l.t.</county>, <county>Ikšķiles nov.</county>, <postalCode>LV-5052</postalCode>, <country>LV</country><censusTract>0740605</censusTract></addr>

Examples of county, parish or rural area address:

* Jaunbirzgaļi, Kārļi, Drabešu pag., Amatas nov., LV-4138

<addr><additionalLocator>Jaunbirzgaļi</additionalLocator>, <city>Kārļi</city>, <county>Drabiešu pag.</county>, <county>Amatas nov.</county>, <postalCode>LV-4138</postalCode>, <country>LV</country><censusTract>0424746</censusTract></addr>

* Upītes, Apes l. t., Apes nov., LV-4337

<addr><additionalLocator>Upītes</additionalLocator>, <county>Apes l.t.</county>, <county>Apes nov.</county>, <postalCode>LV-4337</postalCode>, <country>LV</country><censusTract>0360825</censusTract></addr>

# Permission delegation

From patient perspective medication order and patient profile access is allowed only for the person which is specified in medication order as patient or specified as holder of profile. However System supports medication order and profile access permission delegation via security token delegates.

In order to access other patient medication orders or profiles, security token should contain corresponding *Delegations* attribute. For detailed information on security token structure see [7].

**Security token example:**

<o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">

<saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="\_46ded6f3-7b83-462f-a240-db201f329684" Issuer="https://eves-prod.abc/ip.sts/1.0" IssueInstant="2015-11-20T13:30:51.337Z" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">

…

<saml:AttributeStatement>

…

<saml:Attribute AttributeName="Delegations">

<Actor>

<saml:Attribute AttributeName="PatientType">

<saml:AttributeValue>P8\_DELEGATE</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute AttributeName="privatepersonalidentifier">

<saml:AttributeValue>10077610501</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute AttributeName="givenname">

<saml:AttributeValue>Sanita</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute AttributeName="surname">

<saml:AttributeValue>Vitola</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute AttributeName="role">

<saml:AttributeValue>Patient</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute AttributeName="action">

<saml:AttributeValue>GetProfile</saml:AttributeValue>

<saml:AttributeValue>QueryMedicationDispenses</saml:AttributeValue>

<saml:AttributeValue>QueryMedicationOrders</saml:AttributeValue>

<saml:AttributeValue>SetProfile</saml:AttributeValue>

</saml:Attribute>

</Actor>

</saml:Attribute>

…

</saml:AttributeStatement>

</saml:Assertion>

</o:Security>

# Traceability matrix

Table 100 Traceability matrix

|  |  |
| --- | --- |
| TS Requirement | System Requirement |
| Chapter, identifier, Name |
| **E-Health 1st stage requirements** | |
| (005) Sistēmai jānodrošina šādu pamata dokumentu aprite, par dokumentu uzskatot:   * [Izrakstīto e‑recepti](#_Izrakstītā_recepte), kopā ar tās saistīto informāciju. Šim dokumentam iespējami dažādi veidi – parastā recepte vai īpašā recepte, kas var būt kompensējamā medikamenta recepte vai recepte, kas satur narkotisko un psihotropo vielu saturošus medikamentus. * Ziņojumu par [medikamenta izsniegšanu](#_Izsniegtais_ārstniecības_līdzeklis) pret e‑recepti * Ziņojumu par [e‑receptes anulēšanu](#_Paziņojums_apr_receptes), norādot anulēšanas iemeslu * Ziņojumu par [ārstniecības personas licences apturēšanu vai anulēšanu](#_Ziņojumu_par_ārstniecības) * Ziņojumu par [medikamenta pieejamības tirgū](#_Ziņojumu_par_medikamenta) statusa maiņu * Ziņojumu par [farmaceita statusa](#_Ziņojumu_par_farmaceita) maiņu * Ziņojumu par [medikamenta atsaukšanu](#_Ziņojumu_par_medikamenta_1).   Sistēmā izmantojamo ziņojumu saraksts un to informācijas apjoms, aprakstīts XML shēmu formā, sniegts pielikumā (Ziņojumi un XML shēmas 47.lpp). Minētais saraksts, kā arī pielikumā norādītās shēmas Izstrādātājam jāpapildina ar nepieciešamajām definīcijām vai jāpārstrādā, izveidojot datu apmaiņas standartu ar aptieku un ārsta prakses sistēmām. | Documents are described in 6. Data structures.  These documents are not System responsibility:   * “Ziņojums par ārstniecības personas licences apturēšanu vai anulēšanu”; * “Ziņojums par medikamenta pieejamības tirgū statusa maiņu”; * “Ziņojums par farmaceita statusa maiņu”. |
| (013) Sistēmai automātiski, izmantojot tīmekļa pakalpi, jānosūta informācija par kompensējamiem izsniegtajiem ārstniecības līdzekļiem e‑booking finanšu norēķinu modulim. Piegādātājam, sadarbībā ar e‑booking izstrādātāju, prasību analīzes laikā jāveic datu formāta sagatavošana un saskaņošana. | Data structure for dispense is 6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense. |
| (055) Sistēmā jārealizē datu apmaiņas saskarne ar ārsta prakses IS, kurās ir nodrošināta sadarbspēja darbam e‑receptes tīklā, nodrošinot izrakstīto recepšu nodošanu e‑receptes servisam, kā arī citu saistīto ziņojumu apmaiņu. | Such services:   * 5.1. BookMedicationOrders * 5.2. GetCompensationConditionList * 5.3. RegisterMedicationOrder * 5.4. GetMedicationOrderData * 5.5. CancelMedicationOrder * 5.6. GetMedicationOrderList * 5.8. GetMedicineList * 5.9. GetDiagnosisList |
| (056) Sistēmā jārealizē datu apmaiņas saskarne ar tām aptieku IS, kurās ir nodrošināta sadarbspēja darbam e‑receptes tīklā, nodrošinot informācijas par izsniegto ārstniecības līdzekli nodošanu e‑receptes servisam, kā arī citu saistīto ziņojumu apmaiņu. | Such services:   * 5.4. GetMedicationOrderData * 5.6. GetMedicationOrderList * 5.3. RegisterMedicationOrder * 5.10. BookMedicationDispense * 5.11. RegisterMedicationDispense * 5.12. CancelMedicationDispense * 5.13. GetMedicationDispenseList |
| (107) Ziņojumapmaiņas funkcionalitātei ir jābūt balstītai uz SOA principiem un izmantojot tīkla servisu platformu.  Tīkla servisu platformas pamatu pamatos tiek izmantoti tīkla servisi, kuru saskarnes tiek aprakstītas ar WSDL[[1]](#footnote-1) (valoda tīkla servisu saskarņu aprakstīšanai) un XML Schema (valoda XML struktūru aprakstīšanai)[[2]](#footnote-2) starpniecību, bet datu apmaiņa starp servisiem notiek ar SOAP[[3]](#footnote-3) (datu apmaiņas protokols, kas ļauj apmainīt SOAP ziņojumus jeb elektroniskas aploksnes, izmantojot dažādus transporta protokolus – HTTP, SMTP, FTP u.c.) protokola palīdzību. | 3. Interface description  5. Services  6. Data structures |
| (108) Ziņojumu formātam ir jāizmanto XML[[4]](#footnote-4). | 6. Data structures |
| (109) XML struktūru aprakstam ir jāizmanto XML Schema2 (XSD). | 6. Data structures |
| **E-Health 2nd stage requirements** | |
| REC-02 Ārsta kontaktinformācija e-receptē (Obligāta) | 6.2. COCT\_MT050000UV01.Person  6.3. COCT\_MT230100UV.Medicine  6.14. HL7 standard data structures usage examples  9.1.3.3. Get recipe data  10.1. Error list |
| REC-05 Recepšu izsniegšanas brīdinājumi (Obligāta) | 6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense  9.2.1.1. Book medication dispense  9.2.1.3. Register medication dispense |
| POR-17 E-recepšu izrakstīšanas ekrānformas uzlabojumi (Obligāta) | 6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest  6.3. COCT\_MT230100UV.Medicine  6.6. PORX\_MT010120UV01\_LV02.DispenseRequest  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload  6.14. HL7 standard data structures usage examples  9.1.3.3. Get recipe data  9.2.1.1. Book medication dispense  9.2.1.2. Validate medication dispense  9.2.1.3. Register medication dispense |
| POR-19 E-recepšu izrakstīšanas uzlabojumi sarakstā (Obligāta) | 6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload  6.14. HL7 standard data structures usage examples  9.1.1.6. Book recipe number  9.1.3.3. Get recipe data |
| POR-20 Izsniedzamo ĀL saraksta uzlabojumi (Obligāta) | 6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload  6.14. HL7 standard data structures usage examples |
| POR-23 ĀL izsniegšanas uzlabojumi (Obligāta) | 6.6. PORX\_MT010120UV01\_LV02.DispenseRequest  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload  6.14. HL7 standard data structures usage examples  9.1.3.3. Get recipe data  9.2.1.1. Book medication dispense  9.2.1.2. Validate medication dispense  9.2.1.3. Register medication dispense |
| POR-24 ĀL izsniegšanas vēlamie uzlabojumi (Vēlama) | 6.1. PORX\_MT010120UV01\_LV02.CombinedMedicationRequest  6.7. PORX\_MT020070UV01\_LV02.CombinedMedicationDispense  6.14. HL7 standard data structures usage examples |
| POR-26 Ekstemporāli pagatavojamo zāļu un bezrecepšu zāļu receptes (Obligāta) | 6.3. COCT\_MT230100UV.Medicine  6.13. PORX\_MT000007UV01\_LV02.QueryByParameterPayload  6.14. HL7 standard data structures usage examples |

