

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ 6.050103 «Програмна інженерія»

на тему _____ WEB-застосування для проведення аналізу та оцінки

ефективності профілактичних заходів передачі ВІЛ від матері до дитини

Виконав: студент IV курсу, групи _____ ІІ-51 Реутська С.В.
(прізвище, ім'я, по батькові) _____ (підпис)

Керівник _____ Ліщук К.І., к.т.н. доцент
_____ (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

**Консультант
з графічної
документації** _____ доц., к.т.н., Ліщук К.І.
_____ (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

Рецензент: _____ доц., к.т.н., доц. Лісовиченко О.І.
_____ (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103
«Програмна інженерія» (Програмне забезпечення систем)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

О.А. Павлов
(підпис) (ініціали, прізвище)

“ ” _____ 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Реутській Світлані Віталіївні
(прізвище, ім'я, по батькові)

1. Тема проекту «WEB-застосування для проведення аналізу та оцінки ефективності прфілактичних заходів передачі ВІЛ від матері до дитини»

керівник проекту Ліщук Катерина Ігорівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

2. Термін подання студентом проекту «03» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: основні положення, опис предметної області, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог, постановка завдань

2) Моделювання, архітектура та конструювання програмного забезпечення: аналіз програмного забезпечення, вхідні та вихідні дані, засоби розробки, структура бази даних та класів

3) Аналіз якості та тестування програмного забезпечення

4) Розгортання та впровадження програмного забезпечення

5. Перелік графічного матеріалу

1) Схема структурна класів програмного забезпечення

2) Схема бази даних

3) Креслення вигляду екранних форм

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «12» березня 2018 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	25.03.2019	
2.	Аналіз існуючих методів розв'язання задачі	10.04.2019	
3.	Постановка та формалізація задачі	19.04.2019	
4.	Аналіз вимог до програмного забезпечення	26.04.2019	
5.	Алгоритмізація задачі	03.05.2019	
6.	Моделювання програмного забезпечення	10.05.2019	
7.	Обґрунтування використовуваних технічних засобів	12.05.2019	
8.	Розробка архітектури програмного забезпечення	17.05.2019	
9.	Розробка програмного забезпечення	17.05.2019	
10.	Налагодження програми	24.05.2019	
11.	Виконання графічних документів	24.05.2019	
12.	Оформлення пояснювальної записки	27.05.2019	
13.	Подання ДП на попередній захист	28.05.2019	
14.	Подання ДП рецензенту	06.06.2019	
15.	Подання ДП на основний захист	08.06.2019	

Студент _____ Реутська С.В.
(підпис)

Керівник проекту _____ Ліщук К.І.
(підпис)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 85 сторінок, 13 рисунків, 66 таблиць, 5 додатків, 11 джерел.

Метою розробки web-застосування є зменшення часу на пошук інформації про пацієнтів по всій Україні, ведення їх історій хвороби, а також надання аналітичних матеріалів у вигляді звітів для аналізу результатів, отриманих внаслідок проведення певних профілактичних заходів.

У розділі «Аналіз вимог» проаналізовано популярні рішення перенесення медичних форм в електронний формат. Було проведено аналіз програм, схожих за своєю функціональністю, з метою визначення основних пунктів, які має реалізовувати подібне програмне забезпечення. Також визначено перелік функціональних та нефункціональних вимог та отримані чітко сформульовані цілі, яких необхідно досягнути в процесі розробки ПЗ.

У розділі «Моделювання та конструювання» на основі медичних паперових форм було визначено формат вхідних та вихідних даних. В процесі розробки архітектури ПЗ, було створено модель бази даних та спроектовано схему класів, їх основних методів для досягнення поставлених цілей. Крім того були продумана стратегія забезпечення безпеки персональних даних користувачів системи та пацієнтів.

У розділі «Аналіз якості та тестування» створене WEB-застосування було проаналізовано за різними критеріями якості ПЗ. За допомогою критеріїв оцінювання та проведення функціональних та нефункціональних тестів було доведено, що даний програмний продукт високої якості.

У розділі «Впровадження та супровід» була побудована схема розгортання ПЗ та наведена покрокова інструкція щодо розгортання WEB-застосування на сервери ІІS.

КЛЮЧОВІ СЛОВА: WEB-ЗАСТОСУВАННЯ, ЕЛЕКТРОННА МЕДИЧНА КАРТКА, ШАБЛони ЗВІТНОСТІ, ASP.NET MVC, C#, MS SQL

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ABSTRACT

Structure and scope of work. Explanatory note of the graduation project consists of five sections, contains 85 pages 13 drawings 66 tables 5 Applications 3 sources.

The purpose of developing a web application is to reduce the time it takes to search for information about patients throughout Ukraine, to keep their case histories, and to provide analytical materials in the form of reports to analyze the results obtained as a result of certain preventive measures.

In the “Requirements Analysis” section, popular solutions for transferring medical forms to electronic format are analyzed. The analysis was carried out of programs with similar functionality in order to determine the main points that such software should implement. It also defines a list of functional and non-functional requirements and obtained clearly formulated goals that need to be achieved in the software development process.

In the “Modeling and Design” section, the format of the input and output data was determined on the basis of medical paper forms. In the process of developing software architecture, a database model was created and a diagram of classes was designed, their main methods for achieving the goals set. In addition, there was a well-thought-out strategy to ensure the security of personal data of patients and system's users.

In the section "Analysis of quality and testing" created WEB-application was analyzed according to various criteria of software quality. Using evaluation criteria and conducting functional and non-functional tests, it was proved that this software product is of high quality.

In the “Implementation and Maintenance” section, a software deployment scheme was built and step-by-step instructions were given on how to deploy a WEB application to IIS servers.

KEYWORDS: WEB-applications, electronic medical records, reporting templates, ASP.NET MVC, C #, MS SQL

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Пояснювальна записка до дипломного проекту

на тему: WEB-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини

Київ – 2019 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	13
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	14
1.2.2 ЗВІТНІ ФОРМИ, НЕОБХІДНІ ДЛЯ ДОСЛІДЖЕННЯ ХВОРОБИ [2]	19
1.2.3 <i>Формалізований опис процесів роботи з системою</i>	20
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ	23
1.3.1 <i>Аналіз відомих технічних рішень</i>	23
1.3.2 <i>Аналіз відомих програмних продуктів</i>	24
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
1.4.1 <i>Розроблення функціональних вимог</i>	27
1.4.2 <i>Розроблення нефункціональних вимог</i>	34
1.4.3 <i>Постановка комплексу завдань модулю</i>	34
1.5 ВИСНОВКИ ПО РОЗДІЛУ	35
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	36
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	36
2.1.1 <i>Вхідні та вихідні дані</i>	36
2.1.2 <i>Моделювання бізнес-процесів</i>	43
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	46
2.2.1 <i>Структура бази даних</i>	47
2.2.2 <i>Структура класів</i>	54
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	61
2.3.1 <i>Представлення</i>	61
2.3.2 <i>Контролери</i>	64
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ	65
2.5 ВИСНОВКИ ПО РОЗДІЛУ	66
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	68
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	68

3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....72

3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ74

3.4 ВИСНОВКИ ДО РОЗДІЛУ77

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ78

4.1 РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....78

4.2 РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....79

4.3 ВИСНОВКИ ДО РОЗДІЛУ82

ВИСНОВКИ83

ПЕРЕЛІК ПОСИЛАНЬ.....85

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

АРВ – антиретровірус

АРТ – антиретровірусна терапія

ВІЛ – вірус імунодефіциту людини

ЄДРПОУ – код юридичної особи з єдиного державного реєстру

ЗОЗ – заклад охорони здоров`я

ІФА – ферментний імуносорбентний аналіз

МІС – медичні інформаційні системи

ООН – Організація Об'єднаних Націй

ОШС – організаційно-штатна структура закладів охорони здоров`я

ПЗ – програмне забезпечення

ППМД – профілактика передачі ВІЛ від матері до дитини.

ІІS – (Internet Information Services) – набір серверів для служб інтернету від компанії Microsoft

MVC – (Model View Controller) – архітектурний шаблон для проектування та розробки програмного забезпечення

MS – компанія Microsoft

MS SQL Server – (Microsoft SQL Server) – система керування базами даних з мовою запитів Transact-SQL

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

ВСТУП

Актуальність теми. В сучасному світі інформаційні технології займають дуже важливе місце в житті кожної людини. Не можливо уявити сферу діяльності, яка наразі існувала б без допомоги програмного забезпечення. Його використовують для зберігання та швидкого пошуку інформації, ведення складних обрахунків, аналізу отриманих даних та їх графічного представлення, тощо. Це дає можливість значно зекономити робочий час, зменшити вірогідність помилок через неуважність, автоматизувати процеси, де втручання людини не є необхідним та полегшити роботу, пов'язану з дослідженнями.

Саме тому в Україні дуже гостро стоїть питання про автоматизацію роботи закладів охорони здоров'я. Вирішення цієї задачі стане важливим кроком до становлення України як високорозвиненої держави з гідним рівнем соціального захисту населення.

Зараз існує безліч установ різного типу та призначення, ніяк не пов'язаних один з одним. Більшість інформації зберігається на паперових бланках, хаотично розкиданих по сховищах. Важко за короткий строк знайти потрібний документ, впевнитись в його актуальності. Ще важче збирати статистичні дані, групувати їх за певними ознаками та робити припущення. Часто лікарі та пацієнти стикаються з проблемами втрат цінних паперів, дублювання даних, помилками в записах, недоступністю інформації з іншого закладу для отримання повної картини хвороби.

За останні декілька років було створено декілька систем, метою яких було вирішення більшості проблем, описаних вище. Але вони не є загальнодержавними, тому питання про доступ даних з будь-якої точки України лишається невирішеним.

ООН вважає пошук засобів проти розповсюдження ВІЛ одним з першочергових завдань. Українські вчені також приймають участь у розробці препаратів, створюючи нові варіанти, а потім досліджуючи ефективність їх впливу. Оскільки збереження такого об'єму даних та їх оцінка майже

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

неможливі в умовах паперових карток пацієнтів та існуванні окремих систем в кожному закладі, постає гостра необхідність у створенні програмного продукту, який дозволить мати єдину базу історій хвороби, впливу ліків на їх розвиток та автоматизує доступ до інформації вповноваженим особам.

Мета створення даної роботи – створення загального електронного реєстру медичних карток з обмеженим доступом, автоматизація пошуку інформації та створення звітності на її основі за допомогою використання інтегрованої інформаційної системи.

Для досягнення мети у роботі вирішуються наступні задачі: створення, редагування та видалення електронних версій медичних форм пацієнтів, прив'язка пацієнту до регіону та/або медичної установи, створення профілів користувачів системи (лікарів) з доступом лише до пацієнтів їх установи, перенесення інформації пацієнтів до іншої установи при необхідності, ведення архіву недійсних карток, формування звітів з можливістю вказання параметрів вибірки.

Завданням даної роботи є розробка системи для збору інформації про ВІЛ-інфікованих пацієнток по всій Україні, ведення їх історій хвороби, а також аналіз результатів, отриманих внаслідок проведення певних профілактичних заходів.

Практичне значення. Результатом роботи є інтегроване WEB-застосування, яке можна використовувати в медичних установах України, спеціалізацією якої є хвороба ВІЛ.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

МІС поширені в різних сферах діяльності, в кожній з яких виконують окремі функції. Virізняють декілька найпоширеніших класифікацій [1].

За ступенем автоматизації процесів:

- автоматизовані – збір та первинна обробка інформації виконуються людиною, система виступає лише сховищем даних;
- автоматичні – можливе повне виключення людини з процесу.

За типом інформаційної бази:

- системи даних – базуються на фактах;
- експертні системи – базуються на знаннях фахівців.

За типом інформаційної бази:

- пошукові – здійснюють автоматизацію пошуку фактичних даних та зібраної інформації пацієнтів;
- логічні – системи, що обробляють відомі дані та прогнозують певні результати на їх основі;
- керуючі – системи, що реалізують повний процес роботи з інформацією на різних рівнях (збір, обробка, передача, формування рішення).

За ієрархією:

- базовий – на рівні окремих лікарів;
- лікувально-профілактичного закладу – на рівні поліклінік, диспансерів, стаціонарів;
- територіальний – на рівні органів керування в регіонах, спеціалізованих медичних служб;
- державний – на рівні загальнодержавних закладів та органів управління.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Все більше виникає потреба в створенні МІС на територіальних та державних рівнях, оскільки це дасть можливість створювати не лише системи даних та пошукові, а й впроваджувати керування процесом на всіх рівнях ієрархії, робити прогнози з достатньою вибіркою даних. Дуже важливо мінімізувати неточності, спричинені людським фактором, уникнути дублювання та втрати актуальності інформації. Для діяльності, яка спрямована на аналіз симптомів людей, хворих на невиліковний на даний момент вірус, така система є необхідною складовою, за допомогою якої ефективність боротьби буде збільшена в рази.

1.2 Змістовний опис і аналіз предметної області

Головним завданням даної роботи є перенесення роботи закладів охорони здоров'я в електронний формат. Для цього необхідно надати користувачам (лікарям) звичну для них форму заповнення інформації для пацієнтів.

Було проаналізовано роботу медичних центрів, отримано змістовні описи медичних бланків пацієнтів. Також було визначено, які саме результати спеціалісти хочуть побачити від системи.

1.2.1 Види медичних форм [2]

Таблиця 1.1 – Форма №501-1 Про ВІЛ-інфіковану вагітну

Поле	Тип
Номер	Число
Дата заповнення	Дата
ПІБ	Текст
Дата народження	Дата
Громадянство	Текст
Адреса	Текст
Взяття під нагляд	Дата
Термін вагітності	Число
Освіта	Текст

Продовження таблиці 1.1

Сімейний стан	Текст
Кількість пологів	Число
Кількість абортів	Число
Кількість плодів	Число
Код на обстеження ВІЛ	Текст
Метод підтвердження діагнозу	Текст
Дата встановлення ВІЛ	Дата
Кількість CD4	Число
ЗОЗ в якому перебуває на обліку	Текст
Вірусне навантаження	Число
Стадія ВІЛ	Текст
Ступінь імуносупресії ВІЛ	Текст
Інформація про туберкульоз	Дата, текст
Вживання наркотичних речовин	Текст
Учасниця програми підтримувальної терапії	Так/Ні
Кількість статевих партнерів	Число
Використання методів контрацепції у період вагітності	Текст
Надання згоди про участь у програмі ППМД	Так/Ні
ПІБ лікаря	Текст
Телефон лікаря	Текст

Таблиця 1.2 – Форма №501-2 Про закінчення вагітності

Поле	Тип
Номер	Число
Дата заповнення	Дата

Продовження таблиці 1.2

ПІБ	Текст
Дата народження	Дата
Громадянство	Текст
Адреса	Текст
Загроза переривання	Так/Ні, текст
Затримка розвитку плода	Текст
Наявність захворювань	Текст
Завершення вагітності	Дата, текст
Виявлення ВІЛ статусу	Дата, текст
АРВ-профілактика	Текст
Тривалість курсу	Текст
АРТ до вагітності	Число
АРТ під час вагітності	Число
Прихильність до АРВ/АРТ	Текст
Кількість CD4 перед пологами	Дата, число
Рівень вірусного навантаження	Дата, число
Період доставлення в пологовий будинок	Текст
Спосіб пологів	Текст
Тривалість пологів	Текст
Кількість дітей	Число
Жива	Так/Ні
Стать	Текст
Зріст	Число
Вага	Число
Оцінка АПGAR	Число
Зразок крові на обстеження	Дата, число

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.2

Результат обстеження ВІЛ	Текст
АРВ-профілактика дитини	Дата, текст
Вигодовування	Текст
Смерть	Дата, текст
Заключний діагноз	Текст
ПІБ лікаря	Текст
Телефон лікаря	Текст

Таблиця 1.3 – Форма №501-3(1) Про перебування на обліку дитини

Поле	Тип
Номер	Число
Дата заповнення	Дата
ПІБ	Текст
Дата народження	Дата
Стать	Текст
Адреса, телефон	Текст
ПІБ матері	Текст
Джерело фінансування молочних сумішей	Текст
Ускладнення народження	Текст
Причина взяття на облік	Текст
ПІБ лікаря	Текст
Телефон лікаря	Текст

Таблиця 1.4 – Форма №501-3(2) Про результати діагностики та профілактику ВІЛ

Поле	Тип
Номер	Число
Дата заповнення	Дата

Продовження таблиці 1.4

ПІБ	Текст
Дата народження	Дата
Стать	Текст
Адреса, телефон	Текст
ПІБ матері	Текст
Дослідження ПЛР ДНК	Дата, текст
Рівень вірусного навантаження	Дата, число
ВІЛ-статус	Текст
Профілактика пневмоцистної пневмонії	Текст
ПІБ лікаря	Текст
Телефон лікаря	Текст

Таблиця 1.5 – Форма №501-3(3) Про уточнення ВІЛ-статусу

Поле	Тип
Номер	Число
Дата заповнення	Дата
ПІБ	Текст
Дата народження	Дата
Стать	Текст
Адреса, телефон	Текст
ПІБ матері	Текст
Вигодовування	Текст
ІФА, ІБ	Дата, число, текст
Дата остаточного діагнозу	Дата
Період інфікування	Число
Кількість CD4	Дата, число
Рівень вірусного навантаження	Дата, число

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.5

Схема АРТ	Дата, текст
Зняття з обліку	Дата, текст
Дата смерті	Дата
Заключний діагноз	Текст
Діагноз патологоанатома	Текст
Причина смерті	Текст

Усі дані, які лікарі вписували руками, але при цьому передбачали вибір однієї з шаблонних відповідей, вирішено занести у списки-словники для пришвидшення заповнення картки, а також полегшення утворення звітності. Також було вирішено додати інформацію про переміщення пацієнтів з одного ЗОЗ в інший, створити окремі інформаційні картки про ці заклади для зручності заповнення даних про лікаря та розподілу прав доступу до карток пацієнток та дітей.

1.2.2 Звітні форми, необхідні для дослідження хвороби [2]

Інформація про ВІЛ-інфікованих пацієнтів:

- Форма 1000 – облік офіційно зареєстрованих ВІЛ-інфікованих вагітних у ЗОЗ, за якими ведеться диспансерне спостереження (інформація з форми 502-1);
- Форма 1001 – консультація і проведення тестів ВІЛ-позитивного статусу за звітний рік (вагітні, що вперше провели дослідження);
- Форма 1002 – встановлення діагнозу ВІЛ у вагітних залежно від терміну вагітності у ЗОЗ (нові випадки вагітності);
- Форма 1003 – перебіг вагітності (пацієнтки, вагітність яких була завершена в звітному році);
- Форма 1004 – Стадії ВІЛ-інфекції пацієнтів на момент першого обстеження (інформація з форми 502-1).

Інформація про ВІЛ-інфікованих жінок, що народили:

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- Форма 2000 – завершення вагітності (пологами або абортами);
- Форма 2001 – перебіг пологів (незалежно від перебування в ЗОЗ).

Інформація про АРВ-профілактику:

- Форма 3000 – профілактика під час вагітності і пологів;
- Форма 3001 – профілактика новонароджених (незалежно від їх перебування під наглядом).

Інформація про АРВ-профілактику передачі ВІЛ до дитини:

- Форма 4000 – облік дітей, народжених ВІЛ-інфікованими;
- Форма 4001 – вид вигодовування дітей;
- Форма 4002 – діагностика ВІЛ у дітей та профілактика захворювання;
- Форма 4003 – діти, що перебувають на обліку в ЗОЗ (до 18 років,

інформація з картки 502-1);

- Форма 4004 – соціальний стан ВІЛ-інфікованих дітей (під наглядом ЗОЗ).

Частота зараження ВІЛ дитини матір`ю:

- Форма 5000 – відомості про дітей за зазначений рік;
- Форма 5001 – діти з остаточно встановленим діагнозом;
- Форма 5002 – діти з відсутністю ВІЛ;
- Форма 5003 – діти, які знаходяться на етапі проведення досліджень.

Проведення профілактики передачі ВІЛ:

- Форма 6000 – відомості про дітей за зазначений рік;
- Форма 5001 – профілактичні дії щодо попередження інфекції;
- Форма 5002 – частота передачі ВІЛ при проведенні профілактичних

дій.

1.2.3 Формалізований опис процесів роботи з системою

Процес роботи з системою можна умовно розділити на три узагальнених:

- заповнення та редагування медичної форми;
- формування звіту;

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

– перенесення медичних форм з одного регіону в інший або в архів.

На схемі, представлений на рисунку 1.1 відображено процес заповнення медичних форм. В медичних закладах він відбувається вручну, з заповненням безлічі паперових анкет та копій. Після заповнення анкети потрапляють до архіву медичного закладу, де зберігаються до того моменту, поки не знадобляться лікареві або людині, що відповідає за звітність.

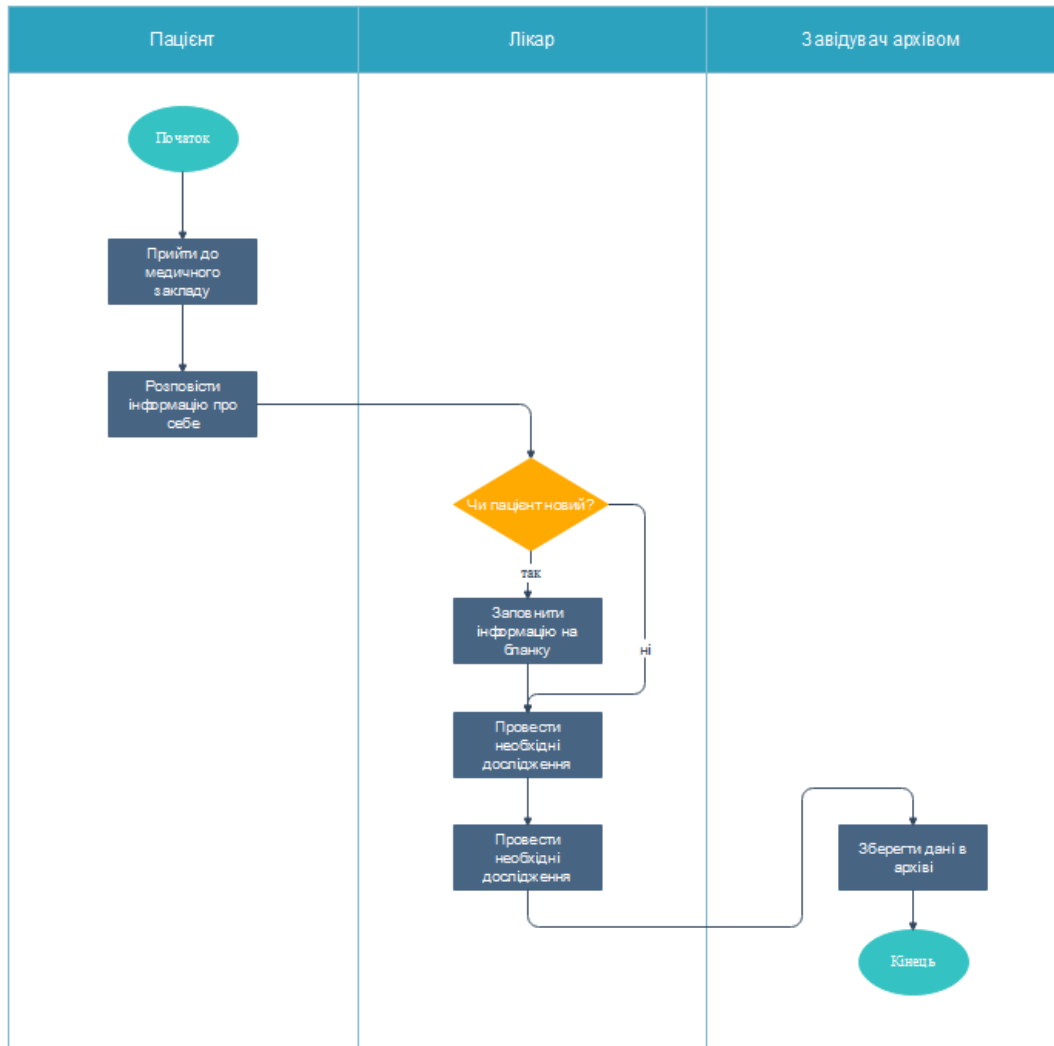


Рисунок 1.1 – Схема структурна активності учасників процесу роботи з медичними формами

На схемі, представлений на рисунку 1.2 відображено процес створення звіту. Людина, що має здати звітність, відвідує архів карток або збирає інформацію від людей з різних центрів охорони здоров'я. Далі вона передивляється всю інформацію та робить висновки про досліджені дані.

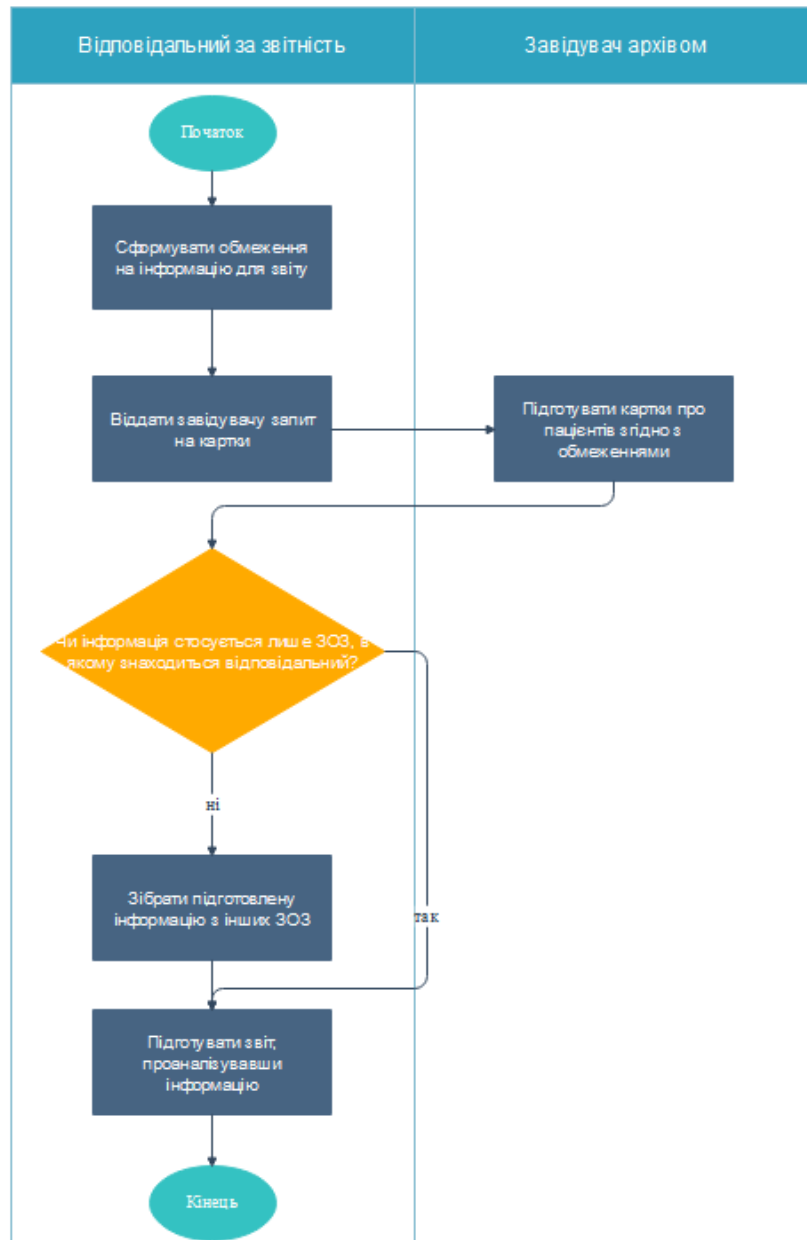


Рисунок 1.2 – Схема структурна активності учасників процесу роботи зі звітністю

На схемі, представлений на рисунку 1.3 відображено процес зміни місця обліку пацієнта. Для отримання даних з попереднього ЗОЗ, лікарю необхідно звернутись до завідуючого відділом минулого ЗОЗ, а тому, в свою чергу, до завідуючого архівом. Цей процес може займати купу часу, оскільки потребує залучення людей сусіднього закладу та часу на пошуки картки в архіві.

