

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інформаційної системи автоматизації обліку товарообігу
засобами PHP, MySQL, HTML5 та CSS3

Виконав: студент IV курсу, групи СНС-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Саган О.А.

(прізвище та ініціали)

Керівник

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Карпінський М.П.

(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«21» червня 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Сагану Олександрю Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи автоматизації обліку товарообігу засобами PHP, MySQL, HTML5, CSS3

Керівник роботи Млинко Богдана Богданівна, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «7» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 22 червня 2023р.

3. Вихідні дані до роботи Літературні та інтернет джерела про методи проектування інформаційних систем, автоматизацію бізнес-процесів та технології веброботки застосунку Для автоматизації обліку товарообігу за допомогою PHP та MySQL.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз аналогів ІСАОТ та постановка завдання. Аналіз існуючих вебзастосунків в галузі обліку складських операцій та доцільності створення ІСАОТ. Обґрунтування використовуваних вебтехнологій реалізації ІСАОТ. Вибір середовища розробки ІСАОТ. Проектування, реалізація, встановлення та тестування ІСАОТ. Вибір оптимального методу реалізації ІСАОТ. Життєвий цикл ІСАОТ. Формалізація вимог. Побудова дерева функцій та проектування структури системи. Модулювання архітектури ІСАОТ. Проектування поведінки ІСАОТ. Проектування бази даних ІСАОТ. Проектування користувацького вебінтерфейсу ІСАОТ. Реалізація основних функцій ІСАОТ. Реалізація користувацького вебінтерфейсу ІСАОТ. Програмна реалізація основних структурних елементів ІСАОТ. Встановлення, налаштування, верифікація та тестування ІСАОТ. 3. Безпека життєдіяльності, основи охорони праці. Особливості електротравматизму, електричний струм як чинник небезпеки для користувачів ІСАОТ. Вимоги ергономіки до організації робочого місця користувача ПК, що використовуватиме ІСАОТ. Висновки. Перелік джерел.

Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний слайд. 2. Мета, об'єкт дослідження. 3. Актуальність обраної теми 4. Аналіз предметної області. 5. Варіанти використання ІСАОТ. 6. Огляд технологій реалізації.

7. Структура. 8. Інформаційні сутності. 9. База даних. 10. Інтерфейс користувача.

11. Висновки.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Безпека життєдіяльності, основи охорони праці | | | |

7. Дата видачі завдання 23 січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|-------|---|--------------------------------|----------|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | 23.01.2023 | Виконано |
| 2. | Підбір джерел про вебтехнології та бази даних | 24.01.2023-26.01.2023 | Виконано |
| 3. | Опрацювання джерел по темі кваліфікаційної роботи | 27.01.2023-31.01.2023 | Виконано |
| 4. | Виконання дослідження щодо розробки інформаційної системи та автоматизації бізнес-процесів обліку товарообігу | 01.02.2023-07.02.2023 | Виконано |
| 5. | Розроблення ІСАОТ | 07.02.2023-08.02.2023 | Виконано |
| 6. | Оформлення розділу «Аналіз аналогів ІСАОТ та постановка завдання» | 08.02.2023-09.02.2023 | Виконано |
| 7. | Оформлення розділу «Проектування, реалізація, встановлення та тестування ІСАОТ» | 10.02.2023-12.02.2023 | Виконано |
| 8. | Виконання завдання до підрозділу «Особливості електротравматизму, електричний струм як чинник небезпеки для користувачів ІСАОТ» | 05.06.2023-06.06.2023 | Виконано |
| 9. | Виконання завдання до підрозділу «Вимоги ергономіки до організації робочого місця користувача ПК, що використовує ІСАОТ» | 07.06.2023-08.06.2023 | Виконано |
| 10. | Оформлення кваліфікаційної роботи | 09.06.2023-11.06.2023 | Виконано |
| 11. | Нормоконтроль | 12.06.2023-13.06.2023 | Виконано |
| 12. | Перевірка на плагіат | | Виконано |
| 13. | Попередній захист кваліфікаційної роботи | | Виконано |
| 14. | Захист кваліфікаційної роботи | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент

(підпис)

Саган О.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційної системи автоматизації обліку товарообігу засобами PHP, MySQL, HTML5 та CSS3 // Кваліфікаційна робота освітнього рівня «Бакалавр» // Саган Олександр Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // С. 57 , рис. – 58, табл. – 11, кресл. – 0, додат. – 10, бібліогр. – 52.

Ключові слова: інформаційні системи, бази даних, зберезувальні процедури, бізнес-процеси, вебтехнології, облік, автоматизація.

Кваліфікаційна робота присвячена дослідженню можливостей автоматизації бізнес-процесів шляхом використання інформаційних технологій. В першому розділі кваліфікаційної роботи описано системи-аналоги інформаційно системи автоматизації обліку товарообігу. Висвітлено їх переваги, недоліки. Розглянуто можливі технології реалізації. Проаналізовано доцільність розробки ІСАОТ шляхом використання МАІ і в підсумку обрано вебтехнології для реалізації ІСАОТ.

В другому розділі кваліфікаційної роботи було досліджено методи розробки вебзастосунків, обрано оптимальний метод, описано життєвий цикл ІСАОТ та формалізовано вимоги до системи. Крім того спроектовано архітектуру ІСАОТ та її поведінку. В рамках проектування вебзастосунку було виділено інформаційні сутності, спроектовано базу даних, рівень логіки та клієнтську частину ІСАОТ. Також розділ описує реалізацію, тестування вебзастосунку та усунення виявлених недоліків.

В третьому розділі кваліфікаційної роботи описано актуальні для користувачів ІСАОТ фактори небезпеки та засоби захисту від них, ергономічні вимоги до облаштування робочого місця.

ANNOTATION

Information System Development Automated Accounting of Commodity by Means of PHP, MySQL, HTML5, CSS3 // Qualification work of the educational level "Bachelor" // Sahan Oleksandr // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-41 // Ternopil, 2023 // P. – 57, fig. – 58, tabl. – 11, chair. – 0, annexes. – 10, references – 52.

Keywords: information systems, databases, storage procedures, business processes, web technologies, accounting, automation.

The qualification work is dedicated to researching the possibilities of automating business processes through the use of information technologies. In the first section of the qualification work, analogue systems of the information system of the automation of merchandise turnover accounting are described. Their advantages and disadvantages are highlighted. Possible implementation technologies are considered. The expediency of developing AAISCC through the use of MAH was analyzed and, as a result, web technologies were chosen for the implementation of AAISCC.

In the second section of the qualification work, the methods of developing web applications were investigated, the optimal method was chosen, the life cycle of AAISCC was described, and the requirements for the system were formalized. In addition, AAISCC architecture and its behavior were designed. As part of the design of the web application, information entities were selected, the database, the logic level and the client part of AAISCC were designed. Also, the section describes the implementation, testing of the web application and elimination of identified shortcomings.

The third section of the qualification work describes the danger factors relevant to users of AAISCC and the means of protection against them. The main ergonomic requirements for arranging the workplace of AAISCC users are also considered.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І ТЕРМІНІВ

AAISCC – Automated Accounting Information System of Commodity Circulation

ACL (англ. access control list) – список керування доступом.

AJAX (англ. Asynchronous JavaScript And XML) – підхід до побудови користувацьких вебзастосунків.

DOM (англ. Document Object Model) – об'єктна модель документу.

HTML (англ. HyperText Markup Language – стандартизована мова гіпертекстової розмітки.

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки.

CSS – cascading style sheets.

JSON (англ. JavaScript Object Notation – запис об'єктів JavaScript.

МАН – method of analysis of hierarchies

ORM – object-relational mapping.

SQL – structured query language.

БД – база даних.

Вольт (В, V) – одиниця вимірювання електричної напруги, електрорушійної сили та різниці потенціалів в системі SI.

ЗП – зберезувальні процедури.

ОС – операційна система.

ПЗ – програмне забезпечення.

СУБД – система управління базами даних.

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 8 |
| РОЗДІЛ 1. АНАЛІЗ АНАЛОГІВ ІСАОТ ТА ПОСТАНОВКА ЗАВДАННЯ..... | 9 |
| 1.1 Аналіз існуючих вебзастосунків в галузі обліку складських операцій та доцільності створення ІСАОТ | 9 |
| 1.2 Обґрунтування використуваних вебтехнологій реалізації ІСАОТ | 15 |
| 1.3 Вибір середовища розробки ІСАОТ | 17 |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ, ВСТАНОВЛЕННЯ ТА ТЕСТУВАННЯ ІСАОТ | 18 |
| 2.1 Вибір оптимального методу реалізації ІСАОТ | 18 |
| 2.2 Життєвий цикл ІСАОТ..... | 18 |
| 2.3 Формалізація вимог | 20 |
| 2.4 Побудова дерева функцій та проєктування структури ІСАОТ | 22 |
| 2.5 Моделювання архітектури ІСАОТ | 24 |
| 2.6 Проєктування поведінки ІСАОТ..... | 25 |
| 2.7 Проєктування бази даних ІСАОТ | 25 |
| 2.8 Проєктування користувацького вебінтерфейсу ІСАОТ | 28 |
| 2.9 Реалізація основних функцій ІСАОТ | 30 |
| 2.10 Реалізація користувацького вебінтерфейсу ІСАОТ..... | 31 |
| 2.11 Програмна реалізація основних структурних елементів ІСАОТ | 34 |
| 2.12 Встановлення та налаштування ІСАОТ | 42 |
| 2.13 Тестування та верифікація ІСАОТ | 43 |
| РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ | 46 |
| 3.1 Особливості електротравматизму, електричний струм як чинник небезпеки для користувачів ІСАОТ..... | 46 |
| 3.2 Вимоги ергономіки до організації робочого місця користувача ПК, що використовуватиме ІСАОТ | 49 |
| 3.3 Висновок до третього розділу | 51 |
| ВИСНОВКИ..... | 52 |
| ПЕРЕЛІК ДЖЕРЕЛ | 53 |
| ДОДАТКИ | |

ВСТУП

Інформаційні технології дозволяють ефективно організувати діяльність людини. Прикладом такої організації може виступати використання вебсервісів, що дозволяють обмінюватися інформацією на різних платформах. Для забезпечення роботи користувача у цих вебсервісах використовуються вебінтерфейси.

Метою дослідження є підвищення ефективності роботи підприємств, які займаються збутом товарів різних категорій, потребують обліку таких операцій та додаткових функцій вебсервісу, що можуть бути потрібні в ході діяльності таких підприємств.

Для досягнення мети було сформовано ряд завдань:

- визначити методи реалізації вебзастосунку для обліку товарообігу;
- виконати проєктування інформаційної системи автоматизація обліку товарообігу;
- обрати вебтехнології, що відповідатимуть потребам ІСАОТ;
- розробити вебзастосунок для автоматизації обліку товарообігу.

Отже, метою даної кваліфікаційної роботи є підвищення ефективності роботи підприємств, що займаються збутом товарів різних категорій, шляхом впровадження вебзастосунку для автоматизації обліку товарообігу та пов'язаних бізнес-процесів, що забезпечить зберігання даних про керівників підприємства, продавців, клієнтів, товари, магазини, роботи, їх стан, пріоритетність, а також забезпечить комунікативну функцію.

РОЗДІЛ 1. АНАЛІЗ АНАЛОГІВ ІСАОТ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз існуючих вебзастосунків в галузі обліку складських операцій та доцільності створення ІСАОТ

В якості аналогів інформаційної системи автоматизації обліку товарообігу було розглянуто 3 вебсервіси: система керування підприємством «Мій склад», система керування складом «Simple WMS» та програма для обліку продажів «РемОнлайн».

На сучасному етапі розвитку суспільства та інформаційних технологій важливою складовою успішного розвитку підприємства є автоматизація облікових операцій та доступність інформації [1]. Потужності обчислювальної техніки надають необхідні інструменти для досягнення певного ступеня автоматизації таких процесів. Вони надають ряд переваг:

- ефективна документація операцій;
- точна інвентаризація;
- наочність товарообігу;
- формування підсумків.

Обчислювальна техніка суттєво покращує та спрощує обробку облікової інформації. Це досягається шляхом зменшення кількості ручних операцій та зниженням впливу людського фактору. Прикладом таких інструментів можуть слугувати розглянуті аналоги [2].

Висновок про доцільність розробки ІСАОТ було зроблено на основі аналізу існуючих вебсервісів та розробкою інформаційної системи за рядом критеріїв шляхом використання МАІ [3]. Виділено 2 основні критерії: «функціонал» та «оформлення».

Функціонал включає в себе наступні підкритерії:

- відповідність (рівень відповідності системи вимогам);

– масштабованість (можливість розширення системи, додавання функцій);

– конфіденційність (менша кількість персональної інформації необхідної для роботи системи забезпечує вищу конфіденційність).

Оформлення включає наступні підкритерії:

– зручність (зрозумілість та зручність інтерфейсу системи);

– адаптивність (наявність мобільних додатків, адаптивного інтерфейсу);

– зрозумілість (початкове навчання, система підказок).

Отже, потрібно обрати найкращу альтернативу за виділеними критеріями. В якості альтернатив розглядається 5 різних сервісів. Переваги та недоліки сервісу «Мій склад» подано на рисунку 1.1.

| | | | | | |
|------------------|---|----------------|------------------------------|---------------------|--------|
| Мій склад | | | | | |
| Відповідність | клієнти | облік поставок | облік продажів | аналіз даних | знижки |
| Масштабованість | розширюваність | | можливість додати функціонал | | |
| Конфіденційність | вимагає малу кількість персональних даних | | | | |
| Зручність | простий інтерфейс | | | доступність функцій | |
| Адаптивність | мобільний додаток | додаток для ПК | адаптивний інтерфейс | | |
| Зрозумілість | навчання | підказки | підтримка | | |

Рисунок 1.1 – Результати аналізу вебсервісу «Мій склад»

Переваги та недоліки інших сервісів наведено в додатку А. В результаті декомпозиції проблеми для подальшого аналізу за допомогою МАІ було отримано ієрархію, як на рисунку 1.2.

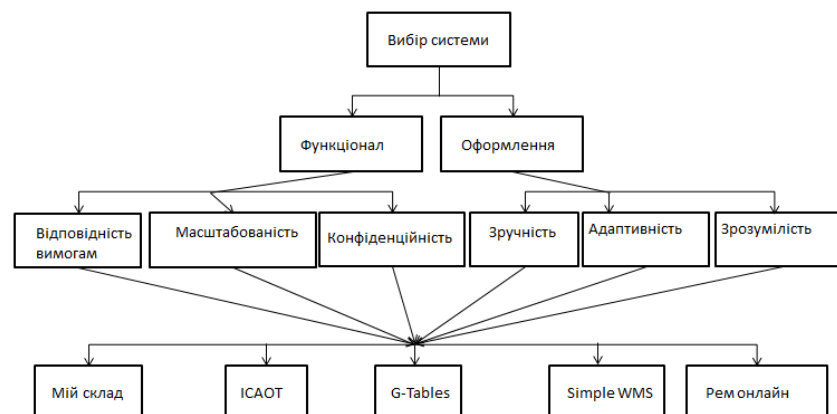


Рисунок 1.2 – Ієрархічна модель доцільності створення ІСАОТ

Після побудови ієрархічної моделі було визначено вектори пріоритетності, які надають перевагу елементам на певному рівні, або роблять їх важливішими і побудовано матриці попарних порівнянь.

Для 2-го рівня ієрархічної моделі було побудовано матрицю та виконано ряд обчислень як у додатку Б. В результаті було отримано власні вектори пріоритетності критеріїв як на рисунку 1.3.

| Функціонал | Оформлення | Власний вектор |
|------------|------------|----------------|
| 1 | 7 | 0,990 |
| 1/7 | 1 | 0,141 |

Рисунок 1.3 – Результати аналізу МПП 2-го рівня ієрархії

Далі було проаналізовано підкритерії функціоналу. На основі обчислень наведених у додатку Б було побудовано власні вектори пріоритетності підкритеріїв критерію «Функціонал» як на рисунку 1.4.

| Відповідність | Конфідентційність | Масштабованість | Власний вектор |
|---------------|-------------------|-----------------|----------------|
| 1 | 2 | 2 | 0,882 |
| 1/2 | 1 | 3 | 0,347 |
| 1/2 | 1/3 | 1 | 0,32 |

Рисунок 1.4 – Результати аналізу МПП 3-го рівня ієрархії за критерієм «Відповідність»

Власні вектори пріоритетності підкритеріїв критерію «Оформлення» отримано на основі обчислення, які подано в додатку Б і наведено на рисунку 1.5.

| Зручність | Зрозумілість | Адаптивність | Власний вектор |
|-----------|--------------|--------------|----------------|
| 1 | 2 | 3 | 0,826 |
| 1/2 | 1 | 3 | 0,520 |
| 1/3 | 1/3 | 1 | 0,218 |

Рисунок 1.5 – Результати аналізу МПП 3-го рівня ієрархії за критерієм «Зручність»

Аналогічним чином було проведено обчислення та побудовано власні вектори за іншими критеріями. Обчислення наведено у додатку Б, результати подано у додатку В.

