

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ
ГЕНЕРАЦІЇ ТА СЕРІАЛІЗАЦІЇ ДАНИХ ЗА СТАНДАРТОМ

HL7/FHIR
(тема)

Виконав:
студент 2 курсу, групи ІНФМ-21-1

Луцков В.А.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Луцкову Владиславу Аркадійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та реалізація методу генерації та серіалізації даних за стандартом HL7/FHIR

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 листопада 2022 р.3. Вихідні дані до роботи стандарт даних HL7, стандарт даних HL7/FHIR, структурні елементи JSON, дані інтернет мережі, мова програмування Java, фреймворк Spring, база даних MariaDB, середовище розробки IntelliJ IDEA, середовище розробки IntelliJ WebStorm, основні поняття серіалізації та десеріалізації, середовище розробки VisualStudioCode.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючий стандартів серіалізації та десеріалізації медичних даних.2. Аналіз структури, термінів та вимог стандарту HL7/FHIR.3. Моделювання алгоритму серіалізації та десеріалізації за стандартом HL7/FHIR.4. Програмна реалізація методу.5. Інтеграція алгоритму до існуючої системи, огляд результатів та формулювання висновків.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) постановка задачі, структура та основні поняття стандарту HL7/FHIR, інтеграція алгоритму до існуючої системи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.11.22	виконано
2	Аналіз завдання, підбір літератури	09.11.22	виконано
3	Аналіз літератури з досліджуваної проблеми	10.11.22-11.11.22	виконано
4	Аналіз технічних засобів	12.11.22	виконано
5	Розробка методу серіалізації та десеріалізації за стандартом HL7/FHIR	13.11.22-14.11.22	виконано
6	Програмна реалізація	14.11.22-15.11.22	виконано
7	Оформлення пояснювальної записки	16.11.22-17.11.22	виконано
8	Перевірка на плагіат	25.11.22	виконано
9	Рецензування	25.11.22-26.11.22	
10	Підготовка презентації та доповіді	26.11.22-27.11.22	
11	Занесення роботи в електронний архів	30.11.22	
12	Попередній захист кваліфікаційної роботи	05.12.22	

Дата видачі завдання 9 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Кобилін О.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 69 с., 7 табл., 41 рис., 53 джерела.

ДОСЛІДЖЕННЯ, МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА, АЛГОРИТМ, СЕРІАЛІЗАЦІЯ, ДЕСЕРІАЛІЗАЦІЯ, МОДЕЛЮВАННЯ, ПРОЕКТУВАННЯ, ІНТЕГРАЦІЯ, HL7, FHIR, БАЗА ДАНИХ, SPRING BOOT, ANGULAR, JAVA, MARIADB.

Об'єктом дослідження є стандарт обміну медичними даними FHIR (Fast Healthcare Interoperability Resources specification).

Метою дослідження є реалізації алгоритму генерації та серіалізації даних за стандартом обміну медичними даними HL7/FHIR. Цей підхід надасть можливість уніфікувати та структурувати дані медичних закладів і відділень різних типів.

Головним нововведенням стосовно реалізації алгоритму має стати імплементація з використанням новітніх технологій та бібліотек з високим рівнем оптимізації, великою швидкістю обробки масивів даних, можливістю кастомізації валідації вхідної та вихідної інформації, високим рівнем безпеки конфіденційних даних користувачів та закладів при інтеграції до вже існуючих систем.

RESEARCH, MEDICAL INFORMATION SYSTEM, ALGORYTHM, SERIALIZATION, DESERIALIZATION, MODELING, DESIGN, INTEGRATION, HL7, FHIR, DATABASE, SPRING BOOT, ANGULAR, JAVA, MARIADB.

The object of the research is the FHIR (Fast Healthcare Interoperability Resources specification) medical data exchange standard.

The aim of the research is to implement an algorithm for generating and serializing data according to the HL7/FHIR medical data exchange standard. This approach will provide an opportunity to unify and structure the data of medical institutions and departments of various types.

The main innovation regarding the implementation of the algorithm should be implementation using the latest technologies and libraries with a high level of optimization, high speed of processing data arrays, the possibility of customizing the validation of input and output information, a high level of security of confidential data of users and institutions when integrating into existing systems.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Проблеми обміну медичною інформацією в США.....	9
1.2 Проблеми обміну медичною інформацією в Європі.....	11
1.3 Існуючі стандарти обміну медичною інформацією	12
1.3.1 Digital Imaging and Communications in Medicine.....	12
1.3.2 OpenEHR	13
1.3.3 HL7.....	13
1.4 Проблеми обміну медичною інформацією в Україні.....	14
1.5 Постановка задачі дослідження.....	15
2 Моделювання алгоритму генерації та серіалізації даних за стандартом HL7/FHIR	17
2.1 Концепція алгоритму.....	17
2.2 Основні поняття	17
2.3 Типи даних.....	17
2.3.1 Примітивні типи даних.....	18
2.3.2 Комплексні типи даних	20
2.3.3 Мета-дані.....	21
2.3.4 Тип даних «Resource».....	21
2.3.5 Тип даних «DomainResource»	22
2.3.6 Ресурсні типи даних. Підкласи типу даних «DomainResource».....	23
2.3.7 Батьківський тип даних «Element».....	24
2.4 Контекстна модель алгоритму.....	25
2.4.1 Патерн «Request».....	25
2.4.2 Патерн «Event»	27
2.4.3 Патерн «Definition»	29

	6
2.5 Валідація даних	31
3 Реалізація методу генерації та серіалізації даних за стандартом HL77/FHIR. Інтеграція реалізованого алгоритму до існуючої системи інформаційного супроводу амбулаторних пацієнтів в медичних установах	32
3.1 Обґрунтування вибору середовища програмної реалізації	32
3.1.1 Angular.js для головних модулів алгоритму.....	32
3.1.2 Java для блоку інтеграції	33
3.1.3 Thymeleaf як прошарок між алгоритмом та інтеграційним блоком	34
3.1.4 Docker	35
3.1.5 Середовище програмної реалізації.....	36
3.2 Програмна реалізація back-end	44
3.3 База даних	46
3.4 Програмна реалізація front-end.....	48
3.5 Інструкція користувача	49
Висновки	63
Перелік джерел посилання	64

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

БД – база даних

ІС – інформаційна система

МІС – медична інформаційна система

FHIR – Fast Healthcare Interoperability Resources specification (ресурси швидкої взаємодії в сфері охорони здоров'я)

ВСТУП

Забезпечення ефективного обміну цифровою медичною інформацією між лікувально-профілактичними установами в наш час є однією із пріоритетних задач в Україні. Введення нових інформаційних технологій дозволяє не тільки суттєво збільшити якість лікувально-профілактичного процесу, а й надає можливість значно зменшити витрати ресурсів, наприклад, за рахунок зменшення дублювання діагностичних досліджень та необхідного персоналу. Однак, на даний момент повсюдне використання цих можливостей перешкоджає ряд проблем, насамперед організаційно-правового та технологічного характеру. Однією з проблем, що носять комплексний характер, є надійна технологічна та правова захищеність конфіденційної медичної інформації, що вимагає використання певних стандартів.

Мета цієї кваліфікаційної роботи полягає в дослідженні та реалізації алгоритму генерації та серіалізації даних за стандартом обміну медичними даними HL7/FHIR. Цей підхід надасть можливість уніфікувати та структурувати дані медичних закладів і відділень різних типів.

В рамках кваліфікаційної роботи об'єктом дослідження є стандарт обміну медичними даними FHIR (Fast Healthcare Interoperability Resources specification), предметом дослідження – алгоритм генерації та серіалізації даних за цим стандартом та інтегрованих до існуючої медичної інформаційної системи.

Актуальність дослідження полягає у імплементації з використанням новітніх технологій та бібліотек з високим рівнем оптимізації, великою швидкістю обробки масивів даних, можливістю кастомізації валідації вхідної та вихідної інформації, високим рівнем безпеки конфіденційних даних користувачів та закладів при інтеграції до вже існуючих систем. Оскільки зараз такими характеристиками володіють лічені алгоритми, то поставлена задача є актуальною.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблеми обміну медичною інформацією в США

Електронні медичні записи (Electronic Health Records, EHRs) вже давно широко використовуються в усіх розвинених країнах. Наприклад, в США ще в 1996 році був прийнятий закон про мобільність та відповідальність медичного страхування (Health Insurance Portability and Accountability Act), згідно якому автор медичних записів (установа охорони здоров'я, приватний лікар і т.д.) несе повну відповідальність за захист EHRs-файлів. При цьому, базою для створення безпечного середовища, необхідного для обміну медичними даними, є електронний цифровий підпис. Певна кількість досліджень показує, що вони повністю відповідають трьом основним вимогам кібербезпеки [1]:

- конфіденційність;
- цілісність;
- безвідмовність EHRs.

Не дивлячись на наявність технічної та законодавчої бази, спеціальні дослідження, проведені у 2009 році показали, що лише близько половини медичних закладів США використовують автоматизовані системи зберігання медичної інформації, та не більше чверті лікарів первинної ланки використовують електронні медичні записи [2]. Через це в 2009 році був прийнятий The Health Information Technology for Economic and Clinical Health (HITECH) Act, що має мінімізувати правові, фінансові та технічні бар'єри в реалізації EHRs та встановлює цифровий підпис як основну гарантію цілісності даних.

Після прийняття HITECH кількість систем обміну інформації про здоров'я збільшилось, та до 2011 року кількість лікувальних закладів, що використовують системи EHRs, збільшилась втричі. Натомість, не всі ці

системи охоплювали достатньо велику частину медичних працівників та пацієнтів, щоб збільшити якість медичної допомоги. Інформація з невеликих комерційних установ частіше за все була неоднозначною та неоднорідною за якістю, і не відображала повноту наявних клінічних даних. Та й самі EHRs не були доступні для всіх медичних працівників, що доглядали пацієнтів [3]. Усі ці проблеми призвели до того, що подальше розповсюдження електронних систем в охорону здоров'я сповільнилось. В результаті в одному з опитувань лікарів, що зробив Центр стратегічних досліджень корпорації RAND, незадоволеність від використання EHRs було на другому місці серед причин незадоволеності від роботи, одразу після неможливості надати пацієнтові високоякісну допомогу, що навіть випереджає фінансові проблеми [4].

Тільки після виявлення таких проблем, наприкінці 2015 року 18 найбільших громадських медичних організацій (NAfTE, CWHNA та інші) запровадили проект по розвитку та адаптації існуючих стандартів медичних даних та створенню операційної системи, що буде пов'язувати усі електронні медичні дані воедино [5]. У свою чергу, урядом США був прийнятий Федеральний стратегічний план по інформаційним технологіям в медицині (Federal Health IT Strategic Plan 2015-2020), а Міністерство охорони здоров'я та соціальних служб у 2016 році оголосило те, що компанії, які не підтримують 90 відсотків EHRs, що використовують американські лікарні, п'ять найбільших приватних систем охорони здоров'я країни та більше десяти провідних професійних асоціацій та груп винайшли три основні принципи, що надають можливість покращити якість інформації про стан здоров'я населення для користувачів та постачальників медичних послуг.

По-перше, доступність для користувачів. Можливість громадянина легко та безпечно отримати доступ до своєї особистої електронної медичної інформації, спрямувати її у будь-яке місце чи дізнатися, як вона може бути ефективно та безпечно використана для особистого здоров'я чи здоров'я спільноти. Більшість великих розробників медичних програм взяли на себе

зобов'язання застосовувати стандартні інтерфейси прикладного програмування та один спільний стандарт для обміну даними – HL7.

По-друге, відсутність блокування даних. При переході пацієнта з однієї медичної організації в іншу, усі його дані повинні бути надані кожного разу, коли в них виникає потреба, відповідно до чинного законодавства.

По-третє, стандартизація. Реалізувати визнані на федеральному рівні національні стандарти сумісності, рекомендації та практики в області електронної медичної інформації, а також використовувати передовий досвід, зокрема пов'язаний з особистим життям та безпекою.

Зараз майже всі громадські лікарні та три чверті приватних лікарів США використовують в роботі сертифіковані електронні медичні ресурси. Це не тільки дало можливість доступу до клінічних даних пацієнтів незалежно від місця знаходження та статусу установи, але й помітно збільшило ефективність медичної допомоги та кінцевий результат лікування. Невирішеною залишається лише проблема ступеня конфіденційності медичної інформації. Наприклад, якщо виникають небезпечні інфекційні захворювання, урядовими службами США та державними установами охорони здоров'я використовуються дані пацієнтів без їх згоди, що суперечить закону про конфіденційність особистих даних. У зв'язку з цим пропонується висунути закон, який дозволяє працювати з великими масивами даних без індивідуальної згоди [6].

1.2 Проблеми обміну медичною інформацією в Європі

У європейській спільноті ситуація з електронними медичними записами є більш різномірною. Медичні заклади Євросоюзу також стикаються з труднощами в організації медичної допомоги через різкі зміни в демографії, та соціально-економічній структурі населення, що відбулися в останні роки. При цьому, хоча ступінь комп'ютеризації більшості великих

європейських лікарень досить висока, в ЄС відсутній узгоджений погляд на актуальні задачі та їх можливі рішення, що не дозволяє розробити єдину концепцію розвитку електронної медицини. Однак, широке розповсюдження стандарту HL7 надає можливість сподіватись на подальшу модернізацію суспільної охорони здоров'я, яка на даний момент упирається в технології обробки великих масивів даних [7].

1.3 Існуючі стандарти обміну медичною інформацією

Серед технологічних аспектів, що мають першорядне значення в реалізації ефективного обміну медичною інформацією, визначальну роль має питання стандартизації. За останні роки по всьому світу була розроблена значна кількість стандартів електронної медицини, зокрема DICOM, openEHR та HL7.

1.3.1 Digital Imaging and Communications in Medicine

Стандарт DICOM (Digital Imaging and Communications in Medicine) призначений для зберігання та передачі графічної і текстової медичної інформації. Він дозволяє організувати цифровий зв'язок між різними діагностичним та терапевтичним медичним обладнанням. Для транспорту в DICOM використовується TCP-протокол, тобто стандартний протокол мережевого зв'язку між комп'ютерними системами. Для підвищення ефективності обробки даних використано комплексний підхід до управління усією діагностичною інформацією, починаючи з вводу зображення та закінчуючи архівацією. Використання DICOM дозволяє знизити собівартість медичних послуг за рахунок таких факторів [8]:

- скорочення часу обслуговування діагностичного обладнання;

- відмова від плівок;
- значне скорочення втрат зображень та результатів досліджень.

1.3.2 OpenEHR

OpenEHR – це стандарт з відкритим вихідним кодом для управління, зберігання та обміну електронними історіями хвороб. Він активно розвивається силами інтернаціональної спільноти. Головний офіс фонду openEHR знаходиться в Лондоні. Найбільший вклад у розвиток стандарту вносить компанія OceanInformatics [9].

У числі базових організаційних концепцій openEHR необхідно відзначити наступні:

- усі дані про здоров'я людини зберігаються за весь період її життя;
- формат даних не повинен залежати від організації, що розміщує цю інформацію;
- розміщена інформація орієнтована на кінцевого користувача.

Таким чином, openEHR вміщує в себе достатньо виразну модель даних, що відповідає сучасним вимогам надання медичної інформації. Крім того, в openEHR передбачений ряд загальноприйнятих способів імпорту та експорту даних (в тому числі в HL7 CDA), генерації програмних та друкованих форм, мовна підтримка та інше.

1.3.3 HL7

Серед протоколів інформаційної взаємодії в наш час все більшу популярність набирає стандарт HL7, що був розроблений некомерційною організацією Health Level Seven, Inc. Однією з найбільш затребуваних редакцій даного стандарту є ISO/HL7 27932:2009 Data Exchange Standards –

HL7 Clinical Document Architecture, Release 2 (скорочено HL7 CDA R2). Для забезпечення можливості машинної обробки персональних та медичних даних використовуються галузеві класифікатори: бібліотека медичних термінів SNOMED CT, класифікатор LOINC, ідентифікатори (документів, пацієнтів, медичного персоналу, організацій тощо). При цьому інформація надається як у вигляді структурованих записів з використанням мови розмітки XML, так і у вигляді будь-якого закодованого об'єкту, включаючи відсканований образ документу, текстовий чи табличний документ [10].

1.4 Проблеми обміну медичною інформацією в Україні

Повертаючись до ситуації в нашій країні, можна відмітити, що у теперішній час рівень розвитку інформаційних технологій надає можливість вирішувати будь-які задачі електронної медицини. Слабким місцем є відсутність актуальної законодавчої бази стосовно електронних медичних документів. Однак, у найближчий час ці питання повинні бути вирішені. Міністерство охорони здоров'я активно займається електронним документообігом, в першу чергу, відмовившись від обов'язкового дублювання документів на паперових носіях. Необхідно також розробити правила використання електронного цифрового підпису в системі охорони здоров'я та вимоги до інформаційної безпеки та захисту персональних даних при використанні ЕЦП. Міністром охорони здоров'я було оголошено, що до кінця 2018 року більше 95% державних медичних закладів будуть підключені до єдиної інформаційної системи електронної медичної карти. У цьому випадку, однією із актуальних задач стане обробка дійсних механізмів обміну цифровою медичною інформацією при реалізації ЕЦП. Одним із відповідних розділів, безумовно, є системи передачі, зберігання та обробки даних HL7. До того ж це єдиний медичний формат, що не залежить від формату даних.

1.5 Постановка задачі дослідження

У наш час в лікарнях існують механізми ведення пацієнтів, що спрямовані на покращення роботи лікарів та інформаційної забезпеченості пацієнтів, але лише лічені з них використовують можливості, які може надати якісно спроектована комп'ютерна система. Задачі реєстрації медичних закладів в спільній системі, інформаційний супровід як лікарів та адміністрації, так і пацієнтів, ведення хворого за зручними та інтуїтивно зрозумілими алгоритмами є невирішеними [11–30]. Для покращення стану обслуговування пацієнтів в медичних закладах потрібна така МІС, яка буде максимально оптимізована, швидка та зручна з усіх сторін. Тому створення Web-застосунку з високою швидкістю обробки запитів та інтуїтивно зрозумілими алгоритмами й інтерфейсом є актуальним.

Об'єктом дослідження є стандарт обміну медичними даними FHIR (Fast Healthcare Interoperability Resources specification).