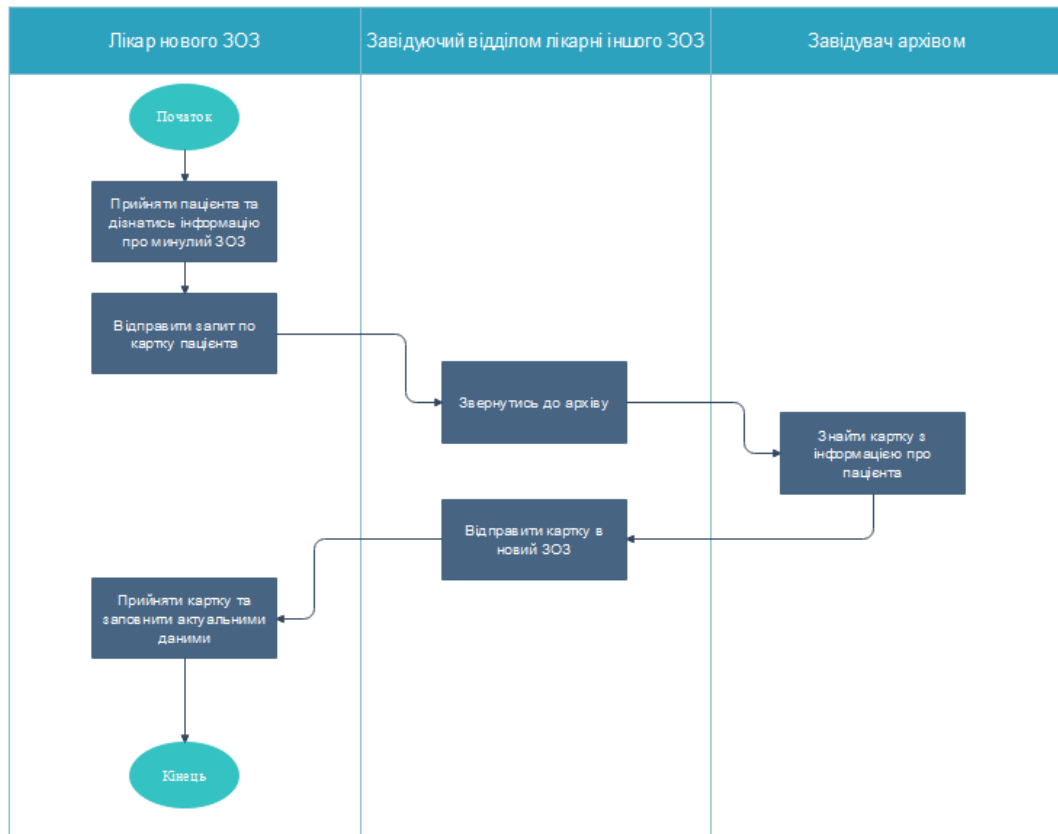


Рисунок 1.3 – Схема структурна активності учасників процесу перенесення картки

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

Все більшого поширення набувають електронні медичні карти. Це сукупність електронних персональних медичних записів пацієнта, що зберігаються на певному носії. Такі карти призначені для того, щоб вирішувати ряд важливих задач:

- збереження в одному місці максимального об`єму інформації;
- забезпечення швидкого доступ до неї пацієнтам та/або мед-працівникам;
- автоматизована побудова статичної та фінансової звітності;
- управління організацією або регіональним ЗОО;
- контроль якості лікування та роботи персоналу;

- проведення дослідів на базі аналізу отриманих даних;
- використання анонімних даних для навчання студентів.

Всі карти зберігаються в електронному медичному архіві – сховищі даних, де зібрана інформація в межах однієї організації. Також в ньому зберігаються списки пацієнтів, довідники, інструменти пошуку та навігації. Сукупність архівів, електронної документації та програмного забезпечення, що здійснює збір та обробку електронних карт, утворюють Єдину систему ведення електронних медичних карт.

Постає питання про стандартизацію інформації для електронного обміну документами, а також для проведення автоматичного аналізу. Паперові медичні форми здебільшого заповнюються у вільному форматі. Для більшості полів електронна система передбачає набір шаблонів. Користувачу лишається лише заповнити дати та обрати необхідну відповідь серед зазначених. Для додаткової інформації передбачені поля-коментарі.

Існує великий ризик, що зловмисники спробують увійти в систему та отримати персональні конфіденційні дані пацієнтів. Щоб запобігти цьому, персональні дані зберігаються на сервері у зашифрованому вигляді. Доступ до системи захищається різними способами:

- електронні ключі, видані кожному лікарю особисто;
- СМС-пароль згенерований перед кожною спробою увійти;
- реєстрація користувача виключно через адміністратора системи.

1.3.2 Аналіз відомих програмних продуктів

Було знайдено декілька медичних платформ, які уособлюють у собі систему ведення електронних карт та інструменти формування звітності.

Archimed+ [3]

Медична система для приватних стоматологічних клінік надає такі послуги:

- розрахунок заробітної плати;

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

- єдина база пацієнтів усіх філіалів;
- маркетингова та фінансова звітність за певний період часу;
- готові шаблони для заповнення карток;
- можливість додавання своїх шаблонних фраз, варіантів відповідей;
- повна інформація про електронну історію хворого.

Система орієнтована як на перенесення історії хвороби в електронний вигляд, так і на полегшення управління підприємством. Серед плюсів можна виділити можливість швидкого розширення варіантів заповнення поля картки. Серед мінусів – єдина база ніяк не поділена на філії, орієнтована на малі підприємства, інформація доступна будь-якому користувачу в повному обсязі. Існує лише звітність для полегшення ведення бізнесу, але не для аналізу лікування хвороби. Неможливе додавання своїх варіантів звітності.

Emcimed [4]

Система обігу медичних документів пропонує створення медичної картки пацієнта, автоматичне створення звітності та контроль правильного заповнення медичних карт.

Серед її основних функцій:

- ведення медичних карт пацієнтів, їх автоматична нумерація;
- управління поточним статусом медичної картки (активна, недійсна, заблокована);
- управління дозволами на перегляд та редагування картки;
- можливість додати файли, де можуть зберігатись скани паперових носіїв, до картки.

Серед плюсів можна виділити управління статусом картки, гнучку систему дозволів на доступ до інформації, автоматичну нумерацію, що дозволить однозначно ідентифікувати картку без викриття персональних даних. Серед мінусів – звітність налаштована автоматично, ніяких параметрів не передбачено.

Доктор Елекс [5]

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Програмне рішення, яке було призначене для оптимізації процесу лікування, поліпшуючи якість обслуговування.

Серед її основних функцій:

- введення даних за допомогою деревовидної структури варіантів;
- система пошуку інформації про пацієнта за різними полями;
- внесення даних в карту відбувається в рекомендованій послідовності, яка зменшує ризик утворення помилок;
- окремий лабораторний кабінет для заповнення результатів обстежень;
- програма виконана у формі WEB-застосування.

Серед плюсів можна виділити систему пошуку інформації по всіх полях картки, за винятком персональних даних, зменшення ризику утворення помилкових даних та WEB-застосування, що полегшує доступ до системи з будь-якого місця і не потребує великих ресурсів комп'ютера. Серед мінусів – відсутність утворення звітності, деревовидна структура варіантів може сповільнити заповнення картки.

Проаналізувавши програми-аналоги, можна відзначити декілька корисних функцій для системи. Важливо якісно налаштувати систему доступу до електронних карток, оскільки користувачі одного ЗОЗ не мають бачити даних про хворого з інших закладів. Система пошуку серед доступних для користувача карток значно пришвидшить роботу, особливо коли після запуску проекту пройде декілька місяців і кількість карток перевищить тисячу. WEB-застосування є найкращим рішенням реалізації програмного продукту, оскільки система є загальноукраїнською і варто надати можливість доступу до неї без додаткових зусиль. Також варто додати автоматичну нумерацію картки, яка б однозначно ідентифікувала її серед інших (не Id запису в базі, оскільки нумерація має бути доступною користувачу).

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

1.4 Аналіз вимог до програмного забезпечення

Вимоги програмного забезпечення напряму залежать від того, які ролі передбачені в системі та від їх сценаріїв роботи. Для роботи з медичною інформацією важливо виділити чотири головних ролі:

- адміністратор;
- лікар;
- працівник;
- неавторизований користувач.

1.4.1 Розроблення функціональних вимог

Загалом, система має виконувати декілька ключових функцій, які направлені на автоматизацію роботи з медичними картками.

Таблиця 1.6 – Вимоги системи в цілому

№	Вимога	Опис
1	Робота з картою ЗОЗ	Система має надати можливість створювати, змінювати, видаляти та переглядати інформацію про ЗОЗ.
2	Робота з картою матері	Система має надати можливість створювати, змінювати, видаляти та переглядати інформацію картки матері. Картка складається з двох розділів: форми 501-1 і 501-2.
3	Робота з картою дитини	Система має надати можливість створювати, змінювати, видаляти та переглядати інформацію картки дитини. Картка складається з двох розділів: форми 501-3(1), 501-3(2) і 501-3(3).

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.6

4	Пошук картки	Система має можливість пошуку карток за обраними полями, або за повним співпадінням ФІО.
5	Формування звіту	Система має надати можливість обирати зі списку потрібний шаблон звіту і формувати його з урахуванням вказаних параметрів.
6	Дерево ЗОЗ	Систему рівнів закладів охорони здоров'я слід представити у вигляді деревовидної структури. Верхні вершини означають більш високий рівень ієрархії.

Неавторизований користувач не має ніяких прав у системі. Єдине що йому доступне – головна сторінка застосунку, на якій він може авторизуватись.

Працівник – роль користувачів, які будуть переглядати картки та формувати звіти. Для працівників визначені окремі додаткові вимоги роботи з системою.

Таблиця 1.7 – Вимоги системи для працівника

№	Вимога	Опис
1	Робота з картою ЗОЗ	Система не дає можливість роботи з картками ЗОЗ.
2	Робота з картою матері	Система має надати можливість лише переглядати інформацію картки матері.
3	Робота з картою дитини	Система має надати можливість лише переглядати інформацію картки дитини.

Лікар – найбільш поширена роль серед усіх користувачів. Вона буде надана усім лікарям, які будуть заповнювати картки. Для лікарів визначені окремі додаткові вимоги роботи з системою.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Таблиця 1.8 – Вимоги системи для лікаря-адміністратора

№	Вимога	Опис
1	Прив'язка картки дитини до матері	В системі має бути можливість однозначної прив'язки картки дитини до картки матері для зручності їх пошуку. Лікар може створювати та знищувати такі зв'язки.
2	Архів карток	В кожному ЗОЗ існує сховище карток, куди потрапляють копії карток пацієнтів, які були зняті з обліку даного ЗОЗ. В картках міститься актуальна інформація на момент зняття. Лікар може перенести картку пацієнта в архів.

Адміністратор – ключова роль в системі. Він має доступ до всіх розділів системи. Для лікарів визначені окремі додаткові вимоги роботи з системою.

Таблиця 1.9 – Вимоги системи для адміністратора

№	Вимога	Опис
1	Робота з користувачами	Адміністратор має право на створення, редагування та видалення профілів користувачів. Також він може змінювати дозволи в системі, які закріплені за певною роллю.
2	Створення шаблонів звітів	Адміністратор має право на створення, редагування та видалення шаблону звіту, зміни його параметрів.

Відображення усіх можливих дій користувача з застосунком показано в структурній схемі варіантів використання на рисунку 1.4.

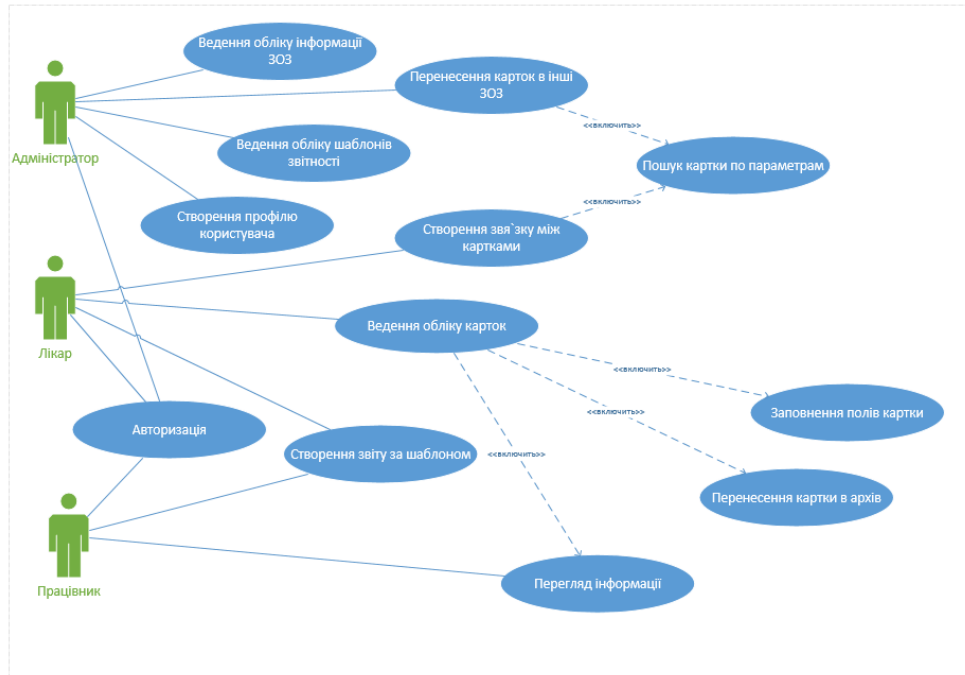


Рисунок 1.4 – Схема структурна варіантів використання

Опис варіантів використання наведено в таблицях 1.10 – 1.22.

Таблиця 1.10 – Варіант використання «Робота з картками пацієнтів»

Ідентифікатор	ВВ-1
Найменування	Робота з картками пацієнтів
Актор	Лікар, працівник
Опис	Можливість додати, відредагувати, переглянути, видалити картку матері або дитини (для працівника – лише переглянути).
Попередні умови	Відкрито сторінку з деревом ЗОЗ, для доступного актору ЗОЗ відображено список карток.
Вихідні умови	Відкрито сторінку додавання, редагування, перегляду картки пацієнта або повідомлення про видалення картки.

Таблиця 1.11 – Варіант використання «Створення картки»

Ідентифікатор	ВВ-2
Найменування	Створення картки

Продовження таблиці 1.11

Первинний актор	Лікар
Опис	Можливість створити нову медичну картку.
Попередні умови	Відкрито сторінку зі списком карток пацієнтів.
Вихідні умови	Створено картку пацієнта в головному сховищі ЗОЗ.

Таблиця 1.12 – Варіант використання «Редагування картки»

Ідентифікатор	ВВ-3
Найменування	Створення картки
Первинний актор	Лікар
Опис	Можливість відредагувати медичну картку.
Попередні умови	Відкрито сторінку зі списком карток пацієнтів.
Вихідні умови	Оновлено картку пацієнта.

Таблиця 1.13 – Варіант використання «Перегляд інформації картки»

Ідентифікатор	ВВ-4
Найменування	Перегляд інформації картки
Первинний актор	Лікар, працівник
Опис	Можливість переглянути інформацію про пацієнта.
Попередні умови	Відкрито сторінку зі списком карток пацієнтів.
Вихідні умови	Відкрита сторінка з відображенням усіх даних про пацієнта

Таблиця 1.14 – Варіант використання «Видалення картки»

Ідентифікатор	ВВ-5
Найменування	Створення картки
Первинний актор	Лікар
Опис	Можливість видалити медичну картку.
Попередні умови	Відкрито сторінку зі списком карток пацієнтів.
Вихідні умови	Видалено картку пацієнта.

Таблиця 1.15 – Варіант використання «Авторизація користувача»

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Ідентифікатор	ВВ-6
Найменування	Авторизація користувача
Первинний актор	Неавторизований користувач
Опис	Здійснення авторизації в системі, щоб отримати доступ до медичних карток.
Попередні умови	Відкрито головну сторінку з формою введення логіну та паролю.
Вихідні умови	Відкрито сторінку зі списком карток пацієнтів.

Таблиця 1.16 – Варіант використання «Створення користувача»

Ідентифікатор	ВВ-7
Найменування	Створення користувача
Первинний актор	Адміністратор
Опис	Створення профілю користувача в системі з призначенням йому необхідної ролі.
Попередні умови	Відкрито сторінку зі списком користувачів системи.
Вихідні умови	Створено нового користувача.

Таблиця 1.17 – Варіант використання «Видалення користувача»

Ідентифікатор	ВВ-8
Найменування	Видалення користувача
Первинний актор	Адміністратор
Опис	Видалення профілю користувача в системі.
Попередні умови	Відкрито сторінку зі списком користувачів системи.
Вихідні умови	Видалено користувача.

Таблиця 1.18 – Варіант використання «Створення нового шаблону звітності»

Ідентифікатор	ВВ-9
Найменування	Створення нового шаблону звітності
Первинний актор	Адміністратор

Продовження таблиці 1.18

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Опис	Створення шаблону звітності з параметрами, яким в подальшому зможуть користуватись лікарі.
Попередні умови	Відкрито сторінку зі списком шаблонів звітів.
Вихідні умови	Створено новий шаблон.

Таблиця 1.19 – Варіант використання «Редагування шаблону звітності»

Ідентифікатор	ВВ-10
Найменування	Редагування шаблону звітності
Первинний актор	Адміністратор
Опис	Редагування шаблону звітності, зміна його параметрів.
Попередні умови	Відкрито сторінку зі списком шаблонів звітів.
Вихідні умови	Відредаговано шаблон.

Таблиця 1.20 – Варіант використання «Перенесення картки в інший ЗОЗ»

Ідентифікатор	ВВ-11
Найменування	Перенесення картки в інший ЗОЗ
Первинний актор	Адміністратор
Опис	Картка стає недоступною для попереднього ЗОЗ і доступною для нового.
Попередні умови	Відкрито сторінку з історією переміщення карток.
Вихідні умови	Картку переміщено в головне сховище нового ЗОЗ. Додано запис про переміщення в таблицю історії.

Таблиця 1.21 – Варіант використання «Перенесення картки в архів ЗОЗ»

Ідентифікатор	ВВ-12
Найменування	Перенесення картки в архів ЗОЗ
Первинний актор	Лікар
Опис	Картка стає недоступною для редагування, неактуальною.
Попередні умови	Відкрито сторінку редагування картки.
Вихідні умови	Картку переміщено в додаткове сховище ЗОЗ з назвою «Архів».

Таблиця 1.22 – Варіант використання «Створення звіту»

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Ідентифікатор	ВВ-12
Найменування	Створення звіту
Первинний актор	Лікар, працівник
Опис	Створення звіту на основі існуючого шаблону.
Попередні умови	Відкрито сторінку списку доступних шаблонів.
Вихідні умови	Створено звіт по шаблону з урахуванням параметрів. Файл звіту автоматично зберігається на пристрій користувача.

1.4.2 Розроблення нефункціональних вимог

WEB-застосування має відповідати таким нефункціональним вимогам:

- персональні дані пацієнтів мають зберігатися в базі у зашифрованому вигляді;
- комп'ютер користувача має бути підключений до інтернету;
- інтерфейс українською мовою;
- інтерфейс має бути зрозумілим (з текстовими підказками);
- програма має коректно обробляти всі аварійні ситуації, викликані помилковими діями користувачів;
- доступ до будь-якої сторінки додатку має відбуватись не більш ніж в 5 дій.

1.4.3 Постановка комплексу завдань модулю

WEB-застосування ведення обліку карток пацієнтів призначене для швидкого внесення та редагування даних, налаштування обмеження доступу до інформації та автоматичної генерації звітності для аналізу проведення профілактичних заходів з метою боротьби з ВІЛ.

Ціль розробки програмного забезпечення – дати можливість працівникам закладів охорони здоров'я пришвидшити роботу, пов'язану з заповненням даних про пацієнтів та їх лікування, забезпечити конфіденційність персональної

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

інформації пацієнтів, зменшити вірогідність помилок, надати зручний інструмент для аналізу ефективності різних профілактичних заходів.

Щоб досягти поставленої цілі, в системі необхідно реалізувати наступні функції:

- створення, редагування, видалення та перегляд медичних форм пацієнтів;
- максимальне спрощення заповнення інформації шляхом введення списків з варіантами відповідей;
 - ведення інформації про заклади охорони здоров'я та їх персонал;
 - формування звітів з можливістю вибору параметрів;
 - створення, редагування та видалення звітних шаблонів;
 - збереження карток в сховищах ЗОЗ, де стоїть на обліку пацієнт;
 - ведення архіву недійсних карток та карток, переведених в інші ЗОЗ.

1.5 Висновки по розділу

В Україні активно проводиться впровадження інформаційних технологій в усі сфери життя людини. В медичній сфері широкого застосування набувають системи електронних медичних карт, які дозволяють пришвидшити роботу лікарів та зібрати максимум інформації в одному місці. Такі карти дозволяють легко обробляти отриману інформацію та формувати інформативні звіти. Але такі звіти можуть вважатись інформативними лише за умови єдиної бази по країні, без дублів та загублених карток.

Питання створення програмного забезпечення, що забезпечуватиме таку функціональність, є актуальним. Було проведено аналіз продуктів-аналогів на ринку, визначено їх недоліки та сильні сторони. З урахуванням проведеного аналізу та зібраної інформації від медичних закладів, було сформульовано вимоги. За допомогою виявлення усіх можливих варіантів використання програмного продукту, було визначено функції, які необхідно реалізувати та цілі, яких бажано досягнути в кінцевому варіанті продукту.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

2.1.1 Вхідні та вихідні дані

Вхідні дані

Сформовані на основі паперових медичних форм №501-1, №501-2, №501-3(1), №501-3(2), №501-3(3), з метою щоб зменшити час адаптації лікарів до нових умов роботи. Для створення єдиної системи обліку електронних карток по Україні, довелося впровадити в форми деякі зміни. Окрім полів, зазначених в цих формах, також були додані поля для позначення приналежності картки до ЗОЗ, додаткова інформація, видозмінено деякі поля на таблиці для зручності. Також було створено структуру «Заклад охорони здоров'я».

Структура вхідних даних ЗОЗ відображена в таблицях 2.1 – 2.2.

Таблиця 2.1 – Поля картки реєстра ЗОЗ

Назва	Детальний опис	Обов'язковість
Прив'язка до ОШС	Технічне поле, що пов'язує ЗОЗ і деревовидну структуру, за якою здійснюється обмеження на перегляд карток.	Ні
Найменування ЗОЗ	Повна назва закладу	Так
Установа ЗОЗ	Повна назва установи	Ні
Код ЄДРПОУ	Код юридичної особи з єдиного державного реєстру	Ні
Тип ЗОЗ	ЗОЗ поділяються на безпосередньо заклади та центри СНІДу	Так
Код КОАТУУ	Однозначний код, за яким можна визначити область, регіон, місто та район реєстрації	Так

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.1

Адреса	Вулиця, будинок	Ні
Поштовий індекс	–	Ні
Лікарі	Таблиця з інформацією про персонал. Дивись таблицю 2.2.	Ні

Таблиця 2.2 – Дані про персонал ЗОЗ

Назва	Детальний опис	Обов'язковість
Прізвище	–	Так
Ім'я	–	Так
По батькові	–	Ні
Посада	Поле має словникові варіанти відповіді. Можливе додавання нового варіанту зі сторінки роботи з інформацією про ЗОЗ.	Так
Телефон	–	Ні
E-mail	–	Ні

Структура вхідних даних картки матері відображена в таблицях 2.3 – 2.9.

Таблиця 2.3 – Додаткові поля електронної форми №501-1, №501-2, додані до полів паперової форми

Назва	Детальний опис	Обов'язковість
Номер документа	Складається з 2 цифр, 2 літер і 2 цифр.	Ні
Дата заповнення	Поточна дата на момент збереження картки	Так
Ідентифікатор матері	Формується автоматично в форматі: порядковий номер картки + 3 літери (імені + фамілії + по батькові + області)	Ні
Форма №025	Номер паперової форми пацієнта	Ні

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.3

Статус картки	Поле має словникові варіанти відповіді.	Ні
Статус форми №501-1	Поле має словникові варіанти відповіді.	Ні
Тип ЗОЗ з якого отримана форма	ЗОЗ поділяються на безпосередньо заклади та центри СНІДу	Ні
Найменування ЗОЗ	Формується список з вибором існуючих в системі карток з інформацією про ЗОЗ	Ні
Центр ЗОЗ, в якому отримали форму №501-1	Формується список з вибором існуючих в системі карток з інформацією про ЗОЗ. Автоматично приймає значення ЗОЗ, в гілці дерева якого було створено картку.	Так
Дата отримання форми	–	Ні
Прізвище лікаря, що опрацьовує форму	Формується список з вибором існуючих в таблиці лікарів з відповідно вибраної картки ЗОЗ.	Ні
Додаткова інформація	–	Ні
Системна інформація	Таблиця з інформацією про зміни в картці. Дивись таблицю 2.4.	Ні

Таблиця 2.4 – Додаткові поля електронної форми №501-1, додані до полів паперової форми

Назва	Детальний опис	Обов'язковість
Лабораторні дані	Таблиця з інформацією про обстеження. Дивись таблицю 2.3.	Ні

Таблиця 2.5 – Лабораторні дані

Назва	Детальний опис	Обов'язковість
Назва обстеження	Поле має словникові варіанти відповіді.	Ні
Етап дослідження	Поле має словникові варіанти відповіді.	Ні
Тип дослідження	Поле має словникові варіанти відповіді.	Ні
Дата забору	–	Ні
Номер зразка	–	Ні
Дата проведення	–	Ні
Результат	Текстове поле для коментаря	Ні

Таблиця 2.6 – Системна інформація

Назва	Детальний опис	Обов'язковість
ПІБ	Поле формується автоматично по профілю користувача.	Так
Посада	Поле формується автоматично по профілю користувача.	Так
Дата та час	Поле формується автоматично по профілю користувача.	Так
Дія	Поле формується автоматично по дії над картою (редагування, створення, тощо)	Так

Таблиця 2.7 – Додаткові поля електронної форми №501-2, додані до полів паперової форми

Назва	Детальний опис	Обов'язковість
Лабораторні дані	Таблиця з інформацією про обстеження. Дивись таблицю 2.3.	Ні
Наявність супутніх захворювань	Таблиця з можливістю додати назву супутнього захворювання. Назви захворювань обираються з можливих варіантів.	Ні

Продовження таблиці 2.7

АРТ лікування	Таблиця з інформацією про лікування АРВ. Дивись таблицю 2.6.	Ні
Діти	Таблиця з інформацією про дітей. Поля повторюють паперову форму. Містить у собі таблицю АРВ профілактика дитини. Дивись таблицю 2.7.	

Таблиця 2.8 – АРТ лікування

Назва	Детальний опис	Обов'язковість
Період АРТ	Поле має словникові варіанти відповіді.	Так
Дата призначення	–	Ні
Тиждень вагітності	Доступне за умови вибору відповідного періоду АРТ	Ні
Схема лікування	Таблиця з можливістю додати назву препарату. Назви препаратів обираються з можливих варіантів.	Ні

Таблиця 2.9 – АРВ профілактика дитини

Назва	Детальний опис	Обов'язковість
Назва препарату	Поле має словникові варіанти відповіді.	Ні
Кількість днів	–	Ні

Структура вхідних даних картки дитини відображена в таблицях 2.10 – 2.11.