В результаті аналізу усіх МПП за допомогою МАІ було отримано локальні вектори пріоритетності. Для обчислення глобальних пріоритетів побудовано матриці на основі локальних, як в додатку Б. Після цього було обчислено їх добуток з власними векторами предків, що знаходяться на 3 рівні ієрархії. На основі отриманих векторів було побудовано ще 1 матрицю, яку далі було перемножено із власним вектором 2 рівня ієрархії. На основі обчислень з додатку Б було отримано вектор глобальних пріоритетів як на рисунку 1.6.

| Система | Пріоритет |
|------------|-----------|
| Мій склад | 0,441 |
| ICOAT | 0,993 |
| G-Tables | 0,988 |
| Simple WMS | 0,347 |
| Рем онлайн | 0,775 |

Рисунок 1.6 – Вектори глобальних пріоритетів розглянутих вебсервісів

За допомогою МАІ було визначено, що система ІСАОТ задовільняє визначені вимоги найкращим чином.

Інформаційна система автоматизації обліку товарообігу забезпечить схожий до аналогів функціонал з рядом додаткових можливостей у місцях, які цього потребують. Також вона забезпечить наступні функції: інформаційна, комунікаційна, контрольна, прогнозна.

Суть інформаційної функції ІСАОТ полягає у забезпеченні керівників підприємства інформацією необхідною для прийняття подальших рішень про дозамовлення товарів [4]. Комунікаційна функція ІСАОТ повинна забезпечити зв'язок між рівнями управління та структурними підрозділами підприємств, на яких вона може бути впроваджена. Контрольна функція відповідає за оцінку результатів діяльності підприємства в цілому. Прогнозна

функція ІСАОТ повинна забезпечити перспективне планування і координування майбутніх рішень та дій.

Як підсумок потрібно відзначити, що автоматизація бізнес-процесів значно прискорює процеси обробки інформації, прийняття рішень на основі аналізу даних та підвищує ефективність роботи в цілому. Саме тому впровадження автоматизованої системи обліку товарообігу є доцільним.

В якості предметної області застосування було розглянуто бізнес-процеси [5], які відбуваються у діючому магазині косметики та побутових товарів. Результати дослідження подано з використанням діаграми потоків даних, яку зображено на рисунку 1.7 [6].

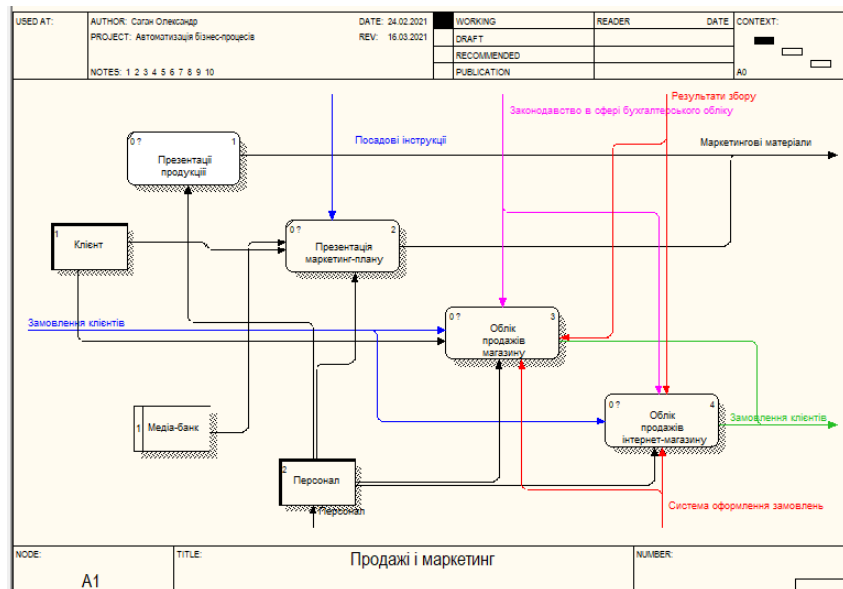


Рисунок 1.7 – Діаграма потоків даних підприємства

Дана діаграма демонструє, які процеси необхідно автоматизувати для підвищення ефективності діяльності підприємства.

Відповідно до визначених вимог потрібно спроектувати наступні функціональні можливості вебзастосунку:

- можливість реєстрації користувачів у ІСАОТ;
- можливість додавання записів про товари і покупки;
- можливість додавання записів про керівників, продавців та клієнтів;

- можливість додавання завдань для користувачів ІСАОТ;
- можливість переглядати свої завдання у ІСАОТ;
- можливість перегляду завдань для себе у ІСАОТ;
- можливість коментувати завдання у ІСАОТ.

З метою захисту інформації було сформовано вимоги до безпеки вебзастосунку:

- валідація інформації отриманої від користувача ІСАОТ;
- фільтрація отриманої інформації для запобігання SQL-ін'єкціям;
- парольна аутентифікація користувачів ІСАОТ;
- захист паролів користувачів ІСАОТ шляхом їх шифрування.

В майбутньому для підвищення рівня безпеки вебзастосунку планується розробити функціонал, що надаватиме доступ до авторизації у ІСАОТ лише визначеному ACL IP-адрес [7].

Відповідно до запроєктованих можливостей було сформовано вимоги до організації доступу користувачів до ІСАОТ:

- доступ через браузер на мобільних пристроях чи ПК;
- шифрування даних при їх передачі від користувача до сервера.

Користувацький інтерфейс ІСАОТ повинен відповідати наступним вимогам:

- коректне відображення контенту у різних браузерах;
- кодування сторінок повинне бути UTF-8;
- стилі відображення елементів інтерфейсу ІСАОТ повинні бути описані засобами CSS;
- генерація елементів вебінтерфейсу користувача відбувається засобами PHP або JS, залежно від наявності набору даних та функціональних можливостей, які забезпечує відповідний компонент вебінтерфейсу;
- архітектура вебзастосунку повинна дозволяти в майбутньому обмежити перелік IP-адрес певним ACL для яких буде доступною ІСАОТ.

Отже, ІСАОТ повинна автоматизувати функції додавання клієнтів, облік поставок, облік продажів, нарахування знижок, отримання інформації за прізвищем клієнта, перегляд інформації про покупки за певний період.

1.2 Обґрунтування використаних вебтехнологій реалізації ІСАОТ

Інформаційна система автоматизації обліку товарообігу має зберігати інформацію у структурованому вигляді, надавати можливість роботи з нею шляхом використання вебінтерфейсів. Це надає актуальності питанню вибору СУБД. Було розглянуто 3 найпопулярніші СУБД, які використовуються в побудові вебзастосунків [8].

Oracle 12c – продукт компанії Oracle. Дана СУБД націлена на хмарні сховища та може бути розміщена на кількох серверах. Це дозволяє оперативно керувати базами даних великих обсягів. Але зважаючи на високу вартість та вимоги до ресурсів, було зроблено висновок про недоцільність використання Oracle 12c для розробки вебзастосунку для автоматизації обліку товарообігу, оскільки, використання цієї СУБД є невиправдано ресурсозатратним і не забезпечить всіх переваг для ІСАОТ.

Далі було розглянуто СУБД MySQL, одну з найпопулярніших в галузі веброзробки. MySQL – це безкоштовний пакет програм від компанії Oracle, який постійно розвивається і удосконалює свій функціонал та безпеку. Недоліки даної СУБД не є критичними для ІСАОТ.

Ще одним популярним представником СУБД є Microsoft SQL Server. Це система керування базами даних, двигунець якої працює на хмарних та локальних серверах.

У випадку з Microsoft SQL Server основним недоліком є потреба у великій кількості ресурсів. У випадку ІСАОТ це недоцільно.

В результаті аналізу варіантів СУБД для розробки вебзастосунку для автоматизації обліку товарообігу було зроблено висновок про доцільність використання MySQL.

В процесі роботи ІСАОТ потрібно буде оперувати різними даними. Для цього потрібний вебінтерфейс. Інтерфейс наданий СУБД не зможе надати потрібних можливостей користувачам без спеціальної підготовки. Саме тому було прийнято рішення розробити вебінтерфейс для ІСАОТ.

Для розробки клієнтської частини було використано HTML та Bootstrap. JS в цьому випадку виступає засобом автоматизації створення елементів користувацького вебінтерфейсу ІСАОТ та виконує операції передачі даних від користувача засобами AJAX. Є багато альтернатив JS, таких як TypeScript, CoffeScript, Roy. Але результатом виконання скриптів написаних на цих мовах є код на JS [9].

Було прийнято рішення про недоцільність використання надбудов, що можуть викликати підвищення вимог до ресурсів клієнта інформаційної системи автоматизації обліку товарообігу.

В якості мови програмування серверної частини було обрано PHP. Вона забезпечить можливість реєстрації, авторизації у ІСАОТ та взаємодії з базою даних через вебінтерфейс розроблений для користувача. Ця мова володіє рядом переваг, а саме: швидка розробка, гнучкість та сумісність з різними ОС, масштабованість проєктів, легкість хостингу [10].

Зважаючи на переваги було зроблено висновок про доцільність використання PHP для реалізації серверної частини ІСАОТ.

ІСАОТ має бути максимально доступною, тому було прийнято рішення створити інтерфейс, який зможе підлаштовуватись під пристрій користувача ІСАОТ, засобами фреймворку Bootstrap [11].

1.3 Вибір середовища розробки ІСАОТ

Для розробки ІСАОТ необхідний набір інструментів: локальний вебсервер та редактор коду. В якості можливих варіантів вебсерверів було розглянуто Denwer та ХАМРР [12].

До переваг Denwer належать простота використання та дуже малий обсяг пам'яті, необхідної для встановлення.

ХАМРР – це повноцінний комплект програмного забезпечення, яке потрібне для локального розгортання ІСАОТ. До переваг належать простота налаштування, безкоштовне поширення та наявність пакетів, що збільшують можливості вебсервера. Для розробки ІСАОТ саме цей локальний сервер підійшов найкращим чином, оскільки, він включає пакети для роботи з РНР та MySQL, які необхідні для програмної реалізації.

Розробка ІСАОТ буде відбуватись на локальному сервері ХАМРР. Для його інсталяції було завантажено виконуваний файл з офіційного сайту.

В якості можливих ІДЕ було розглянуто PHPStorm та VisualStudioCode. В результаті порівняння було виявлено ряд суттєвих переваг VisualStudioCode, а саме: безкоштовне поширення, нижчі системні вимоги, велика кількість безкоштовних розширень, що суттєво збільшують функціонал. На початкових етапах розробки ІСАОТ розмір проєкту буде досить малим і переваги PHPStorm не дадуть настільки великого приросту ефективності як у випадку з великими проєктами. Тому в якості ІДЕ було обрано VisualStudioCode [13] і завантажено з офіційного сайту. Після успішної інсталяції усіх інструментів було розглянуто документацію стосовно їх використання.

РОЗДІЛ 2. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ, ВСТАНОВЛЕННЯ ТА ТЕСТУВАННЯ ІСАОТ

2.1 Вибір оптимального методу реалізації ІСАОТ

Серед можливих варіантів розробки ІСАОТ [14] було обрано метод ручної [15] розробки. Він не має недоліків в формі обмеженості стандартним шаблоном, якщо порівнювати з методами, які передбачають використання програмного забезпечення та SaaS-платформами.

Використання систем керування контентом є недоцільним через малу кількість функціоналу вебзастосунку, яка б вирішувалась стандартними засобами цих систем. Можливість використання фреймворків також було відхилено для економії апаратних ресурсів, що використовуватимуться для впровадження вебзастосунку для автоматизації обліку товарообігу і для того, щоб не створювати залежність інформаційної системи автоматизації обліку товарообігу від конкретної реалізації патернів, які використовуються у певному фреймворку.

2.2 Життєвий цикл ІСАОТ

Інформаційна система автоматизації обліку товарообігу матиме життєвий цикл [16] аналогічний до життєвих циклів схожих вебсервісів. Він включатиме в себе такі етапи:

- розробка концепції інформаційної системи автоматизації обліку товарообігу;
- розробка технічного завдання для інформаційної системи автоматизації обліку товарообігу;
- проєктування інформаційної системи автоматизації обліку товарообігу;
- реалізація інформаційної системи автоматизації обліку товарообігу;

- впровадження ІСАОТ в експлуатацію;
- супровід інформаційної системи автоматизації обліку товарообігу.

Для підвищення продуктивності роботи підприємств шляхом автоматизації бізнес-процесів необхідним етапом є розробка концепції ІСАОТ. Він включає: аналіз предметної області. Далі потрібно виділити інформаційні сутності, що матимуть набори атрибутів, які визначатимуть властивості вебінтерфейсів.

Наступним етапом є розробка технічного завдання. Метою його створення є визначення основних вимог до ІСАОТ та процесу її розробки. Також на цьому етапі можуть розроблятися технічні завдання до окремих вебінтерфейсів ІСАОТ.

На етапі проєктування інформаційної системи автоматизації обліку товарообігу створюється інформаційна база, формуються вимоги до інформаційних масивів, технічних засобів веброзробки та вказуються характеристики вебзастосунку. В результаті проходження цього етапу отримується проєктна документація, що визначає завдання ІСАОТ, алгоритм розробки вебзастосунку, визначає організаційні моменти та набір ПЗ для вирішення завдань інформаційної системи автоматизації обліку товарообігу.

На етапі реалізації ІСАОТ відбувається розробка вебінтерфейсів та серверної частини вебзастосунку для обліку складських операцій, що відповідає положенням визначеним на попередніх етапах і фактично є реалізацією вебзастосунку описаного в технічній документації. В результаті проходження цього етапу отримується інформаційна система автоматизації обліку товарообігу готова до впровадження в експлуатацію.

На етапі впровадження інформаційної системи автоматизації обліку товарообігу в експлуатацію відбувається дослідження вебзастосунку на наявність сценаріїв, що призводять до збоїв в роботі системи. Також відбувається навчання персоналу. Після успішного виконання всіх умов ІСАОТ передається в експлуатацію замовнику.

Супровід вебзастосунку для автоматизації обліку товарообігу відбувається на підставах, що визначаються у відповідних пунктах договору та зобов'язань з боку розробника ІСАОТ. На цьому етапі здійснюється обслуговування ІСАОТ та усуваються недоліки вебінтерфейсів, якщо такі буде виявлено при збереженні стану ІСАОТ, при якому вона була передана в експлуатацію.

2.3 Формалізація вимог

В результаті постановки завдання проєкту, було визначено необхідність проєктування та розробки інформаційної системи, яка дозволить автоматизувати функції додавання клієнтів, облік поставок, облік продажів, нарахування знижок, отримання інформації за прізвищем клієнта, перегляд інформації про покупки за певний період. Користувачами даної системи повинні бути працівники магазину, його власник, клієнти. На рисунку 2.1 подано бізнес-процеси офлайн покупки в нотації IDEF3 [17], які будуть автоматизовані.

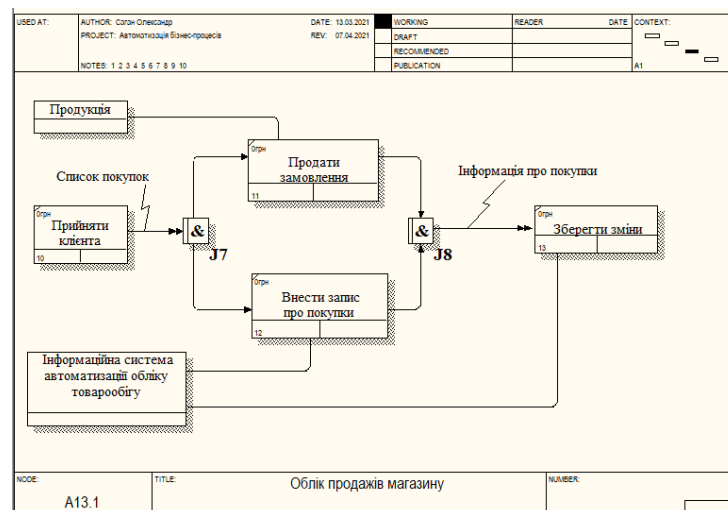


Рисунок 2.1 – Діаграма бізнес-процесу офлайн покупки

Також варто відзначити, що ІСАОТ може використовуватися для ведення обліку товарообігу у випадку виконання операцій продажу шляхом

використання інтернет-магазинів. А в майбутньому при актуальності такого функціоналу буде можливо провести інтеграцію з зовнішніми сервісами.

На рисунку 2.2 подано бізнес-процеси онлайн покупки в нотації IDEF3.

