

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

МЕТОДІВ ОЦІНЮВАННЯ РИЗИКІВ В БІОІНЖЕНЕРІЇ

RESEARCH AND SOFTWARE IMPLEMENTATION

OF METHODS FOR BIOENGINEERING RISK ASSESSMENT

Виконав: студент(ка) 2 курсу, групи КНТ-219м

Спеціальності 122 Комп'ютерні науки

забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Дубецький А.С.

(прізвище та ініціали)

Керівник Табунщик Г.В.

(прізвище та ініціали)

Рецензент Шитікова О.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Дубецький Андрій Станіславович
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів оцінювання ризиків в біоінженерії. Research and Software Implementation of Methods for Bioengineering Risk Assessment
 керівник проєкту (роботи) Табунщик Галина Володимирівна, к.т.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
 затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 5 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Розробка програми. 5. Економіко-організаційна частина. 6. Охорона праці та цивільна безпека. 7. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Табунщик Г.В., професор		
Організаціо-економічна частина	Пожуєва Т.О., професор		
Охорона праці та цивільна безпека	Коробко О.В., ст. викладач		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “ 4 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–9 тижні	Розділ 4
6	Розробка організаційно-економічної частини	10 тиждень	Розділ 5
7	Розробка заходів з охорони праці та цивільної безпеки.	11 тиждень	Розділ 6
8	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
9	Нормоконтроль та рецензування.	14–15 тижні	
10	Захист роботи.	16 тиждень	

Студент(ка)

Дубецький А.С.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

Табунщик Г.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
148 с., 22 табл., 19 рис., 2 дод., 64 джерел.

БІОІНЖЕНЕРІЯ, МЕДИЧНІ СИСТЕМИ, ОЦІНЮВАННЯ РИЗИКІВ,
ЕКСПЕРТНІ МЕТОДИ ОЦІНКИ.

Об'єкт дослідження – ризики в біоінженерії.

Предмет дослідження – методи оцінювання ризиків в біоінженерії.

Мета роботи – дослідження методів оцінювання ризиків медичних систем та їх процесів, що дозволить створити програмну систему для автоматизації та спрощення проведення процедури оцінювання ризиків для медичних систем та їх процесів.

Матеріали, методи та технічні засоби: експертні методи оцінювання ризиків, структурне та об'єктно-орієнтоване програмування, мова програмування JavaScript та фреймворк React, технології Node.js та MySQL, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 7.

Результати. Створено програмний вебзастосунок, за допомогою якого можна проводити опитування експертів, проводити розрахунки та отримувати рекомендації щодо оцінювання ризиків медичних систем та їх процесів.

Висновки. Досліджено методи оцінювання ризиків в біоінженерії, на підставі чого розроблений програмний продукт, який дозволяє автоматизувати та значно прискорити проведення процедури оцінювання ризиків медичних систем та їх процесів з використанням групового експертного оцінювання.

Галузь використання – процес управління ризиками в біоінженерії.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 148 pages, 19 figures, 22 tables, 2 appendixes, 64 sources.

BIOENGINEERING, MEDICAL SYSTEMS, RISK ASSESSMENT, EXPERT ASSESSMENT METHODS.

Object of study are risks for bioengineering.

Subject of study are methods for bioengineering risk assessment.

The purpose of the work is to research of methods for risk assessment of medical systems and their processes, which will create a software implementation to automate and simplify the risk assessment procedure.

Materials, methods, and tools: expert methods of risk assessment, structured and object-oriented programming, programming language JavaScript and framework React, server technologies Node.js and MySQL, IBM-compatible personal computer with an Intel Core 2 Duo processor under the control of the Microsoft Windows 7 operating system.

Results. The Web software application is created, which conducts surveys of experts, calculates and receives recommendations for risk assessment of medical systems and their processes.

Conclusions. Methods for bioengineering risk assessment has been researched and software implementation has been developed. It automates and accelerates the procedure of medical systems risk assessment by an expert group.

The field of use is a process for bioengineering risk management.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Управління ризиками медичних систем та їх процесів	11
1.2 Етапи процесу управління ризиками	14
1.4 Огляд існуючих програмних засобів підтримки процесу управління ризиками	26
1.5 Постановка завдання	30
2 РОЗРОБКА ТА УДОСКОНАЛЕННЯ МЕТОДІВ Й АЛГОРИТМІВ ВИРІШЕННЯ ЗАДАЧІ	32
2.1 Визначення матриці ризиків	32
2.2 Вибір методів експертного оцінювання	34
2.2.1 Загальна постановка задачі експертного оцінювання	34
2.2.2 Оцінка узгодженості думок експертів	35
2.3 Розробка алгоритму роботи програми	37
3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ	39
3.1 Вибір та обґрунтування загальної архітектури програми	39
4 РОЗРОБКА ПРОГРАМИ	48
4.1 Структурна схема програмної системи	48
4.2 Опис функціоналу системи	49
4.3 Опис бази даних	50
4.4 Розробка кабінету користувача	57
4.4.1 Модуль керування інформацією в базі даних	57
4.4.2 Модуль ініціювання експертизи	58
4.4.3 Модуль проведення експертизи	58
4.4.4 Модуль обробки результаті проведеної експертизи	58
4.4.5 Модуль керування даними користувачів	58
4.5 Виконання програмного продукту	59

	7
4.5.1 Головна сторінка	59
4.5.2 Персональний кабінет	60
4.5.3 Сторінки «Про проєкт» та «Контакти»	62
4.5.4 Сторінка «Робота з БД»	64
4.5.5 Сторінка «Проведення експертизи»	64
4.5.6 Сторінка «Оцінювання ризиків»	68
5 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА	69
5.1 Планування розробки програмного продукту	69
5.2 Визначення витрат на розробку програми	70
5.3 Розрахунок економічної ефективності програмного продукту	77
6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	79
6.1 Аналіз потенційних небезпек.	79
6.2 Заходи із забезпечення безпеки.	80
6.3 Заходи з виробничої санітарії та гігієни праці.	87
6.4 Заходи з забезпечення безпеки у надзвичайних ситуаціях.	89
6.4.1 Заходи з пожежної безпеки.	90
6.4.2 Заходи з цивільного захисту.	91
ВИСНОВКИ.....	94
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	95
ДОДАТОК А Текст програми.....	103
ДОДАТОК Б Слайди презентації	137

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AJAX	– Asynchronous Javascript and XML;
API	– Application Programming Interface;
ETA	– Event Tree Analysis;
FMEA	– Failure Mode and Effects Analysis;
FMECA	– Failure Mode, Effects and Critical Analysis;
FTA	– Fault Tree Analysis;
HAZOP	– Hazard and Operability Study;
HTML	– HyperText Markup Language;
HTTP	– HyperText Transfer Protocol;
JSON	– JavaScript Object Notation;
MVC	– Model View Controller;
PDF	– Portable Document Format;
PHA	– Preliminary Hazard Analysis;
SQL	– Structured Query Language;
XML	– eXtensible Markup Language;
БД	– база даних;
ОПР	– особа, що приймає рішення.

ВСТУП

Біоінженерія (bioengineering) – це напрям науки і техніки, що застосовує інженерні принципи, методи та технології у галузях медицини та біології, особливо при вирішенні проблем та поліпшенні обслуговування (як при проектуванні медичних приладів та діагностичного обладнання, так і при створенні біоматеріалів та фармацевтичних препаратів) [1].

Для медичних систем та їх процесів першочергове значення має якість та безпека застосування, бо вони мають безпосередній вплив на життя та здоров'я пацієнтів, персонал, операторів, а також навколишнє середовище.

Безпечним вважається медичний виріб, що використовується згідно передбаченого застосування та для якого відсутній неприпустимий ризик.

Передбачене застосування – використання медичного виробу, процесу або послуги відповідно до технічних умов (специфікацій), інструкцій та інформації, наданих виробником [2].

Ризик – поєднання вірогідності виникнення шкоди і ступеня тяжкості цієї шкоди [3].

На Україні діє стандарт [2], який вимагає від виробників медичних виробів встановлювати, документувати і підтримувати в робочому стані безперервний процес ідентифікації небезпек (ризиків), пов'язаних з медичним виробом, визначення та оцінювання супутніх ризиків, управління даними ризиками і моніторинг у результативності такого управління протягом усього життєвого циклу медичного виробу [4].

Згідно [5] до медичних виробів серед іншого відносять також програмне забезпечення, що застосовується як окремо, так і в поєднанні між собою (включаючи програмне забезпечення, передбачене виробником для застосування спеціально для діагностичних та/або терапевтичних цілей та необхідне для належного функціонування медичного виробу).

Всі ці дії вимагають отримання, аналізування та використання великої кількості різноманітної інформації щодо надійності та безпеки медичних

виробів, можливих факторів впливу та послідовності подій, що здатні привести до небезпечної ситуації. Значно спростити та запобігти отриманню хибного результату при оцінці ризиків здатні системи автоматизації.

Тому розробка програмних систем, які дозволяють автоматизувати процес аналізу та оцінювання ризиків медичних систем та їх процесів є достатньо актуальним завданням.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Управління ризиками медичних систем та їх процесів

Основним призначенням медичних виробів є поліпшення якості і збільшення тривалості людського життя, тому галузь виробництва медичних систем знаходиться під пильним контролем регулюючих органів.

В Україні зазначену функцію виконує Державна служба з лікарських засобів, що здійснює нагляд за обігом медичних виробів на ринку України, в тому числі здійснює перевірку наявності процедури оцінки відповідності згідно законодавства України.

У сфері технічних регламентів та оцінки відповідності встановлено, що всі медичні вироби повинні відповідати нормативним вимогам згідно:

- технічного регламента щодо медичних виробів, затвердженому постановою Кабінету Міністрів України від 02.10.2013 № 753 [5];
- технічного регламента щодо медичних виробів для діагностики *in vitro*, затвердженому постановою Кабінету Міністрів України від 02.10.2013 № 754;
- технічного регламента щодо активних медичних виробів, які імплантують, затвердженому постановою Кабінету Міністрів України від 02.10.2013 № 755 [6].

Згідно вимог [7] до системи управління якістю медичних виробів організація (виробник, представник, імпортер, дистриб'ютор тощо) повинна застосовувати підхід, заснований на оцінці ризиків, для контролю відповідних процесів, необхідних для функціонування системи управління якістю.

Ризик-орієнтовна методологія використовується на всіх етапах життєвого циклу медичних систем, починаючи з етапу проектування та продовжуючи на поствиробничому етапі. На ризики, що виникли на одному етапі життєвого циклу, можна впливати впровадженням деяких заходів на іншому етапі життєвого циклу. Все це є складовими процесу управління ризиком.

Управління ризиком (risk management) – це систематичне застосування політики, процедур і практичних методів управління для задач аналізування, оцінювання, контролювання і моніторингу ризику [2].

Поняття медичної системи (в розумінні медичного виробу) включає в себе широкий спектр обладнання, інструментів, приладдя тощо, конструкція і принципи роботи яких значно між собою відрізняються. Відповідно, рівень потенційного ризику застосування різних типів медичних виробів не може бути однаковий [8]. З огляду на це, була розроблена класифікація медичних виробів залежно від потенційного ризику застосування.

Розрізняють наступні класи медичних виробів: Клас I, Клас IIa, Клас IIb, Клас III, при цьому, чим вищий клас, тим вище рівень потенційного ризику застосування (таблиця 1.1) [5].

У загальному випадку класифікація медичних виробів заснована на критеріях, які характеризують призначення, конструкцію і принципи роботи медичних виробів. До них відносяться:

- тривалість застосування медичного виробу;
- інвазивність медичного виробу;
- наявність контакту з людським тілом або взаємозв'язку з ним;
- спосіб введення медичного виробу в тіло людини (через анатомічні порожнини або хірургічним шляхом);
- застосування для життєво важливих органів і систем (серце, центральна система кровообігу, центральна нервова система);
- застосування джерел енергії [9].

При цьому, треба враховувати що класифікація за потенційним ризиком застосування не має відношення до результатів процесу управління ризиком, але коректність визначення класу ризику застосування є основою для правильного вибору процедур з управління ризиком [5].

Таблиця 1.1 – Класифікація медичних виробів залежно від потенційного ризику застосування [8]

Клас	Опис	Принцип застосування	Ступінь інвазивності	Час застосування
Клас I	Відсутні методичні ризики	Відсутній або не критичний контакт з поверхнею шкіри	Низька ступінь	< 60 хвилин
Клас IIa	Ризик застосування	Короткочасне застосування в організмі (очі, кишківник, хірургічні отвори)	Середня ступінь	< 30 днів
Клас IIb	Підвищений методичний ризик застосування	Системні порушення	Концепція не інвазивного управління	> 30 днів
Клас III	Високий методичний ризик	Вироби для прямого застосування на органах, центральній нервовій системі, системі кровообігу	Концепція інвазивного управління	Довготривало застосування

Конкретні кроки управління ризиком можуть відрізнятися, але обов'язково мають бути присутніми ідентифікація небезпек, оцінка ризику та

при необхідності розробка рекомендацій по зниженню ризику. Розглянемо етапи управління ризиком більш детально.

1.2 Етапи процесу управління ризиками

Схематичне зображення процесу управління ризиком для медичних виробів наведено на рисунку 1.1 [2].

До початку проведення дій з управління ризиком необхідно чітко окреслити галузь застосування: визначити об'єкт оцінки ризиків та активи, що представляють цінність.

Стосовно медичної сфери об'єктом оцінки ризику може бути як окремий медичний виріб, так і процеси, пов'язані з ним.

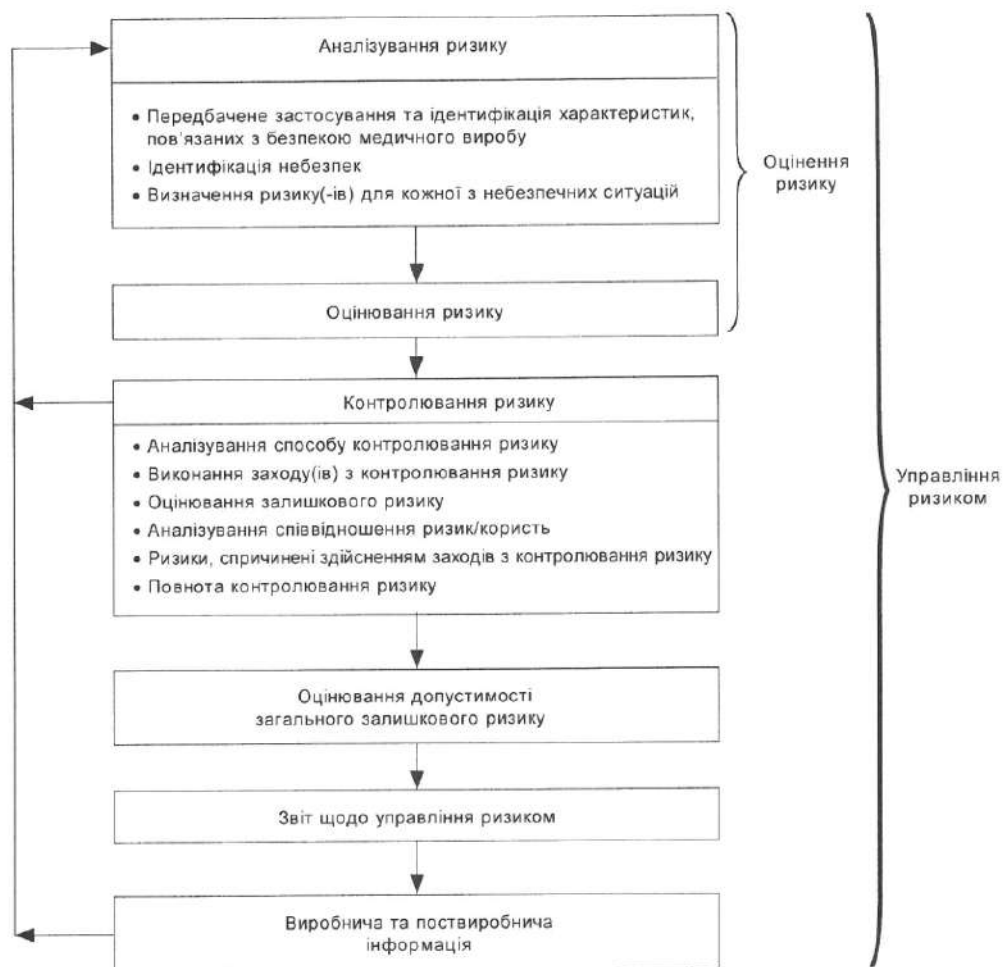


Рисунок 1.1 – Схематичне зображення процесу управління ризиком[2]

При визначенні активів слід враховувати, що основу будь-якого об'єкту чи процесу складають виконувані функції, для здійснення яких потрібні ресурси:

- людські – учасники процесу;
- виробничі;
- матеріальні (в тому числі навколишнє середовище);
- інформаційні;
- інтелектуальні – знання і повноваження учасників.

Управління ризиками для медичних виробів та процесів перш за все має першорядне значення для людських ресурсів: пацієнтів, операторів та інших осіб, що беруть участь в медичному процесі, та для навколишнього середовища.

1.2.1 Аналізування ризиків

Після того, як була окреслена галузь застосування, проводять роботи з аналізування ризику, за якими встановлюють межі передбаченого застосування та ідентифікацію характеристик, пов'язаних з безпекою.

Згідно вимог [2] виробник має задокументувати призначене застосування медичного виробу та передбачуване неправильне застосування. Також виробник визначає характеристики (якісні та кількісні), які можуть впливати на безпеку, та, за можливістю, рівень їх допустимого значення.

В результаті отриманні матеріали стають базисом для ідентифікації небезпек.

Небезпека може бути створена завдяки небезпечній ситуації за умови деяких обставин. Тому треба звертати увагу на послідовності або комбінації подій, що можуть викликати небезпечну ситуацію як за умови нормального стану виробу, так і у стані збою.

Далі для кожної небезпечної ситуації, що була ідентифікована, слід визначити ризик, до якого така ситуація може призвести.

Складові частини ризику, вірогідність та тяжкість наслідків, зазвичай розглядаються окремо.

На практиці доцільно використання дискретної кількості рівнів вірогідності, що значно спрощує процес аналізування. Особливо ефективно присвоєння кількісних значень дискретним рівням. Те саме стосується й ідентифікації тяжкості потенційної шкоди, для якої також доцільно використання дискретних кількісних рівнів тяжкості [2].

Приймається рішення щодо необхідної кількості якісних категорій та їх визначення. Вони можуть бути представлені лінгвістично (наприклад, не потребує медичного втручання, потребує медичного втручання, потребує госпіталізації, призводить до летальних наслідків та ін.) або символічно (L1, L2 тощо). При символічному визначенні кожен символ має бути чітко визначений.

При будь-якому представленні кожна з категорій тяжкості потенційної шкоди не повинна містити елементів ймовірності.

Хоча кількісна оцінка ризику і більш прийнятна, але за умови неможливості її проведення можуть бути достатніми якісні методи визначення ризику.

Коли вірогідність та ступінь тяжкості нанесення шкоди неможливо розрахувати за відсутності достовірної кількісної інформації використовують інші способи визначення ризику.

Наприклад, використовують матриці для опису вірогідності та тяжкості ризиків у кожній небезпечній ситуації.

Чітко визначають кількість рівнів як вірогідності, так і тяжкості та їх наповнення. В таблицях 1.2 та 1.3 представлені приклади якісних рівнів тяжкості та вірогідності [2].

Таблиця 1.2 – Якісні рівні тяжкості

Рівні тяжкості	Опис
Значний	Летальний випадок або зміна анатомічної будови тіла
Помірний	Зворотні чи незначні ушкодження
Незначний	Не призводить до ушкоджень або призводить до незначних ушкоджень

Таблиця 1.3 – Якісні рівні вірогідності

Рівні вірогідності	Опис
Високий	Можливий, трапляється часто
Середній	Можливий, але не часто трапляється
Низький	Навряд трапляється, зрідка

Далі формується матриця ризиків, де в найменуваннях рядків представлені лінгвістичні значення рівнів вірогідності, а в найменуваннях стовпців – рівні тяжкості. До кожної комірки матриці вносять відповідні очікувані ризики. Приклад матриці ризиків наведено у таблиці 1.4.

Таблиця 1.4 – Приклад матриці ризиків

		Якісні рівні тяжкості		
		Значний	Помірний	Незначний
Якісні рівні вірогідності	Високий		R ₂	R ₁
	Середній	R ₃	R ₄	
	Низький			R ₅ , R ₆

Слід враховувати, що матриці з більшим числом якісних рівнів дозволяють отримати більшу точність. Але при цьому матриці, що містять більше ніж п'ять рівнів потребують значно більше інформації для присвоєння ризику до того чи іншого рівня.

Обґрунтування щодо розрядності матриці ризиків, що складається з кількості рівнів вірогідності та кількості рівнів тяжкості наслідків, повинно бути чітко задокументовано виробником [2].

1.2.2 Оцінювання ризику

Для кожного визначеного ризика приймається рішення щодо його допустимості та необхідності його зменшення. Роблять це за допомогою критеріїв допустимості ризиків. Виробник може класифікувати ризики за ступенем допустимості для визначення тих з них, які необхідно зменшити.

Для визначення допустимості ризиків виробник може використовувати відповідні стандарти на конкретний тип виробу (за їх наявності), порівняння рівнів ризику виробів, що знаходяться в експлуатації, дані клінічних досліджень, наявну достовірну інформацію про досвід інших виробників з доступних джерел [2].

Застосування критеріїв допустимості може бути відображено у матриці ризиків. Наприклад за допомогою кольору, як показано у таблиці 1.5.

Таблиця 1.5 – Приклад матриці оцінювання ризиків з колірним показником критерію допустимості

		Якісні рівні тяжкості		
		Значний	Помірний	Незначний
Якісні рівні вірогідності	Високий		R ₂	R ₁
	Середній	R ₃	R ₄	
	Низький			R ₅ , R ₆

Позначки:



– недопустимий ризик;



– допустимий ризик.

Слід зазначити, що зона матриці може бути поділена на більше ніж дві зони критерію допустимості. Зробити це можна за допомогою класифікації кожного визначеного ризику до певної категорії або класу.

1.2.3 Контролювання ризику

На наступному етапі визначають заходи, необхідні для зменшення ризиків до прийнятного рівня. Такі заходи ще називають контрзаходами. Вони здатні зменшувати тяжкість можливої шкоди або її вірогідність, або обидві ці складові ризику разом.

Стандартними способами зі зменшення ризику згідно [10] є:

- безпека передбачена конструкцією;
- запобіжні заходи у самому медичному виробі або виробничому процесі;
- інформування щодо безпеки.

Після визначення всіх необхідних заходів щодо зниження підсумкового ризику їх впроваджують до виробу або процесу, що повинно бути обов'язково перевірено.

Слід враховувати, що деякі з визначених та впроваджених заходів зниження ризиків можуть внести додаткові умови для створення небезпечної ситуації з заподіянням шкоди. Такими новими або збільшеними ризиками, що є наслідком вжитих заходів по контролюванню ризиків, необхідно управляти так само як і будь якими іншими, відповідно до зазначених вище методів та підходів.

Після впровадження заходів зі зниження ризику необхідно оцінити залишковий ризик, використовуючи ті ж самі методи та критерії допустимості ризиків, які були використані на етапі аналізування ризиків.

Якщо ж залишковий ризик після повторного оцінювання попадає в зону допустимого ризику, то виробник має визначити які саме з залишкових ризиків необхідно зазначити, яку інформацію, в якій формі та наскільки

детально необхідно викласти в супровідних документах. Робиться це задля пояснення залишкового ризику так, щоб користувачі активно вживали відповідних заходів для мінімізації впливу залишкового ризику.