Таблиця 2.10 – Додаткові поля електронної форми №501-3(1), №501-3(2), №501-3(3), додані до полів паперової форми

Назва	Детальний опис	Обов'язковість
Номер документа	Складається з 2 цифр, 2 літер і 2 цифр.	Ні

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Продовження таблиці 2.10

Дата заповнення	Поточна дата на момент збереження картки	Так
Ідентифікатор дитини	Формується автоматично в форматі: порядковий номер картки + 3 (літери імені + фамілії + по батькові + області)	Ні
Форма №025	Номер паперової форми пацієнта	Ні
Статус картки	Поле має словникові варіанти відповіді.	Ні
Статус форми №501-3(1)	Поле має словникові варіанти відповіді.	Ні
Тип ЗОЗ з якого отримана форма	ЗОЗ поділяються на безпосередньо заклади та центри СНІДу	Ні
Найменування ЗОЗ	Формується список з вибором існуючих в системі карток з інформацією про ЗОЗ даного типу	Ні
Центр ЗОЗ, в якому отримали форму №501-1	Формується список з вибором існуючих в системі карток ЗОЗ даного типу. Автоматично приймає значення ЗОЗ, в гілці дерева якого було створено картку.	Так
Дата отримання форми	–	Ні
Прізвище лікаря, що опрацьовує форму	Формується список з вибором існуючих в таблиці лікарів з відповідно вибраної картки ЗОЗ.	Ні
Додаткова інформація	–	Ні
Системна інформація	Таблиця з інформацією про зміни в картці. Дивись таблицю 2.4.	Ні

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.11 – Додаткові поля електронної форми №501-3(3), додані до полів паперової форми

Назва	Детальний опис	Обов'язковість
АРТ схема	Таблиця з можливістю додати назву препарату. Назви препаратів обираються з можливих варіантів.	Ні

Вихідні дані

Вихідними даними є сформовані по шаблонам звіти з урахуванням певних параметрів [2]. Приклади результату такого звіту наведений в таблицях 2.12 – 2.13.

Таблиця 2.12 – Структура звітної форми 1001 «Консультація і тестування пацієнок з ВІЛ»

Назва показника	Детальний опис
Кількість вагітних	Враховуються ті, що звернулись до ЗОЗ вперше за звітний період.
Застосування ІФА	Кількість вагітних
Застосування 2 швидких тестів	Кількість вагітних
Застосування 1 швидкого тесту	Кількість вагітних
Інший алгоритм	Кількість вагітних
З підтвердженням ВІЛ	Кількість вагітних
Консультація	Кількість вагітних, яким провели перед та після тесту консультацію.
Не перебувають на обліку ЗОЗ	Кількість вагітних

Таблиця 2.13 – Структура звітної форми 2000 «Завершення вагітності ВІЛ-інфікованих жінок»

Назва показника	Детальний опис
Кількість ВІЛ-інфікованих	Пацієнти, в яких були виявлені нові випадки вагітності в звітному році.
ВІЛ встановлено до вагітності	Кількість вагітних
ВІЛ встановлено до 12 тижнів вагітності	Кількість вагітних
ВІЛ встановлено до 26 тижнів вагітності	Кількість вагітних
ВІЛ встановлено після 26 тижнів вагітності	Кількість вагітних
ВІЛ встановлено у пологах	Кількість вагітних
ВІЛ встановлено після пологів	Кількість вагітних, яким провели перед та після тесту консультацію.

2.1.2 Моделювання бізнес-процесів

Основний бізнес-процес системи – створення звітності на основі отриманих даних про пацієнтів та їх лікування. Діяльність цього процесу відображено на рисунках 2.1 – 2.2 в нотації IDEF0.

На діаграмі А-0 зображено загальний вигляд діяльності.

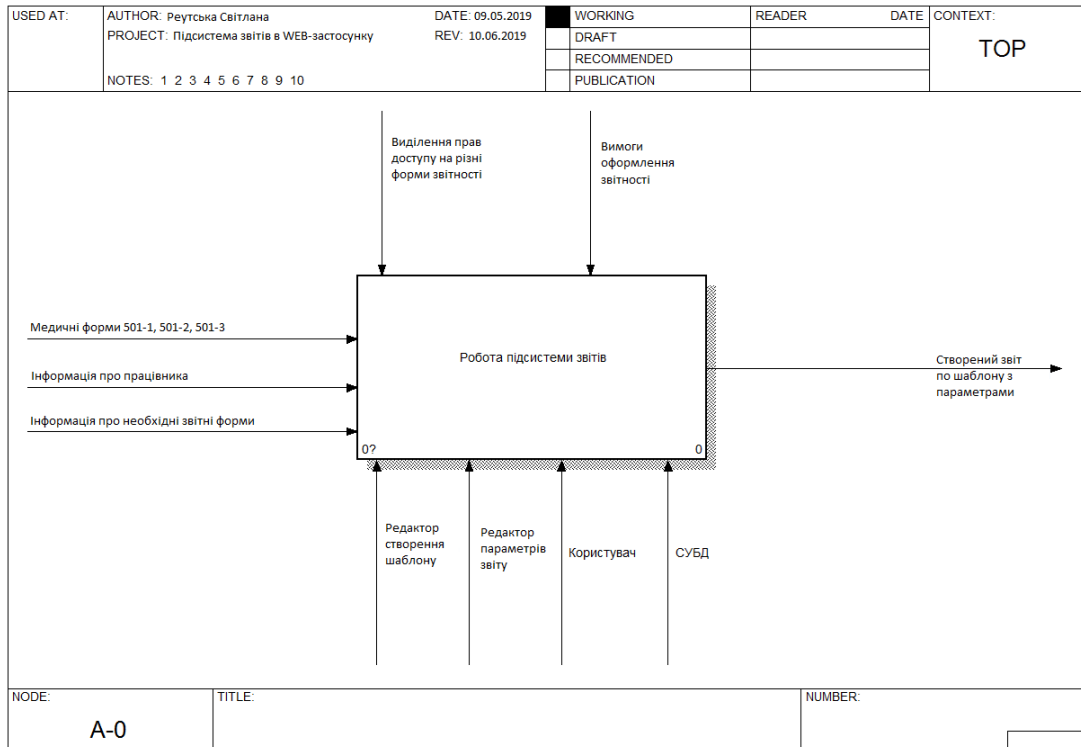


Рисунок 2.1 – Схема структурна бізнес-процесу

На рисунку 2.2 зображено декомпозиції процесу роботи зі звітністю. Створення звіту включає в себе три основні механізми: створення шаблону для звіту, вибір шаблону та заповнення параметрів, генерація файлу звіту. Нижче відображені вхідні та вихідні дані для процесів, а також механізми управління.

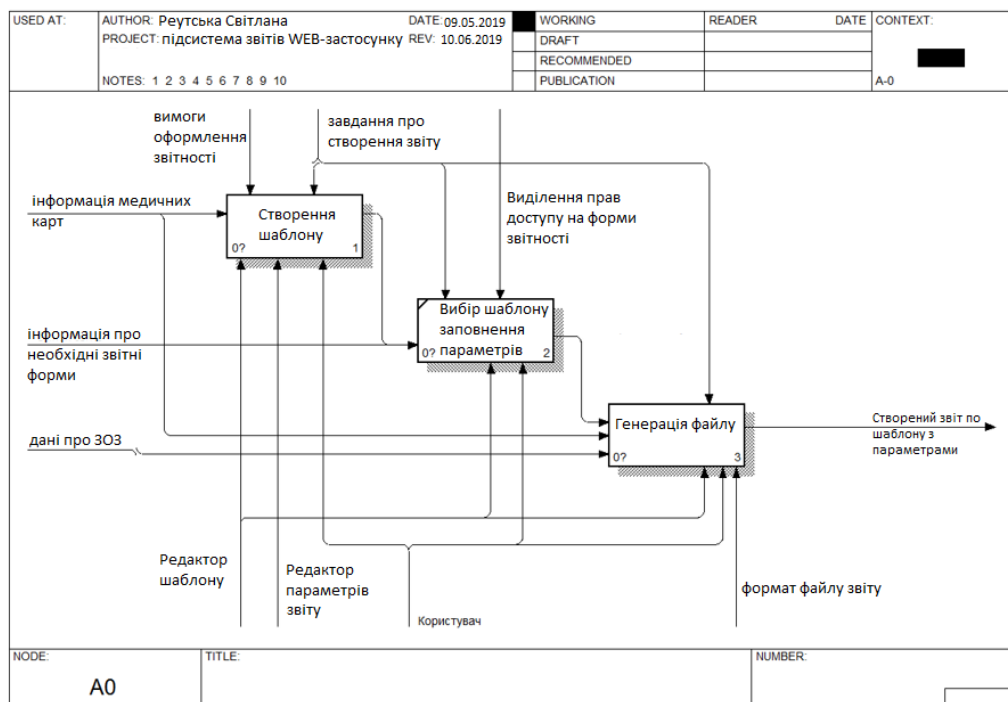


Рисунок 2.2 – Схема структурна бізнес-процесу. Декомпозиція рівень 1

Змн.	Арк.	№ докум.	Підпис	Дата

Важливу роль в системі відіграє процес створення електронних версій медичних карток, на основі яких відбуватиметься процес формування звітності. На рисунках 2.3 – 2.4 зображено структурні схеми станів системи заповнення медичних карт та створення звіту на їх основі.

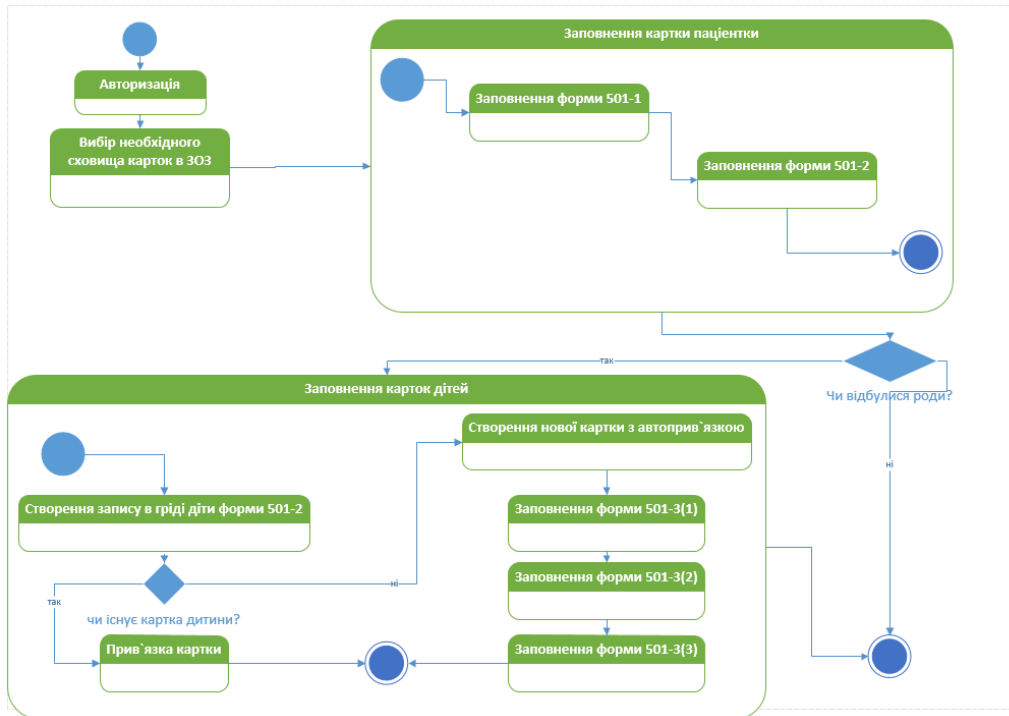


Рисунок 2.3 – Схема структурна заповнення медичних карт

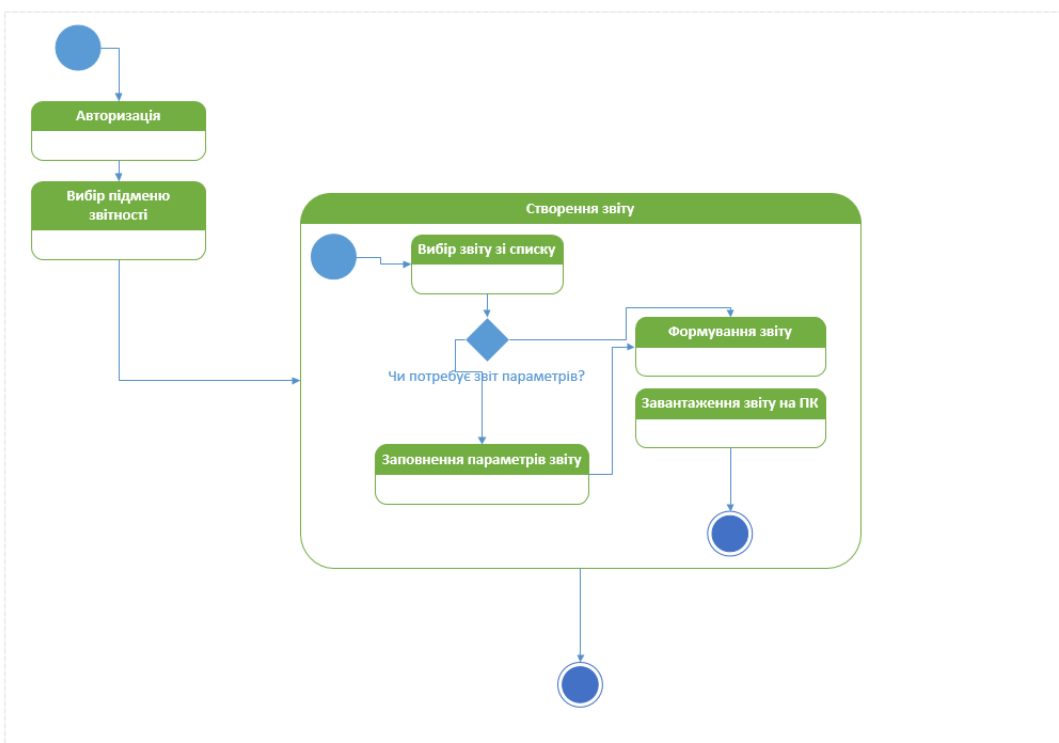


Рисунок 2.4 – Схема структурна створення звіту

Змн.	Арк.	№ докум.	Підпис	Дата

2.2 Архітектура програмного забезпечення

В розробці WEB-застосування було використано архітектурний шаблон MVC (Model-View-Controller) [7].

У відповідності до його правил, застосунок ділиться на наведені нижче три компоненти.

– Модель – структура, що є відображенням даних і бізнес-логіки. Зазвичай модель змінює стани інформації в базі даних та виконує їх обробку.

– Представлення – інтерфейс, який видимий користувачу. Слугує інструментом для відображення інформації в моделі та засоби для роботи з контролером.

– Контролер – обробляє запити, що приходять з частини клієнта на серверну. У відповідності до команд, що приходять з боку клієнта, впливає на модель.

ASP.Net MVC – набір бібліотек, що реалізовує даний архітектурний шаблон.

Представлення створюються в форматі .cshtml, сумісним як з мовою програмування C#, так і з html, css та js командами. Така гнучкість дає змогу відділити клієнтську частину від серверної, без проблем використати js бібліотеки і передати модель на представлення в її звичайному вигляді. Важливим плюсом використання цього фреймворка також є гнучка маршрутизація з можливістю налаштувати її під себе. В результаті URL сторінки стають читабельними та можуть скрити елементи реалізації функціоналу (приховані параметри).

Загалом роботу ASP.Net MVC можна представити у вигляді схеми, представленої на рисунку 2.5.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

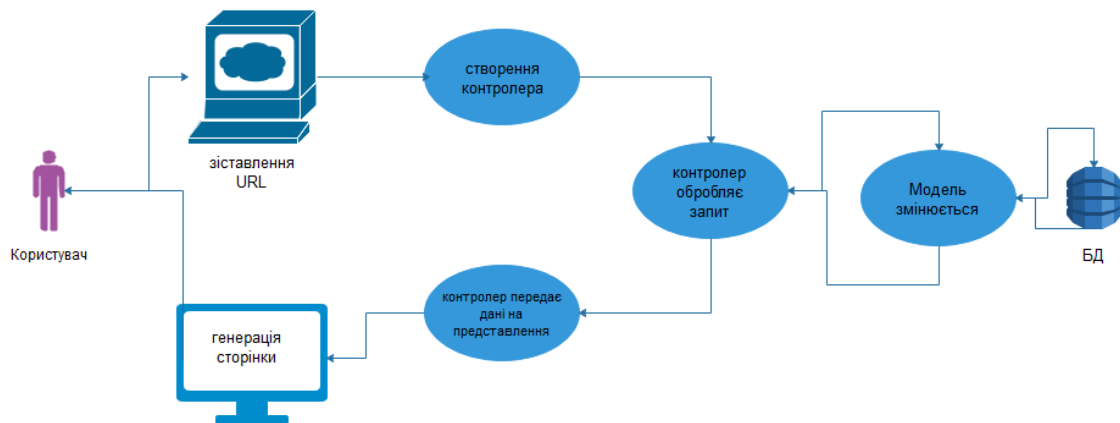


Рисунок 2.5 – Взаємодія компонентів ASP.Net MVC

Процес, представлений на рисунку можна розкласти на декілька етапів.

Етап 1. Користувач вводить URL адресу в браузері. За допомогою маршрутизації визначається контролер, на який подається запит (RegisterRoutes).

Етап 2. Контролер отримує запит і обробляє його. Зазвичай обробка включає в себе проведення маніпуляції з моделлю системи.

Етап 3. Модель змінює свій стан, зберігає його в базі та вертає інформацію про зміни.

Етап 4. Контролер повертає сутність типу ActionResult і передає туди назву представлення та його параметри.

Етап 5. Користувач бачить результат свого запиту на екрані.

2.2.1 Структура бази даних

В якості системи керування базами даних було обрано MS SQL SERVER. Оскільки ASP.Net MVC фреймворк дає широкую інструментальну базу для роботи з ним.

Повна схема Бази даних застосунку наведена в графічних матеріалах.

Опис структури бази даних:

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Таблиця 2.14 – Список основних таблиць та їх призначення

Назва таблиці	Призначення
Dossier	Містить у собі назви усіх карток та їх розділів системи (ЗОЗ, Матері, Дитини).
DossierAttribute	Містить інформацію про назву, тип та їх місце в DossierValue.
DossierList	Містить заголовки створених карток, дати їх створення та видалення.
DossierValue	Містить набір значень полів для карток, описаних в DossierList.
GridView	Містить набір значень полів в таблицях для карток, описаних в DossierList.
Dictionary	Містить заголовки створених словників для заповнення полів карток.
DictionaryValue	Містить значення словників, описаних в Dictionary.
Tree	Містить записи – елементи деревовидної структури для відображення ієрархії закладів та користувачів, що до них прив'язані.
DossierTreeLink	Містить зв'язки між картками та місцем в ієрархії.
DataType	Містить опис типів полів карток системи.
Report	Містить шаблони звітів.
ReportExtraProp	Містить параметри звітів за замовчуванням.
Koatu	Містить назви населених пунктів та їх коди.
TypeName	Містить назви типів полів.

Нижче наведено список колонок для кожної таблиці (таблиці 2.15-2.28).

Таблиця 2.15 – Основні колонки таблиці Dossier

Назва	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Продовження таблиці 2.15

LId	Ідентифікатор картки для її розділу	int	FK
LongName	Повна назва	varchar(1023)	
ShortName	Коротка назва	varchar(1023)	

Таблиця 2.16 – Основні колонки таблиці DossierAttribute

Назва	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DossierId	Ідентифікатор запису таблиці Dossier	int	FK
LId	Ідентифікатор батьківського атрибуту (для таблиць)	int	FK
LongName	Повна назва	varchar(1023)	
ShortName	Коротка назва	varchar(1023)	
DicId	Ідентифікатор запису таблиці Dictionary за потреби	int	
FieldName	Місце поля в таблиці DossierValue	varchar(25)	

Таблиця 2.17 – Основні колонки таблиці DossierList

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DossierId	Ідентифікатор розділу картки з таблиці Dossier	int	FK
CreateDate	Дата створення	datetime	
DeleteDate	Дата видалення	datetime	
LastDate	Дата останньої зміни	datetime	
UserName	Ім'я користувача, що створив картку	varchar(1023)	

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.18 – Основні колонки таблиці DossierValue

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
UserName	Ім'я користувача, що заповнив картку	varchar(1023)	
DossierListId	Ідентифікатор запису таблиці DossierList	int	FK
TXT_1 ... TXT_150	Для запису текстових полів	varchar(25)	
NUM_1 ... NUM_150	Для запису числових, булевих та словникових полів	double	
DATE_1 ... DATE_150	Дата запису полів-дат	datetime	

Таблиця 2.19 – Основні колонки таблиці GridValue

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DossierValueId	Ідентифікатор запису таблиці DossierValue	int	FK
LId	Ідентифікатор запису таблиці GridValue якщо це вкладений грід.		FK
TXT_1 ... TXT_150	Для запису текстових полів	varchar(25)	
NUM_1 ... NUM_150	Для запису числових, булевих та словникових полів	double	

Продовження таблиці 2.19

DATE_1 ... DATE_150	Дата запису полів-дат	datetime	
---------------------------	-----------------------	----------	--

Таблиця 2.20 – Основні колонки таблиці Dictionary

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
LId	Ідентифікатор запису таблиці Dictionary якщо словники вкладені	int	FK
Sort	Сортування в яку сторону відображати	varchar(1023)	
LongName	Повна назва	varchar(1023)	
ShortName	Коротка назва	varchar(1023)	

Таблиця 2.21 – Основні колонки таблиці DictionaryValues

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DictionaryId	Ідентифікатор запису таблиці Dictionary	int	FK
LId	Ідентифікатор запису таблиці DictionaryValues якщо значення залежні	int	FK
Sort	Порядок сортування	int	
LongName	Повна назва	varchar(1023)	
ShortName	Коротка назва	varchar(1023)	

Таблиця 2.22 – Основні колонки таблиці Tree

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
LId	Ідентифікатор запису таблиці StructureTree, який є батьківським	int	

Продовження таблиці 2.22

Type	Тип елемента дерева (головний ЗОЗ, підлеглі ЗОЗ, користувач системи в ЗОЗ)	int	
Code	Код для ідентифікації вершин в межах одного типу по групам	varchar(25)	
LongName	Повна назва	varchar(1023)	
ShortName	Коротка назва	varchar(1023)	

Таблиця 2.23 – Основні колонки таблиці DossierTreeLink

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DossierListId	Ідентифікатор запису таблиці DossierList	int	FK
TreeId	Ідентифікатор запису таблиці StructureTree	int	FK
Date	Дата прив'язки	datetime	
User	Код користувача що здійснив прив'язку	varchar(1023)	

Таблиця 2.24 – Основні колонки таблиці DataType

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
DossierAttributeId	Ідентифікатор запису таблиці DossierAttribute	int	FK
RegularExp	Регулярний вираз для валідації	varchar(1023)	
TypeId	Ідентифікатор типу поля	int	
Message	Повідомлення про помилку в разі невиконання регулярного виразу	varchar(1023)	
MinLong	Мінімальна допустима довжина	int	

Продовження таблиці 2.24

MaxLong	Ммаксимальна допустима довжина	int	
---------	--------------------------------	-----	--

Таблиця 2.25 – Основні колонки таблиці Report

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
Name	Ідентифікатор запису таблиці DosierAttribute	int	
SQLQuery	Тіло шаблону	varchar(1023)	
Parametrs	Параметри звіту у вигляді JSON	varchar(1023)	
NameTemplate	Унікальна назва латиною	varchar(1023)	

Таблиця 2.26 – Основні колонки таблиці ReportExtraProp

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
ReportId	Ідентифікатор запису таблиці Report	int	FK
Code	Кодова назва параметру	varchar(1023)	
Value	Значення параметру	varchar(1023)	

Таблиця 2.27 – Основні колонки таблиці Koatu

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK
Code	Код населеного пункту	int	
Name	Назва населеного пункту	varchar(1023)	

Таблиця 2.28 – Основні колонки таблиці TypeName

Код	Коментар	Тип даних	Key
Id	Ідентифікатор	int	PK

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Продовження таблиці 2.28

Value	Назва типу	varchar(1023)	
-------	------------	---------------	--

Для збереження профілів користувачів, а також їх логінів і паролів, використовуються стандартні рішення ASP.Net MVC. Усі дані зберігаються в таблицях AspNetUsers, AspNetRoles, AspNetUserRoles. Схему цієї частини бази даних можна побачити на рисунку 2.6.

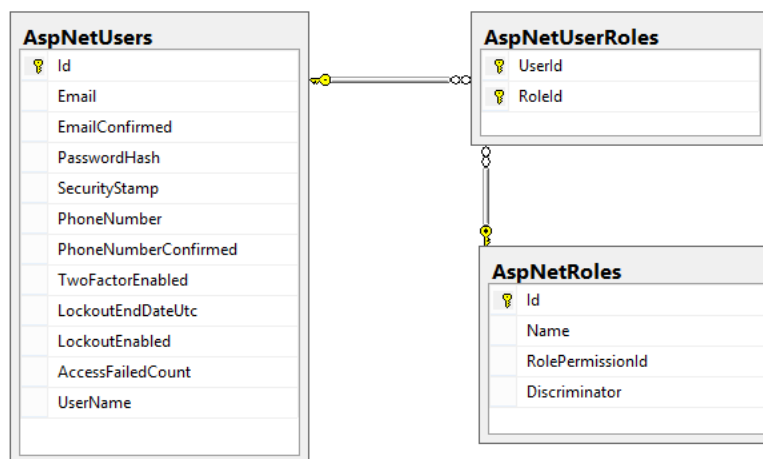


Рисунок 2.6 – Стандартне рішення питання авторизації користувачів від ASP.Net MVC

2.2.2 Структура класів

Структурна схема класів WEB-застосування представлена в графічних матеріалах у вигляді UML діаграми.

Нижче представлений опис основних класів, які призначені безпосередньо для роботи з бізнес-логікою. Допоміжні класи – контролери описані в розділі 2.3.

Таблиця 2.29 – Основні класи WEB-застосування

Назва	Опис
DossierData	Клас, який містить у собі всю інформацію про медичну карту. Містить колекцію DossierSection.

Продовження таблиці 2.29

DossierSection	Клас, який містить у собі всю інформацію про окрему медичну форму. Містить колекцію DossierAttribute.
DossierAttribute	Клас з параметрами поля медичної форми. Може містити колекцію дочірніх DossierAttribute (для таблиць).
DossierService	Клас, що забезпечує роботу з картками пацієнтів.
DossierRepository	Клас, що забезпечує роботу з моделями в базі даних.
DictionaryManager, DictionaryService	Класи, що забезпечують роботу з словниками готових відповідей.
ReportService	Клас, що забезпечує роботу з шаблонами звітності.
Crypt	Клас, що забезпечує шифрування даних пацієнтів
UserInfo	Клас, що містить у собі інформацію про поточного користувача, його ролі.
LinkedDossierService	Клас, що створює та видаляє зв'язки між картками пацієнтів.
DossierAttributeManager	Клас, що забезпечує роботу зі складними полями картки (таблицями).
SectionExtention	Клас, методи якого змінюють стандартну логіку роботи з картою.
DocNumExtention	Клас, методи якого реалізують автоматичне створення унікальної нумерації карток.
PatientService	Клас, що забезпечує логіку роботи та утворення значень медичних карток матері та дитини.
ChildCardModel	Модель для пошуку картки дитини з метою здійснення її прив'язки.
ChildModel	Модель з основними полями картки дитини для копіювання.
DoctoeModel	Модель з інформацією про доктора для копіювання.

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.29

PatientModel	Модель з основними полями картки пацієнтки для копіювання.
ZOZInfoModel	Модель з основними полями картки ЗОЗ для копіювання.
ReportValue	Модель з даними про звіт.

Більшість з класів, наведених вище, реалізують стандартний набір функцій роботи з базою даних. В таблицях 2.30 – 2.37 показане умовне представлення цих методів.

Таблиця 2.30 – Абстрактний опис стандартних методів роботи з базою

Тип повернення	Назва методу	Опис
List<object>	GetAll	Вертає колекцію об'єктів
object	GetByIndex	Вертає об'єкт або null
object	GetByName	Вертає об'єкт або null
–	Insert	Зберігає об'єкт
–	Update	Змінює об'єкт
–	Delete	Видаляє об'єкт

Нижче наведено головні методи класів, що безпосередньо служать цілям програмного забезпечення.

Таблиця 2.31 – Основні методи класу PatientService

Тип повернення	Назва методу	Призначення
List <PatientModel>	GetCardData	Для механізму пошуку картки для перенесення. Шукає співпадіння та вертає інформацію про пацієнток.

