МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут
комп'ютерних наук та управління проектами
(повне найменування інституту, назва факультету)

Кафедра програмного забезпечення автоматизованих систем
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної (магістерської) роботи

за темою "Математична модель для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel, та розробка програми для її реалізація"

Виконав: студент 6 курсу, групи 6151м
спеціальності
121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності)

_____Бєловол І.О._____
(підпис, прізвище та ініціали)
Керівник _____ Приходько С.Б.__
(підпис, прізвище та ініціали)
Рецензент _____
(підпис, прізвище та ініціали)
Завідувач кафедри _____Приходько С.Б.__
(підпис, прізвище та ініціали)

м. Миколаїв – 2020 р.

Міністерство освіти і науки України
Національний університет кораблебудування
імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Освітній ступень Магістр
Галузь 12 «Інформаційні технології»
(шифр і назва)
Спеціальність 121 «Інженерія програмного забезпечення»
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри_____
Приходько С.Б._____
" 26 " 10 2020 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ (МАГІСТЕРСЬКУ) РОБОТУ СТУДЕНТУ

Бєловол Ігор Олександрович
(прізвище, ім'я, по батькові)

1.Тема магістерської роботи
Математична модель для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel, та розробка програми для її реалізація
керівник роботи Приходько Сергій Борисович, доктор наук, професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від " 26 " жовтня 2020 року №1037-уч
2. Строк подання студентом роботи 01.12.2020 року
3. Вихідні дані до роботи

4.Зміст магістерської роботи (МР):
• Титульний аркуш, завдання на кваліфікаційну (магістерську) роботу, реферат (українською, англійською), зміст, перелік умовних позначень, символів, одиниць та термінів (за необхідності).
• Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень**.** Публікації.)
• Огляд літератури за темою, обґрунтування необхідності проведення досліджень за обраною темою, вибір напрямків досліджень, мета дослідження, основні задачі дослідження
• Викладення результатів власних досліджень з висвітленням того нового, що пропонується
• Проект програмного забезпечення
• Результати досліджень та розробки проекту програмного забезпечення
• Організаційно-економічний розділ
• Розділи з охорони праці та охорони навколишнього середовища
• Висновки
• Список використаних джерел
• Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

5. Перелік графічного матеріалу

_____

_____

_____

_____

_____

_____

6. Консультанти розділів магістерської роботи

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

7. Дата видачі завдання___

КАЛЕНДАРНИЙ ПЛАН

Назва етапів кваліфікаційної (магістерської) роботи (МР)	Термін виконання	Примітка
1. Підготовка вступної частини МР	18.10.2020	
2. Підготовка розділу (ів) МР з огляду літератури за темою, обґрунтування необхідності проведення досліджень за обраною темою, вибір напрямів досліджень	19.10.2020	
3. Підготовка розділу (ів) МР з результатів власних досліджень	22.10.2020	
4. Підготовка розділу МР з проекту програмного забезпечення	16.11.2020	
5. Підготовка організаційно-економічного розділу	18.11.2020	
6. Підготовка розділу з охорони праці	20.11.2020	
7. Підготовка розділу з охорони навколишнього середовища	23.11.2020	
8. Підготовка розділу МР – Висновки	25.11.2020	
9. Оформлення списку використаних джерел	27.11.2020	
10. Оформлення додатків	30.11.2020	
11. Підготовка презентації МР та доповіді	01.12.2020	
12. Подання МР на попередній захист	01.12.2020	
13. Подання МР рецензенту	11.12.2020	
14. Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, разом з заявою щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи	14.12.2020	
15. Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК від 19.05.2020 р. за №287-уч); презентації доповіді	18.12.2020	
16. Подання на кафедру ПЗАС письмового відгуку та рецензії на кваліфікаційну роботу	18.12.2020	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Бєловол Ігор Олександрович

«Математична модель для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel, та розробка програми для її реалізація»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2020 р.

Обсяг роботи: 85 стор., 4 табл., 13 рис., 19 використаних джерел, 5 додатків.

Актуальність теми роботи: визначається необхідністю підвищення достовірності оцінювання розміру web застосунків , що створюються з використанням фреймворку Laravel для системи TaskRing та розробка відповідного модуля для цієї системи.

Метою роботи є підвищення достовірності оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel.

Об'єкт дослідження: математична модель для оцінювання розміру веб застосунків.

Предмет дослідження: математична модель для оцінювання розміру веб застосунків що створюються з використанням фреймворку Laravel на основі даних компанії «Micro Logic Company».

Методи дослідження: методи теорії ймовірності та математичної статистики, регресійного аналізу, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів: полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру web застосунків які розроблені з використанням фреймворку Laravel за рахунок використання одновимірного нормалізуючого перетворення у вигляді десяткового логарифму, що дозволило підвищити достовірність оцінювання розміру web застосунків, які розроблені з використанням фреймворку Laravel, в порівнянні з існуючою моделлю.

Практичне значення одержаних результатів: розробка модуля для системи «TaskRing» компанії «Micro Logic Company», для оцінювання розміру веб застосунків що створюються з використанням фреймворку Laravel з використанням удосконаленої регресійної моделі.

Ключові слова: регресійна модель, нелінійна регресія, нормалізуюче перетворення, термін виконання задач, гнучка методологія.

ABSTRACT

Bielovol Ihor

«Mathematical model for estimating the size of web applications created using the Laravel framework and developing the software for its implementation»

Master's work for the educational level of the master's degree in the specialty 121 – «Software engineering». Admiral Makarov National University of Shipbuilding. Mykolaiv, 2020.

Volume of work: 85 pages, 4 tables, 13 figures, 19 used sources, 5 appendices.

Relevance of the theme: determined by the need to increase the reliability of estimating the size of web applications created using the Laravel framework for the system TaskRing and the development of an appropriate module for this system.

The purpose and objectives of the study: is to increase the reliability of estimating the size of web applications created using the Laravel framework.

Object of study: mathematical model for estimating the size of web applications.

Subject of study: mathematical model for estimating the size of web applications created using the Laravel framework based on data from the company " Micro Logic Company ".

Research methods: methods of probability theory and mathematical statistics, regression analysis, object-oriented programming.

Scientific novelty of the obtained results: is to improve the three-factor nonlinear regression model for estimating the size of web applications developed using the Laravel framework through the use of one-dimensional normalizing transformation in the form of a decimal logarithm, which increased the reliability of estimating the size of web applications developed using the Laravel framework, compared to the existing model.

The practical significance of the obtained results: development of a module for the "TaskRing" system of the company "Micro Logic Company", for estimating the size of web applications created using the Laravel framework using an advanced regression model.

Keywords: regression model, nonlinear regression, normalizing transformation, task time, flexible methodology.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8

ВСТУП..... 9

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL 11

1.1 Аналіз особливостей розробки web застосунків з використанням фреймворку Laravel.....11

1.2 Аналіз існуючих моделей оцінювання розміру web застосунків 12

1.3 Удосконалення регресійної моделі оцінювання розміру web застосунків 19

2 УДОСКОНАЛЕННЯ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL 20

2.1 Регресійна модель для оцінювання розміру web застосунків..... 20

2.2 Оцінювання адекватності регресійної моделі 21

2.3 Перевірка вихідних емпіричних даних на викиди..... 22

2.4 Побудова удосконаленої трьохфакторної нелінійної регресійної моделі для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel..... 23

2.5 Постановка задачі на розробку програмного забезпечення для реалізації математичної моделі для оцінювання розміру web застосунків..... 27

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL 28

3.1 Ескізний проект 28

3.1.1 Діаграма варіантів використання 28

3.1.2 Проектування інтерфейсу користувача 29

3.1.3 Концептуальна модель бази даних 31

3.2 Технічний проект 31

3.3 Робочій проект 35

3.3.1 Вибір мови та середовища розробки програмного забезпечення 35

3.3.2 Кодування та випробування ПЗ 35

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL 37

4.1 Опис ПЗ 37

4.2 Розрахунок витрат на створення й експлуатацію ПЗ для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel на основі нормалізуючих перетворень 38

4.3 Економічна ефективність розробки та впровадження ПЗ 41

4.4 Висновки 42

5 ОХОРОНА ПРАЦІ 43

5.1 Визначення, цілі, завдання та актуальність питань, пов'язаних з охороною праці 43

5.2 Аналіз небезпечних і шкідливих факторів, що впливають на людину при роботі з персональним комп’ютером 44

5.3 Розрахунок системи штучного освітлення приміщення для роботи за комп’ютером 49

5.4 Заходи щодо запобігання шкідливих факторів..... 51

6 ОХОРОНА НАВКОЛИЩНЬОГО СЕРЕДОВИЩА 58

6.1 Вплив людини на навколишнє середовище..... 58

6.2 Утилізація відходів комп’ютерної техніки 60

ВИСНОВКИ..... 64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65

Додаток А Технічне завдання на розробку програмного забезпечення для оцінювання розміру web застосунків, що розробленні з використанням фреймворку Laravel..... 68

Додаток Б Текст програми..... 73

Додаток В Опис програми..... 80

Додаток Г Інструація користувача..... 81

Додаток Д Програма і методика випробувань..... 84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД	База даних
ВВ	Випадкова величина
ВЗ	Веб-застосунок
ЕД	Емпіричні дані
МНК	Метод найменших квадратів
ПЗ	Програмне забезпечення
CMM	Capability Maturity Model

ВСТУП

Актуальність теми. Веб-застосунки стали всюдисущими, дозволяючи кожному, навіть маючи лише базові IT-знання, відкрити Інтернет-бізнес, використовуючи безкоштовний (відкритий код) або комерційний ВЗ. Є три основні фактори, які призвели до великого значення сьогоднішніх показників ВЗ. По-перше, поширення широкосмугового Інтернет-з'єднання змінило очікування відвідувачів та терпимість до очікування остаточної публікації. Другим фактором є збільшення обсягу роботи, що створюється постійно зростаючою кількістю користувачів Інтернету. Третім фактором є нова парадигма використання (створення вмісту користувача = ВЗ, що інтенсивно пише), висунута Web 2.0. Все це посилило тиск на ефективність веб-додатків. Це визначається необхідністю підвищення достовірності оцінювання розміру web застосунків з розробки програмного забезпечення для системи TaskRing та розробка відповідного модуля для цієї системи.

Метою роботи є підвищення достовірності оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1) Проаналізувати існуючі алгоритми та методи оцінювання розміру ВЗ, сформулювати постановку задачі;

2) Сформулювати удосконалення математичної моделі оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel;

3) Розробити модуль для програмного забезпечення для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel на основі даних компанії «Micro Logic Company» про виконання задач, використовуючи удосконалену математичну модель оцінювання розміру ВЗ, що створюються з використанням фреймворку Laravel.

Об’єктом дослідження є процес оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel.

Предметом дослідження є нелінійна регресійна модель оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel.

Методи дослідження. Для вирішення поставлених задач були застосовані методи теорії ймовірностей та математичної статистики, регресійного аналізу, побудови нелінійних регресійних моделей на основі нормалізуючих перетворень.

Наукова новизна одержаних результатів полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру web застосунків які розроблені з використанням фреймворку Laravel за рахунок використання одновимірного нормалізуючого перетворення у вигляді десяткового логарифму, що дозволило підвищити достовірність оцінювання розміру веб застосунків що створюються з використанням фреймворку Laravel в порівнянні з існуючою моделлю.

Практичне значення отриманих результатів полягає в розробці ПЗ для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel на основі нелінійної регресійної моделі та за рахунок використання трьохфакторної нелінійної регресійної.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL

1.1 Аналіз особливостей розробки web застосунків з використанням фреймворку Laravel

Laravel - це фреймворк для web-додатків з виразним і елегантним синтаксисом. Він дозволить спростити вирішення основних наболілих завдань, таких як аутентифікація, маршрутизація, сесії і кешування. Laravel - це спроба об'єднати все найкраще, що є в інших PHP фреймворк. Основні переваги Laravel:

1) велика екосистема з миттєвим розгортанням своєї платформи. Офіційний сайт надає безліч мінлива і інформації для ознайомлення;

2) документація Laravel близька до досконалості;

3) у Laravel є свій движок для шаблонів Blade, «гарний» синтаксис мови, який сприяє вирішенню всіх необхідних завдань, таких як аутентифікація, сесії, кешування і маршрутизація RESTful.

Незважаючи на свою молодість (перший реліз вийшов в 2011 році), це вже зовсім зрілий продукт, і, згідно з опитуванням, проведеним порталом SitePoint, він посідає перше місце за популярністю серед розробників на PHP.

Можливості Laravel

Пакети - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткові можливості вже доступні у вигляді таких модулів.

Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent.

Логіка програми - частина розробляється, оголошена або за допомогою контролерів, або маршрутів.

Зворотній маршрутизація пов'язує між собою генеруються додатком посилання і маршрути, дозволяючи змінювати останні з автоматичним оновленням пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL.

REST-контролери - додатковий шар для поділу логіки обробки GET- і POST-запитів HTTP.

Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключати файли їх визначень в include. Завантаження на вимогу запобігає завантаження непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються.

Укладачі уявлень - блоки коду, які виконуються при генерації уявлення (шаблону).

Інверсія управління - дозволяє отримувати екземпляри об'єктів за принципом зворотного управління. Також може використовуватися для створення і отримання об'єктів-одинаків.

Міграції - система управління версіями для баз даних. Дозволяє зв'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми.

Модульне тестування (юніт-тести) - грає дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання помилок.

Сторінковий висновок - спрощує генерацію сторінок, замінюючи різні способи вирішення цього завдання єдиним механізмом, вбудованим в Laravel [1].

1.2 Аналіз існуючих моделей оцінювання розміру web застосунків

Було опубліковано ряд статей щодо несхвалення модернізації традиційних методів оцінки зусиль для сучасного веб-програмного забезпечення. У цьому розділі розглядаються три публікації, які краще виконують вирішену проблему.

Вибір був заснований на трьох критеріях: стаття була цитована на декількох рецензованих публікаціях, вона побудована на визнаному стандарті, і охоплює характеристики веб-додатків.

Створення на визнаному стандарті має першорядне значення для забезпечення якості та надійності. І Рейфер WEBMO, і Abrahão та ін. Системи вимірювання OOmFPWeb є розширенням точок функції Albrecht. Точка функції - це "одиниця вимірювання", яка використовується для обчислення вимірювання функціонального розміру (FSM) програмного забезпечення. Function Point є одним з найбільш широко прийнятих методів і пропонує численні стандарти ISO / IEC. Найбільш широко використовуваним методом FSM є аналіз функціональної точки міжнародної групи користувачів (IFPUG), сертифікований як ISO/IEC 20926:2009 [2].

Останнім критерієм є те, що система вимірювання охоплює характеристики веб-додатків. Ці характеристики засновані на папері Reifer (2000), але все ще діють сьогодні, єдиним доповненням є крос-платформна категорія. Всі методи набрали однаково в 9 категоріях, в будь-якому місці всі методи провалили пункт крос-платформенності. Веб-об'єкти Reifer були створені на замовлення, щоб мати справу з цими характеристиками, тоді як Ruhe та ін. Web-COBRA та Abrahão та ін. Методи OOmFPWeb цитують роботу Reifer та розробляють свої публікації для вирішення тих самих проблем.

Таблиця 1.1 Характеристики розробки web застосувань [3]

	WEBMO	**Web-COBRA**	**OOmFPWeb**
Основною метою є вивести якісну продукцію на ринок якомога швидше	Так	Так	Так
Типовий розмір проекту невеликий (3-5 членів команди)	Так	Так	Так
Типова шкала часу становить 3-6 місяців	Так	Так	Так
Підхід до розвитку, що використовується, є швидким розробка додатків, склеювання будівельних блоків разом, простатипування	Так	Так	Так
Основними інженерними технологіями, що використовуються, є методи, засновані на компонентах, мови 4-го та 5-го покоління (html, Java тощо) візуалізація (рух, анімація) тощо.	Так	Так	Так
Процеси, що використовуються, є	Так	Так	Так
Продукти, розроблені об'єктними системами, багато компонентів для повторного використання (кошики тощо), мало зовнішніх інтерфейсів, відносно прості	Так	Так	Так
Люди, що беруть участь графічні дизайнери, менше досвідчені інженери програмного забезпечення	Так	Так	Так
Кросплатформенний. Візьміть до уваги контекст і пристрій, де ця програма використовується	Ні	Ні	Ні

Розглянутий метод: WEBMO

Система вимірювання WEBMO була розроблена Дональдом Дж. WEBMO покращує раніше використовувану алгоритмічну модель програмного забезпечення під назвою Constructive Cost Model (COCOMO II) та вводить новий показник під назвою Веб-об'єкти. Модель COCOMO II використовує вихідну лінію коду (SLOC) як показник розміру підкреслення, тоді як веб-об'єкти обчислюють розмір з урахуванням кожного елемента, який складають веб-додаток. Веб-об'єкти - це модифікація підрахунку точок функцій, розроблених для інтрамереж та інтернет-середовищ, це комбінує п'ять традиційних предикторів функціональної точки (внутрішні логічні файли, зовнішні файли

інтерфейсу, зовнішні входи, зовнішні виходи, зовнішні запити) плюс чотири веб-специфічні предиктори (кількість рядків XML, HTML та запитів, кількість мультимедійних файлів, кількість сценаріїв, кількість веб-будівельних блоків) в один розмір.

Модель вартості WEBMO не вимагає точних знань про драйвери витрат під час розробки, замість цього загальні драйвери витрат можуть бути визначені для того, щоб оцінити на ранній фазі розробки. Оскільки проекти прогресують і впливові функції дозрівають, вони потім краще визначаються в моделі витрат. Основними перевагами цієї моделі є використання математичної формули прогнозування зусиль та механізм, що дозволяє додавання нових типів операндів і операторів. Математична формула WebMO визначається таким чином:

$$\mathbf{Effort} = \mathbf{A} \prod_{i=1}^{8} \mathbf{cd_i} \ (\mathbf{Size})^{\mathbf{P1}} \quad ; \tag{1.1}$$

$$\mathbf{Duration} = \mathbf{B}(\mathbf{Effort})^{\mathbf{P2}} \quad , \tag{1.2}$$

де A та B константи, P1 та P2 степенні функції, cd_i – фактор затрат, Size web об’єкт [4].

WEBMO - це хороша спроба передбачити зусилля та вартість веб-додатків, так чи не так, коли було вперше задумано, екосистема споживчих пристроїв була дуже обмежена. Ще в 2000 році веб-додатки були обмежені технологіями та підключенням до Інтернету. У період з 2000 по 2015 рік глобальне проникнення в Інтернет зросло в 7 разів з 6,5% до 43%, а мобільний широкосмуговий доступ збільшився в 12 разів з 2007 року [4]. Веб-об'єкти, підкреслена метрика WEBMO, обмежується технологією впровадження та тим, що вона не може бути застосована на ранніх стадіях життєвого циклу розробки веб-додатків.