Якщо ж залишковий ризик все ж таки залишається в зоні неприпустимих ризиків слід застосувати додаткові заходи зі зниження ризику. Ця багаторазова процедура повторюється доки залишковий ризик не буде в зоні допустимого.

Але ймовірна ситуація, коли практичний спосіб зменшення ризику до допустимого рівня відсутній або є недоцільним, тоді виконують аналізування співвідношення користь/ризик, для визначення переваги медичної користі застосування виробу над залишковим ризиком. Для цього застосовують всі зібрані і перевірені дані та наявну літературу.

Якщо докази свідчать, що користь використання медичного виробу не перевищує залишковий ризик, то такий ризик вважають неприпустимим і від такого проекту треба відмовитися.

Однак іноді значні ризики можна вважати допустимими, якщо медичний виріб приносить більше користі ніж шкоди. В такому випадку виробник документально обґрунтовує залишковий ризик з зазначенням заходів безпеки по використанню виробу [2].

При необхідності проведення клінічних досліджень для встановлення допустимості співвідношення користі та залишкового ризику вони регулюються [11]-[12], що дозволяє отримати кількісну оцінку ризиків та користі.

1.2.4 Оцінювання допустимості загального залишкового ризику

Після проведення всіх попередніх етапів виробник повинен поглянути на процес управління ризиками з точки зору широкої перспективи і оцінити сукупний вплив залишкових ризиків та прийняти рішення щодо подальшої доцільності роботи над виробом. Ймовірна ситуація коли загальний

залишковий ризик перевищує встановлені критерії допустимості, тоді як окремі залишкові ризики не перевищували таких критеріїв. Особливо це стосується складних систем з великою кількістю можливих ризиків та їх комбінацій.

Такий аналіз повинні здійснювати спеціалісти, що є експертами в конкретній предметній області та мають відповідний досвід та повноваження.

Рішення щодо доцільності враховує, що всі можливі ризики повинні бути знижені до найнижчого можливого рівня за умови практичної здійсненності.

Практична здійсненність складається з технічної здійсненності та економічної доцільності.

Технічна здійсненність – це здатність виробника знизити ризик незалежно від фінансових витрат.

Економічна доцільність передбачає зниження ризику за умови, що медичний виріб залишається економічно вигідним проектом [2].

1.2.5 Звіт щодо управління ризиком та виробнича і пост виробнича інформація

В результаті всіх проведених робіт формують звіт щодо управління ризиками, де коротко викладають остаточні результати процесу управління ризиком.

Однак на цьому робота не закінчується. Аналіз і оцінка ризиків використовується для забезпечення безперервного управління ризиками.

Принципи управління ризиком повинні бути застосовані з моменту зародження медичного процесу і впродовж всього його існування.

Виробники повинні здійснювати моніторинг виробничої і пост виробничої інформації для отримання даних, що можуть впливати на визначення ризиків, а отже і можуть бути причиною перегляду звіту та покращення процесу управління ризиками. Використання пост виробничої

інформації дозволяє зробити процес управління ризиком повторюваним зі зворотнім зв'язком [2].

1.3 Методи оцінки та управління ризиками

Існують загальні рекомендації щодо доцільності застосування конкретних методів на різних етапах аналізу та оцінки ризику. Вибір відповідного методу залишається за виробником [2].

Аналіз ризиків можна умовно розділити на два взаємодоповнюючі один одного види: якісний і кількісний. Якісний аналіз має на меті ідентифікувати фактори, області й види ризиків. Кількісний аналіз ризиків повинен дати можливість чисельно визначити розміри окремих ризиків та ризику об'єкта в цілому [13].

1.3.1 Якісний аналіз ризиків

Вибір якісного методу аналізу небезпек залежить від мети аналізу, призначення об'єкта, його стану й складності, наприклад:

- попередній аналіз небезпек (Preliminary Hazard Analysis – PHA) [14];
- методи перевірконого листа (Check-list) та «Що буде, якщо ...?» («What – If») [15];
- метод аналізу небезпеки та працездатності (Hazard and Operability Study – HAZOP) [16];
- аналіз видів та наслідків відмов (Failure Mode and Effects Analysis – FMEA та Failure Mode, Effects and Critical Analysis – FMECA) [17];
- графічно-аналітичні методи аналізу небезпек за допомогою дерев: дерева відмов (Fault Tree Analysis – FTA); дерева подій (Event Tree Analysis – ETA); дерева рішень, причин, наслідків тощо [15], [18];
- аналіз помилок персоналу та причинно-наслідковий аналіз [15];

– логіко-ймовірнісне моделювання [19] та багато інших.

Зазвичай вказані методи не використовуються по одинці. Для досягнення найбільшої ефективності якісного аналізу ризиків проводять покрокове аналізування послідовності подій з застосуванням комбінації декількох методів.

1.3.2 Кількісний аналіз ризиків

При кількісному аналізі під терміном “ризик” розуміють багатокомпонентну величину, яка характеризується нанесенням шкоди від впливу того або іншого небезпечного фактору, імовірністю виникнення розглянутого фактору та невизначеністю у величинах як шкоди, так і ймовірності.

Згідно класичному визначенню ризик характеризується як:

$$R = P \cdot U \quad (1.1)$$

де P – імовірність події ;

U – наслідки (наприклад, у вигляді шкоди).

Залежно від мети аналізу для чисельної оцінки ризику можуть використовувати й інші математичні формулювання [13].

Проте необхідно враховувати, що здійснення кількісної оцінки зустрічає і найбільших труднощів, які пов'язані з тим, що для цього виду аналізу необхідна відповідна вхідна достовірна інформація. А для рідкісних подій та для нових розробок, які за своєю суттю є унікальними виробами з новими функціональними властивостями і взаємозв'язками між елементами конструкції, низкою нововведень, що до цього не використовувалися, з одного боку немає достатньої статистики, щоб спиратися на досвід, а з іншого – немає теорії, яка дозволяла б виводити ці величини, виходячи з первинних принципів [13].

В цьому випадку на ряду з клінічними дослідженнями можливо застосування експертного підходу: ймовірності різноманітних подій, зв'язки між ними і наслідки заподіяної шкоди визначають не обчисленнями, а опитуванням досвідчених експертів з використанням тих самих матриць ризику.

1.3.3 Методи експертного оцінювання

Методи експертних оцінок – це методи організації роботи з фахівцями, експертами, і обробки думок експертів.

Основним завданням експертного оцінювання є:

- забезпечення адекватної оцінки об'єкта експертизи;
- виділення альтернативних варіантів рішень, що можуть бути реалізовані та здатні привести до мети;
- обґрунтований вибір самої прийнятної з альтернатив [20].

Колективне експертне оцінювання дозволяє знизити суб'єктивність індивідуальних думок експертів і підвищити ефективність прийняття колективного рішення.

У загальному випадку процедура підготовки та прийняття колективного рішення може бути розбита на наступні етапи: попередній аналіз проблеми та постановка задачі, розробка регламенту проведення експертизи, формування експертної групи, отримання індивідуальних експертних оцінок, перевірка узгодженості думок експертів і формування колективної оцінки, прийняття колективного рішення.

На кожному з цих етапів застосовуються різні методи, які умовно можна розбити на наступні групи:

- методи колективного обговорення, що використовуються для попереднього аналізу проблем (інтерв'ю, анкетування, метод "мозкового штурму", метод Дельфі тощо);

- методи формування груп експертів і визначення компетентності експертів (метод самооцінки, метод взаємооцінки, метод оцінки за формальними показниками анкети, тестування, теорія прецедентів, методи, засновані на результатах колишньої діяльності експертів тощо);

- методи оцінки узгодженості думок експертів (методи знаходження коефіцієнтів конкордації, коефіцієнт рангової кореляції Спірмена і Кендалла тощо);

- методи формування колективної експертної оцінки (методи обчислення середніх (середньоарифметичне, середньгеометричне, медіана, мода тощо), Метод медіани Кемені, обчислення середнього рангу та ін.);

- методи оцінювання альтернатив (метод аналізу ієрархій, методи ранжування альтернатив та ін.) [21].

У методах оцінювання альтернатив широко поширені такі види експертних суджень:

- парні порівняння. З наявної сукупності об'єктів експерту по черзі пред'являються можливі пари об'єктів. Для кожної пари експерт вказує, який з двох об'єктів краще іншого за певною ознакою;

- множинні порівняння. Із сукупності об'єктів складаються не пари, а набори об'єктів. Експерт впорядковує об'єкти в наборі в порядку переваги;

- ранжування. Експерт впорядковує наявні об'єкти за зменшенням (або за зростанням) з точки зору заданої ознаки;

- класифікація. Експерт розділяє всю сукупність об'єктів на кілька (можливо, заздалегідь заданих) класів;

- безпосередня оцінка. Експерт приписує наявним об'єктам числові значення в шкалі значень даної ознаки [20].

Для конкретного завдання експертизи можуть використовуватися різні види суджень або їх поєднання.

1.4 Огляд існуючих програмних засобів підтримки процесу управління ризиками

Процес управління ризиків пов'язаний з достатньо складним поєднанням різноманітних активів, вразливостей, небезпек, захисних заходів, що породжує сотні та тисячі можливих комбінацій, які описують ризики.

На допомогу може прийти програмне забезпечення для проведення аналізу та оцінки ризиків, що призначене істотно підвищити ефективність процесу управління ризиками, забезпечити скорочення часу проведення процедур аналізу та оцінки ризиків за рахунок прискореної обробки великого обсягу інформації та підвищення її точності.

Об'єктивність і достовірність інформації може бути забезпечена тільки в тому випадку, коли роль суб'єктивного фактору, викликаного впливом людини на процес обміну даними, зводиться до мінімуму [22].

В таблиці 1.6 наведений порівняльний аналіз існуючих програмних засобів для оцінки ризиків.

Таблиця 1.6 – Програмні продукти з оцінки ризику

	Назва продукту	Розробник	Особливості призначення	Переваги	Недоліки
1	2	3	4	5	6
1	vsRisk [23-24]	Vigilant Software	призначений для оцінки ризиків інформаційної безпеки згідно з вимогами ISO 27001 та ISO 27005	1) має якісні шкали оцінки, що налаштовуються користувачем; 2) можливо отримати тріальну версію на 15 днів	1) складний інтерфейс для непідготовленого користувача; 2) наявність великих списків можливих загроз, вразливостей і контрзаходів, але при цьому сама система не визначає ніяких взаємозв'язків між ними, це робиться вручну користувачем; 3) відсутня кількісна оцінка ризиків; 4) помилки відображення кирилиці; 5) велика ціна

Продовження таблиці 1.6

1	2	3	4	5	6
2	RA2 art of risk [24]	AEXIS Security Consultants, XiSEC Consultants Ltd	є ефективною системою підтримки прийняття рішень з управління інформаційним і ризиками для сучасного бізнесу, підтримує вимоги ISO 27001, ISO 27005, ISO 17799	1) реалізований простий для розуміння процесний підхід; 2) процес управління ризиками може налаштовуватися під потреби конкретної організації	1) не достатньо опрацьований користувацький інтерфейс; 2) примітивність засобів роботи з моделлю активів, загроз, вразливостей; 3) для успішної оцінки і управління ризиками необхідно збирати інформацію з різних джерел в організації; 4) помилки відображення кирилиці в звітах
3	PTA [23]	PTA Technologies	не прив'язаний до якогось стандарту і реалізує механізм кількісної оцінки ризиків	1) передбачений цікавий механізм оцінки ефективності пропонованих контрзаходів, що дозволяє побачити як змінюється карта ризиків при додаванні чи прибиранні тих чи інших контрзаходів 2) можливо отримати тріальну версію на 30 днів	не передбачена можливість якісної оцінки ризиків
4	Proteus Enterprise [24]	InfoGov	потужна система не тільки для управління ризиками, але і для підтримки процесів управління інформаційною безпекою, що підтримує багато стандартів	1) масштабується від версії на одного користувача до версії для багатьох користувачів; 2) постачається разом з набором шаблонів опитувальників 3) дозволяє проводити онлайн аудити у внутрішніх підрозділах і у зовнішніх постачальників	масштабність продукту направлена на керування бізнесом, де оцінка ризиків одна з процедур

Продовження таблиці 1.6

1	2	3	4	5	6
5	Modulo Risk Manager [23, 25]	Modulo	призначений для поглибленої оцінки ризиків та проектів відповідності	1) підтримка багатьох стандартів; 2) кількісна оцінка ризику; 3) інвентаризація та оцінення активів; 4) збирання та централізація даних, що відносяться до технологічних активів (програмне забезпечення та обладнання) та нетехнологічних активів (люди, процеси та фізичні об'єкти) 5) продукт відзначений нагородою SC Magazine	відсутність тріальної версії, ніяких докладних відео-роликів, досить бідний опис функціоналу
6	RM Studio [23, 26]	RM Studio	інтегрована система управління ризиками соціально-технічних систем, зі зростаючою складністю	можливо отримати тріальну версію на 30 днів, доступні відео-ролики, що демонструють сценарії використання продукту	занадто примітивний функціонал в плані оцінки ризиків, не для суттєвого аналізу
7	РискДетектор [27]	Институт системного анализа РАН	система автоматизації управління ризиками, аудиту, контролю, моніторингу безпеки банківських та інших критичних систем, інфраструктур і бізнес-процесів	1) автоматизує побудову різноманітних моделей; 2) розрахунки ризиків порушення безпеки, залишкових ризиків 3) контроль якості вимог до безпеки системи	не повідомляється

Продовження таблиці 1.6

1	2	3	4	5	6
8	OMNITRACKER Risk Management [28]	OMNINET Software Solutions	Структуроване оцінювання ризиків за вільно визначеними категоріями. Підтримка вимог до управління ризиками відповідно до стандартів ISO 31000	1) визначення контрзаходів та автоматичний контроль графіка їх впровадження; 2) простий для розуміння і розширюваний каталог ризиків; 3) створення документації на основі історично зібраної інформації; 4) адаптована концепція прав і ролей з чіткою відповідальністю за кожен етап процесу управління ризиками	не повідомляється
9	RiskWatch [29]	RiskWatch International	платформа для оцінки ризиків різноманітних галузей	1) простота використання; 2) поєднання кількісного та якісного аналізу; 3) широка база знань по загрозам, вразливостям та контрзаходам, з можливістю поповнення; 4) налаштування звітів; 5) формування листів опитування для отримання даних	достатньо висока ціна

Комерційних програмних продуктів для проведення оцінки ризиків на світовому ринку достатньо багато. До числа більш поширених недоліків програмних продуктів, що обмежує їх практичне застосування, можна віднести наступні:

- складність використання продукту невідготуваному користувачу, інтуїтивно незрозумілий інтерфейс;
- неповне охоплення активів;

- ускладнення процесу усвідомлення ризику через автоматичний розрахунок ризиків, який скритий від користувача;
- проблеми відображення кирилиці для імпортованих програмних продуктів;
- достатньо висока комерційна ціна.

Однак основним їх недоліком для використання оцінки ризиків медичних систем є відсутність орієнтації на медичну інфраструктуру та, як наслідок, відсутність баз знань з типовими для цієї галузі моделями активів, загроз, вразливостей тощо. Також відсутня реалізація можливості групового експертного оцінювання.

Зазначені вище недоліки свідчать про актуальність і своєчасність проведення даної розробки.

1.5 Постановка завдання

Як було зазначено вище аналіз та оцінка ризиків медичних систем є досить трудомісткими та відповідальними етапами процесу управління ризиками. Автоматизація деяких ланок цього процесу здатна значно спростити та підвищити точність, актуальність та достовірність отриманих даних. На сьогоднішній день відсутнє програмне забезпечення, яке б задовольнило потреби в підтримці процесу управління ризиками в медичній галузі, зокрема в оцінюванні ризиків медичних систем та їх процесів.

Тому метою даної роботи є дослідження методів оцінювання ризиків медичних виробів, що дозволить створити програмну систему для автоматизації та спрощення проведення процедури оцінювання ризиків для медичних систем та їх процесів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити етапи аналізу та оцінювання ризиків медичних систем та їх процесів;
- проаналізувати експертні методи оцінювання ризиків;

- зробити вибір мови програмування та середовища розробки;
- розробити архітектуру та реалізувати програмний продукт.

2 РОЗРОБКА ТА УДОСКОНАЛЕННЯ МЕТОДІВ Й АЛГОРИТМІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Визначення матриці ризиків

На етапі аналізування ризиків для кожного з ідентифікованих ризиків слід визначити його складові: ймовірність та тяжкість наслідків.

За відсутності достовірної кількісної інформації вірогідність та ступінь тяжкості нанесення шкоди неможливо розрахувати, тому будемо використовувати якісні методи оцінки.

Як було зазначено в першому розділі доцільно це робити за допомогою якісних дискретних рівнів. Їх кількість слід встановити виходячи з необхідної точності отриманих результатів.

Оскільки три якісні рівні для вірогідності та ступеня тяжкості наведені у таблицях 1.2 та 1.3 дають досить грубий результат при оцінці ризиків було прийнято рішення збільшити кількість рівнів до п'яти, які наведено в таблицях 2.1 та 2.2 [4].

Таблиця 2.1 – Якісні рівні тяжкості

Рівні тяжкості	Опис
1	2
Критична	Летальний результат або зміна анатомічної будови тіла, незворотні пошкодження користувача або пацієнта
Значна	Захворювання, серйозне пошкодження, що закінчилося ускладненням і значним погіршенням здоров'я
Помірна	Захворювання, серйозне пошкодження, що закінчилося повним одужанням і не спричинило ускладнень

Продовження таблиці 2.1

1	2
Низька	Незначні пошкодження без ускладнень і погіршення здоров'я, нанесення значної моральної або матеріальної шкоди
Дуже низька	Відсутність пошкоджень і погіршення здоров'я, але нанесення незначної моральної або матеріальної шкоди

Таблиця 2.2 – Якісні рівні вірогідності

Рівні вірогідності	Опис
Дуже висока	Велика ймовірність виникнення
Висока	Буде виникати не один раз
Низька	Ймовірні поодинокі випадки
Дуже низька	Ймовірне виникнення
Малоймовірно	Малоймовірно, але можливо

Перед тим як сформувати матрицю ризиків слід визначити необхідну кількість критеріїв допустимості – категорій ризику, які будемо називати ранги. У таблиці 2.3 наведено лінгвістичний опис обраних рангів ризику.

Таблиця 2.3 – Ранги ризиків [12]

Ранги ризиків	Опис рангів ризиків
I	Неприпустимий ризик
II	Небажаний ризик, який може бути допустимим тільки, якщо зниження ризику непрактично й веде до непропорційного збільшення витрат у порівнянні з досягнутим поліпшенням
III	Допустимий ризик, якщо витрати на зниження ризику не перевищують досягнутих поліпшень
IV	Нехтовно малий ризик

Остаточно сформована матриця ризиків, за допомогою якої будуть проводити оцінювання ризиків, наведено в таблиці 2.4. Застосовані критерії допустимості у матриці ризиків відображені кольором.

Таблиця 2.4 – Матриця оцінювання ризиків, що поділена на чотири зони за критеріями допустимості

		Якісні рівні тяжкості				
		Критична	Значна	Помірна	Низька	Дуже низька
Якісний рівні вірогідності	Дуже висока					
	Висока					
	Низька					
	Дуже низька					
	Малоймовірно					

Позначки:



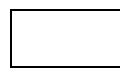
– I ранг;



– II ранг;



– III ранг;



– IV ранг

2.2 Вибір методів експертного оцінювання

Застосування експертного підходу при оцінюванні ризиків медичних систем та їх процесів обумовлено відсутністю необхідної кількості статистичних даних, які могли б бути застосовані при кількісній оцінці ризиків.

До цього процесу долучається група експертів – фахівців з великим досвідом в сфері медичних виробів, що дозволяє знизити суб'єктивність індивідуального судження кожного з експертів та підвищити точність колективного рішення.

2.2.1 Загальна постановка задачі експертного оцінювання

Розглянемо постановку задачі експертного оцінювання в загальному вигляді. Нехай існує множина об'єктів оцінювання $X=\{x_i\}$, $i=1,2,\dots,n$. Тоді Ω –

множина припустимих оцінок цих об'єктів, а $C(\Omega)$ – функція вибору, яка зіставляє кожній підмножині $\Omega' \subseteq \Omega$ її частину. Кожний з p експертів оцінює n альтернатив $a_{ij} = C_{ij}(\Omega) \in \Omega$ як , де $i=1,2,\dots,n$ та $j=1,2,\dots,p$.

В результаті колективного експертного оцінювання формується матриця $n \times p$ $A^{coll} = \|a_{ij}\|$ оцінок i -го об'єкту j -м експертом.

При цьому в залежності від задачі, що вирішується, множина припустимих значень визначається як $\Omega = \{0, 1\}$, може бути визначено завданням попарного порівняння об'єктів a' та b' (2.1)-(2.2):

$$C(\Omega) = \begin{cases} 1, & \text{якщо } a' \text{ краще } b', \\ 0, & \text{якщо інакше.} \end{cases} \quad (2.1)$$

$$\Omega = \{\langle 1, 2, \dots, m \rangle, \langle 1, 3, \dots, m, 2 \rangle, \dots, \langle m, m-1, \dots, 1 \rangle\}, \quad (2.2)$$

яке складається з множини перестановок довжини m , що є характерним для вирішення задачі ранжування. Остання полягає в упорядкуванні об'єктів, що утворюють систему значень ознак. $C(\Omega) = \langle i_1, i_2, \dots, i_m \rangle$, де i_j – номер j -го об'єкту при вказаному упорядкуванні.

$\Omega = \{1, \dots, l\}$. Задача класифікації може полягати в співвідношенні $x \in S$ до однієї з l підмножин $S_1, \dots, S_l$, де $S = S_1 \cup \dots \cup S_l$, та при цьому $C(\Omega) = i$, якщо $x \in S_i$.

$\Omega = E_m$. Відповідне завдання чисельної оцінки може полягати у зіставленні об'єкту оцінювання одного або декількох чисел, причому $C(\Omega) = d$, якщо оцінкою об'єкта є вектор $d \in E_m$ [20].

2.2.2 Оцінка узгодженості думок експертів

При оцінці об'єктів дослідження необхідна оцінка ступеня узгодженості думок експертів. Отримання кількісної міри узгодженості дозволяє більш обґрунтовано інтерпретувати причини розбіжності думок [30].

Для того щоб оцінити узгодженість думок декількох експертів при оцінюванні ряду об'єктів (альтернатив) можна скористатися коефіцієнтом конкордації (узгодженості) Кендалла і Сміта [30], що широко застосовується при ранжуванні (2.3)-(2.4):

$$W = \frac{12 \cdot S_w}{p^2(n^3 - n)}, \quad (2.3)$$

$$S_w = \sum_{j=1}^n \left(\sum_{i=1}^p a_{ij} - \frac{p(n+1)}{2} \right)^2, \quad (2.4)$$

де a_{ij} – ранг i -го об'єкту, який присвоїв йому j -й експерт;

p – кількість експертів;

n – кількість об'єктів (альтернатив), що оцінюються.

При наявності зв'язних рангів (як в нашому випадку), тобто присутності збігу місць, коефіцієнт конкордації (2.3) набуде вигляду (2.5):

$$W = \frac{12 \cdot S_w}{p^2(n^3 - n) - p \sum_{j=1}^p T_j} \quad (2.5)$$

$$T_j = \sum_{i=1}^k (t_i^3 - t_i), \quad (2.6)$$

де k – число зв'язаних (об'єднаних) рангів;

t_i – кількість елементів у i -й зв'язці для j -го експерта.