# Services usage explanation

## Physician processes

Physician goal is to operate with recipes (also known as prescriptions, medication orders or combined medication requests), including its prescription, reservation, cancelation and getting list of recipes.

### Recipe prescription

Recipe prescription is complex process, which consists of such steps as booking recipe number, getting different kind of information during the prescription process (medicines used by patient, frequently used diagnoses, frequently prescribed medicines by physician or to patient) and recipe prescription.

Getting different kind of information during the prescription process is optional; it should be used to provide physician with additional information to eliminate the risk of mistake during prescription process. To submit recipe data to System, there must be called at least 2 mandatory services:

1. BookMedicationOrder – to book and recieve recipe number;
2. RegisterMedicationOrder – to submit recipe with appropriate recipe number.

Further, there are listed most common steps during recipe prescription process.

#### Get patient potentially used medicines

Optional step

Step provides physician with information on patient potentially used medicines ordered by prescription date. These medicines considered to be:

* Partly or totally dispensed medicines based on recipes with valid expiration date;
* Partly or totally dispensed medicines based on recipes with valid or reciently ended treatment course.

To get patient potentially used medicines, use service *GetMedicationOrderList* with parameters:

* patient = [Patient person code] (in example: 01018211119);
* potentiallyAdministrableInd = true (return potentially administrable medicines only);
* scope = PTN, role = SBJ (get recipes where patient is recipe subject);
* retrieve = ORD.MED (fulfill medication data).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<potentiallyAdministrableInd value="true"/>

<scope>PTN</scope>

<role>SBJ</role>

<retrieve>ORD.MED</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get frequently used physician diagnoses

Optional step

Step provides physician with information on frequently used diagnoses by physician during last X months (X – System parameter) ordered by prescription frequency ascending and date descending.

To get frequently used physician diagnoses, use service *GetDiagnosisList* with no parameters:

* scope = USR, role = AUT (get recipes where user is author).

Input data example:

<parameterList>

<scope>USR</scope>

<role>AUT</role>

</parameterList>

Output data example (list with two diagnoses: Z25.1, J10.0):

<diagnosis code="Z25.1" codeSystem="1.3.6.1.4.1.38760.2.159" displayName="Nepieciešamība imunizēt pret gripu"/>

<diagnosis code="J10.0" codeSystem="1.3.6.1.4.1.38760.2.159" displayName="Gripa ar pneimoniju, ja vīruss identificēts"/>

#### Get medicines frequently prescribed to patient

Optional step

Step provides physician with information on medicines frequently prescribed to appropriate patient during last X months (X – System parameter) ordered by prescription frequency ascending and date descending.

To get medicines frequently prescribed to patient, use service *GetMedicineList* with parameters:

* patient = [Patient person code] (in example: 01018211119);
* scope = PTN, role = SBJ (get recipes where patient is recipe subject).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<scope>PTN</scope>

<role>SBJ</role>

</parameterList>

Output data example (list of two medicines: 01-0294, 03-0498):

<medicine classCode="MMAT" determinerCode="INSTANCE">

<code code="01-0294" codeSystem="1.3.6.1.4.1.38760.2.136" displayName="Singulair 4 mg chewable tablets"/>

<desc>Singulair 4 mg košļājamās tabletes</desc>

</medicine>

<medicine classCode="MMAT" determinerCode="INSTANCE">

<code code="03-0498" codeSystem="1.3.6.1.4.1.38760.2.136" displayName="Anzatax 30 mg/5 ml concentrate for solution for infusion"/>

<desc>Anzatax 30 mg/5 ml koncentrāts infūziju šķīduma pagatavošan</desc>

</medicine>

#### Get medicines frequently prescribed by physician

Optional step

Step provides physician with information on medicines frequently prescribed by him patient during last X months (X – System parameter) ordered by prescription frequency ascending and date descending.

To get medicines frequently prescribed by physician, use service *GetMedicineList* with parameters:

* scope = USR, role = AUT (get recipes where user is author).

Input data example:

<parameterList>

<scope>USR</scope>

<role>AUT</role>

</parameterList>

Output data is list of medicine data structures with appropriate data fulfilled; full medicine data structure example shown in chapter “9.1.1.3 Get medicines frequently prescribed to patient” as output data example.