Модель Reifer продемонструвала переваги при застосуванні в некомплексні веб-гіпермедіа-системи, так чи не так, ці системи легше передбачити, ніж веб-додатки, тому що веб-гіпермедіа-системи, як правило, набагато менші з точки зору витрачених зусиль з розробки, і користувач не має

можливості впливати на стан системи на веб-сервері. Вимоги до веб-додатків сьогодні виглядають виключно по-різному і m рудний комплекс, ніж 15 років тому, коли настільний ПК був сольною платформою. Webmo датована формула не пропонує чіткого механізму додавання мультиплатформних операндів або обробки складності веб-додатків.

Розглянутий метод: Web-COBRA

Web-COBRA - це адаптація до веб-домену методу COBRA (оцінка COst, бенчмаркінг і аналіз ризиків), розробленого L. C. Briand, K. El Emam, F. Bomarius в 1998 році. Для створення моделі COBRA метод поєднує в собі знання експертів і минулі дані проекту, зібрані контрольованим способом, а потім використовувати цю інформацію, щоб налаштувати оцінки, що надходять від моделі, яка використовує міру розміру як драйвер вартості [5].

Для того, щоб отримати модель COBRA, людині, яка розраховує зусилля, потрібна інформація про три фактори проекту. По-перше, характеристики проекту, такі як тип проекту, домен програми тощо. По-друге, розмір міри розраховується послідовно серед усіх проектів І по-третє, драйвери витрат, що описують ресурси, як очікується, вплинуть на зусилля, спрямовані на розвиток. Виходячи з цих трьох факторів з випадкової моделі, вимірювання вартості витягується для поточної розробки проекту. Математична формула Web-COBRA визначається для оцінки величини відносної похибки MRE у відсотках від фактичних зусиль проекту [4]

$$\mathrm{MRE} = \left| \frac{Effort_{ACTUAL} - Effort_{ESTIMATED}}{Effort_{ACTUAL}} \right|. \tag{1.3}$$

Крім того, є рівень прогнозування Pred [4]. Цей показник є часткою спостережень за заданим рівнем точності:

$$\mathrm{Pred}\,(l) = \frac{k}{N}. \tag{1.4}$$

Найбільшими перевагами Web-COBRA є використання методу COBRA для веб-домену шляхом об'єднання експертних знань з даними невеликої

кількості проектів для розробки моделей оцінки вартості. На жаль, емпіричні дані про ефективність Web-COBRA були оцінені лише на обмежений набір даних з 12 веб-додатків. Ruhe та ін. Модель Web-COBRA покращилася на WEBMO Reifer , зосередивши свої емпіричні дослідження на більш складних веб-додатках. Він також використовує переваги веб-об'єктів Reifer , щоб передбачити зусилля з розробки проекту за допомогою складеного методу оцінки, який використовує експертні знання та обмежену кількість минулих даних проекту. У всякому чи інакше Web-COBRA також не враховує вимоги до мультиплатформи, оскільки в 2003 році при розробленні моделі мобільні та інші пристрої IoT були навряд чи доступні на споживчому ринку.

Розглянутий метод: OOmFPWeb

Abrahão та ін. Модель вимірювання OOmFPWeb є останнім методом з 3 розглянутих, він був опублікований в 2004 році і був заснований на лабораторному експерименті, який оцінює OO- Method Function Points for The Web. Метод був розроблений відповідно до правил підрахунку IFPUG (International Function Point Users Group) для FPA (IFPUG, 1999) і призначений для використання в контексті веб-систем. OOmFPWeb забезпечує міру функціонального розміру для веб-додатків, які виробляються з використанням об'єктно-орієнтованих веб-рішень (OOWS) (Pastor et al., 2001), цей метод інтегрує навігаційний та презентаційний дизайн із класичним підходом концептуального моделювання OO. OOmFPWeb працює, генеруючи значення на основі кількості функціональних можливостей, які веб-система надає користувачеві. Метод оцінює функціональність веб-систем на основі визначених користувачем функцій, які інкапсульовані в концептуальній моделі проектів веб-додатків. Це прорив від моделі WEBMO Reifer 2000, яка базується виключно на артефактах впровадження, створених після того, як додаток буде повністю розроблений [6].

Емпірична оцінка OOmFPWeb походить від дослідження 2009 року, проведеного S. Abrahão і G. Poels, аналізи на основі чотирьох наборів даних з сімейств експериментів, проведених в Іспанії, Аргентині та Австрії, демонструє

ефективність методу в порівнянні з існуючими галузевими практиками. Метод також був сприйнятий як простий у використанні і корисний учасниками, навіть якщо OOmFPWeb, як вважають, споживає більше часу, ніж встановлений функціональний аналіз точки (FPA). На жаль, оскільки OOmFPWeb є новою процедурою, яка ще не широко використовується на практиці, автори провели свої дослідження на штучній лабораторній установці з використанням студентських учасників, а не практиків.

Вимірювання функціонального розміру OOmFPWeb вимагає двох кроків абстракції, які називаються етапом ідентифікації та кроком вимірювання. На кроці ідентифікації концептуальна схема побудована на основі визначених специфікацій вимог користувача (дані, процес, поведінка, навігація та презентація). На кроці вимірювання всі відповідні ідентифіковані елементи веб-програми визначаються як типи базового функціонального компонента (BFC), як і в стандарті ISO/IEC для вимірювання функціонального розміру. Далі функціональний розмір кожного ідентифікованого БФК оцінюється рейтингом їхнього виходу та перекладом цього рейтингу на значення "Точки функцій". Нарешті, функціональні значення розміру БФК підсумовуються, щоб отримати значення функціонального розміру веб-застосунку.

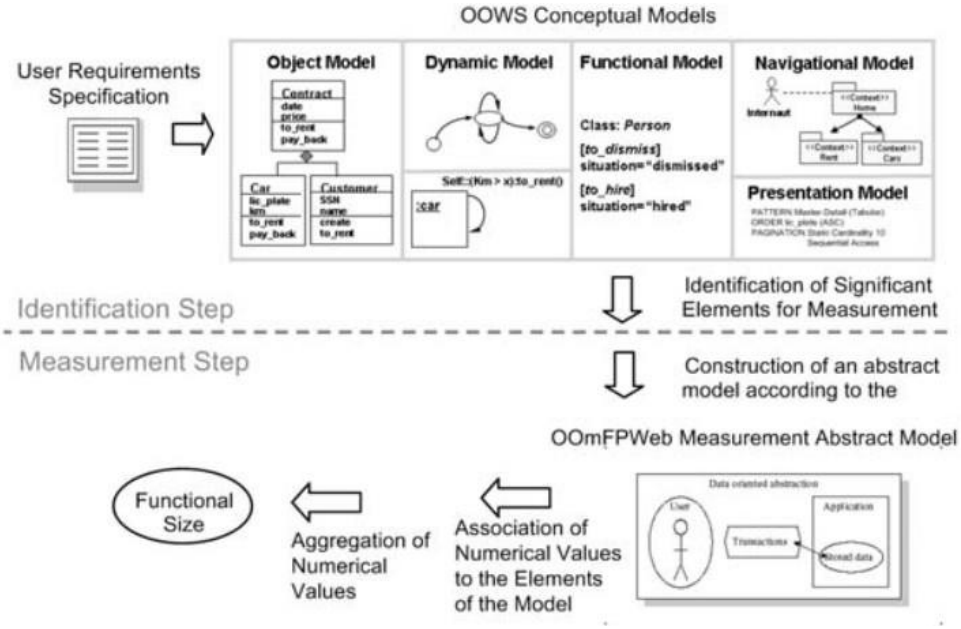


Рисунок 1.1 – Процедура вимірювання OOmFPWeb [3]

OOmFPWeb зусилля вимірювання представлення пропонує більш візуальні сигнали і крок за кроком процедури в порівнянні з WEBMO або WEB-COBRA, його п'ять концептуальних моделей інкапсулювати кожен домен вимоги в вимірювану одиницю функціональних точок значення t капелюх полегшити підрахунок одиниць зусиль, представлено на рисунку 1.4 . У всякому порядку OOmFPWeb також не в змозі описати мультиплатформні вимоги. Метод Моделі презентації описує лише взаємодію користувача з веб-інтерфейсом програми, але він припускає, що контекст і пристрій незмінні. Сучасні веб-додатки побудовані з використанням адаптивного інтерфейсу, були інтерфейс адаптується до контексту взаємодії (біг, сидіння, водіння, так далі) і характеристик платформи (телефон, Smart TV, Smart watch) [7-8].

1.3 Удосконалення регресійної моделі оцінювання розміру web застосунків

Для досягнення мети, що сформульована вище, ми скористалися наступними методами: побудова нелінійних регресійних моделей для оцінювання кількості строк ПЗ та застосування відповідних методів множинного нелінійного регресійного аналізу [9]. Згідно з цим спочатку виконується нормалізація багатовимірних негаусових даних за багатовимірним нормалізуючим перетворенням[10-11].

2 УДОСКОНАЛЕННЯ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL

2.1 Регресійна модель для оцінювання розміру web застосунків

У роботі для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel за рахунок використання трьохфакторної нелінійну регресійну модель яку побудовано із використанням одновимірного нормалізуючого перетворення у вигляді десяткового логарифму, що дозволило підвищити достовірність оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel в порівнянні з існуючою моделлю. Але побудова багатовимірної моделі на основі декількох метрик достатньо складна та затратна в плані зусиль та часу, необхідних для її реалізації. Вона може стати непридатною до використання через відсутність даних, ресурсів або експертних навичок в цій галузі. Саме тому в якості першого наближення для оцінювання кількості рядків коду веб-додатків, реалізованих мовою Java, буде доцільно побудувати нелінійну регресійну модель на основі двох метрик: кількості рядків коду та загальної кількості класів. Для нормалізації негаусівських ВВ можуть бути використані перетворення на основі десяткового або натурального логарифму, перетворення Box-Cox, перетворення Джонсона та інші [12].

У даній роботі буде використовуватись трьохфакторна нелінійна регресійна модель яку побудовано із використанням одновимірного нормалізуючого перетворення у вигляді десяткового логарифму тому, що воно точніше[13-18].

2.2 Оцінювання адекватності регресійної моделі

Для перевірки адекватності лінійного рівняння регресії використаємо коефіцієнт детермінації R^2:

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}\right), \quad (2.1)$$

де y_i – емпіричне значення y;

$\hat{y}_i$ – розрахункове значення y;

$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ – середнє значення ВВ y.

R^2 характеризує частку дисперсії, яка обумовлена регресією, в загальній дисперсії показника y. Коефіцієнт детермінації R^2 приймає значення від 0 до 1. Чим ближче значення коефіцієнта до 1, тим тісніше зв'язок результативної ознаки з досліджуваними факторами [15].

При значенні $R^2 \geq 0{,}5$ можна вважати, що дана модель є прийнятною. Достатньо ефективною та результативною можна вважати модель з показником детермінації $R^2 \geq 0{,}8$. Якщо ж $R^2 = 1$, то лінія регресії точно відповідає усім спостереженням та вимогам, а модель можна вважати адекватною та достовірною.

Величина коефіцієнта детермінації виступає важливим критерієм оцінки якості лінійних і нелінійних моделей. Чим вагоміша частка пояснюваної варіації, тим менше роль інших факторів, а отже, модель регресії краще апроксимує вихідні дані і такою регресійної моделлю можна скористатися для прогнозу значень результативного показника.

Величину відносної похибки MRE (Magnitude of Relative Errors) для лінійного та нелінійного рівнянь регресії можна знайти за формулою [15]:

$$MRE_i = \left|\frac{y_i - \hat{y}_i}{y_i}\right|, \quad (2.2)$$

де $\hat{y}$ – значення y, розраховане за рівнянням регресії;

y_i – фактичне значення ВВ y.

MMRE (Mean of Magnitude of Relative Errors – середня величина відносної похибки) можна розрахувати за формулою [15]:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i. \quad (2.3)$$

Таким чином, маємо ряд показників для перевірки адекватності регресійних моделей та порівняння їх між собою [18].

2.3 Перевірка вихідних емпіричних даних на викиди

Для знаходження викидів використовується відстань Махаланобіса [15]:

$$d_i^2 = (z_i - \bar{z})^T S_n^{-1} (z_i - \bar{z}), \quad (2.4)$$

де $\bar{z}$ – середній вектор вибірки;

S_N – матриця кореляції вибірки, яка визначається за формулою [15]:

$$S_N = \frac{1}{N}\sum_{i=1}^{N} (z_i - \bar{z})(z_i - \bar{z})^T. \quad (2.5)$$

Точка, що має найбільшу відстань Махаланобіса до решти заданих точок, вважається що має найбільшу значимість, так як вона має найбільший вплив на кривизну і на коефіцієнти рівняння регресії [15].

Розрахувавши d_i^2 можна використати або критерій Пірсона χ^2, або критерій Фишера F. При використанні критерію Фишера F потрібно додатково розрахувати T_S (Test statistic). T_S для d_i^2 може бути розрахований таким чином:

$$Ts = \frac{(N-m)N\,d_i^2}{(N^2-1)m}, \qquad (2.6)$$

який має апроксимований розподіл F з m та $N - m$ ступенями свободи [15].

У данній роботі будемо використовувати критерій Фишера F. Тестова статистика для квадрату відстані Махаланобіса порівнюється з квантилем розподілу F, який позначається як $F_{m,N-m,\alpha}$. Тут α – це рівень значущості, який у нашому випадку дорівнюватиме 0,05. Точки, для яких значення T_S, розраховане за формулою (2.6), буде більше, ніж квантиль розподілу F вважаються викидами, і ці значення видаляються з набору даних.

Після усунення викидів отриманий новий набір значень знову нормалізується, знаходимо квадрат відстані Махаланобіса за формулою (2.4) та тестова статистика за формулою (2.5) [15]. Процедура повторюється до того, поки всі значення T_S не будуть менше або дорівнюватимуть квантилю розподілу F .

2.4 Побудова удосконаленої трьохфакторної нелінійної регресійної моделі для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel

Враховуючи те, що Laravel є PHP фреймворком, дані з метрик Web-застосунків, які розміщені на сайті GitHub (https://github.com), були отримані за допомогою інструменту PhpMetrics (https://phpmetrics.org/). Як і в роботі [6], в подальшому ми використовували наступні метрики: кількість класів X_1; суму середньої кількості класів, на які впливає даний клас (Average Afferent Coupling), і середньої кількості класів, з яких даний клас отримує ефекти (Average Efferent Coupling), X_2, та середньої кількості методів X_3. Значення зазначених метрик для 30 Web-застосунків надані в табл. 2.1 Також в табл. 2.1 наведені: розмір Web-застосунків Y у тисячах рядків коду та квадрат відстані Махаланобіса MD^2.

Таблиця 2.1 – Вибірка даних та значення квадрату відстані Махаланобісу MD^2

№	Y	X_1	X_2	X_3	MD^2	№	Y	X_1	X_2	X_3	MD^2
1	20,816	171	9,053	2,485	3,76	16	4,069	107	1,879	2,607	1,05
2	9,482	180	3,100	4,644	3,54	17	7,241	194	3,108	3,010	1,71
3	57,454	1000	4,039	4,612	**24,18**	18	0,515	21	1,238	1,714	3,56
4	19,481	148	9,939	2,466	4,69	19	1,771	57	2,175	2,877	0,70
5	3,532	75	2,773	3,173	0,50	20	16,817	185	11,000	5,119	11,79
6	17,888	98	13,847	1,500	**15,52**	21	1,072	40	1,925	2,500	1,14
7	3,825	81	2,728	2,926	0,42	22	10,774	184	3,065	4,038	1,21
8	22,438	308	5,948	4,127	1,13	23	17,743	313	3,994	4,550	2,33
9	1,881	44	3,000	2,773	0,56	24	1,044	42	1,905	2,952	0,93
10	2,364	44	3,114	3,091	0,62	25	0,805	29	1,552	2,000	2,41
11	67,055	727	4,455	4,344	**21,63**	26	8,618	215	3,019	3,563	1,40
12	9,305	146	3,466	3,158	0,08	27	33,591	365	6,312	4,014	3,13
13	3,364	97	2,299	3,124	0,57	28	28,361	303	6,535	3,743	1,97
14	0,47	23	0,913	1,478	4,78	29	5,732	88	3,500	4,068	2,35
15	3,391	76	2,250	3,500	1,11	30	8,615	210	3,043	3,490	1,23

Розподіл даних з табл. 2.1 є негаусівським тому, що для трьох застосунків (3, 6 та 11) значення MD^2, які відповідно дорівнюють 24,18, 15,52 та 21,63, є більшими ніж величина квантіля розподілу χ^2, що становить 14,86 для рівня значущості 0,005 та чотирьох ступенів свободи. Про негаусівський розподіл даних з табл. 2.1 також свідчить оцінка багатовимірного ексцесу β_2, що дорівнює 52,09. Це значення більш ніж вдвічі перевищує теоретичне, що у нашому випадку дорівнює 24.

Метрики (майбутні фактори) X_1, X_2, та X_3 були перевірені на наявність мультиколінеарності. Наявність мультиколінеарності свідчить про те, що в множинній регресійній моделі два або більше факторів пов‘язані між собою або мають високий ступінь кореляції.

Наявність мультиколінеарності ми визначали за коефіцієнтами впливу дисперсії (VIFs) серед майбутніх факторів в моделі множинної лінійної регресії. Для лінійної моделі множинної регресії з *k*-факторами X_i, VIFs – це діагональні елементи оберненої коваріаційної матриці *k*-факторів. Значення VIFs більше за 10 часто сприймаються як сигнал, що дані мають проблеми з мультиколінеарністю. У разі, якщо значення VIFs знаходяться у межах від 1 до 5, то мультиколінеарності немає. Значення VIFs для X_1, X_2, та X_3 дорівнюють 1,56, 1,03 та 1,52 відповідно. Ці значення вказують на відсутність проблеми мультиколінеарності між обраними факторами.

Спочатку за даними табл. 2.1 було побудовано лінійну регресійну модель для оцінювання розміру Web-застосунків, що створені з використанням фреймворку Laravel, у вигляді

$$Y = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon, \qquad (2.7)$$

де оцінки параметрів за методом найменших квадратів такі: $\hat{b}_0 = -1{,}73631$, $\hat{b}_1 = 0{,}07090$, $\hat{b}_2 = 1{,}27302$, $\hat{b}_3 = -1{,}15451$. Причому в (2.7) ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N\left(0, \sigma_\varepsilon^2\right)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2 = 19{,}74$. Сума квадратів відхилень для моделі (2.7) склала 572,4. Нульова гіпотеза про нормальність розподілу ε моделі (2.7) була перевірена за критерієм Пірсона. Ця гіпотеза була відкинута для рівня значимості 0,05 тому, що значення χ^2, яке дорівнює 8,88, більше за 5,99 – критичне значення χ^2 для рівня значимості 0,05 та двох ступенів свободи. Те, що в (2.7) ε не є випадковою величиною з розподілом Гаусу вказує на відсутність теоретичного обґрунтування застосування лінійної моделі (2.7) для оцінювання розміру Web-застосунків, що створені з використанням фреймворку Laravel, та на необхідність побудови нелінійної регресійної моделі. Також на цю необхідність вказує негаусівський розподіл чотирьохвимірних даних з метрик (залежної та незалежних змінних) табл. 2.1.