Отримані величини порівнюють з критичними значеннями при заданій довірчій ймовірності ($\alpha = 0,95$ и $\alpha = 0,99$). Критичні значення $S_w(\alpha)$ визначають за довідником [30].

При $n > 10 \div 15$ і відсутності кореляції величина $p(n-1)$ W_α розподілена приблизно як χ^2 зі $f = n-1$ ступенями свободи. Отже критичне значення дорівнює (2.7):

$$W_\alpha = \frac{\chi_\alpha^2}{p \cdot (n-1)} \quad (2.7)$$

Таким чином, оцінка думок експертів за допомогою коефіцієнтів конкордації при колективному прийнятті рішень дозволяє виявити і об'єднати різні підходи експертів до оцінки різних явищ (ознак, критеріїв), що дозволяє провести поглиблений аналіз ситуації і прийняти більш зважене обґрунтоване рішення.

2.3 Розробка алгоритму роботи програми

Даний програмний продукт призначений облегшити роботу особи, що приймає рішення (ОПР) під час оцінювання ризиків медичних систем та їх процесів, за рахунок проведення колективного експертного оцінювання.

Для облегшення комунікації між користувачами та зменшення часу проведення експертизи програмний продукт доцільно розробити у вигляді вебзастосунку.

Виходячи з призначення програмного продукту та функціональної необхідності повинно бути організоване розмежовування прав доступу різних груп користувачів (за допомогою персоналізованого входу в систему). Передбачається наступні групи користувачів – експерти, ОПР, адміністратор.

На початку роботи ОПР створює новий об'єкт оцінювання ризиків з заповненням всієї необхідної інформації про нього. Далі для об'єкту оцінювання ризиків проводить ідентифікацію можливих небезпек та ризиків,

що вони викликають, із запропонованого списку типових варіантів. При необхідності додає нові небезпеки та ризики до списку типових варіантів.

Після того, як об'єкт оцінки ризику повністю описаний ОПР ініціалізує проведення експертизи. Встановлює строки проведення експертизи, формує групу експертів та надсилає їм листи з запрошенням до експертизи.

Після отримання листа з запрошенням до прийняття участі в експертизі кожен з експертів повинен зайти до застосунку та провести експертну оцінку шляхом заповнення матриці ризиків.

Отримавши результати роботи експертної групи ОПР проводить перевірку узгодженості думок експертів.

У випадку неузгодженості думок ОПР приймає рішення щодо проведення нової експертизи з уточненими завданням або представників експертної групи.

Якщо ж думки експертів виявилися узгодженими, проводиться усереднення оцінки експертів та формування звіту щодо проведеного експертного оцінювання ризиків.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

Виходячи з поставленого завдання, для забезпечення функціональності було прийняте рішення розробити програмний комплекс з застосуванням вебтехнологій. Особливість реалізації полягає в тому, що логіка застосунку зосереджується на сервері, а функцією браузера є відображення інформації, завантаженої мережею з сервера, і передача назад даних користувача. При такому підході клієнти не залежать від конкретної операційної системи користувача, тому вебзастосунки є міжплатформовими сервісами [31].

Створення вебзастосунку починається з розробки архітектури.

3.1 Вибір та обґрунтування загальної архітектури програми

Було обрано триланкову модель архітектури. Кількість ланок відповідає трьом компонентам: клієнт, сервер та серверна база даних. Схема зазначена на рисунку 3.1.

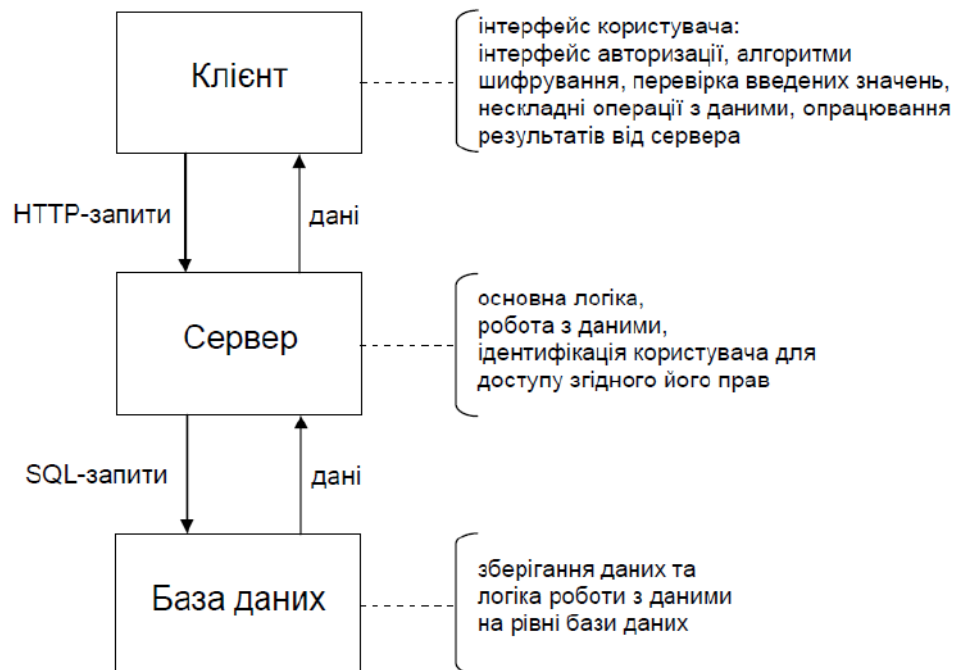


Рисунок 3.1 – Триланкова модель архітектури програми

До переваг вказаної архітектури відносять високу гнучкість, розширюваність, незалежність користувача, готовність і низька вартість відновлення [32].

Верхній рівень триланкової архітектури відповідає за інтерфейс користувача. На цей рівень зазвичай виносять інтерфейс авторизації для обмеження неконтрольованого доступу, алгоритми шифрування, перевірка введених значень на допустимість і відповідність формату, нескладні операції з даними та попередня їх обробка перед відправкою на сервер (сортування, угруповання, підрахунок значень), а також опрацювання результатів від сервера [33].

Користувач під час взаємодії з вебсайтом, що є інтерфейсом клієнтської частини, може здійснювати налаштування програми та перегляд результатів роботи.

Головним рівнем є рівень сервера, на якому зосереджена вся основна логіка, здійснюється робота з даними, ідентифікація користувача для надання йому персонального доступу згідного його прав. Для забезпечення безпеки та цілісності даних сервер є єдиним зв'язком між користувачем та базою даних

Рівень бази даних забезпечує зберігання даних, які будуть використані сервером і при потребі передані на користувацький рівень. Також може бути реалізована нескладна логіка, яка не використовує нічого окрім самої бази даних та її таблиць. Завдяки використанню зовнішніх зв'язків та ключів підтримується цілісність даних.

3.2 Архітектура серверу

Для проектування серверної частини застосунку було використано інтерфейс прикладного програмування API.

API (англ. Application Programming Interface) набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено – це набір чітко визначених методів для взаємодії

різних компонентів, що надає розробнику засоби для швидкої розробки програмного забезпечення за рахунок доступу до функціональності програмного компонента (програми, модуля, бібліотеки) [34].

API є важливою абстракцією, що описує функціональність «у чистому вигляді». При цьому API дозволяє абстрагуватися від того, як саме ця функціональність реалізована. Крім того, API є важливою абстракцією в методології інтегрування (складання) різних програмних компонентів (програм, модулів, систем) і в методології побудови складних комп'ютерних систем на основі принципу «повторного використання коду». За допомогою API програмні компоненти взаємодіють один з одним. При цьому звичайно компоненти утворюють ієрархію – високорівневі компоненти використовують API низькорівневих компонентів, а ті, у свою чергу, використовують API ще більш низькорівневих компонент [35].

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для створення вікон чи іконок на екрані.

При використанні прикладного програмного інтерфейсу в контексті веброзробки, як правило, API визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON) [34].

3.3 Архітектура клієнтського застосунку

При розробці вебзастосунку був використаний шаблон проектування MVC (рисунок 3.2).

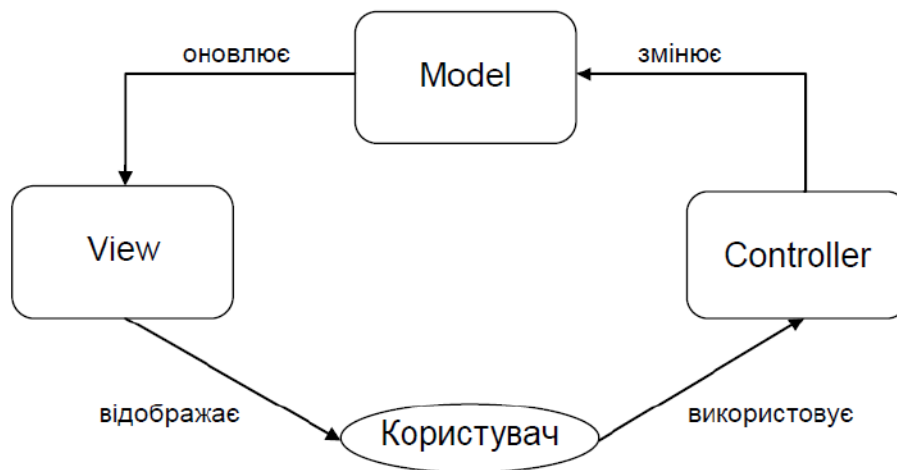


Рисунок 3.2 – Схема шаблону MVC

Шаблон розробки MVC дозволяє розділити застосунок на три функціональні частини:

Model є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.

View – вид або подання (призначений для користувача інтерфейсу). Може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації.

Controller одержує вхідні дані й перетворює їх на команди для Model чи View [36].

В MVC компоненти розділені таким чином, що модифікація одного з них надає мінімальний вплив на інші. Model надає дані і методи роботи з ними, реагує на запити, змінюючи свій стан. View відповідає за візуалізацію. Часто в якості уявлення виступає форма з графічними елементами. Controller забезпечує введення даних користувачем і використовує Model і View для реалізації необхідної реакції [37].

3.4 Інструменти розробки

Одна з основних вимог до програмного продукту – розробка мультиплатформного застосунку з використанням відкритих технологій. Тому кожен з рівнів обраної триланкової архітектури був реалізований з використанням різних технологій.

Технології для клієнтського рівня: мова програмування JavaScript, фреймворк для побудови вебзастосунків React.

Технології для серверного рівня та рівня бази даних було обрано Node.js та MySQL.

3.4.1 Мова програмування JavaScript та фреймворк React

JavaScript – об'єктна та прототипно-побудована мова для розробки програмного забезпечення, що в більшості випадків застосовується у веб-розробці [38].

Основні галузі використання JavaScript:

- створення динамічних сторінок, тобто сторінок, вміст яких може змінюватися після завантаження;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- рішення "локальних" завдань за допомогою сценаріїв, наприклад, перевірка коректності заповнення призначених для користувача форм;
- програмування на стороні сервера (Node.js);
- як основа більшості AJAX-додатків;
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- всередині PDF-документів тощо [39].

JavaScript дозволяє створювати додатки, які виконуються на стороні клієнта, тобто ці програми (сценарії) виконуються браузером на комп'ютері користувача та обробляються вбудованим в цей браузер інтерпретатором [39].

Для JavaScript характерні наступні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

Синтаксис JavaScript схожий з мовою Сі, але має корінні відмінності:

- об'єкти з можливістю інтроспекції (можливість запросити тип і структуру об'єкту під час виконання програми);
- функції як об'єкти першого класу (елементи, які можуть бути передані як параметр, повернуті з функції, присвоєні змінної);
- автоматичне приведення типів (перетворення значення одного типу в значення іншого типу);
- автоматичне прибирання сміття (періодично звільняє пам'ять, видаляючи об'єкти, які вже не будуть використані застосунками);
- анонімні функції (такі, що оголошуються в місці використання і не отримують унікального ідентифікатора для доступу до них) [40].

JavaScript є інтерпретатором без строгої типізації та може мати відмінності при виконанні у різних середовищах, тому необхідно приділяти чимало уваги тестуванню.

Кожен блок сценарію інтерпретатор розбирає окремо. Рекомендовано тримати функції сценарію в окремому блоці коду, або використовувати пов'язані .js файли, що значно спрощує пошук помилок на вебсторінках.

Для JavaScript існують різноманітні фреймворки, кожен з яких має свої інструменти та засоби і використовується в певних умовах. Фреймворки здатні значно спростити процес кодування та підвищити продуктивність за рахунок використання готових функцій та шаблонів. Один з таких фреймворків JavaScript є React.

React – це JavaScript-бібліотека з відкритим вихідним кодом для розробки інтерфейсів користувача, що став доступним у 2013 році і на даний час є лідером у сфері інфраструктури JavaScript UI [41].

React заснований на компоненті багаторазового використання. Простіше кажучи, це блоки коду, які можна класифікувати як класи або функції. Кожен компонент представляє певну частину сторінки, таку як логотип, кнопка або поле введення. Використовувані ними параметри називаються реквізитами, що означає властивості. Говорячи про синтаксис, більшість розробників сходяться на думці, що React легко освоїти, коли ви вже знаєте JavaScript.

React використовує JSX, синтаксис XML, який поєднує в собі JavaScript і HTML [42].

Вибору мови програмування JavaScript посприяло те, що вона є прототипною скриптовою мовою програмування з динамічною типізацією. Фреймворк React володіє багатою бібліотекою готових рішень, також на вибір цього фреймворку вплинула вбудована підтримка та масштабування до розмірів мобільних гаджетів.

3.4.2 Серверна платформа Node.js

Node.js – програмна платформа, заснована на основі інтерпретатору V8 (здійснює трансляцію JavaScript в машинний код), що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення.

Node.js додає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API (написаний на C ++), підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду [43].

Node.js застосовується переважно на сервері, виконуючи роль веб-сервера. Node.js – це асинхронна, однопоточна, неблокуюча модель введення

/ виведення. Пакетна екосистема Node.js, npm, також є найбільшою в світі бібліотечною екосистемою з відкритим вихідним кодом [42].

Серверна платформа Node.js була обрана через її легкість та ефективність моделі введення / виведення, а також за доступність завдяки відкритому вихідному коду.

3.4.3 Система керування базами даних MySQL

Система MySQL — вільна система керування реляційними базами даних, а також компактний багатопотоковий сервер баз даних, який характеризується високою швидкістю, стійкістю і простотою використання [44].

У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається вииграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, тому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць [45].

Мова SQL (мова структурованих запитів) як частина системи MySQL – це стандартна мова програмування для генерації, маніпулювання та отримання інформації з реляційної бази даних [46].

Для некомерційного використання MySQL є безкоштовним. Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Недоліки сервера MySQL:

- не реалізована підтримка транзакцій. Натомість пропонується використовувати LOCK/UNLOCK TABLE;

- відсутня підтримка зовнішніх (foreign) ключів;
- відсутня підтримка тригерів і збережених процедур;
- відсутня підтримка представлень (VIEW).

Зазначені недоліки не є критичними при розробці малих і середніх застосунків [47].

Структура MySQL трирівнева: бази даних – таблиці – записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно таблиця являє собою сукупність записів. А запис – це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці – в межах бази даних, поля – в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем [48].

Для реалізації бази даних було обрано MySQL за велику надійність, швидкість і вбудовані можливості [49].

Зазначені вище технології дозволяють створювати якісні, надійні вебзастосунки та були обрані за зручність використання, актуальність і володіння відкритими вихідними кодами для вільного використання.

4 РОЗРОБКА ПРОГРАМИ

4.1 Структурна схема програмної системи

Система організації та проведення оцінювання ризиків медичних виробів та їх процесів буде складатися з кількох модулів, які в свою чергу будуть складатись зі своїх функціональних підблоків.

Кабінет користувача буде головним модулем, який одночасно буде елементом компонування системи. Усі модулі будуть розташовані в порядку, який притаманний схожим вебзастосункам, що використовують для вирішення задач такого роду. Тому така структура буде сприяти легкому та інтуїтивному користуванню системою.

На рисунку 4.1 представлена структурна схема системи, на якій розташовані основні програмні модулі.

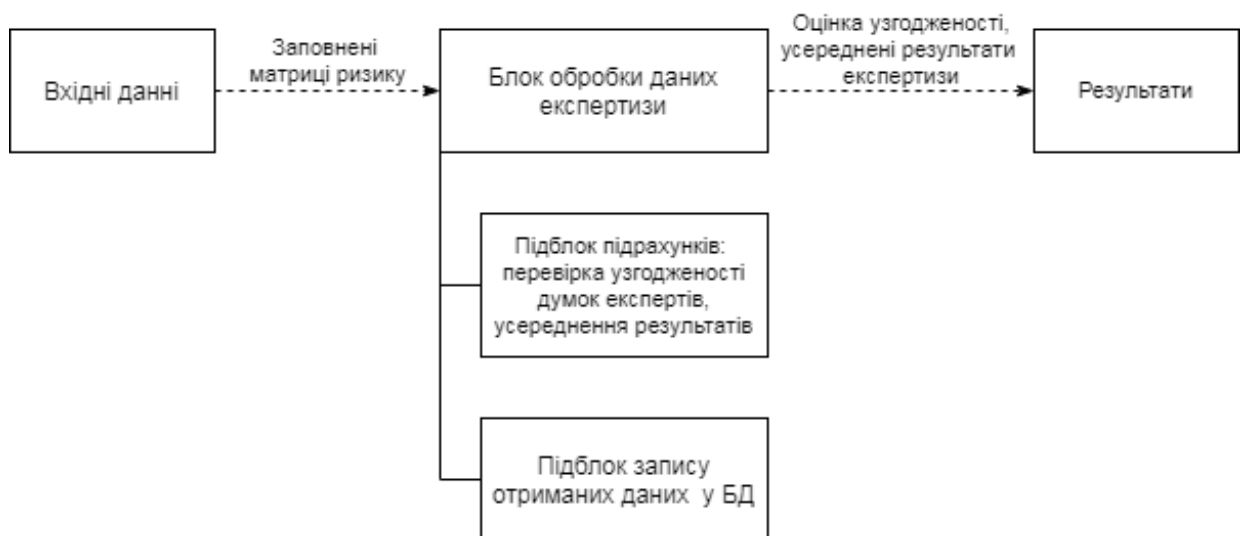


Рисунок 4.1 – Структурна схема системи

У систему заносяться вхідні дані – результати роботи експертної групи (заповнені матриці ризиків для об'єкту оцінки), які обробляються (перевіряються на узгодженість думок експертів та в позитивному випадку розраховуються усередненні значення оцінки ризиків) і виводиться результат

експертизи – показник узгодженості думок експертів та усередненні значення оцінки ризиків (при їх розрахунку).

4.2 Опис функціоналу системи

Програмний застосунок для організації та проведення оцінювання ризиків медичних систем та їх процесів містить трьох акторів – користувачів.

На рисунку 4.2 представлена діаграма прецедентів, яка описує функції та дії користувачів у системі.

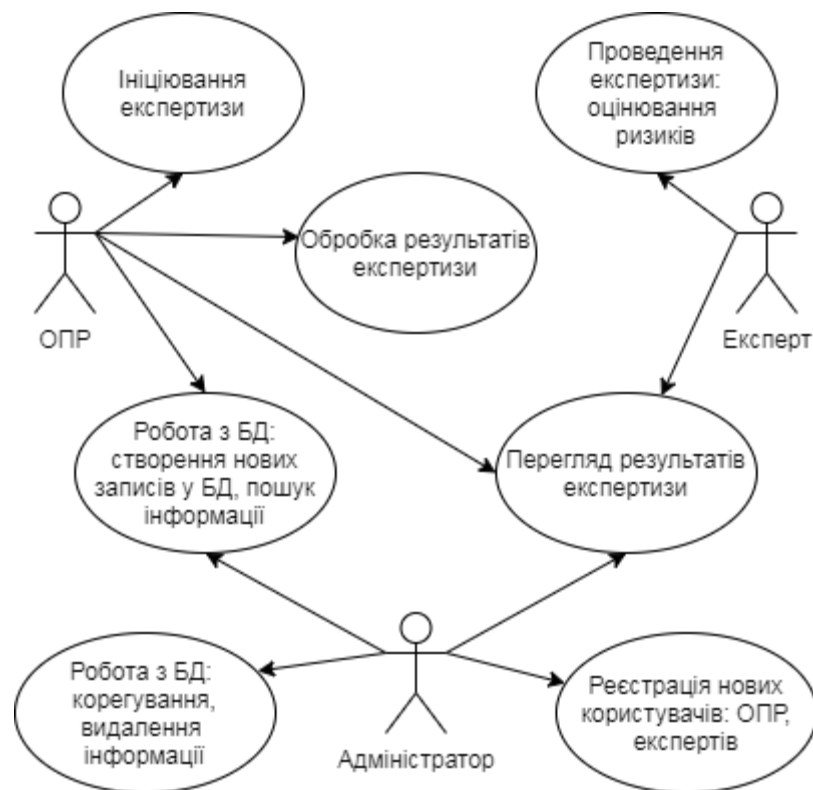


Рисунок 4.2 – Діаграма прецедентів системи

Згідно діаграми присутні три типи користувачів: ОПР, експерти та адміністратор.

До прав ОПР входять наступні функцій:

- отримання розширених прав доступу на сайті з можливістю внесення нових даних до БД;
- додавання інформації щодо нових експертів;

- створення нового об'єкту оцінювання ризиків з заповненням всієї необхідної інформації про нього;
- ідентифікація для об'єкту оцінювання можливих небезпек та ризиків, що вони викликають, із запропонованого списку типових варіантів;
- додавання за необхідності нових небезпек та ризиків до списку типових варіантів;
- ініціалізація проведення експертизи: вибір конкретного об'єкту оцінювання ризиків, встановлення строків проведення експертизи, формування групи експертів;
- по завершенню оцінювання ризиків групою експертів, проведення перевірки узгодженості думок експертів;
- усереднення оцінки експертів, якщо їх думки виявилися узгодженими, або ініціалізація нової експертизи з уточненням завдання або представників групи експертів;
- формування звіту щодо проведеної експертизи з оцінювання ризиків медичної системи або її процесу.

До прав експерта входять такі функції:

- реєстрація на сайті;
- вибір об'єкту оцінювання із запропонованих активних експертиз;
- заповнення матриці ризиків за представленим списком ризиків.

До прав адміністратора входять наступні функції:

- регулювання прав доступу;
- адміністрування бази даних: додавання, редагування та видалення інформації з бази даних, виправлення помилок тощо.

4.3 Опис бази даних

Вся інформація, що отримується та обробляється в застосунку зберігається в базі даних. Концептуальна модель бази даних наведена на рисунку 4.3.