Метою дослідження є реалізації алгоритму генерації та серіалізації даних за стандартом обміну медичними даними HL7/FHIR. Цей підхід надасть можливість уніфікувати та структурувати дані медичних закладів і відділень різних типів.

Предметом дослідження є алгоритм генерації та серіалізації даних за цим стандартом та інтегрованих до існуючої медичної інформаційної системи.

Перш за все треба визначити завдання розробити МІС, що надасть можливість адміністрування медичних закладів та ведення пацієнтів.

Для вирішення такої задачі потрібно вирішити такі суб-задачі:

- моделювання багатомодульного застосунку;
- моделювання бази даних;
- авторизація користувача;
- інтерфейс імпорту даних медичного закладу для подальшої реєстрації;

- інтерфейс реєстрації медичного закладу;
- інтерфейс адміністрування медичним закладом;
- інтерфейс адміністрування відділами медичного закладу;
- інтерфейс адміністрування персоналом;
- реалізувати можливість реєстрації нових працівників за певними правилами безпеки;
- інтерфейс ведення пацієнтів.

Зазначені алгоритми та механізми потребують вхідні дані у вигляді даних медичного закладу, особистих даних його працівників та пацієнтів.

У результаті буде розроблено медичну інформаційну систему у вигляді Web-застосунку для адміністрації медичним закладом із персоналом, з можливістю ведення пацієнтів за стандартами ICD та веденням лікарні за допомогою формулярів стандарту HL7/FHIR.

2 МОДЕЛЮВАННЯ АЛГОРИТМУ ГЕНЕРАЦІЇ ТА СЕРІАЛІЗАЦІЇ ДАНИХ ЗА СТАНДАРТОМ HL7/FHIR

2.1 Концепція алгоритму

Основа концепції будь-якого алгоритму генерації та серіалізації даних за певним стандартом полягає в її поетапному проектуванні та моделюванні кожного його компоненту [31–41]. Концепція базується на функціональних вимогах та документації стандарту.

2.2 Основні поняття

Глосарій предметної області створюється на основі опису стандарту та його документації (табл. 2.1).

Таблиця 2.1 – Основний глосарій HL7/FHIR

Термін	Визначення
FHIR Questionnaire	Структурний елемент FHIR
FHIR Questionnaire Response	Структурний елемент відповідей (може включати самі відповіді) FHIR

2.3 Типи даних

За стандартом HL7/FHIR для усіх типів даних батьківським елементом є «Element».

UML-діаграма усіх існуючих стандартних типів даних HL7/FHIR зазначено на рисунку 2.1.

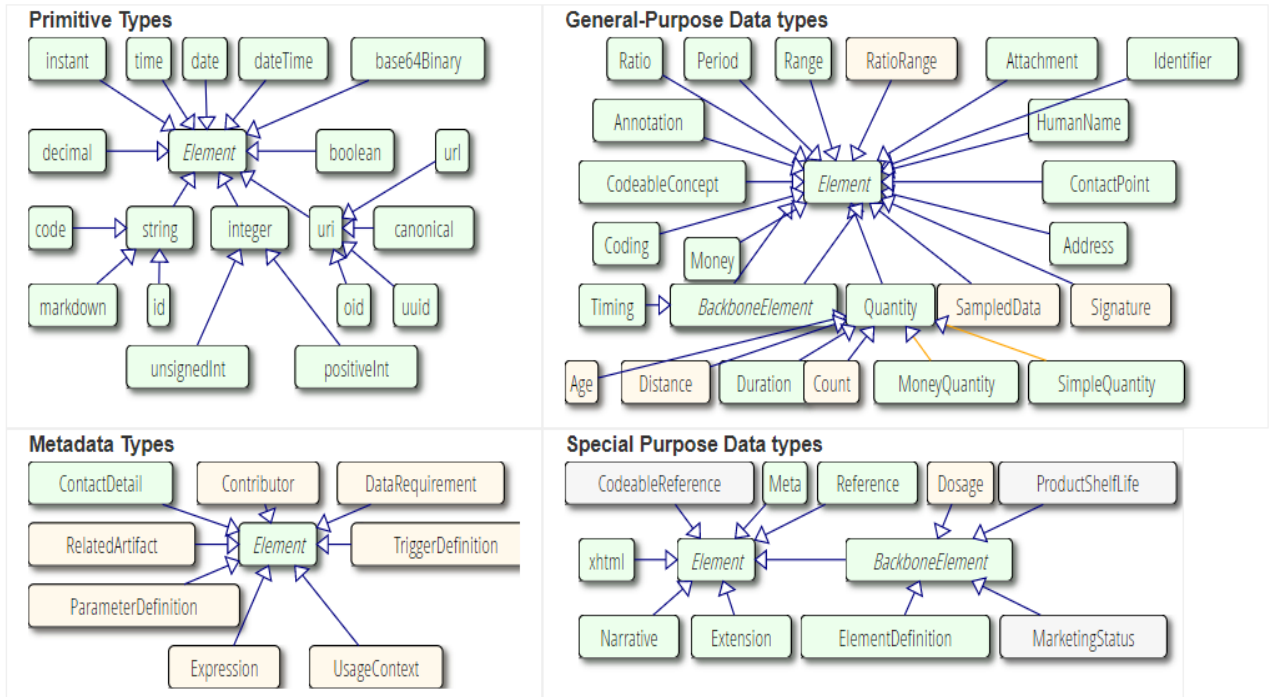


Рисунок 2.1 – UML-діаграма усіх типів даних HL7/FHIR

2.3.1 Примітивні типи даних

Перелік усіх примітивних типів даних зображений на рисунку 2.2.

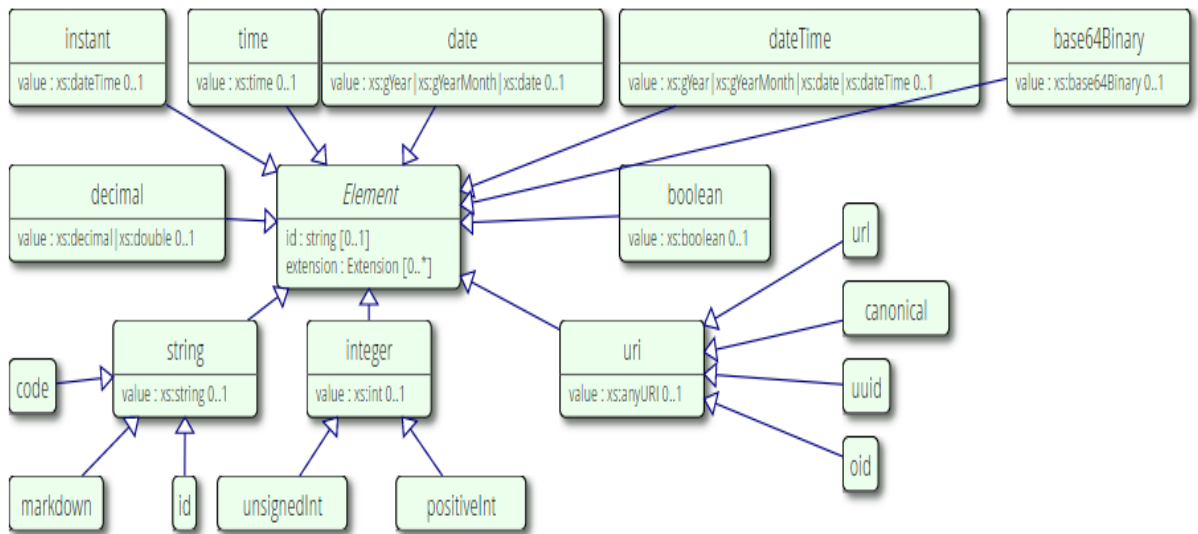


Рисунок 2.2 – UML-діаграма примітивних типів даних HL7/FHIR

Глосарій примітивних типів даних зазначено у таблиці 2.2.

Таблиця 2.2 – Глосарій примітивних типів даних HL7/FHIR

Термін	Визначення
Boolean	BIT(1) тип даних (true/false)
Integer	Структурний елемент відповідей FHIR
String	Строковий тип даних (об'єм до 1 МВ)
Decimal	Раціональне число, 32-bit та вище
URI	Uniform Resource Identifier Reference (за стандартом RFC 3986)
URL	Uniform Resource Locator (за стандартом RFC 1738)
Canonical	Ресурс із URL параметром
Base64Binary	Потоковий тип даних, шифрування base64 (за стандартом RFC 4648)
Instant	Час у форматі «YYYY-MM-DDThh:mm:ss.sss+zz:zz». Використовується лише для системного часу
Date	Дата у форматі «YYYY-MM-DD»
DateTime	Дата та час у форматі «YYYY-MM-DDThh:mm:ss.sss+zz:zz»
Time	Час
Code	Код із словника
OID	OID репрезентація URI (за стандартом EAC 3001)
ID	Унікальний ідентифікатор за системою ASCII
Markdown	FHIR строковий (string) формат, що включає в себе дані форматування
UnsignedInt	Ціле число більше 0 (32-bit)
PositiveInt	Ціле число більше 1 (32-bit)
UUID	UUID чи GUID код у форматі URI RFC 4122)

2.3.2 Комплексні типи даних

За стандартом HL7/FHIR комплексні типи даних складаються із примітивних типів даних. На рисунку 2.3 зображено UML-діаграму усіх комплексних типів даних.

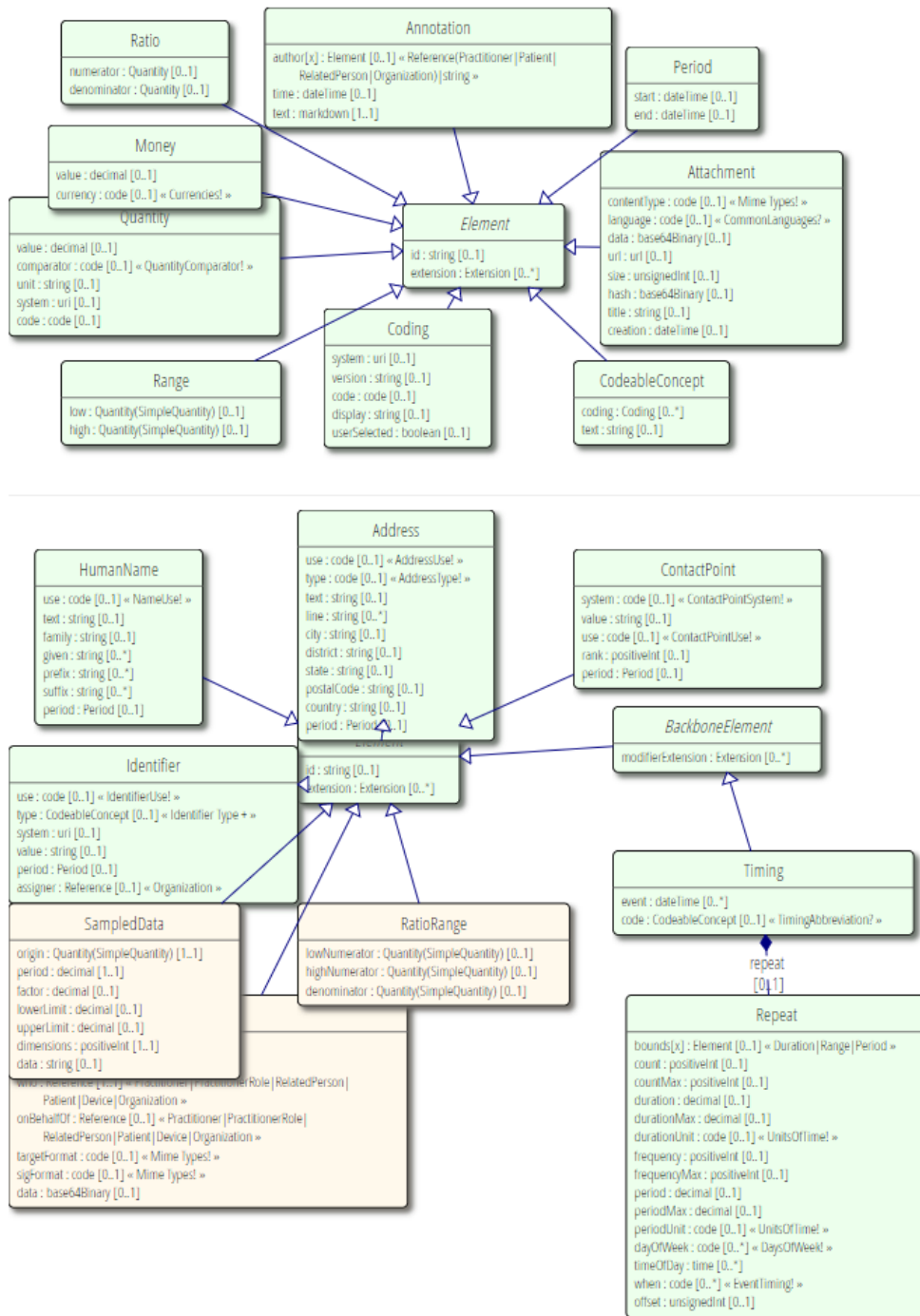


Рисунок 2.3 – UML-діаграма комплексних типів даних HL7/FHIR

2.3.3 Мета-дані

За стандартом HL7/FHIR існують мета-дані які також складаються із примітивних типів даних. На рисунку 2.4 зображено UML-діаграму типів мета-даних:

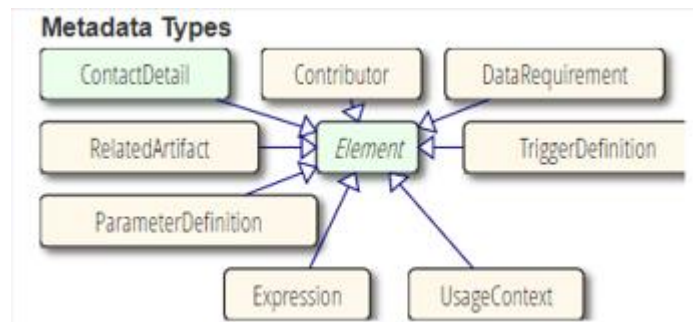


Рисунок 2.4 – UML-діаграма мета-даних HL7/FHIR

2.3.4 Тип даних «Resource»

Ця специфікація визначає низку різних типів ресурсів, які можна використовувати для обміну та/або зберігання даних для вирішення широкого кола проблем, пов'язаних із охороною здоров'я, як клінічних, так і адміністративних. Крім того, ця специфікація визначає кілька різних способів обміну ресурсами. За стандартом HL7/FHIR тип даних «Resource» складається із таких блоків даних:

- ID (String) – примітив, унікальний ідентифікатор;
- Meta – мета-дані;
- ImplicitRules (URI) – набір правил, за якими цей контент був створений;
- Language (Code) – язык ресурсу.

На рисунку 2.5 зображено схематичний опис структури типу даних «Resource».

Structure

Name	Flags	Card.	Type	Description & Constraints
Resource	«A» N		n/a	Base Resource
id	Σ	0..1	string	Logical id of this artifact
meta	Σ	0..1	Meta	Metadata about the resource
implicitRules	?! Σ	0..1	uri	A set of rules under which this content was created
language		0..1	code	Language of the resource content

Рисунок 2.5 – Схематичний опис структури «Resource» HL7/FHIR

На рисунку 2.6 схематично зображено опис структури атрибуту «Meta» типу даних «Resource».

Structure

Name	Flags	Card.	Type
Meta	Σ N		Element
versionId	Σ	0..1	id
lastUpdated	Σ	0..1	instant
source	Σ	0..1	uri
profile	Σ	0..*	canonical(StructureDefinition)
security	Σ	0..*	Coding
tag	Σ	0..*	Coding

Рисунок 2.6 – Схематичний опис атрибуту «Meta» типу даних «Resource» HL7/FHIR

2.3.5 Тип даних «DomainResource»

DomainResource – це абстрактний тип ресурсу. Він слугує базовим типом для створення ресурсів різних типів. На рисунку 2.7 зображено схематичний опис структури типу даних «DomainResource».

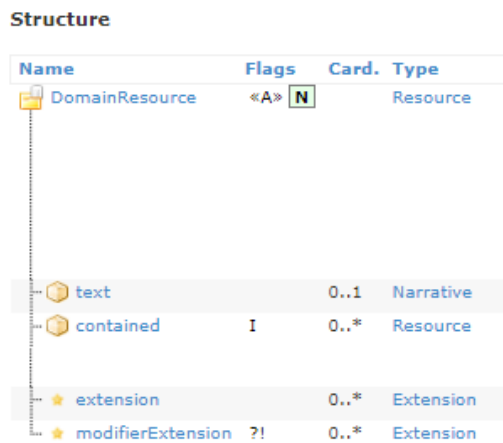


Рисунок 2.7 – Схематичний опис структури «DomainResource» HL7/FHIR

2.3.6 Ресурсні типи даних. Підкласи типу даних «DomainResource»

За стандартом HL7/FHIR тип даних «DomainResource» має значення абстрактного ресурсного типу, тобто батьківського класу, що відображає базову структуру для усіх своїх підтипів.

Підтипи класу «DomainResource» строго специфіковані та уніфіковані і поділяються на 5 основних груп:

- Foundation;
- Base;
- Clinical;
- Financial;
- Specialized.

Кожна із цих груп у свою чергу поділяється на підгрупи:

- Foundation: Conformance, Terminology, Security, Documents та Other;
- Base: Individuals, Entities1, Entities2, Workflow, Management;
- Clinical: Summary, Diagnostics, Medications, Care Provision, Request and Response;
- Financial: Support, Billing, Payment, General;
- Specialized: Public Health and Research, Definitional Artifacts, Evidence.

Based Medicine, Quality Reporting and Testing, Medication Definition.

На рисунку 2.8 зображено схематичний опис структури ресурсних типів даних, що є підкласами типу даних «DomainResource».