Трьохфакторну нелінійну регресійну модель ми побудували із використанням одновимірного нормалізуючого перетворення у вигляді десяткового логарифму.

Для цього негаусівські дані з табл. 2.1 ми нормалізуємо за перетворенням у формі десяткового логарифму. Далі для нормалізованих даних будуємо лінійну регресійну модель

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \qquad (2.8)$$

де параметри моделі (2.14) оцінювалися методом найменших квадратів та їх оцінки є такими: $\hat{b}_0 = -1{,}74968$, $\hat{b}_1 = 1{,}07738$, $\hat{b}_2 = 0{,}76106$, $\hat{b}_3 = -0{,}19497$. Сума квадратів відхилень для моделі (2.14) складає 0,1430. В (2.14) ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N\left(0, \sigma_\varepsilon^2\right)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2$ =0,00493.

Після чого за (2.14) та перетворенням у вигляді десяткового логарифму будуємо нелінійну регресійну модель для оцінювання розміру Web-застосунків, що створені з використанням фреймворку Laravel,

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}. \qquad (2.9)$$

Побудована модель (2.9) була перевірена за множинним коефіцієнтом детермінації R^2, середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, PRED(0,25). Ці показники зазвичай використовуються для оцінювання якості прогнозування за допомогою регресійних моделей і в інженерії програмного забезпечення. Допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Допустиме значення R^2 приблизно таке ж, як для PRED(0,25).

Для моделі (2.9), що була побудована за даними з табл. 2.1, значення R^2, MMRE і PRED(0,25), які дорівнюють відповідно 0,946, 0,135 і 0,900, є задовільними. Причому значення цих показників для моделі (2.9) є кращими за значення R^2, MMRE і PRED(0,25) для лінійної регресійної моделі (2.7), які дорівнюють відповідно 0,926, 0,382 і 0,600. Сума квадратів відхилень для моделі (2.9) складає 404,9, що на 41% менше у порівнянні з відповідною сумою для лінійної регресійної моделі (2.7) [8,11,19-20].

2.5 Постановка задачі на розробку програмного забезпечення для реалізації математичної моделі для оцінювання розміру web застосунків

Виходячи з аналізу існуючих алгоритмів та методів для оцінювання розміру web застосунків, сформулюємо наступну постановку задачі: розробити програмне забезпечення (ПЗ) для оцінювання розміру web застосунків. Для оцінювання розміру web застосунків слід використовувати удосконалену математичну модель оцінювання розміру web застосунків.

ПЗ розробляється як модуль для системи TaskRing.

Користувачам програмного комплексу мають бути доступні функції, відповідно до прав доступу, які зазначені нижче.

Для користувача:

- створення web додатку – введення інформацію про web додаток для оцінки його розміру(LOC, Classes, AEC, Methods);
- перегляд списку web додатків;
- редагування web додатків;
- перегляд звіту оцінки розміру web додатків(точкові оцінки визначені за моделлю (2.7), та інтервальні оцінки за моделлю (2.9));
- обчислення інтервальних оцінок розміру web додатку;
- обчислення точкових оцінок розміру web додатку;
- перегляд точкових оцінок розміру web додатку;
- перегляд інтервальних оцінок розміру web додатку;
- авторизація.

Для адміністратора компанії:

- створення резервної копії БД компанії:
- Перегляд звіту всіх користувачів.

Повний перелік вимог представлено у вигляді технічного завдання у додатку А.

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL

3.1 Ескізний проект

Проектування будь-якого ПЗ розпочинається з розробки ескізного проекту, у якому представляються результати зовнішнього проектування програмного забезпечення.

3.1.1 Діаграма варіантів використання

Контекстна діаграма (рис. 3.1) – це діаграма, розташована на вершині деревовидної структури діаграм, що представляє собою загальний опис системи та її взаємодію з зовнішнім середовищем.

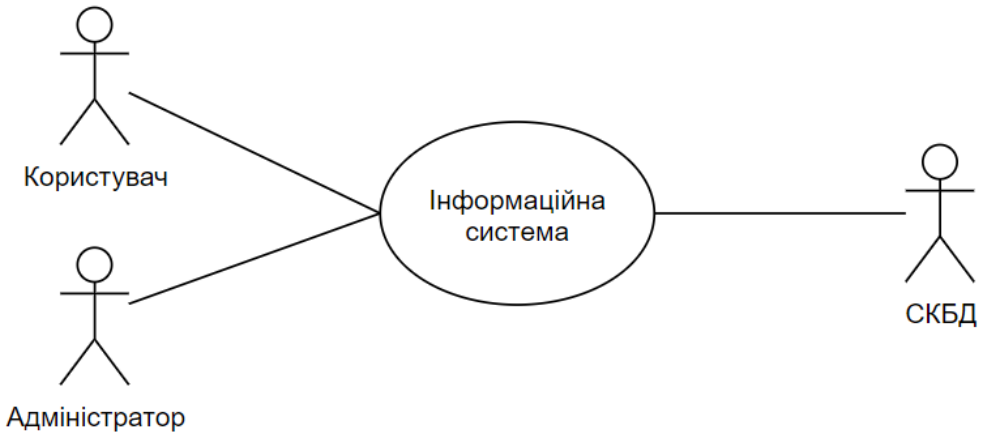


Рисунок 3.1 – Контекстна діаграма

Діаграма варіантів використання – діаграма, на якій зображено відношення між акторами та прецедентами в системі (рис. 3.2).

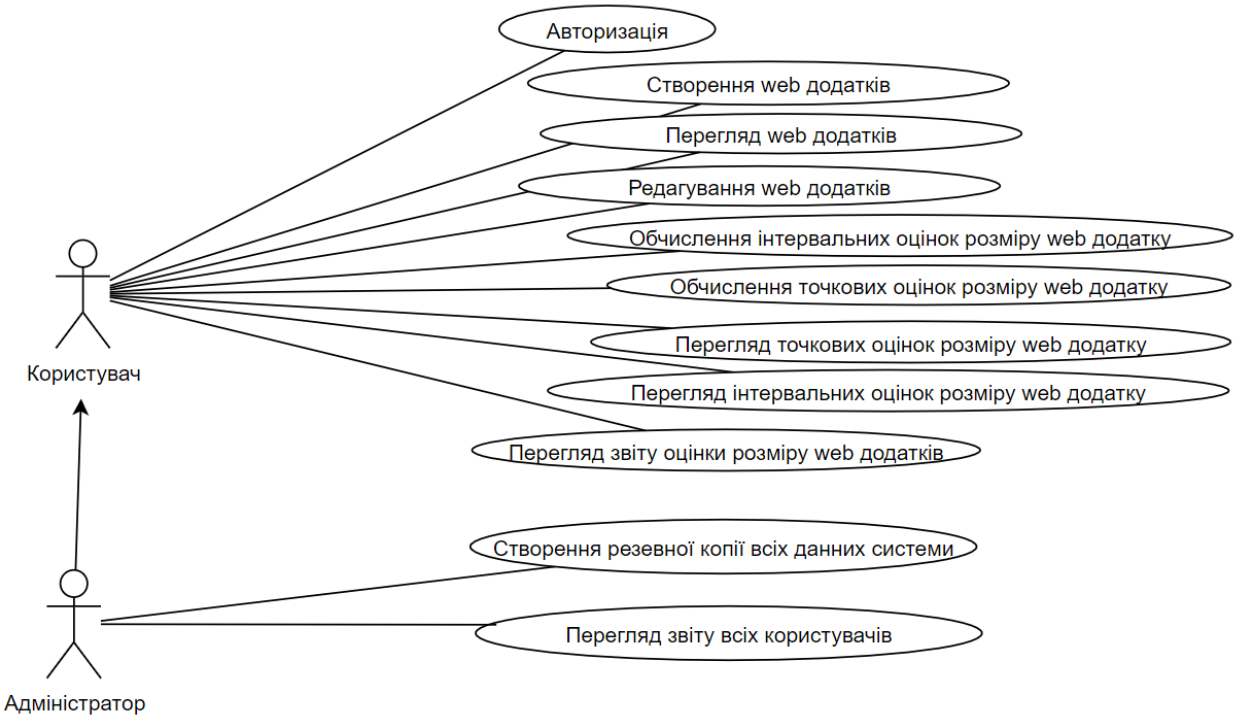


Рисунок 3.2 – Діаграма варіантів використання

Специфікації варіантів використання наведені у технічному завданні, Додаток А.

3.1.2 Проектування інтерфейсу користувача

Проектування інтерфейсу користувача, е важливим етапом розробки програмного забезпечення. Проектування інтерфейсу допомагає наглядно оцінити результат замовником, а також вдосконалитись в зручності інтерфейсу та перевірка виконуваних функцій. Приклади ескізів деяких екранних форма сайту, який буде використовувати розроблюваний веб-сайт, представлені на рисунках 3.3 - 3.4.

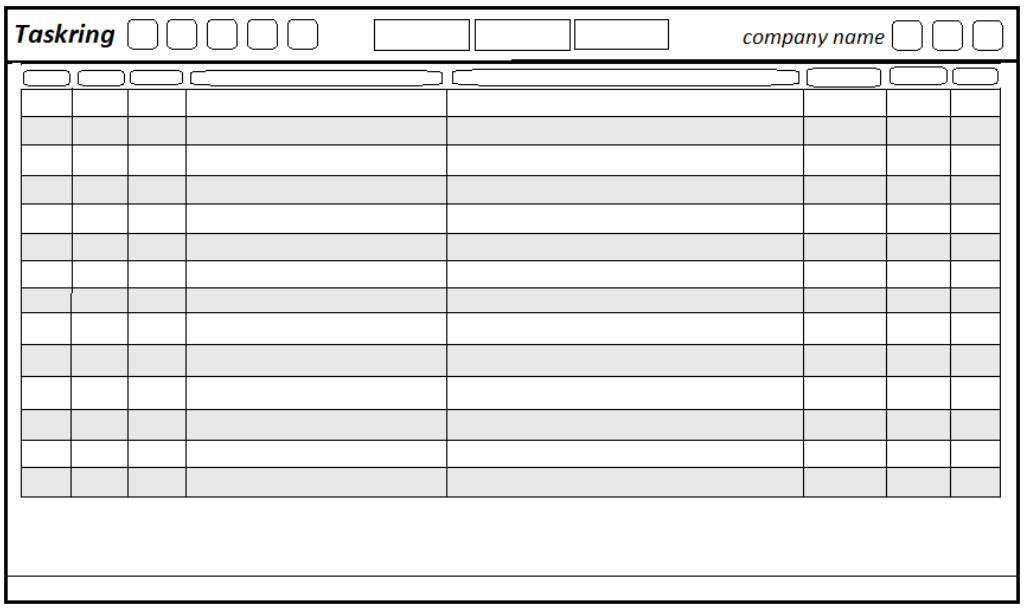


Рисунок 3.3 – Вікно перегляду списку web-застосунків

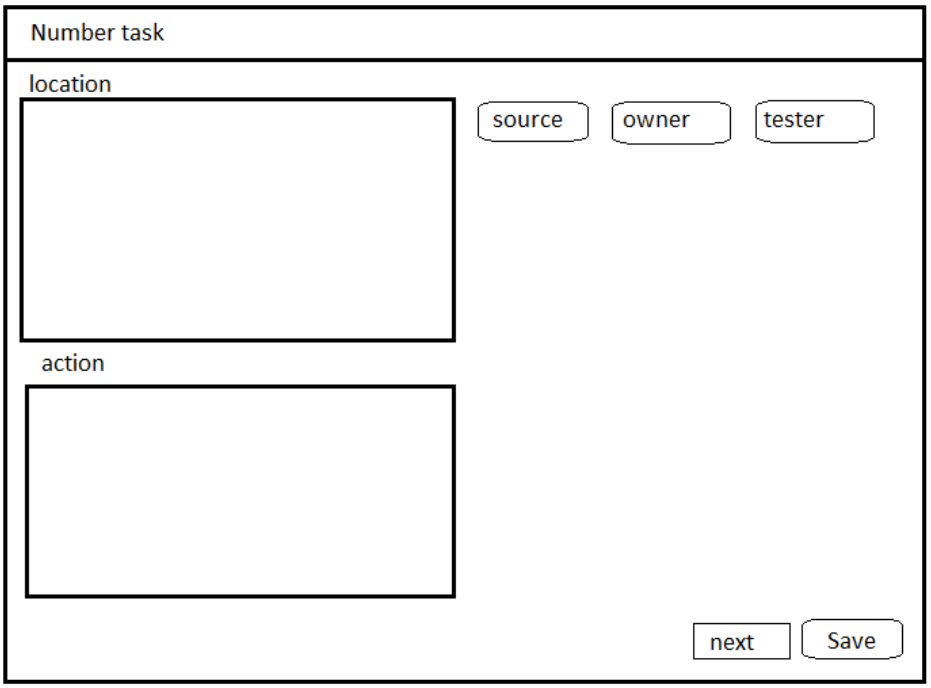


Рисунок 3.4 – Вікно редагування web-застосунків

На рисунках наведених вище зображений інтерфейс роботи з списком задач та окремою задачею.

3.1.3 Концептуальна модель бази даних

Концептуальне проектування – побудова семантичної моделі предметної області. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. Концептуальна модель (рис. 3.5) представлена за допомогою ER-діаграми.

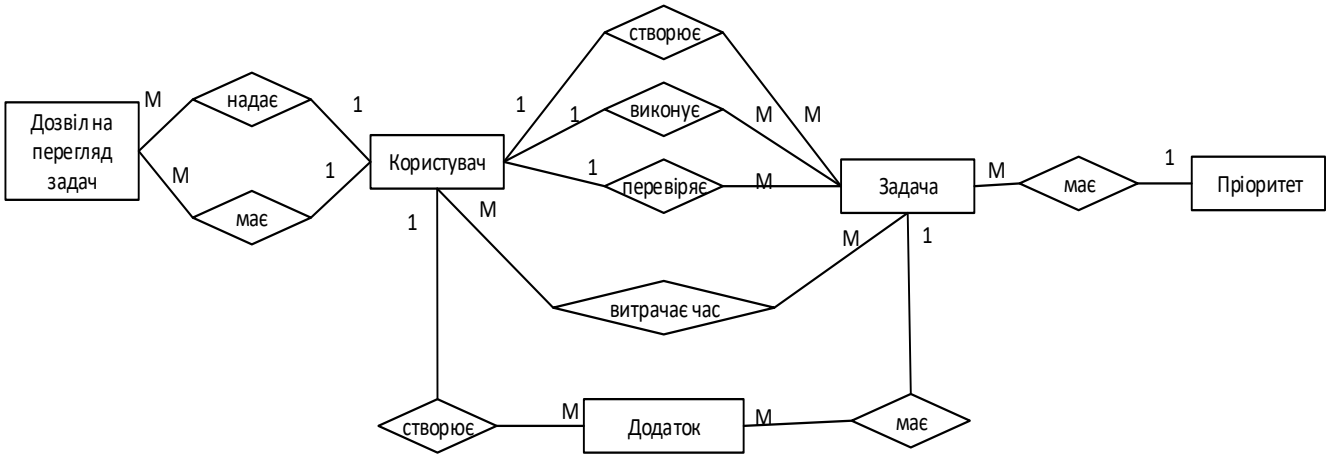


Рисунок 3.5 – Концептуальна модель бази даних

3.2 Технічний проект

Першим етапом технічного проектування є побудова діаграми комплненів системи. Діаграма компонентів необхідна для відображенні основних компонентів та відносин між ними. На рисунку 3.6 відображено діаграму компонентів для модулю розподілення вхідного запиту.

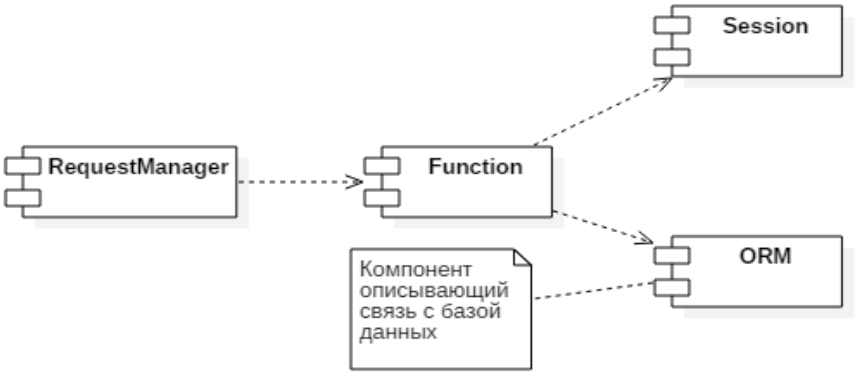


Рисунок 3.6 – Діаграма компонентів

Наступним етапом є уточнення діаграми компонентів набором діаграм класів. Діаграма класів необхідна для відображенні основних класів та типів відносин між ними. На рисунку 3.7 відображено діаграму класів для модулю розподілення вхідного запиту. На рисунках 3.8-3.9 зображено діаграмі класів додаткових модулів розроблюваного ПЗ.

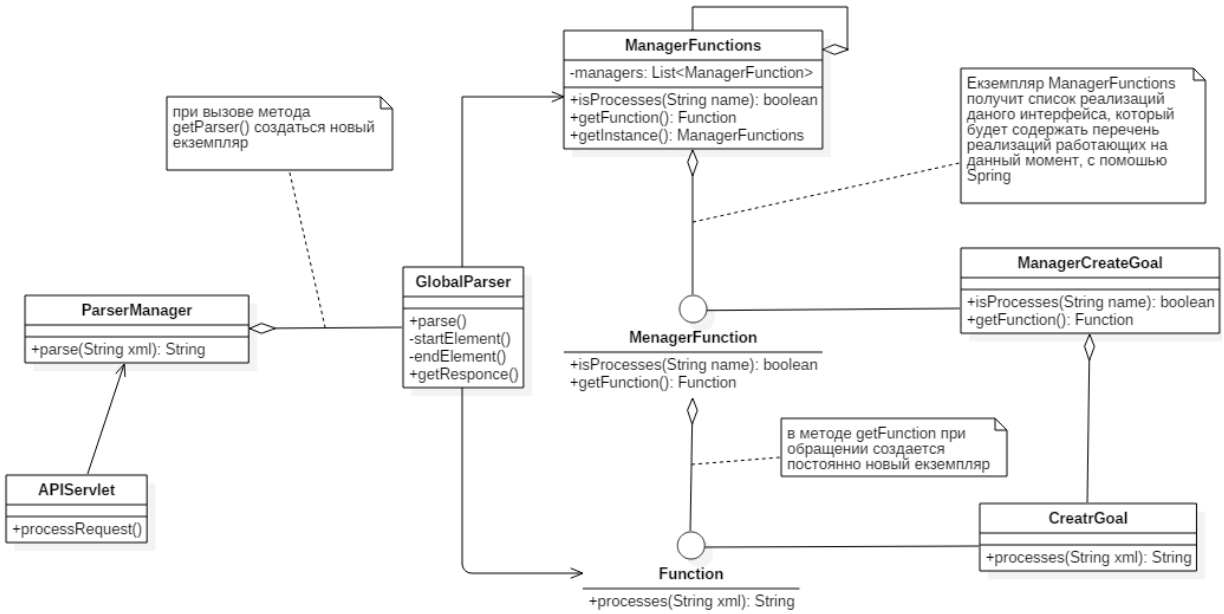


Рисунок 3.7 – Діаграма класів модулю розподілення вхідних запитів

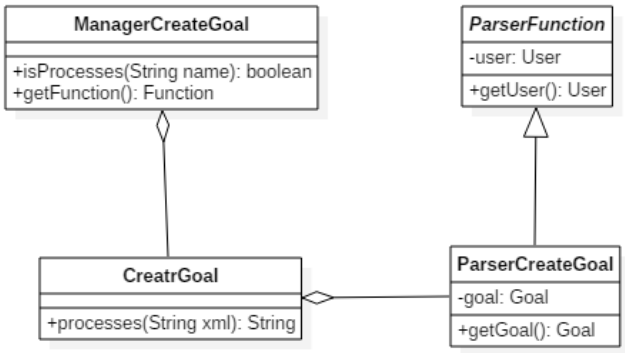


Рисунок 3.8 – Діаграма класів для модуля (функції) «Створення web-застосунка»

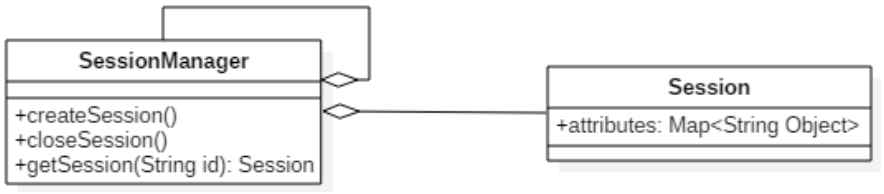


Рисунок 3.9 – Діаграма класів для модуля сесій.