#### Get compensation conditions

Optional step

Step provides physician with information on compensation conditions for requested data (patient data, medication, physician specialty and diagnoses).

To get compensation conditions, use service *GetCompensationConditionList* with parameters:

* effectiveTime = [Today or other date (yyyyMMddHHmmss.ffffzz00)];
* patientId = [Patient person code] (in example: 01018211119)
* patientBirthDate = [Patient birth date (yyyyMMddHHmmss.ffffzz00)];
* patientGenderCode = [Patient gender code (male = V, female = S)];
* diagnosisCode = [Appropriate diagnosis code (ICD-10 code)] (in example: Z25.1 – diagnosis, Z33.0 – additional diagnosis);
* physicianSpecialtyCode = [specialty code physician with which he is working at the moment (logged in)] (in example: A161);
* medicine.code = [Medicine code] (in example: 05-0197);
* medicine.formCode = [Medicine form code] (in example: empty).

Input data example:

<parameterList>

<effectiveTime value="20121122000000.0000-0800"/>

<patientId root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<patientBirthDate value="19471002000000.0000-0700"/>

<patientGenderCode code="S" codeSystem="1.3.6.1.4.1.38760.2.111"/>

<diagnosisCode code="Z25.1" codeSystem="1.3.6.1.4.1.38760.2.159">

<qualifier>

<value code="Z33.0" codeSystem="1.3.6.1.4.1.38760.2.159"/>

</qualifier>

</diagnosisCode>

<physicianSpecialtyCode code="A161" codeSystem="1.3.6.1.4.1.38760.2.38"/>

<medicine classCode="MMAT" determinerCode="INSTANCE">

<code code="05-0197" codeSystem="1.3.6.1.4.1.38760.2.136"/>

</medicine>

</parameterList>

Output data example (compensation code: 3, percentage: 50, special conditions: “grūtniecēm (receptē papildus norādot diagnozes kodu ""Z33"")”):

<compensationCondition>

<code code="3" codeSystem="1.3.6.1.4.1.38760.2.152" displayName="Z25 - Vaccinum influenzae"/>

<compensationPercent value="50"/>

<conditions>grūtniecēm (receptē papildus norādot diagnozes kodu ""Z33"")</conditions>

</compensationCondition>

#### Book recipe number

Mandatory step

Step provides recipe number booking for further data submitting to System. It serves like transaction mechanism to avoid network errors during recipe prescription.

To book recipe number, use service *BookMedicationOrders* with parameters:

* count = 1 (to book one recipe number);
* permanentInd = false (book temporary recipe number).

Input data example:

<bookMedicationOrderRequest>

<count value="1"/>

<permanentInd value="false"/>

</bookMedicationOrderRequest>

Output data example (empty recipe data structure with transcriber data fulfilled):

<combinedMedicationRequest moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<effectiveTime xsi:type="IVL\_TS">

<low value="20121122000000.0000+0200"/>

<high value="20130222000000.0000+0200"/>

</effectiveTime>

<statusCode code="new"/>

<transcriber typeCode="TRANS">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Tatjana</given>

<family>Farbtuha</family>

</name>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="409635213"/>

<name use="L">Viesturu doktorāts</name>

</representedOrganization>

</assignedEntity>

</transcriber>

</combinedMedicationRequest>

#### Prescribe recipe

Mandatory step

Step submits all the recipe data to System and does necessary validations.

To prescribe recipe, use service *RegisterMedicationOrder* with all the gathered recipe information, including recipe number got from *BookMedicationOrders* response.

Input and output data are full recipe data structures; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

### Recipe reservation

Recipe reservation allows booking one or more recipes permanently for further usage at home visit to patients. Recipe reservation allows printing out recipe numbers on paper form for further manipulation with it.

#### Recipe reservation

To reserve recipe permanently for its further usage at home visit to patients, use service *BookMedicationOrders* with parameters:

* count = 1 (reserve one recipe);
* permanentInd = true (recipe reservation create permanent record in database, because it returns recipe number, which is printed on recipe form).

Input data example:

<bookMedicationOrderRequest>

<count value="1"/>

<permanentInd value="true"/>

</bookMedicationOrderRequest>

Output data is empty recipe data structure with transcriber data fulfilled; empty recipe data structure example shown in chapter “9.1.1.6 Book recipe number” as output data example.

### Get list of recipes

System services provide elastic way of getting data by different parameters, for example getting patient recipes or recipes prescribed by physician.

Further, there are listed most common which physician can be interested in.

#### Get recipes prescribed by physician

To get recipes prescribed by physician ordered by prescription date discending, use service *GetMedicationOrderList* with parameters:

* scope = USR, role = AUT (get recipes where user is author);
* retrieve = ORD.DGN, ORD.PTN, ORD.MED (fulfill diagnosis, patient, medication data).

Input data example:

<parameterList>

<scope>USR</scope>

<role>AUT</role>

<retrieve>ORD.DGN</retrieve>

<retrieve>ORD.PTN</retrieve>

<retrieve>ORD.MED</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get patient recipes

To get patient recipes ordered by prescription date discending, use service *GetMedicationOrderList* with parameters:

* patient = [Patient person code] (in example: 01018211119 – if passed, only this patient recipes will be returned; if no patient passed – all the appropriate patients recipes will be returned);
* scope = PTN, role = SBJ (get recipes where patient is recipe subject);
* retrieve = ORD.DGN, ORD.AUT, ORD.MED (fulfill diagnosis, physician, medication data).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<scope>PTN</scope>

<role>SBJ</role>

<retrieve>ORD.DGN</retrieve>

<retrieve>ORD.AUT</retrieve>

<retrieve>ORD.MED</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get recipe data

To get specific recipe data by recipe number, use service *GetMedicationOrderData* with parameters:

* id = [recipe number] (in example: 23345349511101265).

Input data example:

<parameterList>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

</parameterList>

Output data example (full recipe data structure):

<combinedMedicationRequest moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<effectiveTime xsi:type="IVL\_TS">

<low value="20121122000000.0000+0200"/>

<high value="20130222000000.0000+0200"/>

</effectiveTime>

<statusCode code="active"/>

<fulfillmentStatusCode code="unfulfilled"/>

<subject typeCode="SBJ">

<patient classCode="PAT">

<patientPerson>

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<name use="L">

<given>Pēteris</given>

<family>Liepiņš</family>

</name>

<administrativeGenderCode code="V" codeSystem="1.3.6.1.4.1.38760.2.111" displayName="Vīrietis"/>

<birthTime value="19820101000000.0000+0200"/>

<addr><streetName>Bišu iela</streetName> <houseNumber>3</houseNumber>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country></addr>

</patientPerson>

</patient>

</subject>

<directTarget typeCode="DIR">

<medication classCode="ADMM">

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="05-0604" codeSystem="1.3.6.1.4.1.38760.2.136" displayName="Carboplatin "Ebewe" 10 mg/ml concentrate for solution for infusion"/>