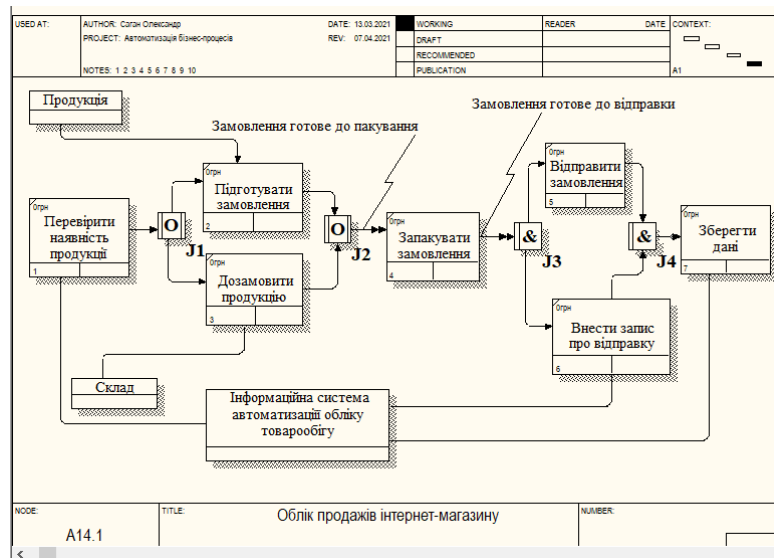


Рисунок 2.2 – Діаграма бізнес-процесу онлайн покупки

Основні елементи узагальненої умовно замкненої моделі подано на рисунку 2.3.

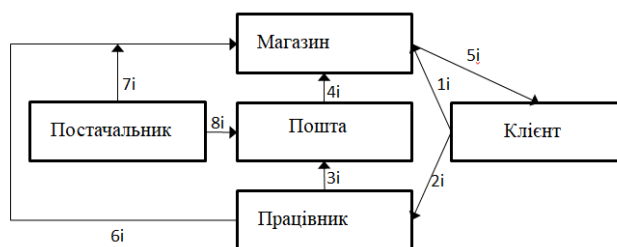


Рисунок 2.3 – Узагальнена умовно замкнута модель інформаційної системи

Зміст інформаційних управляючих зв'язків узагальненої умовно замкненої моделі:

- 1i – внесення персональних даних клієнта;

- 2і – клієнт звернувся до працівника магазину з метою придбання товарів;
- 3і – працівник забрав замовлення з пошти;
- 4і – товари з пошти прибули в магазин;
- 5і – магазин надав замовлені товари;
- 6і – за наявності продукції в магазині працівнику не потрібно звертатись до пошти;
- 7і – можлива адресна доставка від постачальника в магазин;
- 8і – замовлення прийшло від постачальника на пошту.

2.4 Побудова дерева функцій та проєктування структури ІСАОТ

Основним завданням магазину є оперативне оновлення інформації про товари і швидке обслуговування клієнтів. Для цього працівник фірми повинен постійно вести облік продукції, клієнтів, закупок тощо. Крім цього потрібно аналізувати числові показники. Дерево функцій системи подано в додатку Г.

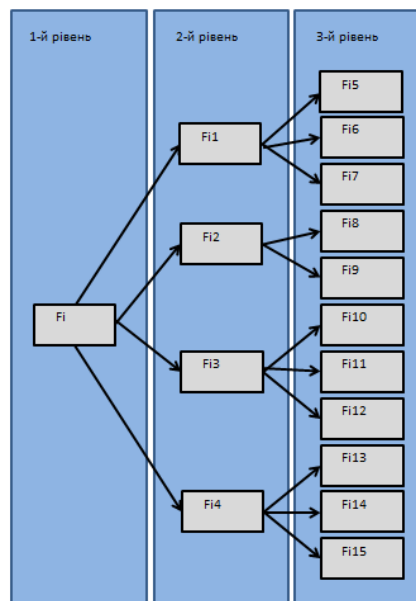


Рисунок 2.4 – Дерево функцій інформаційної системи

Fi1 – облік товарів:

- Fi5 – облік надходжень;
- Fi6 – облік наявності;
- Fi7 – облік продажів.
- Fi2 – інформація про доходи:
- Fi8 – сумарна вартість замовлених товарів;
- Fi9 – доходи за певний період.
- Fi3 – облік клієнтів:
- Fi10 – нові клієнти (ще не зареєстровані);
- Fi11 – постійні клієнти (зареєстровані);
- Fi12 – отримані клієнтами замовлення.
- Fi4 – облік товарообігу:
- Fi13 – замовлені (ще в дорозі);
- Fi14 – на складі (очікують покупця);
- Fi15 – продано.

Відповідно до визначеного функціоналу було спроектовано структуру інформаційної системи, яка подана на рисунку 2.5.

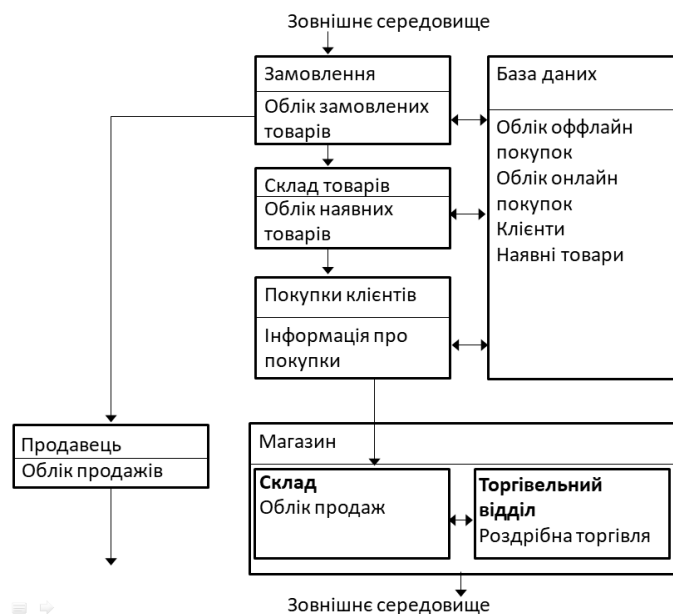


Рисунок 2.5 – Схема компонентів ІСАОТ

2.5 Моделювання архітектури ІСАОТ

Концепція інформаційної системи автоматизації обліку товарообігу передбачає централізацію збереження даних та децентралізацію доступу до них. Було зроблено висновок про доцільність побудови ІСАОТ в стилі трирівневої клієнт-серверної архітектури [18].

Її використання обґрунтовується необхідністю розподілу системи на рівні, кожен з яких матиме своє функціональне призначення. Цей вид архітектури дозволить знизити навантаження на серверну частину ПЗ ІСАОТ за рахунок перенесення частини навантаження на клієнтські місця і в той же час не вимагає великої кількості ресурсів від кінцевих користувачів. Також така архітектура забезпечить зниження витрат на підтримку ІСАОТ [19], оскільки, дає можливість підтримувати усі компоненти централізовано (див. рисунок 2.6).

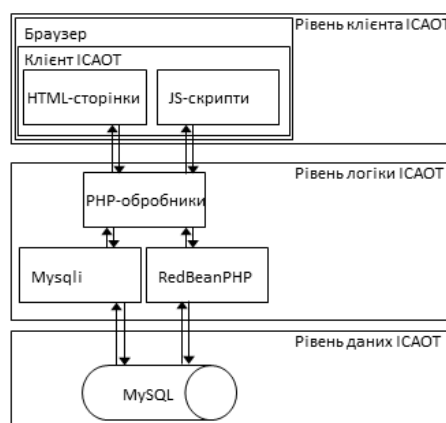


Рисунок 2.6 – Архітектура ІСАОТ

Рівень клієнта ІСАОТ представлений набором HTML-шаблонів, що формуються шляхом виконання PHP-скриптів, які підключають необхідні програмні компоненти: обробники форм, ПЗ RedBeanPHP, яке дозволяє взаємодіяти з БД шляхом використання ORM [20] та повторювані DOM-елементи вебінтерфейсу: футер, меню. Ці шаблони викликають JS-скрипти, що відповідають за побудову DOM-елементів користувацького інтерфейсу ІСАОТ [21].

Шляхом використання описаної клієнтської частини ІСАОТ браузер отримує можливість надіслати дані далі – на рівень логіки. Рівень логіки ІСАОТ представлений набором скриптів та функцій, що забезпечують валідацію даних та їх передачу ORM-інтерфейсам, зокладані ПЗ RedBeanPHP або ж модулю `mysqli`. У випадку дотримання всіх необхідних умов програмні компоненти повернуть відповідь і обробник продовжить дії з нею. Функціонал ІСАОТ передбачає сценарій передачі даних з PHP-скриптів у JS-скрипт, або безпосередньо у браузер, де буде відображено засобами клієнтського рівня ІСАОТ.

2.6 Проєктування поведінки ІСАОТ

Перелік основних компонентів та їх взаємодію у ІСАОТ подано в таблиці у додатку Д та на рисунку в додатку Е.

Кожен компонент ІСАОТ реалізує конкретну функцію шляхом виклику відповідних процедур з PHP-скриптів. Частина функції, що стосуються підсистеми для створення завдань зібрана у файл `utils.php`. Функціонал, що відноситься до інших форм представлений структурою скриптів, що знаходяться у розділах проєкту, які представляють функціональні модулі. Такий підхід не ускладнює структуру ІСАОТ великою кількістю залежностей, дає можливість вдосконалити програмні коди шляхом виявлення неочевидних повторень коду та винесення у нові функції чи модулі. Також є можливість розподілу на складну ієрархію класів при подальшому нарощенні функціоналу та потребі інкапсуляції окремих компонентів.

2.7 Проєктування бази даних ІСАОТ

В результаті аналізу предметної області [22] застосування інформаційної системи автоматизації обліку товарообігу було виділено такі

інформаційні сутності: «Керівник», «Продавець», «Покупець», «Замовлення», «Товари», «Звіт про продані товари», «Магазин», «Рядок замовлення» «Користувач», «Завдання», «Коментар», «Пріоритет», «Статус», «Контакт». Далі було повторно проаналізовано предметну область для встановлення зв'язків між сутностями і проведено подальші дії процесу проектування.

В результаті аналізу інформації, якою оперуватиме інформаційна система автоматизації обліку товарообігу було виділено зв'язки [23]. Між сутностями «Керівник» і «Магазин» було встановлено зв'язок типу «один до багатьох». В рамках поставленої задачі цей зв'язок не має великого значення, оскільки, впровадження системи передбачене для одного магазину, але в майбутньому він дає можливість додавання нових магазинів у випадку їх відкриття. Таким чином виділення керівника в окрему сутність значно підвищує масштабованість проекту в цілому.

У магазині повинен працювати продавець. Наявність відповідної сутності в ІСАОТ дає можливість додавання кількох продавців, якщо у магазині працюватиме більше одного. Тому між сутностями «Магазин» і «Продавець» було встановлено зв'язок аналогічного типу, а саме: «один до багатьох». Суть ІСАОТ полягає в автоматизації основних бізнес-процесів, одним з яких є облік товарообігу. Саме тому важливо виокремити сутність «Наявні товари». Факт наявності означає їх знаходження у магазині, тому було встановлено аналогічний попереднім зв'язок «Наявні товари»-«Магазин». Концептуальну модель бази даних подано на рисунку 2.7 [24].

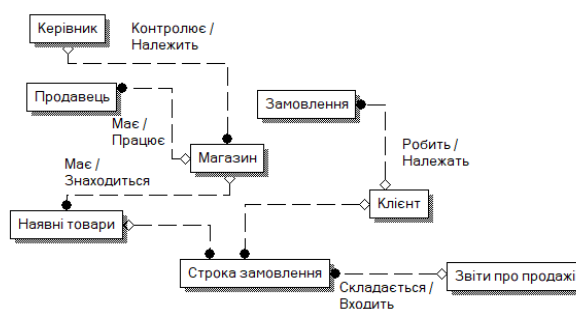


Рисунок 2.7 – Концептуальна модель бази даних

Далі було встановлено зв'язок «багато до багатьох» між сутностями «Клієнт» і «Наявні товари». Він реалізується через проміжну сутність «Рядок замовлення». Було використано неідентифікуючі зв'язки, бо система має надавати можливість переглядати покупки певних покупців і кількість покупок певного товару.

Після виділення сутностей було визначено необхідні ключі, які б давали можливість однозначно ідентифікувати кожну з них. Для кожної сутності було визначено унікальний числовий ідентифікатор, який не повторюється у представників однієї сутності і не пов'язаний з іншими. Далі було визначено первинні ключі та здійснено їх міграцію для реалізації запроєктованих зв'язків у БД ІСАОТ, як на рисунку 2.8 [25].

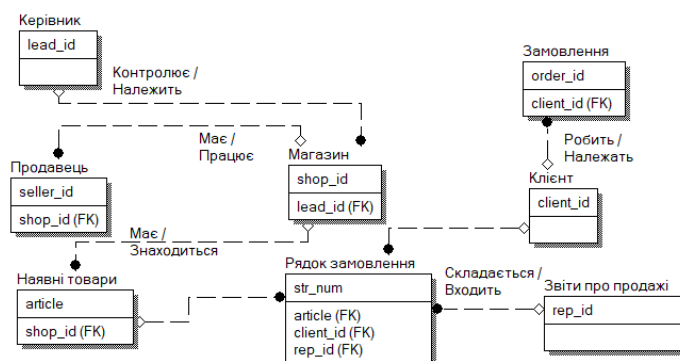


Рисунок 2.8 – Міграція ключів

Після виконання міграції ключів було визначено набір атрибутів для кожної сутності БД ІСАОТ, які забезпечать збереження всіх необхідних даних для обліку товарообігу. Набори атрибутів подано у таблицях додатку Ж.

Після міграції ключів та визначення наборів основних атрибутів було здійснено глибинний аналіз сутностей та їх атрибутів і побудовано логічну моделі реляційної бази даних. Обмеження атрибутів наведено в таблицях додатку З.

Функціонал ІСАОТ, що стосується постановки та роботи з завданнями, а також реєстрацію користувачів було реалізовано засобами ORM наданого ПЗ RedBenaPHP, тому окреме проектування бази даних не проводилось.

2.8 Проектування користувацького вебінтерфейсу ІСАОТ

Відповідно до визначеного переліку ролей користувачів ІСАОТ необхідно розділити функціонал на декілька ролей.

Функції керівника:

- робота з новими керівниками (додавання, редагування, видалення);
- робота з продавцями (додавання, редагування, видалення);
- робота з клієнтами (додавання, редагування, видалення);
- робота з магазинами (додавання, редагування, видалення);
- робота з товарами(додавання, редагування, видалення).

Функції продавця:

- робота з клієнтами (додавання, редагування, видалення);
- робота з товарами(додавання, редагування, видалення).

Функції зареєстрованого користувача:

- додавання замовлень;
- перегляд інформації про свої покупки.

Функції неавторизованого користувача:

- реєстрація;
- авторизація.

Крім того всі типи користувачів, крім незареєстрованих, мають доступ до роботи з завданнями як зображено на рисунку 2.9 [25].

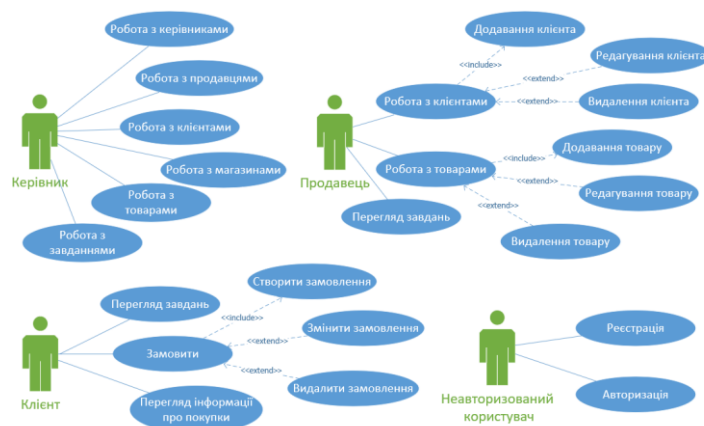


Рисунок 2.9 Діаграма варіантів використання ІСАОТ

Для реалізації даного функціоналу було спроектовано ряд форм, який включає: «Керівник», «Продавець», «Клієнт», «Товар», «Магазин», «Покупки», «Замовлення», «Реєстрація», «Авторизація».

Форма «Керівник» повинна надавати можливість додавати, видаляти і оновлювати інформацію про керівників, а саме: ім'я, телефон, електронна, пошта.

Форми «Продавець», «Клієнт», «Магазин» повинні надавати аналогічний функціонал для роботи з інформацією про відповідні сутності.

Форма «Товар», крім можливості додавання, редагування і видалення одного екземпляру повинна забезпечити внесення групи елементів, в той час як попередні форми такої можливості не потребують.