Foundation	Conformance <ul style="list-style-type: none"> CapabilityStatement N StructureDefinition N ImplementationGuide 1 SearchParameter 3 MessageDefinition 1 OperationDefinition N CompartmentDefinition 1 StructureMap 2 GraphDefinition 1 ExampleScenario 0 	Terminology <ul style="list-style-type: none"> CodeSystem N ValueSet N ConceptMap 3 NamingSystem 2 TerminologyCapabilities 0 	Security <ul style="list-style-type: none"> Provenance 3 AuditEvent 3 Consent 2 	Documents <ul style="list-style-type: none"> Composition 2 DocumentManifest 2 DocumentReference 3 CatalogEntry 0 	Other <ul style="list-style-type: none"> Basic 1 Binary N Bundle N Linkage 0 MessageHeader 4 OperationOutcome N Parameters N Subscription 3 SubscriptionStatus 0 SubscriptionTopic 0 	
	Base	Individuals <ul style="list-style-type: none"> Patient N Practitioner 3 PractitionerRole 2 RelatedPerson 2 Person 2 Group 1 	Entities #1 <ul style="list-style-type: none"> Organization 3 OrganizationAffiliation 0 HealthcareService 2 Endpoint 2 Location 3 	Entities #2 <ul style="list-style-type: none"> Substance 2 BiologicallyDerivedProduct 0 Device 2 DeviceMetric 1 NutritionProduct 0 	Workflow <ul style="list-style-type: none"> Task 2 Appointment 3 AppointmentResponse 3 Schedule 3 Slot 3 VerificationResult 0 	Management <ul style="list-style-type: none"> Encounter 2 EpisodeOfCare 2 Flag 1 List 1 Library 3
		Clinical	Summary <ul style="list-style-type: none"> AllergyIntolerance 3 AdverseEvent 0 Condition (Problem) 3 Procedure 3 FamilyMemberHistory 2 ClinicalImpression 0 DetectedIssue 1 	Diagnostics <ul style="list-style-type: none"> Observation N Media 1 DiagnosticReport 3 Specimen 2 BodyStructure 1 ImagingStudy 3 QuestionnaireResponse 3 MolecularSequence 1 	Medications <ul style="list-style-type: none"> MedicationRequest 3 MedicationAdministration 2 MedicationDispense 2 MedicationStatement 3 Medication 3 MedicationKnowledge 0 Immunization 3 ImmunizationEvaluation 0 ImmunizationRecommendation 1 	Care Provision <ul style="list-style-type: none"> CarePlan 2 CareTeam 2 Goal 2 ServiceRequest 2 NutritionOrder 2 VisionPrescription 2 RiskAssessment 1 RequestGroup 2
	Financial		Support <ul style="list-style-type: none"> Coverage 2 CoverageEligibilityRequest 2 CoverageEligibilityResponse 2 EnrollmentRequest 0 EnrollmentResponse 0 	Billing <ul style="list-style-type: none"> Claim 2 ClaimResponse 2 Invoice 0 	Payment <ul style="list-style-type: none"> PaymentNotice 2 PaymentReconciliation 2 	General <ul style="list-style-type: none"> Account 2 ChargeItem 0 ChargeItemDefinition 0 Contract 1 ExplanationOfBenefit 2 InsurancePlan 0
		Specialized	Public Health & Research <ul style="list-style-type: none"> ResearchStudy 1 ResearchSubject 0 	Definitional Artifacts <ul style="list-style-type: none"> ActivityDefinition 3 DeviceDefinition 0 EventDefinition 0 ObservationDefinition 0 PlanDefinition 3 Questionnaire 3 SpecimenDefinition 0 	Evidence-Based Medicine <ul style="list-style-type: none"> Citation 0 Evidence 1 EvidenceReport 0 EvidenceVariable 1 	Quality Reporting & Testing <ul style="list-style-type: none"> Measure 3 MeasureReport 3 TestScript 2 TestReport 0

Рисунок 2.8 – Схематичний опис структури підкласів типу даних «DomainResource» HL7/FHIR

2.3.7 Батьківський тип даних «Element»

За стандартом HL7/FHIR тип даних «Element» є батьківським для усіх типів даних, тобто «Element» відображає базову структуру (рис. 2.9).

Structure

Name	Flags	Card.	Type
Element	I		n/a
id		0..1	string
extension		0..*	Extension

Рисунок 2.9 – Схематичний опис структури типу даних «Element» HL7/FHIR

2.4 Контекстна модель алгоритму

При виконанні генерації, серіалізації та десеріалізації даних за стандартом HL7/FHIR здійснюється взаємодія трьох основних компонентів, а саме: «Request», «Event» та «Definition». Кожен із цих компонентів є патерном ресурсів, що мають стандартний набір попарних взаємодій.

На рисунку 2.10 зображена схема взаємодії трьох основних компонентів:

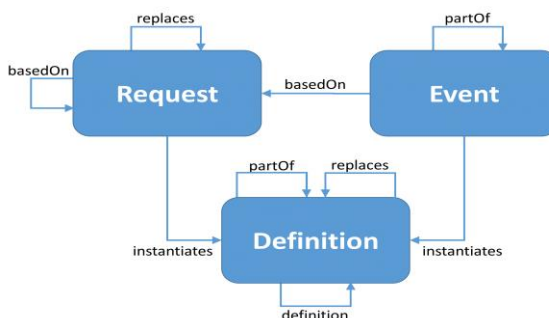


Рисунок 2.10 – Взаємодія компонентів HL7/FHIR

2.4.1 Патерн «Request»

Request – патерн, якому слід слідувати за ресурсами, які представляють конкретну пропозицію, план і/або замовлення на якусь дію чи послугу. Це не ресурс. Він не є частиною схеми FHIR і не може відобразитися

безпосередньо в екземплярах FHIR. Це логічна модель, яка визначає шаблон, якого дотримуються інші ресурси.

Поняття «Request» охоплює всі типи наказів (оригінальні накази, представлення заповнювачів наказів, зворотні накази тощо), а також пропозиції/рекомендації щодо виконання дій, плани, графіки тощо. Будь-який опис діяльності що є «бажаним», якщо опис є конкретним щодо предмета діяльності та приблизний час виконання діяльності вважатиметься «Request».

Ця модель являє собою патерн. Він надає стандартний список елементів даних із потужностями, типами даних, визначеннями, обґрунтуванням і примітками щодо використання, яких ідеально дотримуватимуться ресурси, що належать до категорії робочого процесу «Request». Однак дотримання цієї схеми не є обов'язковим. Не всі сфери охорони здоров'я однакові. Концепції, які можуть бути загальноприйнятими (і, таким чином, включені в цей стандартний шаблон), можуть все ще не бути доречними скрізь або можуть бути настільки рідкісними, що їх доречніше включати як розширення, ніж як основні властивості ресурсу. Робочим групам рекомендується налаштувати описи, примітки щодо використання та обґрунтування відповідно до їхнього ресурсу (наприклад, використовувати термін «діагностичний тест» або «рецепт», а не «запит»). Крім того, примітки щодо дизайну в колонці коментарів, позначені «[]» (квадратними дужками), визначають області, де очікується та заохочується зміна домену.

На наступній діаграмі (рис. 2.11) показано типову діаграму статусів для ресурсів, що відповідають шаблону «Request». Але не всі ресурси підтримуватимуть усі стани, деякі ресурси можуть вибирати різні назви для певних станів, а деякі ресурси можуть вводити підстани до перелічених станів. Крім того, можуть підтримуватися додаткові переходи, включно з термінальними вузлами (наприклад, від «завершеного» назад до «активного»). Тим не менш, більшість ресурсів повинні досить добре узгоджуватися з цією схемою.

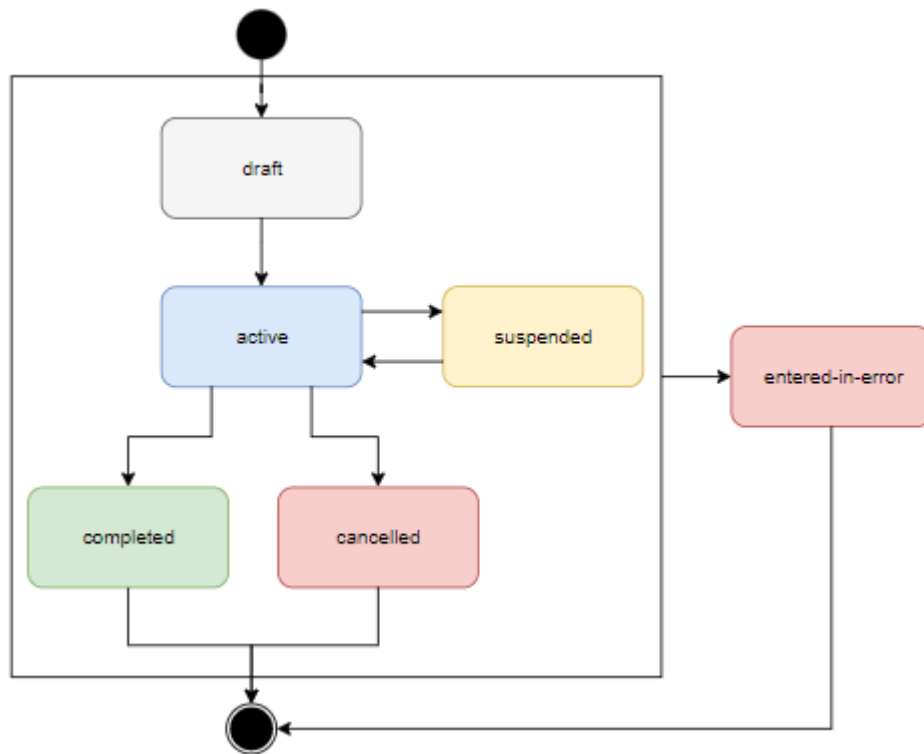


Рисунок 2.11 – Стани «Request» HL7/FHIR

2.4.2 Патерн «Event»

Event – патерн, якому повинні слідувати ресурси, які представляють виконання певної діяльності, можливо, відповідно до запиту або визначення служби. Він не є частиною схеми FHIR і не може відобразитися безпосередньо в екземплярах FHIR. Це логічна модель, яка визначає шаблон, якого дотримуються інші ресурси.

Event – це будь-який опис діяльності, яка вже відбулася або триває. Він включає ресурси, які в основному описують «результат» діяльності або те, що було знайдено (наприклад, стан або спостереження). Приклади включають зустрічі, процедури, заповнені анкети, представлення переходів станів тощо. Це не включає ресурси, які описують об'єкти чи ролі (наприклад, пацієнт, пристрій, місцезнаходження).

Ця модель являє собою патерн. Він надає стандартний список елементів даних із кількістю елементів, типами даних, визначеннями, обґрунтуванням і примітками щодо використання, яких ідеально дотримуватимуться ресурси, що належать до категорії робочого циклу «подія». Однак дотримання цієї схеми не є обов'язковим. Не всі сфери охорони здоров'я однакові. Концепції, які можуть бути загальноприйнятими (і, таким чином, включені в цей стандартний шаблон), можуть все ще не бути доречними скрізь або можуть бути настільки рідкісними, що їх доречніше включати як розширення, ніж як основні властивості ресурсу. Робочі групи заохочуються коригувати описи, примітки щодо використання та обґрунтування відповідно до їхнього ресурсу (наприклад, використовувати термін «процедура» або «спостереження», а не «подія»). Крім того, примітки щодо дизайну в колонці коментарів, позначені «[]» (квадратними дужками), визначають області, де очікується та заохочується зміна домену.

На наступній діаграмі (рис. 2.12) показано типову діаграму статусів для ресурсів, що відповідають шаблону «Event». Але не всі ресурси підтримуватимуть усі стани, деякі ресурси можуть вибирати різні назви для певних станів, а деякі ресурси можуть вводити підстани до перелічених станів. Крім того, можуть підтримуватися додаткові переходи, включно з термінальними вузлами (наприклад, від «завершеного» назад до «активного»). Тим не менш, більшість ресурсів повинні досить добре узгоджуватися з цією схемою.

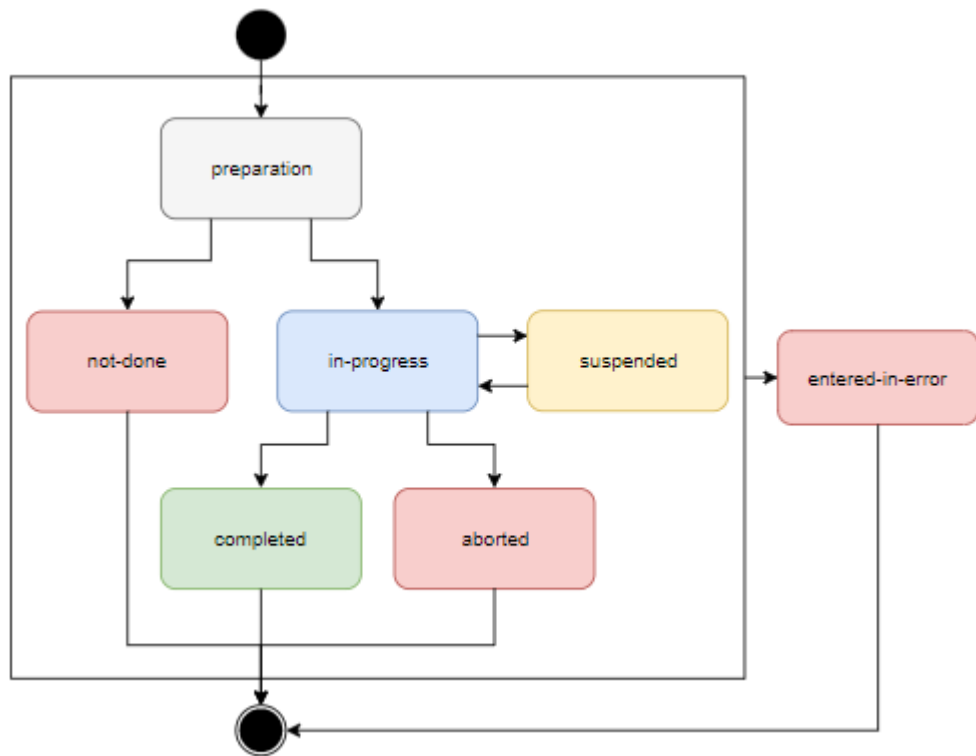


Рисунок 2.12 – Стани «Event» HL7/FHIR

2.4.3 Патерн «Definition»

Event – патерн, якому слід слідувати за ресурсами, які представляють конкретну пропозицію, план і/або замовлення на якусь дію чи послугу. Він не є частиною схеми FHIR і не може відобразитися безпосередньо в екземплярах FHIR. Це логічна модель, яка визначає шаблон, якого дотримуються інші ресурси.

Event – це будь-який опис діяльності, яка вже відбулася або триває. Він включає ресурси, які в основному описують «результат» діяльності або те, що було знайдено (наприклад, стан або спостереження). Приклади включають зустрічі, процедури, заповнені анкети, представлення переходів станів тощо. Це не включає ресурси, які описують об'єкти чи ролі (наприклад, пацієнт, пристрій, місцезнаходження).

Ця модель являє собою патерн. Він надає стандартний список елементів даних із потужностями, типами даних, визначеннями,

обґрунтуванням і примітками щодо використання, яких ідеально дотримуватимуться ресурси, що належать до категорії робочого циклу «подія». Однак дотримання цієї схеми не є обов'язковим. Не всі сфери охорони здоров'я однакові. Концепції, які можуть бути загальноприйнятими (і, таким чином, включені в цей стандартний шаблон), можуть все ще не бути доречними скрізь або можуть бути досить рідкісними, тому їх доречніше включати як розширення, ніж як основні властивості ресурсу. Робочі групи заохочуються коригувати описи, примітки щодо використання та обґрунтування відповідно до їхнього ресурсу (наприклад, використовувати термін «процедура» або «спостереження», а не «подія»). Крім того, примітки щодо дизайну в колонці коментарів, позначені «[]» (квадратними дужками), визначають області, де очікується та заохочується зміна домену.

На наступній діаграмі (рис. 2.13) показано типову діаграму статусів для ресурсів, що відповідають шаблону «Event». Але не всі ресурси підтримуватимуть усі стани, деякі ресурси можуть вибирати різні назви для певних станів, а деякі ресурси можуть вводити підстани до перелічених станів. Крім того, можуть підтримуватися додаткові переходи, включно з термінальними вузлами (наприклад, від «вилучено» назад до «активного»). Тим не менш, більшість ресурсів повинні досить добре узгоджуватися з цією схемою.

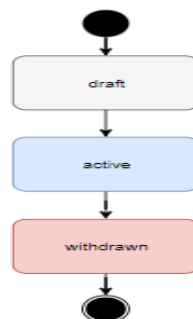


Рисунок 2.13 – Стани «Definition» HL7/FHIR

2.5 Валідація даних

Сервери та клієнти можуть бути налаштовані на перевірку вмісту під час його отримання (наприклад, деякі загальнодоступні служби тестування перевіряють ресурси під час створення/оновлення). Це можна зробити як під час розробки, так і під час виробництва додатків у процесах охорони здоров'я. Хоча використання під час циклу розробки настійно рекомендується, використання під час виробництва може бути не завжди гарною ідеєю:

- валідація – особливо повна перевірка, включаючи термінологію може бути досить вимогливою до обчислень. Це може призвести до неприйнятних часових затримок у виробничій системі;

- під час виробництва перевірка може спричинити втрату важливих даних охорони здоров'я (наприклад, одне поле має неочікуване значення через неправильне введення даних, тому всі дані, пов'язані з ресурсом, буде втрачено);

- під час валідації можуть виникнути невдалі історичні дані, які неможливо виправити, і це зазвичай важко перевірити (і погано перевірено).

З іншого боку, перевірка під час виробничого використання може бути дуже важливою:

- перевірка може знадобитися з міркувань безпеки. Зокрема, може знадобитися перевірка розповіді, щоб запобігти активному вмісту або зовнішнім посиланням;

- перевірка ресурсів у певних контрольних точках у потоці даних може сприяти кращому виявленню та відновленню на відміну від збою певної логіки програми через неправильні дані.