<name>Carboplatin "Ebewe" 10 mg/ml</name>

<desc>Carboplatin "Ebewe" 10 mg/ml koncentrāts infūziju šķīduma p</desc>

<formCode code="150" codeSystem="1.3.6.1.4.1.38760.2.137" displayName="Koncentrāts infūziju šķīduma pagatavošanai"/>

</administrableMedicine>

</medication>

</directTarget>

<author typeCode="AUT">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

<id root="1.3.6.1.4.1.38760.3.1.4" extension="10640008696"/>

<id root="1.3.6.1.4.1.38760.2.1" extension="01015110638"/>

<telecom value="tel:29292929"/>

<telecom value="mailto:vards.uzvards@iestade.lv"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Tatjana</given>

<family>Farbtuha</family>

</name>

<asLicensedEntity classCode="LIC">

<code code="A161" codeSystem="1.3.6.1.4.1.38760.2.38" displayName="onkoloģijas ķīmijterapeits"/>

</asLicensedEntity>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="409635213"/>

<id root="1.3.6.1.4.1.38760.2.23" extension="409635213"/>

<name use="L">Viesturu doktorāts</name>

<addr><additionalLocator>Ziedkalni-8</additionalLocator>, <city>Grenči</city>, <county>Zemītes pag.</county>, <county>Kandavas nov.</county>, <postalCode>LV-3136</postalCode>, <country>LV</country></addr>

</representedOrganization>

</assignedEntity>

</author>

<transcriber typeCode="TRANS">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Tatjana</given>

<family>Farbtuha</family>

</name>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="409635213"/>

<name use="L">Viesturu doktorāts</name>

</representedOrganization>

</assignedEntity>

</transcriber>

<coverage typeCode="COVBY">

<compensationRequest moodCode="RQO">

<substitutionReason nullFlavor="UNC">

<originalText>Aizvietošanas pamatojums</originalText>

</substitutionReason>

</compensationRequest>

</coverage>

<component1 typeCode="COMP">

<substanceAdministrationRequest classCode="SBADM" moodCode="RQO">

<text>Dzert trīs tabletes pirms katras ēdienreizes.</text>

<effectiveTime xsi:type="IVL\_TS">

<width value="2" unit="wk"/>

</effectiveTime>

</substanceAdministrationRequest>

</component1>

<component2 typeCode="COMP">

<dispenseRequest classCode="SPLY" moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.2" extension="ABC123"/>

<effectiveTime xsi:type="IVL\_TS">

<low value="20121122000000.0000+0200"/>

<high value="20130222000000.0000+0200"/>

</effectiveTime>

<quantity value="10" unit="ml" />

<receiver typeCode="RCV">

<assignedPerson classCode="ASSIGNED">

<id extension="01015110638" root="1.3.6.1.4.1.38760.3.1.1"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name>

<given>Vārds</given>

<family>Uzvārds</family>

</name>

</assignedPerson>

</assignedPerson>

</receiver>

<specialFormInd value="false"/>

<treatmentCourseInd value="false"/>

<remainingQuantity value="10" unit="ml" />

</dispenseRequest>

</component2>

<subjectOf4 typeCode="SUBJ">

<substitutionPermission classCode="SUBST" moodCode="PERM">

<code code="N"/>

<reasonCode nullFlavor="UNC">

<originalText>ĀL aizvietošanas aizlieguma pamatojums</originalText>

</reasonCode>

</substitutionPermission>

</subjectOf4>

</combinedMedicationRequest>

### Cancel recipe

System provides to cancel recipe by its number, if it is necessary. Before cancelling recipe, it is important to make sure it is recipe to be canceled. For this purpose, there can be called service *GetMedicationOrderData*, which returns all the recipe data.

#### Get recipe

To get all the recipe data for make sure it is recipe to be canceled, use service *GetMedicationOrderData* with parameters:

* id = [recipe number] (in example: 23345349511101265).

Input data example:

<parameterList>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

</parameterList>

Output data is full recipe data structure; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Cancel recipe

To cancel recipe and add cancellation information to recipe (reason, person, etc), use service *CancelMedicationOrder* with parameters:

* id = [recipe number] (in example: 23345349511101265);
* author.assignedEntity.id = [physician code] (in example: 01015110638);
* effectiveTime = [Today or other date (yyyyMMddHHmmss.ffffzz00)];
* reason = [Cancellation reason] (in example: ERR).

Input data example:

<cancelMedicationOrderRequest>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<author>

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

</assignedEntity>

</author>

<effectiveTime value="20121122000000.0000-0800"/>

<reason code="ERR" codeSystem="1.3.6.1.4.1.38760.2.300"/>

</cancelMedicationOrderRequest>

No specific output data.

## Pharmacist services

Pharmacist goal is to operate with recipe dispensing (also known as dispense, medication dispense or combined medication dispense), including its booking recipe for dispense, dispensing medication, cancelling dispense, importing paper recipe data to System for getting compensation repayment, prescribed medications to pharmacy and dispensed medications from pharmacy.

### Medications dispense and dispense cancellation

Medication dispense is complex process, which consists of booking medication dispense and registration or cancelling of medication dispense.

To submit dispense data to System, there must be called at least 2 mandatory services:

1. BookMedicationDispense – to book and recieve dispense number;
2. RegisterMedicationDispense – to submit dispense with appropriate dispense number.

#### Book medication dispense

Mandatory step

Step provides dispense number booking for further data submitting to System. It serves like transaction mechanism to avoid network errors during process.

To book medication dispense, use service *BookMedicationDispense* with parameters:

* id = [recipe number] (in example: 23345349511101265);
* effectiveTime = [Today or other date (yyyyMMddHHmmss.ffffzz00)].

Input data example:

<bookMedicationDispenseRequest>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<effectiveTime value="20121122000000.0000-0800"/>

</bookMedicationDispenseRequest>

Output data example (empty medication dispense data structure with transcriber and full recipe data fulfilled):

<combinedMedicationDispense>

<id root="1.3.6.1.4.1.38760.3.4.11.3" extension="27656621496472507"/>

<effectiveTime xsi:type="IVL\_TS">

<low value="20121122000000.0000+0200"/>

<high value="20130222000000.0000+0200"/>

</effectiveTime>

<transcriber typeCode="TRANS">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01014511827"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Iraīda</given>

<family>Širovska</family>

</name>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="41503015136"/>

<id root="1.3.6.1.4.1.38760.2.2.5" extension="60290"/>

<name use="L">VSabiedrība ar ierobežotu atbildību “ESPLANĀDE FARM” aptieka Nr.1</name>

</representedOrganization>

</assignedEntity>

</transcriber>

<inFulfillmentOf>

<combinedMedicationRequest moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<statusCode code="active"/>