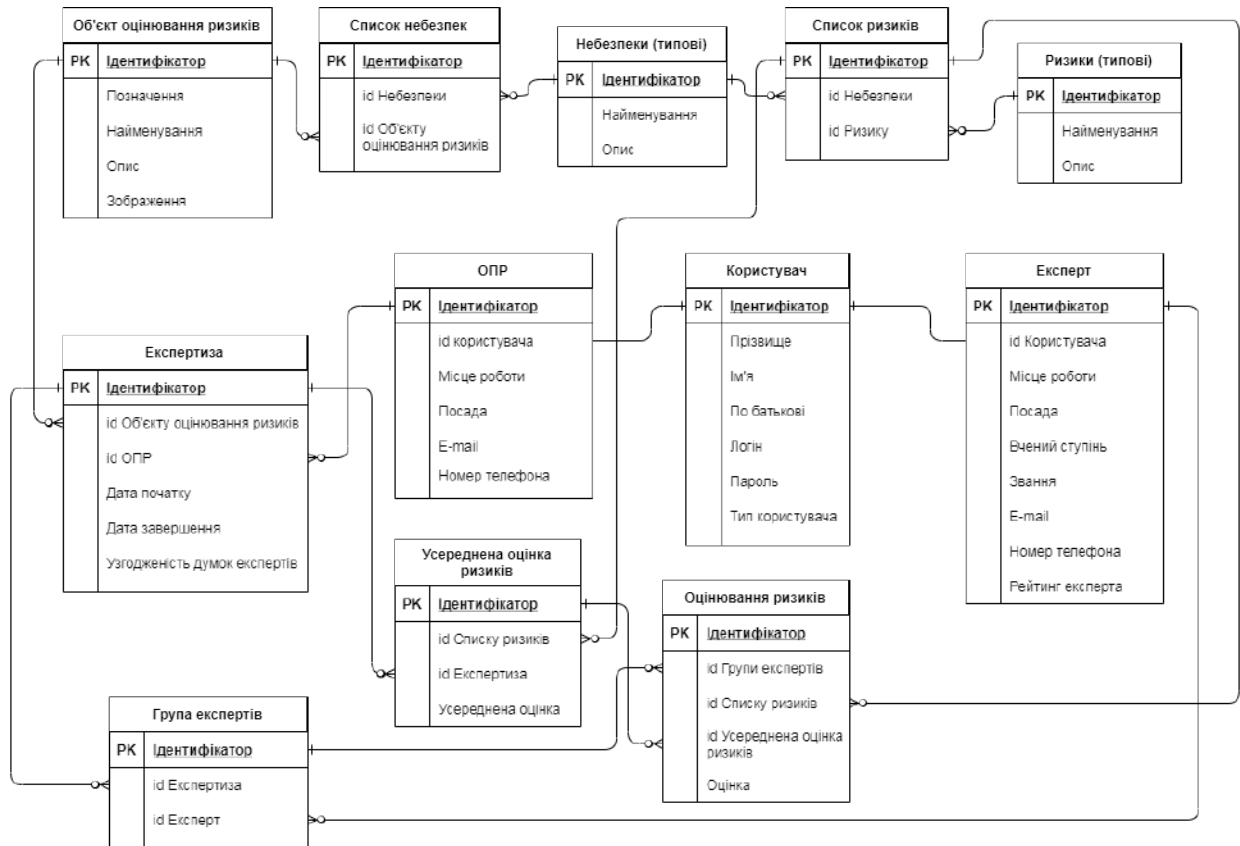


Рисунок 4.3 – Концептуальна модель бази даних

База даних складається з дев'яти основних таблиць: «Користувач», «ОПР», «Експерт», «Експертиза», «Об'єкт оцінювання ризиків», «Небезпеки (типові)», «Ризики (типові)», «Оцінювання ризиків» «Усереднена оцінка ризиків»; та трьох допоміжних таблиць-комутаторів «Група експертів», «Список небезпек», «Список ризиків».

Структура таблиці «Користувач» представлена в таблиці 4.1.

Таблиця 4.1 – Структура таблиці «Користувач»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
Surname	varchar	—	Прізвище
Name	varchar	—	Ім'я
Patronymic	varchar	—	По батькові
UserName	varchar	—	Логін
Password	varchar	—	Пароль
Status	int	—	Тип користувача

Таблиця «ОПР» містить зв'язок з таблицею «Користувач» по ідентифікатору користувача та детальні данні особи, що приймає рішення. Структура таблиці «ОПР» представлена в таблиці 4.2.

Таблиця 4.2 – Структура таблиці «ОПР»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdUser	int	—	Id Користувача
ChiefJob	varchar	—	Місце роботи
ChiePost	varchar	—	Посада
Email	varchar	—	E-mail
Phone	int	—	Номер телефона

Таблиця «Експерт» також містить зв'язок з таблицею «Користувач» по ідентифікатору користувача та детальні данні експертів. Структура таблиці «Експерт» представлена в таблиці 4.3.

Таблиця 4.3 – Структура таблиці «Експерт»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdUser	int	–	Id Користувача
ExpertJob	varchar	–	Місце роботи
ExpertPost	varchar	–	Посада
AcademicDegree	int	–	Вчений ступінь
AcademicTitle	int	–	Звання
Mail	varchar	–	E-mail
Phone	int	–	Номер телефона
ExpertRating	int	–	Рейтинг експерта

Таблиця «Експертиза» (таблиця 4.4) містить усі необхідні данні про експертизу, дати початку та кінця проведення, коефіцієнт узгодженості думок експертів. Зазначена таблиця має зв'язок з таблицями «Об'єкт оцінювання ризиків», «ОПР», «Усереднена оцінка ризиків» та «Група експертів».

Таблиця 4.4 – Структура таблиці «Експертиза»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdObjekt	int	–	id Об'єкту оцінювання ризику
IdChief	int	–	id ОПР
BeginDate	date	–	Дата початку експертизи
EndDate	date	–	Дата завершення експертизи
OpinionsConsistenc y	float	–	Коефіцієнт узгодженості думок експертів

В таблиці «Об'єкт оцінювання ризиків» зберігаються основні дані щодо об'єкту експертизи: шифр, найменування, опис характеристик та

властивостей, є посилання на зображення. Згадана таблиця має з'єднання через свій ідентифікатор з таблицями «Експертиза» та «Список небезпек» (таблиця 4.5).

Таблиця 4.5 – Структура таблиці «Об'єкт оцінювання ризиків»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
CodeObjekt	varchar	–	Позначення об'єкту (шифр)
NameObjekt	varchar	–	Найменування об'єкту
DescriptionObjekt	varchar	–	Опис характеристик, властивостей об'єкту
riskObjekts	varchar	–	Посилання на зображення об'єкту

У таблицях «Небезпеки (типові)» та «Ризики (типові)», структура яких наведена у таблицях 4.6 та 4.7 відповідно, зберігається перелік та опис типових для медичних виробів небезпек та ризиків, які в подальшому будуть обиратись ОПР для конкретного об'єкту оцінювання ризиків.

Ці таблиці мають відносини одна до одної «багато до багатьох» тому для їх з'єднання була застосована таблиця-комутатор «Список ризиків». Таблиця «Небезпеки (типові)» також через таблицю-комутатор «Список небезпек» з'єднана з таблицею «Об'єкт оцінювання ризиків».

Таблиця 4.6 – Структура таблиці «Небезпеки (типові)»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
NameDanger	varchar	–	Найменування небезпеки
DescriptionDanger	varchar	–	Опис небезпеки

Таблиця 4.7 – Структура таблиці «Ризики (типові)»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
NameRisk	varchar	–	Найменування ризику
DescriptionRisk	varchar	–	Опис ризику

Таблиця «Оцінювання ризиків» призначена для зберігання оцінок кожного експерта по кожному ризику для кожного об'єкту оцінювання ризиків (таблиця 4.8). З'єднана через ідентифікатори з таблицями «Усереднена оцінка ризиків», «Групи експертів» та «Список ризиків».

Таблиця 4.8 – Структура таблиці «Оцінювання ризиків»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdExpertGroup	int	–	id Групи експертів
IdRiskList	int	–	id Списку ризиків
IdAverageEstimate	int	–	id Усереднена оцінка ризиків
Estimate	int	–	Оцінка

У таблиці «Усереднена оцінка ризиків» зберігаються усереднені значення суджень експертів по кожному з ризиків для конкретного об'єкту оцінювання (таблиця 4.9). З'єднана с таблицями «Оцінювання ризиків», «Список ризиків» та «Експертиза».

Таблиця 4.9 – Структура таблиці «Усереднена оцінка ризиків»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdRiskList	int	–	id Списку ризиків
IdExpertise	int	–	id Експертиза
AverageEstimate	int	–	Значення усередненої оцінки ризиків

Допоміжні таблиці були створені як комутатори для виключення зв'язків «багато до багатьох». Їх структура наведена нижче (таблиці 4.10 – 4.12).

Таблиця 4.10 – Структура таблиці «Список небезпек»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdDanger	int	–	id Небезпеки
IdObjekt	int	–	id Об'єкту оцінювання ризиків

Таблиця 4.11 – Структура таблиці «Список ризиків»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdDanger	int	–	id Небезпеки
IdRisk	int	–	id Ризику

Таблиця 4.12 – Структура таблиці «Група експертів»

Назва поля	Тип поля	Ключ	Опис
Id	int	+	Первинний ключ
IdExpertise	int	–	id Експертиза
IdExpert	int	–	id Експерта

4.4 Розробка кабінету користувача

Оскільки перелік користувачів системи, що розробляється, є більш-менш постійним, то компонування системи на основі кабінету користувача є виправданим та функціональним рішенням. Це основне робоче місце користувача в системі, де можливий доступ до необхідної інформації згідно закріплених прав.

До кабінету користувача додані програмні підмодулі, що виконують основні функції системи. До них належать наступні:

- модуль керування інформацією в базі даних;
- модуль ініціювання експертизи;
- модуль проведення експертизи;
- модуль обробки результаті проведеної експертизи;
- модуль керування даними користувачів.

4.4.1 Модуль керування інформацією в базі даних

Даний модуль дозволяє користувачам застосунку згідно їх прав доступу керувати інформацією в БД, а саме:

- вносити нову інформацію до БД стосовно об'єкту оцінювання ризиків, типових для медичних виробів небезпек та ризиків, а також переглядати всю наявну інформацію в БД та створювати пошукові запити (за правами доступу ОПР);
- переглядати інформацію з БД щодо об'єкту оцінювання ризиків, виділених для нього небезпек та ризиків, результатів експертизи, а також створювати пошукові запити по доступній інформації (за правами доступу експерта);
- створювати звіти за результатами експертизи (за правами доступу ОПР);

– адміністрування бази даних: додавання, редагування та видалення інформації з бази даних, виправлення помилок тощо (за правами адміністратора).

4.4.2 Модуль ініціювання експертизи

Даний модуль дозволяє користувачу застосунку з правами доступу ОПР ініціювати проведення експертизи.

ОПР встановлює конкретний об'єкт оцінювання ризиків, вибирає для нього із списку типових варіантів небезпеки та ризики, встановлює строки проведення експертизи, формує групу експертів з наявних в БД.

4.4.3 Модуль проведення експертизи

Дозволяє експертам проводити оцінювання (присвоєння відповідного рангу) виділених ризиків для запропонованого об'єкту згідно наведених матриць ризиків.

4.4.4 Модуль обробки результаті проведеної експертизи

Дозволяє ОПР згідно отриманих результатів оцінювання експертів автоматично проводити перевірку узгодженості їх думок та при позитивному результаті проводити усереднення суджень за експертною групою.

4.4.5 Модуль керування даними користувачів

Даний модуль дозволяє користувачу додатку керувати даними, а саме:

- реєструвати у системі нових експертів та ОПР (за правами доступу адміністратора);
- змінювати основні дані експертів (за правами доступу ОПР);

- змінювати основні дані експертів та ОПР (за правами доступу адміністратора);
- видаляти облікові записи користувачів (за правами доступу адміністратора);
- змінювати логін і пароль до облікового запису.

4.5 Виконання програмного продукту

Програмний продукт не потребує інсталяції на комп'ютер користувача оскільки є вебзастосунком, що розроблений з використанням сучасних технологій. Тому для його роботи необхідні браузер, які підтримують актуальні мережеві стандарти, та наявність стабільного інтернет зв'язку зі швидкістю не менше 50 мбіт за секунду.

4.5.1 Головна сторінка

При вході до вебзастосунку відкривається головна сторінка (рисунки 4.4), яка призначена для ознайомлення з тематикою сайту та має посилання на інші сторінки, а також пропонує авторизуватись для входу у особистий кабінет або зареєструватись на сайті для його створення новому користувачу.

Меню сайту містить наступні пункти: «Головна», «Робота з БД», «Проведення експертизи», «Про проєкт», «Контакти». Пункти меню «Робота з БД» та «Проведення експертизи» стають активними тільки при авторизації у вебзастосунку.

4.5.2 Персональний кабінет

При вході у персональний кабінет вебзастосунка клієнту пропонується авторизуватися або пройти реєстрацію. Форми реєстрації та авторизації наведені на рисунках 4.5 та 4.6 відповідно.

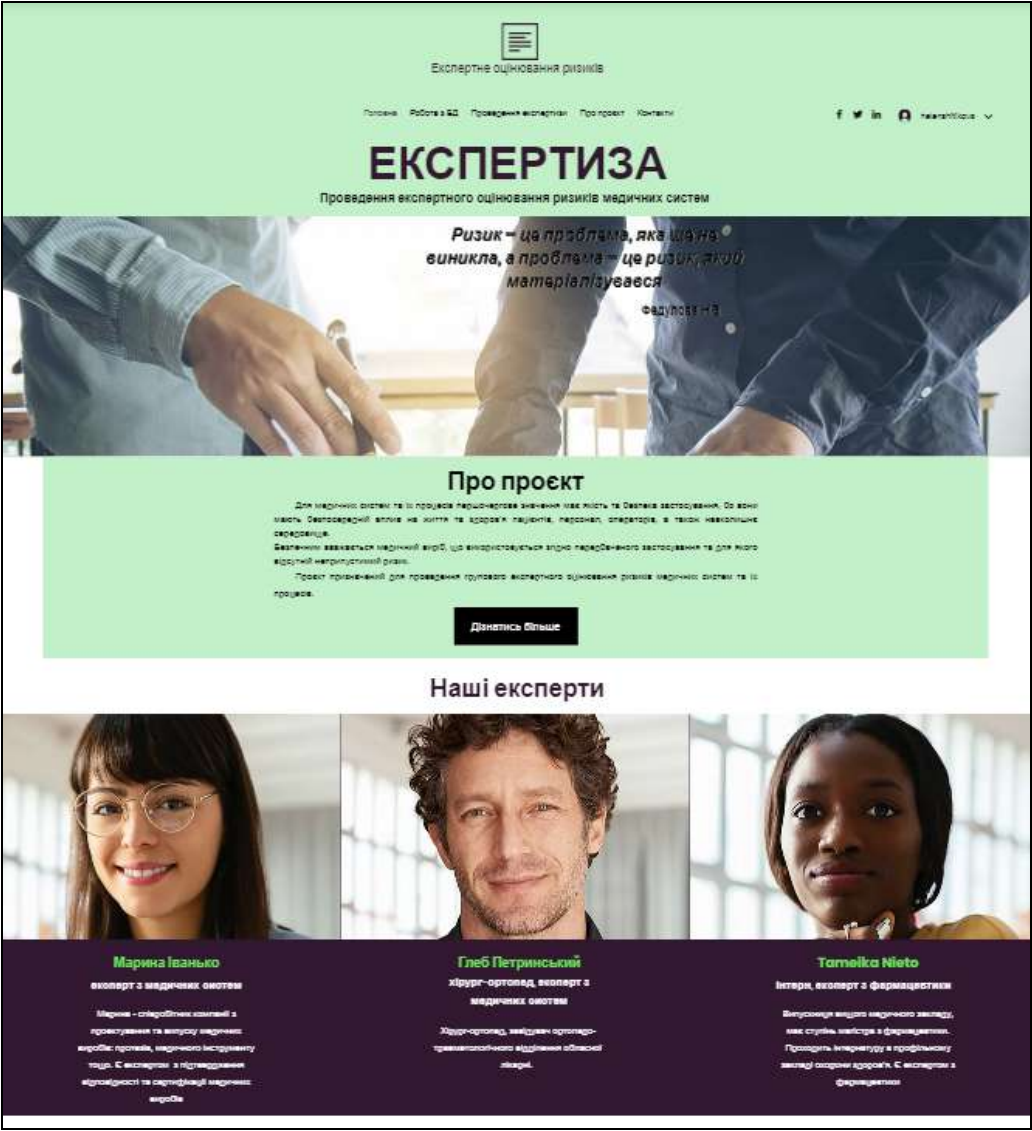


Рисунок 4.4 – «Головна» сторінка застосунку

Реєстрація

Прізвище _____

Ім'я _____

По батькові _____

Ел. пошта _____

Пароль _____

Зареєструватись

Вже є персональний кабінет? **увійти**

Рисунок 4.5 – Форма реєстрації у вебзастосунку

Увійти

Новенький? Зареєструватися

Увійти через Facebook

Увійти через Google+

або

Увійти через ел. пошту

Рисунок 4.6 – Форма авторизації у вебзастосунку

Сторінка персонального кабінету зображена на рисунку 4.7. Вона містить фото користувача та форму, де зазначені його прізвище, ім'я, по батькові, електронна пошта, контактний телефон, місце роботи та посада. При потребі будь які з зазначених даних можуть бути відкориговані.

Експертне оцінювання ризиків

Головна Робота з БД Експертиза Про проєкт Контакти

Мої дані

Перевірте інформацію щодо вас, при потребі відкоригуйте її

Прізвище * Ім'я * По батькові *

Іванько Марина Петрівна

Ел. пошта * Телефон

ivankomarina@mail.com +380951212121

Місце роботи Посада

АТ "УкрПротекс" спеціаліст

Далі

Рисунок 4.7 – Персональний кабінет користувача

4.5.3 Сторінки «Про проєкт» та «Контакти»

Без реєстрації на сайті окрім сторінки «Головна» доступні пункти меню «Про проєкт» та «Контакти».

При виклику пункту меню «Про проєкт» відкривається однойменна сторінка, яка зображена на рисунку 4.8.

Вона містить описову інформацію щодо проєкту, його завдання та обґрунтування актуальності, також є посилання на довідкову інформацію щодо роз'яснення поняття ризику та процесу управління ризиком і правила та методи роботи групового експертного опитування.

Сторінку «Контакти» також можна викликати з головного меню. Вона зображена на рисунку 4.9. На ній містяться контактна інформація та форма зворотного зв'язку «Пишіть нам».



Рисунок 4.8 – Сторінка застосунку «Про проєкт»

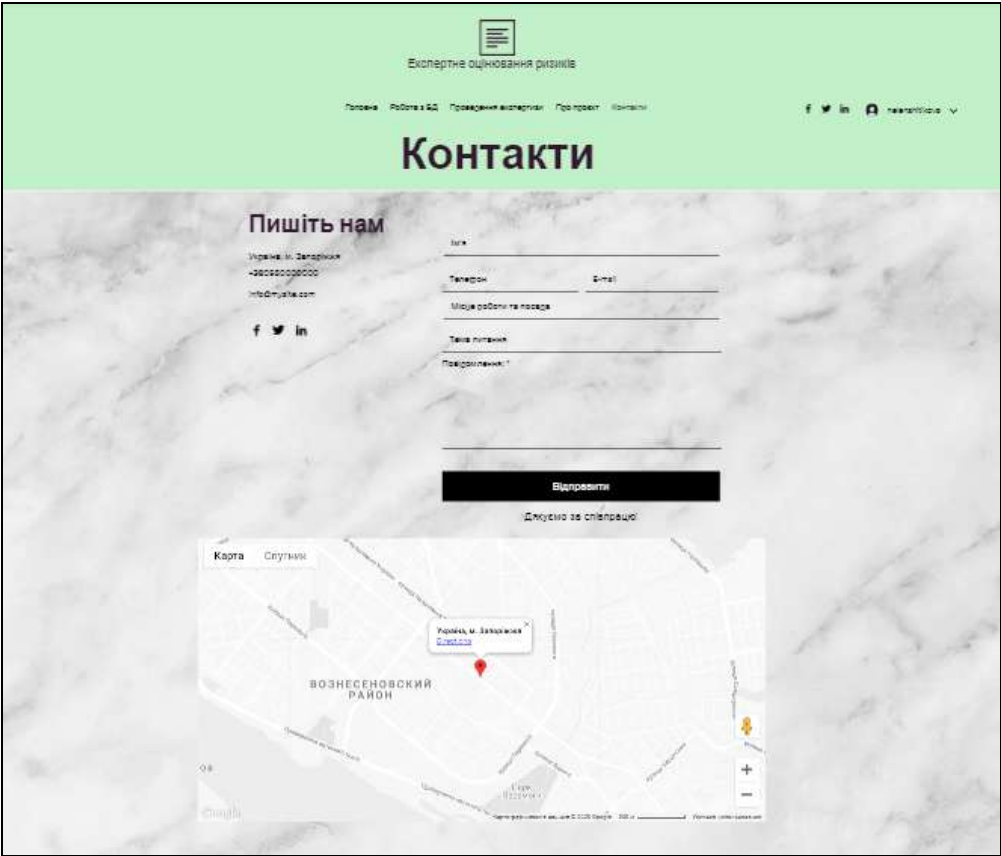


Рисунок 4.9 – Сторінка застосунку «Контакти»

4.5.4 Сторінка «Робота з БД»

Для зареєстрованих на сайті користувачів доступні основні функціональні сторінки. Одна з них – це сторінка «Робота з БД», на якій міститься форма запиту до БД (рисунок 4.10).

Пошуковий запит:

Таблиця: Показати поля:

Виберіть

☐ Додати ще таблиць

Таблиця: Показати поля:

Виберіть

☐ Додати ще таблиць

Таблиця: Показати поля:

Виберіть

☐ Показати зв'язані дані

Відправити

Рисунок 4.10 – Порожня форма запиту

4.5.5 Сторінка «Проведення експертизи»

При заповненні та відправленні запити на сторінці відкривається відповідна інформація (рисунок 4.11).

Об'єкт оцінювання ризиків				Експертиза			
Найменування	Шифр	Список	Зображення	ОПР	Дата початку	Дата закінчення	Коефіцієнт узгодженості
Експертиза комерційного суднобуду	БПК 2874	Інвентаризація комерційного суднобуду	./images/БПК_2874.gif	Петришин	15.11.2020	30.11.2020	0.89
				Петришин	01.12.2020		
Експертиза транспортних суднобуду	ЕТОС	Інвентаризація транспортних суднобуду	./images/ЕТОС.gif	Петришин	20.11.2020	25.11.2020	0.93
				Іванко	25.12.2020		
Набір інструментів	БНЕТС-2	Інвентаризація набору інструментів	./images/БНЕТС_2.gif	Іванко	30.11.2020		

Рисунок 4.11 – Сторінка застосунку «Робота з БД» з заповненою формою запиту

Сторінка «Проведення експертизи» доступна при виклику однойменного пункту меню для користувачів з правами доступу експерта. Вона зображена на рисунку 4.12. На ній відображаються картки для об'єктів оцінювання ризиків для активних на даний час експертиз та кнопка «Нова експертиза». На кожній картці об'єкту окрім посилання на сторінку з безпосередньою процедурою оцінювання ризиків є кнопка «Завершити експертизу».

Кнопки «Нова експертиза» та «Завершити експертизу» активні тільки для користувачів з правами ОПР.

При натисканні на кнопку «Завершити експертизу» програма перевіряє кількість експертів, які вже прийняли участь в оцінюванні. При кількості експертів менше 50% від повної групи виводиться повідомлення про

недостатню кількість даних для завершення експертизи. Вікно повідомлення зображене на рисунку 4.13



Рисунок 4.12 – Сторінка застосунку «Проведення експертизи»



Рисунок 4.13 – Повідомлення про недостатню кількість даних при спробі завершити експертизу

При достатній кількості даних процес оцінювання ризиків для користувачів з правами експертів перестає бути доступним, автоматично виставляється дата завершення експертизи, розраховується коефіцієнт конкордації Кендалла для перевірки узгодженості думок експертів та при середній і високій його значущості розраховуються усереднені значення оцінюваних ризиків. Значення коефіцієнта конкордації Кендалла від 0,4 до 0,7 може свідчити про середній рівень узгодженості, а більший за 0,7 вважається високим рівнем узгодженості.

При середньому та високому рівні узгодженості виводиться повідомлення про успішне завершення експертизи та пропонується переглянути результати (рисунок 4.14), в противному випадку виводиться повідомлення про низьку узгодженість думок експертів та робиться пропозиція провести повторну експертизу з уточненням завдання і/або представників групи експертів (рисунок 4.15).