Форма «Покупки» теж буде повинна забезпечувати можливість внесення інформації, але відрізняється тим, що не повинна надавати можливості перегляду, редагування та видалення.

Форма «Реєстрація» повинна надавати можливість реєстрації в системі, а форма «Авторизація» забезпечує вхід в систему зареєстрованих користувачів.

2.9 Реалізація основних функцій ІСАОТ

Для реалізації функціоналу було розроблено базу даних, яка відповідає фізичній моделі, зображеній на рисунку 2.10

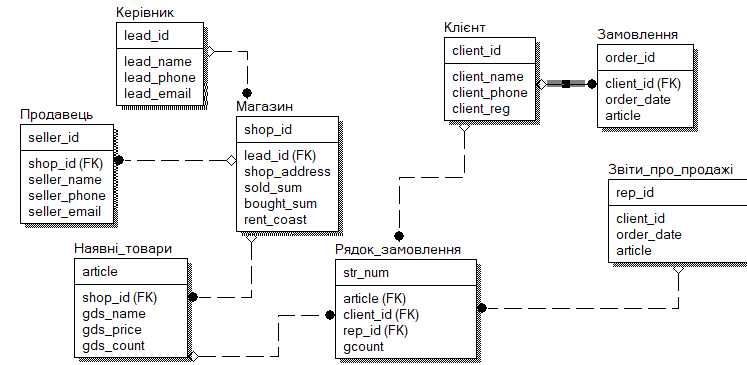


Рисунок 2.10 – Фізична модель бази даних

Відповідно до фізичної моделі було створено запити на створення таблиць як подано в додатку И, та ЗП [26], які дозволяють додавати і змінювати та видаляти дані виконуючи додаткові маніпуляції з вхідними параметрами.

Варто відзначити, що змінні екрануються, серверу передається прототип запиту та значення параметрів. Після чого виконується безпечний запит захищений від SQL-ін'єкцій .

Для сутності «Керівник» було створено 3 ЗП:

- create_lead;
- update_lead;
- delete_lead.

Збережувальна процедура create_lead приймає 3 параметри: ім'я керівника, електронну пошту і телефон. Далі перевіряється наявність керівника, у якого хоч 1 з полів буде співпадати з новим. Якщо збіг знайдено, то запис не буде додано. Якщо ж збігів не виявлено, то буде підготовано прототип запиту, передано його на сервер і окремо передано екрановані параметри, після чого запит буде виконано [28].

Збережувальна процедура `update_lead` приймає 4 параметри: ідентифікатор запису, ім'я користувача, телефон та електронну пошту. Після цього перевіряється можливість зміни даних керівника, на нові, тобто шукаються збіги з іншими записами. Якщо збіг знайдено, то запис не буде оновлено, в іншому випадку на серверу окремо передаються прототип запису та параметри. Після екранування параметрів вони підставляються в запит і виконується оновлення записів.

Збережувальна процедура `delete_lead` приймає 1 параметр: ідентифікатор запису і виконує запит на видалення запису з відповідним ідентифікатором у таблиці керівників.

Для роботи з іншими сутностями було розроблено набір ЗП, які забезпечують виконання аналогічних дій з іншими наборами даних.

2.10 Реалізація користувацького вебінтерфейсу ІСАОТ

Для розмежування прав доступу користувачів було реалізовано реєстрацію, яка потребує мінімум персональної інформації від користувача.

Після реєстрації користувачу потрібно ввести у форму авторизації логін пароль вказані при реєстрації. Після чого виконається запит до БД з метою перевірки наявності відповідного облікового запису.

Вкладка «Додати» надає можливість додавати нових керівників, для цього потрібно заповнити форму, яка зображена на рисунку 2.11 і надіслати дані натисканням кнопки «Додати запис».

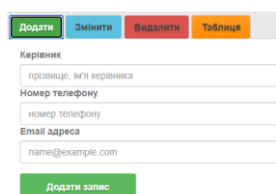


Рисунок 2.11 – Форма додавання керівників

Після успішної авторизації користувачу надається доступ до функцій ІСАОТ, які передбачає роботу з керівниками, магазинами, продавцями, клієнтами, товарами та підсумковими даними, які вичисляються на основі значень таблиць бази даних.

Віджет роботи з керівниками включає в себе 4 вкладки: «Додати», «Змінити», «Видалити» і «Таблиця». Кожна вкладка надає форму для виконання певної дії з базою даних.

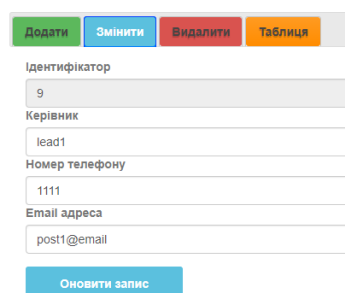
Після натискання кнопки «Додати запис» дані буде надіслано скрипту, який обробляє значення полів форми і вносить їх у БД. При відправці даних оновлення сторінки не відбудеться завдяки використанню технології AJAX [29] при надсиланні форми, активною вкладкою автоматично стане «Таблиця», де користувач зразу може переглянути внесені дані з поміткою «Новий» як на рисунку 2.12.



| id | Керівник | Телефон | Email |
|-------|----------|---------|-------------|
| NOVYI | lead1 | 1111 | post1@email |

Рисунок 2.12 – Табличне подання записів БД

Після оновлення сторінки помітка «Новий» зникає і з’являються кнопки «Змінити» і «Видалити». При натисканні кнопки «Змінити» автоматично буде змінено активну вкладку на «Змінити» і заповнено форму даними відповідного запису як показано на рисунку 2.13.



Додати Змінити Видалити Таблица

Ідентифікатор
9

Керівник
lead1

Номер телефону
1111

Email адреса
post1@email

Оновити запис

Рисунок 2.13 - Форма зміни даних керівника

Після зміни даних і натискання кнопки «Оновити запис» виконується асинхронний обробник події надсилання форми, значення полів передаються скрипту, що їх обробляє і викликає відповідну зберезувальну процедуру. Оновлення сторінки не відбувається і дані в таблиці оновлюються засобами JS. Для візуального виділення рядку таблиці задається інший фон як на рисунку 2.14, поки користувач не наведе курсор, після чого фон буде змінено на попередній.

| id | Керівник | Телефон | Email | | |
|----|-------------|---------|-------------|---------|----------|
| 9 | lead_edited | 1111 | post1@email | Змінити | Видалити |

Рисунок 2.14 – Результат зміни запису

Для видалення записів потрібно натиснути на кнопку «Видалити», після чого виконається обробник відповідної події активною стане вкладка «Видалити» і форма заповниться автоматично.

Після натискання кнопки «Видалити запис» виконається асинхронний обробник події надсилання форми обробник викличе зберезувальну процедуру, яка видалить запис.

Інтерфейси для роботи з іншими таблицями працюють аналогічним чином, але відрізняються наборами полів та логікою зберезувальних процедур.

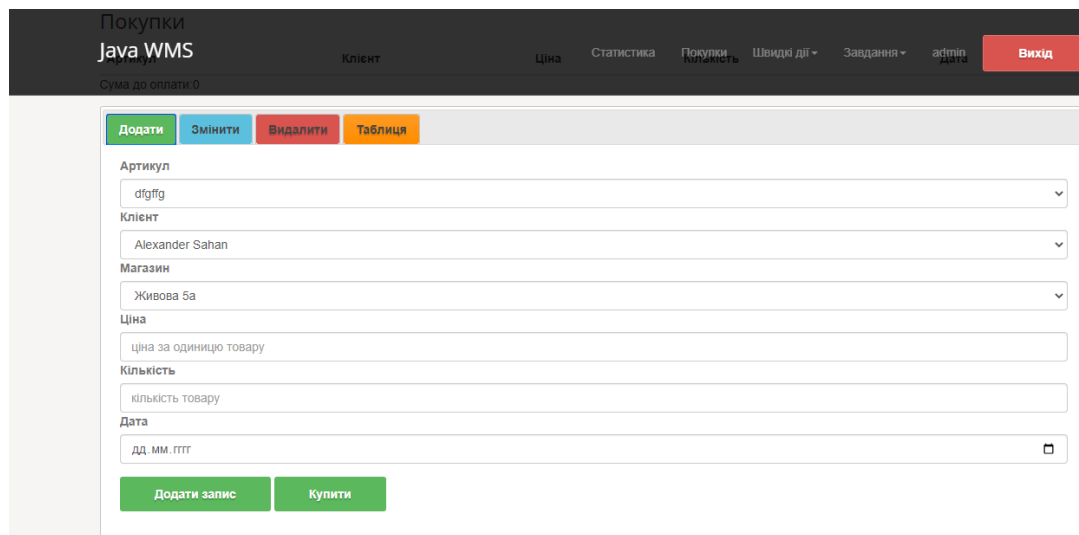
Віджет для роботи з магазинами складається з вкладок «Додати», «Змінити», «Видалити» і «Таблиця».

Віджет для роботи з продавцями включає такі вкладки: «Додати», «Змінити», «Видалити» і «Таблиця».

Віджет для роботи з клієнтами і товарами в свою структуру включають вкладки аналогічні до попередніх віджетів.

Віджет для здійснення покупок відрізняється своїм функціоналом, оскільки дає можливість додавати групу записів. Це було реалізовано за

допомогою накопичення екземплярів сутності «Рядок замовлення». Сам процес накопичення відбувається при заповненні форми «Додати» і натисканні кнопки «Додати запис». При цьому всі одиниці товару та їх сумарну вартість буде відображено в таблиці як на рисунку 2.15.



The screenshot shows a web application interface for 'Покупки Java WMS'. At the top, there is a navigation bar with links for 'Клієнт', 'Ціна', 'Статистика', 'Покупки', 'Швидкі дії', 'Завдання', 'admin', and a red 'Вихід' button. Below the navigation bar, there is a sub-header with 'Додати', 'Змінити', 'Видалити', and 'Таблиця' buttons. The main form contains several input fields: 'Артикул' (with a dropdown menu showing 'dfgffg'), 'Клієнт' (with a dropdown menu showing 'Alexander Sahan'), 'Магазин' (with a dropdown menu showing 'Живова Ба'), 'Ціна' (with a text input field containing 'ціна за одиницю товару'), 'Кількість' (with a text input field containing 'кількість товару'), and 'Дата' (with a date picker showing 'дд. мм. гgg'). At the bottom of the form, there are two green buttons: 'Додати запис' and 'Купити'.

Рисунок 2.15 Віджет для оформлення покупок

Додавання групи записів у БД відбувається після натискання кнопки «Купити», яке викликає AJAX-процедуру, що вносить нові записи у БД. Всі віджети «вкладки» згруповано у віджет «акордеон».

Для коректного відображення всіх елементів було використано каскадні таблиці стилів, а для його адаптації під різні розміри екранів було додано медіа-запити.

2.11 Програмна реалізація основних структурних елементів ІСАОТ

Кожна сторінка ІСАОТ забезпечує виконання певної функції. При переході на певну сторінку ІСАОТ, якщо користувач авторизований – буде відображено елементи користувацького вебінтерфейсу. У протилежному випадку буде виведено форму авторизації у ІСАОТ. Крім того сторінка містить підключення до всіх необхідних компонентів, що забезпечують

підключення до бази даних, розширення набору доступних програмних інструментів, виведення компонентів користувацького інтерфейсу.

Сторінка реєстрації у ІСАОТ забезпечує доступ користувачу до реєстрації в системі. Аналогічно до індексної сторінки вона містить підключення компонентів системи: скрипт побудови форми та її обробник. При завантаженні сторінки реєстрації у ІСАОТ виконується PHP-скрипт, що підключає файл автозавантаження. У свою чергу цей файл підключає бібліотеку для роботи з базою даних, файл підключення до бази даних ІСАОТ, скрипт з набором функцій:

```
require_once 'rb-mysql.php';
require_once 'database.php';
require_once 'utils.php';
```

Підключення до бази даних ІСАОТ відбувається шляхом використання методів модуля mysqlі або ПЗ RedBeanPHP [30]:

```
R::setup( 'mysql:host=host;dbname=name', 'user', 'password' );
```

Також при завантаженні сторінки відбувається перевірка наявності потрібних змінних у суперглобальному масиві \$_POST [31]. Якщо значення змінних встановлено – відбудеться виклик обробника відповідної форми. У випадку зі сторінкою реєстрації відбудеться перевірка чи натискалась кнопка надсилання форми:

```
if( isset($_POST['signup']) ){signup();}
```

Процедуру авторизації у ІСАОТ винесено в окремий скрипт. Спочатку відбувається ініціалізація змінної, у яку будуть записуватись повідомлення про помилки:

```
$errors = array();
```

Далі відбувається перевірка чи заповнені поля форми авторизації у ІСАОТ, якщо ні, то у масив помилок буде додано відповідне повідомлення:

```
if( trim($_POST['login']) == '' ){ $errors[] = 'Login is empty!'; }
```

Поля для введення пошти та паролю на формі реєстрації у ІСАОТ перевіряються аналогічним чином. Якщо якийсь з полів не заповнено – буде отримано повідомлення про це.

Потім виконуються запити до бази користувачів ІСАОТ, що перевіряють наявність користувачів зі внесеним логіном чи поштою шляхом визначення кількості записів з введеними параметрами для логіну [32]:

```
if( R::count('users', "login = ?", array($_POST['login'])) > 0 ) { $errors[] = 'Login exists!'; }
```

Та для адреси електронної пошти:

```
if( R::count('users', "email = ?", array($_POST['email'])) > 0 ) { $errors[] = 'Email exists!'; }
```

Далі відбувається перевірка на наявність помилок. Якщо масив помилок пустий, то відбудеться ініціалізація змінної, що виступає в ролі об'єкта представлення бази даних: таблиці користувачів ІСАОТ [33]:

```
$user = R::dispense('users');
```

Після ініціалізації відбувається присвоєння властивостей, у випадку з формою реєстрації у ІСАОТ буде змінено властивості, що відповідають за

логін, пошту та пароль. Замість пароля зберігається його хеш. Це захищає облікові дані користувачів від розголошення у випадку витоку даних:

```
$user->login = $_POST['login'];
$user->email = $_POST['email'];
$user->password = password_hash($_POST['password']);
```

Після ініціалізації об'єкта ORM БД та визначення його властивостей виконується збереження даних. Воно включає в себе екранування даних отриманих від користувача ІСАОТ, виконання ЗП з підготовчими виразами засобами ПЗ RedBeanPHP [34]:

```
R::store($user);
```

Побудова частини форми ІСАОТ для надсилання даних відбувається шляхом виконання JS-функції, яка приймає один вхідний аргумент – селектор блоку у який буде поміщено DOM-елемент користувацького інтерфейсу:

```
createSignUpForm('.app');
```

Створення форми реєстрації у ІСАОТ відбувається шляхом виклику функції createSignUpForm(). У тілі функції відбувається ініціалізація змінної, якій присвоюється результат виконання функції createForm() з набором параметрів, які визначають кількість та тип полів форми ІСАОТ. Ці параметри включають масив типу «ключ-значення», що описує властивості форми ІСАОТ, масив типів полів форми, масив імен полів форми, масив значень або підказок полів, масив стилів елементів форми та масив написів, що виводяться окремим елементом над кожним з полів форми як показано у додатку Б. Після цього відбувається ініціалізація змінної, якій присвоюється DOM-елемент обраний за отриманим параметром функції:

```
let app = document.querySelector(selector);
```

Далі у обраний елемент відбувається вставка розмітки сформованої форми інформаційної системи автоматизації обліку товарообігу:

```
app.appendChild(form);
```

Код процедури побудови форми реєстрації у ІСАОТ подано в додатку К. Розмітка форми отримується в результаті виконання функції `createForm()`.

Спочатку відбувається ініціалізація змінної, якій присвоюється відповідний DOM-елемент і встановлюються значення певних атрибутів:

```
let form = document.createElement("form");  
form.method = form_conf.method;  
form.action = form_conf.action;
```

Далі ініціалізується змінна, що міститиме блок з кнопками форми :

```
let buttons = document.createElement("div");
```

Після цього починається цикл, що проходить по масиву значень полів, визначених у функції побудови відповідної форми ІСАОТ. На початку циклу створюється контейнер для зберігання елемента форми та надпису, якщо такий буде:

```
let div = document.createElement('div');  
div.setAttribute('class', 'mb-3');  
buttons.setAttribute('class', 'mb-3');
```

Далі відбувається перевірка чи не пусте значення напису на формі ІСАОТ. Якщо воно задане, то буде створено елемент «Label» і додано його у контейнер для елемента відповідної форми ІСАОТ:

```
if( labels[i] != '' ){div.append(createLabel(labels[i]));}
```

Потім відбувається перевірка значення типу елемента форми ІСАОТ. Залежно від цього значення буде виконано певну функцію.

