

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Комп'ютерні науки
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної роботи

магістр

(освітній рівень)

на тему: **Розробка інформаційної системи керування готелем з
автоматизованим багатокритерійним підбором номерів**

Виконав: студент 6 курсу, групи СТМ-61
спеціальності 126 «Інформаційні системи та
технології»
(шифр і назва спеціальності)

_____ **Дідич Н.В.**
(підпис) (прізвище та ініціали)

Керівник _____ **Приймак М.В.**
(підпис) (прізвище та ініціали)

Нормоконтроль _____ **Мацюк О.В.**
(підпис) (прізвище та ініціали)

Рецензент _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційної системи керування готелем з автоматизованим багатокритерійним підбором номерів // Дипломна робота освітнього рівня "Магістр" // Дідич Наталя Василівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТМ-61 // Тернопіль, 2019 // С. , рис. – , табл. – , додат. – , бібліогр. – .

Ключові слова: СИСТЕМА КЕРУВАННЯ ГОТЕЛЕМ, БАГАТОКРИТЕРІЙНИЙ МЕТОД, РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ, АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ.

Дипломна робота присвячена спрощенню процесу бронювання готелю, а також забезпеченню легкого керування ним.

Актуальність роботи проявляється в спрощенні керування даними. Сучасний готель є дуже складним у функціонуванні, і тому не дивно що успішність роботи закладу залежить від максимально злагодженої роботи правників, та швидкого реагування курівництва на проблемні моменти. Багато власників бізнесу зрозуміли що максимально вирішити цю проблему зможе лише впровадження автоматизованої системи керування готелем. Згідно статистики більшість європейських готельних комплексів перейшли на новіші версії програмних продуктів, оскільки кожен власник боїться залишитись останнім в гонці технічного прогресу та конкуренції. Таким чином, якщо давніше готельні комплекси змінювали програмне забезпечення в період 7-9 років, то зараз тенденції стрімко міняються і заради того щоб завчасно не зійти з ринку власники скорочують терміни заміни до 3-4 років, але тенденція зменшення часу зберігається.

В першому розділі дипломної роботи розглянуто автоматизовані системи управління. Було описано впровадження автоматизованих систем в готельному бізнесі та інших сферах людської діяльності. Узагальнення всіх переваг та недоліків вже готових програмних продуктів, які є на ринку. Також було описано найбільш потрібні функції та модулі для розробки актуального програмного забезпечення. В кінці розділу було детально розглянуто речі, які слід враховувати під час вибору програмного забезпечення для управління готелями.

В другому розділі дипломної роботи розглянуто технології для здійснення розробки програмного продукту

В третьому розділі дипломної роботи розглянуто цілі та вимоги розробки, здійсненне проектування системи керування готелем, де описана діаграма класів, прецедентів, баз даних та сама структура програми.

В четвертому розділі дипломної роботи розглянуто набір рекомендацій для розробників на PHP – PHP Standards Recommendations, який слугує стандартизації концепцій програмування в PHP.

Об'єкт дослідження: процес прийняття рішення щодо вибору, найбільш відповідного до вимог клієнта, номеру.

Предмет дослідження: алгоритм багатокритеріального підбору номерів.

Мета роботи: прощення процесу бронювання готелю, а також забезпечення легкого керування ним.

Основні результати: застосування методу автоматизованого багатокритеріального підбору номерів на прикладі системи керування готелем. Результати роботи можуть бути впроваджені в процес розробки будь якої системи керування.

ANNOTATION

Hotel management information system development with automated multi-criteria rooms selection // Diploma work degree “Master” // Didych Natalia // Ternopil Ivan Pul'uj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science, STm-61 // Ternopil, 2019 // P. , fig. - , Table. - , add . - , bibliography . - .

Keywords: HOTEL CONTROL SYSTEM, MULTICRITERIA METHOD, SOFTWARE DEVELOPMENT, AUTOMATED CONTROL SYSTEM.

This thesis is dedicated to simplify the process of booking a hotel and provide easy management.

The urgency of work is manifested in the simplification and management of data. A modern hotel is very difficult to operate, so it is not surprising that the success of the institution depends on the most coordinated work of lawyers, and prompt response of the management to the problematic moments. Many business owners have realized that only the introduction of an automated hotel management system can solve this problem. According to statistics, most European hotel complexes have switched to newer versions of software, as each owner is afraid to remain the latest in the race of technological progress and competition. Thus, if earlier hotel complexes changed the software in the period of 7-9 years, now the trends are changing rapidly and in order not to leave the market in advance the owners shorten the replacement period to 3-4 years, but the trend of decreasing time remains.