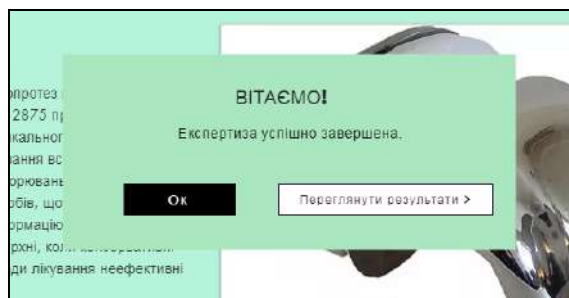


Рисунок 4.14 – Повідомлення про успішне завершення експертизи

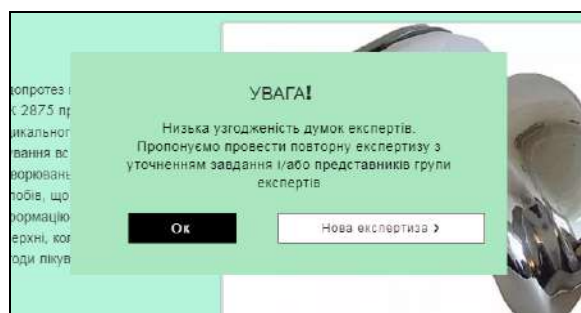


Рисунок 4.15 – Повідомлення про низьку узгодженість думок експертів

5 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У дипломній кваліфікаційній роботі магістра досліджено та виконано програмну реалізацію методу оцінювання ризиків в біоінженерії, зокрема для медичних систем та їх процесів.

Очікується, що впровадження в експлуатацію розробленого програмного забезпечення на підприємстві з виробництва медичних систем дозволить скоротити 0,75 ставки одного інженера відділу конструкторське-технічної документації, що займається проведенням експертизи з оцінювання ризиків для процедур підтвердження відповідності та сертифікації медичних виробів, заробітна платня якого складає 13300 гривень на місяць.

У розробці беруть участь програміст із заробітною платою 15300 грн. в місяць і консультант з заробітною платою 9400 грн. у місяць. На розробку проєкту було витрачено 2 місяці. Розробка програми розпочалася першого вересня і повинна бути виконана до кінця жовтня 2020 року.

5.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи [24]. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. Інформація з характеристики робіт по розробці програмного забезпечення зведена у таблицю 5.1.

Таблиця 5.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд-дн	%		
Аналіз ПЗ	5	7,4	Програміст Консультант	4 1
Визначення вимог до ПП	9	13,4	Програміст Консультант	4 5
Проектування структури програми	30	44,8	Програміст Консультант	16 14
Розробка схеми функціонування програми	3	4,5	Програміст	5
Створення програмного коду	11	16,4	Програміст	11
Тестування і налагодження програмного продукту	5	7,5	Програміст Консультант	3 2
Складання програмної документації	2	3,0	Програміст	2
Разом	67	100	Програміст Консультант	45 22

5.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складають калькуляцію кошторисної вартості робіт, що включає статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання (на оплату машинного часу ЕОМ для написання і налагодження програмних засобів);
- накладні витрати.

Витрати за статтею «Основна заробітна плата» складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми. Розрахунок зарплати на підставі даних про трудомісткість, представлений в таблиці 5.2.

Таблиця 5.2 – Основна заробітна плата

Посада виконавця	Чисельність, чол	місячний оклад	Кількість місяців роботи	Сума ЗП, грн
Програміст	1	15300	2	30600
Консультант	1	9400	1	9400
Разом	2			40000

Додаткову заробітну плату приймають рівною 10% від основної зарплати. Вона розраховується за формулою (5.1):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 10\%. \quad (5.1)$$

Підставивши величину основної заробітної плати в дану формулу отримаємо: $ЗП_{\text{дод}} = 40000 \cdot 0.1 = 4000$ грн.

Відрахування на єдиний соціальний внесок (ЄСВ) визначають у відсотковому відношенні від сум основної та додаткової зарплати з урахуванням премій і доплат. Відрахування вираховуються за формулою (5.2):

$$ЄСВ = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22. \quad (5.2)$$

$$ЄСВ = (40000 + 4000) \cdot 0,22 = 9680 \text{ грн.}$$

Використовується три найменування матеріалів: папір – 300 грн., картридж – 500 грн, флеш-носій – 200 грн.. Витрати на матеріали розраховують за формулою (5.3):

$$M = \sum_{i=1}^n (\text{Ц}_i \cdot N_i (1 + K_{\text{т.з.}}) - \text{Ц}_{io} \cdot \text{Ц}_{io}), \quad (5.3)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн .;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Ц_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн .;

N_i – потреба в i -му матеріалі, напівфабрикаті і комплектуючому;

Ц_{io} – ціна зворотних відходів i -го найменування матеріалу, грн;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

$$\text{Ц}_{io} = 0; N_{io} = 0; K_{\text{т.з.}} = 0,05;$$

$$M = (1 + 0,05) \cdot (300 + 500 + 200) = 1050 \text{ грн.}$$

Таким чином, разом витрати на матеріали та комплектуючі вироби становлять 1050 гривень.

У статтю «Витрати на спеціальне обладнання» входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів. Для цього необхідно скласти кошторис «витрат на утримання і експлуатацію устаткування», виходячи з якого визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на написання і налагодження програми отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (5.4):

$$A = \frac{\Phi_6 \cdot H_a}{100}, \quad (5.4)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн;

N_a – норма амортизаційних відрахувань на повне оновлення обчислювальної техніки,%. Приймаємо, що техніка повинна працювати 4 роки, тому $N_a=25\%$.

Балансова вартість обчислювальної техніки становить 35500 грн. Тому амортизаційні відрахування становлять:

$$A = 35500 \cdot 0,25 = 8875 \text{ грн}$$

Статтю "Експлуатація обладнання" розраховують підсумовуванням витрат на силову електроенергію і допоміжні матеріали. Витрати на електроенергію визначають за формулою (5.5):

$$C_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_{\text{ел.е}}, \quad (5.5)$$

де N_n – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи ЕОМ, ч.;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

$C_{\text{ел.е}}$ – ціна одного кВт·ч електроенергії, грн.

Номінальна Потужність ЕОМ – 0,2 кВт. Річний ефективний фонд часу роботи ЕОМ ставити 1800 годин. Середні коефіцієнти завантаження за часом та за потужністю рівні відповідно 0,9 та 0,6. Ціна однієї кіловат–години електроенергії ставить 2,11 грн. Отримуємо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Приймаємо, що $C_e = 410$ грн.

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (5.6):

$$ЗП_{\text{обсл}} = \Phi ЗП_{\Gamma} \cdot (1 + K_{\text{відрах}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (5.6)$$

де $\Phi ЗП_{\Gamma}$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{\text{відрах}}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідний на технічне обслуговування ЕОМ, год./рік;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу становить 1800 годин

Заробітна плата обслуговуючого персоналу = 9800 грн/місяць.

$$\Phi ЗП_{\Gamma} = 9800 \cdot 12 = 117600 \text{ грн.}$$

$$ЗП_{\text{обсл}} = 117600 \cdot (1 + 0,22) \cdot 36 / 1800 = 2869 \text{ грн.}$$

Суму витрат за статтею "Поточний ремонт обладнання" обчислимо як 3% від балансової вартості обладнання, оскільки неможливо заздалегідь визначити, скільки коштів необхідно витратити на придбання запасних частин наявного обладнання. Балансова вартість обчислювальної техніки становить 35500 грн. Тому сума витрат за статтею "Поточний ремонт обладнання" становить: $35500 \cdot 0,03 = 1065$ грн.

Інших витрат за проектом не передбачено, тому сума витрат за цією статтею дорівнює нуль гривень.

Кошторис витрат на утримання і експлуатацію устаткування наведений в таблиці 5.3.

Таблиця 5.3 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	8875
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	2869
Експлуатація обладнання (крім витрат на поточний ремонт)	410
Поточний ремонт обладнання	1065
Інші витрати	0
Разом	13219

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (5.7):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (5.7)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

$$P_{\text{екс}} = 13219/1800 = 7,34 \text{ грн. / машино–год.}$$

Для написання і налагодження даного програмного продукту ЕОМ експлуатується протягом 45 днів в одну зміну по 8 годин, що складає в сумі 360 годин. Таким чином отримуємо суму за статтею «Витрати на спеціальне обладнання»:

$$C_{\text{мо}} = 7,34 \cdot 360 = 2642 \text{ грн.}$$

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. Для обчислення цієї статті видатків у дипломному проєкті приймемо, що накладні видатки становлять 20% від основної заробітної плати розробників, що складає 40000 грн. Тому сума накладних видатків становить: $40000 \cdot 0,25 = 10000$ грн.

Калькуляція кошторисної вартості робіт з розробки програми наведена в таблиці 5.4.

Таблиця 5.4 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.
Основна заробітна плата	40000
Додаткова заробітна плата	4000
ЄСВ	9680
Матеріали і комплектуючі	1050
Витрати на спеціальне обладнання	2642
Накладні витрати	10000
Разом	67372

5.3 Розрахунок економічної ефективності програмного продукту

Річну економію від впровадження програми, що дозволяє знизити витрати машинного часу ЕОМ для рішення певного завдання в порівнянні із програмою, що застосовувалася раніше, визначають за формулою:

$$E = (T_{m1} - T_{m2}) \cdot B_{\text{екс.КТЗ}} + E_n^a \cdot \frac{\Phi_{\text{б КТЗ}} \cdot (T_{m1} - T_{m2})}{\Phi_{\text{еф КТЗ}}} - \frac{S_{\text{рп}}}{T_c} + E_{\text{зп}}, \quad (5.8)$$

де T_{m1} , T_{m2} – машинний час, необхідний для розв’язання поставлених завдань, відповідно в старому та у новому варіантах, машино–год./рік;

$B_{\text{екс}}$ – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн/ машино–год.;

E_n^a – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б КТЗ}}$ – балансова вартість ЕОМ (КТЗ), грн.;

$\Phi_{\text{еф КТЗ}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год.;

$S_{\text{рп}}$ – сумарні витрати на розробку програми, грн.;

T_c – термін служби впроваджуваної програми до її морального зношування, років. Приймаємо цей термін 5 років;

$E_{\text{зп}}$ – економія на заробітній платі.

До впровадження даного ПЗ дані функції виконувались інженером відділу конструкторське-технічної документації 1800 годин на рік, а після 450 годин на рік (економія складає 0,75 ставки одного інженера, з заробітною платою 13300 гривень на місяць).

$$E = (1800 - 450) \cdot 7,34 + 0,2 \cdot (35500 \cdot (1800 - 450)) / 1800 - 67372/5 + \\ + 13300 \cdot 0,75(1 + 0,22) \cdot 12 = 147794 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E и строк їхньої окупності $T_{ок}$ розраховуються за формулами (5.9) і (5.10):

$$E_{\phi} = \frac{E}{S_{рп}}, \quad (5.9)$$

$$T_{ок} = \frac{S_{рп}}{E}, \quad (5.10)$$

Якщо розрахувати за формулами (5.9) та (5.10) отримаємо:

$$E_{\phi} = 147794 / 67372 = 2,19,$$

$$T_{ок} = 67372 / 147794 = 0,46 \text{ року.}$$

Техніко–економічні показники наведені в таблиці 5.5.

Таблиця 5.5 – Техніко–економічні показники

Показники	Сума	Одиниця виміру
Витрати на розробку програми	67372	грн.
Річна економія	147794	грн.
Строк окупності програми	0,46	рік

Таким чином, впровадження в експлуатацію розробленого програмного забезпечення на підприємстві з виробництва медичних систем дозволяє скоротити 0,75 ставки одного інженера відділу конструкторське-технічної документації, що займається проведенням експертизи з оцінювання ризиків для процедур підтвердження відповідності та сертифікації медичних виробів, заробітна платня якого складає 13300 гривень на місяць. Економічні розрахунки показали, що строк окупності програми 0,46 року, що значно менше часу її нормативного строку окупності (5 років). Крім того, річна економія при використанні програми становить 147794 грн. Виходячи з виконаних розрахунків, можна зробити висновок, що розробка даного програмного продукту є економічно доцільною [50].

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В дипломному проекті проводиться дослідження та програмна реалізація методів оцінювання ризиків в біоінженерії.

6.1 Аналіз потенційних небезпек.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

- негативний вплив випромінювань відео дисплейних терміналів на органи зору людини має основною причиною перенапруження м'язового апарату ока, що пов'язане з недостатньою контрастністю та мерехтінням зображення, відблесками та відбитим світлом від моніторів. Крім того, втома очей настає при частій зміні огляду більш освітленого монітора та менш освітленої клавіатури, що призводить до «астенопії» (відчуття дискомфорту та біль в очах);
- невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця;
- можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразка електрострумом, що може привести до електротравм або летальних наслідків;
- незадовільні умови освітлення робочого місця, в наслідок не рівномірного розподілу освітлювальних приладів;
- невідповідність нормам параметрів мікроклімату;
- небезпеки, пов'язані з підвищеним рівнем шуму, який створюється устаткуванням та людьми, що знаходяться у приміщенні, і які призводить до погіршення слуху;
- можливість загорянь внаслідок порушень правил пожежної безпеки, що може привести до пожеж;

– неправильні дії персоналу під час НС внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків [51].

6.2 Заходи із забезпечення безпеки.

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

Згідно з вимогами щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями передбачені гігієнічні вимоги до організації устаткування робочих місць ВДТ ЕОМ [52].

Конструкція робочого стола відповідає сучасним вимогам ергономіки й забезпечує оптимальне розміщення на робочій поверхні використовуваного устаткування (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого стола із ВДТ регулюється в межах 680...800 мм, а ширина й глибина - забезпечує можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800..1000 мм). Приймаємо робочу поверхню з розмірами 1400x800. Щоб уникнути відблисків на екрані вікон розмістили робоче місце торцем до вікна, не ближче чим на 1 м. Площа дозволяє нам розмістити додаткові меблі для зберігання документів, магнітних дисків, полками, стелажми, тумбами й т.п.

Під час організації робочого простору враховано індивідуальні антропометричні параметри користувача з відповідними допусками на можливі зміни робочих поз та потребу у переміщеннях.

Заходи мінімізації негативного впливу випромінювань від відео дисплейних терміналів на органи зору:

– передбачено такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці;

- для розробників програм із застосуванням ЕОМ призначено регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;
- для операторів із застосування ЕОМ слід призначено регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;
- для операторів комп'ютерного набору призначено регламентовані перерви для відпочинку тривалістю 10 хвилин після кожною години роботи за ВДТ.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не перевищує 4 години.

При 12-годинній робочій зміні регламентовані перерви встановлюються в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин [52].

З метою зменшення негативного впливу монотонності застосовуються чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи).

Для зниження нервово-емоційного напруження, втомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільні деякі перерви використовувати для виконання комплексу вправ.

В окремих випадках - при хронічних скаргах працюючих з ВДТ ЕОМ і ПЕОМ на зорове втомлення, незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей - допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ [52].

Активний відпочинок полягає у виконанні комплексу гімнастичних вправ, спрямованих на зняття нервового напруження, м'язове розслаблення, відновлення функцій фізіологічних систем, що порушуються протягом трудового процесу, зняття втоми очей, поліпшення мозкового кровообігу і працездатності [52].

Згідно ПУЕ приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [53].

Апаратура, що перебуває під небезпечною для життя напругою, припускає забезпечення електробезпечності. Всі металеві частини заземлені на третій провід спеціальних триконтактних розеток. Струмopовідні кабелі розташовані під дерев'яною підлогою, зверху покритою лінолеумом, або у вигляді схованої проводки в стінах, таким чином, контакт із кабелем виключається. Над кожною розеткою присутній попереджувачий надпис.

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. У процесі експлуатації періодично проводиться перевірка якості ізоляції, шляхом виміру її опору. Відповідно до ПУЕ в електроустановках до 1000В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм.

Відповідно до ПУЕ захисне заземлення застосовується у всіх електроустановках змінного струму під напругою 380В и вище й постійного струму 440 В и вище. Захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок відповідно до вимог до захисних заходів від ураження електричним струмом [53].

Для уникнення уражень електричним струмом, серед працівників та обслуговуючого персоналу, виконаємо розрахунок системи захисного заземлення, яке використовується в електромережах з напругою до 1000 В

[53]. Згідно з завданням, вихідні дані для розрахунку захисного заземлення приведені в таблиці 6.1.

Таблиця 6.1 - Вихідні дані для розрахунку

Назва параметру	Позначення, од.	Значення
Напруга в мережі	U , В	380
Розміри приміщення	$L \times B$, м	6x5
Опір природного заземлювача	R_e , Ом	8
Довжина вертикального стрижня	l_b , м	3
Діаметр вертикального стрижня	d , м	0,018
Ширина смуги	b_c , м	0,038
Питомий опір ґрунту	$\rho_{\text{вим}}$, Ом	84
Коефіцієнт сезонності ґрунту	Ψ	1,5
Розрахунковий опір заземлення, при $U < 1000$ В	R_3 , Ом	4

Використовуючи дані таблиці 6.1, виконаємо алгоритм розрахунку захисного заземлення, який передбачає виконання послідовності обчислень.

Визначаємо необхідний опір штучного заземлювача R_u , Ом, якщо передбачається використання також природного заземлювача R_e за формулою (6.1):

$$R_u = \frac{R_e \cdot R_3}{R_e - R_3}, \text{ Ом} \quad (6.1)$$

де R_e – опір розтіканню струму природних заземлювачів, Ом;

R_3 – розрахунковий нормований опір заземлювального пристрою (ЗП),

$R_e = 8$, $R_3 = 4$

Визначимо розрахунковий питомий опір ґрунту за формулою (6.2):

$$\rho = \rho_{\text{вим}} \cdot \Psi, \text{ Ом} \cdot \text{м} \quad (6.2)$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м;

$\rho_{\text{вим}}$ – питомий опір землі, отриманий у результаті вимірів, Ом·м; $\rho_{\text{вим}} = 84$ Ом·м;

Ψ – коефіцієнт сезонності, що враховує промерзання чи висихання ґрунту; $\Psi = 1,5$.

Отримуємо 126 Ом·м.

Обчислимо опір розтіканню струму одиночного вертикального заземлювача $R_{\text{в}}$, Ом. Для стрижневого заземлювача круглого перерізу, заглибленого в ґрунт, розрахункова формула має вигляд (6.3):

$$R_{\text{в}} = \frac{\rho}{2 \cdot \pi \cdot l_{\text{в}}} \left(\ln \frac{2 \cdot l_{\text{в}}}{d} + \frac{1}{2} \cdot \ln \frac{4 \cdot t_{\text{в}} + l_{\text{в}}}{5 \cdot t_{\text{в}} - l_{\text{в}}} \right), \text{ Ом} \quad (6.3)$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м; $\rho = 126$ Ом·м;

$l_{\text{в}}$ – довжина вертикального стрижня, м; $l_{\text{в}} = 3$ м;

d – діаметр перерізу стрижня, м; $d = 0,018$ м

$t_{\text{в}}$ – відстань від поверхні ґрунту до середини довжини вертикального стрижня.

Обчислюємо $R_{\text{в}} = 40,04$ Ом.

Розрахуємо наближену (мінімальну) кількість вертикальних стрижнів (6.4):

$$n' = \frac{R_{\text{в}}}{R_{\text{ш}}}, \text{ шт}, \quad (6.4)$$

де $R_{\text{ш}}$ – опір розтікання струму одиночного вертикального заземлювача, Ом; $R_{\text{ш}} = 40,04$ Ом;

$R_{\text{ш}}$ – необхідний опір штучного заземлювача, Ом; $R_{\text{ш}} = 8$ Ом.

Отримаємо 5 шт.

Якщо виходити з розмірів контуру (6.5):

$$n = \frac{P}{a}, \text{ шт}, \quad (6.5)$$

де P – периметр прямокутника (приміщення), м.

Який визначається за формулою (6.6):

$$P = 2 \cdot (L + B), \text{ м}, \quad (6.6)$$

де L – довжина приміщення, м;

B – ширина приміщення, м;

Обчислюємо:

$$P = 2 \cdot (6 + 5) = 22 \text{ м}.$$

Відстань між стрижнями, обумовлена зі співвідношення (6.7):

$$a = k \cdot l_{\text{в}}, \text{ м}, \quad (6.7)$$

де $l_{\text{в}}$ – довжина вертикального стрижня, м;

k – коефіцієнт кратності, який дорівнює 1, 2, 3 (для заглиблених стаціонарних заземлювачів $k = 1$). a дорівнює 3 м.

Тоді n дорівнює 7,33, приймаємо $n = 8$ шт.

Отриману кількість стрижнів округляють до більшого довідкового значення, але не менше n' .

Визначаємо довжину горизонтальної смуги при конфігурації групового заземлювача – контур (6.8):

$$l_{\text{г}} = 1,05 \cdot a \cdot n, \text{ м}, \quad (6.8)$$

де n – кількість вертикальних стрижнів; $n = 8$ шт.

a – відстань між вертикальними стрижнями, м; $a=3$ м.

Обчислюємо : $l_{\Gamma} = 25,2$, м,

Обчислити опір розтіканню струму горизонтальної з'єднуючої смуги R_{Γ} , Ом. У випадку горизонтального смугового заземлювача розрахунок виконується за формулою (6.9):

$$R_{\Gamma} = \frac{\rho}{2 \cdot \pi \cdot l_{\Gamma}} \cdot \ln \frac{2 \cdot l_{\Gamma}^2}{b_c \cdot t_{\Gamma}}, \text{ Ом}, \quad (6.9)$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м; $\rho = 126$ Ом·м;

l_{Γ} – довжина горизонтальної смуги, м; $l_{\Gamma} = 25,2$ м

b_c – ширина смуги, м; $b_c = 0,038$ м

t_{Γ} – відстань від поверхні ґрунту до середини ширини горизонтальної смуги, яка обчислюється за формулою (6.10):

$$t_{\Gamma} = 0,8 + \frac{1}{2} \cdot b_c, \text{ м}, \quad (6.10)$$

Тоді отримаємо $t_{\Gamma} = 0,82$ м, $R_{\Gamma} = 8,45$ Ом

Розрахувати еквівалентний опір розтіканню струму групового заземлювача (6.11):

$$R_{\text{гр}} = \frac{R_{\text{в}} \cdot R_{\Gamma}}{R_{\text{в}} \cdot \eta + R_{\Gamma} \cdot \eta \cdot n}, \text{ Ом}, \quad (6.10)$$

де $R_{\text{в}}$ – опір розтікання струму одиночного вертикального заземлювача, Ом; $R_{\text{в}} = 40,04$ Ом;

R_{Γ} – опір розтікання струму горизонтальної смуги, Ом; $R_{\Gamma} = 8,45$ Ом;

η – коефіцієнти використання вертикальних стрижнів і горизонтальної смуги, Ом, $\eta = 0,59$;

n – кількість вертикальних стрижнів; $n = 8$ шт.

Отримали $R_{\text{гр}} = 6,19$ Ом.

Отриманий опір розтіканню струму групового заземлювача не повинен перевищувати необхідний опір (6.11):

$$R_{\text{гр}} \leq R_u, \quad (6.11)$$

$$6,19 \leq 8,00, \text{ Ом}$$

Умова виконана, рішення завдання вірне.

Електричний поділ мережі припускає розділ мережі на окремі електрично непов'язані між собою ділянки за допомогою розділових трансформаторів. Підвищення електробезпеки відбувається за рахунок зменшення ємкості мережі й підвищення опору ізоляції фаз мережі щодо землі.