Продовження таблиці 2.31

object	GetDicValueByCode	Для ідентифікації особливих варіантів відповідей передбачений унікальний код. Даний метод вертає Id відповіді по коду.
PacientModel	GetInfo34003	Вертає інформацію, що дублюється на формі 501-2 з форми 501-1.
ChildModel	GetInfoChild	Вертає інформацію про дитину для заповнення таблиці в картці матері
–	Set34009Info	Збирає повну інформацію з форм 501-3(1), 501-3(2), 501-3(3) та заповнює нею останній 4й розділ картки дитини.
ZOZInfoModel	GetInfoFromZOZ	Збирає інформацію про ЗОЗ для автозаповнення в картках матері та дитини.
List <decimal>	GetArchIds	Вертає Id записів деревовидної структури, які є сховищами для карток.
List <ChildCardModel>	LoadChildrens	Вертає інформацію про дитину з урахуванням пошукових параметрів для пошуку прив'язки картки дитини до матері.
–	SetLink	Здійснює прив'язку картки матері до дитини та видаляє стару за необхідності.
decimal	GetCurrentOSS	Вертає Id гілки деревовидної структури, в якій знаходиться сховище для картки.

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.31

string	SendToArchive	Переносить картку до архівного сховища поточного ЗОЗ (вона стає недоступною для редагування)
string	SendFromArchive	Вертає з архівного сховища картку до основного.
string	ChangePlace	Змінює ЗОЗ для картки, автоматично переносить її до нового сховища та створює в архіві старого ЗОЗ її копію.
void	CheckIds	Перевіряє чи всі зв'язки між картками пацієнтів валідні, за необхідності видаляє їх.
decimal	CreateChild	Створює нову картку дитини з автоматичною прив'язкою її до картки матері в тому ж сховищі ЗОЗ.
void	UnlinkChild	Видаляє зв'язок між картою матері та дитини.

Таблиця 2.32 – Основні методи класу SectionExtention для картки ЗОЗ

Тип повернення	Назва методу	Призначення
string	BeforeInsert, BeforeUpdate	Запобігає створенню дублів ЗОЗ для одної вершини деревовидної структури.
void	AfterInsert	Робить запис про ЗОЗ доступним усім користувачам системи для вибору у картці пацієнта.

Таблиця 2.33 – Основні методи класу SectionExtention для картки матері

Тип повернення	Назва методу	Призначення
string	AfterUpdate	Перевіряє зв'язки між картками матері та дитини на валідність.
void	AfterInsert	Зберігає перший запис в таблиці розміщення картки.
Decimal[]	OnUpdate	Копіює дані між формами 501-1 та 501-2.
void	SetGridRows	Автоматично заповнює дані в таблицю про методи дослідження. Користувачеві лишається лише вбити результати в потрібні рядки.
string	BeforeDelete	Не дає видалити картку з прив'язаною дитиною.

Таблиця 2.34 – Основні методи класу GridExtention для картки матері гріда Діти

Тип повернення	Назва методу	Призначення
void	OnSave	Якщо була обрана картка дитини для зв'язку, то зберігає в цій картці посилання на картку матері.
bool	OnDelete	Очищує зв'язок з матір'ю якщо такий мав місце бути.

Таблиця 2.35 – Основні методи класу SectionExtention для картки дитини

Тип повернення	Назва методу	Призначення
void	AfterInsert	Зберігає перший запис в таблиці розміщення картки.

Продовження таблиці 2.35

void	SetGridRows	Автоматично заповнює дані в таблицю про методи дослідження. Користувачеві лишається лише вбити результати в потрібні рядки.
string	BeforeDelete	Видиляє зв'язок з матір'ю.

Таблиця 2.36 – Основні методи класу ViewReports

Тип повернення	Назва методу	Призначення
List <object>	GetReportList	Для механізму пошуку картки для перенесення. Шукає співпадіння та вертає інформацію про пацієток.
List <object>	ViewReportParams	Формує список параметрів, які має ввести користувач для створення звіту.
FileContentResult	DownloadFile	Формує файл, який користувач збереже на свій ПК.
ReportValue	GetOneReport	Шукає звіт з усіма його параметрами по ідентифікатору.
Tuple <Report, List <ReportParams>>	ParseXML	Працює з XML виглядом параметрів звіту.

Таблиця 2.37 – Основні методи класу Crypt

Тип повернення	Назва методу	Призначення
string	Encrypting	Шифрує строку методом AES CBC.
string	Decrypting	Дешифрує строку зашифровану методом AES CBC.

2.3 Конструювання програмного забезпечення

Дане програмне забезпечення розроблялось на мові С# за допомогою фреймворку ASP.Net MVC [7]. Основними перевагами такого рішення можна вважати:

- відокремлена серверна частина від частини клієнта;
- стандартна реалізація авторизації в системі та розподілення користувачів на ролі;
- можливість створювати свої допоміжні засоби для розробки програмного забезпечення;
- наявність системи класів, що спрощують обробку HTTP-запитів;
- можливість використовувати зручну бібліотеку запитів LINQ;
- вільний вибір бібліотек клієнтської частини.

Інтерфейс користувача розроблено за допомогою інструментів: HTML, CSS, JS. Представлення в форматі cshtml дозволили без додаткових зусиль поєднати між собою елементи, написані мовою С# та елементи, написані JS.

Також для отримання красивого та інформативного інтерфейсу було використано бібліотеку jQuery. Вона дала можливість реалізувати відображення таблиць в звичній для користувача формі, полегшити роботу з реалізацією функціоналу обробки подій та спростити використання AJAX.

Для передачі даних з сервера на клієнт було використано актуальний текстовий формат JSON. Він ідеально підходить для передачі складних типів даних, оскільки за своєю структурою нагадує об'єкт класу.

2.3.1 Представлення

Нижче наведено список основних сторінок системи (для кожної AddEdit є відповідна сторінка View):

- MainPage.cshtml – головна сторінка в системі;
- LogOn.cshtml – сторінка авторизації;

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

- AccesDenied.cshtml – сторінка, куди переходить користувач при спробі відкрити картку що йому не доступна;
- OrgStructure/List.cshtml – сторінка відображення деревовидної структури;
- OrgStructure/AddEditDepartment.cshtml – сторінка редагування головного ЗОЗ;
- OrgStructure/AddEditEmployee.cshtml – сторінка редагування підконтрольних ЗОЗ;
- OrgStructure/AddEditClient.cshtml – сторінка редагування працівників, підконтрольних ЗОЗ;
- Docs/AddEdit.cshtml – головна сторінка редагування картки;
- Docs/List.cshtml – сторінка відображення списку карток;
- Docs/DepartmentsTreePartial.cshtml – сторінка відображення деревовидної структури над списком карток;
- ViewReports/List.cshtml – сторінка відображення списку шаблонів при виборі створення звіту;
- ViewReports/ViewReport.cshtml – сторінка введення параметрів шаблону при виборі створення звіту;
- Reporting/List.cshtml – сторінка відображення списку шаблонів в меню створення шаблонів;
- Reporting /AddEdit.cshtml – сторінка редагування шаблону звіту;
- RPPMD/34001/AddEdit.cshtml – сторінка редагування інформації про ЗОЗ;
- RPPMD /34003/AddEdit.cshtml – сторінка редагування форми 501-1;
- RPPMD/34004/AddEdit.cshtml – сторінка редагування форми 501-2;
- RPPMD /34006/AddEdit.cshtml – сторінка редагування форми 501-3(1);
- RPPMD/34007/AddEdit.cshtml – сторінка редагування форми 501-3(2);

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

- РРМД /34008/AddEdit.cshtml – сторінка редагування форми 501-3(3);
- РРМД/34009/AddEdit.cshtml – сторінка редагування форми 501-3 загальної;
- РРМД /34013/AddEdit.cshtml – сторінка інтерфейсу перенесення картки матері або дитини в інший ЗОЗ.

Для полегшення розробки представлень медичних карток, були розроблені html-helpers для різних типів полів. В таблиці 2.38 представлено опис цих інструментів.

Таблиця 2.38 – Допоміжні інструменти розробки представлень

Назва	Параметри	Ціль
Caption	Атрибут	Витягує коротку назву поля з бази даних.
Label	Атрибут	Витягує значення поля в текстовому форматі (для словникових значень, значень-карток та вершин деревовидної структури витягує коротку назву відповідного значення).
Validator	Атрибут	Витягує правила, накладені на поле картки, та текст помилки з бази.
Text	Атрибут	Створює поле для вводу однорядкового тексту.
MultiText	Атрибут, ознака розмітки	Створює поле для вводу однорядкового тексту, залежно від параметра з розміткою.

Продовження таблиці 2.38

Dictionary	Атрибут, ознака додавання нових значень, ознака залежності від інших словників, ознака автозаповнення	Створює поле (випадаючий список) з урахуванням усіх вказаних ознак.
Date	Атрибут	Створює поле з кнопкою (вибір дати за календарем).
Float	Атрибут, Ознака чи є грошовою одиницею	Створює поле для числа з плаваючою комою.
Int	Атрибут	Створює поле для цілого числа.
Dosier	Атрибут, Id картки для вибору, Id поля яке відобразатиметься	Створює випадаючий список з автозаповненням, сформований з карток.
HistoryTable	Id картки	Формує таблицю з історією усіх змін що відбувались з картокою.
Table	Атрибут, кнопки	Формує таблицю зі значень в базі даних (кеші), відображає кнопки (редагування, створення, тощо) відповідно до параметрів.

2.3.2 Контролери

Контролери в системі виконують роль «буферу» між командами клієнта та сервісами, які виконують їх команди. Список основних контролерів застосування та їх призначення:

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

- PacientController – відповідає за зміну взаємозв'язків карток в системі, перенесення карток до інших ЗОЗ, копіювання даних. Вертає значення полів карток у вигляді JSON;
- DocsController – відповідає за збереження, зміну та видалення карток, їх доступність згідно з деревовидною структурою та пошук карток за параметрами;
- ReportingController – відповідає за збереження, зміну та видалення шаблонів звітності, а також за зміну їх додаткових параметрів;
- ViewReportsController – відповідає за формування списку доступних звітів, заповнення параметрів та формування файлу звіту;
- AccountController – відповідає за авторизацію користувача в системі.

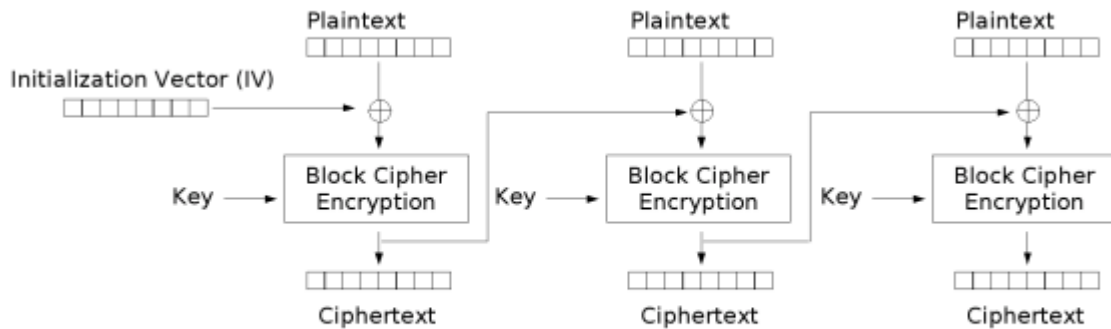
2.4 Аналіз безпеки даних

При створенні системи, що доступна через інтернет-мережу, важливо продумати інструменти захисту персональних даних. Доступ до сторінок системи, окрім головної, здійснюється лише для авторизованих користувачів. Профілі мають чітко визначені ролі, які являються ключем доступу до тієї чи іншої функціональності. Застосування також захищене від недобросовісних спроб користувача ввести команду доступу до даних безпосередньо в адресну строку – кожного разу програма перевіряє чи є права на цю команду для даного профілю.

Оскільки при розробці програмного забезпечення було використано стандартний спосіб авторизації користувачів, стало доцільним використати спосіб збереження паролів, запропонований ASP.Net Identity [9]. Було обрано тип аутентифікації Individual User Account. Паролі зберігаються в базі у вигляді хешу, хешуються з додаванням солі за допомогою алгоритму PBKDF2. Швидкість перебору паролю таким алгоритмом порівняно низька.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Окрім збереження паролів постає питання, що робити з персональною інформацією пацієнтів системи. ВІЛ інфіковані люди, нажаль, все ще викликають неоднозначні емоції у оточення і вимушені скривати свій статус від сторонніх осіб. Очевидно, що вона також має бути зашифрована, щоб не стати загальнодоступною у випадку злиття бази в інтернет. Було обрано симетричний алгоритм блочного шифрування AES. А саме, його CBC режим. Оскільки при його використанні два однакових незашифрованих текста перетворюються на різні зашифровані, що збільшує безпеку даних. Пароль та вектор ініціалізації згенеровані криптографічно стійким генератором випадкових чисел. Схему роботи цього алгоритму можна побачити на рисунку 2.7.



Cipher Block Chaining (CBC) mode encryption

Рисунок 2.7 – Схема роботи AES CBC

2.5 Висновки по розділу

В даному розділі було завершено процес моделювання та конструювання програмного забезпечення. А саме, було опрацьовано ряд питань:

- визначено, яку саме форму мають мати вхідні дані в системі (медичні форми), щоб бути зручними для електронного заповнення;
- здійснено обґрунтований вибір технологій, за допомогою яких буде відбуватись розробка;
- сформовано структуру бази даних;
- створено ієрархію класів та їх методи, які мають реалізовувати логіку роботи з даними програми;

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

- створено допоміжні засоби для полегшення розробки медичних карток, а також спрощення внесення змін в подальшому;
- визначено, які саме сторінки мають бути в WEB-застосуванні щоб повністю надати доступ до функціоналу системи;
- визначено методи шифрування даних, які забезпечать недоторканість персональної інформації пацієнтів та унеможливають вхід до системи з боку сторонньої особи.

Це дало змогу створити цілісний програмний продукт, що відповідає усім вимогам, описаним в першому розділі та має зручну для подальших змін та розширення структуру.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Якість програмного забезпечення можна визначити за допомогою багаторівневої структури характеристик [10]:

- функціональні можливості:
 - функціональна придатність;
 - точність;
 - здатність до взаємодії;
 - захищеність;
- надійність:
 - завершеність;
 - стійкість до відмов;
 - здатність до відновлення;
- практичність:
 - зручність в роботі;
 - зручність у вивченні;
 - зрозумілість;
 - естетична приємність;
- ефективність:
 - ефективність використання ресурсів;
- зручність супровіду:
 - зручність внесення змін;
 - зручність до перевірки роботи;
- мобільність:
 - зручність встановлення;
 - зручність в заміні версії.

В таблиці 3.1 представлена оцінка характеристик WEB-застосунку у відсотковому еквіваленті. Проаналізувавши оцінки отримаємо результат 10.94 з 13.2 можливих. Можна сказати, що програмне забезпечення високої якості.

Таблиця 3.1 – Якісні характеристики ПЗ

Назва	Оцінка	Вага	Коментар
Функціональна придатність	100%	1	Програма реалізовує усі функції, які від неї очікуються.
Точність	100%	1	Програма зберігає медичні картки в потрібному вигляді, максимально націлена на зменшення помилок вводу. Видає правильно побудовані звіти.
Здатність до взаємодії	–	1	Програма не має потреби у взаємодії з іншими системами.
Захищеність	100%	1	Неавторизованому користувачу не доступна жодна сторінка системи. Авторизовані користувачі мають доступ лише до підконтрольним їм карткам.
Завершеність	95%	1	Вірогідність відмови ПО дуже мала і пов'язана лише зі збоями у викладках нових версій. При перезапуску викладки проблема вирішується.
Стійкість до відмов	100%	1	Після виникнення відмови система або продовжує роботу у звичайному режимі, або запускає автоматичну перевикладку версії.
Здатність до відновлення	100%	1	Час відновлення роботи при відмові – до 10 хвилин

Продовження таблиці 3.1

Зручність в роботі	85%	0.9	Інтерфейс зрозумілий. Поля для заповнення мають підказки. Через особливість розміру медичних форм, можливе виникнення незручності з приводу часу заповнення однієї сторінки та її розміру.
Зручність у вивченні	90%	0.8	Інтерфейс карток максимально наближений до паперових медичних форм. Користувачу треба буде вивчити механізми переведення карт і їх прив'язки.
Зрозумілість	90%	1	Більшість елементів інтерфейсу прозора для розуміння, інші мають текстові підказки.
Естетична приємність	–	0.6	Невимірювана характеристика. Було розроблено дизайн системи, що є простим і лаконічним.
Ефективність використання ресурсів	90%	0.5	Система потребує дуже потужного серверу. З іншого боку, для клієнта достатньо звичайного офісного ПК і браузера.
Зручність внесення змін	100%	0.8	Система відкрита до розширення. Можна швидко додавати нові медичні форми, створювати нові звіти.

Продовження таблиці 3.1

Зручність до перевірки роботи	70%	0.8	Система має велику кількість функціоналу, що треба перевіряти. Також перевірка звіту вимагає аналізу існуючих або створення декілька нових медичних карт у повному обсязі. Отже, весь функціонал можна протестувати, але це потребує великих затрат часу.
Зручність встановлення	100%	1	Система встановлюється один раз на сервер спеціалістом.
Зручність в заміні версії	100%	0.8	Нова версія встановлюється автоматичною викладкою поверх старої.

Також варто відмітити показники експлуатаційної якості програмного продукту:

- результативність (виконання програмою своїх функцій, отримання повного результату, якого очікує користувач);
- безпека (захист персональної інформації);
- задоволеність (зручність використання, простота інтерфейсу, естетична краса, тощо);
- продуктивність (швидкість вирішення задач, покладених на програму).

Вплив якісних характеристик на показники експлуатаційної якості можна розглянути на рисунку 3.1.

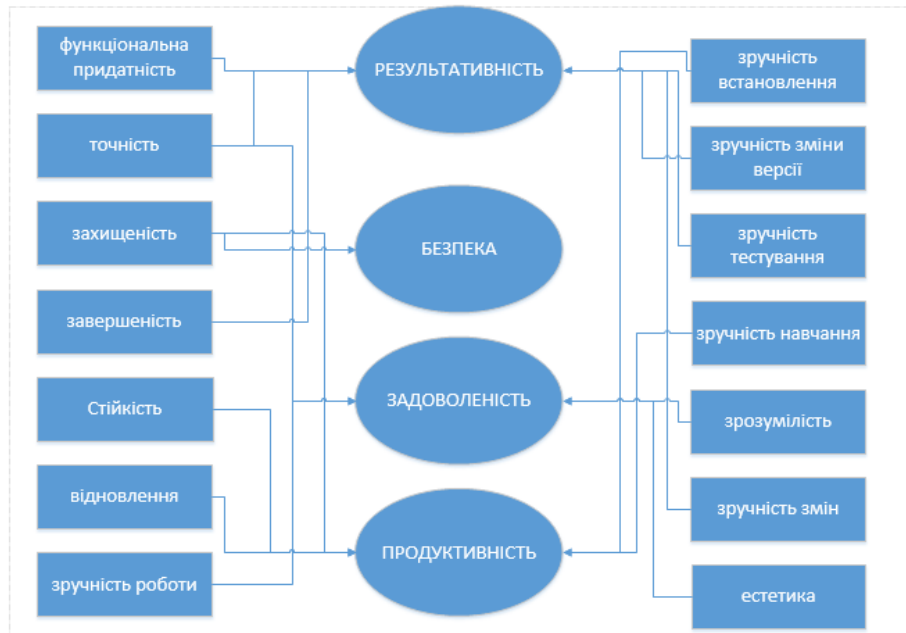


Рисунок 3.1 – Залежність експлуатаційної якості від якісних характеристик ПО

3.2 Опис процесів тестування

Тестування є дуже важливим етапом розробки програмного забезпечення. Саме воно дає можливість оцінити застосунок за критеріями якості. Тому дуже важливо покрити тестами максимальну частину системи, передбачити усі можливі випадки її використання та очікувані поведінки користувача.

Компоненти тестування:

- авторизація;
- створення профілю користувача;
- призначення ролі дозволу на використання функцій системи;
- створення, редагування та видалення карток матері;
- створення, редагування та видалення карток дитини;
- створення, редагування та видалення карток ЗОЗ;
- пошук карток за різними полями;
- доступ до карток різних ЗОЗ для користувача, що знаходиться під керуванням одного ЗОЗ;

- шифрування персональних даних картки матері;
- шифрування персональних даних картки дитини;
- копіювання даних з попередніх медичних форм в наступні;
- прив'язка, відв'язка картки дитини до картки матері;
- перенесення медичної картки до іншого ЗОЗ;
- перенесення та повернення картки в архів поточного ЗОЗ;
- створення, редагування та видалення шаблону звітності;
- формування шаблону з використанням необхідних параметрів.

Підхід:

Тестування проводитиметься у black box режимі. Важливо, щоб користувачі програмного забезпечення могли виконувати усі операції без виникнення помилок або непередбачуваних результатів. Для забезпечення високого рівня надійності, функціонал має бути протестованим як при позитивних, так і при негативних умовах. Особлива увага має приділятися системному тестуванню на базі вимог та сценаріїв користування, оскільки необхідно, щоб програма не ламалась і правильно опрацьовувала дані, введені користувачем, за будь-яких обставин.

Після викладання нової версії варто проводити регресійне тестування, що буде перевіряти роботу функціоналу, доданого в попередніх версіях.

Проводяться наступні функціональні типи тестів:

- критичне (чи зберігаються картки з усіма правильно заповненими полями, чи створюються звіти по шаблонам);
- димне (чи усі сторінки відкриваються, чи працює функціонал заповнення карток та створення шаблону звітності);
- розширене (чи працює прив'язка карток, перенесення до іншого ЗОЗ, чи правильно налаштований доступ, чи коректно працюють звіти з параметрами).

Проводяться наступні нефункціональні типи тестів:

- тестування безпеки (чи шифруються персональні дані);

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

- тестування інтерфейсу (наявність усіх кнопок та їх коректна робота, валідація полів карток, відображення збереженої інформації);
- навантажувальне тестування (чи витримає система наявність 10 тисяч карток в архівах);
- тестування сумісності (чи сумісна програма з найпопулярнішими браузерами InternetExplorer, Opera, Safari, FireFox, Chrome).

Критерії проходження тестів на прийом продукту:

- система повністю відповідає усім зазначеним в специфікаціях вимогам;
- знайдені помилки були виправлені не менш ніж на 100% критичних, 95% значних і 90% незначних;
- функціонал покрито тестами не менш ніж на 90%.

Критерії проходження тестів на обмежений прийом продукту (з умовою швидкого доопрацювання):

- система повністю відповідає усім зазначеним в специфікаціях вимогам;
- знайдені помилки були виправлені не менш ніж на 100% критичних, 95% значних і 80% незначних;
- функціонал покрито тестами не менш ніж на 80%.

Детальний опис процесів тестування наведено в додатку «Програма та методика тестування».

3.3 Опис контрольного прикладу

Під час проведення тестування WEB-застосунку було перевірено усі компоненти, зазначені в пункті 3.2. Повна інформація про приклади проведення таких перевірок відображена в таблицях 3.2 – 3.5.

Таблиця 3.2 – Тест створення форми 501-1 медичної картки матері

Мета	Перевірка можливості створення електронної форми 501-1.
------	---

Продовження таблиці 3.2

Початковий стан	Відкрита сторінка з відображенням деревовидної структури ЗОЗ.
Вхідні дані	Поля форми 501-1
Схема проведення	Обрати необхідний ЗОЗ. Натиснути кнопку «Додати картку». Заповнити усі обов'язкові поля форми 501-1. Натиснути кнопку «Зберегти і вийти»
Очікуваний результат	Створено нову картку пацієнтки в обраному ЗОЗ.
Стан програми після тестування	Відкрита сторінка з обраною попередньою ЗОЗ. На ній відображено таблицю з картами пацієнтів, що знаходяться в ньому. Серед них є збережена картка.

Таблиця 3.3 – Тест прив'язки картки дитини до картки матері

Мета	Перевірка можливості зв'язати між собою медичні карти дітей та матері.
Початковий стан	Відкрита сторінка на редагування картки матері, вкладка форми 501-2. В таблиці «Діти» є один запис. Створено хоча б одну електронну картку дитини в тому ж ЗОЗ.
Вхідні дані	–
Схема проведення	Обрати в таблиці «Діти» запис. Натиснути на кнопку «Прив'язати дитину». У випадяючому вікні обрати з таблиці потрібну дитину та натиснути кнопку «Прив'язати».
Очікуваний результат	У сховищі ЗОЗ матері створено електронну картку дитини. В таблиці «Діти» в графі «посилання» з'явиться URL-посилання на картку дитини, в якій уже заповнена інформація з таблиці «Діти».

Продовження таблиці 3.3

Стан програми після тестування	В сусідній вкладці відкривається створена картка дитини з правильно заповненими полями.
--------------------------------	---

Таблиця 3.4 – Тест перенесення картки в інший ЗОЗ адміністратором

Мета	Перевірка можливості перенесення медичної картки в інший заклад.
Початковий стан	Відкрита сторінка з таблицею історії перенесень карток.
Вхідні дані	Ідентифікатор потрібної картки, ЗОЗ, куди картку треба перенести.
Схема проведення	Натиснути на кнопку «Додати перенесення». Заповнити поле Ідентифікатора та натиснути кнопку «Підтягнути інформацію». Обрати ЗОЗ, в який треба перенести картку. Натиснути «Зберегти і закрити».
Очікуваний результат	У старому ЗОЗ в архівному сховищі створюється копія картки матері та з основного сховища картка матері недоступна. В новому ЗОЗ в основному сховищі картка матері стає доступною. В таблиці перенесень з'являється новий запис відповідно до заповнених полів.
Стан програми після тестування	Відкрита сторінка з таблицею історії перенесень. Серед них є інформація про останнє перенесення.

Таблиця 3.5 – Тест створення звіту з заповненням параметрів

Мета	Перевірка можливості створення звіту.
Початковий стан	Відкрита сторінка з списком доступних користувачеві шаблонів звітів.

Продовження таблиці 3.5

Вхідні дані	Шаблон звіту, параметри звіту.
Схема проведення	Обрати потрібний шаблон та натиснути двічі. Ввести усі необхідні параметри та натиснути кнопку «Сформувати звіт».
Очікуваний результат	Програма формує звіт згідно шаблону та параметрів, він автоматично зберігається на пристрій користувача.
Стан програми після тестування	Відкрита сторінка з таблицею карток для певного ЗОЗ.

3.4 Висновки до розділу

Було проведено аналіз якості розробленого програмного забезпечення та визначено його оцінку. Також було продумано стратегію тестування програмного продукту та наведено контрольні приклади тестів.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

WEB-застосунок необхідно розмістити на сервері, що має наступні характеристики:

- 32Gb RAM 3d generation;
- Intel Xeon 8ядер, 16 потоків, від 20Мб кешу;
- 500Гб SSD.

Для розгортання програми знадобиться backup бази даних, очищеної від усіх тестових значень. При цьому в базі вже мають зберігатись дані про необхідні закладам шаблони звітності та деревовидна структура ЗОЗ.

Діаграму розгортання застосунку можна побачити на рисунку 4.1.

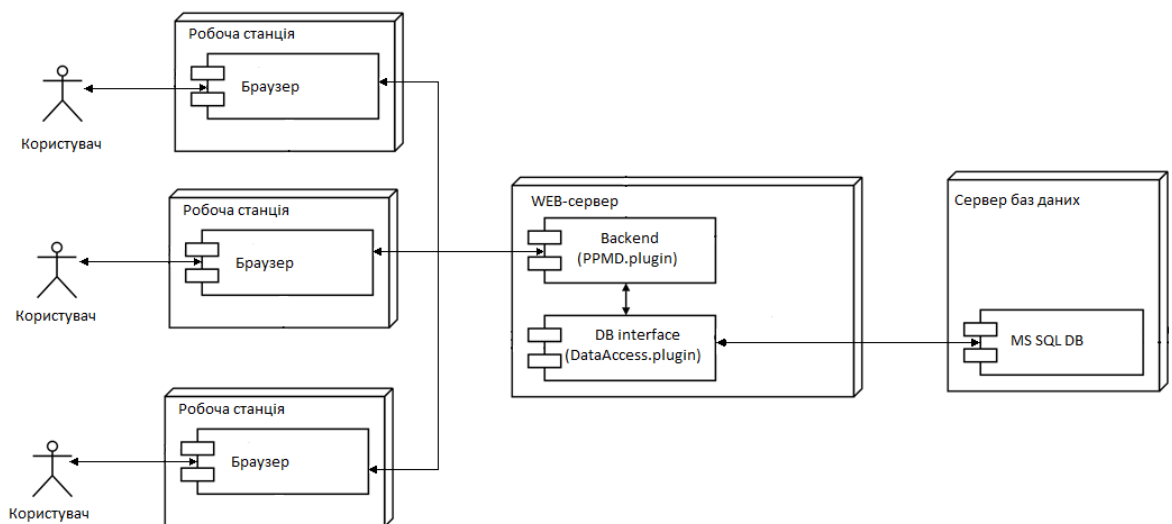


Рисунок 4.1 - Схема структурна розгортання WEB-застосунку

Зазвичай застосування, що використовують ASP.Net MVC, публікують одним з трьох способів:

- сервер IIS;
- сервер nginx;

Змн.	Арк.	№ докум.	Підпис	Дата

- платформа Windows Azure.