The first section of the thesis deals with automated control systems. The introduction of automated systems in the hotel business and other areas of human activity has been described. Generalization of all the advantages and disadvantages of ready-made software products available on the market. The most required features and modules for developing the latest software were also described. At the end of the section, we discussed in detail the things to consider when choosing hotel management software.

The second section of the thesis deals with technologies for the implementation of software development. In the first part of the section, the programming language for writing the software-PHP was selected, the hypertext markup language HTML and the database management system, which allows managing relational databases - MySQL, were selected to display the materials in the work. The second part of the section discusses the software used for development. The IDE Eclipse was selected directly to write the project code

The third section of the thesis deals with the goals and requirements of the development, the implementation of the design of the hotel management system, which describes the diagram of classes, precedents, databases and the structure of the program. In the main part of the section, a code review of several major product modules was carried out, as well as an analysis of the multicriteria automated number dialing method and its implementation. The end of the section details the testing of the program with the illustrations below. The fourth section of the thesis deals with a set of recommendations for developers in PHP - PHP Standards Recommendations , which serves to standardize programming concepts in PHP. The Master's thesis "Hotel Management System with Automated Multicriteria Dialing" used these standards to help you quickly understand the code and systematize it. The purpose of the section is to ensure interoperability of the components and to create a common technical base for the implementation of proven concepts of optimal programming and testing methods for writing the master's code.

Object of study: the process of deciding on the number most appropriate to the client's requirements.

Subject of research: algorithm of multicriteria number selection.

Purpose: To simplify the process of booking a hotel, as well as to provide easy management.

The main results: the application of the method of automated multi-criteria numbering on the example of a hotel management system. The results of the work can be integrated into the process of developing any control system.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSS – (Cascading Style Sheets) - каскадні таблиці стилів.

FTP (File Transfer Protocol) - протокол передачі файлів.

HTML (HyperText Markup Language - Мова розмітки гіпертексту) - стандартна мова розмітки документів у Всесвітній павутині.

MySQL — вільна система керування реляційними базами даних.

PHP (Personal Home Page Tools) - скриптова мова програмування.

WWW (World Wide Web) – глобальна комп'ютерна мережа, всесвітня павутина.

IDE – інтегроване середовище розробки.

ПЗ – програмне забезпечення.

СКБД – система керування базами даних.

АСУ (АСК) – Автоматизована системи управління (керування).

АС – Автоматизована система.

ЗМІСТ

Вступ.....	9
1.3.1 PMS в готельному програмному забезпеченні	19
1.3.2 Види програмного забезпечення готелів.....	20
1.3.3 Потрібні функції програмного забезпечення для управління готелями	21
1.3.4 Переваги програмного забезпечення готелів.....	26
1.3.5 Модулі програмного забезпечення	27
1.3.6 Речі, які слід враховувати під час вибору програмного забезпечення для управління готелями	30
2 ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ ..	32
2.1 Огляд технологій.....	32
2.2 Програмне забезпечення, що використовується для розробки.....	37
3 розробка програмного забезпечення.....	42
3.1 Загальні відомості	42
3.2 Призначення та цілі розробки.....	42
3.3 Вимоги до розробки.....	42
3.5 Застосування методу багатокритеріального підбору	48
3.7 Проектування бази даних	55
3.9 Опис програмної реалізації	63
3.8 Опис класів	67
4 СПЕЦІАЛЬНА ЧАСТИНА	74
4.1 PSR-0 – стандарт автозавантаження.....	74
4.2 PSR-1 – базовий стандарт оформлення коду	75
4.3 PSR-2 – рекомендації по оформленні коду	77
4.4 PSR-3 – інтерфейс протоколювання	79
4.5 PSR-4 – покращення автозавантаження.....	80
4.6 Висновок до четвертого розділу	81
5 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....	82
6 ЕКОЛОГІЯ.....	94
7 Охорона праці та безпека в надзвичайних ситуаціях.....	100

ВСТУП

Актуальність теми дослідження. На даний момент готелі рідко використовують подібні системи для спрощення керування даними. Уся інформація про бронювання, зазвичай, знаходиться в паперовому вигляді, а залишити заявку на бронювання можна лише у телефонному режимі. У такому випадку на отримання необхідної інформації клієнти витрачають зайвий час, а також відволікають адміністратора, більше того не всі клієнти надають перевагу телефонним дзвінками. Також важливою є звітність, адміністратор повинен знати які кімнати є популярніші за інші. В даному програмному забезпеченні є можливість перегляду звітів.