<fulfillmentStatusCode code="unfulfilled"/>

<subject typeCode="SBJ">

<patient classCode="PAT">

<patientPerson>

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<name use="L">

<given>Pēteris</given>

<family>Liepiņš</family>

</name>

<administrativeGenderCode code="V" codeSystem="1.3.6.1.4.1.38760.2.111"/>

<birthTime value="19820101000000.0000+0200"/>

<addr><streetName>Bišu iela</streetName> <houseNumber>3</houseNumber>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country><censusTract>0500201</censusTract></addr>

</patientPerson>

</patient>

</subject>

<directTarget typeCode="DIR">

<medication classCode="ADMM">

<administrableMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="01-0294" codeSystem="1.3.6.1.4.1.38760.2.136"/>

<name>Singulair 4 mg chewable tablets</name>

<formCode code="141" codeSystem="1.3.6.1.4.1.38760.2.137" displayName="Košļājamās tabletes"/>

</administrableMedicine>

</medication>

</directTarget>

<author typeCode="AUT">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

<id root="1.3.6.1.4.1.38760.3.1.4" extension="10640008696"/>

<id root="1.3.6.1.4.1.38760.2.1" extension="01015110638"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Tatjana</given>

<family>Farbtuha</family>

</name>

<asLicensedEntity classCode="LIC">

<code code="A161" codeSystem="1.3.6.1.4.1.38760.2.38" displayName="onkoloģijas ķīmijterapeits"/>

</asLicensedEntity>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="409635213"/>

<id root="1.3.6.1.4.1.38760.2.23" extension="409635213"/>

<name use="L">Viesturu doktorāts</name>

</representedOrganization>

</assignedEntity>

</author>

<transcriber typeCode="TRANS">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01015110638"/>

<id root="1.3.6.1.4.1.38760.3.1.4" extension="10640008696"/>

<id root="1.3.6.1.4.1.38760.2.1" extension="01015110638"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name xsi:type="PN" use="L">

<given>Tatjana</given>

<family>Farbtuha</family>

</name>

</assignedPerson>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.2" extension="409635213"/>

<id root="1.3.6.1.4.1.38760.2.23" extension="409635213"/>

<name use="L">Viesturu doktorāts</name>

</representedOrganization>

</assignedEntity>

</transcriber>

<component1 typeCode="COMP">

<substanceAdministrationRequest classCode="SBADM" moodCode="RQO">

<text>Dzert trīs tabletes pirms katras ēdienreizes.</text>

<effectiveTime xsi:type="IVL\_TS">

<width value="2" unit="wk"/>

</effectiveTime>

</substanceAdministrationRequest>

</component1>

<component2 typeCode="COMP">

<dispenseRequest classCode="SPLY" moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.2" extension="ABC123"/>

<effectiveTime xsi:type="IVL\_TS">

<low value="20121122000000.0000+0200"/>

<high value="20130222000000.0000+0200"/>

</effectiveTime>

<quantity value="10" unit="ml"/>

<specialFormInd value="false"/>

<treatmentCourseInd value="false"/>

<remainingQuantity value="10" unit="ml" />

</dispenseRequest>

</component2>

</combinedMedicationRequest>

</inFulfillmentOf>

<component1 typeCode="COMP">

<substitutionMade classCode="SUBST" moodCode="EVN">

<code code="E"></code>

<reasonCode nullFlavor="UNC">

<originalText>ĀL aizvietošanas pamatojums</originalText>

</reasonCode>

</substitutionMade>

</component1>

<component3>

<supplyEvent>

<statusCode code="new"/>

<effectiveTime value="20121122000000.0000+0200"/>

<receiver typeCode="RCV">

<assignedPerson classCode="ASSIGNED">

<id extension="01014511827" root="1.3.6.1.4.1.38760.3.1.1"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name>

<given>Vārds</given>

<family>Uzvārds</family>

</name>

</assignedPerson>

</assignedPerson>

</receiver>

</supplyEvent>

</component3>

</combinedMedicationDispense>

#### Validate medication dispense

Conditional step

Before register medication dispense, one can validate all the filled data. To validate medication dispense, use service *ValidateMedicationDispnese* with all the gathered dispense information, including dispense number got from *BookMedicationDispense* response.

Input data is full medication dispense data; full medication dispense data structure example shown in chapter “9.2.1.3. Register medication dispense” as input data example.

No specific output data.

#### Register medication dispense

Conditional step

Step submits all the dispense data to System and does necessary validations.

To register medication dispense, use service *RegisterMedicationDispense* with all the gathered dispense information, including dispense number got from *BookMedicationDispense* response.

Input data example:

<combinedMedicationDispense>

<id root="1.3.6.1.4.1.38760.3.4.11.3" extension="27656621496472507"/>

<performer typeCode="PRF">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01014511827"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<asLicensedEntity classCode="LIC">

<code code="F-0324" codeSystem="1.3.6.1.4.1.38760.2.47"/>

</asLicensedEntity>

</assignedPerson>

<telecom value="tel:29292929"/>

<telecom value="mailto:vards.uzvards@iestade.lv"/>

<representedOrganization classCode="ORG" determinerCode="INSTANCE">

<id root="1.3.6.1.4.1.38760.3.2.5" extension="60290"/>

</representedOrganization>

</assignedEntity>

</performer>

<inFulfillmentOf>

<combinedMedicationRequest moodCode="RQO">

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<fulfillmentStatusCode code="fulfilled"/>

</combinedMedicationRequest>

</inFulfillmentOf>

<component1 typeCode="COMP">

<substitutionMade classCode="SUBST" moodCode="EVN">

<code code="E"></code>

<reasonCode nullFlavor="UNC">

<originalText>ĀL aizvietošanas pamatojums</originalText>

</reasonCode>

</substitutionMade>

</component1>

<component3>

<supplyEvent>

<effectiveTime value="20121122000000.0000+0200"/>

<quantity>

<translation value="0.3571"/>

</quantity>

<consumable>

<content classCode="CONT">

<containedMedicine classCode="MMAT" determinerCode="INSTANCE">

<code code="01-0294-03" codeSystem="1.3.6.1.4.1.38760.2.144"/>

</containedMedicine>

</content>

</consumable>

<receiver typeCode="RCV">

<assignedPerson classCode="ASSIGNED">

<id extension="01014511827" root="1.3.6.1.4.1.38760.3.1.1"/>

<assignedPerson classCode="PSN" determinerCode="INSTANCE">

<name>

<given>Vārds</given>

<family>Uzvārds</family>

</name>

</assignedPerson>

</assignedPerson>

</receiver>

<price value="28.13"/>

<totalAmount value="10.05"/>

<paymentAmount value="10.05"/>

<compensatedAmount value="0"/>

</supplyEvent>

</component3>

<component4>

<sociallySupportedInd value="false"/>

</component4>

</combinedMedicationDispense>

Output data is full medication dispense data structures as in input data example, including full recipe data substructure.

#### Cancel medication dispense

Conditional step

Step cancels dispense process, it must be called after *BookMedicationdispense* service to cancel dispense process and clear unnecessary dispense numbers from System.