Для того щоб розмістити додаток на сервер IIS, потрібно виконати декілька простих кроків [11]:

- відкрити адмін-панель IIS, перейти в Панель управління та знайти Диспетчер служб;
- обрати Default Web Site та натиснути кнопку «Додати додаток»;
- заповнити назву проекту та його розташування;
- перевірити коректність підключення до MS SQL бази даних в web.config файлі проекту;
- відкрити проект у Visual Studio, обрати пункт меню «Опублікувати»;
- у відкритому вікні в підменю Connection обираємо папку проекту в IIS та URL нашого сайту.

Після натискання кнопки опублікувати, залишається лише запустити службу веб-публікацій. Після цього WEB-додаток стане доступним по вказаному URL.

Для робочих станцій клієнтів ніякі додаткові кроки по встановленню програми не потрібні. Достатньо забезпечити працівників сучасними комп'ютерами з доступом до мережі та встановленим браузером.

4.2 Робота з програмним забезпеченням

Для успішного освоєння ППМД всі користувачі повинні:

- мати досвід роботи на робочих станціях під управлінням ОС Microsoft Windows (7, 8 або вище);
- володіти навичками роботи з прикладним програмним забезпеченням, що працює в середовищі Windows;
- володіти знаннями елементів роботи з офісними системами типу Microsoft Office (2010 і вище);

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

- мати навички роботи з браузером Microsoft Internet Explorer версії 9.0 та вище або Google Chrome;
- пройти курс навчання роботи з ППМД або детально вивчити інструкцію користувача.

Інструкція щодо основних сценаріїв роботи з WEB-застосуванням наведена в документі «Керівництво користувача».

В разі виникнення відмов необхідно діяти відповідно до інструкцій, наведених в таблиці 4.1.

Таблиця 4.1 – Опис аварійних ситуацій

Код	Опис	Дії користувача
0301	Неможливість роботи в ППМД, пов'язана із зараженням сервера/робочої станції програмами-вірусами.	Користувачеві необхідно звернутися до системного адміністратора ППМД або в Службу технічної підтримки.
0401	Виникають такі помилки: Відмова з боку Microsoft SQL Server в доступі до бази даних від імені користувача SQL сервера; Помилки при виконанні операцій над даними; Помилки розміщення бази даних; Помилки виконання транзакцій.	Користувачеві необхідно звернутися до системного адміністратора ППМД або в Службу технічної підтримки.

Продовження таблиці 4.1

0402 0403	Не відображається сторінка входу в ППМД	Перевірити наявність постійного підключення до мережі Інтернет. При відсутності доступу до інших серверів (наприклад, до пошукової системи www.google.com.ua), необхідно звернутися до адміністратора мережі для вирішення проблеми підключення комп'ютера до мережі. При наявності доступу до інших сторінок найбільш імовірною причиною відсутності доступу до ППМД є зупинка веб-сервера системи. У цьому користувачеві необхідно звернутися до системного адміністратора ППМД. Перевірити правильність введення в адресний рядок браузера адреси ППМД і повторити введення.
0404	Internet Explorer заблокував завантаження файлів на комп'ютер користувача з веб-вузла ППМД.	Дане повідомлення обумовлено політикою безпеки, що встановлена в Internet Explorer. Для виключення появи цього повідомлення надалі, необхідно прописати адресу веб-сайту ППМД в довірені вузли Internet Explorer. Зробіть це самостійно або зверніться до системного адміністратора.

Змн.	Арк.	№ докум.	Підпис	Дата

4.3 Висновки до розділу

В цьому розділі була побудована схема розгортання програмного забезпечення для клієнтів. Також були сформовані покрокові дії-інструкції щодо виконання розгортання. Для користувачів системи було створено детальну інструкцію, вивчення якої робить користування програмою легким та приємним а також створено список дій в разі виникнення аварійних ситуацій.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

ВИСНОВКИ

Основне призначення дипломного проекту – створення WEB-застосування для ведення електронної системи медичних карток хворих на ВІЛ та ведення статистики, пов'язаної з вживанням заходів проти передачі інфекції від матері до дитини.

Мета розробки – створити програмне забезпечення, що буде зручним та зрозумілим в користуванні, пришвидшить заповнення медичних форм пацієнтів та спростить роботу по аналізу отриманих даних. Крім того, однією з важливих вимог до розроблено програмного забезпечення є захист персональної інформації пацієнтів, бути зручною для подальших змін та розширення (мати інструменти для швидкого створення нових медичних форм та нових шаблонів звітності).

У розділі «Аналіз вимог» проаналізовано популярні рішення перенесення медичних форм в електронний формат. Також було проведено аналіз програм, схожих за своєю функціональністю, з метою визначення основних пунктів, які має реалізовувати подібна система, а також визначенню недоліків та сильних сторін конкурентів з метою її вдосконалення. За допомогою отриманою інформації, а також продуманих сценаріїв роботи з системою, було визначено перелік функціональних та нефункціональних вимог. На виході були отримані чітко сформульовані цілі, яких необхідно досягнути в процесі розробки ПЗ.

У розділі «Моделювання та конструювання» на основі медичних паперових форм та інформації, що очікує отримати замовник, було визначено формат вхідних та вихідних даних. Було прийняте рішення про розробку з застосуванням шаблону MVC на мові програмування С#. В процесі розробки архітектури ПЗ, було визначено модель бази даних та спроектовано схему класів, їх основних методів для реалізації поставлених цілей. Крім того вдалося створити допоміжні інструменти для пришвидшення створення полів медичних карток. Крім того були продумана стратегія забезпечення безпеки персональних даних користувачів системи та пацієнтів. На виході був

					КПІ.ІП-5119.045440.02.81	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

отриманий готовий програмний продукт, що виконує усі покладені на нього функції.

У розділі «Аналіз якості та тестування» створене WEB-застосування було проаналізовано за різними критеріями якості ПЗ. Була продумана стратегія тестування та наведені контрольні приклади тестів основної функціональності програми. За допомогою критеріїв оцінювання та проведення функціональних та нефункціональних тестів було доведено, що даний програмний продукт високої якості.

У розділі «Впровадження та супровід» була побудована схема розгортання ПЗ та наведена покрокова інструкція щодо розгортання WEB-застосування на сервери ІІS. Для користувачів програмного забезпечення створено рекомендації по підготовці персоналу, детальну інструкцію по роботі з усіма розділами системи та інструкції щодо дій в разі виникнення аварійних ситуацій.

Результатом дипломної роботи стало WEB-застосування належної якості для ведення обліку електронних медичних карток та створення звітності. Програмний продукт відповідає усім визначеним вимогам, забезпечує цілісність та безпеку даних, є зручним у користуванні. Його архітектура передбачає можливість розширення списку медичних форм та шаблонів звітності, що робить систему перспективною для довготривалого використання.

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		84

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Медична інформаційна система [Електронний ресурс] — Режим доступу: www.uk.wikipedia.org/wiki/Медична_інформаційна_система
- 2) Наказ: Про затвердження форм первинної облікової документації та звітності з питань моніторингу заходів профілактики передачі ВІЛ від матері до дитини. [Електронний ресурс] — Режим доступу: http://search.ligazakon.ua/l_doc2.nsf/link1/RE21795.html
- 3) Медицинская система Archimed+ [Електронний ресурс] — Режим доступу: <https://archimed.pro/>
- 4) Медична інформаційна система Emcimed [Електронний ресурс] — Режим доступу: <https://www.mcmed.ua/ua/about>
- 5) Система «Доктор Елекс» [Електронний ресурс] — Режим доступу: <https://doctor.eleks.com/descriptions/general/>
- 6) Карл И. Вигерс Разработка требований к программному обеспечению. – М: Издательско-торговый дом «Русская Редакция», 2004 – 576 с.
- 7) Адам Фримен, Стивен Сандерсон ASP.Net MVC 5 фреймворк с примерами на С# 5.0 – М: Издательство Вильямс, 2015 – 736 с.
- 8) Джен Харингтон Проектирование реляционных баз данных – М: Издательство Лори, 2006 – 230 с.
- 9) Введение в ASP.Net Identity [Електронний ресурс] — Режим доступу: <https://docs.microsoft.com/ru-ru/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity>
- 10) Качество программного обеспечения [Електронний ресурс] — Режим доступу: <https://qalight.com.ua/baza-znaniy/kachestvo-programmnogo-obespecheniya/>
- 11) IIS 7.0 краткая инструкция для системного администратора [Електронний ресурс] — Режим доступу: <https://blogs.technet.microsoft.com/iwalker/2008/11/24/iis-7-0-1085-2/>

					КПІ.ІП-5119.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		85

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА
ОЦІНКИ ЕФЕКТИВНОСТІ ПРОФІЛАКТИЧНИХ ЗАХОДІВ ДЛЯ
ЗАПОБІГАННЯ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ

Технічне завдання

КП.ІІ-5119.045440.03.91

“ПОГОДЖЕНО”

Керівник проекту:

_____ К.І. Ліщук

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ С.В. Реутська

Київ – 2019 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК.....	6
4.2	ВИМОГИ ДО НАДІЙНОСТІ.....	7
4.3	УМОВИ ЕКСПЛУАТАЦІЇ.....	7
4.4	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ.....	7
4.5	ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ.....	8
4.6	ВИМОГИ ДО МАРКУВАННЯ ТА ПАКУВАННЯ.....	8
4.7	ВИМОГИ ДО ТРАНСПОРТУВАННЯ ТА ЗБЕРІГАННЯ.....	9
4.8	СПЕЦІАЛЬНІ ВИМОГИ.....	9
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....	10
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	11
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	12

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Web-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини.

Галузь застосування:

Наведене технічне завдання поширюється на розробку WEB-застосування [КПІ.ІП-5119.045440.03.81], котре використовується для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини з метою боротьби зі СНІДОМ шляхом збору інформації про проведені профілактичні заходи, їх аналіз та надання інформації медичним працівникам про їх результати.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки Web-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для отримання достовірних результатів про проведені профілактичні заходи для пацієнток для запобігання передачі ВІЛ від матері до дитини. Призначена для медичних закладів, що є учасниками програми «Профілактика передачі ВІЛ від матері до дитини» в Україні, а також для співробітників МОЗ. Метою розробки є зменшення часу на пошук інформації про ВІЛ-інфікованих пацієнток по всій Україні та його спрощення, за рахунок створення єдиної бази електронних карток ВІЛ-інфікованих пацієнток, ведення їх історій хвороби, а також надання аналітичних матеріалів у вигляді звітів для аналізу результатів, отриманих внаслідок проведення певних профілактичних заходів.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для користувача:

- авторизація користувача в системі;
- ведення електронних карток для матері в межах своєї області та/або району;
- ведення електронних карток дитини в межах своєї області та/або району;
- ведення історії проведених профілактичних досліджень для пацієнтки;
- забезпечення можливості копіювання повторюваних даних з попередніх медичних форм пацієнтки в нові;
- пошук карток матері або дитини за заданими реквізитами;
- формування електронної звітності за заданими параметрами;
- прив'язка картки матері до картки дитини.

4.1.1.2 Для адміністратора системи:

- ведення профілю користувача системи;
- ведення довідників областей, регіонів, міст;
- ведення карток закладів охорони здоров'я;
- перенесення медичної картки матері або дитини до іншого регіону та/або медичного закладу.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

4.1.2 Розробку виконати на платформі Windows 10

4.1.3 Додаткові вимоги:

- користувачу доступні лише ті картки пацієнтів, які відносяться до його області та/або його регіону;
- персональні дані пацієнтів (ФІО, адреса, телефон, дата народження) мають зберігатися в зашифрованому вигляді.

4.2 Вимоги до надійності

- Передбачити контроль введення інформації.
- Передбачити захист від некоректних дій користувача.
- Забезпечити цілісність інформації в базі даних.
- Організувати регулярне створення резервних копій бази даних.

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.2 Обслуговування

Вимоги до обслуговування не пред'являються.

4.3.3 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не пред'являються.

4.4 Вимоги до складу і параметрів технічних засобів

Для правильної роботи програмного забезпечення до складу технічних засобів повинні входити:

а) сервер бази даних з наступною конфігурацією:

- процесор з 2-4 ядрами з тактовою частотою не нижче 2.0 ГГц;
- максимальний об'єм оперативної пам'яті (не менше 4 ГБ);
- жорсткі накопичувачі з швидким виконанням операцій читання і запису, або storage-масив;
- в якості СУБД може бути використана MS SQL 2012;

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

б) сервер додатків з наступною конфігурацією:

- 8-ядерний процесора Intel Xeon з 16 робочими потоками від 20 Мб кешу від 2,1 ГГц;
- 32 Гб оперативної пам`яті DDR3-1600 або DDR4-2400;
- SSD на 512 GB;
- .NET Framework 4.0 або вище;

в) клієнтські робочі місця:

- процесор 1.6 ГГц і вище, 2-4 ядра;
- 4 Гб оперативної пам`яті DDR3-1600 або DDR4-2400;
- Інтернет-з`єднання швидкістю не менше 15Мбіт/с.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства Windows та підтримувати браузері IE, Google Chrome.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:

- електронна картка пацієнтки та дитини з атрибутами відповідно медичних паперових форм № 501-1/о, №501-2/о, № 501-3(I)/о, № 501-3(II)/о, № 501-3(III)/о, № 501-4/о;
- інформація про області та регіони;
- інформація про заклади охорони здоров`я.

4.5.3 Результати повинні бути представлені в наступному форматі:

- історія проведення профілактичних заходів для матері;
- звітні форми у форматі pdf або xls про результати.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред`являються.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 70 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Програма та методика тестування.

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема бази даних

5.4.2 Схема структурна класів програмного забезпечення

5.4.3 Креслення вигляду екранних форм

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	22.02.2019	
2.	Розробка технічного завдання	04.03.2019	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	15.03.2019	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	02.04.2019	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму ...)
5.	Програмна реалізація програмного забезпечення	21.04.2019	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	01.05.2019	Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту	10.05.2019	Пояснювальна записка.
8.	Розробка матеріалів графічної частини проекту	16.05.2019	Графічний матеріал проекту
9.	Оформлення технічної документації проекту	24.05.2019	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5119.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА
ОЦІНКИ ЕФЕКТИВНОСТІ ПРОФІЛАКТИЧНИХ ЗАХОДІВ ДЛЯ
ЗАПОБІГАННЯ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ

Опис програми

КПІ.ІІ-5119.045440.04.13

“ПОГОДЖЕНО”

Керівник проекту:

_____ К.І. Ліщук

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ С.В. Реутська

Київ – 2019 року

Тексти програмного коду

Система ППМД

(Найменування програми (документа))

DVD-R

(Вид носія даних)

30 арк, 13 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

1 КОНТРОЛЕР

```

public class PatientController : Controller
{
    private IPatientService _service;

    public PatientController(IPatientService service)
    {
        _service = service;
    }

    public JsonResult GetArchIds()
    {
        return Json(_service.GetArchIds(), JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetDicValueByCode(decimal? attribute, decimal? localCode)
    {
        decimal attr = attribute != null ? Convert.ToDecimal(attribute) : 0;
        decimal locC = localCode != null ? Convert.ToDecimal(localCode) : 0;
        var result = (attr != 0 && locC != 0) ? _service.GetDicValueByCode(attr, locC) : null;

        return Json(result, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetInfo34003(decimal? dossierId)
    {
        PatientModel m;
        if (dossierId != null && dossierId != 0)
        {
            m = _service.GetInfo34003(dossierId.Value);
        } else
        {
            m = new PatientModel();
        }
        return Json(m, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetCardData(string Identity, decimal? Mis, int? CardType)
    {
        List<CardDataModel> list;
        if (CardType == 0 || (Mis == 0 && Identity == string.Empty))
        {
            list = null;
        } else
        {
            list = _service.GetCardData(Identity, Mis, Convert.ToInt32(CardType));
        }
        return Json(list, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetCurrentZOZ(decimal? deptId)
    {
        if (deptId != null && deptId != 0)
        {
            return Json(_service.GetCurrentZOZ(deptId.Value), JsonRequestBehavior.AllowGet);
        } else
        {
            return Json("", JsonRequestBehavior.AllowGet);
        }
    }

    public JsonResult CheckChilds(decimal dossierId)
    {
        _service.CheckChilds(dossierId);

        return Json(null, JsonRequestBehavior.AllowGet);
    }

    public JsonResult Set34009Info(decimal? dossierId)

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

    {
        if (dossierId != null && dossierId != 0)
        {
            _service.Set34009Info(dossierId.Value);
        }
        return Json(null, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetInfoChild(decimal? dossierId, bool is34007)
    {
        ChildModel m;
        if (dossierId != null && dossierId != 0)
        {
            m = _service.GetInfoChild(dossierId.Value, is34007);
        }
        else
        {
            m = new ChildModel();
        }
        return Json(m, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetInfoFromZOZ(decimal? CardId)
    {
        ZOZInfoModel m;
        if (CardId.HasValue)
            m = _service.GetInfoFromZOZ(CardId.Value);
        else m = new ZOZInfoModel();

        return Json(m, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetInfoDoctors(decimal? CardId)
    {
        Dictionary<string, DoctorModel> result = new Dictionary<string, DoctorModel>();
        if (CardId.HasValue)
        {
            var doctors = _service.GetInfoDoctors(CardId.Value);
            if (doctors != null)
            {
                result = doctors.ToDictionary(x => x.id.ToString("F0"), x => x);
            }
            else return null;
        }
        return Json(result, JsonRequestBehavior.AllowGet);
    }

    public JsonResult LoadChildrens(string Identific, string BirthDate, string Place, string Sex)
    {
        List<ChildCardModel> childs = _service.LoadChildrens(Identific, BirthDate, Place, Sex);
        return Json(childs, JsonRequestBehavior.AllowGet);
    }

    public JsonResult SetLink(decimal motherId, decimal childId, decimal childCardId, decimal? prevId)
    {
        return Json(null, JsonRequestBehavior.AllowGet);
    }

    public JsonResult createChild(decimal motherId, decimal childId)
    {
        var result = _service.createChild(motherId, childId);
        return Json(result.ToString(), JsonRequestBehavior.AllowGet);
    }

    public JsonResult unlinkChild(decimal motherId, decimal childMId, decimal childId)
    {
        _service.unlinkChild(motherId, childMId, childId);
        return Json(null, JsonRequestBehavior.AllowGet);
    }

    public JsonResult GetIdentificator(bool Mother, string Id)
    {
        string id = _service.GetIdentificator(Mother, Convert.ToDecimal(Id.Split(',')[0]));
        return Json(id, JsonRequestBehavior.AllowGet);
    }
}

```

					KPI.IP-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

public JsonResult GetCardById(decimal? DosId)
{
    var card = _service.GetCardById(Convert.ToDecimal(DosId));
    return Json(card, JsonRequestBehavior.AllowGet);
}

public JsonResult GetStatus(decimal? dossierId, bool isMother)
{
    var result = _service.GetStatus(Convert.ToDecimal(dossierId), isMother);
    return Json(result, JsonRequestBehavior.AllowGet);
}

public JsonResult GetMis(decimal? dossierId, bool isMother)
{
    var result = _service.GetMis(Convert.ToDecimal(dossierId), isMother);
    return Json(result, JsonRequestBehavior.AllowGet);
}

public JsonResult GetOSS()
{
    var result = _service.GetOSS();
    return Json(result, JsonRequestBehavior.AllowGet);
}

public JsonResult GetCurrentOSS(decimal dossierId)
{
    var result = _service.GetCurrentOSS(dossierId);
    return Json(result, JsonRequestBehavior.AllowGet);
}

public JsonResult changePlace(decimal dossierId, decimal? CurOSSId, decimal OSSId, string OSSVal,
string comment, decimal sectionId, decimal gridId, decimal sectionDB)
{
    string result = _service.changePlace(dossierId, CurOSSId, OSSId, OSSVal, comment, sectionId,
gridId, sectionDB);
    return Json(result, JsonRequestBehavior.AllowGet);
}

[HttpGet]
public virtual ActionResult AddChildPartial()
{
    return PartialView("34004/AddChildPartial");
}

public JsonResult GetZOZArea(decimal? dossierId)
{
    string m;
    if (dossierId != null && dossierId != 0)
    {
        m = _service.GetZOZArea(dossierId.Value);
    }
    else
    {
        m = string.Empty;
    }
    return Json(m, JsonRequestBehavior.AllowGet);
}

public JsonResult GetParrentIds(decimal dosItemId, decimal gridId, string dosId)
{
    return Json(_service.GetLastId(dosItemId, gridId, dosId), JsonRequestBehavior.AllowGet);
}

public JsonResult sendToArchive(decimal dossierId, decimal sectionDB)
{
    string result = _service.sendToArchive(dossierId, sectionDB);
    return Json(result, JsonRequestBehavior.AllowGet);
}

public JsonResult backSend(decimal dossierId, decimal sectionDB)
{
    string res = _service.backSend(dossierId, sectionDB);
    return Json(res, JsonRequestBehavior.AllowGet);
}

public FileResult DownloadFile(string file_name)

```

					КПІ.ІП-5119.045440.04.13	Арк. 5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        string path = Server.MapPath("~/Modules/PPMD.Plugin/Documents/" + file_name);
        byte[] file = System.IO.File.ReadAllBytes(path);
        if (file != null)
        {
            return File(file, System.Net.Mime.MediaTypeNames.Application.Octet, file_name);
        }
        else
            return null;
    }
}

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

2 ЛОГІКА

```

using CommonDocs.Plugin.Services.Abstract;
using Core.AuthServices.Abstract;
using Core.Entities;
using DataAccess;
using DataAccess.Entities;
using DataAccess.Managers.Abstract;
using DataAccess.Repositories.Abstract;
using DataAccess.Services.Abstract;
using DictionariesUDB.Plugin.Services.Abstract;
using PPMD.Plugin.Entities;
using PPMD.Plugin.Services.Abstract;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Linq.SqlClient;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Configuration;

namespace PPMD.Plugin.Services.Concrete
{
    public class PacientService : IPacientService
    {
        protected IDictionaryValuesCacheManager dictionaryManager;
        protected IDocsService service;
        protected IDossierRepository _repository;
        private IDictionariesUDBService _dicUdbService;
        private IDossierItemManager _dossierItemManager;
        private IDossierAttributeManager _dossierAttributeManager;
        private IDossierService _dossierService;
        private ILinkedDocsService _linkedDocsManager;
        private readonly UserInfo _userInfo;
        protected readonly IUserExtensionDispatcher _userExtension;
        private readonly IKoatuuService _koatuuService;

        public PacientService(IDictionaryValuesCacheManager dictionaryManager, IDocsService service,
            IDossierRepository _repository,
            IDictionariesUDBService _dicUdbService, IDossierItemManager _dossierItemManager,
            IDossierAttributeManager _dossierAttributeManager,
            IDossierService _dossierService, ILinkedDocsService _linkedDocsManager, IUserDataProvider
            userDataProvider, IUserExtensionDispatcher userExtension
            , IKoatuuService koatuuService)
        {
            this.dictionaryManager = dictionaryManager;
            this.service = service;
            this._repository = _repository;
            this._dicUdbService = _dicUdbService;
            this._dossierItemManager = _dossierItemManager;
            this._dossierAttributeManager = _dossierAttributeManager;
            this._dossierService = _dossierService;
            this._linkedDocsManager = _linkedDocsManager;
            this._userInfo = userDataProvider.GetCurrentUserInfo();
            this._userExtension = userExtension;
            this._koatuuService = koatuuService;
        }

        public List<CardDataModel> GetCardData(string Identity, decimal? Mis, int CardType)
        {
            List<CardDataModel> list;
            if (CardType == 34002)
            {
                var qw = (from d in _repository.Table<Dossier_List>()
                    join d1 in _repository.Table<Dossier_List>() on d.DLS_IDP equals d1.DLS_LID

```

					КП.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

join dv in _repository.Table<Dossier_Value>() on d1.DLS_IDP equals dv.DLS_IDP
where d.DLS_DDAT == null && d1.DLS_PID == 34003
select new CardDataModel
{
    Mis = dv.DVL_NUM57 != null ? dv.DVL_NUM57.ToString() : string.Empty,
    CardIdentity = dv.DVL_TEXT36 != null ? dv.DVL_TEXT36.ToString() :
string.Empty,
    CardId = d.DLS_IDP,
    DateFirst = dv.DVL_DATE1 != null ? Convert.ToDateTime(dv.DVL_DATE1).Date
: DateTime.Now,
    Form = dv.DVL_TEXT1 != null ? dv.DVL_TEXT1.ToString() : string.Empty,
    SectionId = d1.DLS_IDP
});
list = qw.ToList();
foreach (var card in list) card.Mis = card.Mis.Split('.')[0].Split(',')[0];
if (Identity != string.Empty) list = list.Where(x => x.CardIdentity ==
Identity).ToList();
if (Mis != null) list = list.Where(x => x.Mis == Mis.ToString()).ToList();
}
else
{
    var qw = (from d in _repository.Table<Dossier_List>()
join d1 in _repository.Table<Dossier_List>() on d.DLS_IDP equals d1.DLS_LID
join dv in _repository.Table<Dossier_Value>() on d1.DLS_IDP equals dv.DLS_IDP
where d.DLS_DDAT == null && d1.DLS_PID == 34006
select new CardDataModel
{
    Mis = dv.DVL_NUM33 != null ? dv.DVL_NUM33.ToString() : string.Empty,
    CardIdentity = dv.DVL_TEXT38 != null ? dv.DVL_TEXT38.ToString() :
string.Empty,
    CardId = d.DLS_IDP,
    DateFirst = dv.DVL_DATE2 != null ? Convert.ToDateTime(dv.DVL_DATE2).Date
: DateTime.Now,
    Form = dv.DVL_TEXT1 != null ? dv.DVL_TEXT1.ToString() : string.Empty,
    SectionId = d1.DLS_IDP
});
list = qw.ToList();
foreach (var card in list) card.Mis = card.Mis.Split('.')[0].Split(',')[0];
if (Identity != string.Empty) list = list.Where(x => x.CardIdentity ==
Identity).ToList();
if (Mis != null) list = list.Where(x => x.Mis == Mis.ToString()).ToList();
}
foreach (var card in list)
{
    var section = card.SectionId;
    var gridId = CardType == 34002 ? 34715 : 34722;
    card.DateFirstString = card.DateFirst.ToString("dd.MM.yyy");
    var currentOSS = (from dtsl in _repository.Table<DossierTreeStaff_Link>()
where card.CardId == dtsl.DLS_IDP
select dtsl.STT_UID).FirstOrDefault();
    card.ZOZ = Convert.ToDecimal((from ts in _repository.Table<Tree_Staff>()
where currentOSS == ts.STT_UID
select ts.STT_LID).FirstOrDefault());
    card.Arch = (from ts in _repository.Table<Tree_Staff>()
where currentOSS == ts.STT_UID
select ts.STT_CODE).FirstOrDefault().ToString();

    var grdAttr =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(card.CardId), section, gridId,
false, 0)).Value;

    foreach (DataRow dr in grdAttr.Rows)
    {
        if (CardType == 34002 && Convert.ToInt32(dr["34848"]) == grdAttr.Rows.Count)
        {
            card.DateFrom = Convert.ToDateTime(dr["34717"]).Date;
            card.DateFromString = Convert.ToDateTime(dr["34717"]).Date.ToString("dd.MM.yyy");
            card.PersonFrom = dr["34718"].ToString();
        }
    }
}

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