З введенням подібної системи у роботу буде вирішено вище згадані проблеми, а також забезпечено адміністраторам системи зручне, швидке і інтуїтивно зрозуміле керування базою бронювання.

Мета і завдання дослідження. Метою магістерської роботи є спрощення процесу бронювання готелю, а також забезпечення легкого керування ним.

Для досягнення цієї мети необхідно:

- дослідити предметну область та здійснити постановку проблеми;
- побудувати модель автоматизованої системи;
- реалізувати програмне забезпечення для керування базою бронювання;
- провести тестування розробленої системи.

Для зручності клієнтів було обрано скриптову мову PHP, яка була створена для генерації HTML-сторінок на стороні веб-сервера. Таким чином клієнт може залишити заявку на бронювання з будь-якого пристрою (ПК, планшета, смартфона), який підтримує доступ до мережі Інтернет.

Наукова новизна отриманих даних полягає в вдосконаленні методу підбору номерів шляхом визначення найбільш відповідного до вимог та найдешевшого з допустимо можливих номерів готелю за допомогою

багатокритеріального методу підбору.

Об'єктом дослідження є процес прийняття рішення щодо вибору, найбільш відповідного до вимог клієнта, номеру.

Предметом дослідження є алгоритм багатокритеріального підбору номерів.

Практичне значення отриманих результатів. Застосування методу автоматизованого багатокритеріального підбору номерів на базі системи керування готелем. Результати роботи можуть бути впроваджені в процес розробки будь якої системи керування.

1 ТЕНДЕНЦІЇ РОЗВИТКУ АВТОМАТИЗОВАНИХ СИСТЕМ КЕРУВАННЯ

1.1 Огляд автоматизованих систем керування

На даний час широкого росту у всіх сферах діяльності людини набули автоматизовані системи (АС). Без них не можливо уявити сучасні сфери такі як виробництво, наука, медицина, техніка та освіта (наведено на рисунку 1.1). Розвиток новітніх технологій поклав початок появі новітніх методів опрацювання та передачі інформації. Для якіснішого та швидшого опрацювання інформації і потрібне створення автоматизованих систем за різним функціоналом.