Якщо значення рівне «textarea», то відбудеться виклик функції для створення даного елемента з передачею параметрів необхідних для оформлення елемента інтерфейсу користувача ІСАОТ:

```
div.append(createTextArea(names[i], placeholders[i], css[i]));
form.append(div);
```

У випадку з випадаючим списком логіка аналогічна за винятком того, що замість значення передається URL для отримання даних засобами fetch, якими буде наповнено випадаючий список, що використовується на формах ІСАОТ та може містити різні набори даних:

```
div.append(createSelect(names[i], placeholders[i], css[i]));
form.append(div);
```

Якщо ж передано тип, що передбачає створення кнопки для форми ІСАОТ, то відбудеться виклик функції для її створення:

```
buttons.append(createButton(types[i], names[i], placeholders[i],
css[i]));
```

Останнім з можливих елементів форми ІСАОТ є поле для введення даних, що мають відношення до визначених інформаційних об'єктів-сутностей:

```
div.append(createInput(names[i], types[i], placeholders[i],
css[i]));
form.append(div);
```

Після ініціалізації всіх елементів та додавання їх у відповідні блоки відбувається повернення форми та її відображення у відповідному розділі. Побудова елемента напису на формі ІСАОТ відбувається шляхом виклику функції `createLabel()`. Вона приймає єдиний вхідний параметр та ініціалізує змінну, у яку поміщається DOM-елемент. Також у функції задаються його властивості – значення та клас:

```
let label = document.createElement('label');
label.append(val);
label.setAttribute('class', 'form-label');
```

Робота функцій для створення інших елементів форми ІСАОТ аналогічна до створення напису. Відмінність полягає лише у наборах властивостей DOM-елементів. Крім того, у випадку з випадаючим списком виконується запит для отримання значень списку засобами JS [35]:

```
fetch(url, {method: 'GET', headers: {'Content-Type':
'application/json'}})
```

Якщо в результаті запиту отримується масив значень, то формується масив ключів JSON-елементів [36]:

```
let keys = [];
for (let key in options[0]) {keys.push(key);}
```

Цей масив потрібний для подальшої побудови випадаючого списку.

Циклічний перебір елементів формує опції списку та ітеративно заповнює їх значеннями:


```
option.value = options[i][keys[0]];
option.append(options[i][keys[1]]);
```

На кожній ітерації новий елемент списку додається у елемент «випадаючий список»:

```
select.appendChild(option);
```

Принцип роботи інших сторінок ІСАОТ, що передбачають заповнення форм аналогічний описаному і відрізняється лише наборами даних. Варто відзначити, що доступ до функціоналу ІСАОТ користувач отримає лише після авторизації. Вона реалізується шляхом додавання об'єкту-користувача, що містить всі його властивості у суперглобальний масив сесії [37]:

```
if( $user ){password_verify($_POST['password'],
$user['password']) ? $_SESSION['logged_user'] = $user :
$errors[] = 'Enter password again!';}
```

Додавання, редагування та видалення записів реалізовані аналогічним до додавання користувача чином. При завантаженні сторінки відбувається підключення необхідних скриптів та компонентів. Виконується перевірка авторизації користувача у ІСАОТ. У випадку якщо користувач не авторизований відбудеться виклик процедури побудови форми авторизації. Якщо ж користувач авторизований, то буде відображено контент відповідно до того, в який розділ ІСАОТ відбувся запит.

Виведення контактів та завдань у ІСАОТ відбувається засобами PHP. Для перевірки чи авторизований користувач у ІСАОТ відбувається пошук даних у суперглобальному масиві \$_SESSION:

```
if( isset($_SESSION['logged_user']) ):
```

Якщо користувач авторизований у інформаційній системі автоматизації обліку товарообігу, то виконується запит до БД, який повертає вибірку з таблиці контактів. Це реалізовано шляхом використання програмного забезпечення RedBeanPHP:

```
$contacts = R::getAll('SELECT * FROM contacts');
```

Після чого в циклі відбувається виведення HTML-розмітки та даних отриманих в результаті запиту:

```
<?php foreach($contacts as &$contact): ?>
    <tr > ... </td></tr>
<?php endforeach; ?>
```

Редагування і видалення контактів реалізуються викликом методів реалізованих в ORM. При передачі даних отриманих від користувача ІСАОТ також відбувається виклик зберезувальних процедур, екранування даних. Ці заходи захищають інформаційну систему автоматизації обліку товарообігу від SQL-ін'єкцій.

2.12 Встановлення та налаштування ІСАОТ

Встановлення ІСАОТ передбачає налаштування сервера, на якому буде розгорнуто проєкт. З боку клієнта потрібен лише браузер та доступ до мережі «Інтернет». Налаштування сервера для ІСАОТ включає:

- встановлення PHP 7.4.30 або вище;
- завантаження програмного забезпечення для об'єктно-реляційного відображення – RedBeanPHP та його переміщення у папку з файлами проєкту;
- підключення MySQL БД у відповідному файлі ІСАОТ;
- встановлення вебсервера з підтримкою PHP.

Після успішної інсталяції усіх необхідних компонентів та виконання налаштувань ІСАОТ готова до роботи і підтримує віддалені підключення у випадку використання сервера доступного через мережу «Інтернет».

У випадку з файлами, що представляють певні розділи ІСАОТ не потрібно виконувати заходи для обмеження доступу до файлів, оскільки, для неавторизованого користувача буде виведено форму авторизації.

2.13 Тестування та верифікація ІСАОТ

Тестування JS-скриптів було проведено за допомогою вебсервісу «Involta». В результаті перевірки було отримано повідомлення про валідність описаних процедур побудови форм та їх елементів, що використовуються у інформаційній системі автоматизації обліку товарообігу(див. рисунок 2.16).

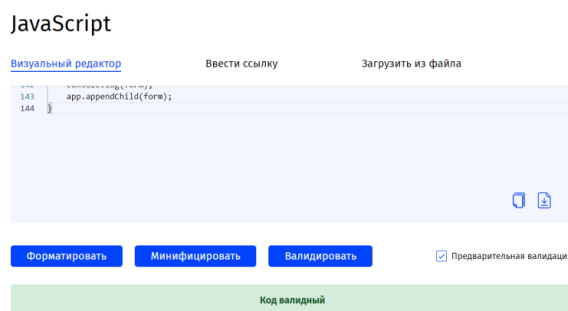


Рисунок 2.16 – Результат валідації JS-скриптів ІСАОТ

Валідація каскадних таблиць стилів не проводилась, оскільки, було використано стандартні таблиці стилів фреймворку Bootstrap.

Валідація PHP-скриптів проводилась у сервісі «PiliApp». Після вибору версії PHP та запуску валідації було отримано повідомлення про валідність описаних процедур, що використовуються у ІСАОТ (див. рисунок 2.17).

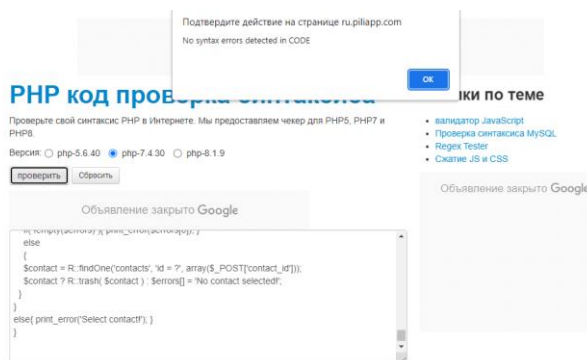


Рисунок 2.17 – Результати валідації PHP-скриптів ІСАОТ

Для тестування зберезувальних процедур було розроблено ряд запитів, які реалізують ручне позитивне тестування [39]. Для перевірки коректності роботи ЗП `create_lead` її було викликано з передачею необхідних параметрів, як на рисунку 2.18. Тестування інших ЗП наведено у додатку Л.

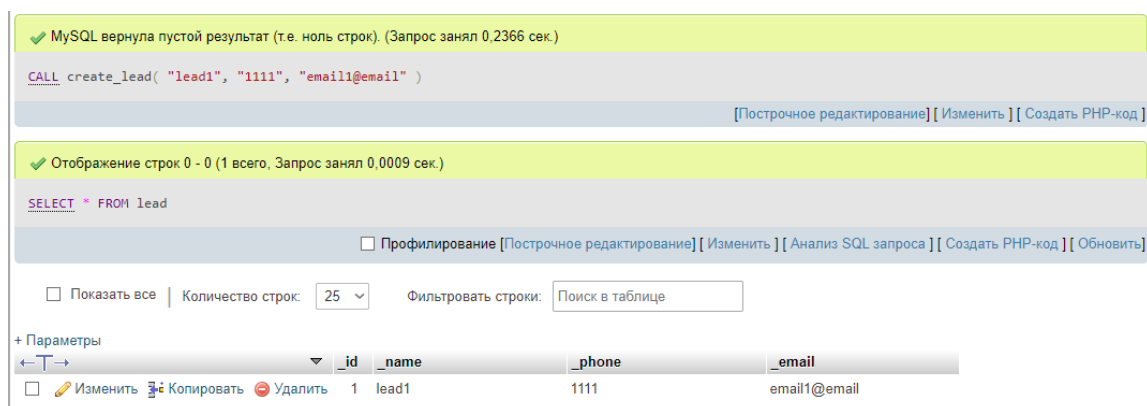


Рисунок 2.18 – Результат виконання ЗП `create_lead`

В результаті тестування не було виявлено недоліків, які б могли завадити коректній роботі інформаційної системи автоматизації обліку товарообігу.

В процесі хостингу вебзастосунку для автоматизації обліку товарообігу було виявлено сценарій, при якому отримувалось повідомлення про неможливість почати сесію, оскільки, метадані вже було надіслано. Проблема полягала в тому, що код, який розпочинав сесію знаходився нижче

HTML-коду. Рішенням стало переміщення PHP-коду вище гіпертекстової розмітки сторінки.

В решті сценаріїв, які виконувались у ході тестування ІСАОТ не було виявлено суттєвих помилок, які б частково або повністю перешкоджали виконанню запроєктованих функцій інформаційної системи автоматизації обліку товарообігу.

Набір доступних дій у ІСАОТ відповідає списку функціональних можливостей зазначених на етапі проєктування ІСАОТ. На основі цього було зроблено висновок, про відповідність реалізованої інформаційної системи автоматизації обліку товарообігу поставленим вимогам.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Особливості електротравматизму, електричний струм як чинник небезпеки для користувачів ІСАОТ

Використання інформаційної системи автоматизації обліку товарообігу передбачає використання комп'ютерної техніки, що забезпечує доступ до необхідних сервісів. Це надає актуальності дослідженню особливостей електротравматизму та електричного струму як чинника небезпеки при використанні ІСАОТ.

Користувач ІСАОТ може отримати електротравму при потраплянні під напругу в результаті прямого або опосередкованого контакту з елементами, електричний потенціал яких відрізняється від електричного потенціалу поверхні, на якій він знаходиться. Результатом такого контакту є утворення електричної дуги між елементами обладнання та користувачем ІСАОТ, який має контакт із землею.

Електротравматизм складає близько 1% виробничого травматизму і 15-20% серед випадків з летальними наслідками. Це свідчить про те, що електротравми за своїм характером класифікують як важкі. Також особливістю електротравматизму є те, що на електроустановки напругою нижче 1 кВ припадає до 70-80% смертельних електротравм, а на електроустановки, напругою, що перевищує 1 кВ. – до 20-30% [40]. Тому вивчення питань стосовно електротравматизму є важливим для користувачів ІСАОТ.

Такий розподіл величин напруги в загальній кількості електротравм зумовлений більшою розповсюдженістю електроустановок напругою до 1 кВ. Також на це впливає те, що установки з нижчою напругою доступні більшому загалу потенційних користувачів ІСАОТ, які мають недостатньо чіткі уявлення про небезпеки електричного струму та вимоги, які необхідні

для підтримання рівня безпеки на високому рівні при експлуатації інформаційної системи автоматизації обліку товарообігу.

Також слід відзначити, що користувачі ІСАОТ не використовуватимуть установок, напруга на яких перевищує 1 кВ.

Загалом виділяють такі особливості електротравматизму актуальні для користувачів ІСАОТ:

- людина не в змозі дистанційно, без спеціальних приладів, визначати наявність напруги, а тому дія струму, зазвичай, є раптовою, і захисна реакція організму проявляється тільки після потрапляння під напругу;

- струм, що протікає через тіло людини, діє на тканини і органи не тільки в місцях контакту зі струмовідними частинами і на шляху протікання, але й рефлекторно, як надзвичайно сильний подразник, впливає на весь організм, що може призводити до порушення функціонування життєво важливих систем організму – нервової, серцево-судинної систем, дихання, тощо;

- електротравми можливі без дотику людини до струмовідних частин – внаслідок утворення електричної дуги при пробіі повітряного проміжку між струмовідними частинами, або між струмовідними частинами і людиною, чи землею;

- розслідуванню, обліку і аналізу, в основному, доступні тяжкі електротравми та електротравми зі смертельними наслідками, що негативно впливає на профілактику електротравм.

З метою запобігання електротравмам користувачів ІСАОТ розглянуто різні заходи [41], до них відносять: недоступність струмопровідних частин для випадкового дотику. Вона досягається ізоляцією струмонепровідними матеріалами. Провідники електричного струму повинні мати робочу ізоляцію, яка повинна захистити користувачів ІСАОТ від контакту з провідниками під напругою. Наступним заходом є застосування малих напруг – дуже ефективний захист від ураження електричним струмом. Для живлення кіл управління технологічним обладнанням, яке використовується

для експлуатації ІСАОТ, використовують напругу не вище 42 В. На шафах і пультах управління обладнанням розміщують штепсельні розетки з напругою не вище 12 В для включення переносних світильників, які використовуються при необхідності забезпечення додаткового освітлення у приміщеннях для користувачів ІСАОТ.

Захисне заземлення – свідоме електричне з'єднання з землею чи її еквівалентом металевих частин електроустаткування потрібного для експлуатації ІСАОТ [42].

Занулення – свідоме електричне з'єднання з нульовим захисним провідником металевих струмонепровідних частин комп'ютерної техніки потрібної користувачам ІСАОТ, які можуть виявитися під напругою. Занулення повинно відповідати вимогам, що зазначені у відповідних стандартах [43].

Ще одним з можливих заходів є захисне вимкнення – швидкодійний захист, що забезпечує автоматичне відключення комп'ютерної техніки, яку використовують користувачі ІСАОТ, коли в ньому виникає небезпека ураження струмом. Така небезпека може виникнути у випадку:

- замикання фази на корпус комп'ютерної техніки;
- пониження опору ізоляції фаз відносно землі;
- появи в мережі більш високої напруги;
- торкання користувачів ІСАОТ до струмопровідних частин.

Захист користувачів ІСАОТ від електричних статичних полів полягає в знятті з ізольованих від землі металевих частин устаткування електричної напруги, яка виникає внаслідок статичної електризації під час технологічних процесів, що супроводжуються тертям, роздрібненням твердих тіл, пересипанням сипучих матеріалів, переливанням рідин-діелектриків. Дія статичної електрики на користувачів ІСАОТ може проявлятися у вигляді слабого, помірного чи навіть сильного уколу, який сам по собі не являє небезпеки, але може бути непрямою причиною нещасного випадку внаслідок рефлекторного руху користувача ІСАОТ до небезпечної зони.

Захист від атмосферної електрики – комплекс захисних пристроїв, призначених для забезпечення безпеки користувачів ІСАОТ, які працюють з комп'ютерною технікою, від ураження електричним струмом, дії електричної дуги та електромагнітного поля.

Основні ізоляційні електрозахисні засоби здатні довгий час витримувати робочу напругу комп'ютерної техніки потрібної для експлуатації ІСАОТ, тому ними дозволяється торкатися до струмопровідних частин. До таких засобів відносять: діелектричні гумові рукавиці, інструменти з ізольованими ручками та. Ізоляційні штанги, ізоляційні і струмовимірювальні кліщі, показники високої напруги – в електроустаткуванні з напругою вище 1000 В.

3.2 Вимоги ергономіки до організації робочого місця користувача ПК, що використовуватиме ІСАОТ

Загальні ергономічні вимоги для організації робочого місця користувача ІСАОТ встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт [44-45]. Параметри робочого місця користувача ІСАОТ повинні бути наступними [46]:

- площа кабінету не менш 6 м², а об'єм не менш 24 м³;
- для обробки приміщення повинні використовуватися дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі – 0,7-0,8; для стін – 0,5-0,6; для підлоги – 0,3-0,5;
- робочий стіл повинен забезпечувати оптимальне розміщення обладнання необхідного для функціонування ІСАОТ;
- крісло повинне підтримувати раціональну робочу позу під час роботи з ІСАОТ та давати можливість змінювати позу для зменшення статичного напруження м'язів шийно-плечової області та спини для попередження розвитку втоми працюючого як зазначено у відповідних

стандартах [47]. Рекомендовані параметри робочого місця подано на рисунку 3.1 [48].