To cancel medication dispense, use service *BookMedicationDispense* with parameters:

* medicationOrderId = [recipe number] (in example: 23345349511101265);
* medicationDispenseId = [dispense number, got from *BookMedicationDispense* response] (in example: 27656621496472507).

Input data example:

<cancelMedicationDispenseRequest moodCode="RQO">

<medicationOrderId root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

<medicationDispenseId root="1.3.6.1.4.1.38760.3.4.11.3" extension="27656621496472507"/>

</cancelMedicationDispenseRequest>

No specific output data.

### Paper recipe importing

Pharmacist can import paper only recipe to System, providing registration of paper recipe to System and bar code printout on paper recipe. Taking into account that during the paper recipe import process, pharmacist must fulfill all the data according to paper recipe, which can be written wrong, there is ability to edit imported paper recipe for pharmacist. Editing differs from importing only with recipe number passed to service.

#### Get paper recipe data and number

To get paper recipe data and its number, use service *GetMedicationOrderData* with parameters:

* id = [recipe number] (in example: 23345349511101265).

Input data example:

<parameterList>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

</parameterList>

Output data is full recipe data structure; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Import or edit paper recipe data

To import or edit paper recipe data, use service *RegisterMedicationOrder* with all the gathered recipe information, no need to query *BookMedicationOrders* before.

Input and output data are list of recipe data structures; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

### Pharmacy medication

Within e‑Health project, patient is able to set its favorite pharmacy. This allows pharmacies to plan medication purchase. Additionally, pharmacy can get all the medications dispensed from it.

#### Get medications prescribed to pharmacy

To get all the medications prescribed to appropriate pharmacy, use service *GetMedicationOrderList* with parameters:

* prescription.low = [Date from which get medications in format (yyyyMMddHHmmss.ffffzz00)];
* prescription.high = [Date till which get medications in format (yyyyMMddHHmmss.ffffzz00)];
* potentiallyFulfillableInd = true (return potentially fulfilable medications only – based on medication orders, which are not expired and not dispensed or partly dispensed);
* scope = ORG (get recipes for organization pharmacy);
* retrieve = ORD.DIS, ORD.MED (fulfill dispense request and medication data).

Input data example:

<parameterList>

<prescriptionTime>

<low value="20130101000000.0000+0200"/>

<high value="20130201000000.0000+0200"/>

</prescriptionTime>

<potentiallyFulfillableInd value="true"/>

<scope>ORG</scope>

<retrieve>ORD.DIS</retrieve>

<retrieve>ORD.MED</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get medications dispensed from pharmacy

To get all the medications dispensed form appropriate pharmacy, use service *GetMedicationDispenseList* with parameters:

* prescription.low = [Date from which get medications in format (yyyyMMddHHmmss.ffffzz00)];
* prescription.high = [Date till which get medications in format (yyyyMMddHHmmss.ffffzz00)];
* scope = ORG (get recipes for organization pharmacy).

Input data example:

<parameterList>

<dispenseTime>

<low value="20130101000000.0000+0200"/>

<high value="20130201000000.0000+0200"/>

</dispenseTime>

<scope>ORG</scope>

</parameterList>

Output data is list of medication dispense data structures with appropriate data fulfilled; full medication dispense data structure example shown in chapter “9.2.1.3 Register medication dispense” as output data example.

### Patient recipes

System provides service for getting patient recipes by patient person code.

Note:

This is demonstration process only; it shows the opportunities of centralized System. This process will be able to use legally for dispensing only after complete dematerialization of recipes in Latvia, for now pharmacist must demand paper recipe from patient and dispensing medication by it.

#### Patient recipes by person code (demo)

To get patient recipes by person code ordered by prescription date, use service *GetMedicationOrderList* with parameters:

* patient = [Patient person code] (in example: 01018211119);
* scope = PTN, role = SBJ, DLG (get recipes where patient is recipe subject or delegate);
* retrieve = ORD.DGN, ORD.AUT, ORD.MED, DIS.ALL (fulfill diagnosis, physician, medication, dispense data).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<scope>PTN</scope>

<role>SBJ</role>

<role>DLG</role>

<retrieve>ORD.DGN</retrieve>

<retrieve>ORD.AUT</retrieve>

<retrieve>ORD.MED</retrieve>

<retrieve>DIS.ALL</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

## Patient services

Patient can observe his and his delegates recipes, also providing his favorite pharmacy, from where patient consider getting medications. Patient pharmacy can be used by pharmacies to plan medication purchase.

### Patients recipes

Patient can observe his recipes and its’ dispenses, additionally he can observe his delegates recipes (children etc).

#### Get patient recipes

To get patient recipes ordered by prescription date, use service *GetMedicationOrderList* with parameters:

* patient = [Patient person code] (in example: 01018211119);
* potentiallyFulfillableInd = true (return potentially fulfilable medicines only– based on medication orders, which are not expired and not dispensed or partly dispensed);
* scope = USR, role = SBJ, DLG (get recipes where patient is recipe subject or delegate);
* retrieve = ORD.DGN, ORD.AUT, ORD.MED, DIS.ALL (fulfill diagnosis, physician, medication, dispense data).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<potentiallyFulfillableInd value="true"/>

<scope>USR</scope>

<role>SBJ</role>

<role>DLG</role>

<retrieve>ORD.DGN</retrieve>

<retrieve>ORD.AUT</retrieve>

<retrieve>ORD.MED</retrieve>

<retrieve>DIS.ALL</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get recipe data

To get all the information of recipe, use service *GetMedicationOrderData* with parameters:

* id = [recipe number] (in example: 23345349511101265).

Input data example:

<parameterList>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

</parameterList>

Output data is full recipe data structure; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

### Patient pharmacy

Patient can set or get his favorite pharmacy to or from his user profile.

#### Get pharmacy

To get pharmacy from patient user profile, use service *GetProfile* with parameters:

* patient = [Patient person code] (in example: 01018211119).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

</parameterList>

Output data example (full profile data structure):

<profileSetupRequest moodCode="RQO">

<subject typeCode="SBJ">

<patient classCode="PAT">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

</patient>

</subject>

<directTarget typeCode="DIR">

<profile>

<pharmacy root="1.3.6.1.4.1.38760.2.134" extension="60290"/>

</profile>

</directTarget>

</profileSetupRequest>

#### Set pharmacy

To set new pharmacy to patient user profile, use service *SetProfile* with parameters:

* subject.patientid = [Patient person code] (in example: 01018211119);
* directTarget.profile.pharmacy = [Pharmacy code] (in example: 123456).

Input data is full profile data structure; full profile data structure example shown in chapter “9.3.2.1 Get pharmacy” as output data example.

No specific output data.

## Supervisor institution services

Supervisor institution goal is to supervise and control recipe chain process, by identifying wrong recipes and cancelling them, managing warnings on medications, getting patients data in emergency cases (when there has been prescribed medications which are cancelled).