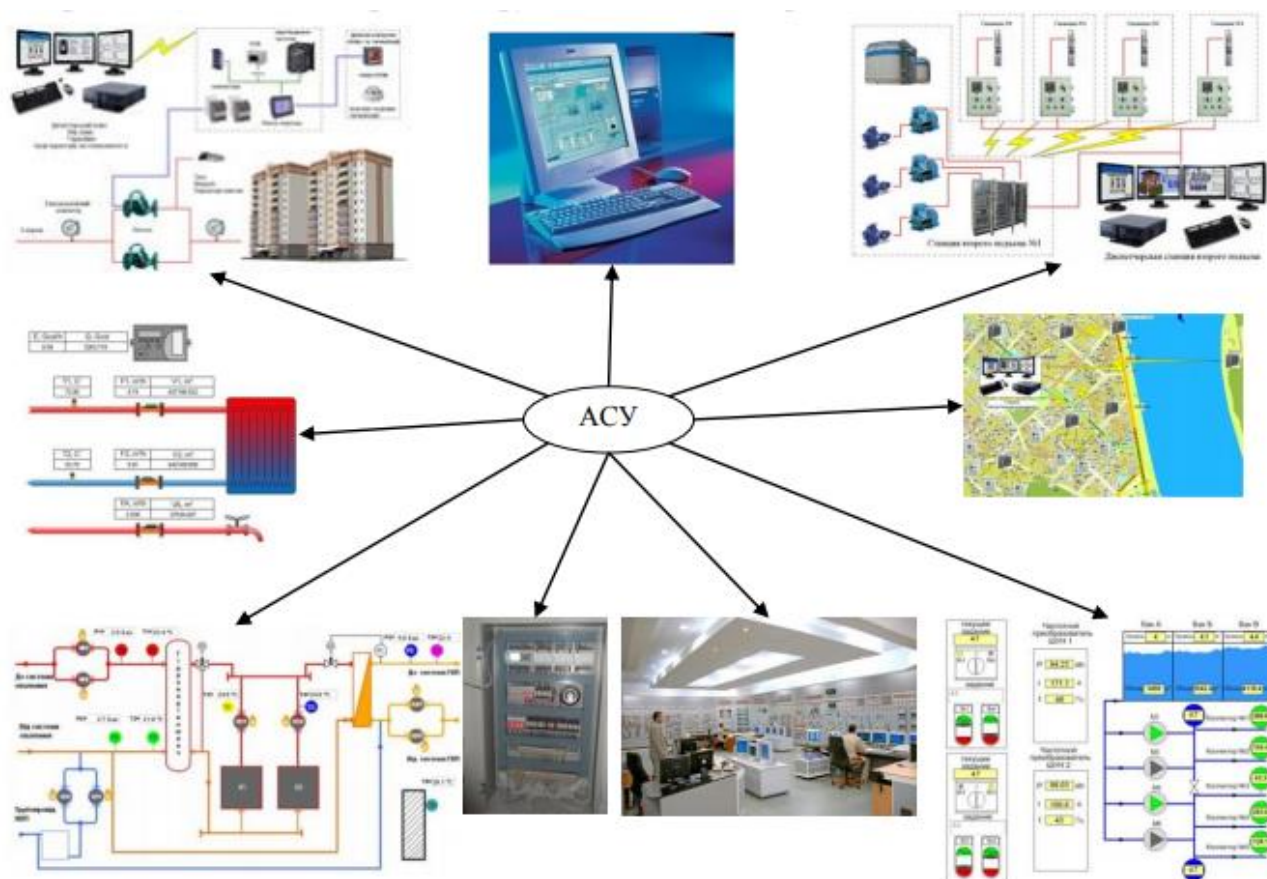


Рисунок 1.1 – Галузі використання АС

В даний час на кожному підприємстві появляється проблема

оптимізації та керування даними, що в свою чергу покращує ефективність роботи. Для того щоб ефективно керувати організацією та оптимально виконувати робочі процеси для керівників потрібна актуальна та достовірна інформація. Цю проблему в теперішній час можуть вирішити засоби і методи автоматизації інформаційних потоків (рисунок 1.2).

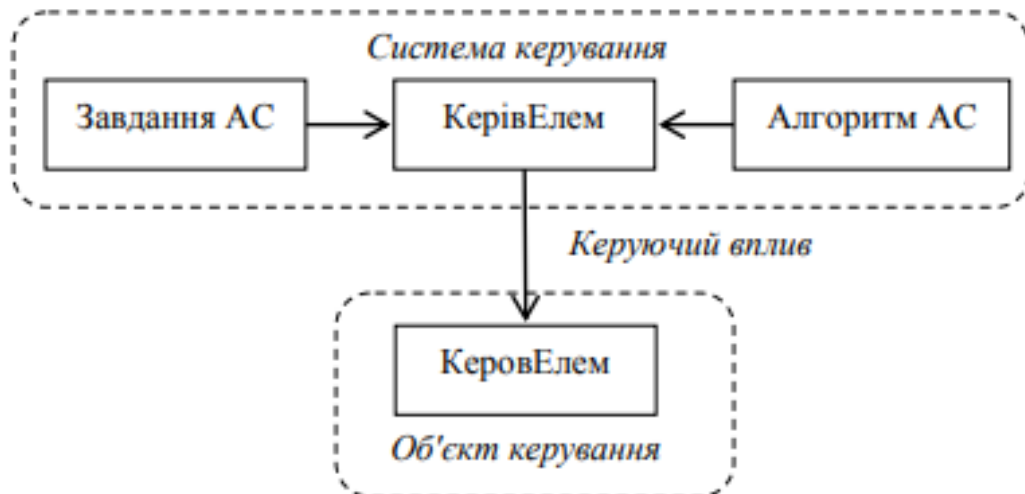


Рисунок 1.2 – Діаграма засобів і методів автоматизації інформаційних потоків

У всіх системах існує 2 види управління це об'єкт (рисунок 1.3), якому дають задачі (верстат, підприємство, галузь) і орган, який дає ці задачі (технічний засіб, людина). Протягом процесу роботи управляючий орган отримує інформацію щодо зовнішнього стану середовища, де на даний момент присутній об'єкт і з яким він взаємодіє. Уся інформація направляється до управляючого органу, який в свою чергу прораховує на її основі всі варіанти та приймає рішення. Після прийнятого рішення виконавчий орган починає вплив на об'єкт управління.