На донному етапі проектування проводиться вибір типу сховища даних і перетворення концептуальної моделі в модель даних відповідно до вибраного типу сховища.

І з за відносної легкості в проектуванні, обсягом інформації, та можливостями, що надають реляційні БД. Було вибрано саму цей тип сховища даних. Відповідно до якого була розроблена логічна модель (рис. 3.10).

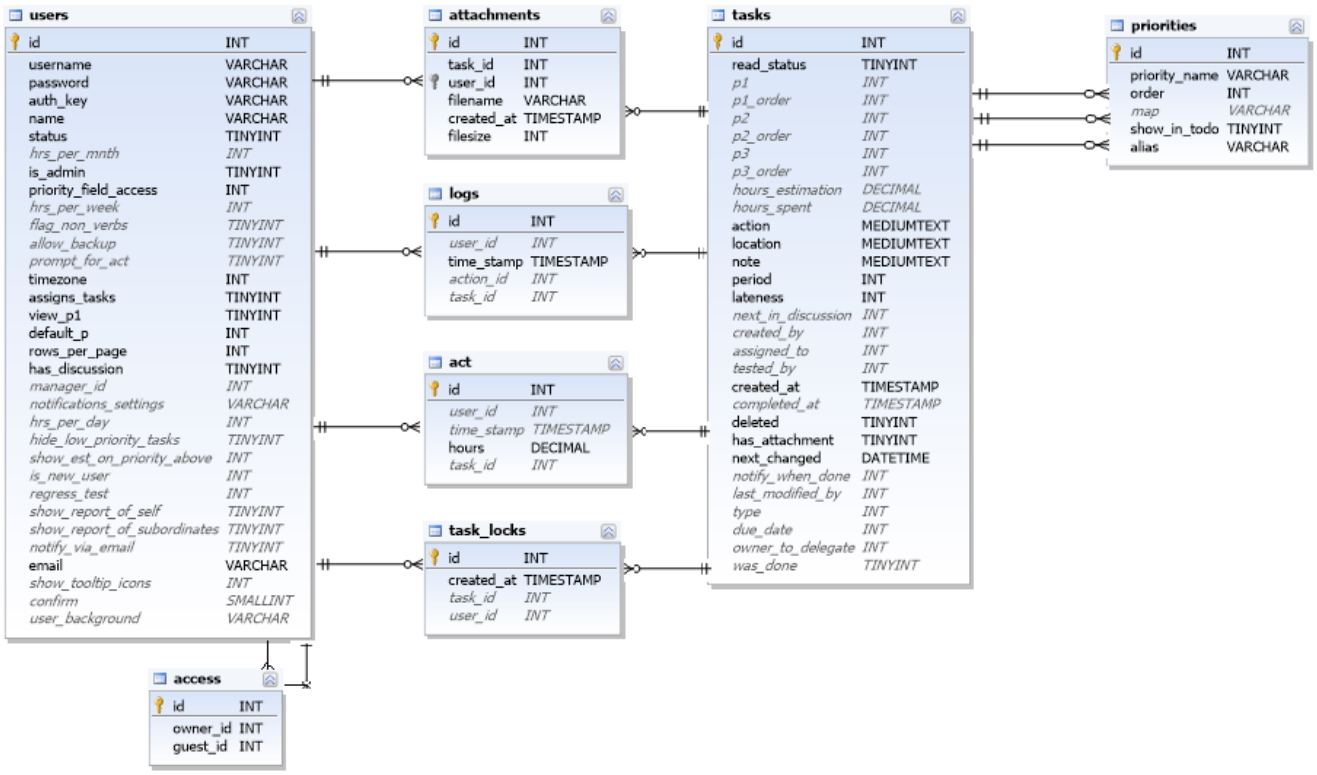


Рисунок 3.10 – Логічна модель БД

Для вдосконалення якості відображення інтерфейсу користувача і зменшенням трафіку використовуваного веб сервісом. Логіка керування

інтерфейсом користувача, перенесена на сторону клієнта. В зв’язку з чим виникає необхідність в проектуванні архітектури клієнтської частини сервісу.

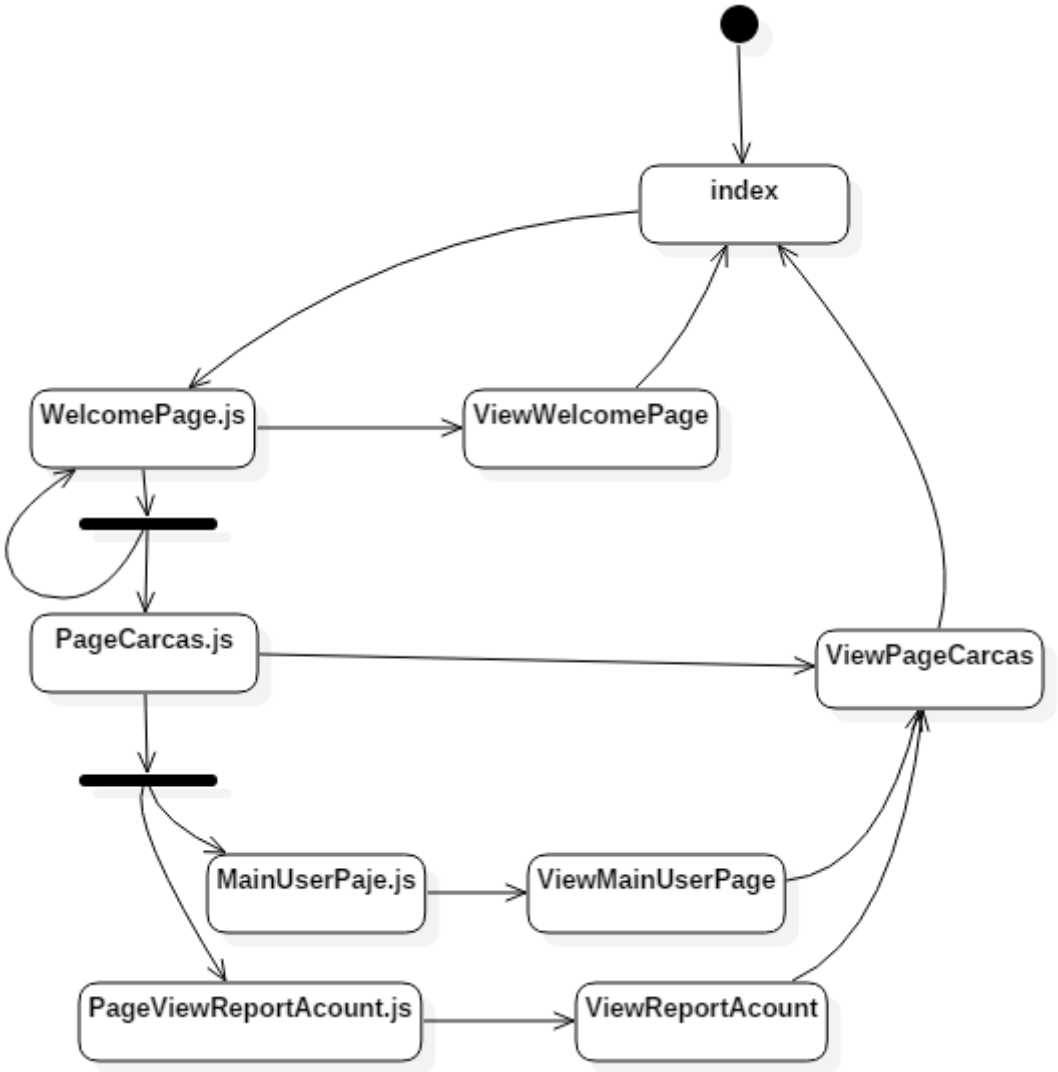


Рисунок 3.11 – Загальна модель передачі керування між скриптами при генерації сторінки користувача.

На рисунку 3.11 відображено загальну модель передачі керування між скриптами при генерації сторінки користувача, для веб сайту, що буде використовувати розроблюваний веб сервіс.

3.3 Робочій проект

3.3.1 Вибір мови та середовища розробки програмного забезпечення

Вибір мови програмування було виконано на етапі написання технічного завдання – скриптова мова PHP. Вибір обумовлено тим, що система TaskRing, для якої розробляється підсистема для оцінки розміру web додатків, написано мовою PHP версії 5.6.

Середовищем розробки було обрано PHPStorm від JetBrains, тому що ця середа розробки була створена спеціально для мови PHP, та має велику кількість корисних функцій, які прискорюють процес розробки програмного забезпечення. Крім того, PHPStorm входить до множини програмних продуктів компанії IntelliJ, які розповсюджуються за студентською ліцензією, як, наприклад IntelliJ Idea, яка неодноразово використовувалася в процесі навчання для написання програмних продуктів.

3.3.2 Кодування та випробування ПЗ

Під час випробувань було проведено повне функціональне тестування, а також навантажувальне тестування і тестування на відмову всього програмно-апаратного комплексу.

Всі виявлені недоліки ПЗ були зафіксовані в протоколах і усунуті до моменту впровадження ПЗ у дію.

Програму і методику випробувань наведено у Додатку Д.

Методи випробувань

Випробування було проведено за стратегією «чорного ящика». Через велику кількість функції, які повинно було випробувати, представлено результати розробки випробувань основних функції додатку.

Протокол тестування:

1) Проведено перевірку програми на відповідність технічному завданню:

Для користувача:

- Створення web додатку – введення інформацію про web додаток для оцінки його розміру(перевірено);

- перегляд списку web додатків(перевірено);
- редагування web додатків(перевірено);
- перегляд звіту оцінки розміру web додатків(перевірено);
- обчислення інтервальних оцінок розміру web додатку;
- обчислення точкових оцінок розміру web додатку;
- перегляд точкових оцінок розміру web додатку(перевірено);
- перегляд інтервальних оцінок розміру web додатку(перевірено);
- авторизація(перевірено).

Для адміністратора компанії:

- створення резервної копії БД компанії(перевірено);
- Перегляд звіту всіх користувачів(перевірено).

2) Проведено перевірку програми на її програмно-апаратну сумісність: виконано тестування на різних апаратно-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; виконане перевірку на наявність і усунення всіх помилок, зазначених у п.1

3) Проведено візуальний огляд програми: виконано остаточне налагодження на предмет виявлення та усунення дрібних помилок.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL

4.1 Опис ПЗ

Успіх програмних проектів залежить від якості даних, на основі яких приймаються рішення під час стратегічного та тактичного планування. Останнім часом для зменшення ризиків керування проектами по розробці програмного забезпечення використовуються гнучкі методології розробки.

Одним із найважливіших факторів, що впливають на прийняття рішення по пріоритезації задач, включення задач до backlog і т.д. є оцінювання розміру web застосунків.

Основними методами, які використовуються для оцінювання розміру web застосунків, є система вимірювання WEBMO, модель вимірювання Web-COBRA, Модель вимірювання OOmFPWeb. Однак вказані моделі не завжди адекватно враховують розподіл емпіричних даних про тривалість та трудомісткість задач, виконаних з використанням гнучкої методології. Тому задача удосконалення методів оцінювання розміру web застосунків є актуальною.

У зв’язку з вузькою спеціалізацією питання не існує готового програмного забезпечення, яке б в повній мірі вирішувало всі необхідні задачі. Впровадження програмного забезпечення для тривалості виконання задач з розробки програмного забезпечення перш за все націлене на зменшення витрат та підвищення ефективності роботи.

Створення програмного забезпечення вимагає одноразових витрат на розробку, придбання необхідних технічних засобів і поточних витрат на

функціонування. Економія від функціонування ПЗ визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного забезпечення характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

4.2 Розрахунок витрат на створення й експлуатацію ПЗ для оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel на основі нормалізуючих перетворень

Витрати на розробку системи складаються з витрат на заробітню плату розробника, на амортизацію ПЕОМ, на якій виконується розробка, на експлуатацію цієї ПЕОМ, на засоби розробки та витрат на матеріали і комплектуючі.

Розробка програмного забезпечення виконується програмістом, місячний оклад якого складає 4173 грн. Додаткова заробітна плата складає 20% від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювача системи 5007,6 грн/міс, вартість сучасної ПЕОМ складає 10000 грн. (приведена вартість машини на базі Intel Core i5). При вартості кіловат-години електроенергії рівної 1,68 грн., розраховується вартість розробки програми. Витрати на допоміжні матеріали приведені в табл. 4.1

Таблиця 4.1 – Витрати на допоміжні матеріали

Пункти витрат	Сума грн.
Папір	70.00
Заправлення картриджа до принтера	130.00
Диск	6.00
Література	300.00
Непередбачені витрати	50.00
Разом матеріали і комплектуючі вироби.	556.00

Вартість програми розрахуємо по формулі:

$$Спр = (Взп + Всф + Взг + Ве) * Т + Вм, \quad (4.1)$$

де:

Взп – основна і додаткова заробітна плата обслуговуючого персоналу, грн.

Всф – відрахування в соціальні фонди (38% від основної і додаткової заробітної плати), грн.

Взг – загальногосподарські витрати (10% від основної заробітної плати), грн.

Ве – витрати на електроенергію.

Т – тривалість розробки, міс.

Вм – витрати на основні і допоміжні матеріали.

Ве при споживанні потужності 500 Вт, тривалості роботи на місяць, рівної 21*8 = 168 годин, вартості кіловат-години електроенергії 0,98 грн.,

$$Ве = 168 * 1.68 * 0,5 = 282,24 \text{ грн.}$$

Таблиця 4.2 – Витрати на розробку системи

Найменування витрат	Одиниця виміру	Кількість
Тривалість розробки	Міс	3
Основна і додаткова заробітна	Грн.	5007,6
Відрахування в соціальні	Грн.	1902,88
Загальногосподарські витрати	Грн.	834,6
Витрати на допоміжні	Грн.	556,0
Витрати на електроенергію	Грн.	682,24

Відповідно до формули 4.1 вартість програми:

$$Спр = (5007,6 + 1902,88 + 834,6 + 82,32) * 3 + 556,0 = 24038,2 \text{ грн.}$$

Експлуатаційні витрати для ПЕОМ розрахуємо по формулі:

$$Взр = Аоб + Вм + Ве, \quad (4.2)$$

де:

Аоб – амортизаційні відрахування, грн.

Вм – річні витрати на основні і допоміжні матеріали, грн.

Ве – витрати на електроенергію, грн.

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$Аоб = 10000 * 0,6 = 6000 \text{ грн.}$$

У масштабах закладу річні витрати на основні і допоміжні матеріали (носії, папір, …) визначаються в розмірі 5% вартості основного устаткування:

$$Вм = 10000 * 0,05 = 500 \text{ грн.}$$

Річний обсяг робіт ПЕОМ у годинах визначається в такий спосіб:

$$Фм = 253,3 * Тз, \quad (4.3)$$

де Тз – це середнє денне завантаження устаткування (близько 7 годин)

253,3 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПЕОМ складе:

$$Фм = 253,3 * 7 = 1773,1 \text{ годин.}$$

Витрати на електроенергію Ве при 1773,1 годинах роботи устаткування в рік складуть

$$Ве = 1773,1* 0,98 * 0,5 = 868,81 \text{ грн.}$$

Експлуатаційні витрати для ПЕОМ за рік складуть:

$$Взр = 6000 + 500 + 868,81= 7368,81 \text{ грн.}$$

Отже, у перший рік витрати на створення й експлуатацію програми складуть:

Все = 24038,2 + 7368,81 = 31407,01 грн.

4.3 Економічна ефективність розробки та впровадження ПЗ

Основним показником економічної ефективності функціонування ПЗ є зменшення витрат, підвищення ефективності роботи та удосконалена ступень захисту.

До числа найголовніших факторів, що визначають ефективність роботи в зв'язку з впровадженням ПЗ, відносяться:

– підвищення продуктивності праці;

– підвищення швидкості обробки інформації;

– вивільнення робочого часу;

– умовно-річна економія від підвищення якості захисту інформації.

Розрахуємо пряму економічну ефективність від використання ПЗ, ґрунтуючись на тому, що впровадження програмного продукту зменшує кількість працівників (за експертною оцінкою фахівців установи близько двох чоловік), яких потрібно було б задіяти для обробки інформації на старому ПЗ.

Зарплата 2 працівників в рік складає:

4173 * 12 * 2 = 100 152 грн.

Економічний ефект першого року розраховується за формулою :

$$A_{экон.эф.} = \Delta C_n - E_n * k \text{ ,} \qquad (4.4)$$

де ΔC_n – вивільнені кошти після впровадження програмного продукту (100 152 грн.) мінус експлуатаційні витрати (31407,01 грн.) ;

ΔC_n = 100 152– 31407,01 = 68 744,99 грн.

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації (0,6));

k – одноразові витрати на впровадження продукту (12742,62 грн).

$$A_{экон.эф.} = 68\,744{,}99 - 0.6 * 24038{,}2 = 54\,322{,}07 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{П} = \frac{24038{,}2}{100152} \approx 0.24 \text{ (року),}$$

де П – вивільнені кошти після впровадження програмного продукту (100152 грн.);

k – одноразові витрати на впровадження системи (24038,2 грн.)

Отже строк окупності системи складає приблизно 2,8 місяці.

4.4 Висновки

Економічний ефект проявляється в отриманні економії витрат шляхом зменшення витрат, підвищення ефективності роботи та підвищенні якості оцінювання часу виконання задач з розробки програмного забезпечення.