3 РЕАЛІЗАЦІЯ МЕТОДУ ГЕНЕРАЦІЇ ТА СЕРІАЛІЗАЦІЇ ДАНИХ ЗА СТАНДАРТОМ HL77/FHIR. ІНТЕГРАЦІЯ РЕАЛІЗОВАНОГО АЛГОРИТМУ ДО ІСНУЮЧОЇ СИСТЕМИ ІНФОРМАЦІЙНОГО СУПРОВОДУ АМБУЛАТОРНИХ ПАЦІЄНТІВ В МЕДИЧНИХ УСТАНОВАХ

3.1 Обґрунтування вибору середовища програмної реалізації

3.1.1 Angular.js для головних модулів алгоритму

У рамках кваліфікаційної роботи був створений алгоритм генералізації та серіалізації даних за стандартом HL7/FHIR. Була досліджена його структура, основні і поняття. Програмна реалізація алгоритму генерації та серіалізації даних за стандартом HL7/FHIR була реалізована у модульному вигляді (складається із кількох бібліотек – функціональних модулів) та інтегрована у медичну інформаційну систему супроводу амбулаторних пацієнтів в медичних установах. Розроблений алгоритм дозволяє використовувати стандарт HL7/FHIR для генерації, відображення, серіалізації та десеріалізації формулярів в медичній інформаційній системі.

Для реалізації функціональних модулів була використана мова програмування Angular.js.

Angular.js – це середовище виконання JavaScript з відкритим вихідним кодом, як виконує код поза браузером. Платформа була створена в 2009 році Райаном Далем для швидких мережевих застосунків, які можна легко масштабувати. Раніше цю роль виконував Apache HTTP Server, але з меншою ефективністю.

Крім того, завдяки Node.js розробники JavaScripts використовують серверні сценарії для створення динамічних вебсторінок. Ці сценарії виконують запити на стороні сервера або вебсервера а не на стороні клієнта

чи у браузері. Це пришвидшує завантаження та зменшує навантаження на центральний процесор.

Для тих же результатів можна використовувати статичні вебсторінки, але це вдарить по контенту – він буде менш інтерактивним та більш важким.

Основні переваги Angular.js:

- легкість. Angular.js дуже доступний. Його легко освоїти, але лише тільки після JavaScript та принципів об'єктно-орієнтованого програмування.

- простота. Angular.js використовує слоган «JavaScript everywhere». Тобто розробнику не потрібно перемикає мови програмування між серверною та зовнішньою розробкою, що призводить до меншої ваги проекту.

- швидкість. Програмування в середовищі Angular.js відбувається швидше й простіше за багатьох конкурентів, тому скорочується час на розробку й загалом виведення продукту на ринок.

- масштабування. Це взагалі було однією з головних причин для створення Angular.js – запропонувати більш масштабовану альтернативу Apache. Angular.js може обробляти кілька одночасних підключень.

- ком'юніті. Основна перевага широкої спільноти – підтримка та зворотній зв'язок. Крім того, учасники спільноти використовують усі переваги відкритого вихідного коду Angular.js, створюючи чисельні та нескінченні інструменти та бібліотеки для полегшення процесу розробки.

3.1.2 Java для блоку інтеграції

Java – це об'єктно-орієнтована мова програмування, випущена у 1995 році як основний компонент платформи Java із незалежністю від архітектури.

Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи

системній платформі без змін у початковому коді та додаткової перекомпіляції.

Цього можна досягти, компілюючи початковий Java код у байт-код, який є спрощеними машинними командами. Потім програму можна виконати на будь-якій платформі, що має встановлені віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної операційної системи і процесора. Зараз віртуальні машини Java існують для більшості процесорів і операційних систем.

Стандартні бібліотеки забезпечують загальний спосіб доступу до таких особливостей, як обробка мультимедії, багатопотоковість та роботу з мережею. Також Java має об'єктно-орієнтовану концепцію, надзвичайний рівень безпеки та відкритий вихідний код із незкінченною кількістю бібліотек, які розробляє спільнота.

3.1.3 Thymeleaf як прошарок між алгоритмом та інтеграційним блоком

Thymeleaf – це Java XML/XHTML/HTML5 шаблонізатор, який може працювати як в веб (за допомогою сервлета), так і не у вебсередовищі. Від добре підходить для обслуговування XHTML/HTML5 на рівні уявлення веб-додатків на основі MVC (Model View Controller), але він може й обробляти будь-який файл XML навіть в автономному середовищі. Він повністю підтримує інтеграцію Spring Framework.

У веб-додатках Thymeleaf виступає як заміна JSP (JavaServer Pages) та реалізує концепцію Natural Templates: файли шаблонів, які можна відкрити напряму в браузері та які все ще відображаються правильно як вебсторінки.

Thymeleaf має відкритий вихідний код та ліцензію Apache License 2.0.

Можливості Thymeleaf:

- Java шаблонізатор форматів XML, XHTML, HTML5;
- працює як у вебсередовищі, так і в автономних системах;

- побудований на наборі модульних функцій, що називають діалектами. Один із стандартних діалектів – SpEL (Spring Standard);
- декілька режимів шаблону: перевірка XML та XHTML на відповідність DTD (Document type definition), перевірка HTML5 із заміною застарілого коду на XML;
- повна чи розширювана інтернаціоналізація;
- кеш шаблону, який скорочує ввід та вивід – висока швидкість;
- надзвичайно масштабований та розширюваний. Якщо треба, то можна використовувати як структуру механізмів шаблонів;
- повна документація із прикладами.

3.1.4 Docker

Docker – це програмне забезпечення з відкритим кодом, що є найпопулярнішою платформою для управління ізольованими контейнерами з використанням Linux. Він доповнюється високорівневими API, які дозволяють керувати контейнерами на рівні ізоляції окремих процесів чи груп процесів. Зокрема, Docker дозволяє не переймаючись вмістом контейнера запускати дозвільні процеси в режимі ізоляції і потім переносити і клонувати сформовані для даних процесів контейнери на інші сервери, беручи на себе всю роботу зі створення, обслуговування та підтримки контейнерів.

Основні можливості Docker:

- можливість розміщення в ізольованому оточенні різної начинки, що включає різні комбінації файлів, бібліотек, конфігурацій, скриптів тощо;
- підтримка роботи на будь-якому комп'ютері на базі архітектури x86 та x64, починаючи від портативних систем, таких як ноутбуки, та закінчуючи великими серверами та віртуальними машинами;

- використання легковагих контейнерів для ізоляції процесів від інших процесів і основної системи;
- оскільки контейнери використовують свою власну самодостатню файловою систему, не важливо де і коли вони запускаються;
- ізоляція на рівні файлової системи;
- ізоляція ресурсів: пам'яті, CPU тощо;
- ізоляція на рівні мережи: кожен ізольований процес має доступ тільки до пов'язаного з контейнером мережевого простору імен, включаючи мережевий інтерфейс і його IP-адреси;
- коренева файлова система контейнера створюється з використанням механізмів «сору-on-write»;
- можливість використовувати інтерактивну програмну оболонку та команди;
- можливість створення контейнерів, що містять складні програмні стеки, через зв'язування між собою вже існуючих контейнерів, що містять складові частини формованого стека. Зв'язування здійснюється не через злиття вмісту, а через забезпечення взаємодії між контейнерами (мережевий тунель).

Docker складається із чотирьох основних елементів:

- демон Docker;
- клієнт Docker;
- образ Docker;
- контейнер Docker.

3.1.5 Середовище програмної реалізації

Для реалізації алгоритмів генерації та серіалізації даних за стандартом HL7/FHIR на мові програмування Angular.js було обране середовище

програмування IntelliJ WebStorm. Це обумовлено тим, що WebStorm повністю адаптований під розробку на мовах Web, і в тому числі Angular.js.

IntelliJ WebStorm – інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, розроблена на основі платформи IntelliJ IDEA. WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей.

WebStorm забезпечують автодоповнення, аналіз коду «на льоту», навігацію по коду, автоматичний рефакторинг, зневадження та інтеграції з системами управління версіями. Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами. Підтримуються множинна вкладеність (коли в документ на HTML вкладений скрипт на JavaScript, в який вкладено інший код HTML, всередині якого вкладений JavaScript) – в таких конструкціях підтримується коректний рефакторинг.

Основні можливості IntelliJ WebStorm:

- інтеграція з системами управління проектами Git, GitHub, Mercurial;
- інтеграція з системами відстеження помилок;
- модифікація файлів css, html, js з одночасним переглядом результатів (Live Edit чи система «on-fly»);
- віддалене розгортання за протоколами FTP та SFTP, на монтованих мережевих дисках тощо, з можливістю автоматичної синхронізації;
- можливості «Zen Coding» та «Emmet».

Для реалізації інтеграційного модуля було використано середовище Програмування IntelliJ IDEA 2021.3.3. Це обумовлено тим, що IDEA дуже добре адаптована під програмування мовою Java і значно краща та зручніша ніж конкурентні рішення.

IntelliJ IDEA – це особливе середовище програмування або інтегроване середовище розробки (IDE), багато в чому призначене для Java. Це середовище використовується спеціально для розробки програм. Він розроблений компанією під назвою JetBrains, яку офіційно називали IntelliJ. Він доступний у двох виданнях: Community Edition, що має ліцензію Apache

2.0, і комерційне видання, відоме як Ultimate Edition. Обидва вони можуть бути використані для створення програмного забезпечення, яке можна продати. Що робить IntelliJ IDEA настільки відмінним від своїх аналогів, це його легкість у використанні, гнучкість та міцний дизайн [42].

IntelliJ IDEA був розроблений JetBrains, раніше відомим як IntelliJ. Він був вперше випущений в 2001 році, і він похвалився такими функціями, як розширена навігація по коду і можливість рефакторних кодів, що зробило його дуже популярним. Він навіть отримав відзнаку того, що був визнаний найкращим інструментом програмування на базі Java в 2010 році, виділяючи такі інструменти, як NetBeans, Eclipse та JDeveloper [42].

Було використано реляційну SQL базу даних MariaDB для зберігання персональних даних користувачів, даних HL7/FHIR «Survey», та отриманих з документів даних.

Для зборки проекту було використано використана системи на основі Maven та NPM.

Для тестування базового та GUI функціоналу було використано Junit, Mockito та зв'язку Selenium із Cucumber.

Як базовий фреймворк було використано Spring Boot.

Spring Framework – це програмний каркас (фреймворк) з відкритим кодом та контейнери з підтримкою інверсії управління для платформи Java [42].

Spring Boot Framework надає можливість використання моделі програмування та автоматичної конфігурації сучасних EE Java-програм на будь-якій платформі розгортання із JVM.

Ключовим елементом концепції Spring є підтримка на рівні додатків: Spring фокусується на дрібних та не таких важливих як бізнес логіка частинах корпоративних додатків, щоб розробник мог зосередитись на бізнес логіці та предметній області проекту без зайвих зв'язків із конкретними середовищами розгортання.

Spring Framework складається з декількох основних модулів.

Spring Data. Цей фреймворк надає звичну та послідовну модель програмування для доступу до бази даних, зберігаючи при цьому особливі та базові риси цієї бази даних.

Spring Security – це потужна і налаштована оболонка забезпечення захисту додатку із аутентифікацією та контролем доступу. Фактично, це стандарт забезпечення безпеки проєктів, що побудовані на базі Spring Framework.

У таблиці 3.1 наведені версії усіх використовуваних технологій та компонентів інтеграційного модуля.

Таблиця 3.1 – Версії технологій розробки

Технологія	Версія	Мета
spring-boot-starter-parent	2.4.0	Базовий модуль Spring
spring-boot-starter-web	2.4.3	Базовий модуль Spring для створення Web-застосунку
spring-boot-starter-actuator	2.4.3	Базовий модуль Spring
spring-boot-starter-cache	2.4.3	Базовий модуль Spring для управління кешем
springfox-swagger2	2.9.2	Тестування контролерів
springfox-swagger-ui	2.9.2	Графічне відображення тестування контролерів
Lombok	1.18.12	Прискорення роботи з класами
commons-lang3	3.11	Набір інструментів для роботи з базовими колекціями
commons-csv	1.8	Набір інструментів для роботи з .csv файлами
commons-collections4	4.4	Набір колекцій
Caffeine	2.9.0	Набір інструментів для роботи з кешем

Продовження таблиці 3.1

Технологія	Версія	Мета
liquibase-core	4.3.1	Автоматизація адміністрування бази даних
mariadb-java-client	2.7.2	Клієнт MariaDB
spring-data-jpa-datatables	5.0.0	Модуль Spring для роботи з базами даних у вигляді datatables
hibernate-types-52	2.10.3	Додаткові типи для роботи з базами даних
spring-boot-starter-thymeleaf	2.4.3	Модуль Spring для з'єднання back-end та front-end
thymeleaf-extras-springsecurity5	3.0.4	Модуль Spring для забезпечення безпеки з'єднання back-end та front-end
thymeleaf-layout-dialect	2.5.2	Діалект для створення шаблонів
thymeleaf	3.0.12	Фреймворк для роботи з front-end
spring-boot-starter-data-jpa	2.4.3	Модуль Spring для роботи з базами даних
spring-boot-starter-mail	2.4.3	Модуль Spring для роботи з email
commons-email	1.5	Набір інструментів для роботи з email
javax.mail	1.6.2	Набір інструментів для роботи з email
org.apache.commons	4.4	Набір інструментів для роботи з базовими типами даних Java
jjwt	0.9.1	Модуль для формування JsonWebToken

Кінець таблиці 3.1

Технологія	Версія	Мета
jjwt-impl	0.11.2	Імплементатії модулю для формування JsonWebToken

У таблиці 3.2 наведені версії усіх використовуваних технологій та компонентів модулів HL7/FHIR.

Таблиця 3.2 – Версії технологій розробки

Технологія	Версія
@lhncbc/ucum-lhc	4.1.3
angular	1.8.2
angular-animate	1.8.2
angular-ui-bootstrap	1.3.3
autocomplete-lhc	18.1.1
bootstrap	3.4.1
Bootstrap-ui-datetime-picker	2.6.4
bootstrap-ui-datetime-picker	1.1.15
element-resize-detector	2.6.4
fast-deep-equal	3.1.3
fhirpath	2.9.1
jquery	3.6.0
jquery-ui	1.12.1
js-class	0.0.3
moment	2.24.0
ngSmoothScroll	2.0.1
promise-polyfill	8.2.0
whatwg-fetch	3.0.0

Продовження таблиці 3.2

Технологія	Версія
@babel/core	7.1.2
@babel/preset-env	7.1.0
babel-loader	8.0.4
bower	1.7.9
chai	3.5.0
clean-css	4.2.1
clean-css-cli	4.3.0
css-loader	2.1.1
eslint	7.32.0
eslint-plugin-security	1.4.0
file-loader	3.0.1
grunt	1.4.1
grunt-angular-templates	1.2.0
grunt-cli	1.4.3
grunt-contrib-connect	2.0.0
grunt-contrib-jshint	2.1.0
grunt-contrib-watch	1.1.0
grunt-karma	3.0.1
grunt-newer	1.1.0
grunt-protractor-runner	5.0.0
grunt-wiredep	3.0.1
jasmine-core	2.3.4
jasmine-spec-reporter	2.7.0
jit-grunt	0.10.0
jshint-stylish	1.0.0
karma	6.3.4
karma-chai	0.1.0

Кінець таблиці 3.2

Технологія	Версія
karma-chrome-launcher	2.0.0
karma-jasmine	0.3.6
karma-mocha	2.0.0
karma-mocha-reporter	2.2.5
karma-requirejs	1.1.0
mini-css-extract-plugin	0.5.0
mocha	9.1.3
moment-locales-webpack-plugin	1.0.7
node-sass	4.14.1
phantomjs-prebuilt	2.1.16
protractor	5.4.2
requirejs	2.3.2
serve-static	1.13.2
style-loader	0.23.1
time-grunt	1.0.0
tmp	0.2.1
uglify-js	3.4.10
webpack	4.42.1
webpack-cli	3.3.11
wiredep	4.0.0

У таблиці 3.3 наведені версії усіх використовуваних технологій та компонентів фронтенд частини.

Таблиця 3.3 – Версії технологій розробки

Технологія	Версія
css-loader	6

Кінець таблиці 3.3

Технологія	Версія
sass	1.54
sass-loader	13
style-loader	3
webpack	5.74
webpack-cli	4
jquery	3.6.0
mocha	10
jsdom	20
chai	*
chai-spies	*
mocha-multi-reporters	*
mocha-sonarqube-reporter	*
nyc	*
sinon	*

3.2 Програмна реалізація back-end

Реалізація back-end частини алгоритму генерації та серіалізації даних за стандартом HL7/FHIR полягає в створенні таких модулів:

- модуль формування структури;
- модуль редрерінгу;
- модуль валідації;
- модуль показу.

На рисунку 3.1 схематично показано взаємодію цих модулів.

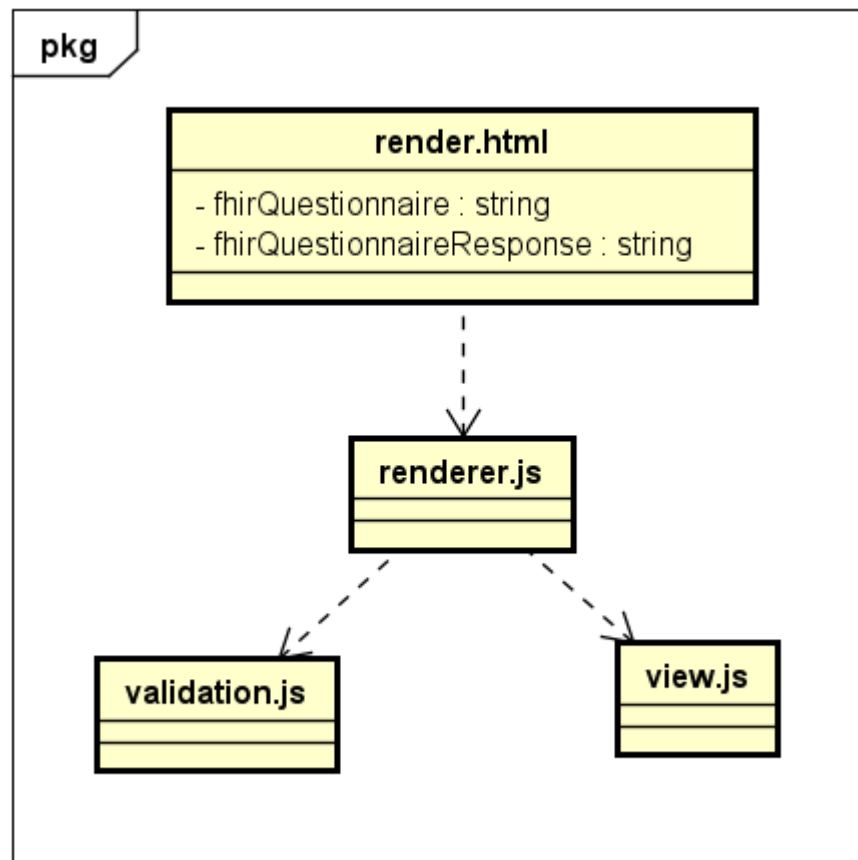


Рисунок 3.1 – Модулі алгоритму генерації та серіалізації даних HL7/FHIR

У свою чергу модуль формування структури повинен мати такі підмодулі:

- підмодуль обробки діагностичних записів;
- підмодуль експорту загальних частин;
- підмодуль процесору виразів;
- підмодуль обробки загальних частин FHIR;
- підмодулі обробки SDC;
- підмодуль підстандарту R4;
- підмодуль підстандарту STU3;
- підмодуль загальних функціональних блоків (утілітарний);
- блок обробки темплейтів HTML.