### Recipe management

Supervisor institution can get list of recipes, filter them and search by different parameters to find required information and recipes. If required, supervisor institution can cancel recipe.

#### Get recipes

To get all the available recipes, use service *GetMedicationOrderList* with parameters (attention: to avoid unnecessary load on System, use more precise filter parameters, for example, specific patient etc.):

* subject.patientid = [Patient person code] (in example: 01018211119);
* scope = ALL (get all recipes);
* retrieve = ORD.DGN, ORD.AUT, ORD.MED, DIS.ALL (fulfill diagnosis, physician, medication, dispense data).

Input data example:

<parameterList>

<patient root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<scope>ALL</scope>

<retrieve>ORD.DGN</retrieve>

<retrieve>ORD.AUT</retrieve>

<retrieve>ORD.MED</retrieve>

<retrieve>DIS.ALL</retrieve>

</parameterList>

Output data is list of recipe data structures with appropriate data fulfilled; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Get recipe data

To get all the recipe information, use service *GetMedicationOrderData* with parameters:

* id = [recipe number] (in example: 23345349511101265).

Input data example:

<parameterList>

<id root="1.3.6.1.4.1.38760.3.4.11.1" extension="23345349511101265"/>

</parameterList>

Output data is full recipe data structure; full recipe data structure example shown in chapter “9.1.3.3 Get recipe data” as output data example.

#### Cancel recipe

To cancel recipe and add cancellation information to recipe (reason, person, etc), use service *CancelMedicationOrder* with parameters:

* subject.combinedMedicationRequest.id = [recipe number];
* author.assignedEntity.id = [person code];
* effectiveTime = [Today];
* reason = [Cancellation reason].

Input data is the same as in chapter “9.1.4.2 Cancel recipe”.

No specific output data.

### Medication management

Supervisor institution can manage medication warnings by registering them. If required, supervisor institution can get patients contact information to whom there were prescribed medications, which are cancelled or dangerous, for further actions (sending mails, calling etc.).

#### Get medication warnings

To get medication warnings for its review and managing, use service *GetMedicationWarningList* with parameters:

* prescription.low = [Date from which get warnings in format (yyyyMMddHHmmss.ffffzz00)];
* prescription.high = [Date till which get warnings in format (yyyyMMddHHmmss.ffffzz00)];
* status = [Warning status] (in example: active).

Input data example:

<parameterList>

<effectiveTime>

<low value="20130101000000.0000+0200"/>

<high value="20130201000000.0000+0200"/>

</effectiveTime>

<status code="active"/>

</parameterList>

Output data is list of full medication warning data structures; full medication warning data structure example shown in chapter “9.4.2.2 Register medication warning” as input data example.

#### Register medication warning

To register new medication warning, use service *RegisterMedicationWarning* with parameters:

* medicineCode = [Madicine code to register warning to] (in example: 99-0863);
* warning = [Warning text] (in example: Dangerous medicine);
* eventCode = [Event code to show for prescriptors and dispensers] (in example: REG, DSP);
* effectiveTime = [Tomorrow];
* author.assignedEntity.id = [Person code].

Input data example:

<medicationWarning moodCode="RQO">

<medicineCode code="99-0863" codeSystem="1.3.6.1.4.1.38760.2.136"/>

<warning>Dangerous medicine</warning>

<eventCode>REG</eventCode>

<eventCode>DSP</eventCode>

<effectiveTime value="20120810000000.0008+0300"/>

<author typeCode="AUT">

<assignedEntity classCode="ASSIGNED">

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

</assignedEntity>

</author>

</medicationWarning>

No specific output data.

#### Get patients contact information by medication

To get patients contact information by specific medication, use service *GetPatientContactList* with parameters:

* prescribedMedicine.code = [Medicine code to get by] (in example: 99-0863)
* scope = ALL (get all recipes).

Input data example:

<parameterList>

<prescribedMedicine classCode="MMAT" determinerCode="KIND">

<code code="99-0863" codeSystem="1.3.6.1.4.1.38760.2.136"/>

</prescribedMedicine>

<scope>USR</scope>

</parameterList>

Output data example (list of two persons with ids: 01018211119, 02024222206):

<person>

<id root="1.3.6.1.4.1.38760.3.1.1" extension="01018211119"/>

<name use="L">

<given>Andris</given>

<family>Bērziņš</family>

</name>

<administrativeGenderCode code="V" codeSystem="1.3.6.1.4.1.38760.2.111"/>

<birthTime value="19820810000000.0000+0300"/>

<addr><streetName>Bišu iela</streetName> <houseNumber>3</houseNumber>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country</addr>

</person>

<person>

<id root="1.3.6.1.4.1.38760.3.1.1" extension="02024222206"/>

<name use="L">

<given>Anna</given>

<family>Kalniņa</family>

</name>

<administrativeGenderCode code="S" codeSystem="1.3.6.1.4.1.38760.2.111"/>

<birthTime value="19420202000000.0000+0300"/>

<addr><streetName>Bišu iela</streetName> <houseNumber>3</houseNumber>, <city>Gulbene</city>, <county>Gulbenes nov.</county>, <postalCode>LV-4401</postalCode>, <country>LV</country>addr>

</person>

# Appendixes

## Error list

Errors are divided into two groups:

* Base errors – can occur in any of services;
* Services errors – service specific errors.