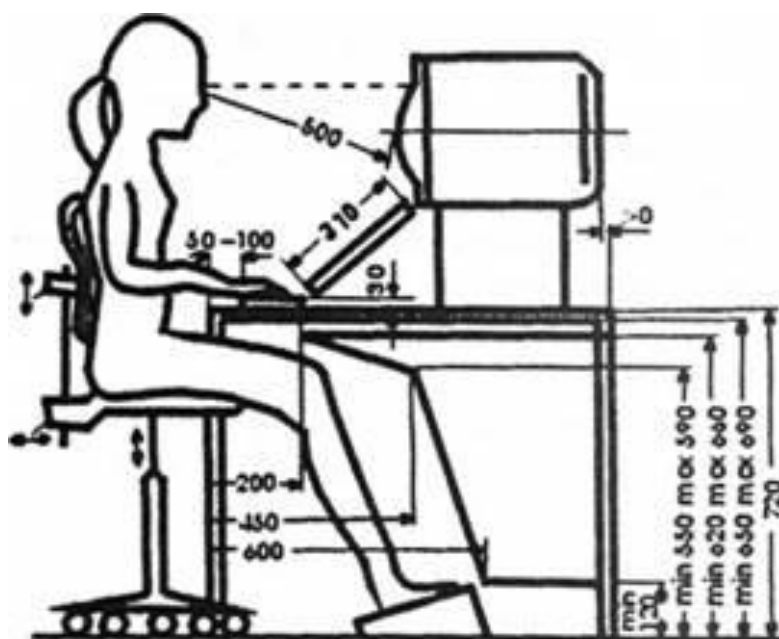


Рисунок 3.1 – Рекомендовані параметри робочого місця користувача ІСАОТ

Висота робочої поверхні столу користувача ІСАОТ, повинна складати 725 мм і мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, не менше 450 мм в глибину на рівні колін і на рівні простягнутої ноги – не менше 650 мм. Відстань від очей користувача ІСАОТ до екрану дисплея має становити 500-700 мм. Кут зору 10-20, але не більше 40; кут між верхнім краєм дисплея і рівнем очей користувача має становити не менше 10. Кращим є розташування екрану перпендикулярно до лінії зору користувача [49]. Також до ергономічних вимог обладнання робочого місця користувача ІСАОТ відносять вимоги до освітлення, вони повинні відповідати вимогам, що визначені у відповідних стандартах [50].

Повітряне середовище в робочій зоні користувачів ІСАОТ визначається мікрокліматом виробничого приміщення. Величини температури, відносної вологості та швидкості руху повітря на робочих місцях з дисплеями повинні відповідати допустимим значенням, які встановлені [51].

Допустимі рівні звуку та еквівалентні рівні звуку на робочих місцях користувачів ІСАОТ повинні відповідати вимогам санітарних норм та допустимих рівнів шуму на робочих місцях. Згідно з цими нормами в приміщенні, де працює людина рівень шуму не повинен перевищувати 60 дБ [52]. Висока температура повітря негативно позначається на функціональному стані людини, тому при створенні комфортних умов роботи особливу увагу необхідно приділити шляхам відводу повітря.

3.3 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано травми, що загрожують користувачам інформаційної системи автоматизації обліку товарообігу при порушенні норм, що зазначені у відповідних стандартах, що стосуються експлуатації електроустаткування. Також розглянуто заходи унеможливлення ураження користувачів ІСАОТ електричним струмом. Крім того розглянуто норми, що стосуються ергономічних вимоги до обладнання робочого місця користувачів ІСАОТ. Вони включають: площу кабінету, освітлення, облаштування робочого місця та інші параметри.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було реалізовано інформаційну систему автоматизації обліку товарообігу, проведено аналіз аналогічних вебсервісів, який довів актуальність створення вебзастосунку, засобів реалізації. Також було детально досліджено предметну область, в результаті чого було виділено інформаційні об'єкти-сутності, які стали основою для подальшої розробки ІСАОТ. Також було розглянуто системи управління базами даних і можливі вебтехнології для реалізації ІСАОТ.

Засобами PHP, JS, Bootstrap та MySQL було реалізовано базу даних та вебзастосунок, які входять до складу розроблюваної інформаційної системи автоматизації обліку товарообігу. ІСАОТ містить набір інструментів, який забезпечує виконання запроєктованих функцій і дає можливість в майбутньому реалізувати розширення функціоналу.

Після впровадження та тестування вебзастосунку було виправлено помилки, що призводили до проблем з авторизацією користувачів інформаційної системи автоматизації обліку товарообігу.

Отже, результатом виконання кваліфікаційної роботи стала інформаційна система автоматизації обліку товарообігу, яка відповідає поставленим вимогам, виконує запроєктовані функції та дає можливість для подальшої розробки та розширення функціоналу.

ПЕРЕЛІК ДЖЕРЕЛ

1 Автоматизація бізнес-процесів: суть і переваги. Нетсофт. URL: <https://techexpert.ua/automate-business-processes/> (дата звернення: 04.02.2023).

2 Автоматизація бізнес-процесів: плюси, проблеми, можливості - TechExpert. TechExpert. URL: <https://techexpert.ua/automate-business-processes/> (дата звернення: 06.02.2023).

3 Метод аналізу ієрархій - Система підтримки прийняття рішень ІТ-компанії "Тріумф ІТ". Система підтримки прийняття рішень ІТ-компанії "Тріумф ІТ". URL: <https://dss.tg.ck.ua/ahp-help> (дата звернення: 06.02.2023).

4 Структура та функції інформаційних систем. Pidru4niki. URL: https://pidru4niki.com/2015101166602/menedzhment/struktura_funktsiyi_informatsiynih_sistem (дата звернення: 12.02.2023).

5 Що таке бізнес-процеси в підприємницькій діяльності?. Школа бізнесу. URL: <https://online.novaposhta.education/blog/scho-take-biznes-protsesi-v-pidpriyemnitskij-diyalnosti> (дата звернення: 12.02.2023).

6 Діаграми потоків даних (dfd). StudFiles. URL: <https://studfile.net/preview/5462334/page:11/> (дата звернення: 12.02.2023).

7 Creating a Custom ACL in PHP | Developer.com. Developer.com. URL: <https://www.developer.com/open-source/creating-a-custom-acl-in-php/> (date of access: 04.03.2023).

8 СУБД (Системы управления базами данных) - что это, виды и функции. Highload.today - медиа для разработчиков. URL: <https://highload.today/subd-kakie-byvayut-kak-vybrat/> (дата звернення: 04.03.2023).

9 Davert. Альтернативы для JavaScript. Хабр. URL: <https://habr.com/ru/articles/163033/> (дата звернення: 04.03.2023).

10 7 причин, по которым PHP хорош для разработки бизнес-проектов. Custom Software Development Company | Stfalcon.com. URL:

<https://stfalcon.com/ru/blog/post/PHP-advantages-for-business> (дата звернення: 09.03.2023).

11 Bootstrap: що це, з чого почати вивчення і як використовувати. ДРУЗІ. URL: <https://druzy.com.ua/bootstrap-sho-ce-z-chogo-pochati-vivchennia-i-iaak-vikoristovuvati/> (дата звернення: 10.03.2023).

12 Що таке локальний сервер і для чого він потрібен - Українське програмування. Українське програмування - Українське програмування. URL: <http://programming.in.ua/web-design/allphp/68-local-serv-priority.html> (дата звернення: 11.03.2023).

13 Найкращі середовища для розробки на Python та редакторів коду у 2021 році - Інший. Огляди, Ігри, Розваги, Червень 2023. URL: <https://uk.myservname.com/12-best-python-ides> (дата звернення: 11.03.2023).

14 Три способа разработки веб-сайта – Разработка на vc.ru. vc.ru. URL: <https://vc.ru/dev/78714-3-sposoba-razrabotki-veb-sayta> (дата звернення: 12.03.2023).

15 Що таке вебдодаток? – WebCase. URL: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/> (дата звернення 12.03.2023).

16 Учасники проектів Вікімедіа. Життєвий цикл програмного забезпечення – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Життєвий_цикл_програмного_забезпечення (дата звернення: 14.03.2023).

17 Учасники проектів Вікімедіа. IDEF – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/IDEF> (дата звернення: 24.03.2023).

18 Учасники проектів Вікімедіа. Триярусна архітектура – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Триярусна_архітектура (дата звернення: 24.03.2023).

19 <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>

20 Welcome. RedBeanPHP. URL: <https://www.redbeanphp.com/index.php> (дата звернення: 24.03.2023).

21 DOM - об'єктна модель документа - JavaScript довідка. JavaScript - JavaScript довідка. URL: <http://xn--80adth0aefm3i.xn--j1amh/DOM> (дата звернення: 01.04.2023).

22 Аналіз предметної області. Stud. URL: https://stud.com.ua/77208/informatika/analiz_predmetnoyi_oblasti (дата звернення: 04.04.2023).

23 Бази даних. ER-модель. Поняття зв'язку. Потужність зв'язку. Типи зв'язків. Приклади | BestProg. BestProg | Програмування: теорія та практика. URL: <https://www.bestprog.net/uk/2019/01/27/er-model-the-concept-of-relationship-the-relationship-capacity-types-of-relationships-examples-ua/> (дата звернення: 04.04.2023).

24 Учасники проектів Вікімедіа. Концептуальна схема – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Концептуальна_схема (дата звернення: 04.04.2023).

25 Учасники проектів Вікімедіа. Діаграма прецедентів – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Діаграма_прецедентів (дата звернення: 07.04.2023).

26 Збережені процедури | e-helper.com.ua. e-helper.com.ua | Учат здесь!. URL: <http://www.e-helper.com.ua/node/40> (дата звернення: 09.04.2023).

27 Тестування безпеки: SQL-ін'єкції | Безкоштовний онлайн-курс від компанії QATestLab. Безкоштовний онлайн-курс від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/technical-articles/security-testing-sql-injection/> (дата звернення: 10.04.2023).

28 MS SQL Server и T-SQL | Хранимые процедуры. METANIT.COM - Сайт о программировании. URL: <https://metanit.com/sql/sqlserver/11.1.php> (дата звернення: 15.04.2023).

29 jQuery.ajax() | jQuery API Documentation. jQuery API Documentation. URL: <https://api.jquery.com/jquery.ajax/> (date of access: 16.04.2023).

30 Setup | RedBeanPHP. RedBeanPHP :: Welcome. URL: https://redbeanphp.com/manual3_0/index.php?p=/manual3_0/setup (дата звернення: 18.04.2023).

31 PHP: \$_POST - Manual. PHP: Hypertext Preprocessor. URL: <https://www.php.net/manual/ru/reserved.variables.post.php> (дата звернення: 24.04.2023).

32 RedBeanPHP Counting. RedBeanPHP Welcome. URL: <https://redbeanphp.com/index.php?p=/counting> (дата звернення: 27.04.2023).

33 RedBeanPHP CRUD. RedBeanPHP Welcome. URL: <https://redbeanphp.com/index.php?p=/crud> (дата звернення: 27.04.2023).

34 RedBeanPHP Quick Tour. RedBeanPHP Welcome. URL: https://redbeanphp.com/index.php?p=/quick_tour (дата звернення: 27.04.2023).

35 https://developer.mozilla.org/ru/docs/Web/API/Fetch_API/Using_Fetch

36 Учасники проектів Вікімедіа. JSON – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/JSON> (дата звернення: 27.04.2023).

37 PHP: \$_SESSION: Hypertext Preprocessor. URL: <https://www.php.net/manual/en/reserved.variables.session.php> (дата звернення: 29.04.2023).

38 RedBeanPHP Querying. RedBeanPHP Welcome. URL: <https://redbeanphp.com/index.php?p=/querying> (дата звернення: 02.05.2023).

39 Учасники проектів Вікімедіа. Ручне тестування – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Ручне_тестування (дата звернення: 02.05.2023).

40 Електротравматизм та дія електричного струму на організм людини - Бібліотека BukLib.net. Головна - Бібліотека BukLib.net. URL: <https://buklib.net/books/31413/> (дата звернення: 05.05.2023).

41 Методи і засоби захисту від ураження електричним струмом - Бібліотека BukLib.net. Головна - Бібліотека BukLib.net. URL: <https://buklib.net/books/35195/> (дата звернення: 06.05.2023).

42 ДСТУ Б В.2.5-82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. Чинний від

2017-04-01. Вид. офіц. Науково-виробн. підприємство «Укренергоналадкавимірювання».

43 ДСТУ 7237:2011. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту. Чинний від 2011-08-01. Вид. офіц. Держспоживстандарт України.

44 ДСТУ ISO 9241-1:2003. Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 1. Загальні положення (ISO 9241-1:1997, IDT). Чинний від 2004-10-01. Вид. офіц. Держспоживстандарт України.

45 ДСТУ 7234:2011. Дизайн і ергономіка. Обладнання виробниче. Загальні вимоги дизайну та ергономіки. Чинний від 2011-08-01. Вид. офіц. Укр. НДІ дизайну та ергономіки НАУ.

46 ДСТУ 7299:2013. Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки. Чинний від 2014-01-01. Вид. офіц.

47 ДСТУ 7951:2015. Дизайн і ергономіка. Крісло оператора. Загальні ергономічні вимоги. Чинний від 2014-01-01. Вид. офіц.

48 ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. Чинний від 2017-07-01. Вид. офіц.

49 Ергономічні вимоги для організації робочого місця. Головна. URL: <https://www.gpp.in.ua/robota/ergonomichni-vimogi-dlya-organizatsiji-robochogo-mistsya> (дата звернення: 04.06.2023).

50 ДСТУ EN 12464-1:2016. Світло та освітлення. Освітлення робочих місць. Частина 1. Внутрішні робочі місця (EN 12464-1:2011, IDT). Чинний від 2017-12-01. Вид. офіц.

51 ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. Чинний від 1999-12-01. Вид. офіц.

52 Атаманчук П.С., Мендерецький В.В., Панчук О.П., Чорна О.Г. Інтегрований курс безпеки життєдіяльності (теоретичні основи): Навч. посіб. - Кам'янець-Подільський: Буйницький О .А. ,2009 . - 200 с .

ДОДАТКИ

Результати аналізу переваг та недоліків аналогів ІСАОТ

| | | | | | |
|------------------|---|----------------|--------------------------------|----------------------|--------|
| ІСАОТ | | | | | |
| Відповідність | клієнти | облік поставок | облік продажів | аналіз даних | знижки |
| Масштабованість | розширюваність | | можливість додавати функціонал | | |
| Конфіденційність | вимагає малу кількість персональних даних | | | | |
| Зручність | простий інтерфейс | | | доступність функцій | |
| Адаптивність | мобільний додаток | | додаток для ПК | адаптивний інтерфейс | |
| Зрозумілість | навчання | | підказки | підтримка | |

Рисунок А1 – Результати аналізу вебсервісу «ІСАОТ»

| | | | | | |
|------------------|---|----------------|--------------------------------|----------------------|--------|
| G-Tables | | | | | |
| Відповідність | клієнти | облік поставок | облік продажів | аналіз даних | знижки |
| Масштабованість | розширюваність | | можливість додавати функціонал | | |
| Конфіденційність | вимагає малу кількість персональних даних | | | | |
| Зручність | простий інтерфейс | | | доступність функцій | |
| Адаптивність | мобільний додаток | | додаток для ПК | адаптивний інтерфейс | |
| Зрозумілість | навчання | | підказки | підтримка | |

Рисунок А2 – Результати аналізу вебсервісу «G-Tables»

| | | | | | |
|------------------|---|----------------|--------------------------------|----------------------|--------|
| Simple WMS | | | | | |
| Відповідність | клієнти | облік поставок | облік продажів | аналіз даних | знижки |
| Масштабованість | розширюваність | | можливість додавати функціонал | | |
| Конфіденційність | вимагає малу кількість персональних даних | | | | |
| Зручність | простий інтерфейс | | | доступність функцій | |
| Адаптивність | мобільний додаток | | додаток для ПК | адаптивний інтерфейс | |
| Зрозумілість | навчання | | підказки | підтримка | |

Рисунок А3 – Результати аналізу вебсервісу «Simple WMS»

| | | | | | |
|------------------|---|----------------|--------------------------------|----------------------|--------|
| Рем онлайн | | | | | |
| Відповідність | клієнти | облік поставок | облік продажів | аналіз даних | знижки |
| Масштабованість | розширюваність | | можливість додавати функціонал | | |
| Конфіденційність | вимагає малу кількість персональних даних | | | | |
| Зручність | простий інтерфейс | | | доступність функцій | |
| Адаптивність | мобільний додаток | | додаток для ПК | адаптивний інтерфейс | |
| Зрозумілість | навчання | | підказки | підтримка | |