```

else if (CardType == 34005 && Convert.ToInt32(dr["34849"]) == grdAttr.Rows.Count)
{
    card.DateFrom = Convert.ToDateTime(dr["34724"]).Date;
    card.DateFromString = Convert.ToDateTime(dr["34724"]).Date.ToString("dd.MM.yyy");
    card.PersonFrom = dr["34725"].ToString();
}
}
}

list = list.Where(x => x.Arch.StartsWith("1")).ToList();

return list;
}

public DictionaryValue GetDicValueByCode(decimal attribute, decimal localCode)
{
    var value = dictionaryManager.GetValues(attribute).Where(x => x.LocalCode ==
localCode.ToString()).FirstOrDefault();
    return value;
}

public object GetCurrentZOZ(decimal deptId)
{
    var ossId = (from dv in _repository.Table<Tree_Staff>()
        where dv.STT_UID == deptId
        select dv.STT_LID).FirstOrDefault();

    var Id = (from dv in _repository.Table<vw_Dossier_Value>()
        join d1 in _repository.Table<Dossier_List>() on dv.DLS_IDP equals d1.DLS_IDP+1
        where dv.DOS_IDP_MAIN == 34000 &&
            dv.DOS_IDP_SECTION == 34001 && dv.DVL_NUM4 == ossId && d1.DLS_DDAT
== null
        select new { Id = dv.DLS_IDP_MAIN, Text = dv.DVL_TEXT1 }).FirstOrDefault();
    return Id;
}

public PacientModel GetInfo34003(decimal dossierId)
{
    DossierData document = service.LoadDoc(dossierId);
    PacientModel m = new PacientModel
    {
        name = document.Sections["34003"].Attributes["34030"].Value != null ?
document.Sections["34003"].Attributes["34030"].Value.ToString() : String.Empty,
        surname = document.Sections["34003"].Attributes["34029"].Value != null ?
document.Sections["34003"].Attributes["34029"].Value.ToString() : String.Empty,
        secName = document.Sections["34003"].Attributes["34031"].Value != null ?
document.Sections["34003"].Attributes["34031"].Value.ToString() : String.Empty,
        date = document.Sections["34003"].Attributes["34032"].Value != null ?
document.Sections["34003"].Attributes["34032"].Value.ToString().Substring(0, 10) : String.Empty,
        place = document.Sections["34003"].Attributes["34036"].Value != null ?
document.Sections["34003"].Attributes["34036"].Value.ToString() : String.Empty,
        placeNow = document.Sections["34003"].Attributes["34062"].Value != null
?document.Sections["34003"].Attributes["34062"].Value.ToString() : String.Empty,
        street = document.Sections["34003"].Attributes["34037"].Value != null ?
document.Sections["34003"].Attributes["34037"].Value.ToString() : String.Empty,
        home = document.Sections["34003"].Attributes["34038"].Value != null ?
document.Sections["34003"].Attributes["34038"].Value.ToString() : String.Empty,
        phone = document.Sections["34003"].Attributes["34039"].Value != null ?
document.Sections["34003"].Attributes["34039"].Value.ToString() : String.Empty,
        mail = document.Sections["34003"].Attributes["34065"].Value != null ?
document.Sections["34003"].Attributes["34065"].Value.ToString() : String.Empty,
        elseContact = document.Sections["34003"].Attributes["34068"].Value != null ?
document.Sections["34003"].Attributes["34068"].Value.ToString() : String.Empty,
        typeOfplace = document.Sections["34003"].Attributes["34040"].Value != null ?
Convert.ToDecimal(document.Sections["34003"].Attributes["34040"].Value) : 0,
        Ukrainian = document.Sections["34003"].Attributes["34034"].Value != null ?
Convert.ToBoolean(document.Sections["34003"].Attributes["34034"].Value) : false,
        elseCountry = document.Sections["34003"].Attributes["34035"].Value != null ?
Convert.ToDecimal(document.Sections["34003"].Attributes["34035"].Value) : 0,
    };
}

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

```

        withoutCountry = document.Sections["34003"].Attributes["34043"].Value != null ?
Convert.ToBoolean(document.Sections["34003"].Attributes["34043"].Value) : false,
        chSurname = document.Sections["34003"].Attributes["34046"].Value != null ?
document.Sections["34003"].Attributes["34046"].Value.ToString() : String.Empty
    };
    return m;
}

public ChildModel GetInfoChild(decimal dossierId, bool is34007)
{
    DossierData document = service.LoadDoc(dossierId);
    ChildModel m;

    if (is34007)
    {
        var dossierM = document.Sections["34006"].Attributes["34462"].Value != null ?
_dossierItemManager.GetDossierItem(Convert.ToDecimal(document.Sections["34006"].Attributes["34462"].Value
), true) : null;
        m = new ChildModel
        {
            name = document.Sections["34006"].Attributes["34247"].Value != null ?
document.Sections["34006"].Attributes["34247"].Value.ToString() : String.Empty,
            surname = document.Sections["34006"].Attributes["34248"].Value != null ?
document.Sections["34006"].Attributes["34248"].Value.ToString() : String.Empty,
            secName = document.Sections["34006"].Attributes["34249"].Value != null ?
document.Sections["34006"].Attributes["34249"].Value.ToString() : String.Empty,
            date = document.Sections["34006"].Attributes["34250"].Value != null ?
document.Sections["34006"].Attributes["34250"].Value.ToString().Substring(0, 10) : String.Empty,
            place = document.Sections["34006"].Attributes["34252"].Value != null ?
document.Sections["34006"].Attributes["34252"].Value.ToString() : String.Empty,
            placeNow = document.Sections["34006"].Attributes["34258"].Value != null ?
document.Sections["34006"].Attributes["34258"].Value.ToString() : String.Empty,
            street = document.Sections["34006"].Attributes["34253"].Value != null ?
document.Sections["34006"].Attributes["34253"].Value.ToString() : String.Empty,
            home = document.Sections["34006"].Attributes["34254"].Value != null ?
document.Sections["34006"].Attributes["34254"].Value.ToString() : String.Empty,
            phone = document.Sections["34006"].Attributes["34255"].Value != null ?
document.Sections["34006"].Attributes["34255"].Value.ToString() : String.Empty,
            mail = document.Sections["34006"].Attributes["34282"].Value != null ?
document.Sections["34006"].Attributes["34282"].Value.ToString() : String.Empty,
            motherName = document.Sections["34006"].Attributes["34260"].Value != null ?
document.Sections["34006"].Attributes["34260"].Value.ToString() : String.Empty,
            motherSurname = document.Sections["34006"].Attributes["34261"].Value != null ?
document.Sections["34006"].Attributes["34261"].Value.ToString() : String.Empty,
            motherSecName = document.Sections["34006"].Attributes["34262"].Value != null ?
document.Sections["34006"].Attributes["34262"].Value.ToString() : String.Empty,
            elseContact = document.Sections["34006"].Attributes["34256"].Value != null ?
document.Sections["34006"].Attributes["34256"].Value.ToString() : String.Empty
        };
        if (dossierM != null)
        {
            var grdAttr =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(document.Sections["34006"].Attribute
s["34462"].Value), dossierM.Sections["34004"].ItemId, 34190, false, 0)).Value;
            foreach (DataRow dr in grdAttr.Rows)
            {
                if (!dr.IsNull("34220") &&
Convert.ToDecimal(dr["34220"].ToString().Split('.')[0].Split(',')[0]) == dossierId)
                {
                    m.dataPlr = dr["34197"] != null ? dr["34197"].ToString().Split(' ')[0] :
String.Empty;
                    m.numPlr = dr["34198"] != null ?
dr["34198"].ToString().Split(',')[0].Split('.')[0] : String.Empty;
                    m.dataResPlr = dr["34739"] != null ? dr["34739"].ToString().Split(' ')[0] :
String.Empty;
                }
            }
        }
    }
}

```

					КП.ІП-5119.045440.04.13	Арк.
ЗМН.	Арк.	№ докум.	Підпис	Дата		10

```

        m.resPlr = dr["34199"] != null ?
dr["34199"].ToString().Split(',').[0].Split('.')[0] : String.Empty;
    }
}
}
else
{
    m = new ChildModel
    {
        name = document.Sections["34007"].Attributes["34324"].Value != null ?
document.Sections["34007"].Attributes["34324"].Value.ToString() : String.Empty,
        surname = document.Sections["34007"].Attributes["34323"].Value != null ?
document.Sections["34007"].Attributes["34323"].Value.ToString() : String.Empty,
        secName = document.Sections["34007"].Attributes["34325"].Value != null ?
document.Sections["34007"].Attributes["34325"].Value.ToString() : String.Empty,
        date = document.Sections["34007"].Attributes["34326"].Value != null ?
document.Sections["34007"].Attributes["34326"].Value.ToString().Substring(0, 10) : String.Empty,
        place = document.Sections["34007"].Attributes["34327"].Value != null ?
document.Sections["34007"].Attributes["34327"].Value.ToString() : String.Empty,
        placeNow = document.Sections["34007"].Attributes["34335"].Value != null ?
document.Sections["34007"].Attributes["34335"].Value.ToString() : String.Empty,
        street = document.Sections["34007"].Attributes["34328"].Value != null ?
document.Sections["34007"].Attributes["34328"].Value.ToString() : String.Empty,
        home = document.Sections["34007"].Attributes["34329"].Value != null ?
document.Sections["34007"].Attributes["34329"].Value.ToString() : String.Empty,
        phone = document.Sections["34007"].Attributes["34330"].Value != null ?
document.Sections["34007"].Attributes["34330"].Value.ToString() : String.Empty,
        mail = document.Sections["34007"].Attributes["34337"].Value != null ?
document.Sections["34007"].Attributes["34337"].Value.ToString() : String.Empty,
        motherName = document.Sections["34007"].Attributes["34332"].Value != null ?
document.Sections["34007"].Attributes["34332"].Value.ToString() : String.Empty,
        motherSurname = document.Sections["34007"].Attributes["34331"].Value != null ?
document.Sections["34007"].Attributes["34331"].Value.ToString() : String.Empty,
        motherSecName = document.Sections["34007"].Attributes["34333"].Value != null ?
document.Sections["34007"].Attributes["34333"].Value.ToString() : String.Empty,
        elseContact = document.Sections["34007"].Attributes["34339"].Value != null ?
document.Sections["34007"].Attributes["34339"].Value.ToString() : String.Empty
    };
}
return m;
}

public void Set34009Info(decimal dossierId)
{
    DossierData documentChild = service.LoadDoc(dossierId);
    var dossier = _dossierItemManager.GetDossierItem(Convert.ToDecimal(dossierId), true);
    var dossierM = dossier.Sections["34006"].Attributes["34462"].Value != null ?
_dossierItemManager.GetDossierItem(Convert.ToDecimal(dossier.Sections["34006"].Attributes["34462"].Value)
, true) : null;

    #region set values
    dossier.Sections["34009"].Attributes["34478"].Value =
dossier.Sections["34006"].Attributes["34284"].Value;
    dossier.Sections["34009"].Attributes["34479"].Value =
dossier.Sections["34008"].Attributes["34419"].Value;
    dossier.Sections["34009"].Attributes["34480"].Value =
dossier.Sections["34006"].Attributes["34250"].Value;
    dossier.Sections["34009"].Attributes["34481"].Value =
dossier.Sections["34006"].Attributes["34251"].Value;
    dossier.Sections["34009"].Attributes["34482"].Value =
dossier.Sections["34008"].Attributes["34386"].Value;
    dossier.Sections["34009"].Attributes["34483"].Value =
dossier.Sections["34008"].Attributes["34387"].Value;

    dossier.Sections["34009"].Attributes["34484"].Value =
dossier.Sections["34008"].Attributes["34433"].Value;

```

					КП.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11


```

dossier.Sections["34009"].Attributes["34819"].Value =
dossier.Sections["34007"].Attributes["34771"].Value;
dossier.Sections["34009"].Attributes["34820"].Value =
dossier.Sections["34007"].Attributes["34340"].Value;
dossier.Sections["34009"].Attributes["34821"].Value =
dossier.Sections["34007"].Attributes["34342"].Value;
dossier.Sections["34009"].Attributes["34822"].Value =
dossier.Sections["34007"].Attributes["34343"].Value;

dossier.Sections["34009"].Attributes["34636"].Value =
dossier.Sections["34008"].Attributes["34406"].Value;
dossier.Sections["34009"].Attributes["34813"].Value =
dossier.Sections["34008"].Attributes["34407"].Value;
dossier.Sections["34009"].Attributes["34823"].Value =
dossier.Sections["34008"].Attributes["34835"].Value;
dossier.Sections["34009"].Attributes["34639"].Value =
dossier.Sections["34008"].Attributes["34409"].Value;
dossier.Sections["34009"].Attributes["34818"].Value =
dossier.Sections["34008"].Attributes["34410"].Value;
dossier.Sections["34009"].Attributes["34824"].Value =
dossier.Sections["34008"].Attributes["34836"].Value;
dossier.Sections["34009"].Attributes["34648"].Value =
dossier.Sections["34008"].Attributes["34418"].Value;
dossier.Sections["34009"].Attributes["34649"].Value =
dossier.Sections["34008"].Attributes["34420"].Value;
dossier.Sections["34009"].Attributes["34825"].Value =
dossier.Sections["34008"].Attributes["34784"].Value;
dossier.Sections["34009"].Attributes["34658"].Value =
dossier.Sections["34008"].Attributes["34429"].Value;
dossier.Sections["34009"].Attributes["34660"].Value =
dossier.Sections["34007"].Attributes["34344"].Value;
dossier.Sections["34009"].Attributes["34661"].Value =
dossier.Sections["34007"].Attributes["34345"].Value;
#endregion

if (dossierM != null)
{
#region set other values
dossier.Sections["34009"].Attributes["34491"].Value =
dossierM.Sections["34003"].Attributes["34032"].Value;
dossier.Sections["34009"].Attributes["34492"].Value =
dossierM.Sections["34003"].Attributes["34034"].Value;
dossier.Sections["34009"].Attributes["34493"].Value =
dossierM.Sections["34003"].Attributes["34035"].Value;
dossier.Sections["34009"].Attributes["34791"].Value =
dossierM.Sections["34003"].Attributes["34043"].Value;
dossier.Sections["34009"].Attributes["34494"].Value =
dossierM.Sections["34003"].Attributes["34040"].Value;

dossier.Sections["34009"].Attributes["34498"].Value =
dossierM.Sections["34003"].Attributes["34041"].Value;
dossier.Sections["34009"].Attributes["34499"].Value =
dossierM.Sections["34003"].Attributes["34042"].Value;
dossier.Sections["34009"].Attributes["34854"].Value =
dossierM.Sections["34003"].Attributes["34055"].Value;
dossier.Sections["34009"].Attributes["34855"].Value =
dossierM.Sections["34003"].Attributes["34056"].Value;
dossier.Sections["34009"].Attributes["34501"].Value =
dossierM.Sections["34003"].Attributes["34047"].Value;
dossier.Sections["34009"].Attributes["34502"].Value =
dossierM.Sections["34003"].Attributes["34048"].Value;
dossier.Sections["34009"].Attributes["34503"].Value =
dossierM.Sections["34003"].Attributes["34049"].Value;
dossier.Sections["34009"].Attributes["34504"].Value =
dossierM.Sections["34003"].Attributes["34050"].Value;
dossier.Sections["34009"].Attributes["34788"].Value =
dossierM.Sections["34003"].Attributes["34051"].Value;
dossier.Sections["34009"].Attributes["34506"].Value =
dossierM.Sections["34003"].Attributes["34052"].Value;

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13


```

dossier.Sections["34009"].Attributes["34581"].Value =
dossierM.Sections["34004"].Attributes["34168"].Value;
dossier.Sections["34009"].Attributes["34803"].Value =
dossierM.Sections["34004"].Attributes["34144"].Value;

dossier.Sections["34009"].Attributes["34587"].Value =
dossierM.Sections["34004"].Attributes["34152"].Value;
dossier.Sections["34009"].Attributes["34588"].Value =
dossierM.Sections["34004"].Attributes["34153"].Value;
dossier.Sections["34009"].Attributes["34804"].Value =
dossierM.Sections["34004"].Attributes["34137"].Value;
dossier.Sections["34009"].Attributes["34805"].Value =
dossierM.Sections["34004"].Attributes["34138"].Value;
dossier.Sections["34009"].Attributes["34589"].Value =
dossierM.Sections["34004"].Attributes["34176"].Value;
dossier.Sections["34009"].Attributes["34590"].Value =
dossierM.Sections["34004"].Attributes["34177"].Value;
dossier.Sections["34009"].Attributes["34591"].Value =
dossierM.Sections["34004"].Attributes["34178"].Value;
dossier.Sections["34009"].Attributes["34592"].Value =
dossierM.Sections["34004"].Attributes["34179"].Value;
dossier.Sections["34009"].Attributes["34789"].Value =
dossierM.Sections["34004"].Attributes["34181"].Value;
dossier.Sections["34009"].Attributes["34595"].Value =
dossierM.Sections["34004"].Attributes["34182"].Value;
dossier.Sections["34009"].Attributes["34596"].Value =
dossierM.Sections["34004"].Attributes["34183"].Value;
#endregion

#region set attribute 34832
var grdAttr =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34004"].ItemId, 34190, false, 0)).Value;
var grdAttrARV = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34832, false, 0)).Value;

foreach (DataRow dr in grdAttr.Rows)
{
    if (!dr.IsNull("34220") &&
Convert.ToDecimal(dr["34220"].ToString().Split('.')[0].Split(',')[0]) == dossierId)
    {
        dossier.Sections["34009"].Attributes["34597"].Value = dr["34193"];
        dossier.Sections["34009"].Attributes["34598"].Value = dr["34194"];
        dossier.Sections["34009"].Attributes["34600"].Value = dr["34196"];
        dossier.Sections["34009"].Attributes["34628"].Value = dr["34212"];
        dossier.Sections["34009"].Attributes["34627"].Value = dr["34211"];
        dossier.Sections["34009"].Attributes["34806"].Value = dr["34201"];

        var gridARV =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34004"].ItemId, 34757, false, 0)).Value;
        var childARV = gridARV.AsEnumerable().Where(x => Convert.ToDecimal(x["PId"]) ==
Convert.ToDecimal(dr["Id"])).AsEnumerable();
        grdAttrARV.Clear();

        int rId = 0;
        foreach (var arv in childARV)
        {
            DataRow grow = grdAttrARV.NewRow();
            grow["Id"] = rId > 0 ? -1 : --rId;
            grow["34833"] = arv["34758"];
            grow["34834"] = arv["34759"];

            grdAttrARV.Rows.Add(grow);
        }
        dossier.Sections["34009"].Attributes["34832"].Value = grdAttrARV;
    }
}
#endregion

```

					KPI.IП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

```

#region set attribute 34837
var grdAttrOld =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34003"].ItemId, 34730, false, 0)).Value;
var grdAttrNew = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34837, false, 0)).Value;

grdAttrNew.Clear();
int rowId = 0;

foreach (DataRow dr in grdAttrOld.Rows)
{
    DataRow grow = grdAttrNew.NewRow();
    grow["Id"] = rowId > 0 ? -1 : --rowId;
    grow["34838"] = dr["34731"];
    grow["34839"] = dr["34732"];
    grow["34840"] = dr["34733"];
    grow["34841"] = dr["34734"];
    grow["34842"] = dr["34735"];
    grow["34843"] = dr["34736"];
    grow["34844"] = dr["34737"];
    grow["34845"] = dr["34738"];
    grow["34846"] = dr["34762"];
    grow["34847"] = dr["34763"];

    grdAttrNew.Rows.Add(grow);
}
dossier.Sections["34009"].Attributes["34837"].Value = grdAttrNew;
#endregion

#region set attribute 34864
var grdAttrOldHv =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34004"].ItemId, 34862, false, 0)).Value;
var grdAttrNewHv = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34864, false, 0)).Value;

grdAttrNewHv.Clear();
rowId = 0;

foreach (DataRow dr in grdAttrOldHv.Rows)
{
    DataRow grow = grdAttrNewHv.NewRow();
    grow["Id"] = rowId > 0 ? -1 : --rowId;
    grow["34865"] = dr["34863"];
    grdAttrNewHv.Rows.Add(grow);
}
dossier.Sections["34009"].Attributes["34864"].Value = grdAttrNewHv;
#endregion

#region set attribute 34826
var grdAttrOld3 =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34004"].ItemId, 34751, false, 0)).Value;
var grdAttrNew3 = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34826, false, 0)).Value;

grdAttrNew3.Clear();
rowId = 0;

foreach (DataRow dr in grdAttrOld3.Rows)
{
    DataRow grow = grdAttrNew3.NewRow();
    grow["Id"] = rowId > 0 ? -1 : --rowId;
    grow["34827"] = dr["34752"];
    grow["34828"] = dr["34753"];
    grow["34829"] = dr["34754"];

    grdAttrNew3.Rows.Add(grow);
}

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

```

dossier.Sections["34009"].Attributes["34826"].Value = grdAttrNew3;

// save parent rows for next table
_repository.BeginTransaction();
try
{
    _dossierService.SaveDossierUncommitted(_repository, dossier, new decimal[] { 34009 });
    _repository.Commit();
}
catch (Exception e)
{
    _repository.Rollback();
    throw e;
}
#endregion

#region set attribute 34830
var grdAttrOld3Ch =
(DataTable)(_dossierAttributeManager.LoadAttribute(Convert.ToDecimal(dossier.Sections["34006"].Attributes
["34462"].Value), dossierM.Sections["34004"].ItemId, 34755, false, 0)).Value;
var grdAttrNew3Ch = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34830, false, 0)).Value;

grdAttrNew3Ch.Clear();

rowId = 0;
foreach (DataRow dr in grdAttrNew3.Rows)
{
    var OId = grdAttrOld3.AsEnumerable().Where(x => Convert.ToDecimal(x["34752"]) ==
Convert.ToDecimal(dr["34827"]) && Convert.ToString(x["34753"]) == Convert.ToString(dr["34828"])) /*&&
Convert.ToDecimal(x["34754"]) == Convert.ToDecimal(dr["34829"])*).FirstOrDefault()["Id"];
    foreach (DataRow ch in grdAttrOld3Ch.Rows)
    {
        if (Convert.ToDecimal(ch["PId"]) == Convert.ToDecimal(OId))
        {
            DataRow grow = grdAttrNew3Ch.NewRow();
            grow["Id"] = rowId > 0 ? -1 : --rowId;
            grow["PId"] = dr["Id"];
            grow["34831"] = ch["34756"];
            grdAttrNew3Ch.Rows.Add(grow);
        }
    }
}
dossier.Sections["34009"].Attributes["34830"].Value = grdAttrNew3Ch;
#endregion
}

if (dossier.Sections["34008"].ItemId != 0)
{
    #region set attribute 34856
    var grdAttrOld2 = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34008"].ItemId, 34786, false, 0)).Value;
    var grdAttrNew2 = (DataTable)(_dossierAttributeManager.LoadAttribute(dossierId,
dossier.Sections["34009"].ItemId, 34856, false, 0)).Value;

    grdAttrNew2.Clear();
    int rowId = 0;

    foreach (DataRow dr in grdAttrOld2.Rows)
    {
        DataRow grow = grdAttrNew2.NewRow();
        grow["Id"] = rowId > 0 ? -1 : --rowId;
        grow["34857"] = dr["34787"];

        grdAttrNew2.Rows.Add(grow);
    }
    dossier.Sections["34009"].Attributes["34856"].Value = grdAttrNew2;
    #endregion
}

```

					КП.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

```

        _repository.BeginTransaction();
        try
        {
            _dossierService.SaveDossierUncommitted(_repository, dossier, new decimal[] { 34009 });
            _repository.Commit();
        }
        catch
        {
            _repository.Rollback();
            throw;
        }
        _dossierItemManager.DeleteDossierFromChache(dossierId);
    }

    public ZOZInfoModel GetInfoFromZOZ(decimal CardId)
    {
        DossierData document = service.LoadDoc(CardId);

        ZOZInfoModel m = new ZOZInfoModel
        {
            name = document.Sections["34001"].Attributes["34232"].Value != null ?
            document.Sections["34001"].Attributes["34232"].Value.ToString() : String.Empty,
            index = document.Sections["34001"].Attributes["34015"].Value != null ?
            document.Sections["34001"].Attributes["34015"].Value.ToString() : String.Empty,
            town = document.Sections["34001"].Attributes["34003"].Value != null ?
            document.Sections["34001"].Attributes["34003"].Value.ToString() : String.Empty,
            adress = document.Sections["34001"].Attributes["34004"].Value != null ?
            document.Sections["34001"].Attributes["34004"].Value.ToString() : String.Empty,
            code = document.Sections["34001"].Attributes["34005"].Value != null ?
            document.Sections["34001"].Attributes["34005"].Value.ToString() : String.Empty
        };
        return m;
    }

    public string ControllEditSignals(decimal? id, int CloseEditTimeout)
    {
        if (id != 0 && id != null)
        {
            Dossier_List model = _repository.Table<Dossier_List>().FirstOrDefault(x => x.DLS_IDP ==
            id);

            if (model != null)
            {
                if (model.DLS_EDIT == 1)
                {
                    if (DateTime.Now.AddMinutes(-CloseEditTimeout) > model.DLS_EDAT.Value)
                    {

                        _repository.BeginTransaction();
                        try
                        {
                            model.DLS_NEDI = _userInfo.SurName;
                            model.DLS_EDAT = DateTime.Now;
                            model.DLS_EDIT = 1;

                            _repository.SubmitChange();
                            _repository.Commit();
                        }
                        catch (Exception e)
                        {
                            _repository.Rollback();
                            throw;
                        }
                    }
                    return "";
                }
                else
                {
                    if (DateTime.Now.AddMinutes(-1) > model.DLS_EDAT.Value)
                    {

```

					КП.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

3 ПРЕДСТАВЛЕННЯ

```

@model DataAccess.Entities.DossierSection
@using CommonDocs.Plugin.Html.Entities
@using CommonDocs.Plugin.Html.Helpers
@using PPMD.Plugin.Resources

<style>
    .ui-accordion {
        width: 900px;
    }

    #btnTreewrapper {
        display: none;
    }
    #gridBtns34715.top-space{ margin-top: 0px; }
</style>
<input type="hidden" id="34101" name="34101" value='@Model.Attributes["34101"].Value' />
<input type="hidden" id="34024" name="34024" value='@Model.Attributes["34024"].Value' />
<input type="hidden" id="34108" name="34108" value='@Model.Attributes["34108"].Value' />
<input type="hidden" id="34098" name="34098" value='@Model.Attributes["34098"].Value' />
<input type="hidden" id="34304" name="34304" value='@Model.Attributes["34304"].Value' />
<input type="hidden" id="34682" name="34682" value='@Model.Attributes["34682"].Value' />
<input type="hidden" id="34685" name="34685" value='@Model.Attributes["34685"].Value' />
<input type="hidden" id="34873" name="34873" value='@Model.Attributes["34873"].Value' />
<input type="hidden" id="34874" name="34874" value='@Model.Attributes["34874"].Value' />
<input type="hidden" id="34868" name="34868" value='@Model.Attributes["34868"].Value' />
<div id="Title1"></div>

<div>
    @*<div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34024"])
        @Html.CommonDocs().Dictionary(Model.Attributes["34024"], dp => { d }, new { @style = "width:
500px" })
    </div>*@
    <div class="left r-space">
        <div class="caption">@PPMDResources.DocNumber:</div>
        @Html.CommonDocs().Text(Model.Attributes["34025"], new { @style = "width: 50px; margin-
right:10px;", @class = "left" })
        @Html.CommonDocs().Float(Model.Attributes["34026"], new { @style = "width: 50px; margin-
right:10px;", @class = "left" })
        @Html.CommonDocs().Text(Model.Attributes["34027"], new { @style = "width: 50px", @class =
"left" })
    </div>
    <div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34028"])
        @Html.CommonDocs().Date(Model.Attributes["34028"], new { @class = "34028_v" })
    </div>
    <div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34681"])
        @Html.CommonDocs().Text(Model.Attributes["34681"], new { @style = "width: 170px", @readonly =
true, @class = "left" })
        <input type="button" id="formIdentity" onclick="formIdentity1()" disabled="disabled"
class="btn btn-gnrt-id left l-space-btn" />
        <div class="clear"></div>
    </div>
    <div class="left">
        @Html.CommonDocs().Caption(Model.Attributes["34871"])
        @Html.CommonDocs().Text(Model.Attributes["34871"], new { @style = "width: 150px" })
    </div>
    <div class="right">
        @Html.CommonDocs().Caption(Model.Attributes["34703"])
        @Html.CommonDocs().Int(Model.Attributes["34703"], new { @style = "width: 150px" })
    </div>
    <div class="clear"></div>