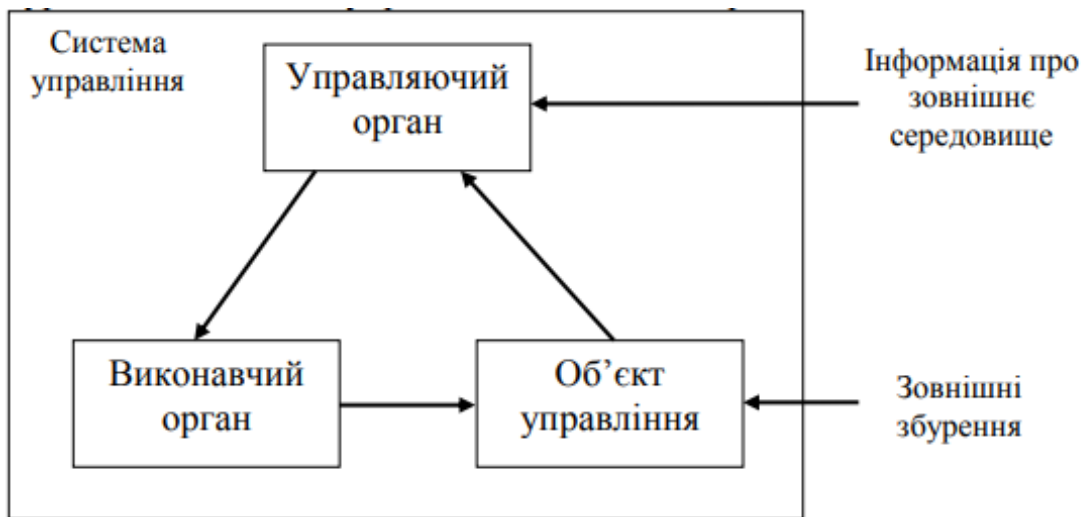


Рисунок 1.3 – Діаграма системи управління

Всі автоматизовані системи управління під час своєї роботи повинні здійснювати наступні дії:

- отримання, опрацювання та виконання аналізу даних (сигналів, повідомлень, документів тощо) про конкретне становище об'єкта управління;
- постановка впливових рішень (програм, планів і т.д.);
- передачу даних (сигналів, вказівок, документів) на виконання і контроль їх передачі;
- виконання і перевірка реалізації поставлених задач;
- інформаційний обмін (документами, повідомленнями тощо) з суміжними автоматизованими системами.

Система автоматичного керування ACS MP призначена для: автоматизації збору даних про складні комплекси технологічного обладнання та для управління виробничим процесом, підтримання оптимального режиму роботи технологічних апаратів та обліку проміжних даних, формування та виведення бухгалтерська та архівна документація, діагностика вимірювальної апаратури.

Гнучка модульна структура ACS MP дозволяє забезпечити для кожного технологічного об'єкта оптимальний рівень автоматизації, достатній для ефективного та безпечного проведення виробничого процесу.

До розробки зазвичай пропонують адаптацію програмного забезпечення ACS MP за замовленнями клієнтів з посиланням на конкретні об'єкти та обладнання.

Впровадження даної системи дозволяє:

- автоматизувати виробничий процес;
- забезпечити дистанційне управління та управління;
- покращити умови та підвищити безпеку праці.

1.1.1 Класифікація АСУ

Велика кількість АС повинна бути систематизована, тому найкраще в цьому може допомогти класифікація за різними найбільш поширенішими ознаками. Класифікація помагає більш покроково здійснити вирішення найбільших проблем проектування АС, оскільки підбір технічних засобів залежить від того до якого класу належить АС.

Класифікація АС може бути дуже різним. Зазвичай воно підбирається за різними ознаками, тому в значній мірі воно є доволі умовним.

Приклади класифікацій АСУ за поширеними признаками:

1. Сфера діяльності: державні, територіальні (регіональні), галузеві, підприємств або організацій, технологічних процесів.
2. Рівень автоматизації процесів: інформаційно-пошукові, інформаційно-керівні, інформаційно-довідкові, СППР, інтелектуальні, експериментальні.

Розглянемо декілька пунктів класифікації та дамо їм наступний підподіл.

- За призначенням або особливістю об'єктів управління: адміністративні, виробничо-технічні, соціальні, транспортні тощо.
- За ступенем централізації обробки інформації: централізовані, децентралізовані, колективного використання.
- За характером керованого виробничого процесу: для виробництва з дискретним процесом (машинобудування, приладобудування); з неперервним

процесом (хімічна, нафтопереробна і газова промисловості); з неперервно дискретним процесом (металургійна, цементна, харчова промисловість).

- За ступенем інтеграції функції: багаторівневі з інтеграцією за рівнями планування, багаторівневі з інтеграцією за рівнями управління; комплексні (інтеграція з відмінними задачами).

1.1.2 Впровадження автоматизованих систем управління

У 1963 році розпочалось створення та аналіз автоматизованих інформаційних систем. Це був рік введення у великих підприємствах використання ЕОМ (електроннообчислювальні машини) для опрацювання побутовоекономічного керування. А наступні роки розпочалось покрокове покращення інформаційних систем (приклад наведений і таблиці 1.1).