Рисунок А4 – Результати аналізу вебсервісу «Рем Онлайн»

Обчислення коефіцієнтів ваг критеріїв та векторів пріоритетності

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & 7 \\ \frac{1}{7} & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} & \text{size} &:= 2 & I_2 &:= 0 \\
 & & L_0 &:= 2 & & & & \\
 w &:= \text{eigenvec}(A, L_0) & & & CI &:= \frac{L_0 - \text{size}}{\text{size} - 1} = 0 \\
 & & & & CR &:= \frac{CI}{I_2} = 0 \\
 w &= \begin{pmatrix} 0.99 \\ 0.141 \end{pmatrix} & & & & & & &
 \end{aligned}$$

Рисунок Б1 – Аналіз 2-го рівня ієрархічної моделі

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & 2 & 2 \\ \frac{1}{2} & 1 & 2 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 3.054 \\ -0.027 + 0.404i \\ -0.027 - 0.404i \end{pmatrix} & \text{size} &:= 3 \\
 & & & & I_3 &:= 0.5247 \\
 & & & & L_0 &:= 3.054 \\
 w &:= \text{eigenvec}(A, L_0) & w &= \begin{pmatrix} 0.802 \\ 0.505 \\ 0.318 \end{pmatrix} & CI &:= \frac{L_0 - \text{size}}{\text{size} - 1} = 0.027 \\
 & & & & CR &:= \frac{CI}{I_3} = 0.051
 \end{aligned}$$

Рисунок Б2 – Аналіз 3-го рівня ієрархії за 1 критерієм

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & 2 & 3 \\ \frac{1}{2} & 1 & 3 \\ \frac{1}{3} & \frac{1}{3} & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 3.054 \\ -0.027 + 0.404i \\ -0.027 - 0.404i \end{pmatrix} & \text{size} &:= 3 \\
 & & & & I_3 &:= 0.5247 \\
 & & & & L_0 &:= 3.054 \\
 w &:= \text{eigenvec}(A, L_0) & w &= \begin{pmatrix} 0.826 \\ 0.52 \\ 0.218 \end{pmatrix} & CI &:= \frac{L_0 - \text{size}}{\text{size} - 1} = 0.027 \\
 & & & & CR &:= \frac{CI}{I_3} = 0.051
 \end{aligned}$$

Рисунок Б3 – Аналіз 3-го рівня ієрархії за 2 критерієм

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & \frac{1}{3} & \frac{1}{3} & 2 & \frac{1}{3} \\ 3 & 1 & 1 & 3 & 1 \\ 3 & 1 & 1 & 3 & 1 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{2} \\ 3 & 1 & 1 & 3 & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 5.171 \\ -0.086 + 0.616i \\ -0.086 - 0.616i \\ 0 \\ 0 \end{pmatrix} & \text{size} &:= 5 \\
 & & & & I_5 &:= 1.1086 \\
 & & & & L_0 &:= 5.171 \\
 w &:= \text{eigenvec}(A, L_0) & CI &:= \frac{L_0 - \text{size}}{\text{size} - 1} = 0.043 \\
 & & CR &:= \frac{CI}{I_5} = 0.039 \\
 w &= \begin{pmatrix} 0.22 \\ 0.553 \\ 0.553 \\ 0.181 \\ 0.553 \end{pmatrix} & & & & & & &
 \end{aligned}$$

Рисунок Б4 – Аналіз 4-го рівня ієрархії за 1 критерієм

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & \frac{1}{7} & \frac{1}{4} & 1 & 1 \\ 7 & 1 & 1 & 3 & 3 \\ 4 & 1 & 1 & 3 & 3 \\ 1 & \frac{1}{3} & \frac{1}{3} & 1 & 1 \\ 1 & \frac{1}{3} & \frac{1}{3} & 1 & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 5.078 \\ -0.039 + 0.628i \\ -0.039 - 0.628i \\ 0 \\ 0 \end{pmatrix} & \begin{matrix} \text{size} := 5 \\ \text{IS} := 1.1086 \\ \text{L}_0 = 5.078 \end{matrix} \\
 \\ \\
 w &:= \text{eigenvec}(A, L_0) & \text{CI} &:= \frac{\text{L}_0 - \text{size}}{\text{size} - 1} = 0.02 \\
 w &= \begin{pmatrix} 0.16 \\ 0.715 \\ 0.621 \\ 0.197 \\ 0.197 \end{pmatrix} & \text{CR} &:= \frac{\text{CI}}{\text{IS}} = 0.018
 \end{aligned}$$

Рисунок Б5 – Аналіз 4-го рівня ієрархії за 2 критерієм

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & 3 & 1 \\ 2 & 1 & \frac{1}{2} & 3 & 2 \\ 3 & 2 & 1 & 3 & 2 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} & 2 & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 5.131 \\ -0.04 + 0.784i \\ -0.04 - 0.784i \\ -0.026 + 0.228i \\ -0.026 - 0.228i \end{pmatrix} & \begin{matrix} \text{size} := 5 \\ \text{IS} := 1.1086 \\ \text{L}_0 = 5.131 \end{matrix} \\
 \\ \\
 w &:= \text{eigenvec}(A, L_0) & \text{CI} &:= \frac{\text{L}_0 - \text{size}}{\text{size} - 1} = 0.033 \\
 w &= \begin{pmatrix} 0.31 \\ 0.502 \\ 0.731 \\ 0.161 \\ 0.302 \end{pmatrix} & \text{CR} &:= \frac{\text{CI}}{\text{IS}} = 0.03
 \end{aligned}$$

Рисунок Б6 – Аналіз 4-го рівня ієрархії за 3 критерієм

$$\begin{aligned}
 \underline{\underline{A}} &:= \begin{pmatrix} 1 & 1 & 2 & 3 & \frac{1}{2} \\ 1 & 1 & 2 & 3 & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 & 3 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} \\ 2 & 1 & 2 & 3 & 1 \end{pmatrix} & \underline{\underline{L}} &:= \text{eigenvals}(A) = \begin{pmatrix} 5.117 \\ 0.707i \\ -0.707i \\ -0.058 + 0.307i \\ -0.058 - 0.307i \end{pmatrix} & \begin{matrix} \text{size} := 5 \\ \text{IS} := 1.1086 \\ \text{L}_0 = 5.117 \end{matrix} \\
 \\ \\
 w &:= \text{eigenvec}(A, L_0) & \text{CI} &:= \frac{\text{L}_0 - \text{size}}{\text{size} - 1} = 0.029 \\
 w &= \begin{pmatrix} 0.467 \\ 0.528 \\ 0.31 \\ 0.156 \\ 0.619 \end{pmatrix} & \text{CR} &:= \frac{\text{CI}}{\text{IS}} = 0.026
 \end{aligned}$$

Рисунок Б7 – Аналіз 4-го рівня ієрархії за 4 критерієм

$$A := \begin{pmatrix} 1 & \frac{1}{2} & 1 & \frac{1}{3} & \frac{1}{2} \\ 2 & 1 & 3 & 1 & 1 \\ 1 & \frac{1}{2} & 1 & \frac{1}{4} & \frac{1}{4} \\ 2 & 1 & 4 & 1 & 1 \\ 2 & 1 & 4 & 1 & 1 \end{pmatrix} \quad L := \text{eigenvals}(A) = \begin{pmatrix} 5.078 \\ -0.039 + 0.478i \\ -0.039 - 0.478i \\ 0 \\ 0 \end{pmatrix} \quad \begin{matrix} \text{size} := 5 \\ I5 := 1.1086 \\ L_0 = 5.078 \end{matrix}$$

$$w := \text{eigenvec}(A, L_0) \quad CI := \frac{L_0 - \text{size}}{\text{size} - 1} = 0.02$$

$$w = \begin{pmatrix} 0.226 \\ 0.527 \\ 0.189 \\ 0.564 \\ 0.564 \end{pmatrix} \quad CR := \frac{CI}{I5} = 0.018$$

Рисунок Б8 – Аналіз 4-го рівня ієрархії за 5 критерієм

$$A := \begin{pmatrix} 1 & 3 & 2 & 2 & 2 \\ \frac{1}{3} & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 2 & 1 & 1 & \frac{1}{2} \\ \frac{1}{2} & 2 & 1 & 1 & \frac{1}{2} \\ \frac{1}{2} & 2 & 2 & 2 & 1 \end{pmatrix} \quad L := \text{eigenvals}(A) = \begin{pmatrix} 5.088 \\ -0.025 + 0.666i \\ -0.025 - 0.666i \\ -0.037 \\ 0 \end{pmatrix} \quad \begin{matrix} \text{size} := 5 \\ I5 := 1.1086 \\ L_0 = 5.088 \end{matrix}$$

$$w := \text{eigenvec}(A, L_0) \quad CI := \frac{L_0 - \text{size}}{\text{size} - 1} = 0.022$$

$$w = \begin{pmatrix} 0.707 \\ 0.198 \\ 0.324 \\ 0.324 \\ 0.5 \end{pmatrix} \quad CR := \frac{CI}{I5} = 0.02$$

Рисунок Б9 – Аналіз 4-го рівня ієрархії за 6 критерієм

$$W31 := \begin{pmatrix} 0.882 \\ 0.347 \\ 0.32 \end{pmatrix} \quad W32 := \begin{pmatrix} 0.826 \\ 0.520 \\ 0.218 \end{pmatrix}$$

$$M41 := \begin{pmatrix} 0.220 & 0.160 & 0.320 \\ 0.553 & 0.715 & 0.502 \\ 0.553 & 0.621 & 0.731 \\ 0.181 & 0.197 & 0.161 \\ 0.553 & 0.197 & 0.302 \end{pmatrix} \quad M42 := \begin{pmatrix} 0.467 & 0.226 & 0.707 \\ 0.528 & 0.527 & 0.198 \\ 0.310 & 0.189 & 0.324 \\ 0.156 & 0.564 & 0.324 \\ 0.619 & 0.564 & 0.500 \end{pmatrix}$$

Рисунок Б10 – Матриці локальних пріоритетів

$$M_{41} \cdot W_{31} = \begin{pmatrix} 0.352 \\ 0.896 \\ 0.937 \\ 0.28 \\ 0.653 \end{pmatrix} \quad M_{42} \cdot W_{32} = \begin{pmatrix} 0.657 \\ 0.753 \\ 0.425 \\ 0.493 \\ 0.914 \end{pmatrix} \quad W_{21} := \begin{pmatrix} 0.990 \\ 0.141 \end{pmatrix}$$

$$M := \begin{pmatrix} 0.352 & 0.657 \\ 0.896 & 0.753 \\ 0.937 & 0.425 \\ 0.28 & 0.493 \\ 0.653 & 0.914 \end{pmatrix} \quad M \cdot W_{21} = \begin{pmatrix} 0.441 \\ 0.993 \\ 0.988 \\ 0.347 \\ 0.775 \end{pmatrix}$$

Рисунок Б11 – Обчислення глобальних пріоритетів

$$W_{31} := \begin{pmatrix} 0.882 \\ 0.347 \\ 0.32 \end{pmatrix} \quad W_{32} := \begin{pmatrix} 0.826 \\ 0.520 \\ 0.218 \end{pmatrix}$$

$$M_{41} := \begin{pmatrix} 0.220 & 0.160 & 0.320 \\ 0.553 & 0.715 & 0.502 \\ 0.553 & 0.621 & 0.731 \\ 0.181 & 0.197 & 0.161 \\ 0.553 & 0.197 & 0.302 \end{pmatrix} \quad M_{42} := \begin{pmatrix} 0.467 & 0.226 & 0.707 \\ 0.528 & 0.527 & 0.198 \\ 0.310 & 0.189 & 0.324 \\ 0.156 & 0.564 & 0.324 \\ 0.619 & 0.564 & 0.500 \end{pmatrix}$$

Рисунок Б12 – Матриці локальних пріоритетів

$$M_{41} \cdot W_{31} = \begin{pmatrix} 0.352 \\ 0.896 \\ 0.937 \\ 0.28 \\ 0.653 \end{pmatrix} \quad M_{42} \cdot W_{32} = \begin{pmatrix} 0.657 \\ 0.753 \\ 0.425 \\ 0.493 \\ 0.914 \end{pmatrix} \quad W_{21} := \begin{pmatrix} 0.990 \\ 0.141 \end{pmatrix}$$

$$M := \begin{pmatrix} 0.352 & 0.657 \\ 0.896 & 0.753 \\ 0.937 & 0.425 \\ 0.28 & 0.493 \\ 0.653 & 0.914 \end{pmatrix} \quad M \cdot W_{21} = \begin{pmatrix} 0.441 \\ 0.993 \\ 0.988 \\ 0.347 \\ 0.775 \end{pmatrix}$$

Рисунок Б13 – Обчислення глобальних пріоритетів

Результати обчислення векторів пріоритетності

| 4 рівень (масштабованість) | Мій склад | ICOAT | G-Tables | Simple WMS | Рем онлайн | Власний вектор |
|----------------------------|-----------|-------|----------|------------|------------|----------------|
| Мій склад | 1 | 1/3 | 1/3 | 1 | 1 | 0,160 |
| ICOAT | 3 | 1 | 1 | 3 | 3 | 0,715 |
| G-Tables | 3 | 1 | 1 | 3 | 3 | 0,621 |
| Simple WMS | 1 | 1/3 | 1/3 | 1 | 1 | 0,197 |
| Рем онлайн | 1 | 1/3 | 1/3 | 1 | 1 | 0,197 |

Рисунок В1 – Власні вектори 4-го рівня ієрархії за критерієм «Масштабованість»

| 4 рівень (конфіденційність) | Мій склад | ICOAT | G-Tables | Simple WMS | Рем онлайн | Власний вектор |
|-----------------------------|-----------|-------|----------|------------|------------|----------------|
| Мій склад | 1 | 1/2 | 1/3 | 3 | 1 | 0,320 |
| ICOAT | 2 | 1 | 1/2 | 3 | 2 | 0,502 |
| G-Tables | 3 | 2 | 1 | 3 | 2 | 0,731 |
| Simple WMS | 1/3 | 1/3 | 1/3 | 1 | 1/2 | 0,161 |
| Рем онлайн | 1 | 1/2 | 1/2 | 2 | 1 | 0,302 |

Рисунок В2 – Власні вектори 4-го рівня ієрархії за критерієм «Конфіденційність»

| 4 рівень (зручність) | Мій склад | ICOAT | G-Tables | Simple WMS | Рем онлайн | Власний вектор |
|----------------------|-----------|-------|----------|------------|------------|----------------|
| Мій склад | 1 | 1 | 2 | 3 | 1/2 | 0,467 |
| ICOAT | 1 | 1 | 2 | 3 | 1 | 0,528 |
| G-Tables | 1/2 | 1/2 | 1 | 3 | 1/2 | 0,310 |
| Simple WMS | 1/3 | 1/3 | 1/3 | 1 | 1/3 | 0,156 |
| Рем онлайн | 2 | 1 | 2 | 3 | 1 | 0,619 |

Рисунок В3 – Власні вектори 4-го рівня ієрархії за критерієм «Зручність»

| 4 рівень (адаптивність) | Мій склад | ICOAT | G-Tables | Simple WMS | Рем онлайн | Власний вектор |
|-------------------------|-----------|-------|----------|------------|------------|----------------|
| Мій склад | 1 | 1/2 | 1 | 1/3 | 1/2 | 0,226 |
| ICOAT | 2 | 1 | 3 | 1 | 1 | 0,527 |
| G-Tables | 1 | 1/2 | 1 | 1/4 | 1/4 | 0,189 |
| Simple WMS | 2 | 1 | 4 | 1 | 1 | 0,564 |
| Рем онлайн | 2 | 1 | 4 | 1 | 1 | 0,564 |

Рисунок В4 – Власні вектори 4-го рівня ієрархії за критерієм «Адаптивність»

| 4 рівень (зрозумілість) | Мій склад | ICOAT | G-Tables | Simple WMS | Рем онлайн | Власний вектор |
|-------------------------|-----------|-------|----------|------------|------------|----------------|
| Мій склад | 1 | 3 | 2 | 2 | 2 | 0,707 |
| ICOAT | 1/3 | 1 | 1/2 | 1/2 | 1/2 | 0,198 |
| G-Tables | 1/2 | 2 | 1 | 1 | 1/2 | 0,324 |
| Simple WMS | 1/2 | 2 | 1 | 1 | 1/2 | 0,324 |
| Рем онлайн | 1/2 | 2 | 2 | 2 | 1 | 0,500 |

Рисунок В5 – Власні вектори 4-го рівня ієрархії за критерієм «Зрозумілість»