З обрахованих вище даних видно, що програмний продукт з точки зору річних експлуатаційних даних може мати практичне застосування.

Реалізація даного програмного продукту є доцільною, як бачимо з отриманих значень періоду окупності та показників ефективності.

5 ОХОРОНА ПРАЦІ

5.1 Визначення, цілі, завдання та актуальність питань, пов'язаних з охороною праці

Охорона праці — це система законодавчих актів соціально-економічних, організаційно-технічних і санітарно-гігієнічних заходів, що забезпечують безпеку, збереження здоров'я і працездатність людини в процесі праці.

Загальне керівництво роботою по охороні праці на підприємстві покладається на директора і головного інженера, а безпосереднє керівництво й організацію цієї роботи здійснює особисто головний інженер, а на великих підприємствах – його заступник по охороні праці. У їхньому підпорядкуванні знаходиться відділ охорони праці, на який покладаються наступні основні функції:

- контроль за дотриманням керівниками цехів, відділів і інших структурних підрозділів діючого законодавства, стандартів, правил і норм по охороні праці; за виконанням розпоряджень контролюючих органів і наказів по підприємству;
- розробка заходів щодо створення здорових і безпечних умов праці в цехах, відділах, на споруджуваних об'єктах;
- проведення вступного інструктажу для людей, які надходять на підприємство і контроль за своєчасним і якісним проведенням інструктажу на робочих місцях;
- участь у роботі комісій з перевірки знань інженерно-технічних працівників в області техніки безпеки і виробничої санітарії, по розслідуванню причин аварій і нещасних випадків, по розгляду проектів будівництва, капітального ремонту цехів і устаткування;
- проведення паспортизації санітарно-технічного стану цехів;

- облік потерпілих від нещасних випадків на виробництві, проведення аналізу і складання звітів про виробничий травматизм, контроль за освоєнням коштів і впровадженням заходів, спрямованих на поліпшення умов праці;
- організація виставок і стендів по охороні праці.

Поліпшення умов праці сприяє:

- підвищенню ефективності виробництва;
- підтримці працездатності людини на високому рівні і його якнайшвидшому відновленні;
- скороченню виробничого травматизму;
- зменшенню професійних захворювань.

5.2 Аналіз небезпечних і шкідливих факторів, що впливають на людину при роботі з персональним комп'ютером

В проектованому приміщенні, в якому передбачається робота персоналу за ЕОМ, можливе виникнення наступних небезпечних і шкідливих факторів:

- підвищена небезпека поразки електричним струмом;
- підвищена пожежна небезпека;
- шкідливе електромагнітне випромінювання;
- недостатнє освітлення.

Одним з найбільш розповсюджених небезпечних виробничих факторів є електричний струм. Дія електричного струму на живу тканину має різнобічний і своєрідний характер. Проходячи через організм людини, електрострум здійснює термічний, електролітичний, механічний і біологічний вплив.

Термічна дія струму проявляється опіками окремих ділянок тіла, нагріванням до високої температури органів, розташованих на шляху струму, викликаючи в них значні функціональні розлади. Електролітична дія виражається в розкладанні органічної рідини, у тому числі крові, у порушенні її фізико-хімічиого складу. Механічна дія струму приводить до розшарування, розриву

тканин організму в результаті електродинамічного ефекту, а також миттєвого вибухоподібного утворення пари з рідини. Біологічна дія струму виявляється подразненням і порушенням живих тканин організму, а також порушенням внутрішніх біологічних процесів.

Припустимим вважається струм, при якому людина може самостійно звільнитися від електричного ланцюга. Його величина залежить від швидкості проходження струму через тіло людини: при тривалості дії більш 10 с - 2 мА, при 1 менш - 6 мА. Струм, при якому потерпілий не може самостійно відірватися від струмоведучих частин називається струмом, що не відпускає, і складає 10...20 мА. Змінний струм небезпечніше постійного, при високій напрузі не безпечнішим є постійний струм.

Пожежі на обжитих людиною територіях, на підприємствах виникають у більшості випадків у зв'язку з порушенням технологічного режиму. Це, на жаль, часте явище і державою передбачені спеціальні документи, що описують основи протипожежного захисту. Це стандарти: ДСТ 12.1.004-76 "Пожежна безпека" і ДСТ 12.1.010-76 "Вибухобезпека".

Заходи щодо пожежної профілактики розділяються на організаційні, технічні, режимні й експлуатаційні.

Організаційні заходи передбачають правильну експлуатацію машин і внутрішньозаводського транспорту, правильний зміст будинків, території, протипожежний інструктаж робітників та службовців, організацію добровільні пожежних дружин, пожежно-технічних комісій, видання наказів по питаннях посилення пожежної безпеки та ін.

До технічних заходів відносяться дотримання протипожежних правил, норм при проектуванні будинків, встановленні електропроводів і устаткування, опалення, вентиляції, освітлення, правил розміщення устаткування.

Заходи режимного характеру - це заборона паління в невстановлених місцях, проведення зварювальних і інших вогневих робіт у пожежно небезпечних приміщеннях та ін.

Експлуатаційними заходами є своєчасні профілактичні огляди, ремонти й випробування технологічного устаткування.

Шкідливий вплив електромагнітного випромінювання, здійснюваного обладнанням комп‘ютерних мереж і систем, обумовлений взаємодією людини як біологічного об‘єкта з оточуючим середовищем. Для нейтралізації цього впливу застосовують спеціальні екрани чи покриття моніторів, розташовують робочі місця таким чином, щоб працівник не знаходився ззаду дисплея тощо. Крім того, при розробці інтер‘єру приміщення повинні враховуватися ергономічні вимоги, зокрема, забезпечуватись правильне розташування клавіатури відносно рук оператора і монітора відносно очей людини (не менше 80 см).

Важливим фактором, що впливає на продуктивність праці, є освітленість робочого місця. Організація раціонального освітлення робочого місця - одне з основних питань охорони праці. При незадовільному освітленні різко знижується продуктивність праці, можливі нещасливі випадки, поява короткозорості, швидка стомленість.

Основною задачею виробничого освітлення є підтримка на робочому місці освітленості, що відповідає характеру зорової роботи. Збільшення освітленості робочої поверхні поліпшує видимість об'єктів за рахунок підвищення їхньої яскравості, збільшує швидкість розрізнення деталей, що позначається на зростанні продуктивності праці.

При організації виробничого освітлення необхідно забезпечити рівномірний розподіл яскравості на робочій поверхні і навколишніх предметах. Переведення погляду з яскраво освітленої на слабко освітлену поверхню змушує око переадаптуватися, що веде до стомлення зору і, відповідно, до зниження продуктивності праці. Для підвищення рівномірності природного освітлення великих приміщень здійснюється комбіноване освітлення. Світле фарбування стелі, стін і устаткування сприяє рівномірному розподілу яскравостей у полі зору працюючого.

Виробниче освітлення повинне забезпечувати відсутність у полі зору працюючого різких тіней. Наявність різких тіней спотворює розміри і форми

об'єктів розрізнення і тим самим підвищує стомлюваність, знижує продуктивність праці. Особливо шкідливі тіні, що рухаються, які можуть привести до травм.

Для поліпшення умов видимості об'єктів у полі зору працюючого повинна бути відсутня пряма і відбита блисткість. Блисткість - це підвищена яскравість світлих поверхонь, що викликає порушення зорових функцій (засліпленість), тобто погіршення видимості об'єктів. Блисткість обмежують зменшенням яскравості джерела світла, правильним вибором типу світильника.

Коливання освітленості на робочому місці, викликані, наприклад, різкою зміною напруги в мережі, обумовлюють переадаптацію ока, приводячи до значного стомлення.

Виробниче освітлення повинне забезпечувати необхідний спектральний склад світлового потоку. Це вимога особлива істотно для забезпечення правильної передачі кольору, а в окремих випадках - для посилення колірних контрастів. Оптимальний спектральний склад забезпечує природне освітлення.

Освітлювальні установки повинні бути зручні і прості в експлуатації, довговічні, відповідати вимогам естетики, електробезпечності, а також не повинні бути причиною вибуху чи пожежі.

Штучне освітлення застосовується як у темний час доби, так і у світлий час, коли для нормальних умов по освітленості природне освітлення неприпустиме чи недостатньо неможливо.

Система штучного освітлення — це група світильників з розміщеними в них електричними джерелами світла, спроектована по визначеному принципу в залежності від виконуваних задач. При розміщенні джерел світла над освітлюваною площею часто виникає необхідність одночасного рішення питання вибору світильників по таких характеристиках, як дальність дії, припустима висота підвісу, одинична потужність джерел світла і фону.

Системи штучного освітлення класифікуються по двох основних ознаках:

- конструктивне виконання;
- функціональне призначення.

По конструктивному виконанню розрізняють дві системи штучного освітлення:

- загальне (рівномірне або локалізоване);
- комбіноване, коли до загального освітлення додається місцеве, що концентрує світловий потік безпосередньо на робочому місці.

Загальне рівномірне освітлення забезпечує рівномірний розподіл світлового потоку без врахування розташування устаткування, а загальне локалізоване (нерівномірне) - з урахуванням розташування робочих місць (наприклад, у складальних цехах при технічній неможливості устаткування місцевого освітлення).

За функціональним призначенням штучне освітлення ділиться на робоче, аварійне, охоронне і чергове.

Робоче освітлення призначене для забезпечення нормальних зорових умов при виконанні робіт, проходу людей і руху транспорту. Воно забезпечує нормовані характеристики (освітленість, яскравість освітлення) у приміщеннях і місцях провадження робіт поза будинками. Для приміщень, що мають зони з різними умовами природного освітлення і зони різних режимів роботи влаштовується роздільне керування освітленням таких зон.

Аварійне освітлення поділяється на освітлення безпеки, застосовуване для продовження роботи, і евакуаційне — для евакуації з приміщення при аварійному відключенні робочого освітлення. Світильники систем аварійного освітлення підключаються до мережі, незалежно від робочого освітлення, починаючи від щита підстанції.

Чергове освітлення призначене для приміщень, а охоронне — для освітлення площадок підприємства, що охороняються в неробочий час, який збігається з темним часом доби. Для чергового освітлення виділяють частину світильників робочого чи аварійного освітлення, які забезпечують мінімальну освітленість для несення чергувань і охорони.

5.3 Розрахунок системи штучного освітлення приміщення для роботи за комп’ютером

Приміщення включає в себе лише одну кімнату і має наступний вигляд (рис. 5.1):

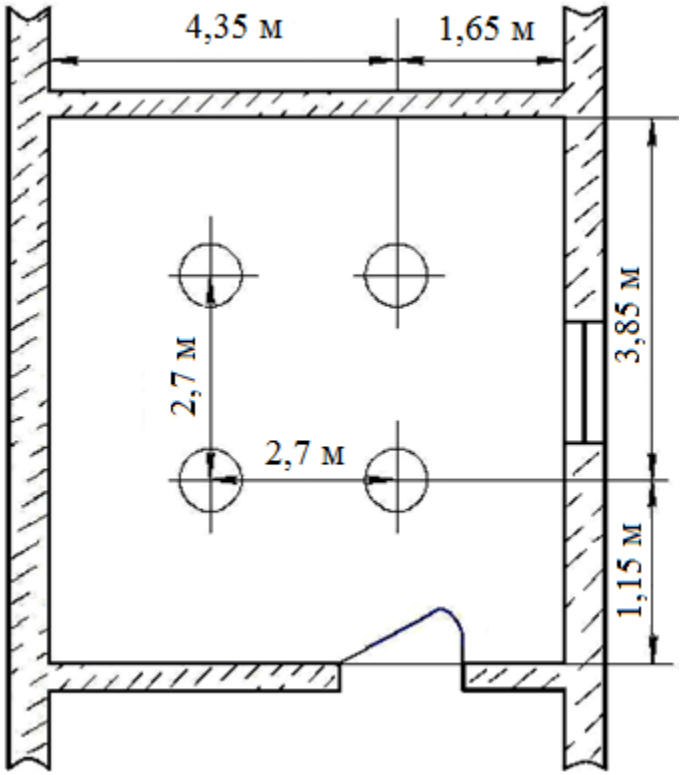


Рисунок 5.1 – Розташування світильників

Тож розрахунок буде проводитись лише для неї. Оскільки конфігурація приміщення відділу має просту форму, то розрахунок спрощується тільки до простого розрахунку.

Для виконання розрахунку використовуємо навчальний посібник “Виробниче освітлення та його розрахунок”, А.М. Тубальцев, Миколаїв 2001.

Розрахунок освітлення для приміщення:

Дано:

- довжина А = 6 м;
- ширина В = 5 м;
- висота Н = 3 м;

- висота робочої поверхні $h_р = 0.75$ м;
- для освітлення приймаємо світильники типу ПВЛМ-Р;
- мінімальна освітленість люмінесцентної лампи за нормами E_{min} = 300 лк;
- коефіцієнт відображення стелі $\rho_{стелі}$ = 70%, стін $\rho_{стін}$ = 50%, робочої поверхні $\rho_{пов}$ = 30%;
- напруга мережі 220В.

Рішення:

Відстань від стелі до робочої поверхні:

$$H_0 = H – h_р = 3 – 0.75 = 2.25 \text{ м.}$$

Відстань від стелі до світильника:

$$h_с = 0.2 \cdot H_0 = 0.2 \cdot 2.25 = 0.45 \text{ м.}$$

Висота підвішування світильника над освітлюваною поверхнею:

$$h = H_0 – h_с = 2.25 – 0.45 = 1.8 \text{ м.}$$

Висота підвішування світильника над підлогою:

$$H_п = h + h_р = 1.8 + 0.75 = 2.55 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення L / h = 1.5.

Тоді відстань між центром світильників:

$$L = 1.5 \cdot h = 1.5 \cdot 1.8 = 2.7 \text{ м.}$$

Необхідна кількість світильників:

$$N = S / L^2 = A \cdot B / L^2 = 6 \cdot 5 / 2.7^2 = 4.1 \approx 4 \text{ світ,}$$

де $S = A \cdot B$ – площа приміщення.

Приймаємо 4 світильника для кімнати, в 2 ряда по 2 шт.

$$L_1 = (A – L) / 2 = (6 – 2.7) / 2 = 1.65 \text{ м.}$$

$$L_2 = (B – L) / 2 = (5 – 2.7) / 2 = 1.15 \text{ м.}$$

Індекс приміщення:

$$i = (A \cdot B) / (h \cdot (A + B)) = (6 \cdot 5) / (1.8 \cdot (6 + 5)) = 30 / 19.8 = 1.52.$$

За таблицею «Коефіцієнт використання світлового потоку» визначаємо, що для світильників типу ПВЛМ-Р та коефіцієнтів відображення стелі $\rho_{стелі}$ = 70%, стін $\rho_{стін}$ = 50%, робочої поверхні $\rho_{пов}$ = 30% та при і = 1.52 коефіцієнт використання світового потоку $\eta = 0.6$.

Світловий потік однієї лампи:

$$\Phi_р = (E_{min} \cdot S \cdot Kз \cdot Z) / (N \cdot \eta \cdot n) = (300 \cdot 6 \cdot 5 \cdot 1.5 \cdot 1.1) / (4 \cdot 0.6 \cdot 2)$$

$$\Phi_р = 3093.75 \text{ лм.}$$

де $К_З$ = 1.5 – коефіцієнт запасу, що враховує експлуатаційне зниження освітленості, в порівнянні з запроектованою, внаслідок забруднення підволоки, переборок, світильників і ламп, а також зменшення світлового потоку ламп у процесі їхньої експлуатації;

Z – коефіцієнт мінімальної освітленості, що враховує нерівномірність освітленості. Для люмінесцентних ламп значення цього коефіцієнту приймають рівним 1.1;

n – кількість ламп у світильніках; n = 2.

За знайденим світловим потоком з таблиці «Параметри лампи розжарювання загального призначення» обираємо лампу типу ЛБ потужністю 40 Вт, що має світловий потік Ф = 3040 лм, найбільш близький до розрахункового.

При цьому фактична освітленість:

$$E_ф = E_{min} \cdot (\Phi_л / \Phi_р) = 300 \cdot (3040 / 3093.75) = 294.79 \text{ лк.}$$

Загальна потужність освітлювальної установки:

$$P_{заг} = P_л \cdot N \cdot n = 40 \cdot 4 \cdot 2 = 320 \text{ Вт} = 0.32 \text{ кВт.}$$

5.4 Заходи щодо запобігання шкідливих факторів

Усе комп’ютерне устаткування повинне відповідати міжнародним стандартам якості і безпеки.

Згідно з результатами досліджень англійської фірми RELs TEMPEST, екрани дисплеїв інтенсивно випромінюють, починаючи з частоти 10 кГц, найбільш потужне випромінювання відповідає частоті термінової розгортки – 15

кГц. Для того, щоб уникнути цього, необхідно придбати дисплеї із металевим корпусом, що зменшує потужність випромінювань на 12%.

Дисплеї на рідких кристалах (LSD-Liquid Crstal Display), що випускаються нідерландською фірмою Philips, абсолютно позбавлені будь-якого випромінювання, що дозволяє зняти питання про шкідливість низькочастотного випромінювання екрана.

З метою зниження ризику для здоров'я різними організаціями були розроблені рекомендації з параметрів моніторів, випливаючи яким виробники моніторів борють за наше здоров'я. Усі стандарти безпеки для моніторів регламентують максимально припустимі значення електричних і магнітних полів, створюваних монітором при роботі. Практично в кожній розвитій крайні є власні стандарти, але особливу популярність в усьому світі (так склалося історично) завоювали стандарти, розроблені у Швеції і відомим під ім'ям TCO.

5.4.1 TCO

TCO (The Swedish Confederation of Professional Employees, Шведська Конфедерація Професійних Колективів Робітників), членами якої є 1.3 мільйона шведських професіоналів, організаційно складається з 19 об'єднань, що працюють разом з метою поліпшення розумів роботи своїх членів. Ці 1.3 млн. членів представляють широкий спектр робітників та службовців з державного і приватного сектора економіки.

TCO ніяк не зв'язана з чи політикою релігією, що є однією з визначальних причин , що дозволяє поєднуватися різними колективним членам під дахом однієї організації.

Справа у тім, що більш 80% службовців і робітників у Швеції мають справу з комп'ютерами, тому головна задача TCO – це розробити стандарти безпеки при роботі з комп'ютерами, тобто забезпечити своїм членам і всім іншим безпечне і комфортне робоче місце. Крім розробки стандартів безпеки, TCO бере доля в створенні спеціальних інструментів для тестування моніторів і комп'ютерів.

Стандарти ТСО розроблені з метою гарантувати користувачам комп'ютерів безпечну роботові. Цим стандартом повинний відповідати кожен монітор, проданий у Швеції й у Європі. Рекомендації ТСО використовуються виробниками моніторів для створення більш якісних продуктів, що менш небезпечні для здоров'я користувачів. Суть рекомендацій ТСО складається не тільки у визначенні припустимих значень різного типові випромінювань, але й визначенні мінімально прийнятих параметрів моніторів, наприклад, підтримуваних дозволів, інтенсивності світіння люмінофора, запас яскравості, енергоспоживання, гучність і т.д. Більш того, крім вимог, і документах ТСО приводяться докладні методики тестування моніторів.