```

					КП.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

```

    @*@Html.CommonDocs().Validator(Model.Attributes["34024"])*@
    @Html.CommonDocs().Validator(Model.Attributes["34025"])
    @Html.CommonDocs().Validator(Model.Attributes["34026"])
    @Html.CommonDocs().Validator(Model.Attributes["34027"])
    @Html.CommonDocs().Validator(Model.Attributes["34028"])
    @Html.CommonDocs().Validator(Model.Attributes["34703"])
    @Html.CommonDocs().Validator(Model.Attributes["34871"])
    @Html.CommonDocs().Validator(Model.Attributes["34681"])
</div>
<div class="group w100 top-space clean-bg">
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34109"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34109"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 204px" })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34705"])
            @Html.CommonDocs().Text(Model.Attributes["34705"], new { @style = "width: 204px", @class
= "left" })
            <input type="Button" id="ArchiveBtn" title="@PPMDResources.MoveToArchive"
onclick="sendToArchive()" class="btn btn-to-archive left l-space-btn" />
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34109"])
        @Html.CommonDocs().Validator(Model.Attributes["34705"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34706"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34706"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 204px" })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34707"])
            @Html.CommonDocs().Text(Model.Attributes["34707"], new { @style = "width: 204px" })
        </div>
        <div class="left r-space">
            <div class="caption">@PPMDResources.CaptionForStatus501_2:</div>
            <input id="status501_2" type="text" style="width: 204px" disabled />
        </div>
        <div class="left">
            <div class="caption">@PPMDResources.ReasonForInvalidity:</div>
            <input id="status501_2reason" type="text" style="width: 206px" disabled />
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34706"])
        @Html.CommonDocs().Validator(Model.Attributes["34707"])
    </div>
</div>
<div id="accordion_0" class="top-space accordion-padding-top-clear overflow-hidden">
    <h3>@PPMDResources.F5011AccordionTitle0</h3>
    <div>
        <div>
            <div class="left r-space">
                @Html.CommonDocs().Caption(Model.Attributes["34308"])
                @Html.CommonDocs().Dictionary(Model.Attributes["34308"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 200px" })
            </div>
            <div id="forElse" class="left">
                @Html.CommonDocs().Caption(Model.Attributes["34016"])
                @Html.CommonDocs().Dossier(Model.Attributes["34016"], dp => { dp.OutputAttribute =
34000; dp.AutoComplete.Enable = true; dp.AutoComplete.Length = 2; }, new { @style = "width: 639px" })
            </div>
            <div id="forZ0Z" class="left">
                @Html.CommonDocs().Caption(Model.Attributes["34017"])
                @Html.CommonDocs().Text(Model.Attributes["34017"], new { @style = "width: 658px" })
            </div>
            <div class="clear"></div>
            @Html.CommonDocs().Validator(Model.Attributes["34308"])
        </div>
    </div>
</div>

```

					KPI.IP-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

```

        @Html.CommonDocs().Validator(Model.Attributes["34016"])
        @Html.CommonDocs().Validator(Model.Attributes["34017"])
    </div>
    <div>
        @Html.CommonDocs().Caption(Model.Attributes["34298"])
        @Html.CommonDocs().Text(Model.Attributes["34298"], new { @readonly = "readonly", @class =
"w100" })
        @Html.CommonDocs().Validator(Model.Attributes["34298"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34106"])
            @Html.CommonDocs().Int(Model.Attributes["34106"], new { @readonly = "readonly",
@style = "width: 110px" })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34018"])
            @Html.CommonDocs().Text(Model.Attributes["34018"], new { @readonly = "readonly",
@style = "width: 350px" })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34019"])
            @Html.CommonDocs().Text(Model.Attributes["34019"], new { @readonly = "readonly",
@style = "width: 268px" })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34020"])
            @Html.CommonDocs().Text(Model.Attributes["34020"], new { @readonly = "readonly",
@style = "width: 90px" })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34106"])
        @Html.CommonDocs().Validator(Model.Attributes["34018"])
        @Html.CommonDocs().Validator(Model.Attributes["34019"])
        @Html.CommonDocs().Validator(Model.Attributes["34020"])
    </div>
</div>
</div>
<div id="accordion_1" class="top-space accordion-padding-top-clear overflow-hidden">
    <h3>@PPMDResources.F5011AccordionTitle1</h3>
    <div>
        <div>
            <div class="left r-space">
                @Html.CommonDocs().Caption(Model.Attributes["34029"])
                @Html.CommonDocs().Text(Model.Attributes["34029"], new { @style = "width: 205px" })
            </div>
            <div class="left r-space">
                @Html.CommonDocs().Caption(Model.Attributes["34030"])
                @Html.CommonDocs().Text(Model.Attributes["34030"], new { @style = "width: 205px" })
            </div>
            <div class="left r-space">
                @Html.CommonDocs().Caption(Model.Attributes["34031"])
                @Html.CommonDocs().Text(Model.Attributes["34031"], new { @style = "width: 205px" })
            </div>
            <div class="left">
                @Html.CommonDocs().Caption(Model.Attributes["34046"])
                @Html.CommonDocs().Text(Model.Attributes["34046"], new { @style = "width: 203px" })
            </div>
            <div class="clear"></div>
            @Html.CommonDocs().Validator(Model.Attributes["34029"])
            @Html.CommonDocs().Validator(Model.Attributes["34030"])
            @Html.CommonDocs().Validator(Model.Attributes["34031"])
            @Html.CommonDocs().Validator(Model.Attributes["34046"])
        </div>
        <div>
            <div class="left r-space" style="width: 120px">
                @Html.CommonDocs().Caption(Model.Attributes["34032"])
                @Html.CommonDocs().Date(Model.Attributes["34032"])
            </div>

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

```

<div class="left r-space" style="width: 141px">
    @Html.CommonDocs().Caption(Model.Attributes["34033"])
    @Html.CommonDocs().Int(Model.Attributes["34033"], new { @style = "width: 40px" })
</div>
<div class="left r-space" style="margin-top: 21px; width: 155px">
    <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34034"])</div>
    @Html.CommonDocs().CheckBox(Model.Attributes["34034"], new { @class = "left", @style
= "margin: 9px 0px 0 5px" })
</div>
<div class="left r-space">
    @Html.CommonDocs().Caption(Model.Attributes["34035"])
    @Html.CommonDocs().Dictionary(Model.Attributes["34035"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 200px" })
</div>

<div class="left" style="margin-top: 21px; width: 170px">
    <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34043"])</div>
    @Html.CommonDocs().CheckBox(Model.Attributes["34043"], new { @class = "left", @style
= "margin: 9px 0px 0 5px" })
</div>
<div class="clear"></div>
@Html.CommonDocs().Validator(Model.Attributes["34032"])
@Html.CommonDocs().Validator(Model.Attributes["34033"])
@Html.CommonDocs().Validator(Model.Attributes["34035"])
</div>
<div class="caption bold">@PPMDResources.FullPostAddress</div>
<div class="group w100 clean-bg">
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34036"])
            @Html.CommonDocs().Text(Model.Attributes["34036"], new { @style = "width: 280px"
        })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34037"])
            @Html.CommonDocs().Text(Model.Attributes["34037"], new { @style = "width: 411px"
        })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34038"])
            @Html.CommonDocs().Text(Model.Attributes["34038"], new { @style = "width: 125px"
        })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34036"])
        @Html.CommonDocs().Validator(Model.Attributes["34037"])
        @Html.CommonDocs().Validator(Model.Attributes["34038"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34062"])
            @Html.CommonDocs().Text(Model.Attributes["34062"], new { @style = "width: 280px"
        })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34039"])
            @Html.CommonDocs().Text(Model.Attributes["34039"], new { @style = "width: 200px"
        })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34065"])
            @Html.CommonDocs().Text(Model.Attributes["34065"], new { @style = "width: 336px"
        })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34062"])
        @Html.CommonDocs().Validator(Model.Attributes["34039"])
        @Html.CommonDocs().Validator(Model.Attributes["34065"])
    </div>
</div>
<div>
</div>

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

```

    <div class="left r-space">
      @Html.CommonDocs().Caption(Model.Attributes["34068"])
      @Html.CommonDocs().Text(Model.Attributes["34068"], new { @style = "width: 280px"
    })
  </div>
  <div class="left">
    @Html.CommonDocs().Caption(Model.Attributes["34040"])
    @Html.CommonDocs().Dictionary(Model.Attributes["34040"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 200px" })
  </div>
  <div class="clear"></div>
  @Html.CommonDocs().Validator(Model.Attributes["34068"])
  @Html.CommonDocs().Validator(Model.Attributes["34040"])
</div>
</div>
</div>
</div>
<div id="accordion_2" class="top-space accordion-padding-top-clear overflow-hidden">
  <h3>@PPMDResources.F5011AccordionTitle11</h3>
  <div>
    <div class="caption bold">@PPMDResources.F5011GroupTitle4</div>
    <div class="group clean-bg">
      <div>
        <div class="left r-space">
          @Html.CommonDocs().Caption(Model.Attributes["34044"])
          @Html.CommonDocs().Text(Model.Attributes["34044"], new { @style = "width: 418px"
        })
      </div>
      <div class="left">
        @Html.CommonDocs().Caption(Model.Attributes["34045"])
        @Html.CommonDocs().Text(Model.Attributes["34045"], new { @style = "width: 418px"
      })
    </div>
    <div class="clear"></div>
    @Html.CommonDocs().Validator(Model.Attributes["34044"])
    @Html.CommonDocs().Validator(Model.Attributes["34045"])
  </div>
  <div>
    <div class="left r-space">
      <div class="caption">@PPMDResources.PregnancyTerm:</div>
      <div class="group clean-bg">
        <div class="left r-space">
          @Html.CommonDocs().Caption(Model.Attributes["34869"])
          @Html.CommonDocs().Int(Model.Attributes["34869"], new { @style = "width:
45px" })
        </div>
        <div class="left">
          @Html.CommonDocs().Caption(Model.Attributes["34042"])
          @Html.CommonDocs().Int(Model.Attributes["34042"], new { @style = "width:
45px" })
        </div>
      </div>
    </div>
    <div class="left r-space" style="margin-top: 28px">
      @Html.CommonDocs().Caption(Model.Attributes["34041"])
      @Html.CommonDocs().Date(Model.Attributes["34041"])
    </div>
    <div class="left r-space" style="margin-top: 46px">
      <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34055"])</div>
      @Html.CommonDocs().CheckBox(Model.Attributes["34055"], new { @class = "left",
@style = "margin: 9px 0px 0 5px" })
    </div>
    <div class="left" style="margin-top: 46px">
      <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34056"])</div>
      @Html.CommonDocs().CheckBox(Model.Attributes["34056"], new { @class = "left",
@style = "margin: 9px 0px 0 5px" })
    </div>
    <div class="clear"></div>
    @Html.CommonDocs().Validator(Model.Attributes["34041"])
  </div>

```

```

        @Html.CommonDocs().Validator(Model.Attributes["34042"])
        @Html.CommonDocs().Validator(Model.Attributes["34869"])
    </div>
    <div>
        <div class="left r-space" style="width: 360px">
            @Html.CommonDocs().Caption(Model.Attributes["34057"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34057"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 360px" })
        </div>
        <div class="left r-space" style="margin-top: 12px">
            @Html.CommonDocs().Caption(Model.Attributes["34091"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34091"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 200px" })
        </div>
        <div class="left r-space" style="width: 220px">
            @Html.CommonDocs().Caption(Model.Attributes["34058"])
            @Html.CommonDocs().Date(Model.Attributes["34058"])
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34091"])
        @Html.CommonDocs().Validator(Model.Attributes["34057"])
        @Html.CommonDocs().Validator(Model.Attributes["34058"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34047"])
            @Html.CommonDocs().Int(Model.Attributes["34047"], new { @style = "width: 145px"
        })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34048"])
            @Html.CommonDocs().Int(Model.Attributes["34048"], new { @style = "width: 125px"
        })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34049"])
            @Html.CommonDocs().Int(Model.Attributes["34049"], new { @style = "width: 125px"
        })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34050"])
            @Html.CommonDocs().Int(Model.Attributes["34050"], new { @style = "width: 125px"
        })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34051"])
            @Html.CommonDocs().Int(Model.Attributes["34051"], new { @style = "width: 125px"
        })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34047"])
        @Html.CommonDocs().Validator(Model.Attributes["34048"])
        @Html.CommonDocs().Validator(Model.Attributes["34049"])
        @Html.CommonDocs().Validator(Model.Attributes["34050"])
        @Html.CommonDocs().Validator(Model.Attributes["34051"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34052"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34052"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 290px" })
        </div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34053"])
            @Html.CommonDocs().Int(Model.Attributes["34053"], new { @style = "width: 190px"
        })
        </div>
        <div class="left" style="margin-top: 21px">
            <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34059"])</div>

```

					KPI.IП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

```

        @Html.CommonDocs().CheckBox(Model.Attributes["34059"], new { @style = "margin:
9px 0 0 5px", @class = "left" })
    </div>
    <div class="clear"></div>
    @Html.CommonDocs().Validator(Model.Attributes["34053"])
    @Html.CommonDocs().Validator(Model.Attributes["34052"])
</div>
</div>
</div>
<div class="caption bold">@PPMDResources.F5011AccordionTitle2</div>
<div class="group w100 clean-bg">
    <div class="left r-space" style="margin-top: 12px">
        @Html.CommonDocs().Caption(Model.Attributes["34060"])
        @Html.CommonDocs().Dictionary(Model.Attributes["34060"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 365px" })
    </div>
    <div class="left r-space" style="width: 105px">
        @Html.CommonDocs().Caption(Model.Attributes["34054"])
        @Html.CommonDocs().Dictionary(Model.Attributes["34054"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 100px" })
    </div>
    <div class="left" style="margin-top: 33px">
        <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34061"])</div>
        @Html.CommonDocs().CheckBox(Model.Attributes["34061"], new { @style = "margin: 9px 0 0 5px",
@class = "left" })
    </div>
    <div class="clear"></div>
    @Html.CommonDocs().Validator(Model.Attributes["34060"])
    @Html.CommonDocs().Validator(Model.Attributes["34054"])
</div>
<div>
    @Html.CommonDocs().Table(Model.Attributes["34730"], new TableParams { DialogWidth = 900,
GridWidth = 1300, SortOrder="asc", SortName = "GRD_NUM7", GridScrollWidth = 900, IsAddAction = false,
LeftButtons = new[] { new TableBtns() { Type = "Edit" } }, null)
</div>
<div>
    <div class="left r-space" style="width: 320px">
        @Html.CommonDocs().Caption(Model.Attributes["34071"])
        @Html.CommonDocs().Dictionary(Model.Attributes["34071"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 150px" })
    </div>
    <div class="left r-space" style="margin-top: 12px; width: 130px;">
        @Html.CommonDocs().Caption(Model.Attributes["34072"])
        @Html.CommonDocs().Date(Model.Attributes["34072"])
    </div>
    <div class="left" style="margin-top: 12px">
        @Html.CommonDocs().Caption(Model.Attributes["34064"])
        @Html.CommonDocs().Dictionary(Model.Attributes["34064"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 410px" })
    </div>
    <div class="clear"></div>
    @Html.CommonDocs().Validator(Model.Attributes["34071"])
    @Html.CommonDocs().Validator(Model.Attributes["34072"])
    @Html.CommonDocs().Validator(Model.Attributes["34064"])
</div>
<div>
    @Html.CommonDocs().Caption(Model.Attributes["34461"])
    @Html.CommonDocs().Dictionary(Model.Attributes["34461"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 900px" })
    @Html.CommonDocs().Validator(Model.Attributes["34461"])
</div>

<div id="accordion_3" class="top-space accordion-padding-top-clear overflow-hidden">
    <h3>@PPMDResources.F5011AccordionTitle3</h3>
    <div>
        <div class="caption bold" style="width: 760px">@PPMDResources.F5011GroupTitle1</div>
        <div class="group clean-bg w100">

```

					KPI.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

```

<div>
  <div class="left r-space">
    @Html.CommonDocs().Caption(Model.Attributes["34073"])
    @Html.CommonDocs().Float(Model.Attributes["34073"], new { @style = "width: 200px"
  })
  </div>
  <div class="left r-space">
    @Html.CommonDocs().Caption(Model.Attributes["34074"])
    @Html.CommonDocs().Float(Model.Attributes["34074"], new { @style = "width: 170px"
  })
  </div>
  <div class="left r-space">
    @Html.CommonDocs().Caption(Model.Attributes["34075"])
    @Html.CommonDocs().Date(Model.Attributes["34075"])
  </div>
  <div class="left" style="margin-top: 21px;">
    <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34076"])</div>
    @Html.CommonDocs().CheckBox(Model.Attributes["34076"], new { @class = "left",
@style = "margin: 9px 0px 0 5px" })
  </div>
  <div class="clear"></div>
  @Html.CommonDocs().Validator(Model.Attributes["34073"])
  @Html.CommonDocs().Validator(Model.Attributes["34074"])
  @Html.CommonDocs().Validator(Model.Attributes["34075"])
</div>
<div>
  <div class="left r-space" style="width: 175px">
    @Html.CommonDocs().Caption(Model.Attributes["34077"])
    @Html.CommonDocs().Float(Model.Attributes["34077"], new { @style = "width: 175px"
  })
  </div>
  <div class="left r-space" style="margin-top: 33px">
    <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34882"])</div>
    @Html.CommonDocs().CheckBox(Model.Attributes["34882"], new { @class="left",
@style="margin: 9px 0 0 5px" })
  </div>
  <div class="left r-space" style="margin-top: 12px">
    @Html.CommonDocs().Caption(Model.Attributes["34063"])
    @Html.CommonDocs().Date(Model.Attributes["34063"])
  </div>
  <div class="left r-space" style="margin-top: 12px">
    @Html.CommonDocs().Caption(Model.Attributes["34078"])
    @Html.CommonDocs().Date(Model.Attributes["34078"])
  </div>
  <div class="left" style="margin-top: 33px;">
    <div class="left">@Html.CommonDocs().Caption(Model.Attributes["34079"])</div>
    @Html.CommonDocs().CheckBox(Model.Attributes["34079"], new { @class = "left",
@style = "margin: 9px 0px 0 5px" })
  </div>
  <div class="clear"></div>
  @Html.CommonDocs().Validator(Model.Attributes["34077"])
  @Html.CommonDocs().Validator(Model.Attributes["34078"])
  @Html.CommonDocs().Validator(Model.Attributes["34063"])
</div>
</div>
</div>
</div>
<div>
  <div class="left r-space">
    <div style="width: 470px">@Html.CommonDocs().Caption(Model.Attributes["34080"])</div>
    @Html.CommonDocs().Dictionary(Model.Attributes["34080"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 450px" })
  </div>
  <div class="left">
    <div style="width: 300px">@Html.CommonDocs().Caption(Model.Attributes["34087"])</div>
    @Html.CommonDocs().Dictionary(Model.Attributes["34087"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 410px" })
  </div>
  <div class="clear"></div>
  @Html.CommonDocs().Validator(Model.Attributes["34087"])

```

					КПІ.ІП-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

```

        @Html.CommonDocs().Validator(Model.Attributes["34080"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34067"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34067"], new { @style = "width: 590px" })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34090"])
            @Html.CommonDocs().Dictionary(Model.Attributes["34090"], dp => { dp.Behavior.SortOrder =
SortOrderType.DatabaseAsc; }, new { @style = "width: 290px" })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34067"])
        @Html.CommonDocs().Validator(Model.Attributes["34090"])
    </div>
    <div id="accordion_4" class="top-space accordion-padding-top-clear overflow-hidden">
        <h3>@PPMDResources.F5011AccordionTitle4</h3>
        <div>
            <div>
                <div class="left r-space" style="width: 130px">
                    @Html.CommonDocs().Caption(Model.Attributes["34457"])
                    @Html.CommonDocs().Date(Model.Attributes["34457"])
                </div>
                <div class="left r-space">
                    @Html.CommonDocs().Caption(Model.Attributes["34853"])
                    @Html.CommonDocs().Dictionary(Model.Attributes["34853"], dp => {
dp.Behavior.SortOrder = SortOrderType.DatabaseAsc; }, new { @style = "width: 429px" })
                </div>
                <div class="left">
                    @Html.CommonDocs().Caption(Model.Attributes["34459"])
                    @Html.CommonDocs().Text(Model.Attributes["34459"], new { @style = "width: 279px" })
                </div>
                <div class="clear"></div>
                @Html.CommonDocs().Validator(Model.Attributes["34457"])
                @Html.CommonDocs().Validator(Model.Attributes["34853"])
                @Html.CommonDocs().Validator(Model.Attributes["34459"])
            </div>
            <div>
                @Html.CommonDocs().Caption(Model.Attributes["34460"])
                @Html.CommonDocs().Text(Model.Attributes["34460"], new { @class = "w100" })
                @Html.CommonDocs().Validator(Model.Attributes["34460"])
            </div>
            <div class="caption bold">@PPMDResources.F5011GroupTitle5</div>
            <div id="moveBlock" class="group">
                <div id="movePermit">
                    <div class="left r-space">
                        <div class="caption">@PPMDResources.Zoz:</div>
                        <select id="OSSId" name="selectBox" style="width: 500px"></select>
                    </div>
                    <div class="left">
                        <div class="caption">@PPMDResources.ReasonForTransfer:</div>
                        <input type="Text" id="Reason" class="left" style="width: 300px" />
                        <input type="Button" id="ReasonBtn" title="@PPMDResources.Transfer"
onclick="changePlace()" class="btn btn-redirect left 1-space-btn" />
                    </div>
                    <div class="clear"></div>
                </div>
            </div>
            <div>
                @Html.CommonDocs().Table(Model.Attributes["34715"], new TableParams { DialogWidth =
900, IsAddAction = false, GridWidth = 856, SortName = "GRD_DATE1" }, null)
            </div>
        </div>
    </div>
    <div>
        @Html.CommonDocs().Caption(Model.Attributes["34097"])
        @Html.CommonDocs().MultiText(Model.Attributes["34097"], new MultiTextParams { IsPlain = true },
new { @style = "width: 100%; height: 50px" })
        @Html.CommonDocs().Validator(Model.Attributes["34097"])
    </div>

```

					<p>KPI.IП-5119.045440.04.13</p>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

```

</div>

<div id="accordion_9" class="top-space accordion-padding-top-clear overflow-hidden">
  <h3>@PPMDResources.F5011AccordionTitle8</h3>
  <div>
    <div>
      <div class="left r-space" style="margin-top: 12px">
        @Html.CommonDocs().Caption(Model.Attributes["34099"])
        @Html.CommonDocs().Dossier(Model.Attributes["34099"], dp => { dp.OutputAttribute =
34000; dp.AutoComplete.Enable = true; dp.AutoComplete.Length = 2; }, new { @style = "width: 671px" })
      </div>
      <div class="left" style="width: 165px">
        @Html.CommonDocs().Caption(Model.Attributes["34100"])
        @Html.CommonDocs().Date(Model.Attributes["34100"])
      </div>
      <div class="clear"></div>
      @Html.CommonDocs().Validator(Model.Attributes["34099"])
      @Html.CommonDocs().Validator(Model.Attributes["34100"])
    </div>
    <div>
      <div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34022"])
        @Html.CommonDocs().Text(Model.Attributes["34022"], new { @style = "width: 310px" })
      </div>
      <div class="left">
        @Html.CommonDocs().Caption(Model.Attributes["34023"])
        @Html.CommonDocs().Text(Model.Attributes["34023"], new { @style = "width: 150px" })
      </div>
      <div class="clear"></div>
      @Html.CommonDocs().Validator(Model.Attributes["34022"])
      @Html.CommonDocs().Validator(Model.Attributes["34023"])
    </div>
  </div>
</div>

<div id="accordion_10" class="top-space accordion-padding-top-clear overflow-hidden">
  <h3>@PPMDResources.F5011AccordionTitle9</h3>
  <div>
    <div>
      <div class="left r-space" style="margin-top: 12px">
        @Html.CommonDocs().Caption(Model.Attributes["34303"])
        @Html.CommonDocs().Dossier(Model.Attributes["34303"], dp => { dp.OutputAttribute =
34000; dp.AutoComplete.Enable = true; dp.AutoComplete.Length = 2; }, new { @style = "width: 671px" })
      </div>
      <div class="left" style="width: 165px">
        @Html.CommonDocs().Caption(Model.Attributes["34021"])
        @Html.CommonDocs().Date(Model.Attributes["34021"])
      </div>
      <div class="clear"></div>
      @Html.CommonDocs().Validator(Model.Attributes["34303"])
      @Html.CommonDocs().Validator(Model.Attributes["34021"])
    </div>
    <div>
      <div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34101"])
        @Html.DropDownList("Doctors", Enumerable.Empty<SelectListItem>(), "", new { @style =
"width: 310px" })
      </div>
      <div class="left r-space">
        @Html.CommonDocs().Caption(Model.Attributes["34102"])
        @Html.CommonDocs().Text(Model.Attributes["34102"], new { @readonly = "readonly",
@style = "width: 264px" })
      </div>
      <div class="left">
        @Html.CommonDocs().Caption(Model.Attributes["34103"])
        @Html.CommonDocs().Text(Model.Attributes["34103"], new { @readonly = "readonly",
@style = "width: 264px" })
      </div>
      <div class="clear"></div>
      @Html.CommonDocs().Validator(Model.Attributes["34102"])
    </div>
  </div>
</div>

```

					KPI.II-5119.045440.04.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

```

        @Html.CommonDocs().Validator(Model.Attributes["34103"])
    </div>
    <div>
        <div class="left r-space">
            @Html.CommonDocs().Caption(Model.Attributes["34104"])
            @Html.CommonDocs().Text(Model.Attributes["34104"], new { @readonly = "readonly",
@style = "width: 310px" })
        </div>
        <div class="left">
            @Html.CommonDocs().Caption(Model.Attributes["34105"])
            @Html.CommonDocs().Text(Model.Attributes["34105"], new { @readonly = "readonly",
@style = "width: 264px" })
        </div>
        <div class="clear"></div>
        @Html.CommonDocs().Validator(Model.Attributes["34104"])
        @Html.CommonDocs().Validator(Model.Attributes["34105"])
    </div>
</div>
<div>
    @Html.CommonDocs().Caption(Model.Attributes["34107"])
    @Html.CommonDocs().MultiText(Model.Attributes["34107"], new MultiTextParams { IsPlain = true },
new { @style = "width: 900px; height: 50px" })
    @Html.CommonDocs().Validator(Model.Attributes["34107"])
</div>

<div>
    @Html.CommonDocs().HistoryTable(new HistoryTableParams
    {
        DossierSectionId = Model.ItemId,
        Width = 900,
        InAccordion = true,
        RequiredCols = new List<SysColType>()
        {
            SysColType.FullName, SysColType.Position, SysColType.Organization, SysColType.Date, SysColType.Action
        })
    </div>

```

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА
ОЦІНКИ ЕФЕКТИВНОСТІ ПРОФІЛАКТИЧНИХ ЗАХОДІВ ДЛЯ
ЗАПОБІГАННЯ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ

Програма та методика тестування

КП.ІІ-5119.045440.05.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ К.І. Ліщук

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ С.В. Реутська

Київ – 2019 року

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАНЬ	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	7

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Web-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини, яке являє собою web-сайт, створений на платформі Microsoft .Net з використанням технологій ASP.NET MVC, JQuery, AJAX.

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- функціональна працездатність елементів сторінок web-ресурсу;
- наявність доступу до бази електронних медичних карток;
- можливість генерації звітності відповідно до інформації в базі даних;
- забезпечення належного рівня безпеки даних;
- зручність роботи з web-сайтом;
- відповідність дизайну вимогам Технічного завдання.

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 МЕТОДИ ТЕСТУВАННЯ

Тестування виконується методом Black Box Testing. Перевіряється безпосередньо програмний продукт на відповідність функціональним вимогам. Для надійності функціонал буде протестований як при позитивних, так і при негативних умовах.