Основні компоненти ІСАОТ

Таблиця Д1 – Компоненти ІСАОТ

| Назва файлу | Призначення |
|------------------|-----------------------------------|
| charts.php | Виводить статистику продажів |
| clients.php | Дії з клієнтами ІСАОТ |
| create_task.php | Додавання завдання |
| data.php | Дані для статистики |
| delete_task.php | видалення завдання |
| goods.php | Дії з товарами |
| index.php | Індексна сторінка ІСАОТ |
| leads.php | Дії з керівниками |
| market.php | Сторінка покупок |
| my_tasks.php | Завдання користувача |
| sellers.php | Робота з продавцями |
| shops.php | Робота з магазинами |
| tasks_for_me.php | Завдання для користувача |
| update_task.php | Оновлення завдання |
| view.php | Перегляд завдань |
| bootstrap.php | Підключення CSS та JS |
| footer.php | Файл з футером |
| menu.php | Файл з меню |
| actions.php | Файл з обробниками подій |
| autoload.php | Файл автозавантаження скриптів |
| database.php | Підключення до бази даних ІСАОТ |
| rb-mysql.php | ПЗ RedBeanPHP |
| utils.php | Реалізація основних функцій ІСАОТ |

Взаємодія компонентів ІСАОТ

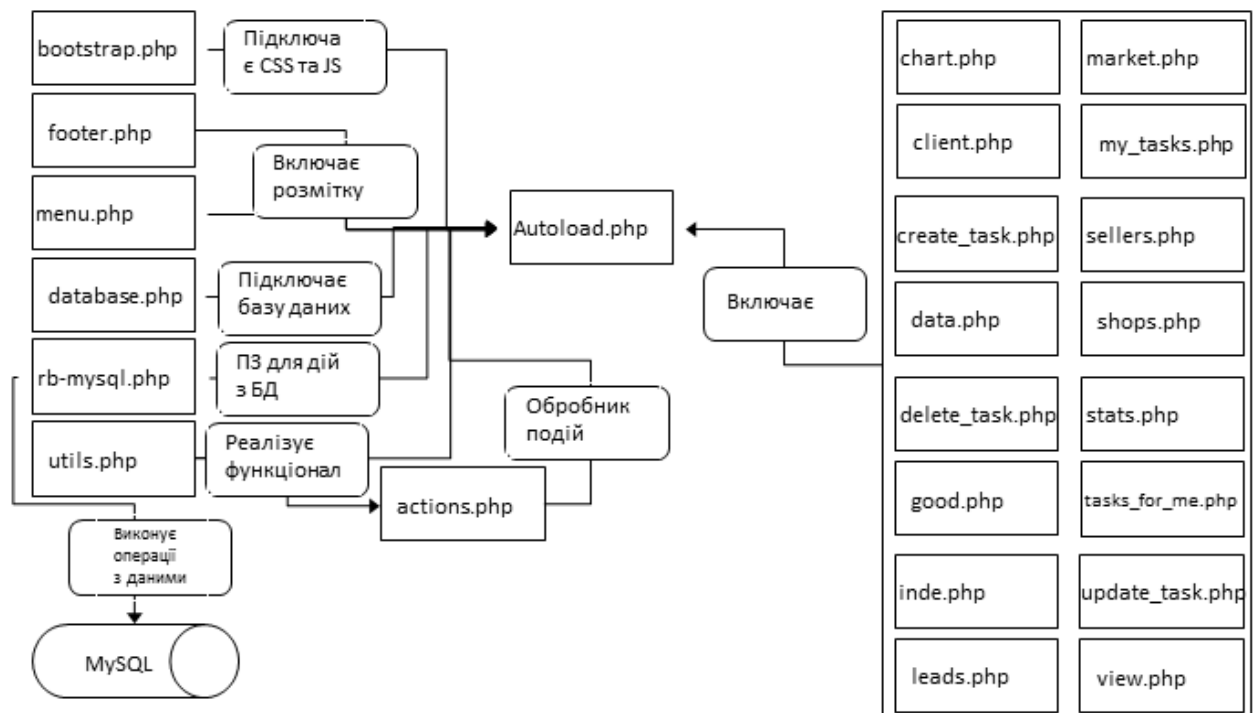


Рисунок Е1 – Схема взаємодії компонентів ІСАОТ

Набори атрибутів інформаційних сутностей

Таблиця Ж1 – Аналіз інформаційних об'єктів

| Інформаційний об'єкт | Атрибути | Характеристики |
|----------------------|---|---|
| Керівник | lead_id lead_name lead_phone lead_email | Дані про керівника магазину (ім'я, телефон, ел. Адреса). |
| Клієнт | client_id | Дані про клієнта (ім'я, телефон, інформація про реєстрацію). |
| | client_name | |
| | client_phone | |
| | client_reg | |
| Магазин | shop_id shop_address sold_sum bought_sum rent_coast | Дані про магазин (адреса, вартість проданої та закупленої продукції, вартість оренди). |
| Наявні товари | article gds_name gds_price gds_count | Дані про наявні товари (артикул, назва, ціна, кількість). |
| Замовлення | order_id client_id order_date article | Дані про замовлення (клієнт, дата, артикул). |
| Рядок замовлення | str_num article client_id gcount rep_id | Дані про покупки (ідентифікатор рядка, артикул, клієнт, кількість певного товару, ідентифікатор звіту). |
| Звіти про замовлення | order_id client_id order_date article | Дані про покупки конкретного клієнта (ідентифікатор звіту, ідентифікатор клієнта, дата, артикул). |

Таблиця Ж2 – Глибинний аналіз об'єктів-сутностей, та їх атрибутів

| Сутність | Атрибут | Ідентифікатор | Ключі |
|------------------|---------------|---------------|-------|
| Керівник | lead_id | lead_id | P |
| | lead_name | lead_name | - |
| | lead_phone | lead_phone | - |
| | lead_email | lead_email | - |
| Продавець | seller_id | seller_id | P |
| | seller_name | seller_name | - |
| | seller_phone | seller_phone | - |
| | shop_id | shop_id | F |
| | seller_email | seller_email | - |
| Клієнт | client_id | client_id | P |
| | client_name | client_name | - |
| | clientr_phone | clientr_phone | - |
| | client_reg | client_reg | - |
| Магазин | shop_id | shop_id | P |
| | shop_address | shop_address | - |
| | sold_sum | sold_sum | - |
| | bought_sum | bought_sum | - |
| | lead_id | lead_id | F |
| | rent_coast | rent_coast | - |
| Наявні товари | article | article | P |
| | shop_id | shop_id | F |
| | gds_name | gds_name | - |
| | gds_price | gds_price | - |
| | gds_count | gds_count | - |
| Замовлення | order_id | order_id | P |
| | client_id | client_id | F |
| | order_date | order_date | - |
| | article | article | - |
| Рядок замовлення | str_num | str_num | P |
| | article | article | F |
| | client_id | client_id | F |
| | gcount | gcount | - |
| | rep_id | rep_id | F |

Таблиця ЖЗ – Аналіз та опис зв'язків між сутностями

| Зв'язок | Назва | Тип | Зміст |
|-----------------------|----------------------|-----|---|
| Керівник-магазин | Контролює / Належить | 1:Б | Керівник може контролювати декілька магазинів. |
| Магазин-продавець | Має / Працює | 1:Б | В одному магазині може працювати кілька продавців. покупець може мати кілька замовлень. |
| Магазин-Наявні товари | Має / Знаходиться | 1:Б | Багато звернень можуть бути прийняті одним працівником; |
| Клієнт-Замовлення | Робить / Належить | 1:Б | Один клієнт може зробити кілька замовлень. |
| Клієнт-Наявні товари | Купує / Належать | Б:Б | Багато клієнтів можуть купити багато товарів. |

Обмеження атрибутів сутностей

Таблиця 31 «Керівники»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _id | Лічильник | P |
| _name | Текстовий | - |
| _phone | Текстовий | - |
| _email | Текстовий | - |

Таблиця 32 «Клієнти»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _id | Лічильник | P |
| _name | Текстовий | - |
| _phone | Текстовий | - |
| _email | Текстовий | - |

Таблиця 33 «Продавці»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _id | Лічильник | P |
| _shop | Числовий | F |
| _name | Текстовий | - |
| _phone | Текстовий | - |
| _email | Текстовий | - |

Таблиця 34 «Магазини»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _id | Лічильник | P |
| _lead | Числовий | F |
| _address | Текстовий | - |
| _sold | Грошовий | - |
| _bought | Грошовий | - |
| _rent | Грошовий | - |

Таблиця 35 «Товари»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _article | Текстовий | P |
| _shop | Числовий | F |
| _name | Текстовий | - |
| _price | Грошовий | - |
| _count | Числовий | - |

Таблиця 36 «Рядок замовлення»

| Атрибути | Обмеження атрибутів | |
|----------|---------------------|------|
| | Тип даних | Ключ |
| _id | Лічильник | P |
| _article | Числовий | F |
| _client | Текстовий | F |
| _count | Числовий | - |
| _date | Дата і час | - |

Запити на створення таблиць

Лістинг И1 – Запити на створення таблицьБД

```
CREATE TABLE client
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _name VARCHAR(50) NOT NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(30) NULL
)ENGINE=InnoDB;
```

```
CREATE TABLE lead
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _name VARCHAR(50) NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(50) NULL
)ENGINE=InnoDB;
```

```
CREATE TABLE shop
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _lead INT NULL,
  _address VARCHAR(50) NOT NULL,
  _sold FLOAT NULL,
  _bought FLOAT NULL,
  _rent FLOAT NULL
)ENGINE=InnoDB;
```

```
CREATE TABLE seller
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _shop INT NOT NULL,
  _name VARCHAR(50) NOT NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(50) NULL
)ENGINE=InnoDB;
```

```
CREATE TABLE goods
(
  _article VARCHAR(6) NOT NULL PRIMARY KEY,
  _shop INT NULL,
  _name VARCHAR(50) NOT NULL,
  _price FLOAT NULL,
  _count INT NULL
)ENGINE=InnoDB;
```

```
CREATE TABLE order_str
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _article VARCHAR(6) NOT NULL,
  _client INT NULL,
  _price FLOAT NULL,
```



```
_count INT NOT NULL,  
_date DATE NULL  
)ENGINE=InnoDB;
```

```
CREATE TABLE users  
(  
_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
_name VARCHAR(50) NOT NULL,  
_phone VARCHAR(10) NULL,  
_email VARCHAR(30) NULL,  
_pass VARCHAR(32) NULL  
)ENGINE=InnoDB;
```

```
ALTER TABLE shop ADD UNIQUE (_id);  
ALTER TABLE seller ADD UNIQUE (_id);  
ALTER TABLE goods ADD UNIQUE (_article);  
ALTER TABLE order_str ADD UNIQUE (_article);  
ALTER TABLE lead ADD UNIQUE (_id);  
ALTER TABLE client ADD UNIQUE (_id);
```

```
ALTER TABLE shop ADD INDEX (_lead);  
ALTER TABLE seller ADD INDEX (_shop);  
ALTER TABLE goods ADD INDEX (_shop);  
ALTER TABLE order_str ADD INDEX (_article);  
ALTER TABLE order_str ADD INDEX (_client);
```

```
ALTER TABLE shop  
ADD FOREIGN KEY (_lead)  
REFERENCES lead (_id)  
ON DELETE CASCADE  
ON UPDATE CASCADE ;
```

```
ALTER TABLE seller  
ADD FOREIGN KEY (_shop)  
REFERENCES shop (_id)  
ON DELETE CASCADE  
ON UPDATE CASCADE ;
```

```
ALTER TABLE goods  
ADD FOREIGN KEY (_shop)  
REFERENCES shop (_id)  
ON DELETE CASCADE  
ON UPDATE CASCADE ;
```

Процедура побудови форми реєстрації

```
function createSignUpForm(selector){
  let form = createForm(
    {method: 'POST',action: 'index.php', name: 'signup_form'},
    ['text', 'email', 'password', 'password', 'submit', 'reset'],
    ['login', 'email', 'password', 'repeat_password', 'signup', 'cleaner'],
    ['Логін', 'Email', 'Пароль', 'Пароль ще раз', 'Реєстрація', 'Очистити'],
    ['form-control', 'form-control', 'form-control', 'form-control', 'btn btn-
success', 'btn btn-danger'],
    ['Логін:', 'Email:', 'Пароль:', 'Пароль знову:', '', '']
  );
  let app = document.querySelector(selector);
  app.appendChild(form);
}
```

Тестування збережувальних процедур

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0575 сек.)

```
CALL update_lead ( 1, "lead_update", "222", "email12@email" )
```

[[Построчное редактирование]] [[Изменить]] [[Создать PHP-код]]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0008 сек.)

```
SELECT * FROM lead
```

Профилирование [[Построчное редактирование]] [[Изменить]] [[Анализ SQL запроса]] [[Создать PHP-код]] [[Обновить]]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _id | _name | _phone | _email |
|--------------------------|-----|-------------|--------|---------------|
| <input type="checkbox"/> | 1 | lead_update | 222 | email12@email |

Рисунок Л1 – Результат виконання ЗП update_lead

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0568 сек.)

```
CALL create_shop(1, "address1", 100.1, 140.7, 50 )
```

[[Построчное редактирование]] [[Изменить]] [[Создать PHP-код]]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM shop
```

Профилирование [[Построчное редактирование]] [[Изменить]] [[Анализ SQL запроса]] [[Создать PHP-код]] [[Обновить]]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _id | _lead | _address | _sold | _bought | _rent |
|--------------------------|-----|-------|----------|-------|---------|-------|
| <input type="checkbox"/> | 1 | 1 | address1 | 100.1 | 140.7 | 50 |

Рисунок Л2 – Результат виконання ЗП create_shop

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0524 сек.)

```
CALL update_shop( 1, 1, "address_edited", 101.6, 130.9, 70 )
```

[[Построчное редактирование]] [[Изменить]] [[Создать PHP-код]]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM shop
```

Профилирование [[Построчное редактирование]] [[Изменить]] [[Анализ SQL запроса]] [[Создать PHP-код]] [[Обновить]]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _id | _lead | _address | _sold | _bought | _rent |
|--------------------------|-----|-------|----------------|-------|---------|-------|
| <input type="checkbox"/> | 1 | 1 | address_edited | 101.6 | 130.9 | 70 |

Рисунок Л3 – Результат виконання ЗП update_shop

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1523 сек.)

```
CALL create_seller ( 1, "seller1", "1111", "email@gmail.com" )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0008 сек.)

```
SELECT * FROM seller
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки:

+ Параметры

| | <u>_id</u> | <u>_shop</u> | <u>_name</u> | <u>_phone</u> | <u>_email</u> |
|-----------------------------------|------------|--------------|--------------|---------------|------------------|
| <input type="checkbox"/> Изменить | 1 | 1 | seller1 | 1111 | email1@gmail.com |

Рисунок Л4 – Результат виконання ЗП create_seller

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0495 сек.)

```
CALL update_seller ( 1, 1, "seller_edited", "2222", "email_edited@email" )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM seller
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки:

+ Параметры

| | <u>_id</u> | <u>_shop</u> | <u>_name</u> | <u>_phone</u> | <u>_email</u> |
|-----------------------------------|------------|--------------|---------------|---------------|--------------------|
| <input type="checkbox"/> Изменить | 1 | 1 | seller_edited | 2222 | email_edited@email |

Рисунок Л5 – Результат виконання ЗП update_seller

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0558 сек.)

```
CALL create_client ( "client1", "1111", "email@email" )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0008 сек.)

```
SELECT * FROM client
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки:

+ Параметры

| | <u>_id</u> | <u>_name</u> | <u>_phone</u> | <u>_email</u> |
|-----------------------------------|------------|--------------|---------------|---------------|
| <input type="checkbox"/> Изменить | 1 | client1 | 1111 | email@email |

Рисунок Л6 – Результат виконання ЗП create_client

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0615 сек.)

```
CALL update_client ( 1, "client_edited", "2222", "email_edited@email" )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM client
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _id | _name | _phone | _email |
|--|-----|---------------|--------|--------------------|
| <input type="checkbox"/> Изменить Копировать Удалить | 1 | client_edited | 2222 | email_edited@email |

Рисунок Л7 – Результат выполнения ЗП update_client

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0602 сек.)

```
CALL create_goods ( "000001", 1, "goods1", 10.5, 5 )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0005 сек.)

```
SELECT * FROM goods
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _article | _shop | _name | _price | _count |
|--|----------|-------|--------|--------|--------|
| <input type="checkbox"/> Изменить Копировать Удалить | 000001 | 1 | goods1 | 10.5 | 5 |

Рисунок Л8 – Результат выполнения ЗП create_goods

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0709 сек.)

```
CALL update_goods ( "goods2", 11, 2, "000001" )
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM goods
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

| | _article | _shop | _name | _price | _count |
|--|----------|-------|--------|--------|--------|
| <input type="checkbox"/> Изменить Копировать Удалить | 000001 | 1 | goods2 | 11 | 2 |

Рисунок Л9 – Результат выполнения ЗП update_goods