Рекомендації ТСО застосовуються як у Швеції, так і у всіх Європейських країнах для визначення стандартних параметрів, яким повинні відповідати всі монітори. До складові розроблених ТСО рекомендацій сьогодні входять три стандарти: ТСО'92, ТСО'95 і ТСО'99.

ТСО'99 пред'являє більш тверді вимоги, чим ТСО'95, а наступних областях: ергономіка (фізична, візуальна і зручність використання), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище й екологія, а також пожежна й електрична безпеку. Стандарт ТСО'99 поширюється на традиційні CRT-монітори, плоскопанельні монітори (Flat Panel Displays), портативні комп'ютери (Laptop і Notebook), системні блоки і клавіатури.

Екологічні вимоги містять у собі обмеження на присутність важких металів, бромінатів і хлоринатів, фреонів (CFC) і хлорованих речовин усередині матеріалів, більш зручні умови для роботи.

Уже зроблено клавіатури, які можуть згинатися, складатися, розділятися, скручуватися, що забезпечує більш природне положення рук та зап'ясть при друкуванні і значно знижує ймовірність професіональних захворювань, пов'язаних з монотонністю рухів.

Однак спеціалісти різних країн вважають, що остаточно вирішити цю проблему можна буде тільки при повній відмові від клавіатури і створенні комп'ютерів, які друкують і виконують команди, що подаються голосом.

Загальні рекомендації-характеристики при роботі на ПК:

1) Дотримання обмежень за медичними вказівками.

2) Уважне ставлення до характеристик дисплея.

3) Правильна організація робочого місця.

4) Раціональна організація робочого часу.

Часткові рекомендації при роботі на ПК:

- Необхідно дотримуватися часових обмежень щодо роботи з ПК для людей, які страждають захворюваннями опорно-рухового апарату, очей, шкіри, а також вагітних жінок.
- Надавати перевагу використанню дисплеїв з високою дозволяючою можливістю (дозволом) і раціональним розміром екрана (15 дюймів і більше). Не використовуйте CGA, EGA, HGA, MGA монітори.
- Краще вибирати відеоадаптери з високим дозволом і частотою оновлення екранного зображення (регенерація) не менше 72 Гц, а краще – 85 Гц і вище.
- Обов’язково ставити на незахищений дисплей екранні поляризаційні фільтри (сіточні, плівочні, пластмасові, скляні), які зменшують шкідливе електромагнітне та ультрафіолетове випромінювання, роблять мерехтіння менш помітним, захищають дисплей від осідання зарядженого пилу, знижують електростатичний заряд, затримують рентгенівське випромінювання.
- Сидіти не ближче 70 см від дисплея (приблизно на відстані витягнутої руки).
- Екран дисплея не повинен бути орієнтованим у бік джерела світла (вікон, настольних ламп та ін.).
- При розміщенні робочого місця поряд із вікном кут між екраном дисплея і площиною вікна повинен становити не менше 90 градусів (для виключення блимань), ближню частину вікна бажано зашторити.
- Не слід розміщувати дисплей безпосередньо під джерелом освітлення чи поряд із ним, уникнути виникнення блимань на екрані.

- Освітленість робочого місця не повинна перевищувати 2/3 нормальної освітленості приміщення.
- Стіна позаду дисплея повинна освітлюватися приблизно так, як і його екран.
- При розміщенні в кімнаті декількох ПК відстань між ними повинна складати не менше 1.2 м (особливо між задніми і боковими стінками сусідніх ПК, бо через них відбувається найбільш сильне випромінювання від блоків розгортки зображення).
- Робоче місце повинно бути обладнано так, щоб виключити незручні пози і довгочасні статичні напруження тіла.
- Загальний час роботи з ПК не повинен перевищувати 50% усього робочого часу, тобто не більше 4 годин на день для дорослих (за кордоном для операторів ПК передбачено скорочений робочий день – 6 годин) і 2 години для дітей.
- Не слід перевищувати темпи роботи порядку 10 тис. Нажимів клавіш за годину (приблизно 1500 слів).
- При роботі з ПК необхідно робити 15-хвилинні перерви через кожних 2 години, а при інтенсивній роботі – через 1 годину і навіть півгодини.
- Під час роботи на ПК необхідно слідкувати, щоб лінія руки в області зап’ястя залишалась прямою, зап’ястя були розслаблені, а пальці трохи зігнуті; не напружуйте руки, зігніть їх у ліктях приблизно під прямим кутом; слідкуйте, щоб удари по клавішах були не занадто сильними.
- Підлокітники робочого місця повинні бути опорою для рук як при роботі з клавіатурою, так і користуванні мишею (при цьому клавіатура і миша повинні розміщуватися так, щоб до них не потрібно було тягнутися)
- При виникненні напруження або спазмів у м’язах слід припинити роботу і зробити декілька вправ для розслаблення.
- Відрегулюйте висоту стільця так, щоб стегна розміщувалися паралельно підлозі, а ноги – твердо стояли на ній.

- Під час роботи сидіть прямо чи нахиліть корпус вперед, намагаючись зберегти природній згин тіла в поясниці; не сутультесь і не зводьте плечі; щоб не втомлювалась шия, не нахиляйте голову; верхній край монітора при цьому повинен знаходитися на рівні очей.
- Регулярно робіть вправи для очей; „переключайте" зір на різні відстані; працюючи з текстом, встановіть крупний шрифт із розміром символів не менше 12 чи 14 пунктів.
- Рекомендується розмістити біля монітора кактуси, які частково поглинають випромінення, що виділяється.
- Поверхня стола, клавіатури і корпуси ПК повинні бути одного кольору.
- Якщо можливо, з дисплеєм необхідно працювати в режимі „темні символи на світлому фоні" (найшкідливіший режим – „оранжеві символи на темному фоні")
- У приміщенні, де встановлено ПК, необхідно підтримувати відносну вологість повітря не нижче 40% з метою зменшення шкідливого впливу електростатичного поля.
- Рекомендується відсунути все обладнання ПК (і взагалі все, що включено в електричну розетку) на 60-90 см від робочого місця.
- Не рекомендується стояти перед пристроєм, що працює (наприклад, принтером при виводі документів на друг, особливо перед лазерним).

5.4.2 Забезпечення електробезпеки

Електробезпека забезпечується відповідною конструкцією електроустаткування, застосуванням технічних засобів захисту, організаційними і технічними заходами.

Конструкція електроустаткування повинна відповідати умовам його експлуатації, забезпечувати захист персоналу від зіткнення зі струмоведучими чистинами й устаткуванням – від влучення усередину сторонніх предметів і води.

Найбільш розповсюдженими технічними засобами захисту є захисне заземлення і занулення.

Організаційні і технічні заходи щодо забезпечення електробезпечності полягають в основному у відповідному навчанні, інструктажі і допуску до роботи людей, що пройшли медичний огляд, виконанні ряду технічних заходів при проведенні робіт з електроустаткуванням, дотриманні особливих вимог при роботі з частинами, що находяться під напругою.

5.4.3 Захист від електромагнітного випромінювання

Для постійного магнітного поля допустимим рівнем на робочому місці є напружність, що не повинна перевищувати 8 кА/м.

Одним з найбільш ефективних і часто застосовуваних методів захисту від низькочастотних і радіо випромінювань є екранування. Для екранів використовуються, головним чином, матеріали з великою електричною провідністю.

Як засоби індивідуального захисту застосовуються спецодяг, виготовлений з металізованої тканини у виді комбінезонів, халатів, фартухів, курток з каптурами і вмонтованими в них захисними окулярами.

6 ОХОРОНА НАВКОЛИЩНЬОГО СЕРЕДОВИЩА

6.1 Вплив людини на навколишнє середовище

Охорона навколишнього природного середовища є системою державних і суспільних заходів, направлених на збереження, відтворювання і раціональне використання природних ресурсів і поліпшення полягання природного середовища, і є частиною прикладної екології.

Відносини у галузі охорони навколишнього природного середовища в Україні регулюються Законом України № 1268-XII від 26.06.91р. «Про охорону навколишнього природного середовища», а також розроблюваними відповідно до нього земельним, водним, лісовим законодавством, законодавством про надра, про охорону атмосферного повітря, про охорону і використання рослинного і тваринного світу та іншим спеціальним законодавством.

В міру прискорення темпів науково-технічного прогресу дія людей на природу стає все більш могутньою. І в даний час воно вже сумарно із дією природних чинників, що приводить до якісної зміни співвідношення сил між суспільством і природою. На сучасному етапі людство поставлено перед чинником виникнення в природі незворотних процесів, нових шляхів переміщення і перетворення енергії і речовини. В природу втручається все більше і більше нових речовин, чужих їй, часом сильно токсичних для організмів. Частина з них не включається в природний круговорот і нагромаджується в біосфері, що приводить до небажаних екологічних наслідків.

Накопичення промислових відходів сприяє підвищенню захворюваності людей і тварин, прискоренню корозії машин і металевого устаткування, зниженню врожайності сільськогосподарських культур і продуктивності тваринництва, прискореному і нераціональному використовуванню природних ресурсів і енергії, погіршенню багатьох властивостей екологічних систем,

загибелі деяких унікальних природних територіальних комплексів, зникненню окремих видів тварин і рослин.

Основну загрозу для навколишнього середовища в межах галузі розробки програмного забезпечення становлять комплекси електронно-обчислювальних машин (надалі ЕОМ), а саме електромагнітні випромінювання від них.

Дослідження останніх років показали, що електромагнітні випромінювання вищезгаданих електронних пристроїв містять торсіонову компоненту, котра несе інформацію про процеси, що відбуваються в тому чи іншому електронному пристрої.

Торсіонові поля мають високу проникаючу здатність і їх неможливо заекранувати.

Останніми роками при проектуванні (конструюванні), виготовленні (будівництві) і експлуатації технічних і систем управління в різних сферах діяльності надзвичайно широко застосовуються персональні комп'ютери і всілякі комп’ютерні програми.

Робота з комп'ютерами програмістів, операторів і інших користувачів пов'язана з додатковими шкідливими і небезпечними чинниками, що негативно впливають на організм людини. Наприклад, шкідлива дія на зір надає не при тримання стандартних візуальних ергономічних параметрів екрану, розмір мінімального елементу відображення, мерехтіння зображення, відбивна здатність (відблиски) і ін. Працюючий комп’ютер створює електромагнітне поле, що шкідливо діє на організм людини. Це поле може викликати радіоперешкоди, тобто заважати роботі радіо і телеаппаратури, що призводить до зниження надійності технічної системи або системи управління, до збільшення риски виникнення аварійної ситуації у виробничому середовищі. Для забезпечення безпеки роботи з комп’ютером розроблені і повинні повсюдно застосовуватися стандарти на монітори, вимоги до приміщень для експлуатації комп'ютерів і до організації і устаткування робочих місць.

6.2 Утилізація відходів комп’ютерної техніки

Використання комп'ютерів вимагає вирішення таких важливих питань, як утилізація відходів (мікросхеми з вмістом кольорових металів, плати, дискети).

При утилізації старих комп'ютерів відбувається їх розділ на сім фракцій: метали, пластмаси, штекери, дроти, батареї, стекло. Жодна деталь не йде для повторного використання, оскільки не можна гарантувати їх надійність, але у формі вторинної сировини вони йдуть на виготовлення нових комп'ютерів або інших пристроїв.

Детально розглянемо декілька прикладів переробки відходів обчислювальної техніки.

Гадолінієво-галлеві гранати (ГГГ) використовуються у виробництві компонентів пристроїв, що запам'ятовують. В ході обробки біля 80% вихідного матеріалу перетворюється на відходи або відбраковується. ГГГ мають високу вартість і їх виділення з відходів представляє інтерес з економічної точки зору.

При здобутті досить чистих продуктів можливі повторне їх використання як вихідний матеріал. При цьому значно підвищується економічність виробництва заготівок з ГГГ. Під терміном відходи маються на увазі кристалічні залишки (залишки середовища для зростання кристалів, частини кристалів, виробництва, що утворюються на різних стадіях), а також дрібний порошок, що виходить при різанні, шліфовці, поліровці кристалів граната або подібних матеріалів.

Переробка цих відходів протягом ряду останніх років викликає труднощі і не вирішена до цих пір. Всі попередні спроби були безуспішними із-за низької розчинності цих складних оксидів.

Вдосконалений процес, розроблений Е.Гуссетом (патент США 4-198231 від 15 квітня 1980 року, фірма «Свісс Алюміній Лтд.», Швейцарія), призначений для виділення галію і гадолінія з відходів, що містять обидва ці елементи у вигляді оксидів або з'єднань, що перекладаються в оксиди. Відходи дрібно подрібнюються і потім розчиняються в сильних мінеральних кислотах. Гадоліній

осідає з очищених розчинів у вигляді оксалата, галій виділяється в металевому вигляді електролітично. Електролітичне виділення галію може проводитися до виділення гадолінія у вигляді оксалата з розчину.

Розглянемо приклад проведення такого процесу. Відходи піддаються переробці, є залишки завантаження в пристрої для зростання кристалів, розколені частини кристалів зі всіх стадій переробки, дрібнозернисті порошки і пудри після операції різання, шліфовки і поліровки гранатів GdGaO. Змельчений порошок після обробки кристалів ГГГ висушується при 1200 С і потім нагрівається при 6000 С протягом декількох годин для розкладання летких забруднень.

Дрібнозернисті відходи в кількості 1000 г розміром менше 40 мкм., що містять 34% галію і 46% гадолінія кип'ятяться із зворотним холодильником протягом двох годин в 2100 мл 35%-ной соляної кислоти. Це відповідає 99% завантаження. Після кип'ячення частина відходів, що не розчинилася, фільтрується і промивається. Після висушування залишок важить 20 р. Залишки такого типа об'єднуються і піддаються повторній обробці кип'ятінням. Фільтрат, що включає промивні води, об'ємом 2300 мл, обробляється 4000 г металевого галію при 500 С в течії 45 хвилин. Метал має максимально диспергувати.

В ході цієї операції благородніші елементи, присутні в розчині, виділяються і частково розчиняються в галії до його насичення і далі охолоджуються у вигляді інтерметалевих включень або в елементарній формі. Висадження можна проводити з меншою кількістю галію. Метал може періодично замінюватися на нову порцію до досягнення необхідної міри очищення. В результаті процесу очищення виходить розчин з вмістом галію 140 г/л і гадолінія 190 г/л.

Встановлюється величина *ph*= 1 шляхом додавання 900 мл 4%-ного розчину перекису водню для окислення домішок заліза. Осадження гадолінія проводиться при 500 С шляхом додавання 1500 г кристалічної технічної щавлевої кислоти COH*2HO; суспензія акуратно перемішується 12 годин для повторного осадження у вигляді оксалата гадолінія.

Оксалат гадолінія Gd(CO)*10HO відділяється центрифугуванням, промивається 20 мл розбавленої щавлевої кислоти (6 г/л) і висушується при 1300 С; перетворення на оксид гадолінія досягається прожаренням при 8000С. Подальше очищення може проводитися розчиненням в кислоті і осадженням у вигляді окасалата гадолінія. До рідини після центріфугованія і промивань осаду (3300 мл) додають 2300 г КОН при інтенсивному перемішуванні до рн= 12,6000 мл розчину з вмістом галію 55 г/л і гадолінія 1 г/л піддається електролізу при 600 Із з використанням катода з нержавеющей сталії і графітового анода при щільності струму близько 0,1 А/кв.см. Після 48 годин осідає 325 г галію і залишається розчин з вмістом 0,4 г/л, що не піддається подальшій переробці. Обложений метал має чистоту 99,99% і може бути безпосередньо перетворений в оксид.

При експлуатації персонального комп'ютера витрачаються наступні ресурси:

- електроенергія;
- папір для принтера;
- картріджи з фарбувальною стрічкою.

Для раціонального використання електроенергії не слід залишати включеним комп'ютер і принтер, якщо вони не потрібні в даний час. Друкувати можна з двох сторін. Витрати на папір навряд чи вдасться скоротити удвічі, проте економія буде вельми істотною. Проблему з утилізацією паперових відходів може вирішити вторинна переробка.

Для повторного використання картріджів з фарбувальною стрічкою необхідно купити пристрій для просочення стрічок і тоді картріджи можна буде використовувати 20-40 раз.

Сучасна технологія виготовлення елементів засобів обчислювальної техніки (СВТ) дозволяє досягти дуже низького рівня відмов елементів під час експлуатації. У зв'язку з цим відпадає необхідність проведення ремонтних робіт на місці експлуатації сучасних засобів обчислювальної техніки і як наслідок не

утворюються відходи (несправні мікросхеми), що містять дорогоцінні і рідкоземельні метали. Природно, в сервісних центрах, що спеціалізуються на ремонті і технічному обслуговуванні СВТ, має бути організований збір і облік матеріалів, що містять коштовні метали, з подальшою обробкою цих матеріалів на спеціалізованих заводах з метою їх вилучення. У зв'язку з тим, що вітчизняне виробництво сучасних компонентів інформаційних технологій знаходиться в сьогоднішні дні лише в зачатковому стані, СВТ складаються з парку імпортних машин і устаткування. Із-за відсутності інформації про вміст дорогоцінних металів в елементах устаткування, строгий облік не представляється можливим і має бути покладений на фахівців експортних фірм. В умовах ринкової економіки підприємства мають бути самі зацікавлені у вторинній переробці, що містять дорогоцінні метали вузлів і елементів за умови неможливості їх використання.

Сумарна маса дорогоцінних металів в комп'ютері типа IBM PC/XT, використовуваному в розробці даного дипломного проекту складає: золото - 0,22968 г; срібло - 5, 091336 г.

Технологічний процес витягання дорогоцінних металів з друкарських плат здійснюється за наступною схемою. Друкарські плати сортуються по переважанню в них кількості дорогоцінних металів, дробляться і подрібнюються, обпалюються і плавляться. В процесі випалення перолітіческому розкладанню піддається пластмасова основа, а основа дорогоцінних металів у вигляді металевих залишків відновлюється до оксидів. Металевий залишок подрібнюється, гранулюється, проходіт магнітну сепарацію і відбувається відділення магнітних від немагнітних часток. Отриманий таким чином порошок, розділений по видах дорогоцінних металів, у вигляді гранул розплавляється в індукційних плавильних печах з подальшим розділенням кожного металу окремо.

ВИСНОВКИ

В результаті виконання даної роботи було удосконалено існуючу модель оцінювання розміру web застосунків, що створюються з використанням фреймворку Laravel. Також, було розроблено програмне забезпечення для реалізації удосконаленої моделі.

Основні результати магістерської роботи полягають у такому:

1. проаналізовано існуючі алгоритми та моделі оцінювання розміру web застосунків;
2. удосконалено математичну модель оцінювання розміру web застосунків;
3. виконано порівняння результатів отриманих удосконаленою моделлю та існуючими моделями оцінювання розміру web застосунків;

В результаті досліджень було удосконалено математичну модель оцінювання розміру web застосунків, та в результаті проведення тестів за допомогою метрик розглянутих в розділі 4, можна сказати що оцінювання за удосконаленою моделлю більш достовірно в порівняно з іншими моделями, які були розглянуті.