Проводяться усі рівні тестування, від модульного до приймального (під час інтеграції проекту в реальні заклади здоров'я України). Особлива увага приділена системному тестуванню на базі вимог та випадків користування. Після викладки кожної нової версії програми, обов'язкове проведення регресійного тестування, яке буде перевіряти правильну роботу функціональності, доданої в попередніх версіях.

Також передбачене проведення Бета-тестування за допомогою добровольців з 5 центрів ЗОЗ перед кожним релізом програми.

Будуть проводитись наступні функціональні типи тестів:

- димне тестування (падіння системи при переході на різні сторінки, зберігання, редагування та видалення карточки);
- критичне тестування (зберігання належним чином даних з усіх полів картки).
- розширене тестування (прив'язки карток, надання доступу різним користувачам);

Будуть проводитись наступні нефункціональні типи тестів:

- тестування інтерфейсу (наявність усіх кнопок, які необхідні для працювання з системою, їх робота, коректне відображення інформації карток, валідація полів);
- тестування безпеки (шифрування даних про пацієнта згідно з вимогами);

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

– навантажувальне тестування (прив'язка до картки матері до 5ти дітей, перенесення картки з однієї області в іншу до 10 разів, наявність тисячі карток у архіві);

– тестування установки (доступ для лікаря з комп'ютера ЗОЗ до закритого серверу, на якому лежить програма);

тестування сумісності (з різними видами браузерів (Chrome, FireFox, Safari, Opera, InternetExplorer)).

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Працездатність web-ресурсу перевіряється шляхом:

- динамічного ручного тестування – введенням граничних та недопустимих значень в поля медичної картки, які можна редагувати;
- динамічного ручного тестування – відслідковуванням коректності автоматично заповнюваних полів медичних карток;
- динамічного ручного тестування – створенням зв'язку між картками матері та дитини;
- динамічного ручного тестування – формуванням звітності на основі введених в систему даних;
- динамічного ручного тестування – перенесенням картки в інший ЗОЗ;
- динамічного ручного тестування на відповідність функціональним вимогам;
- тестування шифрування персональних даних пацієнта в базі;
- тестування web-ресурсу в різних web-браузерах;
- тестування при максимальному навантаженні;
- тестування стабільності роботи при різних умовах;
- тестування зручності використання;
- тестування інтерфейсу.

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА
ОЦІНКИ ЕФЕКТИВНОСТІ ПРОФІЛАКТИЧНИХ ЗАХОДІВ ДЛЯ
ЗАПОБІГАННЯ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ

Керівництво користувача

КП.ІІ-5119.045440.06.34

“ПОГОДЖЕНО”

Керівник проекту:

_____ К.І. Ліщук

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ С.В. Реутська









Київ – 2019 року

ЗМІСТ

1	ЗАГАЛЬНИЙ ОПИС РОБОТИ ІНТЕРФЕЙСУ.....	3
1.1	РОБОТА З ТАБЛИЦЯМИ.....	3
1.2	ПОШУК ІНФОРМАЦІЇ.....	4
1.3	ОБОВ`ЯЗКОВІСТЬ ПОЛІВ.....	5
2	АВТОРИЗАЦІЯ КОРИСТУВАЧА.....	6
2.1	ВХІД У СИСТЕМУ.....	6
2.2	МЕНЮ СИСТЕМИ.....	7
3	РОБОТА З КАРТКАМИ.....	8
3.1	РОЗДІЛ «РЕЄСТР ЗОЗ».....	8
3.2	РОБОТА З КАРТКАМИ ПАЦІЄНТІВ.....	9

1 ЗАГАЛЬНИЙ ОПИС РОБОТИ ІНТЕРФЕЙСУ

Управляючі елементи:

-  кнопка календар для введення дати;
-  кнопка додати картку;
-  кнопка обрати файл;
-  кнопка завантажити файл;
-  кнопка перегляду картки;
-  кнопка редагування картки;
-  кнопка видалення картки;
-  кнопка меню медичних форм картки.

1.1 Робота з таблицями



Для переміщення по сторінкам таблиці використовується пейджинговий навігаційний блок, а також дозволяє задавати кількість елементів на сторінці.




Рисунок 1.1 – Навігаційний блок таблиці

Для зручності роботи користувача є можливість сортування даних в таблиці за алфавітом для полів типу "текст" і сортування даних за значенням дат для полів типу "дата" за всіма реквізитами таблиці, для чого необхідно натиснути на заголовок відповідного стовпчика таблиці.

Повторне натиснення по заголовку стовпчика приведе до сортування в зворотному напрямку.

У кожному рядку таблиці розташовані кнопки для роботи з рядками таблиці. Так наприклад, кнопка "Перегляд"  дозволяє переглянути інформацію з відповідної картки таблиці, кнопка "Редагувати" 

					КПІ.ІП-5119.045440.06.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

призначена для редагування інформації відповідної картки таблиці, кнопка "Видалити"  призначена для видалення картки з таблиці та з БД. Для виконання операції з записами таблиці необхідно обрати потрібний рядок і натиснути відповідну кнопку.

1.2 Пошук інформації

На сторінках з таблицями розташовані два інструменти пошуку і фільтрації інформації: фільтр по полям картки та контекстний пошук.

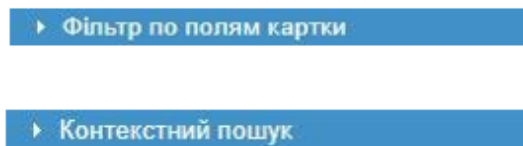


Рисунок 1.2 – Пошук

Інструмент «Контекстний пошук» допомагає знайти інформацію по прізвищу та імені ВІЛ-інфікованої особи. Слід звернути увагу, що з метою дотримання конфіденційності даних «Контекстний пошук» працює по принципу «дорівнює», тобто шукає лише при точно введеному прізвищі або імені особи. У випадку допущення хоча б однієї помилки, пошук не дасть результату.

Інструмент пошуку і фільтрації інформації в розгорнутому вигляді представлений нижче.

Рисунок 1.3 Фільтр по полям картки

Рисунок 1.4 Контекстний пошук

1.3 Обов'язковість полів

Обов'язкові поля карток позначаються червоною зірочкою «*».

При спробі збереження картки після заповнення або редагування її, якщо хоча б один з обов'язкових реквізитів не вказано, з'являється нагадування про обов'язковість заповнення.

2 АВТОРИЗАЦІЯ КОРИСТУВАЧА

2.1 Вхід у систему

ППМД надає доступ до функціональності за допомогою технології "тонкого клієнта" (використовуючи браузер). Це означає, що всі операції при роботі з програмним забезпеченням системи відбуваються на веб-сторінках у вікні браузера.

Для того, щоб запустити ППМД, необхідно пройти процес авторизації користувачів, використовуючи механізм аутентифікації, який служить єдиним вікном доступу до Системи.

Достатньо ввести в адресний рядок браузера адресу Системи. Відкривається головна сторінка ППМД, яка містить блок авторизації у Системі.



Рисунок 2.1 Головна сторінка

Необхідно ввести "Логін" і "Пароль" і натиснути на кнопку "Вхід". Логін і пароль можна отримати у адміністратора системи. У разі вдалої аутентифікації користувач потрапляє на головну сторінку ППМД, на якій він може вибрати необхідний функціонал.

					КПІ.ІП-5119.045440.06.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

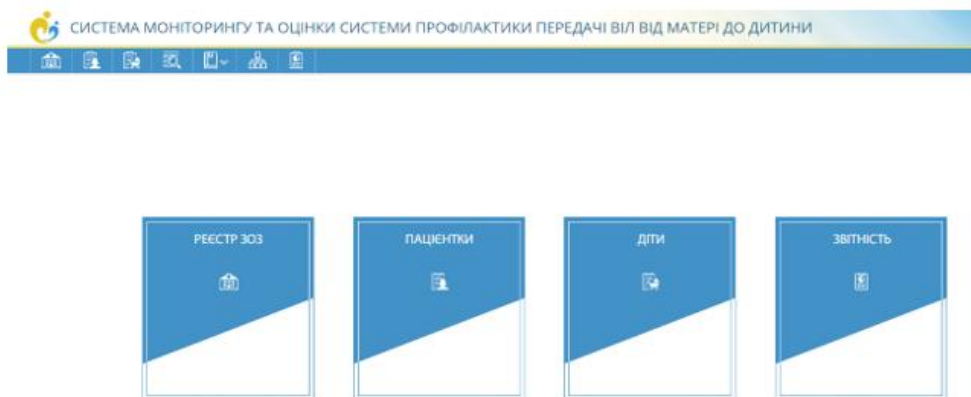


Рисунок 2.2 - Головна сторінка авторизованого користувача

2.2 Меню системи

Структура розділів системи:


- реєстр ЗОЗ;
- пацієнтки;
- діти;
- переміщення карток;
- довідники;
- звітність.

Після отримання користувачем доступу до певного функціоналу основним інструментом навігації є головне меню. Вихід з ППМД, або зміна користувача системи здійснюється натисканням кнопки "Вийти".

					КПІ.ІП-5119.045440.06.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

3 РОБОТА З КАРТКАМИ

3.1 Розділ «Реєстр ЗОЗ»


Для входу у розділ «Реєстр ЗОЗ» необхідно підвести курсор до значка  на головному меню системи або натиснути на функціональному блоці «Реєстр ЗОЗ».

РЕЄСТР ЗОЗ

Фільтр по всім карткам

Найменування ЗОЗ	Область	Поштовий індекс	Район	Населений пункт	Вулиця, будинок	Код ЄДРПОУ	Тип ЗОЗ
Центральна санітарно-епідеміологічна станція на повітряному транспорті	КІРОВОГРАДСЬКА ОБЛАСТЬ	38702		МОЛЕКСАНДРІЯ		151	Центр СНІДу
центр реабілітації репродуктивної функції людини	КІРОВОГРАДСЬКА ОБЛАСТЬ			МОЛЕКСАНДРІЯ		160	Заклад охорони здоров'я
Фельдшерсько-акушерський пункт	КІРОВОГРАДСЬКА ОБЛАСТЬ			МОЛЕКСАНДРІЯ		167	Центр СНІДу
Українська станція екстреної експресної консультативної медичної допомоги	КІРОВОГРАДСЬКА ОБЛАСТЬ			МОЛЕКСАНДРІЯ		169	Заклад охорони здоров'я
Спеціалізована профілактика ІНЗІ України при Національному університеті	ІВАНО-ФРАНКІВСЬКА ОБЛАСТЬ			М.ВОЛЕХІВ		165	Заклад охорони здоров'я

Рисунок 3.1 Список реєстрів

Для створення нової картки «ЗОЗ» необхідно натиснути кнопку , що розміщена над таблицею.

РЕЄСТР ЗОЗ

Зберегти Закрити

Приїздки до ОШС:

Найменування ЗОЗ: *

Пологовий будинок №5

Найменування установи, у сфері управління якої перебуває ЗОЗ:

Міністерство

Код ЄДРПОУ: 34343456 Тип ЗОЗ: * Заклад охорони здоров'я

Адреса

Код КОАТУУ: * 0710100000 Область: ВОЛИНСЬКА ОБЛАСТЬ Район: Місто: М.ЛУЦЬК

Вулиця, будинок: Баумана 43 Поштовий індекс: 56543

Лікарі


Прізвище	Ім'я	По батькові	Посада	Контактний телефон	E-mail
Костюченко	Юрій	Миколайович	головний лікар		

Системна інформація

Рисунок 3.2 Картка «Реєстру ЗОЗ»

Після остаточного заповнення форми необхідно натиснути кнопку «Зберегти» для збереження внесених даних або змін у картці.

3.2 Робота з картками пацієнтів

Для входу у розділ «Пацієнтки» необхідно підвести курсор до значка  на головному меню системи або натиснути на функціональному блоці «Пацієнтки».

Користувач має можливість перегляду карток розділу «Архів», а також створення, редагування та перегляд карток розділу «Пацієнтки».

«Архів» - це картки пацієнток, які стояли на обліку в цьому Центрі, але зараз з якихось причин зняті з обліку.

КАРТКА ПАЦІЄНТКИ

Підрозділи


- Архів (3)
- Картки (11)


Фільтр по полові картки

Контекстний подив

Ідентифікатор картки	Унікальний код в ЛІС	Дата народження	Місце реєстрації	дата виходу на облік	Найменування ЗОЗ	Клінічна стадія ВІЛ-інфекції на момент першого обстеження у ЗОЗ, що здійснює медичний нагляд за ВІЛ-позитивними особами, під час даного відвідування
мінСлосРабгт154	970111	10.06.1976	Харсон	02.07.2018	Інформаційно-аналітичний центр медичної статистики	IV стадія
мінСлосРабгт151	970040	01.06.2018	Харсон	03.06.2018	Інформаційно-аналітичний центр медичної статистики	ГРВ
мінСлосРабгт148	870110	17.11.1982	Харсон	03.06.2018	Інформаційно-аналітичний центр медичної статистики	ГРВ
мінСлосРабгт14	970040	01.06.1972	Харсон	03.07.2018	Інформаційно-аналітичний центр медичної статистики	IV стадія


Рисунок 3.3 Список карток

Користувач в залежності від своєї ролі має можливість редагувати дані усіх полів картки. Для редагування реквізитів картки пацієнтки необхідно натиснути кнопку редагування  у рядку відповідної картки пацієнтки.

Для створення нової картки «Пацієнтки» необхідно натиснути кнопку , що розміщена над таблицею.


Усі поля картки заповнюються згідно правил, що описані у розділі 3.5 цього документа.

Рисунок 3.4 Картка «Пацієнтка»

Після першого збереження картки, з'являється можливість згенерувати «Ідентифікатор матері» за допомогою кнопки . «Ідентифікатор пацієнтки» користувач формує одноразово за допомогою кнопки «ID», після чого кнопка «ID» стає недоступною. Після генерування ідентифікатора карту необхідно обов'язково «Зберегти». Алгоритм формування ідентифікатора: XXX – перші три літери «Область» з реєстра ЗОЗ, XXX – перші три літери прізвища матері, XX – перші дві літери імені матері, XXX – перші три літери по батькові матері, наскрізний номер.

ПОВІДОМЛЕННЯ № 1 про ВІЛ-позитивну вагітну

Номер документа:

Дата заповнення: * 



Ідентифікатор матері: 

Рисунок 3.5 Згенерований ідентифікатор

Перенесення картки до архіву. Картки, що втратили чинність, переміщуються до архіву. Для цього необхідно натиснути кнопку  «Переміщення картки в архів». Картка з'являється у підрозділі «Архів» в режимі перегляду.


					КПІ.ІП-5119.045440.06.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Переміщення пацієнтки в інший ЗОЗ. На вкладці існує «Таблиця переміщення». Користувач бачить історію переміщення картки пацієнтки. Право перемістити картку має тільки Адміністратор системи.

Переміщення пацієнтки в інший ЗОЗ

Таблиця переміщення						
№ з/п	Найменування ЗОЗ	Дата взяття на облік	Користувач	Дата зняття з обліку	Користувач	Примітка
1	КУ "Херсонський обласний центр профілактики та боротьби зі СНІДом"	27.06.2018 17:22	Адмін			

Рисунок 3.6 Таблиця переміщення

Таблиця «Діти». Для створення нового запису необхідно натиснути кнопку , що розміщена над таблицею. З'являється спливаюче вікно, в якому обираються записи з довідника.

Користувач додає стільки записів, скільки дітей народилося.

РЕДАГУВАННЯ ЗАПИСУ

Дитина: жива Стать: дівчинка зріст, см: 53 вага, г: 2900 Обстеження на ВІЛ дитини після народження у пологовому будинку (відділенні) проведено: так

Якщо так, зазначити метод дослідження:

Дослідження методом СКК: дата забору: 07.06.2018 Номер зразка: 654 дата проведення: 07.06.2018 Результат: Позитивний

Дослідження за допомогою одного ШТ: дата проведення: 08.06.2018 номер зразка: 21 результат: Негативний

зазначити з якої причини:

Дослідження за допомогою ІМ: дата забору: 08.06.2018 номер зразка: 23 дата проведення: 09.06.2018 результат: Негативний

АРВ-профілактика дитини

Назва препарату	Кількість днів
AZT	2

Причина, з якої не було АРВ-профілактики:

Вид вигодовування дитини під час перебування у пологовому будинку (зазначити кількість днів/годин):

грудне: 2 Од. виміру: годин змішане: 3 Од. виміру: годин штучне: 5 Од. виміру: годин Проведення медикаментозного припинення лактації породіллі: так

якщо ні, зазначити причину, з якої медикаментозне припинення лактації в пологовому будинку (відділенні) не проводилося: не проводилося

Доля дитини: виписана разом з матір'ю вказати на яку добу: 4

Рисунок 3.7 Таблиця діти

Прив'язка матері до дитини. Для прив'язки картки дитини до карти матері необхідно обрати запис дитини в таблиці «Діти» та натиснути кнопку



«Створити і прив'язати картку дитини». У записі заповниться поле «Посилання на дитину».

При натисненні на посилання «Картка дитини» створюється нова картка дитини з автоматично скопійованими полями персональних даних матері і датою народження дитини (з правом редагування).

КАРТКА ДИТИНИ Зберегти

ПОВІДОМЛЕННЯ № 3 (I) про перебування на обліку дитини, народженої ВІЛ-позитивною жінкою
[Картка матері](#)

Номер документа: Дата заповнення: * 09.11.2017 Ідентифікатор дитини: ХЕРСтаНаВікЗ ID Унікальний код в МІС: 1342345

Статус картки: Причина недійсності:

Статус форми № 501-3(II)/о: Причина недійсності: Статус форми 501-3(III)/о: нова Причина недійсності:

Статус форми 501-3(III)/о: нова Причина недійсності: Статус форми 501-4/о: Причина недійсності:

▶ Дані ЗОЗ, з якого отримана форма № 501-3 (Уо)

▼ Персональні дані дитини, народженої ВІЛ-інфікованою жінкою

Прізвище дитини: Стайко Ім'я: Надія По батькові: Вікторівна Дата народження: 06.10.2017 Стать дитини: дівчинка

Повна поштова адреса

Місце реєстрації: Скадовськ Вулиця: Чкалова Будинок, квартира: 12

Місце фактичного проживання: Скадовськ Телефон: 121-211 E-mail: stayko@bigmir.net

Інша контактна інформація: a/c 232

Прізвище матері: Стайко Ім'я: Яніна По батькові: Владіславівна Унікальний код матері в МІС: 2342321

Рисунок 3.8 - Створена картка дитини

					КП.ІП-5119.045440.06.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ___ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ДЛЯ ПРОВЕДЕННЯ АНАЛІЗУ ТА
ОЦІНКИ ЕФЕКТИВНОСТІ ПРОФІЛАКТИЧНИХ ЗАХОДІВ ДЛЯ
ЗАПОБІГАННЯ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ

Графічні матеріали

КПІ.ІІ-5119.045440.07.99

“ПОГОДЖЕНО”

Керівник проекту:

_____ К.І. Ліщук

Нормоконтроль:

_____ К.І. Ліщук

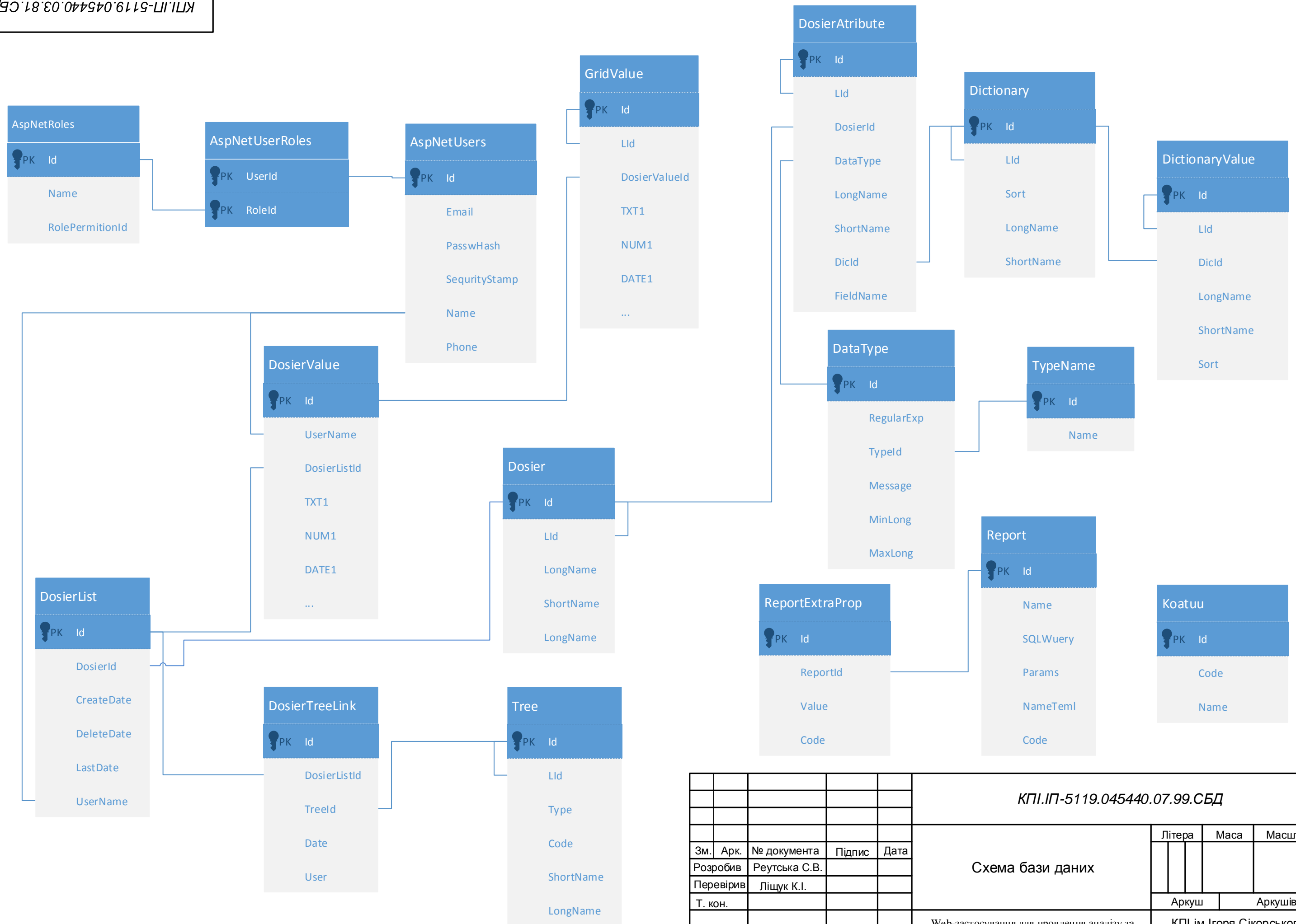
Виконавець:

_____ С.В. Реутська

Київ – 2019 року

КПІ.ІП-5119.045440.05.51

					КПІ.ІП-5119.045440.05.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2



					КПІ.ІП-5119.045440.07.99.СБД							
					Схема бази даних							
										Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Реутська С.В.										
Перевірів		Ліщук К.І.										
Т. кон.					Аркуш		Аркушів					
Н. кон.		Ліщук К.І.			КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51							
Затвердив		Ліщук К.І.										

Web-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини

СИСТЕМА МОНІТОРИНГУ ТА ОЦІНКИ СИСТЕМИ ПРОФІЛАКТИКИ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ



РЕЄСТР ЗОЗ

Зберегти і закрити Зберегти Закрити

Прив'язка до ОШС:

КЗ "Вінницький обласний клінічний Центр профілактики та боротьби зі СНІДом" № «»

Найменування ЗОЗ: *

Заклад охорони здоров'я Вінниця

Найменування установи, у сфері управління якої перебуває ЗОЗ:

Код ЄДРПОУ: Тип ЗОЗ: *

Заклад охорони здоров'я

Адреса

Код КОАТУУ: * 0510100000 Область: ВІННИЦЬКА ОБЛАСТЬ Район: Місто: М.ВІННИЦЯ

Вулиця, будинок: Лесі Українки, 32 Поштовий індекс: 04545

Лікарі

Прізвище	Ім'я	По батькові	Посада	Контактний телефон	E-mail
Петров	Іван	Сидорович	головний лікар	0445677656	aa@gmail.com

Системна інформація

Зберегти і закрити Зберегти Закрити

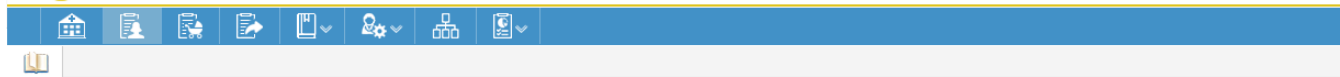
СИСТЕМА МОНІТОРИНГУ ТА ОЦІНКИ СИСТЕМИ ПРОФІЛАКТИКИ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ



ЗВІТНІСТЬ

- Профілактика передачі ВІЛ від матері до дитини
 - Форма № 63 Таблиця 1000
 - Форма № 63 Таблиця 1001
 - Форма № 63 Таблиця 1002
 - Форма № 63 Таблиця 2000
 - Форма № 63 Таблиця 2001
 - Форма № 63 Таблиця 3000
 - Форма № 63 Таблиця 3001
 - Форма № 63 Таблиця 4000
 - Форма № 63 Таблиця 4001
 - Форма № 63 Таблиця 4002
 - Форма № 63 Таблиця 4003
 - Форма № 63 Таблиця 4004
- Частота передачі ВІЛ від матері до дитини
 - Форма № 63-1 Таблиця 1000
 - Форма № 63-1 Таблиця 2000
 - Форма № 63-1 Таблиця 2001
 - Форма № 63-1 Таблиця 3000

СИСТЕМА МОНІТОРИНГУ ТА ОЦІНКИ СИСТЕМИ ПРОФІЛАКТИКИ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ



КАРТКА ПАЦІЄНТКИ

Зберегти і закрити Зберегти Закрити

Форма № 501-1/о Форма № 501-2/о Пов'язані документи та файли

ПОВІДОМЛЕННЯ № 1 про ВІЛ-позитивну вагітну
 Номер документа: Дата заповнення: * 01.05.2019 Ідентифікатор матері: Форма № 025/о: Унікальний код в МІС:

Статус картки: нова Причина недійсності:
 Статус форми № 501-1/о: нова Причина недійсності: Статус форми № 501-2/о: нова Причина недійсності:

Дані ЗОЗ, з якого отримана форма № 501-1/о

Тип ЗОЗ: Найменування ЗОЗ:
 Найменування установи, у сфері управління якої перебуває ЗОЗ:
 Поштовий індекс: Назва міста чи населеного пункту, де знаходиться ЗОЗ: Вулиця, № будинку, де знаходиться ЗОЗ: Код ЄДРПОУ:

Персональні дані ВІЛ-позитивної вагітної

Прізвище: Ім'я: По батькові: Змінене прізвище:
 Дата народження: кількість повних років: Громадянство: Україна: Інша країна: Особа без громадянства:
 Повна поштова адреса
 Місце реєстрації: Вулиця: Будинок, квартира:

СИСТЕМА МОНІТОРИНГУ ТА ОЦІНКИ СИСТЕМИ ПРОФІЛАКТИКИ ПЕРЕДАЧІ ВІЛ ВІД МАТЕРІ ДО ДИТИНИ



КАРТКА ПЕРЕМІЩЕННЯ

Зберегти і закрити Зберегти Закрити

Картка пацієнтки Картка дитини

Ідентифікатор: Унікальний код в МІС: 1 Форма № 025/о: 7777 Підтягнути картку

Найменування ЗОЗ (звідки): * КЗ "Вінницький обласний клінічний Центр профілактики та боротьби зі СНІДом" № «» Дата заповнення 1-ої вкладки: 01.05.2019

Дата взяття на облік електронної картки: 01.05.2019 Користувач: Админ Дата взяття з обліку електронної картки:

Користувач: Причина зняття з обліку:

Найменування ЗОЗ (куди): *

Примітка:

Зберегти і закрити Зберегти Закрити

					КПІ.ІП-5119.045440.07.99.KE				
Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм		Літера	Маса	Масштаб
Розробив		Реутська С.В.							
Перевірив		Ліщук К.І.					Аркуш	Аркушів	
Т. кон.									
Н. кон.		Ліщук К.І.			Web-застосування для проведення аналізу та оцінки ефективності профілактичних заходів для запобігання передачі ВІЛ від матері до дитини		КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51		
Затвердив		Ліщук К.І.							