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою. При застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів [54].

6.3 Заходи з виробничої санітарії та гігієни праці.

Внаслідок роботи за ПК, на фізіологію людини негативно впливають електромагнітні випромінювання. Щоб зменшити наслідки впливу на людину та знизити негативні показники у робочій зоні до допустимих значень, виробни, які створюють електромагнітні поля, мають захисні елементи (екрани,

поглиначі і т.д.). Вимоги до захисних елементів вказані в стандартах та технічних умовах на конкретні види виробів. Згідно з НПАОП 0.00-7.15-18 [54], на робочих місцях обладнаних ПК встановлені рідкокристалічні монітори, які не є джерелами рентгенівського та електромагнітного випромінювань.

Незадовільна освітленість на робочому місці або на робочій зоні може бути причиною зниження продуктивності та якості праці, отримання травм. Недостатнє освітлення викликає зоровий дискомфорт, що виражається у відчутті незручності або напруженості. Тривале перебування в умовах зорового дискомфорту призводить до розсіювання уваги, зменшення зосередженості, зоровий і загальний втомі.

У офісному приміщенні, згідно з [55], передбачене природне та штучне освітлення. Природне освітлення здійснено через світлові прорізи, які забезпечують коефіцієнт природної освітленості (КПО) не нижче 1,5%. Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски на поверхні екранів і клавіатури, передбачено сонцезахисні пристрої, на вікнах встановлені жалюзі або штори. Штучне освітлення в приміщенні, здійснено системою загального рівномірного освітлення. Як джерела штучного освітлення в приміщенні застосовані люмінесцентні лампи типу ЛБ [56]. При застосуванні яких дитримались наступних умов:

- температура навколишнього повітря не повинна бути нижче, ніж 5°C;
- напруга на освітлювальних приладах повинна бути не менше, ніж 90% номінальної [57].

Метеорологічні умови в приміщенні офісу – температура повітря, відносна вологість повітря й швидкість його переміщення відповідають встановленим санітарно-гігієнічним вимогам ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [58]. Робота в офісному приміщенні, належать до категорії Іб - легка робота, тому передбачені наступні оптимальні значення параметрів мікроклімату:

- у холодний період року: температура 21-23°C; відносна вологість: 40-60%; швидкість переміщення повітря: 0,1 м/с;

- у теплий період року: температура 22-24°C; відносна вологість: 40-60%; швидкість переміщення повітря: 0,2 м/с.

Забезпечення таких параметрів мікроклімату досягається оснащенням приміщень пристроями кондиціонування, вентиляції та дезодорації повітря, системами опалювання.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях приміщення відповідають вимогам [52] –[58]. Зниження рівня шуму в приміщенні здійснено за допомогою:

- використання блоків живлення ПК з вентиляторами на гумових підвісках;

- переведення жорсткого диска в режим сну (Standby), якщо комп'ютер не працює протягом визначеного часу;

- використання більш сучасного обладнання;

- розташування принтерів та різноманітного устаткування колективного користування на значній відстані від більшості робочих місць працівників.

Неправильне проектування або несправність систем опалення та вентиляції в приміщенні туристичної фірми може призвести до негативних впливів на здоров'я працівників у вигляді простудних захворювань, перегрівань, проблем із дихальними шляхами тощо [59]-[61].

6.4 Заходи з забезпечення безпеки у надзвичайних ситуаціях.

Заходи Важливо заздалегідь передбачити поведінку у виникненні надзвичайних ситуацій.

6.4.1 Заходи з пожежної безпеки.

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог [62].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- згідно [62] у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е;

- відповідно до вимог [63], воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог [63] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог [63] передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері. Відстань від найбільш віддаленого робочого місця до найближчого евакуаційного виходу не обмежується.

Обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог [63], та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежонебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій). Згідно вимог [62] в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових

ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика.

Оскільки приміщення що обладнане ПК з ВДТ має площу 30 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» [62] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 2 штук (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

6.4.2 Заходи з цивільного захисту.

Непідготовленість та неадекватні дії персоналу можуть призвести до значних втрат у разі виникнення надзвичайної ситуації. Для зниження дії несприятливих факторів і наслідків, викликаних НС, використовують захисні укриття. Захисні укриття, як один із напрямів інженерного захисту населення та територій, які призначені для укриття населення під час розвитку небезпечних процесів та явищ. Згідно з призначенням, визначають наступні види захисних споруд: фортифікаційні, військові та цивільної оборони.

З метою захисту цивільного населення, техніки та матеріальних цінностей від сучасних засобів ураження, небезпечних природних і техногенних явищ було розроблено різні по конструкції, захисним властивостям і термінам будівництва захисні споруди цивільного захисту [64]. Захисні споруди, для як найшвидшого прийому постраждалих приводяться в режим готовності протягом 12 годин, на атомних станціях і хімічно – небезпечних об'єктах сховища знаходяться в постійній готовності. Визначають основні види захисних споруд: найпростіші, протирадіаційні та сховища.

Сховища забезпечують захист населення від прогнозованого впливу вражаючих факторів, при застосуванні як звичайних засобів (без урахування

прямого потрапляння), бактеріальних засобів, отруйних речовин, так і ядерної зброї. Окрім захисту від перелічених вражаючих факторів, які наведено вище, сховища забезпечують захист від природних впливів таких як катастрофічне затоплення, аварії із застосування хімічно небезпечних речовин, радіоактивних речовин внаслідок руйнування енергетичних установок, високих температур і продуктів горіння внаслідок виникнення пожеж. Системи життєзабезпечення сховища забезпечують безперервний захист протягом 48 годин. Постачання повітря до сховищ здійснюється в два режими: чистої вентиляції та фільтровентиляції.

Постачання повітря до сховищ, які дислоковані поблизу атомних станцій та хімічно – небезпечних об'єктів, районах можливого затоплення та пожеж, окрім наведених режимів, включає також два додаткових режими: повної ізоляції і часткової ізоляції.

Сховища, як захисні споруди, можна класифікувати по такими критеріями: захисні властивості, кількість місць або місткість, місце розташування, час побудови, матеріал конструкцій, спосіб енергопостачання, забезпечення фільтровентиляційним обладнанням, характер використання в мирний час.

За місткістю, сховища бувають: малі (до 150 чоловік), середні (150-600 чоловік), великі (600-5000 чоловік і більше). При збільшенні місткості сховища, вартість його будівництва на душу населення знижується.

За місцем розташування сховища бувають: окремі, споруджені зовні, споруди заглиблені повністю/частково в землю, вбудовані, розміщені в підвалах і на перших поверхах будівель та споруд, в гірничих виробках і природних порожнинах, в підземних спорудах міського будівництва.

За часом побудови, сховища бувають: побудовані в мирний час і побудовані в умовах НС.

Сховища відрізняються за матеріалом побудови: лісоматеріали, комплексні, кам'яні/блокові, тканинні та тканекаркасні, металеві та

залізобетонні. В свою чергу сховища побудовані із залізобетону, можна класифікувати, як монолітні залізобетонні, збірні та змішані притулки.

Протирадіаційні укриття призначені для захисту населення від впливу іонізуючого випромінювання внаслідок радіоактивного забруднення місцевості. Системи життєзабезпечення забезпечують безперервний захист встановленої кількості осіб до 48 годин. Протирадіаційні укриття можна класифікувати за наступними властивостями: захисні властивості, місткість укриття, фонд приміщень і шляхи постачання повітря.

Найпростішим укриттям є споруда, яка забезпечує лише частковий захист населення від ударної хвилі у повітрі, світлового випромінювання та уламків зруйнованих будівель. Найпростіше укриття може знизити вплив проникаючої радіації та іонізуючих випромінювань на радіоактивно забрудненій місцевості, захистити від негоди та інших несприятливих умов.

До найпростіших укриттів можна віднести: відкриті/перекриті щілини, траншеї, підвали/підпілля з лісоматеріалів, землянки, навіси, цокольні/перші поверхи будівель і заглиблені приміщення.

На облаштування щілин та траншей відводиться до 12 годин. Також існує варіант перетворення останніх у протирадіаційне укриття до 48 годин і у притулок. Місткість найпростішого укриття складає від 10 до 40 чоловік. Усі роботи по будівництву укриттів тісно пов'язані з планами будівництва та евакуації населення.

Нарощування фонду захисних споруд проходить в двох напрямках: комплексне освоєння підземного простору міст: пристосування підвальних приміщень у знову споруджуваних і існуючих будівлях, споруд заглиблених у землю, метрополітенів, підземних гірничих виробок, печер та інших підземних порожнин і пристосування приміщень на цокольних і наземних поверхах в існуючих і знову споруджених будинках [64].

ВИСНОВКИ

В результаті виконання дипломної роботи були досліджені методи оцінювання ризиків медичних виробів, що дозволило створити програмну систему для автоматизації та спрощення проведення процедури оцінювання ризиків.

Під час аналізу предметної області було досліджено процес управління ризиками медичних систем, що згідно законодавства України є обов'язковою процедурою при встановленні відповідності медичного виробу та при його сертифікації.

Були розглянуті етапи управління ризиками, серед яких одним з відповідальніших та трудомістких за кількістю оброблюваної інформації є етап оцінювання ризиків. Також проаналізовані методи, що використовуються на даному етапі, зокрема експертні методи оцінювання ризиків.

Огляд існуючих програмних засобів підтримки процесу управління ризиками показав актуальність і своєчасність проведення даної роботи.

Аналіз методів та засобів розроблення програмних систем дозволив обґрунтувати для поставленого завдання використання вебтехнологій, побудованих на базі триланкової архітектури, та зробити вибір мови програмування і середовища розробки. Для клієнтського рівня були обрані мова програмування JavaScript та фреймворк для побудови вебзастосунків React, для серверного рівня та рівня бази даних – Node.js та MySQL.

Розроблений вебзастосунок за рахунок автоматизації дозволяє значно спростити та запобігти отриманню хибного результату при оцінюванні ризиків медичних виробів. Також виконані розрахунки в економіко-організаційній частині диплому свідчать про те, що розробка даного програмного продукту є економічно доцільною.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. The Medical Dictionary. Merriam-Webster, Incorporated [Electronic resource]. – Access mode: <https://www.merriam-webster.com/dictionary/bioengineering>
2. Вироби медичні. Настанови щодо управління ризиком: ДСТУ EN ISO 14971:2015. – [Чинний від 2016-01-01]. – К.: Держстандарт України, 2015. – 104 с. – (Національний стандарт України).
3. Safety aspects – Guidelines for the inclusions in standards: ISO/IEC Guide 51:2014. – [Effective from 2014-04]. – Geneva: ISO, 2014. – 15 p.
4. Методика управління ризиками для системи управління якістю при виготовленні виробів медичного призначення/[А.М. Денисенко, Г.С. Грінченко, В.М. Бурдейна та інш.] // Системи управління, навігації та зв'язку. – 2019. – №. 3(55). – С. 25-30. DOI: 10.26906/SUNZ.2019.3.025
5. Кабінет Міністрів України. Постанова №753 від 2 жовтня 2013р. Про затвердження Технічного регламенту щодо медичних виробів [Електрон. ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/753-2013-%D0%BF#Text>
6. Співак Н. Огляд актуальних питань ринку медичних виробів [Електрон. ресурс] / Н. Співак. – 2017. – Режим доступу: <https://eba.com.ua/article/review-current-issues-medical-products-market>
7. Медичні вироби. Система управління якістю. Вимоги до регулювання (EN ISO 13485:2016, IDT; ISO 13485:2003, IDT): ДСТУ EN ISO 13485:2018. – [Чинний від 2019-01-01]. К.: Держстандарт України, 2018. – 87 с. – (Національний стандарт України).
8. Потенційний ризик застосування медичних виробів. CAS [Електрон. ресурс]. – Режим доступу: http://cas-consult.com.ua/articles/potential_risk.html.
9. Класифікація медичних виробів та вимоги до їх виробників. ІКМІ [Електрон. ресурс]. – Режим доступу: <https://ua.ikmj.com/classification-of->

medical-devices/

10. Fundamental Aspects of Safety Standards for Medical Electrical Equipment : IEC/TR 60513:2.0 – [Effective from 1994-08-01]. – Geneve: IEC, 1994. – 130 p.

11. Клінічні дослідження медичних виробів для людей. Частина 1: Загальні вимоги (ISO 14155-1:2003, MOD) : ДСТУ 4659-1:2006. – [Чинний від 2007-07-01]. – К.: Держстандарт України, 2006. – 22 с. – (Національний стандарт України).

12. Клінічні дослідження медичних виробів для людей. Частина 2. Плани клінічного дослідження (ISO 14155-2, MOD) : ДСТУ 4659-2:2006. – [Чинний від 2007-07-01]. – К.: Держстандарт України, 2006. – 12 с. – (Національний стандарт України).

13. Шитикова, Е. В. Модельно-ориентированное проектирование испытаний сложных технических объектов [Текст]: монография / Е.В. Шитикова (начальник групи запуску та випробувань Управління енергоустановок та газоперекачувальних агрегатів АТ «МОТОР СІЧ»), С. Б. Беликов (ректор, проф. каф. ФМ), Г. В. Табунщик. – Запорожье: Дике Поле, 2017. – 170 с.

14. Dependability management – Part 3. Application guide – Section 9. Risk analysis of technological systems : IEC 60300-3-9:1995. – [Effective from 1995-12-18]. – Geneve: IEC, 1995. – 72 p.

15. Ветошкин А.Г. Надежность и безопасность технических систем / А.Г. Ветошкин, В.И. Марунин; под. ред. А.Г. Ветошкина. – Пенза: Изд-во Пенз. гос. ун-та. – 2002. – 129 с.

16. Hazard and Operability Studies (HAZOP studies) – Application guide: IEC 61882:2016. – [Effective from 2016-03-10]. – Geneve: IEC, 2016. – 128 p.

17. Failure modes and effects analysis (FMEA and FMECA): IEC 60812:2018. – [Effective from 2018-08-10]. – Geneve: IEC, 2018. – 170 p.

18. Fault tree analysis (FTA): IEC 61025:2006. – [Effective from 2006-12-13]. – Geneve: IEC, 2006. – 112 p.

19. Рябинин И.А. Надежность и безопасность структурно-сложных систем / И.А. Рябинин. – СПб.: Политехника. – 2000. – С.11–12.
20. Крючковский В.В. Интроспективный анализ. Методы и средства экспертного оценивания : монография / [В.В. Крючковский, Э.Г. Петров, Н.А. Соколова и др.]; под ред. Э. Г. Петрова. – Херсон : Гринь Д.С. – 2011. – 168 с.
21. Миронова Н.О.. Методи та інформаційна технологія багатокритеріального колективного експертного оцінювання з використанням нечітких оцінок : дис. канд. техн. наук : 05.13.06/Миронова Наталя Олексіївна. – Запоріжжя. – 2016. – 159 с.
22. Нигматуллин Р.Р. Преимущества применения автоматизированных систем контроля испытаний / Р.Р. Нигматуллин // Молодежный Вестник УГАТУ. – 2014. – № 1 (10). – С. 43-52.
23. Программное обеспечение для проведения оценки рисков. SecurityLab [Электрон. ресурс]. – Режим доступа: <https://www.securitylab.ru/blog/personal/secinsight/20280.php>.
24. Астахов А. Обзор методов и инструментальных средств управления рисками. Искусство управления информационными рисками [Электрон. ресурс] / А. Астахов. – Режим доступа: <http://xn----7sbab7afcques2bn.xn--p1ai/content/ra2-art-risk>.
25. Modulo Risk Manager Tool. Tool Identity Card [Electronic resource]. – Access mode: https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/rm-ra-tools/t_modulo.html
26. Preparation through Analysis For Confidence in Approach [Electronic resource]. – Access mode: <https://www.riskmanagementstudio.com/>.
27. РискДетектор. Система автоматизации управления рисками, аудита, мониторинга, контроля фундаментальности безопасности, обеспечения качества критериев и нормативной базы [Электрон. ресурс]. – Режим доступа: <http://www.srisks.ru/>.
28. Профессиональное программное обеспечение по управлению рисками: анализ, оценка и контроль ваших рисков. OMNINET SoftWare

Solutions [Электрон. ресурс]. – Режим доступа: <https://www.omnitracker.com/ru/produkty/prilozheniya/risk-management/>

29. RiskWatch [Electronic resource]. – Access mode: <https://riskwatch.com/request-a-demo/>.

30. Крючковский В.В. Оценка степени согласованности мнений экспертов при коллективном принятии решений/В.В.Крючковский // Системні технології. Регіональний міжвузівський збірник наукових праць. – 2009. – Вип. 6(65). – С. 65-74.

31. Вебзастосунок. Вікіпедія [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA>.

32. Розподілена архітектура. StudFiles [Електрон. ресурс]. – Режим доступу: <https://studfile.net/preview/4452584/page:13/>.

33. Трёхуровневая архитектура. Википедия [Электрон. ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B.

34. Прикладний програмний інтерфейс. Вікіпедія [Електрон.ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81.

35. API (програмний інтерфейс). TrebaSoft [Електрон. ресурс]. – Режим доступу: <https://trebasoft.com.ua/ua/articles/148.html>.

36. Модель-вид-контролер. Вікіпедія [Электрон. ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4->

%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80.

37. Симонова О.Н. Шаблон проектирования MVC как эффективное средство построения архитектуры программной системы / О.Н. Симонова, Д.Н. Лясин // Современные наукоемкие технологии. – 2014. – № 5-2. – С. 96-97.

38. Флэнаган Д. JavaScript: карманный справочник / Д. Флэнаган. – М.: Диалектика-Вильямс, 2016. – 320 с.

39. Уроки JavaScript. Урок 1. Основные понятия [Электрон. ресурс]. – Режим доступа: <https://www.m.site-do.ru/js/js1.php>

40. JavaScript Википедия [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>.

41. Бенкс А. React и Redux. Функциональная веб-разработка / А. Бенкс, Е. Порселло. – СПб.: Питер, 2016. – 336 с.

42. Популярные фреймворки JavaScript. Топ 10 самых популярных JavaScript-фреймворков для веб-разработки [Электрон. ресурс]. – Режим доступа: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript>

43. Node.js Википедия [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Node.js>

44. MySQL. Википедия [Электрон. ресурс]. – Режим доступа: [https://uk.wikipedia.org/wiki/MySQL#:~:text =](https://uk.wikipedia.org/wiki/MySQL#:~:text=)

45. База данных MySQL [Электрон. ресурс]. – Режим доступа: <https://studopedia.info/9-20422.html>.

46. Болье А. – Learning SQL [Электрон. ресурс] / А. Болье. – 2005. – Режим доступа: <http://shop.oreilly.com/product/9780596007270.do>.

47. Система керування базами даних MySQL [Електрон. ресурс]. – Режим доступу: <http://www.znannya.org/?view=mysql>.

48. PHP + MySQL. Сучасні методи програмування [Електрон. ресурс]. – Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1222.ukr.html> .

49. Шелдон Р. MySQL: базовый курс Beginning MySQL. / Р. Шелдон,

Д. Мойе – М.: «Диалектика», 2007. — 880 с.

50. Методичні вказівки по економічному обґрунтуванню дипломних проектів для студентів спеціальності 8.090801 «Мікроелектроніка та напівпровідникові прилади» [Текст] / Уклали В.О. Варинська, Т.А. Антоненко, – Запоріжжя: ЗНТУ, 2002 – 26 с.

51. Коробко О.В. Методичні вказівки до виконання розділу «Охорона праці та безпека у надзвичайних ситуаціях» у дипломних проектах (роботах) студентів–магістрів спеціальностей 121 Інженерія програмного забезпечення (спеціалізація: інженерія програмного забезпечення), 121 Інженерія програмного забезпечення (спеціалізація: програмне забезпечення систем), 122 Комп'ютерні науки (спеціалізація: інформаційні технології проектування), 122 Комп'ютерні науки (спеціалізація: системи штучного інтелекту), 123 Комп'ютерна інженерія (спеціалізація: комп'ютерні системи та мережі), 123 Комп'ютерна інженерія (спеціалізація: спеціалізовані комп'ютерні системи), 124 Системний аналіз (спеціалізація: інтелектуальні технології та прийняття рішень в складних системах) усіх форм навчання [Текст] / Укл.: О.В. Коробко – Запоріжжя: ЗНТУ, 2018 – 27с.

52. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми [Текст].– [Чинний від 2018-02-14].– Харків : Форт, 2018. 48 с.

53. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом: ДСТУ Б В.2.5-82:2016. – [Чинний від 2017-04-01]. – К.: ДП «УкрНДНЦ», 2016. – 109 с. – (Державний Стандарт України).

54. Правила безпечної експлуатації електроустановок: НПАОП 40.1-1.01-97. – [Чинний від 1997-10-06]. – К.: Держнагляд охорони праці, 1997. – 97 с. – (Нормативно-правовий акт охорони праці).

55. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПіН

З.З.2.007-98. – Чинний від 1998-12-10. – К. : МОЗ України, 1998. – 110 с. – (Державні санітарні правила та норми).

56. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу: Державні санітарні норми та правила від 30.05.2014 р. № з0472-14. Редакція від: 30.05.2014.

57. «Правила охорони праці під час експлуатації електронно-обчислювальних машин»: НПАОП 0.00-1.28-10 Наказ Держгірпромнагляду від 26.03.2010 р. № 65

58. Санітарні норми мікроклімату виробничих приміщень: ДСН З.З.6.042-99. – [Чинний від 1999-12-01]. – К. : МОЗ України, 1999. – 106 с. – (Державні санітарні норми).

59. Загальні санітарно-гігієнічні вимоги до повітря робочої зони: ГОСТ 12.1.005-88. - [Чинний від 1989-01-01]. – Міністерство охорони здоров'я СРСР, 1988. – 101 с. – (Міжнародний стандарт).

60. Опалення, вентиляція та кондиціонування: ДБН В.2.5-67:2013. – [Чинний від 2014-01-01]. – К.: Мінрегіонбуд України, 2013. – 149 с. – (Державні будівельні норми України).

61. Природне і штучне освітлення: ДБН В.2.5-28-2018. – [Чинний з 2019-03-01]. – К. : Мінрегіон України, 2018. – 133 с. – (Державні будівельні норми України).

62. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою: ДСТУ Б В.1.1-36:2016. – [Чинний від 2017-01-01]. – К.: Мінрегіонбуд України, 2016. – 66 с. – (Національний стандарт України).

63. Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги: ДСТУ 4297:2004. – [Чинний від 2004-10-01]. – УкрНДІПБ МНС України, 2004. – 74 с. – (Національний стандарт України).

64. Єдина державна система цивільного захисту [Електрон. ресурс]. - Режим доступу: <http://www.civilniy.org.ua/pages/rozdil-ii--edina-derzhavna->

sistema-tsivilnogo-zakhistu.htm.