Модуль рендерінгу також повинен поділятися на такі підмодулі:

- підмодуль налаштування загальних атрибутів рендереру;
- підмодуль настройки функціональних кнопок та посилань;
- підмодуль налаштування івентів.

Модуль показу буде мати загальні налаштування для роботи та відображення структури даних FHIR, стилі сторінок.

Модуль валідації за документацією стандарту HL7/FHIR обов'язково повинен мати функціональні підмодулі, що відповідають за виконання валідації, формування повідомлень, формування звіту валідації, обробку івентів та відправлення звіту валідації.

3.3 База даних

У якості бази даних використовується реляційний підхід збереження даних, а також була необхідна висока швидкість обміну та обробки інформації, тому було обрано базу даних MariaDB.

MariaDB – це реляційна база даних, яка використовується для зберігання невеликих за обсягом даних за певною реляційною моделлю, як і в інших традиційних реляційних база даних [43].

Ця модель впорядковує дані в одну або кілька таблиць (або «відносин») стовпців і рядків з унікальним ключем, що ідентифікує кожен рядок. Рядки також називають записами або кортежами. Стовпці також називаються атрибутами. Як правило, кожна таблиця / відношення представляє один «тип сутності». Рядки представляють екземпляри цього типу сутності, а стовпці представляють значення, віднесені до цього екземпляра [44].

Оскільки дані зберігаються за канонами реляційних баз даних, то сутність відповідає всім стандартам таких баз [45–49]:

- кожен атрибут відносини має унікальне в даному відношенні ім'я;
- кожен атрибут визначений на одному домені;

- на одному і тому ж домені може бути визначено декілька атрибутів;
- ім'я атрибута може збігатися з ім'ям домена;
- порядок проходження атрибутів не встановлюється (атрибути у визначенні схеми відносини не впорядковані);
- у відношенні немає співпадаючих кортежів (кожен кортеж унікальний);
- порядок проходження кортежів не встановлюється (кортежі можуть стосуватися не впорядковані);
- сутність має ім'я, яке в схемі бази даних відрізняється від імен всіх інших відносин.

Для інтеграції алгоритму генерації та серіалізації даних за алгоритмом HL7/FHIR необхідно модифікувати існуючу базу даних МІС.

Структура існуючої бази даних МІС зображена на рисунку 3.2.

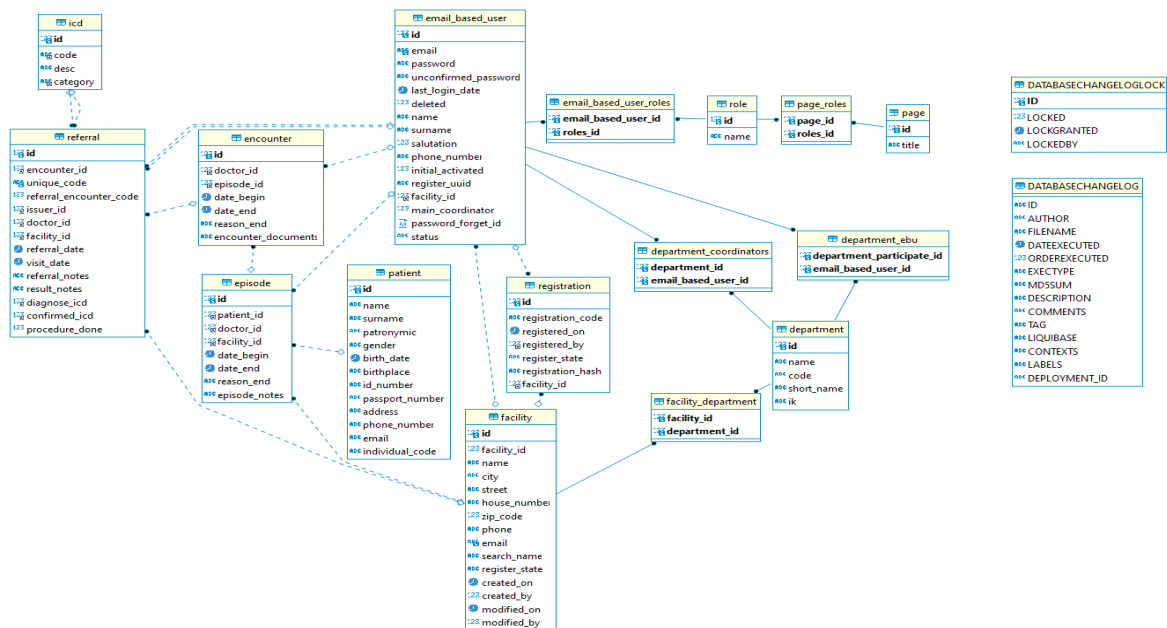


Рисунок 3.2 – Структура бази даних

Для реалізації правильної роботи алгоритму було прийняте рішення про додавання нових таблиць та зовнішніх зв'язків, як показано на рисунку 3.3.

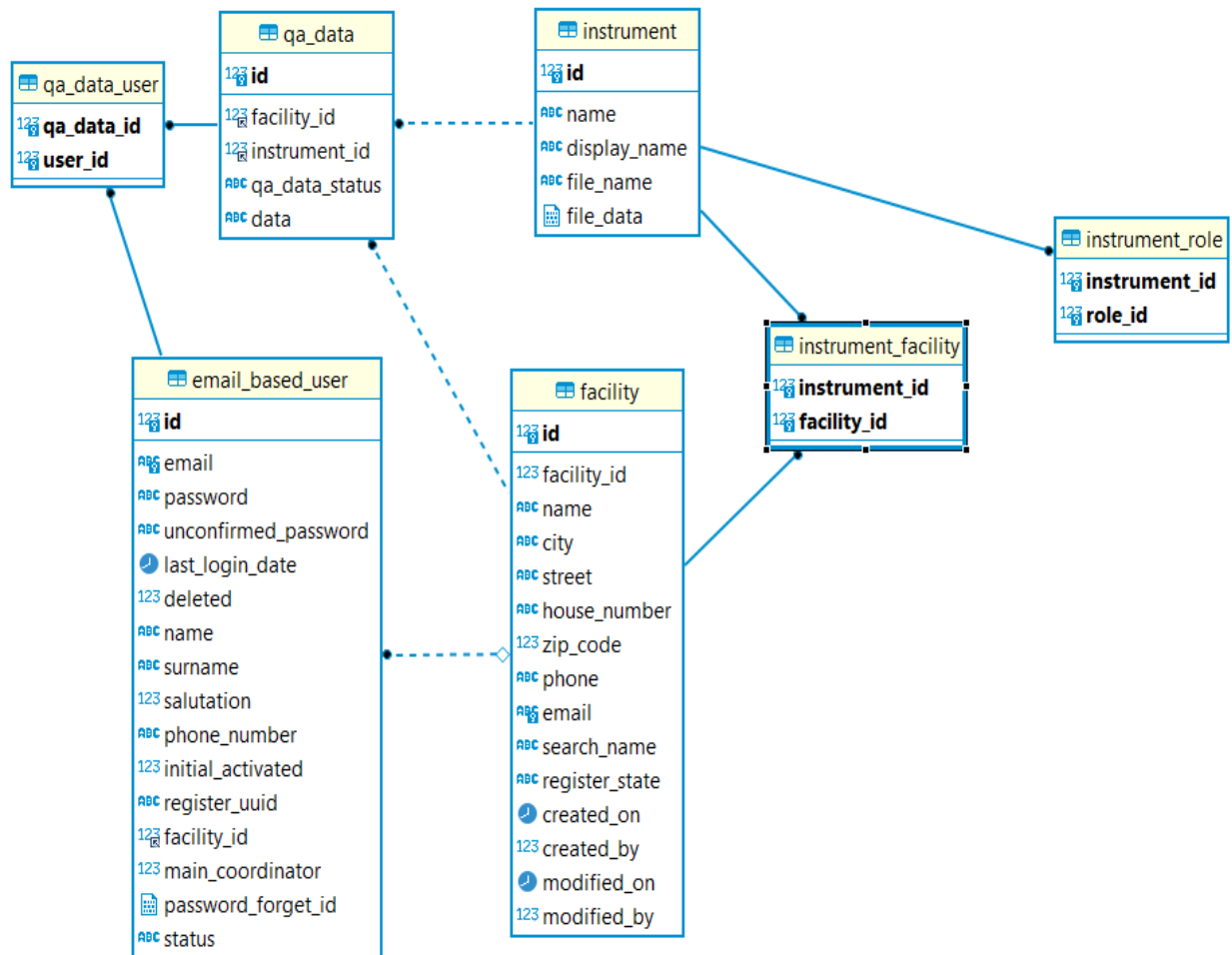


Рисунок 3.3 – Нові таблиці та зв'язки у структурі бази даних

3.4 Програмна реалізація front-end

Front-end частина створена за допомогою Spring Boot Web та thymeleaf.

Thymeleaf – це сучасний серверний механізм Java для вебсторінок.

З модулями для Spring Framework, безліччю інтеграцій з улюбленими інструментами та можливістю підключення власного функціоналу Thymeleaf ідеально підходить для сучасної веб-розробки HTML5 JVM в Інтернеті, хоча він може набагато більше.

Thymeleaf – надзвичайно розширюваний механізм (насправді його можна назвати платформою шаблонів), який дозволяє вам визначати і налаштовувати спосіб обробки ваших шаблонів до тонкого рівня деталізації.

Об'єкт, який застосовує деяку логіку до артефакту розмітки (тегу, текст, коментарю) називається процесором. Артефакти – це те, з чого складається діалект. З коробки основна бібліотека Thymeleaf надає діалект, званий стандартним діалектом, якого повинно бути достатньо для більшості користувачів.

3.5 Інструкція користувача

Для розпочатку роботи у програмному Web-застосунку МІС [50 – 52] користувачу необхідно перш за все скласти документ у форматі .csv із такими даними, як зазначено у таблиці 3.4, та відправити його системному адміністратору, зазначивши дані головного координатора для подальшої реєстрації.

Таблиця 3.4 – Поля файлу формату «.csv»

Поле	Розшифровка
ID	Ідентифікатор медичної установи
Facility	Назва медичної установи
Facility city	Город медичної установи
Facility street	Адрес медичної установи
Department	Відділення медичної установи

Необхідно зазначити, що ключем є комбінація двох полей: «ID» та «Department». Також відділення обов'язково мають бути у такому форматі: «id shortname [- name]». Усі доступні відділення можна знайти у таблиці 3.5.

Таблиця 3.5 – Відділення

Code	shortname
1000	DERMAAMB
1100	GASTROAMB
1200	KARDIOAMB
1300	MSKAMB
1400	ONKOAMB
1500	PNEUMOAMB
1600	GERIAAMB
1700	NEUROAMB
1800	PSYCHAMB
1900	CHYOAMB

Адміністратор у свою чергу повинен увійти до системи. Після відкриття Web-застосунку користувачу надається форма входу з можливістю відновити пароль. Вона зображена на рисунку 3.4.

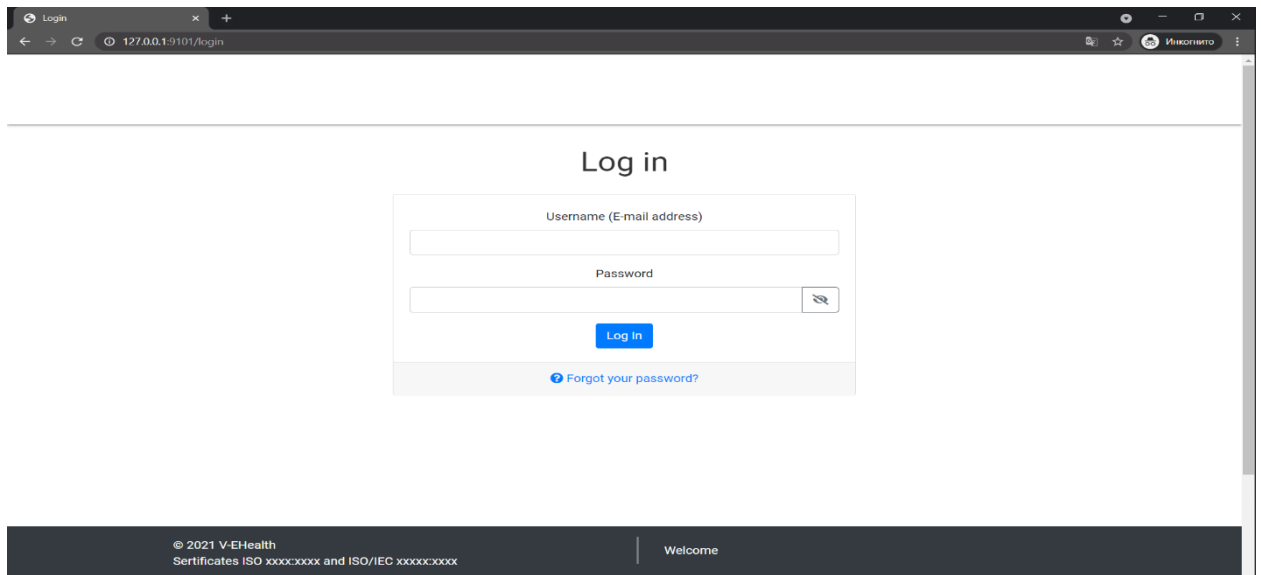


Рисунок 3.4 – Форма авторизації МІС

За необхідністю можна зайти на сторінку відновлення паролю, клікнув на надпис «Forgot your password?». Сторінка відновлення паролю зображена на рисунку 3.5.

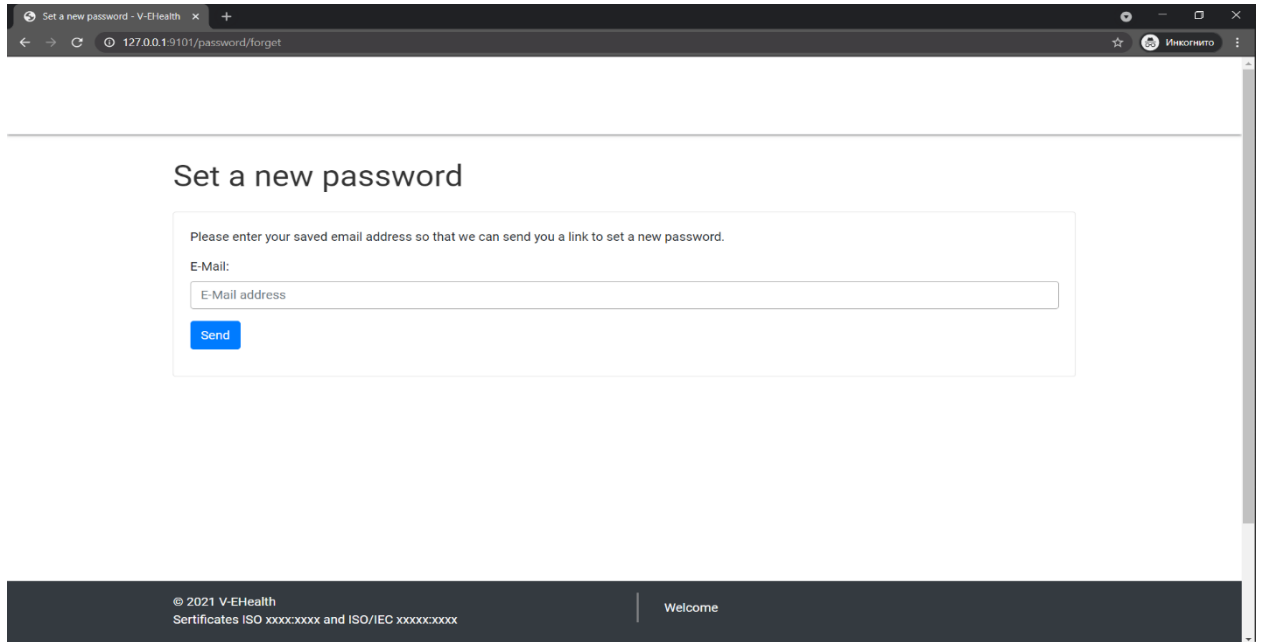


Рисунок 3.5 – Форма відновлення паролю МІС

Після авторизації адміністратор побачить головну сторінку з меню (рис. 3.6).