4. розроблено модуль для системи TaskRing, що виконує оцінювання розміру web застосунків;
5. результати, отримані в роботі, в подальшому можуть бути використані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Laravel [Електронний ресурс] : лидер среди PHP фреймворков, одобренный разработчиками. Режим доступу : https://webformyself.com/laravel-lider-sredi-php-frejmvorkov-odobrennyj-razrabotchikami/ – Назва з екрану.

2. Scientific-practical mag. / Practical guide: IFPUG. Function Point Counting Practices Manual, release 4.1, International Function Point Users Group. - 1999. Westerville, Ohio, USA.

3. Scientific-practical mag. / Editor-in-Chief D. Reifer – 2000- . Web-Development: Estimating Quick-Time-to- Market Software. IEEE software, 2000, № 8.

4. Estimating Effort for Cross-platform Web Application Development [Електронний ресурс]. Режим доступу : https://www.diva-portal.org/smash/get/diva2:1095777/FULLTEXT01.pdf – Назва з екрану.

5. Міжнародний союз електрозв'язку [Електронний ресурс] : «ІКТ Ф дієта цифри: світ у 2015 році». Режим доступу : https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf – Назва з екрану.

6. Martino S. Di An Empirical Study on the Use of Web- COBRA and Web Objects to Estimate Web Application Development Effort / S. Di Martino, F. Ferrucci, C. Gravino // Web Engineering, Lecture Notes in Computer Science : international scientific and technical conf. 2009 p. : abstracts of rep. – 2009. - P. 213-220.

7. García Díaz V. Handbook of Research on Innovations in Systems and Software Engineering / [24] V. García Díaz, J. Manuel Cueva Lovelle, B. Cristina Pelayo García-Bustelo, 2014. – 315 p.

8. Abrahão S. A family of experiments to evaluate a functional size measurement procedure for Web applications / S. Abrahão, G. Poels // In the materials of the international conference on advanced computer information technologies, February 2009. - P. 253–269.

9. Prykhodko N.V. Constructing the non-linear regression models on the basis of multivariate normalizing transformations / N.V. Prykhodko, S.B. Prykhodko // Electronic modeling. – 2018. – Vol. 40, No. 6. – P. 101 – 110.

10. Chatterjee S. Regression analysis by example: scientific article / S. Chatterjee, B. Price, New York: John Wiley & Son, 1977, 228 p.

11. Трьохфакторна нелінійна регресійна модель для оцінювання розміру php-застосунків із відкритим кодом [Електронний ресурс]. Режим доступу: http://www.tech.vernadskyjournals.in.ua/journals/2020/1_2020/part_1/25.pdf – Назва з екрану.

12. Звіт про тенденції інтернету 2016 [Електронний ресурс] / Мері Мейкер // Режим доступу: http://www.slideshare.net/kleinerperkins/2016-internet-trends-report – Назва з екрану.

13. Ericsson AB "Звіт про мобільність Ericsson" [Електронний ресурс]. Режим доступу: http://www.ericsson.com/res/docs/2015/mobility-report/ericsson-mobility-report-nov – Назва з екрану.

14. Prykhodko S.B. Building The Non-Linear Regression Equations On The Basis Of Multivariate Normalizing Transformations / S. Prykhodko, N. Prykhodko, L. Makarova // Proceedings of First International Conference on System Analysis & Intelligent Computing (SAIC-2018), Kyiv, Ukraine, October 08–12, 2018. – P. 48 – 52.

15. Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою java [Електронний ресурс] / Л.М. Макарова, Н.В. Приходько, О.О. Кудін. Режим доступу: http://visnyk.kntu.net.ua/index.php/visnyk/article/viewFile/417/447 – Назва з екрану.

16. Інформаційні технології. Вимірювання програмного забезпечення. Вимірювання функціонального розміру. Частина 1: Визначення концепцій : ДСТУ ISO / IEC 14143-1:2013 - ДСТУ ISO / IEC 14143-1:2013. - [Чинний від 2014-07-01]. – К. : Державний стандарт України, 2014. – 181с. – (Національні стандарти України).

17. Kiciman E. AjaxScope: A platform for remotely monitoring the client-side behavior of Web 2.0 applications / E. Kiciman, B. Livshits // 21 21st Symp. on

Operating Sys. Principles: international scientific and technical symp. 2007. abstracts of rep. 2007. – P. 17-30.

18. Abrahão S. Evaluating a functional size measurement method for Web applications/ S. Abrahão, G. Poels, O. Pastor // Symposium: empirical analysis in program metrics: international scientific and technical symp. 2004, abstracts of rep. 2004. – P. 358-369.

19. Prykhodko S. Estimating the software size of open-source PHP-based systems using non-linear regression analysis / S. Prykhodko, N. Prykhodko, L. Makarova // in Proceedings of International Conference on Advanced Computer Information Technologies (ACIT 2018), June 1-3, 2018, Ceske Budejovice, Czech Republic. – P. 199 – 202.

Додаток А Технічне завдання на розробку програмного забезпечення для оцінювання розміру web застосунків, що розробленні з використанням фреймворку Laravel

Вступ

Назва проекту: Розробка модуля для оцінювання розміру web застосунків для ПЗ TaskRing

У таблиці А.1 представлена загальна інформація про програмний комплекс, який розробляється.

Таблиця А.1 – Загальна інформація про програмний комплекс.

Повна назва програмного комплексу	Модуль для оцінювання розміру web застосунків.
Умовне позначення програмного комплексу	«TaskRing»
Розробники	Бєловол Ігор

1. Підстави для розробки

«TaskRing» розробляється на підставі завдання на магістерську роботу видане кафедрою ПЗАС НУК ім. адмірала Макарова.

2. Призначення розробки

2.1. Функціональне призначення

Функціональним призначенням розробки є спрощення процесу постановки задач кожному з розробників, відслідковування процесу виконання задач, відслідковуванні скільки часу витрачається на вирішення задач, пріоритизації задач, організації комунікації розробників.

2.2. Експлуатаційне призначення.

Сервіс повинний бути доступний як для веб додаток.

3. Вимоги до програмного продукту

3.1. Вимоги до функціональних характеристик

3.1.1. Вимоги до складу виконуваних функцій

Користувачам програмного комплексу мають бути доступні функції, відповідно до прав доступу, які зазначені нижче.

Для користувача:

- створення web додатку – введення інформацію про web додаток для оцінки його розміру(LOC, Classes, AEC, Methods);
- перегляд списку web додатків;
- редагування web додатків;
- перегляд звіту оцінки розміру web додатків(точкові оцінки визначені за моделлю (2.7), та інтервальні оцінки за моделлю (2.9));
- обчислення інтервальних оцінок розміру web додатку;
- обчислення точкових оцінок розміру web додатку;
- перегляд точкових оцінок розміру web додатку;
- перегляд інтервальних оцінок розміру web додатку;
- авторизація.

Для адміністратора компанії:

- створення резервної копії БД компанії:
- Перегляд звіту всіх користувачів.

3.1.2. Вимоги до організації вхідних та вихідних даних.

Структура вхідних ті вихідних даних визначається структурою HTML сторінки.

3.2. Вимоги до надійності

Передбачити обробку ситуацій не коректного вводу інформації.

Час на відновлення працездатності у разі відмови не більше години.

3.3. Умови експлуатації

Умови експлуатації серверної частини згідно з ASHRAE:

- рекомендована температура в приміщенні 18 - 27 ° С, для цього необхідно кондиціонування повітря;
- вологість повітря в серверній повинна бути в межах від 20% до 80% без конденсації вологи; швидкість зміни вологості 6% в годину;

- запиленість не повинна перевищувати 0,75 мг / м³;
- тиск в серверній повинен перевищувати тиск в сусідніх приміщеннях. Рекомендується перевищення тиску не менше 14.7 Па.;
- рівень освітлення має становити не менше 500 лк, виміряному на висоті 1 метр в горизонтальній площині;
- рівень електромагнітного випромінювання не повинні перевищувати 3 В / м в усіх діапазонах частот;
- для певних видів обладнання і кросів необхідно обмежити вібрацію.

3.4. Вимоги до складу та характеристикам технічних засобів.

Вимоги до технічних засобів клієнтської частини співпадають з вимогам клієнту(програми).

Вимоги до технічних характеристик серверної частини представлені в таблиці А.2.

Таблиця А.2 - Вимоги до технічних характеристик серверної частини.

Параметр	Значення
Размер файла базы данных, Гб	5-25
Количество одновременных подключений	10-40
Рекомендуемый минимальный размер ОЗУ	32 Гб
Операционная система Windows Server 2003 R2 или Windows Server 2008 R2 (для персональной БД Windows XP или любая более новая версия)	64 bit
Файловая система	NTFS
Количество ядер процессоров	>= 4
Количество жестких дисков для размещения базы данных	4 HDD на 15K rpm

3.5. Вимоги до інформаційної та програмної сумісності

Вимог до інформаційної та програмної сумісності не представляється.

4. Вимоги до програмної документації

Всі регламентуючі документи є елементами програмної системи «TaskRing», які безпосередньо входять в неї. Регламент програмної системи

описує правила її ефективного використання. У таблиці А.3 наведено склад документів[1].

Таблиця А.3 – Склад документів організаційно-методичного і експлуатаційного забезпечення

№	Назва	Стадія
1	Опис структури програмного забезпечення «TaskRing»	Технічне проектування
2	Керівництво користувача	Реалізація системи
3	Текст програми	Реалізація системи
4	Опис програми	Реалізація системи
5	Програма і методика випробувань	Тестування системи

5. Стадії та етапи розробки

В таблиці А.4 представлені стадії та етапи створення системи.

Таблиця А.4 – Стадії та етапи робіт по створенню системи

№	Стадія	Документи
1	Аналіз предметної області	Звіт с описом предметної області
2	Системне проектування	Технічне завдання
3	Ескізне проектування	Ескізний проект
4	Технічне проектування	Технічний проект
5	Робоче проектування	Робочий проект
6	Здача проекту	Програмна методики випробувань. Акт проведення випробувань.

6. Порядок прийому та контролю

Для контролю та прийому повинен бути наданий опис програмного продукту, а також сам програмний продукт і методика випробування. Порядок контролю та прийому даної розробки здійснюється представником розробника згідно з програмою та методикою випробувань. Якщо програма не пройшла

випробування, виконавець зобов’язаний виправити помилки та недоліки у строк, не більш ніж один місяць від дня випробування. За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації-розробника. У випадку виявлення помилок під час прийому програмного продукту складаються акт про виявлені помилки, який підписується представниками замовника та розробника ї утверджується керівниками організації – замовника й організації – розробника. Розробник повинен у термін, який становить не більше як один місяць виправити вказані зауваження й повідомити замовника про повторне проведення перевірки, не пізніше як за два тижні до початку прийому програмного продукту.

Додаток Б Текст програми

Лістинг В.1 – Запити на створення таблиць БД відповідно до розробленої фізичної моделі

```
CREATE TABLE user (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  confirmation VARCHAR(50) DEFAULT NULL,
  dateRegistration VARCHAR(50) DEFAULT NULL,
  hashpass VARCHAR(50) NOT NULL,
  mail VARCHAR(150) NOT NULL,
  name VARCHAR(50) NOT NULL,
  PRIMARY KEY (id)
)
ENGINE = INNODB
AUTO_INCREMENT = 8
AVG_ROW_LENGTH = 8192
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE acount (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  annitation LONGTEXT NOT NULL,
  name VARCHAR(150) NOT NULL,
  user_id BIGINT(20) NOT NULL,
  type TINYINT(1) NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_e57un1oftohad5mkh5nclybmf FOREIGN KEY (user_id)
    REFERENCES user(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 1
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE fixing_acount (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  date VARCHAR(255) NOT NULL,
  user_id BIGINT(20) NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_t4kybkkecx9mk2we1h72r4iww FOREIGN KEY (user_id)
```

```
    REFERENCES user(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 1
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE goal (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  annotation VARCHAR(255) DEFAULT NULL,
  endDate VARCHAR(255) DEFAULT NULL,
  name VARCHAR(255) DEFAULT NULL,
  startDate VARCHAR(255) DEFAULT NULL,
  user_id BIGINT(20) DEFAULT NULL,
  state VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_7b7j83l6dquot72lsg25y8323 FOREIGN KEY (user_id)
    REFERENCES user(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 21
AVG_ROW_LENGTH = 4096
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE plan_to_day (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  date VARCHAR(255) DEFAULT NULL,
  user_id BIGINT(20) DEFAULT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_2fcaxdfpxb6300y9b0sqmyf4h FOREIGN KEY (user_id)
    REFERENCES user(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 18
AVG_ROW_LENGTH = 1820
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE fixing_hash_acount (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  value DOUBLE NOT NULL,
  acount_id BIGINT(20) NOT NULL,
  fixing_acout_id BIGINT(20) NOT NULL,
  PRIMARY KEY (id),
```

```
  CONSTRAINT FK_26e396o8t0bshlgwsldpikq37 FOREIGN KEY (acount_id)
    REFERENCES acount(id) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT FK_ltlj0n2ot83vfgxoggtx5rxem FOREIGN KEY (fixing_acout_id)
    REFERENCES fixing_acount(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 1
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE item_goal_plan (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  anotation VARCHAR(255) DEFAULT NULL,
  end_date VARCHAR(255) NOT NULL,
  name VARCHAR(255) NOT NULL,
  start_date VARCHAR(255) NOT NULL,
  acount_id BIGINT(20) DEFAULT NULL,
  goal_id BIGINT(20) DEFAULT NULL,
  state VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_b4gdxpqap53w6ngdxwru48ahh FOREIGN KEY (goal_id)
    REFERENCES goal(id) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT FK_bfacaqha2kfjqrwg6o08m8mre FOREIGN KEY (acount_id)
    REFERENCES acount(id) ON DELETE RESTRICT ON UPDATE RESTRICT
)
ENGINE = INNODB
AUTO_INCREMENT = 24
AVG_ROW_LENGTH = 2340
CHARACTER SET utf8
COLLATE utf8_general_ci;
CREATE TABLE item_plan_to_day (
  id BIGINT(20) NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL,
  status VARCHAR(255) DEFAULT NULL,
  time_notifice VARCHAR(255) DEFAULT NULL,
  item_goal_plan_id BIGINT(20) DEFAULT NULL,
  plan_to_day_id BIGINT(20) DEFAULT NULL,
  anotation VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (id),
  CONSTRAINT FK_5sce36yuykcr4gdv33elbovvs FOREIGN KEY (item_goal_plan_id)
    REFERENCES item_goal_plan(id) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT FK_heohxmesm3ou68j85wfla5xtn FOREIGN KEY (plan_to_day_id)
    REFERENCES plan_to_day(id) ON DELETE RESTRICT ON UPDATE RESTRICT
```

```
)
ENGINE = INNODB
AUTO_INCREMENT = 46
AVG_ROW_LENGTH = 1638
CHARACTER SET utf8
COLLATE utf8_general_ci;
```

Лістинг В.2 – Код модулю виконуючого функцію реєстрації користувач

```
public class AuthorizationFunction implements Function {

    @Override
    public String processes(String xml) throws Exception {
        ParserAuthorization handler = new ParserAuthorization();
        try {
            InputSource source = new InputSource(new StringReader(xml));
            SAXParserUtil.getParser().parse(source, handler);
        } catch (Exception e) {
            e.printStackTrace();
        }
        User user = handler.getUser();

        SessionManager sessionManager = SessionManagerImpl.getInstance();
        Object tempObj = SpringUtil.getInstance().getBean("sessionManager");
        if (tempObj instanceof SessionManager) {
            sessionManager = (SessionManager) tempObj;
        } else {
            throw new ExceptionInInitializerError("Не соответствие типов при
инициализации");
        }
        org.hibernate.Session                    sessiondao                    =
HibernateUtil.getSessionFactory().openSession();
        List<User>                                                          users
=sessiondao.createCriteria(User.class).add(Example.create(user)).list();
        sessiondao.close();
        if (users.size()>0) {
            user = users.get(0);
            Session session = sessionManager.createSession();
            session.setAttribute("user", user.getId());
            return  "<authorization><sessionId>"  +  session.getIdSession()  +
"</sessionId></authorization>";
        } else {
```

```
            return "<error>User not found</error>";
        }
    }
}
```

Лістинг В.3 – структура запитів до веб сервісу

```
<?xml version="1.0" encoding="UTF-8"?>

<request
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xsi:noNamespaceSchemaLocation='APIRequest.xsd'>
    <!--Пример запроса на регистрацию-->
    <registration>
        <name>YuriPanasenco</name>
        <mail>Panasenco@outlook.com</mail>
        <hashPass>12345</hashPass>
    </registration>
    <!--Пример запроса на авторизацию-->
    <authorization>
        <name>ebtb</name>
        <hashPass>sdfvrf</hashPass>
    </authorization>
    <!--пример запроса на создание цели-->
    <createGoal>
        <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
        <name>Название цели</name>
        <anotation>Краткое описание цели</anotation>
        <beginDate>07.09.2016</beginDate>
        <endDate>01.07.2018</endDate>
        <plan>
            <item>
                <number>1</number>
                <name>Получить высшее образование (Уровень магистр)</name>
                <beginDate>01.09.2016</beginDate>
                <endDate>01.07.2018</endDate>
            </item>
            <item>
                <number>2</number>
                <name>Накопить 5 тыс. доларов</name>
                <beginDate>01.09.2016</beginDate>
                <endDate>01.07.2018</endDate>
```

```
            <idAcount>4</idAcount>
        </item>
    </plan>
</createGoal>
<!--пример запроса на закрытие сесии-->
<closeSession>
    <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
</closeSession>
<!--пример запроса на получение информации о пользователе-->
<userState>
    <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
</userState>
<!--пример запроса на получение списка пользовательских целей-->
<userGoals>
    <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
</userGoals>
<!--пример запроса на добавление задачи на завтра-->
<addTaskOnTommorow>
    <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
    <task>
        <name>название</name>
    </task>
    <task>
        <name>название </name>
        <notification>15:30</notification>
        <idItemGoalPlan>35</idItemGoalPlan>
    </task>
</addTaskOnTommorow>
<!--пример запроса на изменения или удаления задачи-->
<updateTaskOnDay>
    <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
    <task>
        <operation>update</operation>
        <id>345</id>
        <name>название задачи</name>
        <notification>15:30</notification>
        <idItemGoalPlan>35</idItemGoalPlan>
    </task>
    <task>
        <operation>delete</operation>
        <id>3435</id>
    </task>
```

```
            <task>
                <operation>update</operation>
                <id>632</id>
                <state>2</state>
            </task>
        </updateTaskOnDay>
        <!--пример запроса на получения задач пользователя-->
        <globalplan>
            <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
        </globalplan>
        <!--пример запроса на получение информации по конкретной задачи-->
        <goalInfo>
            <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
            <golaID>234</golaID>
        </goalInfo>
        <!--Возвращает список дел выполненых для достижения поставленой задачи->
        <goalReport>
            <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
            <goalId>234</goalId>
        </goalReport>
        <!--пример запроса на получение статистики пользователя-->
        <statistics>
            <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
        </statistics>
        <acountReport>
            <sessionId>rgthryujtikyt</sessionId>
            <acountId>45</acountId>
        </acountReport>
        <!--Пример запроса для подтверждения регистрации-->
        <confirmation>
            <sessionId>tnlekmgregreko54ymn6oyh5k4my</sessionId>
            <key>uijonuhibjonot6f5ft7</key>
        </confirmation>
    </request>
```

Додаток В Опис програми

Розроблена програма необхідна для оцінювання розміру web застосунків з використанням фреймворку Laravel. Програма має клієнт-серверну архітектуру та виконується на сервері за адресою https://app.TaskRing.com.