Table 101 Error list

|  |  |
| --- | --- |
| # | Error text |
| **Common errors** | |
| **System errors** | |
| 1 | System error. |
| 2 | Outer system error. |
| **Request validation errors** | |
| 100 | Request sent to the wrong IS. |
| 101 | Invalid query ID or query continuation expired. |
| **Security token validations** | |
| 111 | Person specified in security token can not be found in physician registry. |
| 112 | Organization specified in security token can not be found in medical institution registry. |
| 113 | Person specified in security token can not be found in pharmacist registry. |
| 114 | Organization specified in security token can not be found in pharmacy registry. |
| 115 | Person specified in security token doesn’t represent organization specified in security token. |
| **Data enterer validations** | |
| 121 | Person specified in data enterer structure can not be found in physician registry. |
| 122 | Organization specified in data enterer structure can not be found in medical institution registry. |
| 123 | Person specified in data enterer structure can not be found in pharmacist registry. |
| 124 | Organization specified in data enterer structure can not be found in pharmacy registry. |
| 125 | Person specified in data enterer structure doesn’t represent organization specified in security token. |
| **No permission to execute operation errors** | |
| 200 | No permissions to execute operation. |
| 201 | No permissions to execute operation with specific input data. |
| 202 | No permissions to retrieve object. |
| 203 | No permissions to update object. |
| **Incorrect input data errors** | |
| 300 | Mandatory attribute is missing. |
| 301 | Warning: Attribute will be ignored. |
| 302 | Incorrect attribute value. |
| 303 | Incorrect attribute value: Future date/time specified. |
| 304 | Incorrect attribute value: Specified date/time is too long in the past. |
| 305 | Incorrect attribute value: Invalid time interval specified. |
| 306 | Incorrect attribute value: Invalid identity specified. |
| 307 | Incorrect attribute value: Invalid person name specified. |
| 308 | Incorrect attribute value: Invalid identity sheme (root) specified. |
| 309 | Incorrect attribute value: Invalid classifier sheme (code system) specified. |
| 310 | Incorrect attribute value: Specified value can not be found in classifier. |
| 311 | Incorrect attribute value: Unsupported type. |
| 312 | Incorrect attribute value: Value too long. |
| 313 | Incorrect attribute value: Values is not in URL format. |
| 314 | Incorrect attribute value: Values is not in supported URL format. |
| 315 | Incorrect attribute value: Value too short. |
| 316 | Incorrect attribute value: Multiple conflicting identities specified. |
| **Service errors** | |
| **BookMedicationOrders errors** | |
| 10100 | Booked order limit exceeded. |
| **GetMedicationOrderData errors** | |
| 10200 | e-Rescription not found. |
| **GetCompensationConditionList errors** | |
| 10300 | Specified medication not compensable. |
| 10301 | Specified patient diagnosis does not meet compensation conditions. |
| 10302 | Specified patient age does not meet compensation conditions. |
| 10303 | Specified patient gender does not meet compensation conditions. |
| 10304 | Specified physicians specialty does not meet compensation conditions. |
| 10305 | Common medication name or prescribed medicine substitution reason will be required in prescription. |
| *Get Patient card validations* | |
| 10400 | Patient not found. |
| 10401 | Patient data conflicts with EHR. |
| 10402 | Patient deceased. |
| **RegisterMedicationOrder errors** | |
| 10500 | e-Rescription already registered. |
| 10501 | Narcotic substances can be prescribed only on special prescription form. |
| 10502 | Teratogenic substances can be prescribed only on special prescription form. |
| 10503 | Compensable medication can be prescribed only on special prescription form. |
| 10504 | Long administration period medications can be prescribed only on normal prescription form. |
| 10505 | Administration perdion can not be longer than 12 months. |
| 10506 | Specified presciprion validity date does not comply with regulations. |
| 10507 | Specified medican can not be prescribed for administration period longer than 3 months. |
| 10508 | Prescribed narcotic substance quantity exceeds limit specified by regulations. |
| 10509 | Specified medication can not be prescribed. |
| 10510 | Medication prescription warning. |
| 10512 | Administration period too long. |
| *Author validations* | |
| 10520 | Specified author information conflicts with security token. |
| 10521 | Specified author can not be found in physician registry. |
| 10522 | Organization represented by specified author can not be found in medical institution registry. |
| 10523 | Specified author does not represents specified organization. |
| 10524 | Specified author does not has specified specality. |
| 10525 | Specified author can not prescribe medications. |
| *Compensation validations* | |
| 10530 | Compensated medications can be prescribed for patients with EHR. |
| 10200 | *See GetMedicationOrderData errors* |
| 10300 - 10304 | *See GetCompensationConditionList errors* |
| 10900 - 10915 | *See RegisterMedicationDispense errors* |
| 10400 - 10402 | *Get Patient card validations* |
| **CancelMedicationOrder errors** | |
| 10600 | e-Prescription already cancelled. |
| 10601 | Specified author information conflicts with security token. |
| 10602 | e-Prescription already completed. |
| 10200 | *See GetMedicationOrderData errors* |
| **BookMedicationDispense errors** | |
| 10700 | Can no dispense imported prescriptions. |
| 10701 | e-Prescription cancelled. |
| 10702 | e-Prescription expired. |
| 10703 | e-Prescription fully dispensed. |
| 10704 | e-Prescription blocked for dispension in other pharmacy. |
| 10705 | Patient is deceased. |
| 10706 | e-Prescription not compensable. |
| 10707 | Special dispense conditions. |
| 10708 | Dispense warning. |
| 10200 | *See GetMedicationOrderData errors* |
| 10402 | *See Get Patient card validations* |
| *Get medication dispense validations* | |
| 10800 | Invalid dispense transaction ID. |
| **RegisterMedicationDispense errors** | |
| 10900 | Specified quanity unit does not match quanity unit in prescription. |
| 10901 | Specified total amount does not match patient and compensation amounts. |
| 10902 | Specified total amount does not match package price multifying package amount. |
| 10903 | Invalid substitution certificate date. |
| 10904 | Substitution certificate expired. |
| 10905 | e-Prescription ID doesn’t match reserved one. |
| 10906 | Medication dispense date doesn’t match reserved one. |
| 10911 | e-Prescription not covered by state. |
| 10912 | State coverage percent does not match with specified in e-Prescription. |
| 10913 | Compensated amount does not match coverage percent. |
| 10914 | Compensation amount can be specified only for special prescriptions. |
| 10915 | EHIC certificate issue date in the future. |
| 10916 | Partial dispense can not be performed for “special” prescriptions. |
| 10917 | Only medicine products from classifier "Kompensējamo zāļu saraksts" (1.3.6.1.4.1.38760.2.151) or related products from classifier "Medikamentu saraksts" (1.3.6.1.4.1.38760.2.144) are allowed for state covered dispenses. |
| *Author validations* | |
| 10920 | Specified author information conflicts with security token. |
| 10921 | Specified author can not be found in pharmacist registry. |
| 10922 | Organization represented by specified author can not be found in pharmacy registry. |
| 10923 | Specified author does not represents specified organization. |
| 10924 | Specified author does not has specified specality. |
| 10925 | Specified author can not dispense medications. |
| 10200 | *See GetMedicationOrderData errors* |
| 10800 | *See Get medication dispense validations* |
| **CancelMedicationDispense errors** | |
| 10200 | *See GetMedicationOrderData errors* |
| 10800 | *See Get medication dispense validations* |
| **GetMedicationOrderList** | |
| **GetMedicineList** | |
| **GetDiagnosisList** | |
| **GetMedicationDispenseList** | |
| **GetProfile** | |
| **SetProfile** | |
| **GetPatientContactList** | |
| **RegisterMedicationWarning errors** | |
| 11000 | Medication warning author conflicts with user data. |
| **GetMedicationWarningList** | |
| **ValidateMedicationDispense errors** | |
| 11101 | Medication dispense already cancelled. |
| 11102 | Medication dispense already registered. |

1. W3C (World Wide Web Consortium) ieteiktais standarts sk. <http://www.w3.org/TR/wsdl> [↑](#footnote-ref-1)
2. W3C (World Wide Web Consortium) ieteiktais standarts sk. <http://www.w3.org/XML/Schema> [↑](#footnote-ref-2)
3. W3C (World Wide Web Consortium) ieteiktais standarts sk. <http://www.w3.org/TR/soap/> [↑](#footnote-ref-3)
4. W3C (World Wide Web Consortium) ieteiktais standarts sk. <http://www.w3.org/XML/> [↑](#footnote-ref-4)