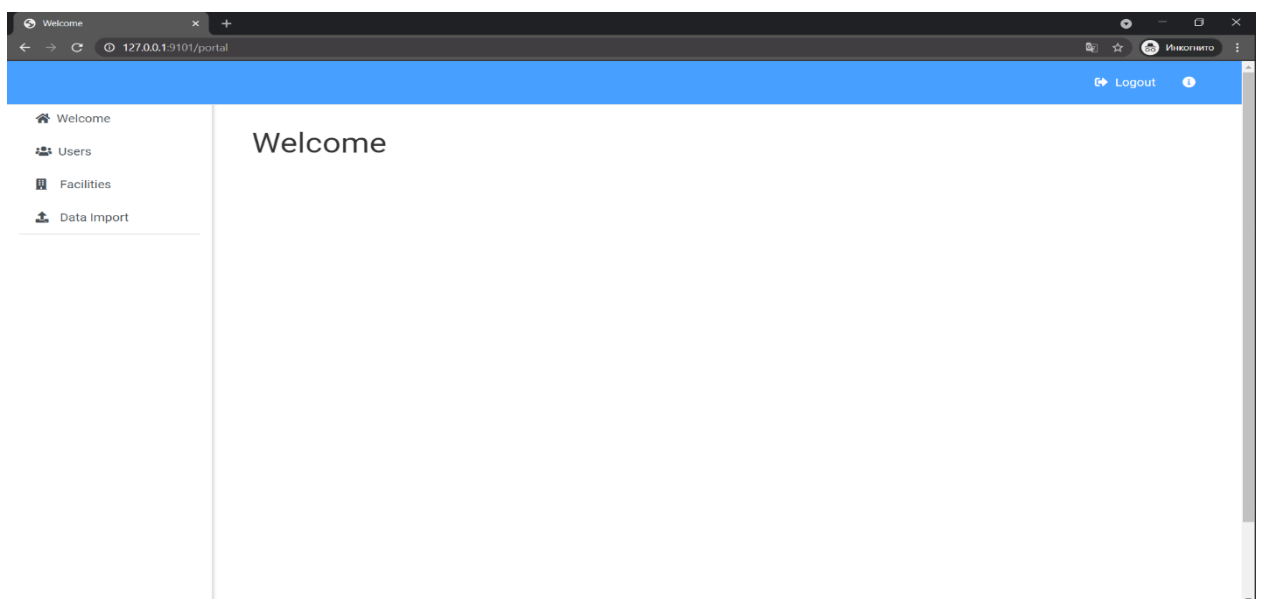


Рисунок 3.6 – Головне меню адміністратора МІС

У меню потрібно клікнути на кнопку «Data import» та обрати файл імпорту даних медичної установи у форматі .csv. За необхідності можна скачати файл прикладу, клікнувши на кнопку «Download sample file» та прийняти завантаження (рис. 3.7).

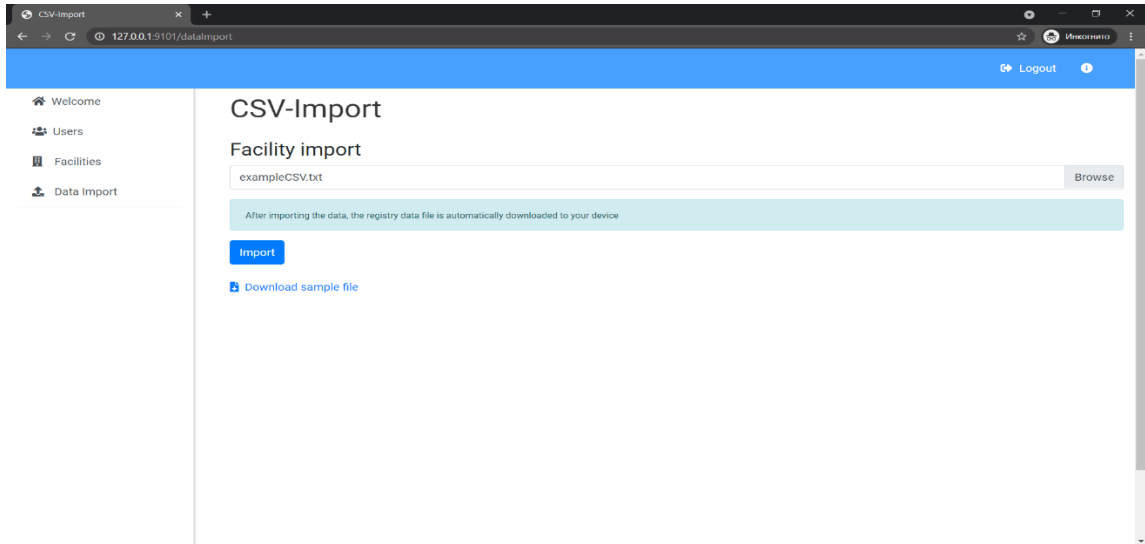


Рисунок 3.7 – Сторінка імпорту даних мед закладів

Після натискання кнопки «Import» необхідно завантажити файл до бази даних, та використати згенеровані посилання та коди для подальшої реєстрації (рис. 3.8) та ввести дані головного координатора (рис. 3.9).

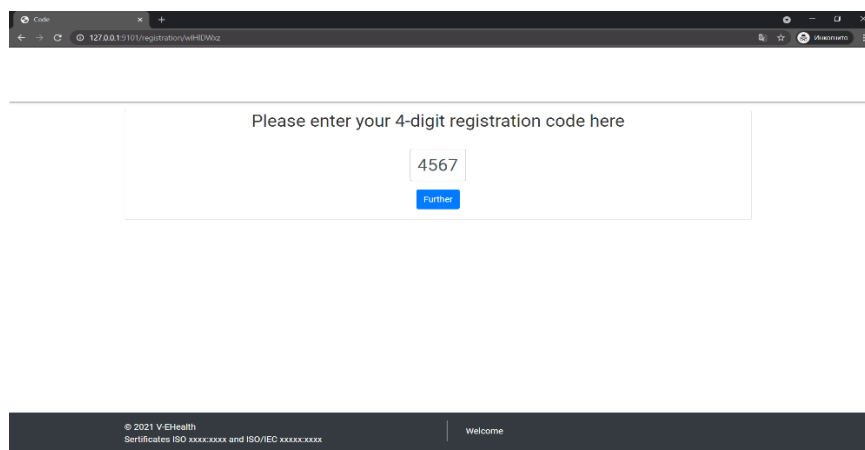


Рисунок 3.8 – Сторінка підтвердження початку реєстрації медичного закладу

Please create the coordinator

Name

Surname

E-mail address

Phone number (optional)

[Further](#)

© 2021 V-EHealth
Certificates ISO xxxxxxxx and ISO/IEC xxxxxxxx

Welcome

Рисунок 3.9 – Сторінка підтвердження початку реєстрації медичного закладу

Наступним кроком адміністратору буде повідомлено, що email із подальшими інструкціями (рис. 3.10) був відправлений на адрес, який був зазначений при реєстрації головного координатора (рис. 3.11).

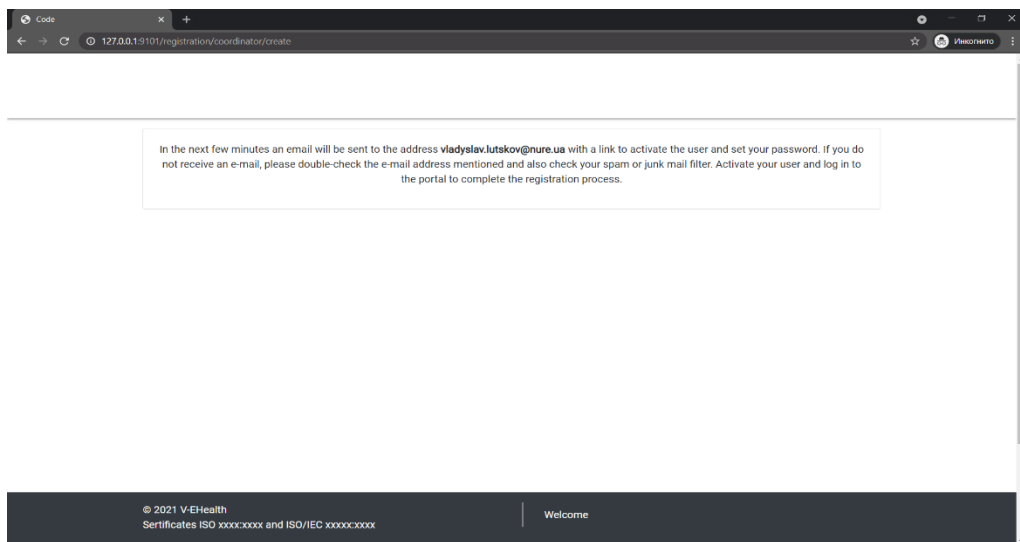


Рисунок 3.10 – Сторінка підтвердження запиту на реєстрацію головного координатора

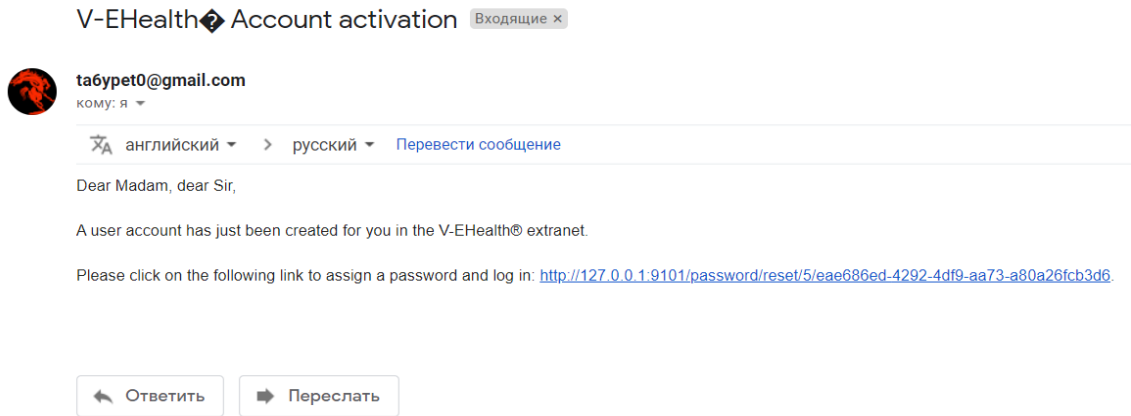


Рисунок 3.11 – Інформаційний лист з інструкцією

Після кліку по посиланню у листі відкриється вікно створення нового паролю для головного координатора (рис. 3.12).

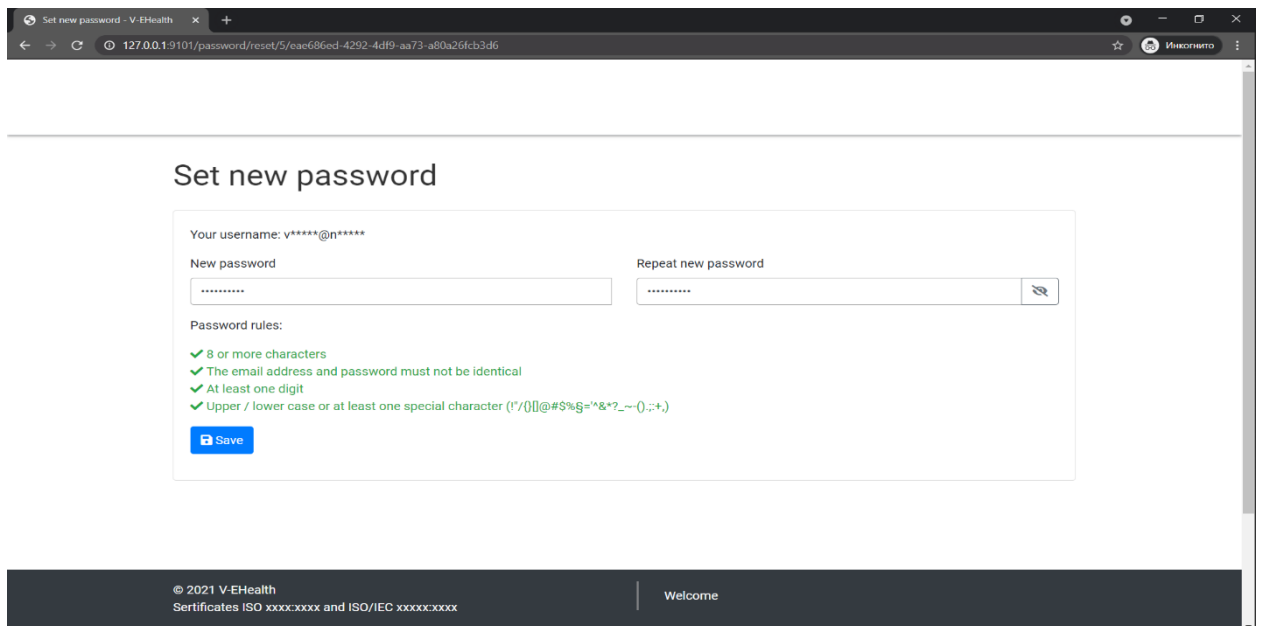


Рисунок 3.12 – Встановлення нового паролю

Коли пароль буде змінено, тільки що зареєстрований головний координатор повинен увійти до системи з зазначеним паролем та email для продовження проходження реєстрації (рис. 3.13–3.18).

Master data of the facility

Check and confirm the master data of your facility

Support-ID: 1001001

Facility name: Test Facility

Street and house number: Naukova str. 1

Postcode: 1

City: Kharkiv

E-Mail Address of the facility:

Phone number. The facility:

Confirm

© 2021 V-EHealth
Certificates ISO xxxxxxxx and ISO/IEC xxxxxxxx

Welcome

Рисунок 3.13 – Сторінка підтвердження даних медичного закладу

Participating departments

Description	Confirmation
1000 - DERMAAMB - Outpatient dermatology	<input checked="" type="radio"/> yes <input type="radio"/> no
1100 - GASTROAMB - Outpatient gastroenterology / nephrology / metabolic disease	<input checked="" type="radio"/> yes <input type="radio"/> no
1200 - KARDIOAMB - Outpatient cardiology outpatient	<input checked="" type="radio"/> yes <input type="radio"/> no
1300 - MSKAMB - Outpatient musculoskeletal diseases (including orthopedics, rheumatology)	<input checked="" type="radio"/> yes <input type="radio"/> no
1600 - GERIAAMB - Outpatient geriatrics	<input checked="" type="radio"/> yes <input type="radio"/> no
1800 - PSYCHAMB - Outpatient psychosomatic / addiction diseases	<input checked="" type="radio"/> yes <input type="radio"/> no
1900 - CHYOAMB - Outpatient children and youth	<input checked="" type="radio"/> yes <input type="radio"/> no

Add departments Save

© 2021 V-EHealth
Certificates ISO xxxxxxxx and ISO/IEC xxxxxxxx

Welcome

Рисунок 3.14 – Сторінка підтвердження даних відділень

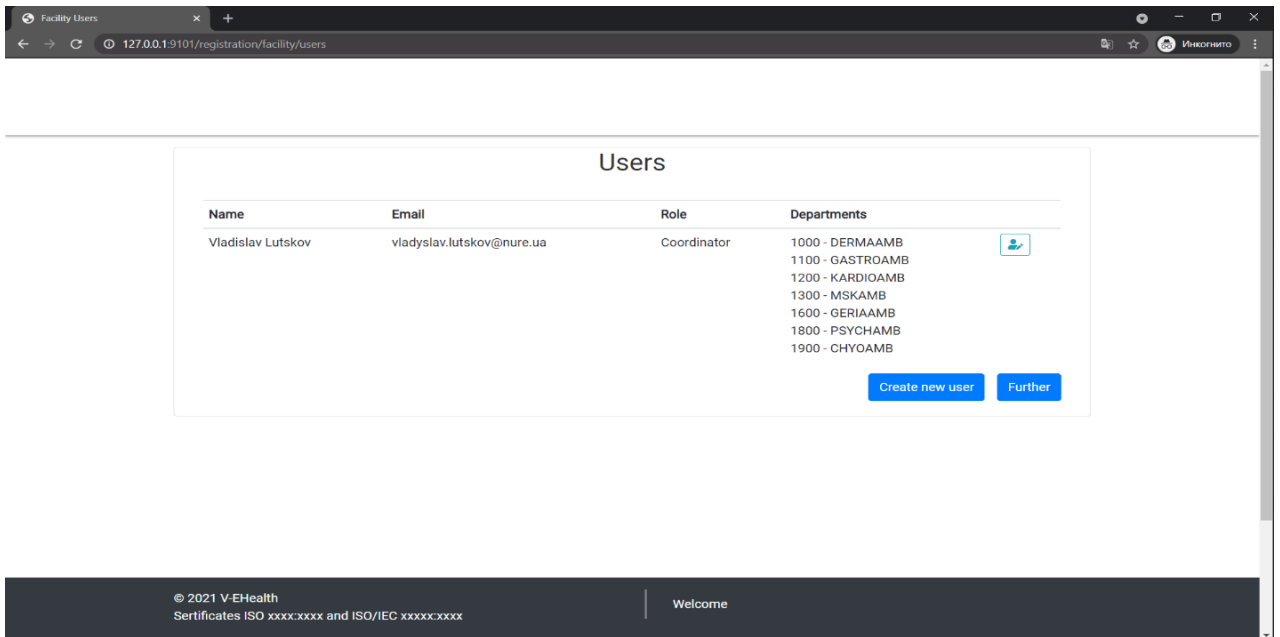


Рисунок 3.15 – Сторінка підтвердження даних замів головного координатора

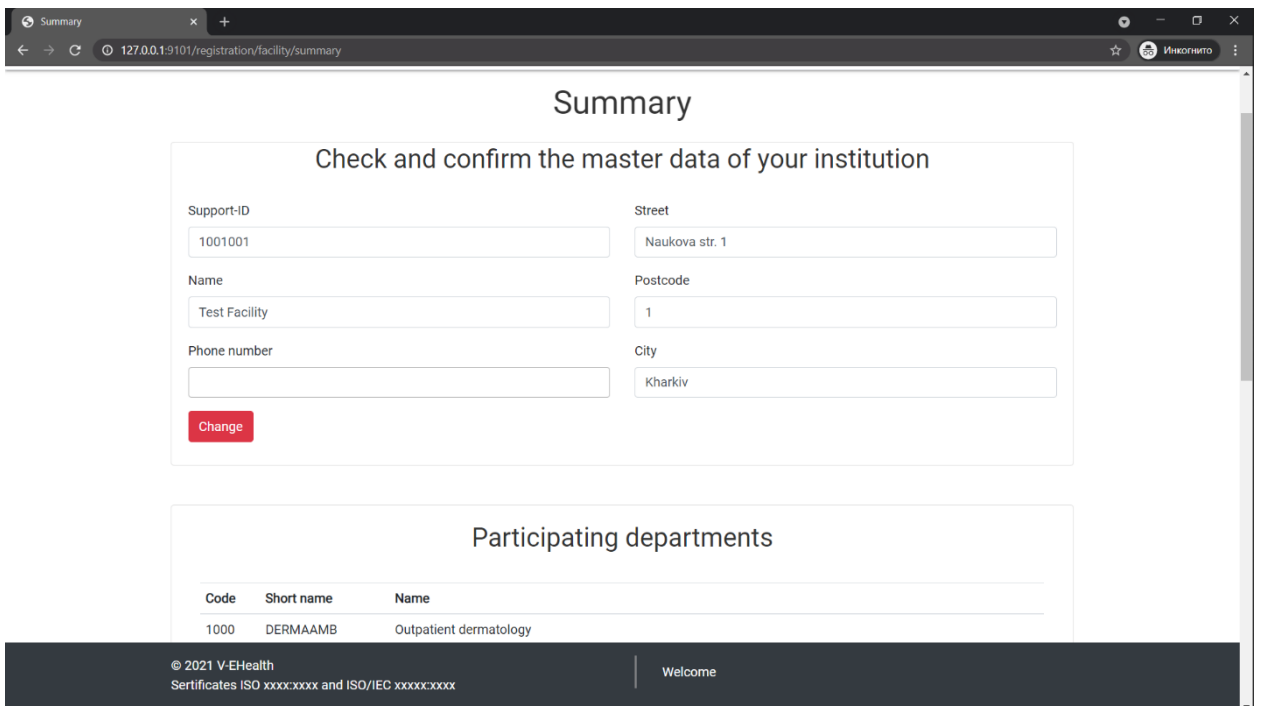


Рисунок 3.16 – Сторінка підтвердження даних реєстрації (мед. закладу)