Програма написана мовою програмування PHP з використанням елементів JavaScript. Та дозволяє виконувати наступні дії:

Для користувача:

- створення web додатку – введення інформацію про web додаток для оцінки його розміру(LOC, Classes, AEC, Methods);
- перегляд списку web додатків;
- редагування web додатків;
- перегляд звіту оцінки розміру web додатків(точкові оцінки визначені за моделлю (2.7), та інтервальні оцінки за моделлю (2.9));
- обчислення інтервальних оцінок розміру web додатку;
- обчислення точкових оцінок розміру web додатку;
- перегляд точкових оцінок розміру web додатку;
- перегляд інтервальних оцінок розміру web додатку;
- авторизація.

Для адміністратора компанії:

- створення резервної копії БД компанії:
- Перегляд звіту всіх користувачів.

Додаток Г Інструація користувача

Для початку використання веб-додатку необхідно перейти за наступним посиланням https://app.TaskRing.com/login (див. рис. Г.1), де система запропонує ввести назву компанії.

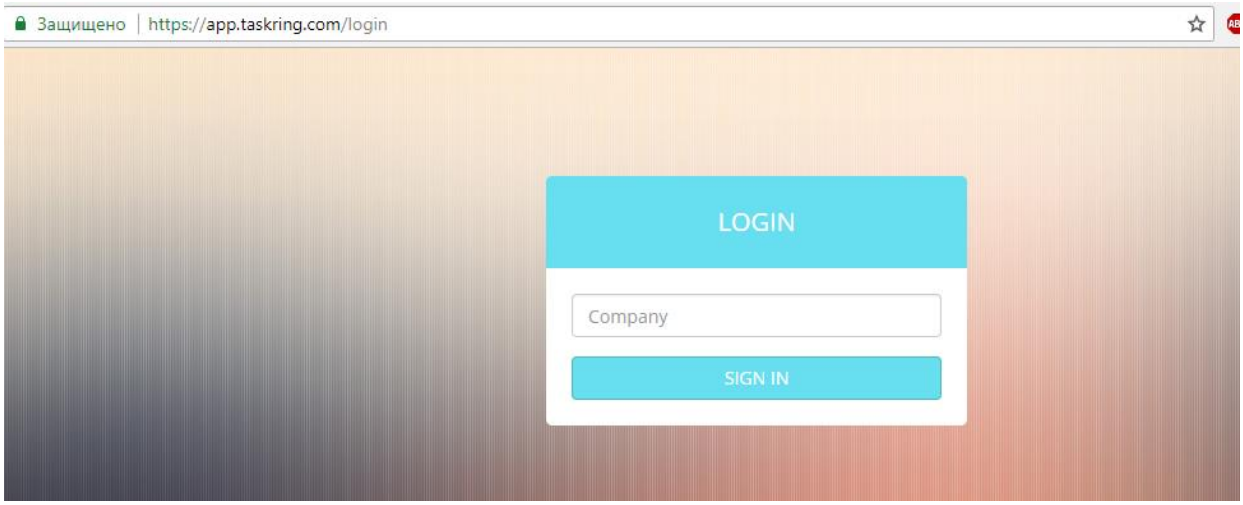


Рисунок Г.1 – Форма 1-го етапу авторизації

Після чого система запропонує ввести логін та пароль користувача (див. рис. Г.2)

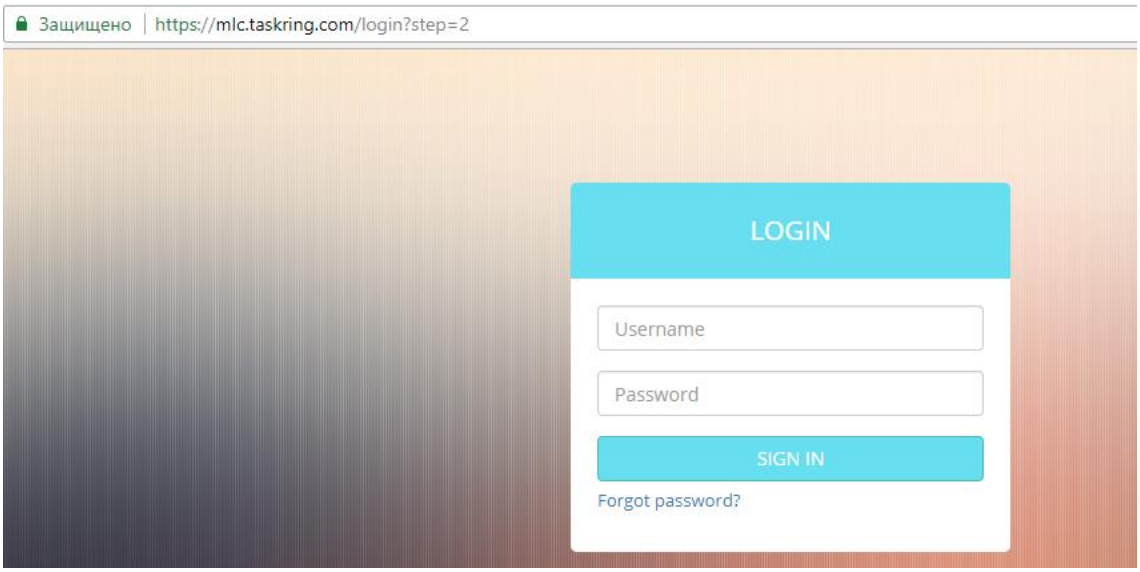
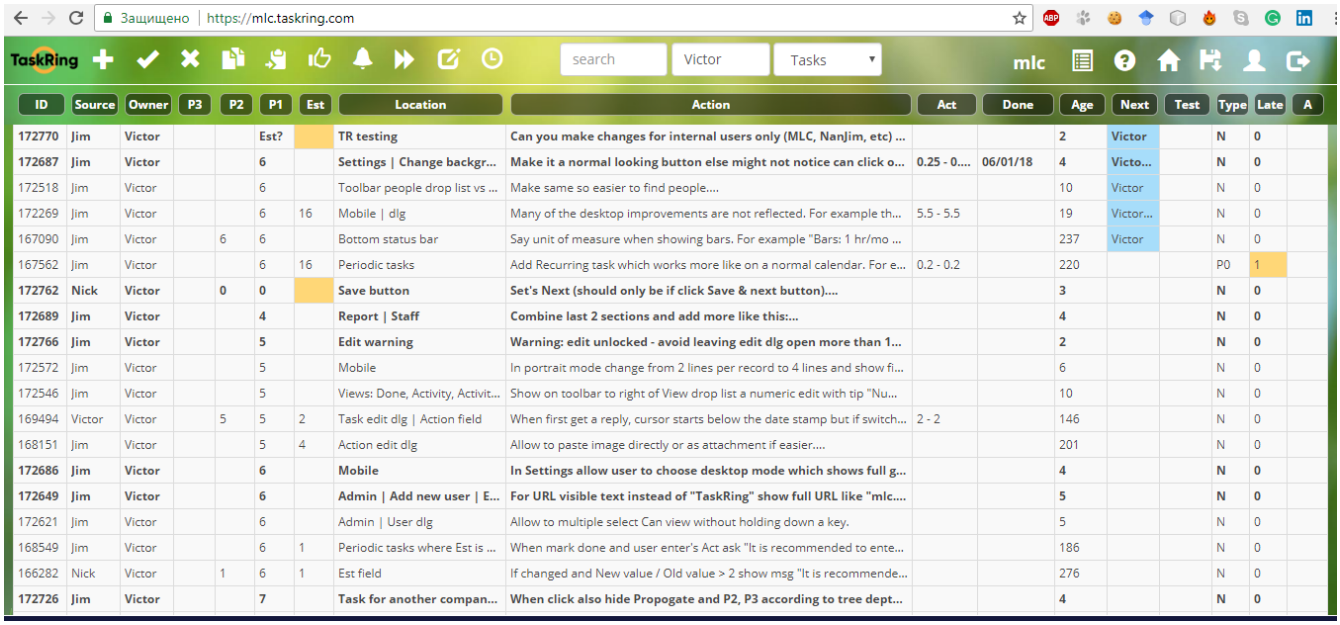


Рисунок Г.2 – Форма другого етапу авторизації

Після успішного проведення авторизації система користувача направить до головної сторінки (див. рис. Г.3)



ID	Source	Owner	P3	P2	P1	Est	Location	Action	Act	Done	Age	Next	Test	Type	Late	A
172770	Jim	Victor			Est?		TR testing	Can you make changes for internal users only (MLC, NanJim, etc) ...			2	Victor		N	0	
172687	Jim	Victor			6		Settings | Change backgr...	Make it a normal looking button else might not notice can click o...	0.25 - 0....	06/01/18	4	Victo...		N	0	
172518	Jim	Victor			6		Toolbar people drop list vs ...	Make same so easier to find people....			10	Victor		N	0	
172269	Jim	Victor			6	16	Mobile | dlg	Many of the desktop improvements are not reflected. For example th...	5.5 - 5.5		19	Victor...		N	0	
167090	Jim	Victor		6	6		Bottom status bar	Say unit of measure when showing bars. For example "Bars: 1 hr/mo ...			237	Victor		N	0	
167562	Jim	Victor			6	16	Periodic tasks	Add Recurring task which works more like on a normal calendar. For e...	0.2 - 0.2		220			P0	1	
172762	Nick	Victor		0	0		Save button	Set's Next (should only be if click Save & next button)....			3			N	0	
172689	Jim	Victor			4		Report | Staff	Combine last 2 sections and add more like this:...			4			N	0	
172766	Jim	Victor			5		Edit warning	Warning: edit unlocked - avoid leaving edit dlg open more than 1...			2			N	0	
172572	Jim	Victor			5		Mobile	In portrait mode change from 2 lines per record to 4 lines and show fi...			6			N	0	
172546	Jim	Victor			5		Views: Done, Activity, Activit...	Show on toolbar to right of View drop list a numeric edit with tip "Nu...			10			N	0	
169494	Victor	Victor		5	5	2	Task edit dlg | Action field	When first get a reply, cursor starts below the date stamp but if switch...	2 - 2		146			N	0	
168151	Jim	Victor			5	4	Action edit dlg	Allow to paste image directly or as attachment if easier....			201			N	0	
172686	Jim	Victor			6		Mobile	In Settings allow user to choose desktop mode which shows full g...			4			N	0	
172649	Jim	Victor			6		Admin | Add new user | E...	For URL visible text instead of "TaskRing" show full URL like "mlc....			5			N	0	
172621	Jim	Victor			6		Admin | User dlg	Allow to multiple select Can view without holding down a key.			5			N	0	
168549	Jim	Victor			6	1	Periodic tasks where Est is ...	When mark done and user enter's Act ask "It is recommended to ente...			186			N	0	
166282	Nick	Victor		1	6	1	Est field	If changed and New value / Old value > 2 show msg "It is recommende...			276			N	0	
172726	Jim	Victor			7		Task for another compan...	When click also hide Propogate and P2, P3 according to tree dept...			4			N	0	

Рисунок Г.3 – Головна сторінка користувача.

На головній сторінці користувача представлено перелік web додатків. Для перегляду web додатку необхідно вибрати відповідний рядок, та натиснути кнопку редагування на верхній панелі керування. Після чого системі відобразить форму редагування web додатку(див. рис. Г.5.).

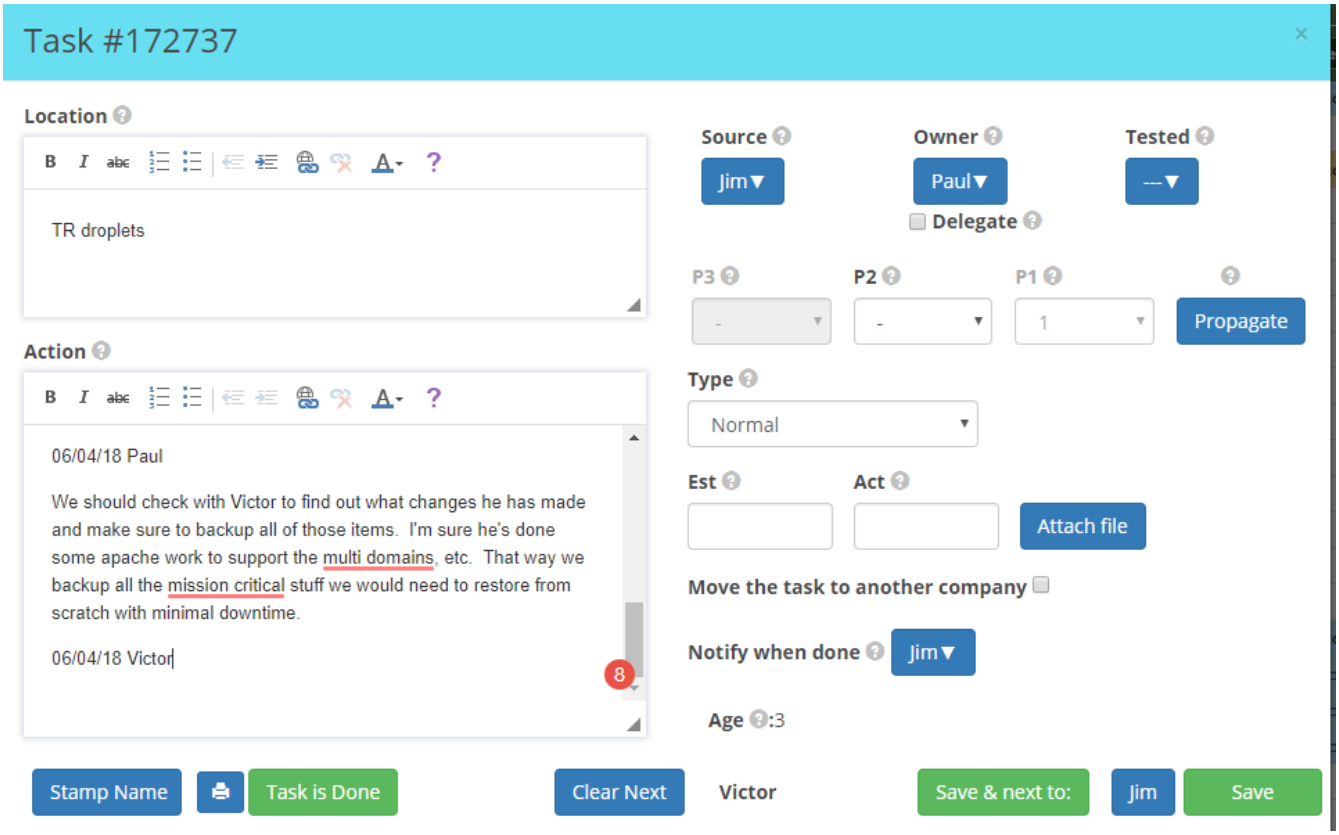


Рисунок Г.4 – Форма редагування web додатків

Ні формі редагування web додатків користувач має можливість ввести інформації про web додаток

Також в системі присутня можливість перегляду звіту оцінки розміру web додатків (див. рис. Г.5).

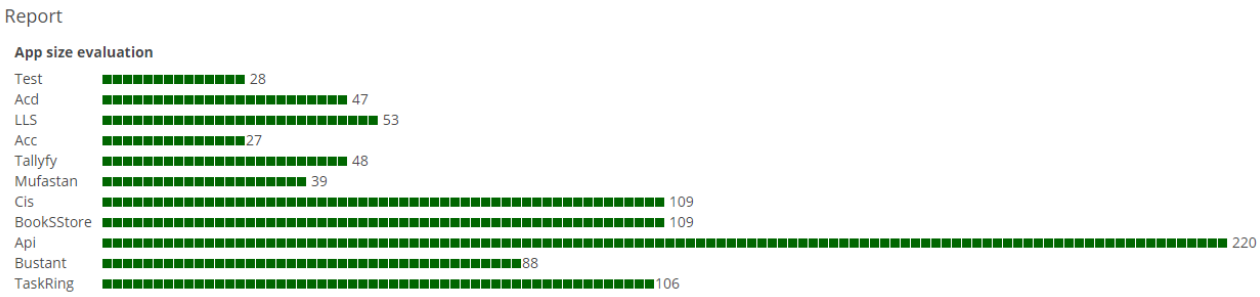


Рисунок Г.6 – Звіт оцінки розміру web додатків

Додаток Д Програма і методика випробувань

1. Обєкт випробувань

Назва розроблюваного проекту: «ПЗ для оцінювання розміру web застосунків з використанням фреймворку Laravel TaskRing». Галузь застосування – підприємство «Micro Logic Corporation».

2. Мета випробувань

Мета проведення випробувань оцінки експлуатаційних характеристик програми, перевірка і підтвердження працездатності програми в умовах, максимально наближених до у умов реальної експлуатації.

3. Вимоги до програми

Програма має реалізувати функції, описані в технічному завданні, яке наведено в додатку А.

4. Вимоги до програмної документації

Для проведення тестування програми, повинна бути надана наступна документація:

- Технічне завдання на розробку програми (вимоги до складання документу визначаються ДСТ 19.101-77);
- Текст програми (зміст і оформлення документу визначається ДСТ 19.101-77)
- Опис програми
- Інструкція користувача
- Програма і методика випробувань ПЗ (вимоги до складання документу визначаються ДСТ 19.101-77);

5. Склад порядок та методи випробувань

5.1 Перевірка програми на відповідність технічному завданню:

Для користувача:

- створення web додатку – введення інформацію про web додаток для оцінки його розміру;
- перегляд списку web додатків;
- редагування web додатків;
- перегляд звіту оцінки розміру web додатків(точкові оцінки визначені за моделлю (2.7), та інтервальні оцінки за моделлю (2.9));
- обчислення інтервальних оцінок розміру web додатку;
- обчислення точкових оцінок розміру web додатку;
- перегляд точкових оцінок розміру web додатку;
- перегляд інтервальних оцінок розміру web додатку;
- авторизація.

Для адміністратора компанії:

- створення резервної копії БД компанії;
- Перегляд звіту всіх користувачів.

Перегляд звіту всіх користувачів.

5.2 Перевірка програми на її апаратно-програмну сумістність: тестування на апаратних-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; перевірка на наявність і усунення всіх помилок, зазначених у п. 5.1.

5.3 Візуальних перегляд програми: остаточне налагодження на предмет виявлення на усунення подібних помилок