Табличка 1.1 - Етапи створення систем управління.

Номер етапу	Період, роки	Назва етапу в нашій країні	Назва етапу в іноземній літературі
Перший	1963–1972	Створення АСУ (позадачний підхід)	Системи обробки даних
Другий	1972–1985	Створення і розвиток АСУ згідно з концепцією баз даних	Управлінські інформаційні системи
Третій	Початок 1985 (триває досі)	Створення інтегрованих АСУ, обчислювальних систем і мереж	Системи підтримки прийняття рішень

У 80-х роках в організаційному управлінні, було створено цілу систему автоматизованої побудови технологій. Це стало причиною введення у керування підприємства програмних АСУ.

Найбільш використовувані процеси автоматизованих систем управління:

- планування і прогнозування;
- облік, контроль, аналіз;
- координацію і регулювання.

За характером об'єкта управління АСУ поділяються на:

- АСУ підприємства (АСУП);
- АСУ технологічними процесами (АСУ ТП);
- АСУ територіальними організаціями;
- АСОУ (автоматизована система організаційного управління) – для управління колективами людей в економічних і соціальних системах.
- АСОД (автоматизована система обробки інформації);
- АСНТД – АС управління науково-технічними дослідженнями[5].

1.2 АСУ в готельному бізнесі

На сьогоднішній час основною візитівкою готелю стає не матеріальні чинники (ціна, інтер'єр номерів), а дещо більш особистніше, таке як рівень сервісу, індивідуальний підхід до клієнта, максимальний комфорт гостя. В погоні за першістю і розвиває дану сферу, що не може не тішити клієнтів. Тому найбільшою тенденцією сьогодні є максимальне прискорення часу роботи з клієнтом. Наприклад, реєстрація, виїзд, бронювання також індивідуальний підхід та врахування всіх побажань клієнта.

Сучасний готель є дуже складним у функціонуванні, і тому не дивно що успішність роботи закладу залежить від максимально злагодженої роботи правників, та швидкого реагування курівництва на проблемні моменти. Отже кожний власник такої нелегкої справи хоче отримати максимум від своєї діяльності, і старається здивувати своїх клієнтів. Найбільш оціненим плюсом теперішніх готельних комплексів є забезпечення оперативності і точності роботи персоналу та готельного комплексу в цілому. Багато власників бізнесу зрозуміли що максимально вирішити цю проблему зможе лише впровадження автоматизованої системи керування готелем (Property Management System (PMS)).

Автоматизована система керування для готелів пов'язує в собі не один комплекс підсистем, які взаємодіють між собою. Це створюється для

полегшення роботи персоналу, оскільки вони зможуть на пряму взаємодіяти з туристичними агентствами, туроператорами, ресторанами та іншими корпоративними клієнтами. Згідно статистики більшість європейських готельних комплексів перейшли на новіші версії програмних продуктів, оскільки кожен власник боїться залишитись останнім в гонці технічного прогресу та конкуренції. Таким чином, якщо давніше готельні комплекси змінювали програмне забезпечення в період 7-9 років, то зараз тенденції стрімко міняються і заради того щоб завчасно не зійти з ринку власники скорочують терміни заміни до 3-4 років, але тенденція зменшення часу зберігається, оскільки ринок готельного бізнесу має доволі жорстоку конкуренцію.

На даний момент нойбільшої популярності мають системи: "Micros Fidelio", Hospitality Enterprise Resource Planning "Cenium", Epitome PMS, Amadeus PMS, OPERA, "KEI Hotel", "UCS Shelter".

Зазвичай більшість європейських імпортерів програмних продуктів мають свої версії автоматизованих систем керування, яку використовують для дистанційного керування продуктом [4].

Програмне забезпечення систем управління готелем спрямоване на те, щоб звільнити персонал та керувати їх життєвими завданнями, автоматизуючи їх і синхронізувавши їх між різними системами, що використовуються у вашій щоденній роботі. До завдань, які можна вирішити, належать:

- автоматизація бронювання
- управління відносинами з гостями
- управління людськими ресурсами
- автоматизація аналітики

Залежно від потреб та специфіки вашого готельного бізнесу необхідні функції та їх реалізація можуть відрізнятися. У багатьох компаніях пропонують розробку будь-якого розміру рішення з нуля, а також

допомагають на етапі перед розробкою у створенні бізнес-аналізу та оновлення, обслуговування програмного забезпечення після розробки.