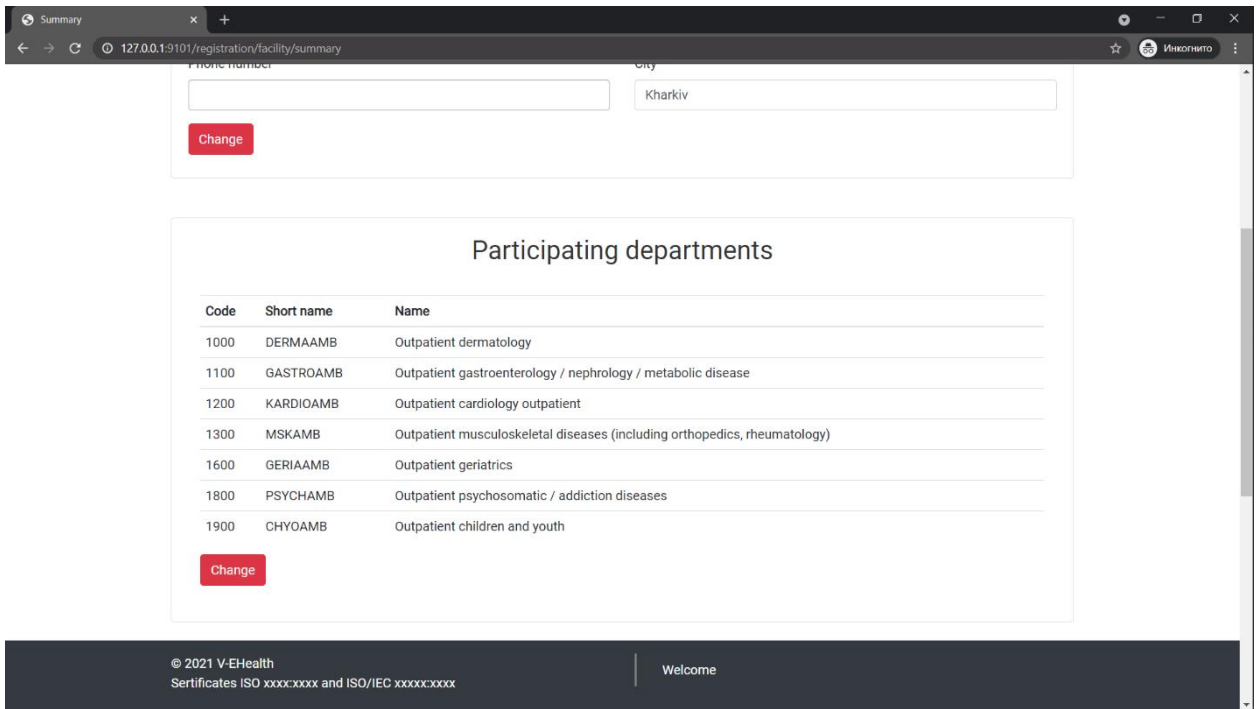


Рисунок 3.17 – Сторінка підтвердження даних реєстрації (відділень)

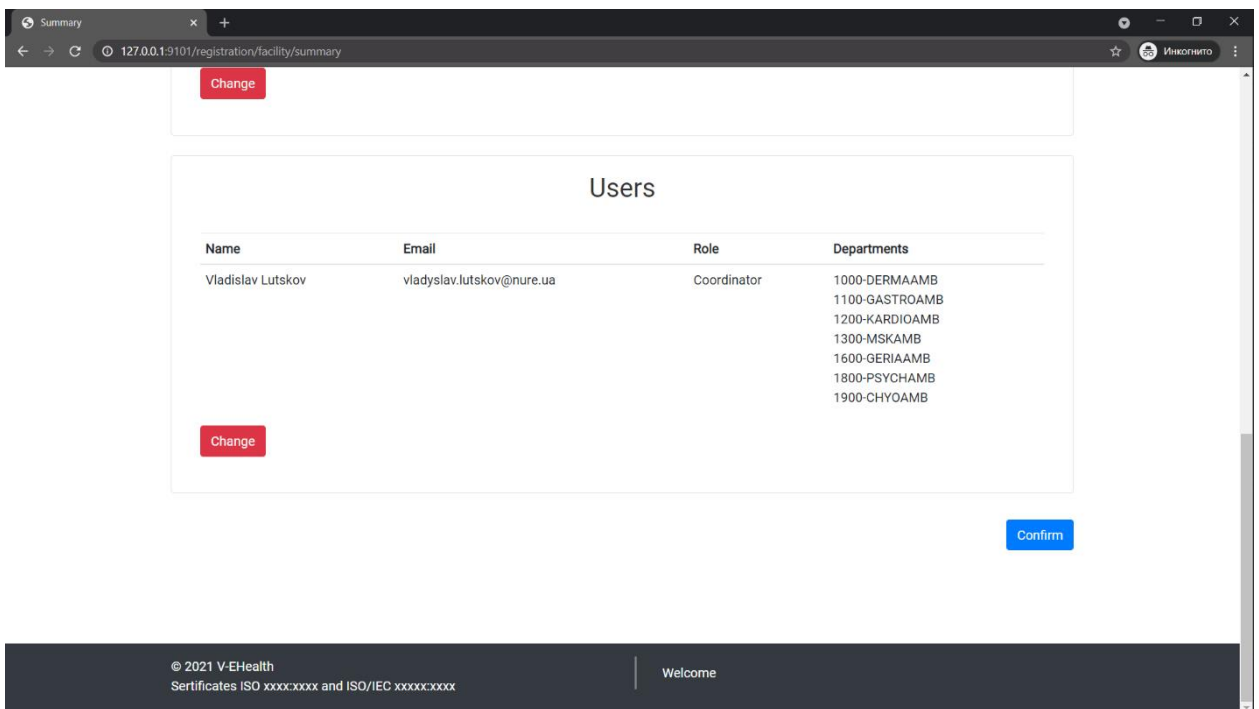


Рисунок 3.18 – Сторінка підтвердження даних реєстрації (користувачів)

Після вдалого процесу реєстрації, користувач отримає повідомлення (рис. 3.19).

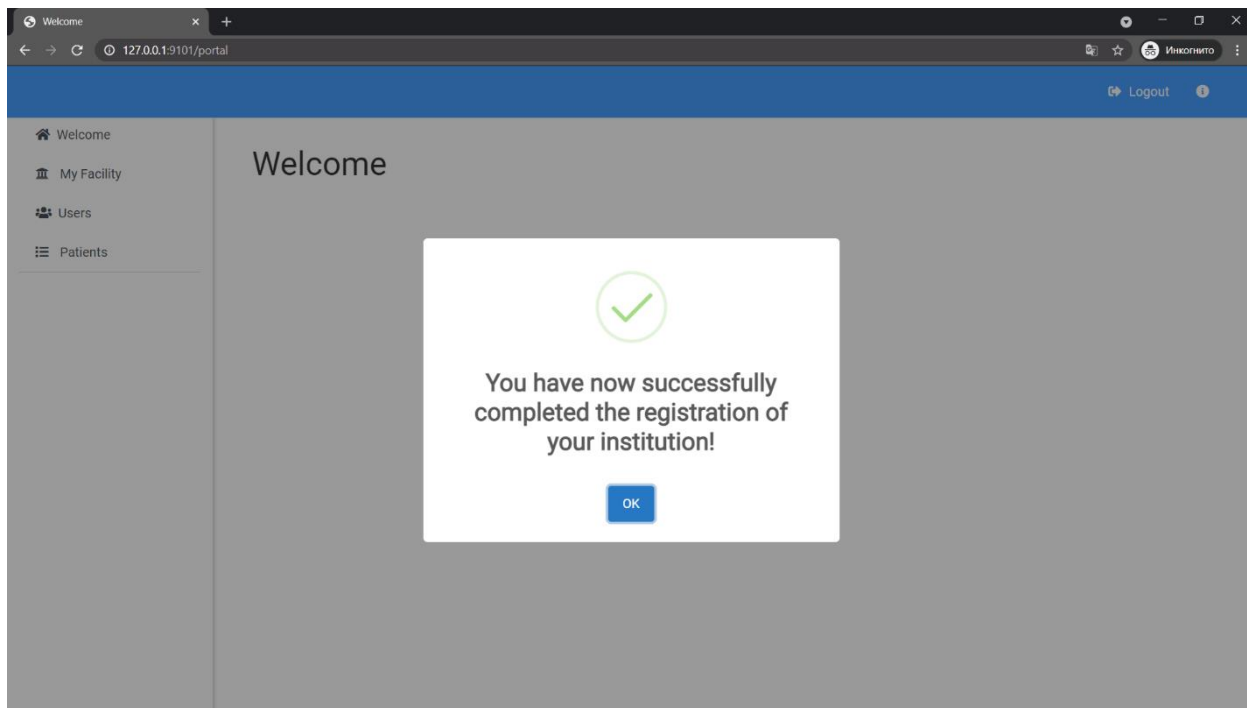


Рисунок 3.19 – Повідомлення про успішну реєстрацію медичного закладу

Слід зазначити, що одразу ж після реєстрації до системи можуть зайти лише головний координатор та його зами. Для входу інших лікарей необхідно їх спочатку зареєструвати в інформаційній системі.

Після процесу реєстрації адміністратор повинен створити запис «Instrument», використовуючи попередньо підготовлений структурний елемент FHIR та додати його до існуючої Facility (рис 3.20–3.24).

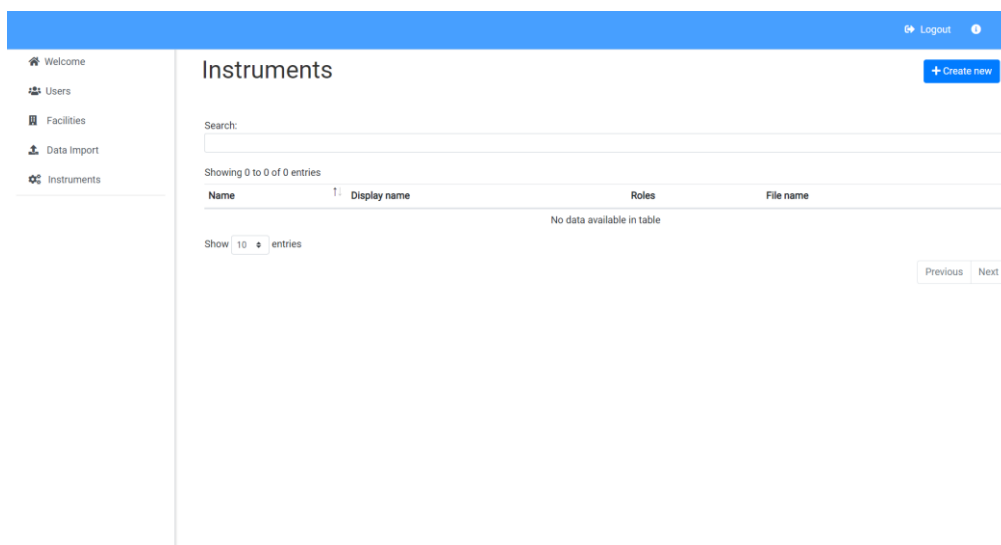


Рисунок 3.20 – Сторінка з інструментами

Create Instrument

Name
test

Display name
test

Roles
ADMINISTRATOR
COORDINATOR_MAIN
COORDINATOR_DEP
DOCTOR

Upload file
fhir_questionnaire_C (1) (1).json

Save Cancel

Рисунок 3.21 – Створення нового інструменту

Add instrument

Search:

Showing 1 to 1 of 1 entries

Name	Display name	Roles	File name
test	test	COORDINATOR_DEP COORDINATOR_MAIN DOCTOR	fhir_questionnaire_C (1) (1).json

Show 10 entries

Previous 1 Next

Showing 1 to 1 of 1 entries

Name	Display name	Roles	File name
No matching records found			

Show 10 entries

Previous 1 Next

© 2022 V-EHealth
Certificates ISO xxxxxxxx and ISO/IEC xxxxxxxx

Welcome

Рисунок 3.22 – Модальне вікно додавання нового інструменту до існуючої Facility на сторінці «Facility details»

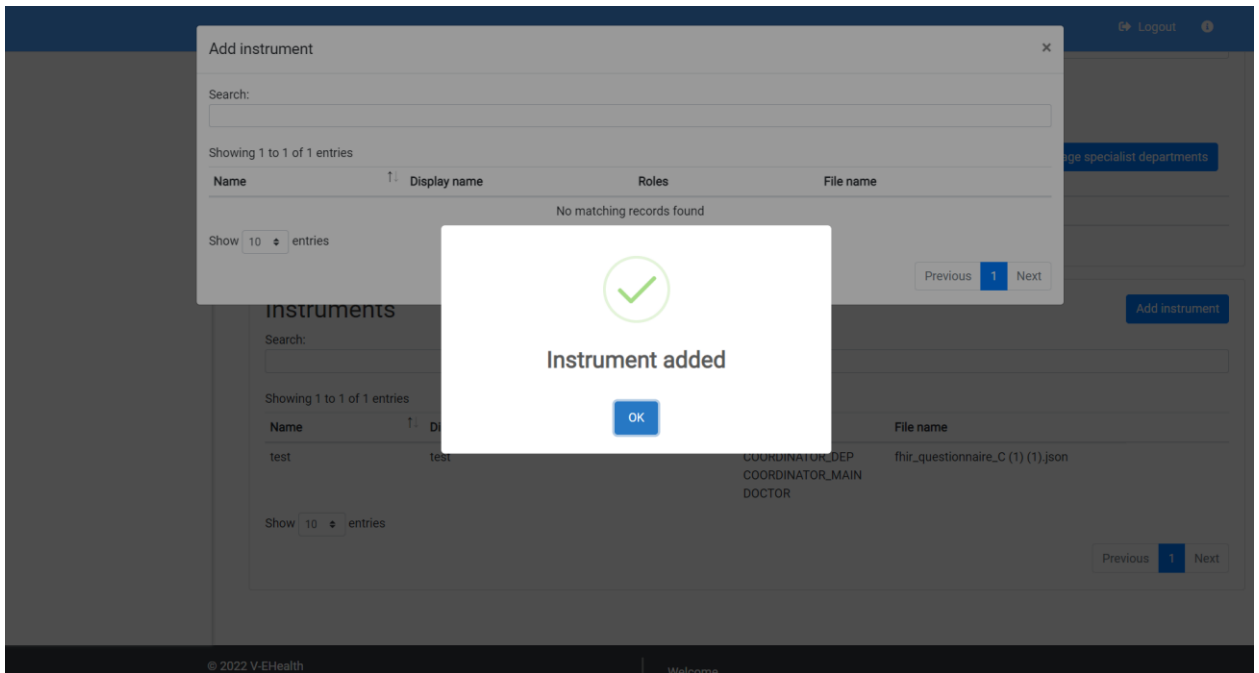


Рисунок 3.23 – Повідомлення про вдале додавання нового інструменту до Facility

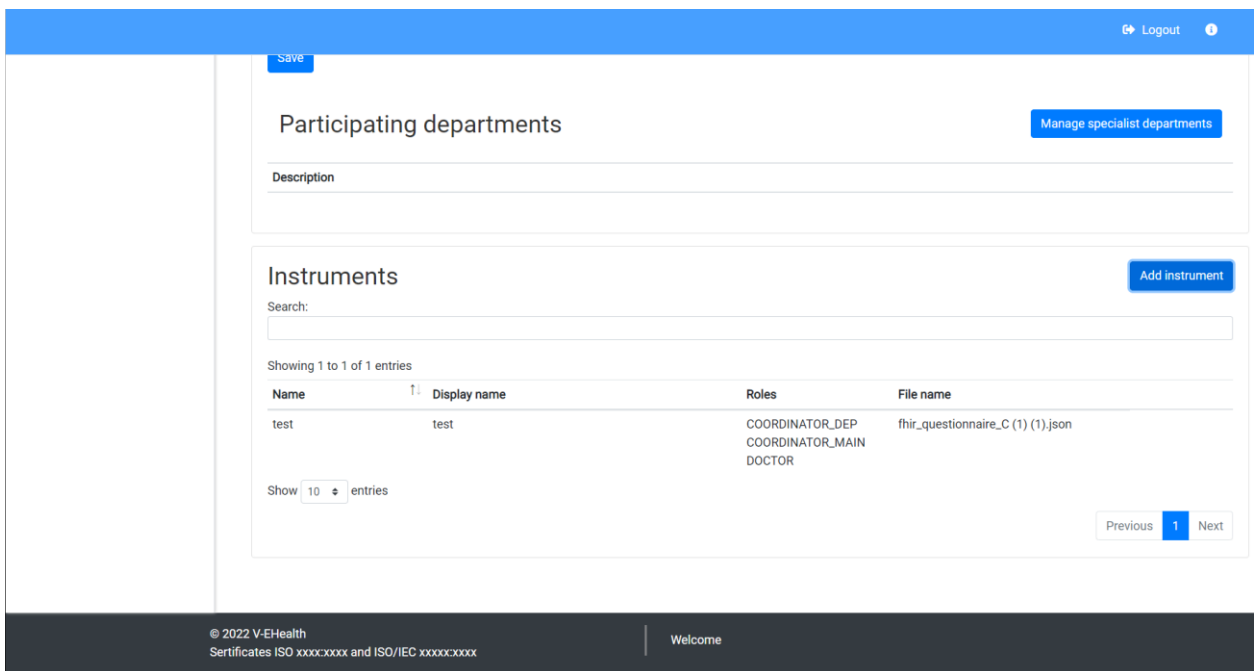


Рисунок 3.24 – Оновлені дані про інструменти, що належать до поточної Facility

Після чого користувачі із тими ролями, що були зазначені у інструменті, зможуть вільно користуватись усіма знегерованими FHIR

формулярами за цими інструментами через сторінку «Survey» (рис. 3.25 – рис. 3.28).