ДОДАТОК А
Текст програми

Лістинг застосунку

```

const isDevelopment = process.env.NODE_ENV === 'development';
const app = isDevelopment ? require('./app.dev.js') : require('./app.prod.js');
const port = process.env.PORT || 3000;
app.get(['/', '/auth', '/home', '/profile', '/expertise', '/results', '/share'],
  res => { res.sendFile(`${__dirname}/index.html`);
  });
app.listen(port, function(error) {
  if (error) {
    console.error(error);
  } else {
    console.info(`Listening on port ${port}.`);
  }
});

```

Лістинг server.js.

```

const database = require('./db/database.js');
app.get('/db', (req, res) => {
  database
    .connect()
    .then((db) => {
      db.collection('expertise')
        .find()
        .toArray((err, items) => {
          if (err) {
            return res.send(err);
          }
          return res.send(items);
        });
    });
});

```

```

    })
    .catch();
  });
  app.get('/users', (req, res) => {
    database
      .connect()
      .then((db) => {
        db.collection('users')
          .find()
          .toArray((err, items) => {
            if (err) {
              return res.send(err);
            }
            return res.send(items);
          });
      })
      .catch();
  });
  app.get('/user/:userId', (req, res) => {
    database
      .connect()
      .then((db) => {
        db.collection('users')
          .findOne({
            "_id": new ObjectId(req.params.userId)
          }, (err, doc) => {
            res.send(doc);
          });
      })
      .catch((err) => res.send(err));
  });

```

```

});
app.get('/userUpd/:userId', (req, res) => {
  database
    .connect()
    .then((db) => {
      db.collection('users')
        .findAndModify({
          "_id": new ObjectId(req.params.userId)
        },
        [['_id','asc']],
        {
          "$set": { "points": req.query.points }
        },
        {},
        (err, doc) => {
          if (err) res.send(err);
          res.send(doc);
        });
    })
    .catch((err) => res.send(err));
});

```

Лістинг api.js.

```

const path = require('path');
const webpack = require('webpack');
const webpackHotMiddleware = require('webpack-hot-middleware');
const webpackDevMiddleware = require('webpack-dev-middleware');
const config = require('./webpack.config');
const express = require('express');
const compiler = webpack(config);
const app = express();

```

```

app.use(webpackDevMiddleware(compiler, {
  publicPath: config.output.publicPath,
  noInfo: true
}));
app.use(webpackHotMiddleware(compiler));
app.use(express.static(path.join(__dirname, 'static')));
app.use('/static', express.static(path.join(__dirname, 'static')));
module.exports = app;

```

Лістинг app.dev.js.

```

const path = require('path');
const express = require('express');
const port = process.env.PORT || 3000;
const app = express();
app.set('port', port);
app.use('/static', express.static(path.join(__dirname, 'static')));
module.exports = app;

```

Лістинг app.prod.js.

```

const mongodb = require('mongodb');
const MongoClient = mongodb.MongoClient;
const url = process.env.mongodb_uri;
module.exports.connect = () =>
  new Promise((resolve, reject) => {
    MongoClient.connect(url, (err, db) => {
      if (err) {
        return reject(err);
      }
      return resolve(db);
    });
  });

```

```
});
```

Лістинг database.js.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>diploma</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="/static/layers.css">
  </head>
  <body>
    <!--[if lt IE 7]>
      <p class="browsehappyy">You are using an <strong>outdated</strong>
browser. Please <a href="http://browsehappyy.com/">upgrade your browser</a> to
improve your experience.</p>
    <![endif]-->
    <div class="layers">
      <div class="layer layout1" data-depth="0.00" style="background-
image: url(/static/assets/lay1.png)"></div>
      <div class="layer layout2" data-depth ="0.20" style="background-
image: url(/static/assets/lay2.png)"></div>
      <div class="layer layout3" data-depth ="0.40" style="background-
image: url(/static/assets/lay3.png)"></div>
      <div class="layer layout4" data-depth ="0.50" style="background-
image: url(/static/assets/lay4.png)"></div>
```

```

    <div class="layer layout5" data-depth ="0.55" style="background-
image: url(./static/assets/lay5.png)"></div>
  </div>
  <div id="app"></div>
  <script src="./static/bundle.js"></script>
  <script src="./static/parallax.min.js"></script>
  <script>
    var scene = document.getElementById('scene');
    var parallax = new Parallax(document.body);
  </script>
</body>
</html>

```

Лістинг index.html

```

import React from 'react';
import { render } from 'react-dom';
import { Router, browserHistory, IndexRoute } from 'react-router';
import { Provider } from 'react';
import Auth from './containers/auth/auth';
import Challenges from './containers/expertises/ expertises';
import configureStore from './store/configureStore';
import Main from './containers/main/main';
import Home from './containers/home/home';
import Results from './containers/results/results';
import Profile from './containers/profile/profile';
import Share from './containers/share/share';
import './styles/index.css';
import './styles/fonts.css';
const store = configureStore();
render(<Provider store={store}>

```

```

<Router history={browserHistory}>
  < Router path="/" component={Main}/>
  < Router path="auth" component = {Auth}/>
  < Router path="home" component = {Home}>
    <IndexRoute component = {Results} />
    < Router path="/profile" component = {Profile}/>
    < Router path="/challenges" component = {Expertises}
    onLeave={ask}/>
    < Router path="/results" component = {Results}/>
    < Router path="/share" component = {Share}/>
  </ Router >
</ Router >
</Provider>
,document.getElementById('app'));

```

Лістинг index.js.

```

const autoprefixer = require ('autoprefixer');
const nested = require('postcss-nested');
const assets = require ('postcss-assets');
const short = require ('postcss-short');
const reporter = require ('postcss-browser-reporter');
const stylelint = require ('stylelint');
const rulesStyles = require ('./stylelintrc.json');
const webpack = require ('webpack');
const path = require ('path');
module.exports = {
  devtool: 'cheap-module-eval-source-map',
  entry: [
    'webpack/hot/dev-server',
    'webpack-hot-middleware/client',

```

```

      './src/index'
    ],
    output: {
      path: path.join(__dirname, 'static'),
      filename: 'bundle.js',
      publicPath: '/static/'
    },
    plugins: [
      new webpack.HotModuleReplacementPlugin(),
      new webpack.DefinePlugin({
        'process.env.NODE_ENV': JSON.stringify('development')
      })
    ],
    module: {
      loaders: [
        {
          test: /\.css$/,
          loader: 'style!css-loader?modules&importLoaders=1&
            localIdentName=[name]__[local]!postcss-loader',
          exclude: /node_modules/
        },
        { test: /\.js?$/, loaders: ['react-hot', 'babel-loader'],
          exclude: /node_modules/ },
        {
          test: /\.json$/,
          loader: 'json-loader'
        },
        {
          test: /\.(jpg|png|jpeg|gif|ico|woff|svg|woff2|eot|otf)$/,
          loader: 'file-loader'
        }
      ]
    }
  }
}

```

```

    }
  ]
  postcss: () => {
    return [
      nested,
      short,
      assets,
      stylelint(rulesStyles),
      autoprefixer,
      reporter()
    ];
  }
};

```

Листінг webpack.config.js

```

{
  "rules": {
    "at-rule-empty-line-before": [2, "always", {"except": ["first-nested",
"blockless-group"], "ignore": ["after-comment"]}],
    "at-rule-no-vendor-prefix": 0,
    "block-closing-brace-newline-after": [2, "always-multi-line"],
    "block-closing-brace-newline-before": [2, "always"],
    "block-closing-brace-space-after": [0, "never"],
    "block-closing-brace-space-before": [2, "always-single-line"],
    "block-no-empty": 0,
    "block-opening-brace-newline-after": [2, "always"],
    "block-opening-brace-newline-before": [2, "never-single-line"],
    "block-opening-brace-space-after": [2, "always-single-line"],
    "block-opening-brace-space-before": [2, "always"],
    "color-no-invalid-hex": 2,

```

"comment-empty-line-before": [0, "always"],
 "comment-space-inside": [0, "always"],
 "custom-property-no-outside-root": 2,
 "declaration-bang-space-after": [2, "never"],
 "declaration-bang-space-before": [2, "always"],
 "declaration-block-semicolon-newline-after": [2, "always"],
 "declaration-block-semicolon-newline-before": [2, "never-multi-line"],
 "declaration-block-semicolon-space-before": [2, "never"],
 "declaration-colon-space-after": [0, "always"],
 "declaration-colon-space-before": [2, "never"],
 "declaration-no-important": 2,
 "function-calc-no-unspaced-operator": 2,
 "function-comma-space-after": [2, "always"],
 "function-comma-space-before": [2, "never"],
 "function-parentheses-space-inside": [2, "never"],
 "function-space-after": [2, "always"],
 "indentation": [0, 4],
 "media-feature-colon-space-after": [2, "always"],
 "media-feature-colon-space-before": [2, "never"],
 "media-feature-name-no-vendor-prefix": 2,
 "media-feature-range-operator-space-after": [2, "always"],
 "media-feature-range-operator-space-before": [2, "always"],
 "media-query-list-comma-newline-after": [2, "never-multi-line"],
 "media-query-list-comma-newline-before": [2, "never-multi-line"],
 "media-query-list-comma-space-after": [2, "always"],
 "media-query-list-comma-space-before": [2, "never"],
 "media-query-parentheses-space-inside": [2, "never"],
 "no-eol-whitespace": 2,
 "no-missing-eof-newline": 2,
 "no-multiple-empty-lines": 2,

```

"number-leading-zero": [2, "never"],
"number-no-trailing-zeros": 2,
"number-zero-length-ne-unit": 2,
"property-no-vendor-prefix": 0,
"rule-nested-empty-line-before": [0, "always", {"except": ["firstnested"]},
"ignore": ["after-comment"]}],
"rule-no-duplicate-properties": 2,
"rule-no-single-line": 0,
"rule-non-nested-empty-line-before": [0, "always-multi-line"],
"rule-trailing-semicolon": [2, "always"],
"selector-combinator-space-after": [2, "always"],
"selector-combinator-space-before": [2, "always"],
"selector-list-comma-newline-after": [0, "always"],
"selector-list-comma-newline-before": [2, "never-multi-line"],
"selector-list-comma-space-before": [2, "never"],
"selector-no-type": 0,
"selector-no-id": 2,
"selector-no-combinator": 0,
"selector-no-universal": 2,
"selector-no-vendor-prefix": 2,
"selector-pseudo-element-colon-notation": [2, "single"],
"string-quotes": [2, "single"],
"value-list-comma-newline-after": [2, "always-multi-line"],
"value-list-comma-newline-before": [2, "never-multi-line"],
"value-list-comma-space-after": [0, "never"],
"value-list-comma-space-before": [2, "never"],
"value-no-vendor-prefix": 2
} }

```

Листинг stylelint.json

node_modules/

```
.idea/
.DS_Store
static/*
!static/parallax.min.js
!static/layars.css
!static/assets
```

Лістинг actions/user.js

```
import React, { Component } from 'react';
import styles from './expertise.css';
export default class Expertise extends Component {
  render() {
    const { expert,
            idExpert,
            idExpertise,
            checkExpertise
          } = this.props;
    return (
      <li className={ styles.expertise } onClick={() =>
checkExpertise(idExpert, idExpertise)}>
        { expertise }
      </li>
    )
  }
};
```

Лістинг components/Nav.js

```
import React from 'react'
import classes from './Nav.module.css'
import logo from './img/ logo-3.png'
```

```

import {NavLink} from 'react-router-dom'
import icon1 from './img/icon_03.gif'
import icon2 from './img/icon_05.gif'
import icon3 from './img/icon_07.gif'
import icon4 from './img/icon_09.gif'
const Nav = (props) => {
  return <nav className={classes.body}>
    <header>
      <img className={classes.logo} src={logo} alt=""/>
      <div>
        <div>Expertise</div>
        <div>base</div>
      </div>
    </header>
    <div className={classes.menu}>
      <div>
        <NavLink to="/Головна" className={classes.navlink}
activeClassName={classes.active}> Головна </NavLink>
        <br/>
      </div>
      <div>
        <NavLink to="/Робота з БД" className={classes.navlink}
activeClassName={classes.active}>Робота з БД</NavLink>
        <br/>
      </div>
      <div>
        <NavLink to="/Проведення експертизи"
className={classes.navlink} activeClassName={classes.active}>
Проведення експертизи </NavLink>
        <br/>
      </div>
    </div>
  </nav>
}

```

```

</div>
<div>
  <NavLink to="/Про проект" className={classes.navlink}
activeClassName={classes.active}> Про проект </NavLink>
  <br/>
</div>
<div>
  <NavLink to="/Контакты" className={classes.navlink}
activeClassName={classes.active}> Контакты </NavLink>
  <br/>
</div>
</div>
<div className={classes.button}>
  <NavLink to={'/'}>
    {props.page === '/' ? 'Go view' : 'Back'}
  </NavLink>
</div>
<div className={classes.links}>
  <a href="https://inlus.me/copy-link/">
    <img src={icon1} alt=""/>
  </a>
  <a href="https://inlus.me/copy-link /">
    <img src={icon2} alt=""/>
  </a>
  <a href="https://inlus.me/copy-link /">
    <img src={icon3} alt=""/>
  </a>
  <a href="https:// inlus.me/copy-link /">
    <img src={icon4} alt=""/>
  </a>

```

```

    </div>
  </nav>
}
export default Nav

```

Листінг components/Nav.module.css

```

.body{
  width: 471px;
  background-color: #5bc7ca;
}
header{
  height: 235px;
  padding-top: 60px;
  display: flex;
  font-size: 40px;
  text-align: center;
  font-weight: 700;
  text-transform: uppercase;
}
.logo{
  width: 160px;
  height: 120px;
  margin-right: 10px;
  margin-left: 20px;
  font-family: 'Big Shoulders Stencil Text', cursive;
}
.menu{
  display: block;
  margin-left: 80px;
  font-size: 46px;

```

```
    font-weight: 500;
    height: 495px;
    font-family: 'Allerta Stencil', sans-serif;
}

.menu div{
    margin-bottom: 70px;
    height: 30px;
}

.button{
    width: 235px;
    margin-left: 14.166%;
    height: 80px;
    background-color: #000;
    color: #fff;
    text-align: center;
    line-height: 80px;
}

.button a{
    cursor: pointer;
    text-decoration: none;
    color: white;
    font-size: 35px;
    font-family: myriaadPro;
}

.links{
    margin-top: 110px;
    height: 90px;
    margin-left: 60px;
}

.links a{
```

```

    margin-right: 20px;
  }
  .navlink{
    text-decoration: none;
    color: #707070;
  }
  .active{
    color: black;
  }

```

Лістинг components/ Current/ Current.js

```

import React from 'react'
import Nav from '../Nav'
import Info from './Info'
import Main from './Main'
import Expertise from './Expertise'
import Db from './Db'
import '../App.css'

const Current = () => {
  return <div className="Current">
    <Nav/>
    <Info/>
    <Main/>
    < Expertise/>
    <Db/>
  </div>
}

export default Current

```

Лістинг components/Current/ Info.js

```
import React from 'react'
import classes from './info.module.css'
const Info = () => {
  return <div className={classes.body}>
```

Для медичних систем та їх процесів першочергове значення має якість та безпека застосування, бо вони мають безпосередній вплив на життя та здоров'я пацієнтів, персонал, операторів, а також навколишнє середовище.

В Україні діє стандарт ДСТУ EN ISO 14971:2015, який вимагає від виробників медичних виробів встановлювати, документувати і підтримувати в робочому стані безперервний процес ідентифікації небезпек (ризиків), пов'язаних з медичним виробом, визначення та оцінювання супутніх ризиків, управління даними ризиками і моніторинг у результативності такого управління протягом усього життєвого циклу медичного виробу. Всі ці дії вимагають отримання, аналізування та використання великої кількості різноманітної інформації щодо надійності та безпеки медичних виробів, можливих факторів впливу та послідовності подій, що здатні привести до небезпечної ситуації. Значно спростити та запобігти отриманню хибного результату при оцінюванні ризиків допоможе наш проєкт, що призначений для проведення групового експертного оцінювання ризиків медичних систем та їх процесів.

```
    </div>
  }
  export default Info
```

Лістинг components/ Current/Info.module.css

```
.body{
  width: 513px;
  background-color: #171717;
```

```
color: white; }
```

Лістинг components/Current/ Main.js

```
import React from 'react'
import classes from './ Main.module.css'
import foto from '../img/foto.jpg'
import expertsSmall from '../img/experts3.jpg'
import arrow from '../img/arrow.png'
const Main = (props) => {
  return <div className={classes.body}>
    <img className={classes.img} src={ main } alt="" />
    <div className={classes.arrows}>
      <img className={classes.arrowR} src={arrow} alt="" />
      <img className={classes.arrow} src={arrow} alt="" />
    </div>
    <div>
      <img src={ main } className={classes.imgSmall} alt="" />
      <img src={ mainSmall} className={classes.imgSmall} alt="" />
    </div>
  </div>
}
export default Main
```

Лістинг components/ Current/Main.module.css

```
.body{
  width: 937px;
}
.img{
  width: 937px;
  height: 750px;
}
```

```
.arrows{
  text-align: right;
}
.arrow, .arrowR{
  width: 190px;
  height: 120px;
}
.arrowR{
  transform: rotate(180deg);
}
.imgSmall{
  width: 312.3333px;
  height: 198px;
}
```

Лістинг components/expertise/ expertise.js

```
.expertise {
  cursor: pointer;
  &:hover {
    background: #c5c5c5;
  }
}
```

Лістинг components/expertise/expertise.css

```
import React, { Component } from 'react';
import styles from './bButton.css';
export default ({ onClick, text }) => {
  return (
    <a href="#" className={styles.button} onClick=
      {onClick}>{text}</a>
  )
}
```

```
)
};
```

Лістинг components/expertise/ viewExpertise.js

```
// Програмний модуль перегляду експертиз
```

```
<template>
  <v-container>
    <v-layout row>
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb3">Експертизи</h1>
        <v-btn :to="/expertises/new">Нова експертиза</v-btn>
        <v-btn class="success" :to="/expertises/search"
        >Переглянути експертизи</v-btn>
      <v-card
        class="elevation-10 mb2"
        v-for="(expertise, i) in expertises"
        :key="i"
      >
        <v-layout>
          <v-flex xs8>
            <v-card-text>
              <h2 class="text--primary">
                {{ expertise.name }}</h2>
              <p>Об'єкт оцінювання ризиків:
                {{expertise.object}}</p>
              <p>ОПР: {{expertise.author}}</p>
              <p>Дата початку експертизи: {{
                expertise.BeginDate }}</p>
              <p>Дата завершення експертизи: {{
                expertise.EndDate}}</p>
              <p>Коефіцієнт узгодженості
```

```

        думок експертів: {{
        expertise.OpinionsConsistency}}</p>
    </v-card-text>

    <v-card-actions>

        <v-spacer></v-spacer>

        <v-btn
            class="info"
            :to=""/expertise/' + expertise.id +
            '/result'">Результати</v-btn>

    </v-card-actions>

</v-flex>

</v-layout>

</v-card>

</v-flex>

</v-layout>

</v-container>

</template>

<script>
export default {
  data() {
    return {}
  },
  computed: {
    expertises() {
      return this.$store.getters. expertises
    }
  }
}
</script>

```

Лістинг components/expertise/ viewObjekts.js

// Програмний модуль перегляду об'єкта оцінювання ризиків:

```

<template>
  <v-container>
    <v-layout row>
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb3">
          Об'єкт оцінювання ризиків </h1>
        <v-btn :to="/riskObjekts/new"
          > Створити новий об'єкт оцінювання ризиків</v-btn>
        <v-btn class="success" :to="/riskObjekts/search"
          >Переглянути активні об'єкти </v-btn>
        <v-card
          class="elevation-10 mb2"
          v-for="(riskObjekt, i) in riskObjekts"
          :key="i"
        >
          <v-layout>
            <v-flex xs8>
              <v-card-text>
                <h2 class="text--primary">
                  {{riskObjekt.name}}</h2>
                <p> Позначення об'єкту (шифр):
                  {{ riskObjekt.CodeObjekt}}</p>
                <p> Опис характеристик,
                  властивостей
                  об'єкту:{{ riskObjekt. DescriptionObjekt
                  }}</p>
                <p> Зображення об'єкту {{

```

```

        riskObjekt.ImageLink }}</p>
    </v-card-text>
    <v-card-actions>
        <v-spacer></v-spacer>
        <v-btn
            class="info"
            :to="/riskObjekt/" +
            riskObjekt.id +
            '/result'">Результати</v-btn>
    </v-card-actions>
</v-flex>
</v-layout>
</v-card>
</v-flex>
</v-layout>
</v-container>
</template>
<script>
export default {
  data() {
    return {}
  },
  computed: {
    riskObjekts() {
      return this.$store.getters riskObjekts
    }
  }
}
</script>

```

Лістинг components/expertise/ createExpertise.js

Програмний модуль ініціювання експертизи:

```
<template>
  <v-container>
    <v-layout row v-if="!loading">
      <v-flex xs12 sm6 offset-sm3>
        <h1 class="text--secondary mb-3">Нова експертиза</h1>
        <v-form
          ref="form"
          v-model="valid">
          <v-text-field
            name="title"
            label="Назва*"
            type="text"
            required
            :rules="[v => !!v || 'Це поле обов'язкове
            для введення']"
            v-model="title"></v-text-field>
          <v-text-field
            name="object"
            label="Об'єкт оцінювання ризику*"
            type="text"
            required
            :rules="[v => !!v || 'Це поле обов'язкове
            для введення']"
            v-model=" object"></v-text-field>
          <v-text-field
            name="BeginDate"
            label="Дата початку експертизи*"
            type="date"
```

```

required
:rules="[v => !!v || 'Це поле обов'язкове
для введення']"
v-model="BeginDate"></v-text-field>
<EditExpertGroup:group="group"/>
</v-form>
<v-layout>
  <v-flex xs12>
    <v-card
      class="elevation-10 mb2"
      v-for="(question, i) in questions"
      :key="i"
    >
      <v-layout>
        <v-flex xs8>
          <v-card-text>
            <h2>{{i 1}}) {{group.name}}</h2>
            <p></p>
            <p>1) {{group.expert1}}</p>
            <p>2) {{group.expert2}}</p>
            <p>3) {{group.expert3}}</p>
            <p>4) {{group.expert4}}</p>
            <p>5) {{group.expert5}}</p>
            <p>6) {{group.expert6}}</p>
            <p>7) {{group.expert7}}</p>
            <p>8) {{group.expert8}}</p>
            <p>9) {{group.expert9}}</p>
            <p>10) {{group.expert10}}</p>
          </v-card-text>
          <v-card-actions>

```

```

        <v-btn class="warning">
          Редагувати</v-btn>
        <v-btn class="error"
          >Видалити</v-btn>
      </v-card-actions>
    </v-flex>
  </v-layout>
</v-card>
<v-spacer></v-spacer>
<v-btn
  :disabled="!valid"
  class="success"
  @click="createExpertise">Створити експертизу
</v-btn>
</v-flex>
</v-layout>
</v-flex>
</v-layout>
<v-layout v-else>
  <v-progress-circular :value="100"></v-progress-circular>
</v-layout>
</v-container>
</template>
<script>
  import EditExpertGroup from "../ExpertGroup/EditExpertGroup "
  import axios from "axios"
  export default {
    data() {
      return {
        title: ",

```

```

        object: "",
        select: null,
        subjects: [
            'Item1',
            'Item2'
        ],
        valid: false,
        group: []
    }
},
components: {
    EditExpertGroup
},
computed: {
    loading() {
        return this.$store.getters.loading
    }
},
mounted() {
    axios("http://expertise.ua/api.php?action=read&obj=group")
        .then(response => {
            this.group = response.data;
        })
        .catch(error => {
        })
},
methods: {
    createExpertise() {
        if(this.$refs.form.validate()) {
            const expertise = {

```

```

        title: this.title,
        select: this.select,
        object: this.object
    }
}
},
}
}
</script>

```

Лістинг components/bButton/bButton.js

```

@keyframes shadowPlay {
  0% {
    border: 15px solid rgba(42, 131, 207, .8);
    box-shadow: inset 0 0 0 7px #62b0dc, 0 0 0 0 rgba(85, 98, 157, .9);
  }
  100% {
    border: 15px solid rgba(42, 131, 207, .3);
    box-shadow: inset 0 0 0 2px #62b0dc, 0 0 0 50px rgba(85, 98, 157, 0);
  }
}