До переваг розробки спеціального програмного забезпечення для управління готелями відноситься те, що отримує клієнт:

- Унікальні функціональні можливості, які відповідають потребам бізнесу, не йдучи на компроміс, як у випадку купівлі позапрограмного забезпечення.

- Безпечна і надійна система, яка гарантує, що всі збережені конфіденційні дані захищені від хакерів, зловмисних програм або проблем із сервером.

- Гнучке та масштабоване рішення, яке можна змінити в міру розширення бізнесу.

А середині 90-х років на український ринок зайшли багато автоматизованих систем керування готелем. Проте програмне забезпечення вітчизняного бізнесу значно поступалося західним продуктам. Розробка таких продуктів може вартувати близько 110 мільйонів доларів щорічно, але такі вклади вважаються вигідними з точки зору прибутковості одного готелю.

Багато власників бізнесу бояться інвестувати свої кошти в технічні продукти, оскільки не впевнені в їх результативності. Але є базові критерії за якими можна оцінити доцільність інвестицій в АСУ:

1. Підвищення конкурентної переваги.
2. Покращення продуктивності роботи.
3. Найраціональніше використання ресурсів готелю.

Якщо ці критерії доцільності співпадають з бажаннями клієнта інвестиція в автоматизовану систему управління готелями буде одним з найкращих рішень замовника. Отже, застосування автоматизованої системи управління сприяє покращенню рівня обслуговування і управління персоналом та підвищенню продуктивності конкретного підприємства.

1.3 Програмне забезпечення готелів

Найкраще програмне забезпечення для управління готелем - це ідеальне рішення для підтримки гостинності та управлінні відносинами з клієнтами в готелі. Система управління готелем може бути використана для управління номерами для взаємодії з гостями в індустрії гостинності, як у ресторанах, гуртожитках, будиночках, апартаментах, курортах тощо.

Технологія - це не те, чого повинні боятися в цьому часі, а те, чого повинні навчитися сприймати. Це може по-справжньому перетворити досвід для клієнтів і відрізнити готель від усіх інших.

Програмне забезпечення для системи управління - це програмне рішення, розроблене для управління готельєрами. Він надає просте у користуванні програмне забезпечення для готельних рахунків, програмне забезпечення системи бронювання готелів та онлайн-програмне забезпечення бронювання готелів.

Готельна індустрія процвітає роками. З появою нових підприємців у бізнесі ринок адаптується до умов конкуренції. Для того, щоб керувати готелем та забезпечити його безперебійне функціонування, готелю необхідно включити програмне забезпечення Hotel PMS. Адже готель без програмного забезпечення - це як людське тіло без скелета.

Такі функції, як фронт-офіс, POS, інвентар, нарахування заробітної плати, програмне забезпечення для бронювання готелів в Інтернеті, програмне забезпечення для виставлення рахунків у готелі та інтерфейси сприяють кращим програмним рішенням HMS для готелів, роблячи його повноцінною системою, яка стає основою будь-якого готелю.

1.3.1 PMS в готельному програмному забезпеченні

Система управління майном у програмному забезпеченні готелю піклується про онлайн-систему бронювання готелів та інші адміністративні завдання. Однією з найважливіших функцій, яку виконуватиме програмне

забезпечення PMS готелю, є операції на стійці реєстрації, бронювання, управління каналами, ведення господарства, управління тарифами та заповненням, обробка платежів тощо. Програмне забезпечення PMS готелю також відповідає за контроль за бронюванням і транзакцій. Крім цього, він також допомагає вам вести господарство та виконувати функції, пов'язані з персоналом.

Системи управління готельними властивостями використовуються великими мережами готелів, маленькими хостелами, котеджами, курортами, орендами тощо. Використовуючи ці програмні засоби реєстрації готелів, власники готелів можуть побачити статус бронювання номерів та контролювати це. Їх функціональність не закінчується, використовуючи готельне програмне забезпечення PMS, готельєри також можуть керувати операціями в офісах, послугами харчування та напоїв та відстежувати рівень зайнятості номерів.

1.3.2 Види програмного забезпечення готелів

Існує чотири типи програмного забезпечення для управління готелями:

1. Системи управління нерухомістю - Системи управління власністю готелів відповідають за керування щоденними видами діяльності, такими як приймання бронювання, скасування бронювання та створення щоденних маніфестів. Це ефективний інструмент для управління готелем будь-якого розміру.

2. Менеджери каналів - Менеджер каналів - це програма, яка допоможе власникам готелів зв'язатися з різними агентами. Ці агенти мають інформацію в реальному часі про наявність номерів, і на їх основі вони можуть перепродавати ваші кімнати з більшою ефективністю за комісію.