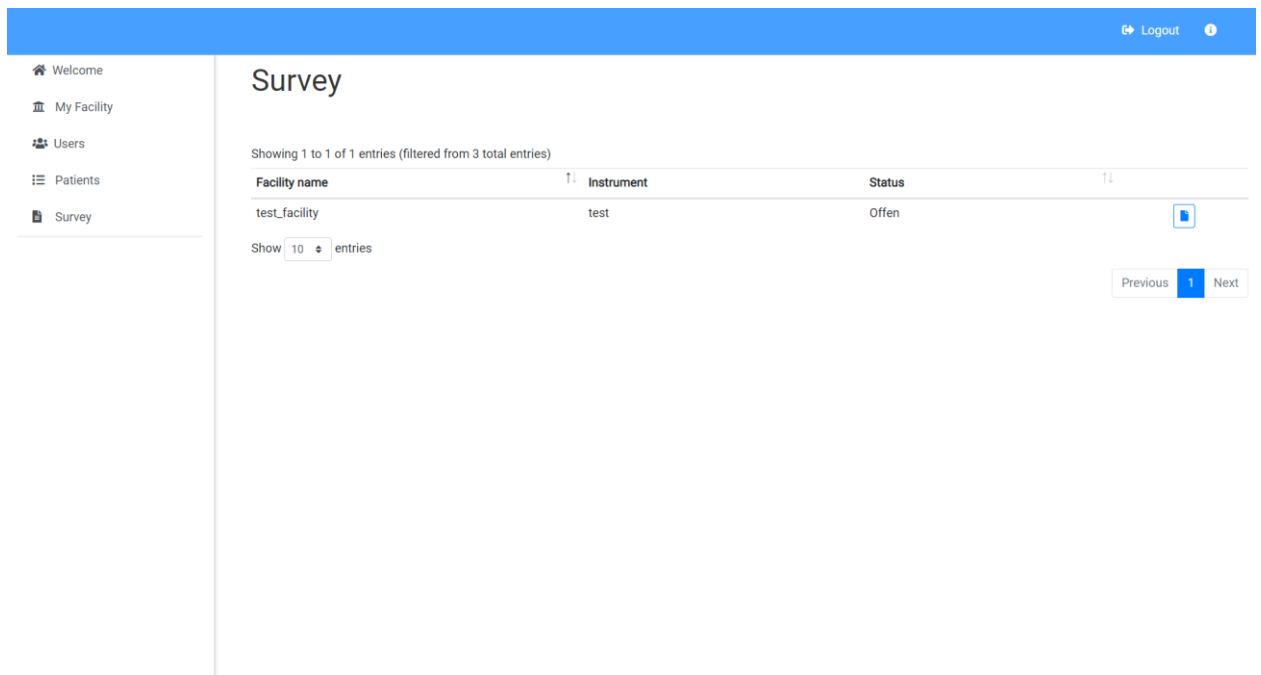


Рисунок 3.25 – Сторінка «Survey» із згенерованим записом FHIR

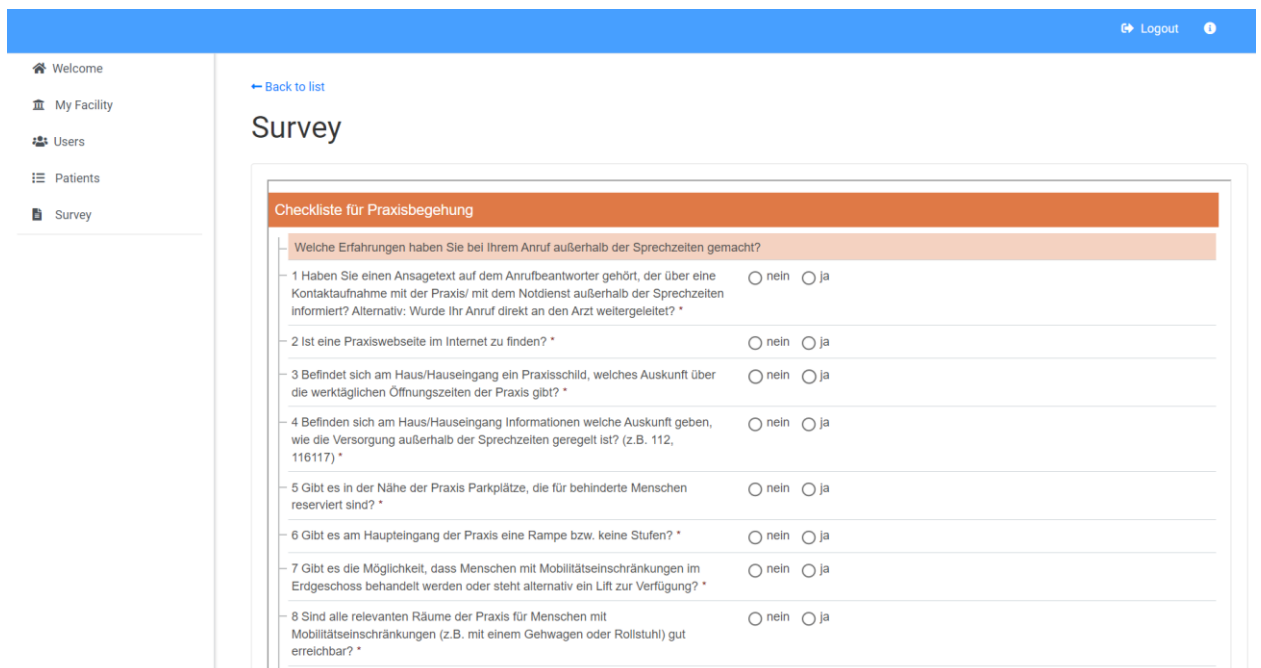


Рисунок 3.26 – Формуляр на сторінці «Survey» із згенерованим формуляром FHIR після кліку по кнопці «View»

ein schriftlich festgelegtes Verfahren? *

64 Gibt es eine Liste der Notfallmedikamente, die in der Praxis verfügbar sein sollten? * nein ja

65 Gibt es ein Verfahren in der Praxis zum Auffüllen der Notfallmedikamente? * nein ja nicht zutreffend

66 Gibt es ein Verfahren zur Kontrolle der Haltbarkeitsdaten der Notfallmedikamente? * nein ja nicht zutreffend

67 Regelt dieses Verfahren, wer für a) das Auffüllen der Notfallmedikamente und b) die Überprüfung der Haltbarkeitsdaten der Notfallmedikamente verantwortlich ist? * nein ja nicht zutreffend

68 Ist in der Praxis eine Arzttasche oder ein Notfalkoffer für dringende Notfälle verfügbar? * nein ja

69 Befindet sich in der Praxis ein funktionstüchtiger Kühlschrank zur Aufbewahrung von Medikamenten? * nein ja

70 Bewahrt die Praxis Medikamente und Rezepte, die unter das Betäubungsmittelgesetz fallen in einem abgeschlossenen Schrank auf? * nein ja nicht zutreffend

[← Back to list](#) [→ To top](#)

© 2022 V-Health
Certificates ISO xxxxxxxx and ISO/IEC xxxxxxxx

Welcome

Рисунок 3.27 – Нижня частина фікна на сторінці «Survey» із згенерованим формуляром FHIR та функціональними кнопками «Validate» та «Save»

aktualisiert? *

37 Wenn die Antiviren-Software manuell aktualisiert wird, erfolgt dies täglich? * nein ja nicht zutreffend

Input error

Haben Sie einen Ansagetext auf dem Anruferantworter gehört, der über eine Kontaktaufnahme mit der Praxis/ mit dem Notdienst außerhalb der Sprechzeiten informiert? Alternativ: Wurde Ihr Anruf direkt an den Arzt weitergeleitet? - input required

Ist eine Praxiswebseite im Internet zu finden? - input required

Befindet sich am Haus/Hauseingang ein Praxisschild, welches Auskunft über die werktäglichen Öffnungszeiten der Praxis gibt? - input required

Befinden sich am Haus/Hauseingang Informationen welche Auskunft geben, wie die Versorgung außerhalb der Sprechzeiten geregelt ist? (z.B. 112, 116117) - input required

Gibt es in der Nähe der Praxis Parkplätze, die für behinderte Menschen reserviert sind? - input required

Gibt es am Haupteingang der Praxis eine Rampe bzw. keine Stufen? - input required

Gibt es die Möglichkeit, dass Menschen mit Mobilitätseinschränkungen im Erdgeschoss behandelt werden oder steht alternativ ein Lift zur Verfügung? - input required

Sind alle relevanten Räume der Praxis für Menschen mit Mobilitätseinschränkungen (z.B. mit einem Gehwagen oder Rollstuhl) gut erreichbar? - input required

Besitzt die Praxis einen sicheren und trockenen Platz für Kinderwagen, Rollstuhl und/oder Gehwagen? - input required

Ist die Patiententoilette, ggf. auch eine separate Toilette in der Praxis, barrierefrei gestaltet? - input required

Verfügt die Praxis über eine Möglichkeit zum Wickeln mit Schutz gegen Herunterfallen? - input required

Ist die Praxis zur Vermeidung von Stürzen hinreichend ausgeleuchtet? - input required

Werden potenzielle Stolperfallen in der Praxis identifiziert und wenn möglich beseitigt? - input required

Werden Patienten auf potenzielle Stolperfallen hingewiesen? - input required

Werden Patienten auf eine bestehende Rutschgefahr in den Praxisräumen (z.B. auf feuchtem Boden) hingewiesen? - input required

Wird gehbehinderten Patienten geholfen, indem Sie beispielsweise beim Gehen innerhalb der Praxisräumen unterstützt werden? - input required

Hängt in den Praxisräumen ein Fluchtplan deutlich sichtbar aus? - input required

Gibt es weitere Maßnahmen, die sicherstellen, dass im Ernstfall Notausgänge eindeutig zu erkennen sind (z.B. durch selbstleuchtende oder langleuchtende Piktogramme, welche den Fluchtweg deutlich und leicht verständlich markieren)? - input required

Sind die Feuerlöscher an einem sichtbaren, gut gekennzeichneten und leicht zugänglichen Ort vorschriftsmäßig angebracht? - input required

Рисунок 3.28 – Приклад повідомлення валідації

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено, оптимізовано та інтегровано до існуючої МІС алгоритм генерації, серіалізації та десеріалізації за стандартом HL7/FHIR.

У ході кваліфікаційної роботи було здійснено дослідження стандарту HL7/FHIR, його структури, правил, робочих процесів тощо.

Для розв'язання проблем була спроектована модель алгоритму генерації та серіалізації даних за стандартом HL7/FHIR.

Для реалізації методу було розроблено поетапну модель розробки та проектування кожного модуля, компоненти та підмодуля системи, опис вимог. В якості програмного компонента було використано існуючу МІС із окремо розробленим модулем інтеграції.

Під час реалізації було використано новітні технології.

Було розроблено інтерфейс користування додатком для генерації, редагування та валідації формулярів для усіх доступних ролей користувачів.

У рамках розробки системи було створено та адаптовано:

- нові таблиці та зв'язки у базі даних;
- функціональні модулі що забезпечують роботу алгоритму генерації та серіалізації даних за стандартом HL7/FHIR;
- алгоритм генерації та серіалізації даних за стандартом HL7/FHIR.

Також було розширено механізм інформаційного забезпечення користувачів.

Результати дослідження апробовано у вигляді тез доповіді під час I інтернаціональної наукової та практичної конференції «CURRENT TRENDS IN THE DEVELOPMENT OF MODERN SCIENTIFIC THOUGHT» [53].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kahn, S., & Sheshadri, V. (2008). Medical record privacy and security in a digital environment. *IT professional*, 10(2), 46-52.
2. Blumenthal D. The Federal Role in Promoting Health Information Technology. The Commonwealth Fund, January, 2009. URL: <http://www.commonwealthfund.org/publications/perspectives-on-health-reform-briefs/2009/jan/the-federal-role-in-promoting-health-information-technology> (дата звернення 10.11.2022).
3. Bitton, A., Flier, L. A., & Jha, A. K. (2012). Health information technology in the era of care delivery reform: to what end?. *JAMA*, 307(24), 2593-2594.
4. Friedberg, M., Crosson, F. J., & Tutty, M. (2014). Physicians' concerns about electronic health records: implications and steps towards solutions. *Health Affairs blog*, 11, 963.
5. Jacob, J. A. (2015). GAO issues report on health information interoperability. *JAMA*, 314(18), 1906-1906.
6. Sharfstein, J. M. (2015). Using health care data to track and improve public health. *Jama*, 313(20), 2012-2013.
7. Herland, M., Khoshgoftaar, T. M., & Wald, R. (2014). A review of data mining using big data in health informatics. *Journal of Big data*, 1(1), 1-35.
8. PACS система и протокол DICOM. URL: <http://www.medialog.ru> (дата звернення 10.11.2022).
9. Стандарт openEHR. URL: <http://www.openehr.org> (дата звернення 9.11.2022)
10. Гайдуків, А. І., Балюк, А. С., & Реймеров, С. Ю. (2012). Підхід к формуванню, храненню и обробці частично структурованих медичинських документів в форматі ISO/HL7 27932: 2009 (HL7 CDA R2) в

медицинских информационных системах. *Врач и информационные технологии*, (5), 6-15.

11. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

12. Работягов, А. В., Ляшенко, В. В., & Кобылин, О. А. (2016). Сегментация сложных изображений цитологических препаратов.

13. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.

14. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 5(57).

15. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

16. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

17. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.

18. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed

observations in Data Stream Mining tasks. *Information Technology and Management Science*, 19(1), 23-28.

19. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. *International Journal of Advanced Trends in Computer Science and Engineering*. 9(4). – 4701-4706.

20. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). *Advances in Spatio-Temporal Segmentation of Visual Data* (Vol. 876). Springer Nature.

21. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

22. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, 79(10).

23. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform.

24. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

25. Gorokhovatskiy, V. A., Kobylin, O. A., & Kulikov, Y. A. (2015). Application of Granulation of Feature Descriptions in Structural Image Recognition. *Telecommunications and Radio Engineering*, 74(6).

26. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications* (pp. 210-230). Springer, Cham.

27. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In *Eleventh International Conference on Machine Vision (ICMV 2018)* (Vol. 11041, pp. 176-183). SPIE.

28. Бодянский, Е. В., Винокурова, Е. А., Пелешко, Д. Д., Кобылин, И. О., & Кобылин, О. А. (2017). Нечёткая кластеризация временных рядов с неравномерными и асинхронными тактами квантования. *Системы обработки информации*, (5), 47-54.

29. Lyashenko, V., Matarneh, R., Kobylin, O., & Putyatin, Y. (2016). Contour detection and allocation for cytological images using Wavelet analysis methodology.

30. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform.

31. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, and Vyacheslav Lyashenko (2019) Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach, *International Journal of Emerging Trends in Engineering Research*, 7(11), pp. 460-465.

32. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.

33. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.

34. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19). p. 6777.

35. Tvoroshenko, I. S. (2004) Structure and functions of intelligent decision-making tools in complex systems. *Artificial Intelligence*, 4, 462-470.

36. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical

System HEALTH, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6), pp. 3221-3227.

37. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки. *Системи обробки інформації*, (2), 76-82.

38. Творошенко І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.

39. Творошенко, И. С. (2010). Анализ процессов принятия решений в интеллектуальных системах. *Системи обробки інформації*, (2), 248-253.

40. Творошенко, И. С., Дехтярь, А. П. (2005, June) Информационные технологии в задачах компьютерной диагностики с использованием интеллектуальных систем. In *Клиническая информатика и Телемедицина. Компьютерная Медицина–2005: материалы междунар. научн.-технич. конф., Харьков* (p. 138).

41. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

42. Definition - What does IntelliJ IDEA mean? (2020). URL: <https://www.conceptdraw.com/examples/idef3-diagram-software> (дата звернення 01.11.2022).

43. What is MongoDB? Introduction, Architecture, Features & Example. URL: <https://www.guru99.com/what-is-mongodb.html> (дата звернення 01.11.2022).

44. Брацький, В. О., & М'якшило, О. М. (2016). Дослідження особливостей застосування реляційних і нереляційних баз даних на прикладі SQL Server та MongoDB. *Наукові праці Національного університету харчових технологій*, (22, № 5), 15-24.

45. Ситников, Д. Э., Титов, С. В., & Титова, Е. В. (2011). Семантические свойства информативности ассоциативных зависимостей между признаками объектов в базах данных.

46. Титова, Е. В. (2003). Сравнительная характеристика простых и расширенных ассоциативных правил для признаков объектов в базах данных. Системы обробки інформації.–Х.: ХВУ, 2003.–Вип. 2, 31-37.

47. Sitnikov, D., Titova, O., Minukhin, S., Kovalenko, A., & Titov, S. (2018, October). Informativity of association rules from the viewpoint of information theory. In 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) (pp. 595-598). IEEE.

48. Луганский, А. М., Машталир, В. П., & Шляхов, В. В. (2004). Связь отношений толерантности и эквивалентности в метатекстовых информационноаналитических системах. Радиоэлектроника и информатика, (1 (26)).

49. Богучарский, С. И., & Машталир, В. П. (2014). Сегментация изображений в больших базах данных с использованием плотности распределения информации. *Electrotechnic and Computer Systems*, (14 (90)), 119-123.

50. Машталир, В. П., & Шляхов, В. В. (2006). Индуцированная согласованность отношений в задачах грануляции информации.

51. Луганский, А. М., & Машталир, В. П. (2005). Метатекстовые модели в интерактивных информационных системах. Системы обробки інформації, (2), 202-210.

52. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, and Vyacheslav Lyashenko (2019) Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach, *International Journal of Emerging Trends in Engineering Research*, 7(11), pp. 460-465.

53. Луцков, В. (2022). Аналіз стандарту HL7/FHIR.