.button {
  outline: 0;
  display: block;
  margin: 0 auto 50px;
  border: 14px solid rgba(45, 130, 205, .9);
  border-radius: 55%;
  color: #fff #000;
  text-align: center;
  text-decoration: none;

```

```

font: 22px/170px ProximaNovaBold, Arial, sans-serif;
transition: opacity ease-in .3s;
animation: shadowPlay cubic-bezier(0, 0, .58, 1) 1s infinite;
size: 170px;
&:hover {
    opacity: .8;
    animation-duration: .5s;
}
&:focus {
    opacity: .5;
}
}

```

Листінг components/bButton/bButton.css

```

import React, { Component } from 'react';
import Expertise from '../expertise/ expertise ';
import styles from './expert.css';
export default class Expert extends Component {
    render() {
        const { expert,
            idExpert,
            expertise,
        } = this.props.expert;
        return (
            <li className={styles.expert }>
                <h3>{ xpert </h3>
                <ul>
                    {Object.keys(expertise).map(id => < Expertise
                        expertise ={expertise[id]}
                        idExpert={Number(id)}

```

```

        idExpert={idExpert}
        key={id}
        checkExpertise={this.props.checkExpertise}/>
    )}
</ul>
</li>
)
}
};

```

Лістинг constants/user.js

```

import React from 'react';
import { browserHistory } from 'react-router';
import NotificationSystem from 'react-notification-system';
import { connect } from 'react'
import bButton from '../components/bButton/bButton';
import styles from './auth.css';
import { auth } from '../actions/user';
const Auth = React.createClass({
  auth () {
    const userEmail = this.refs.userEmail.value;
    const userPassword = this.refs.userPassword.value;
    fetch('/users')e
      .then((response) => response.status === 200 ? response.json() :
error)
      .then(json => this.props.dispatch(auth(json, userEmail,
userPassword)));
  },
  _addNotification(event) {
    event.preventDefault();

```

```

    this._notificationSystem.addNotification({
      message: 'Notification message',
      level: 'success'
    });
  },
  componentDidMount() {
    this._notificationSystem = this.refs.notificationSystem;
  },
  shouldComponentUpdate(nextProps) {
    if (nextProps.user.authAnswer) {
      this._notificationSystem.addNotification({
        message: 'Авторизація пройшла успішно!',
        level: 'success'
      });
      setTimeout(() => browserHistory.push('/home'), 1200);
    } else {
      this._notificationSystem.addNotification({
        message:
`Авторизація не пройдена, спробуйте ще раз!`, level: 'error'
      });
    }
    return nextProps.user.authAnswer ? true : false;
  },
  render() {
    return (
      <div className={styles.inner}>
        <NotificationSystem ref="notificationSystem" />
        <div>
          <input ref="userEmail"
            className={styles.input}
            type="text"

```

```

        placeholder="Email"
        tabindex="0"
        autoFocus/>
      <input ref="userPassword"
        className={styles.inputPassword}
        type="password"
        placeholder="Пароль"
        tabindex="1"/>
      <bButton      onClick={this.auth}      text="Увійти"
tabindex="2"/>
    </div>
  </div>
)
}
});
const mapStateToProps = (state) => {
  return {
    user: state.user
  };
};
export default connect(mapStateToProps)(Auth);

```

ДОДАТОК Б
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА
РЕАЛІЗАЦІЯ МЕТОДІВ ОЦІНЮВАННЯ
РИЗИКІВ В БІОІНЖЕНЕРІЇ

Виконав: студент 2 курсу, групи КНТ-219м

Дубецький А.С.

Керівник: Проф. Кафедри ПЗ

Табунщик Г.В.

Рисунок Б.1 – Титульний слайд презентації

ПОСТАНОВКА ЗАВДАННЯ

Досліджено методи оцінювання ризиків медичних систем, на підставі чого розроблений програмний продукт, який дозволяє автоматизувати та значно прискорити проведення процедури оцінювання ризиків медичних систем та їх процесів з використанням групового експертного оцінювання.

Об'єкт дослідження – ризики в біоінженерії.

Предмет дослідження – методи оцінювання ризиків в біоінженерії.

Метою даної роботи є дослідження методів оцінювання ризиків медичних виробів, що дозволить створити програмну систему для автоматизації та спрощення проведення процедури оцінювання ризиків для медичних систем та їх процесів.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- а) дослідити етапи аналізу та оцінювання ризиків медичних систем та їх процесів;
- б) проаналізувати експертні методи оцінювання ризиків;
- в) зробити вибір мови програмування та середовища розробки;
- г) розробити архітектуру та реалізувати програмний продукт.

Рисунок Б.2 – Постановка завдання

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Медичні вироби належать до біоінженерних систем.

Для медичних систем та їх процесів першочергове значення має якість та безпека застосування.

Безпечним вважається медичний виріб, що використовується згідно передбаченого застосування та для якого відсутній неприпустимий ризик.

Ризик – поєднання вірогідності виникнення шкоди і ступеня тяжкості цієї шкоди.



Рисунок Б.3 – Дослідження предметної області

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Згідно ДСТУ EN ISO 14971:2015 виробники медичних виробів зобов'язані:

- ідентифікувати небезпеки (ризики), пов'язані з медичним виробом;
- визначати та оцінювати супутні ризики;
- управляти даними ризиками;
- проводити моніторинг результативності такого управління протягом усього життєвого циклу медичного виробу.



Рисунок Б.4 – Дослідження предметної області

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Таблиця 1 – Класифікація медичних виробів залежно від потенційного ризику застосування

Клас	Опис	Принцип застосування	Ступінь інвазивності	Час застосування
Клас I	Відсутні методичні ризики	Відсутній або не критичний контакт з поверхнею шкіри	Низька ступінь	< 60 хвилин
Клас IIa	Ризик застосування	Короткочасне застосування в організмі (очі, кишечник, хірургічні отвори)	Середня ступінь	< 30 днів
Клас IIb	Підвищений методичний ризик застосування	Системні порушення	Концепція не інвазивного управління	> 30 днів
Клас III	Високий методичний ризик	Вироби для прямого застосування на органах, центральній нервовій системі, системі кровообігу	Концепція інвазивного управління	Довготривалого застосування

Рисунок Б.5 – Класифікація медичних виробів

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

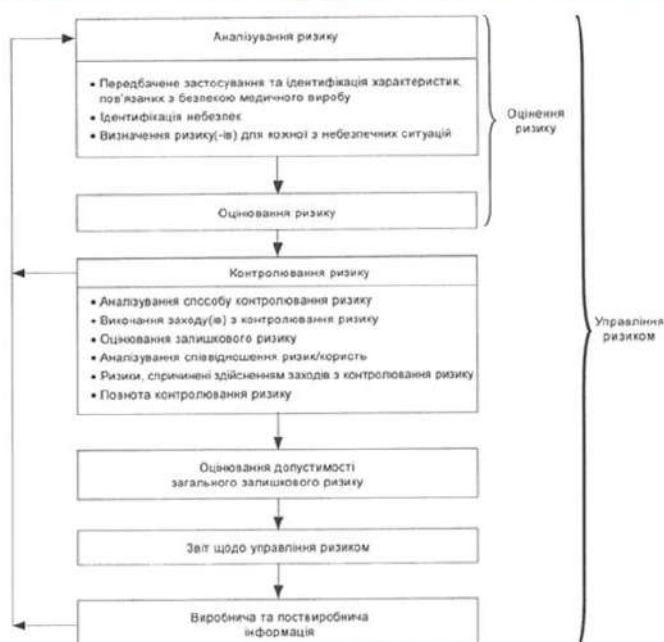


Рисунок 1 – Схематичне зображення процесу управління ризиком

Рисунок Б.6 – Схема процесу управління ризиком

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Таблиця 2 – Програмні продукти з оцінки ризику

Назва продукту Розробник	Переваги	Недоліки
vsRisk Vigilant Software	1) має якісні шкали оцінки, що налаштовуються користувачем; 2) є тріальна версія	1) складний інтерфейс; 2) великі списки загроз, вразливостей і контрзаходів, але без взаємозв'язків між ними; 3) відсутня кількісна оцінка ризиків; 4) помилки відображення кирилиці; 5) велика ціна
RA2 art of risk AEXIS Security	1) простий процесний підхід; 2) є налаштування під потреби конкретної організації	1) Слабкий інтерфейс; 2) примітивність засобів роботи з моделлю активів, загроз, вразливостей; 3) помилки відображення кирилиці в звітах
PTA PTA Technologies	1) цікавий механізм оцінки ефективності пропонованих контрзаходів; 2) є тріальна версія	не передбачена можливість якісної оцінки ризиків
Proteus Enterprise InfoGov	1) масштабується від одного користувача до багатьох; 2) є набір шаблонів опитувальників; 3) онлайн аудити у внутрішніх підрозділах і у зовнішніх постачальників	масштабність продукту направлена на керування бізнесом, де оцінка ризиків одна з процедур
RiskWatch RiskWatch International	1) простота використання; 2) поєднання кількісного та якісного аналізу; 3) широка база знань по загрозам, вразливостям та контрзаходам, з можливістю поповнення; 4) налаштування звітів; 5) формування листів опитування для отримання даних	достатньо висока ціна

Рисунок Б.7 – Програмні продукти з оцінки ризику

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Недоліки розглянутих програмних продуктів для проведення оцінки ризиків:

- складність використання продукту непідготовленому користувачу, інтуїтивно незрозумілий інтерфейс;
- неповне охоплення активів;
- ускладнення процесу усвідомлення ризику через автоматичний розрахунок ризиків, який скритий від користувача;
- проблеми відображення кирилиці для імпортних програмних продуктів;
- достатньо висока комерційна ціна;
- відсутність орієнтації на медичну інфраструктуру.

Рисунок Б.8 – Недоліки розглянутих програмних продуктів

АНАЛІЗ МЕТОДІВ ОЦІНКИ ТА УПРАВЛІННЯ РИЗИКАМИ

Аналіз ризиків умовно поділяють на два взаємодоповнюючі один одного види:

Якісний аналіз. Має на меті ідентифікувати фактори, області й види ризиків.

Кількісний аналіз ризиків повинен дати можливість чисельно визначити розміри окремих ризиків та ризику об'єкта в цілому.

Классификация методов экспертного оценивания

методы коллективного обсуждения, используемые для предварительного анализа проблем	• интервью, • анкетирование, • метод "мозгового штурма", • метод Дельфи и т.д.
методы формирования групп экспертов и определения компетентности экспертов	• метод самооценки, • метод взаимнооценки, • метод оценки по формальным показателям анкеты, • тестирование, • теория прецедентов и т.д.
методы оценки согласованности мнений экспертов	• методы нахождения коэффициентов конкордации, • коэффициент ранговой корреляции Спирмена и Кендалла и др.
методы формирования коллективной экспертной оценки	• методы вычисления средних (медиана, мода, среднееарифметическое, среднеегеометрическое и др.), • метод медианы Кемени, • вычисление среднего ранга и т.д.
методы оценивания альтернатив	• метод анализа иерархий, • методы ранжирования альтернатив, • классификация и др.

Рисунок Б.9 – Аналіз методів та управління ризиками

АНАЛІЗ МЕТОДІВ ОЦІНКИ ТА УПРАВЛІННЯ РИЗИКАМИ

Таблиця 3 – Ранги ризиків

Ранги	Опис рангів ризиків
I	Неприпустимий ризик
II	Небажаний ризик, який може бути допустимим тільки, якщо зниження ризику непрактично й веде до непропорційного збільшення витрат
III	Допустимий ризик, якщо витрати на зниження ризику не перевищують досягнутих поліпшень
IV	Нехтовно малий ризик

		Матриця оцінювання ризиків, що поділена на чотири зони за критеріями допустимості				
		Якісні рівні тяжкості				
		Критична	Значна	Помірна	Низька	Дуже низька
Якісні рівні вірогідності	Дуже висока	I	II	III	IV	IV
	Висока	I	II	III	IV	IV
	Низька	I	II	III	IV	IV
	Дуже низька	II	III	IV	IV	IV
	Малоймовірно	III	III	IV	IV	IV

Позначки:
■ – I ранг; ■ – II ранг; ■ – III ранг; ■ – IV ранг

Рисунок Б.10 – Аналіз методів та управління ризиками

АНАЛІЗ МЕТОДІВ ОЦІНКИ ТА УПРАВЛІННЯ РИЗИКАМИ

Коефіцієнт конкордації при наявності зв'язних рангів:

$$W = \frac{12 \cdot S_w}{p^2(n^3 - n) - p \sum_{j=1}^p T_j} \quad (1)$$

$$\text{де } S_w = \sum_{j=1}^n \left(\sum_{i=1}^p a_{ij} - \frac{p(n+1)}{2} \right)^2 \quad (2)$$

$$T_j = \sum_{i=1}^k (t_i^3 - t_i) \quad (3)$$

a_{ij} – ранг i -го об'єкту, який присвоїв йому j -й експерт;

p – кількість експертів;

n – кількість об'єктів (альтернатив), що оцінюються;

k – число зв'язаних (об'єднаних) рангів;

t_i – кількість елементів у i -й зв'язці для j -го експерта.

Рисунок Б.11 – Аналіз методів та управління ризиками

РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ



Рисунок 2 – Триланкова модель архітектури програми

Рисунок Б.12 – Розробка архітектури програми

РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

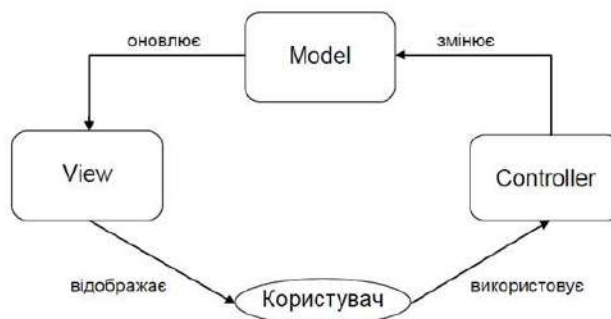


Рисунок 3 – Схема шаблону MVC

Рисунок Б.13 – Схема шаблону MVC

ІНСТРУМЕНТИ РОЗРОБКИ

Одна з основних вимог до програмного продукту – розробка мультиплатформного застосунку з використанням відкритих технологій. Тому кожен з рівнів обраної триланкової архітектури був реалізований з використанням різних технологій.

Технології для клієнтського рівня: мова програмування JavaScript, фреймворк для побудови вебзастосунків React.

Технології для серверного рівня та рівня бази даних було обрано Node.js та MySQL.

Рисунок Б.14 – Інструменти розробки

РОЗРОБКА ПРОГРАМИ

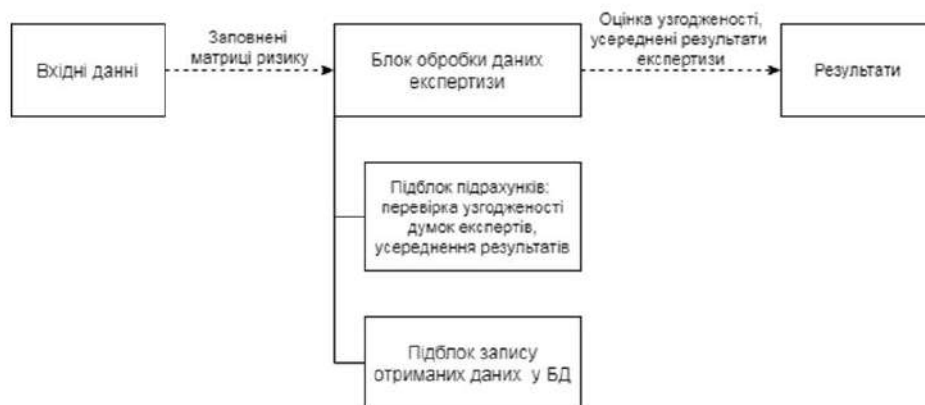


Рисунок 4 – Структурна схема розробленої системи

Рисунок Б.15 – Структурна схема розробленої системи

РОЗРОБКА ПРОГРАМИ

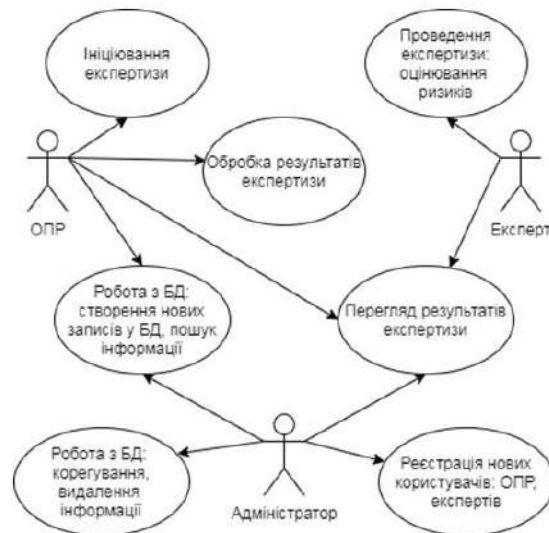


Рисунок 5 – Діаграма прецедентів системи

Рисунок Б.16 – Діаграма прецедентів системи

РОЗРОБКА ПРОГРАМИ

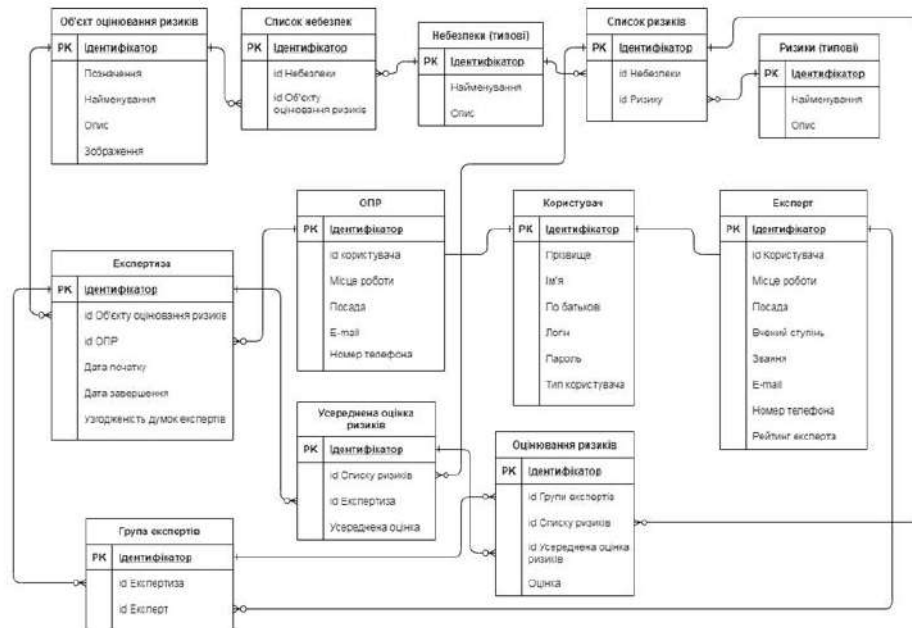


Рисунок 6 – Концептуальна модель бази даних

Рисунок Б.17 – Концептуальна модель бази даних

РОЗРОБКА ПРОГРАМИ



Рисунок 7 – «Головна» сторінка застосунку

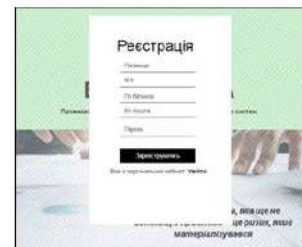


Рисунок 8 – Форма реєстрації

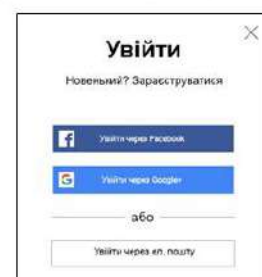


Рисунок 9 – Вікно входу в систему

Рисунок Б.18 – Вікна та форми вебзастосунку

РОЗРОБКА ПРОГРАМИ

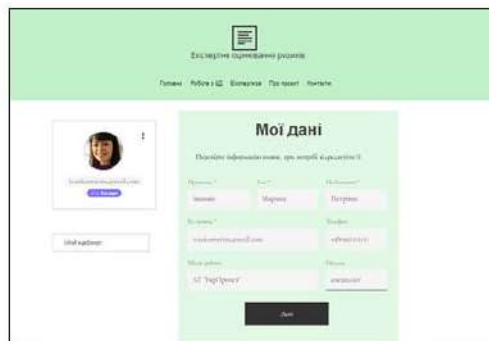


Рисунок 10 – Персональний кабінет користувача



Рисунок 11 – Сторінка застосунку «Про проєкт»

Рисунок Б.19 – Вікна вебзастосунку

РОЗРОБКА ПРОГРАМИ

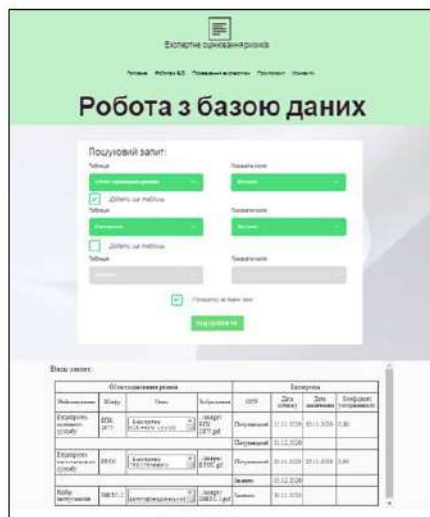


Рисунок 12 – Сторінка «Робота з БД» з формою запити



Рисунок 13 – Сторінка «Проведення експертизи»

Рисунок Б.20 – Вікна вебзастосунку

РОЗРОБКА ПРОГРАМИ

Оцінювання ризиків

Ендопротез колінного суглобу

Методика	Ризик	Оцінювання ризику риз
Ризикова оцінка операції	R ₁ - Збільшення кількості операцій	Високий
	R ₂ - Високий ступінь ризику до операції	Високий
	R ₃ - Ускладнення операції	Високий
	R ₄ - Застосування	Високий
Оцінювання складових	R ₅ - Кількість операцій	Високий
	R ₆ - Ускладнення операції	Високий
	R ₇ - Ускладнення операції	Високий
	R ₈ - Ускладнення операції	Високий
	R ₉ - Ускладнення операції	Високий
	R ₁₀ - Ускладнення операції	Високий
	R ₁₁ - Ускладнення операції	Високий
	R ₁₂ - Ускладнення операції	Високий
Результативна оцінка	R ₁₃ - Високий ступінь ризику	Високий
	R ₁₄ - Високий ступінь ризику	Високий

Відповісти далі

Рисунок 14 – Сторінка «Оцінювання ризиків»

Рисунок Б.21 – Вікно вебзастосунку

ВИСНОВКИ

- 1) Було досліджено процес управління ризиками медичних систем.
- 2) Проаналізовані методи, що використовуються на етапі оцінювання ризиків, зокрема експертні методи.
- 3) Зроблено огляд існуючих програмних засобів підтримки процесу управління ризиками, що показав актуальність і своєчасність проведення даної роботи.
- 4) Обґрунтовано для поставленого завдання використання вебтехнологій, побудованих на базі триланкової архітектури, та зроблений вибір мови програмування і середовища розробки.
- 5) Був розроблений вебзастосунок, який завдяки автоматизації дозволяє значно спростити та запобігти отриманню хибного результату при оцінюванні ризиків медичних виробів.
- 6) Виконані розрахунки економіко-організаційної частини диплому, які показали, що розробка даного програмного продукту є економічно доцільною.
- 7) Розроблена система може використовуватись для оцінювання ризиків при встановленні відповідності медичного виробу та при його сертифікації, що згідно законодавства України є обов'язковою процедурою для виробників.

Рисунок Б.22 – Висновки