3. Онлайн-бронювання - Це особливий тип інформаційної системи готелів, який допоможе вам приймати бронювання безпосередньо з веб-сайту. Це обов'язковий компонент майже для кожного типу готелю.

4. Інструменти ціноутворення - інструменти ціноутворення допомагають вам розробити ефективну стратегію управління доходами. Вам більше не доведеться турбуватися про зміну ставок вручну або обчислення доходу, який слід отримувати на кожній кімнаті, ви можете використовувати ці інноваційні інструменти ціноутворення, щоб автоматизувати процес і позбутися цих виснажливих завдань.

1.3.3 Потрібні функції програмного забезпечення для управління готелями

1. Панель приладів:

Після входу в програмне забезпечення для управління готелем перше, що вам слід побачити, - це привітна, корисна інформаційна панель. На приладовій панелі ви зможете зробити короткий огляд усіх заходів, які проводяться у вашому готелі. Він надасть вам таку інформацію, як зайнятість, кількість заїздів, виїзних номерів, наявність вільного персоналу тощо.

2. Підтримка прямого бронювання:

Онлайн-канали розповсюдження можуть допомогти вам розширити свій бізнес, але вони коштують, і ці витрати збільшуються зі збільшенням відсотків комісійних. Але якщо мандрівникові вдасться знайти ваш веб-сайт, ви можете легко зафіксувати їхнє пряме бронювання. Клієнти готові заповнити форму запити і чекати, коли ви повернетесь назад. Хороший механізм бронювання вашого програмного забезпечення для рахунків у готелі дозволить вашому гостю вивчити різні варіанти, перевірити наявність, здійснити оплату та отримати інформацію за один сеанс.

Навіть великі мережі готелів, які мають міцні стосунки з онлайн-турфірмами, протягом останніх кількох років зробили потужний поштовх до двигуна прямого бронювання.

3. Найкращі доступні ціни (BAR):

BAR допомагає вам контролювати свої тарифні плани. Використовуючи функціонал BAR, ви можете керувати рекламними

акціями, ставками працівників та легко реагувати на ринкові умови. Ваше програмне забезпечення для управління готелями повинно бути достатньо ефективним, щоб забезпечити VAR відповідно до різних сезонів, тобто встановити їх вище для напружених сезонів і навпаки.

Якщо у вас в Інтернеті система управління готелем функціонує VAR, вам не доведеться турбуватися про вручну змінювати кожен ціну. Натомість ви можете встановити потрібні тарифи на певний проміжок часу, а потім залишити так.

Поряд з VAR, програмне забезпечення для готельних номерів дозволяє легко змінювати ціни та пакети. Створення промо-кодів та спеціальних тарифів не повинно бути великим завданням.

4. Прибирання та обслуговування:

Вам, мабуть, цікаво, яке програмне забезпечення готельної реєстрації стосується ведення господарства та обслуговування?

Використовуючи програмне забезпечення для реєстрації готелів, ви можете відстежувати номери, які потребують прибирання чи якісь інші препарати до приїзду гостей. Мало того, що він також використовується для відстеження кімнат, які прибираються в даний день.

Якщо говорити про технічне обслуговування програмного забезпечення для управління готелем, то слідкуйте за областями, які потребують уваги, а також планує майбутні ремонти, щоб гарантувати, що водопровід та електроенергія працюють ефективно та ефективно. Програмне забезпечення поєднує ведення господарства та ведення часу, гарантуючи, що послуги надаються гостям в порядку, допомагає менеджеру тримати зусилля готелю впорядкованому в правильному напрямку.

5. Звітність:

Модуль звітності є обов'язковим для будь-якого програмного забезпечення управління готелем. Існує безліч звітів, які обов'язкові для роботи готелю щовечора.

Вони поділяються на три категорії:

- Звіти про виробництво - це допомагає зрозуміти, як працює ваша система онлайн-управління готелем за тарифними тарифами, каналом розповсюдження, ADR та розрахунками кількості. Звіти використовуються для прийняття стратегічних рішень на основі цін на розподіл тощо.

- Фінансові звіти - вони звітують про доходи, податки, комісійні, чистий дохід тощо. Цей звіт допоможе вам зрозуміти, які гроші надходять та виходять. Вони також використовуються для податкової та урядової звітності.

- Щоденні звіти про активність - звіти про прибуття, виїзд та внутрішні записи відстежують гостей та допомагають безперебійно працювати.

6. Спільний інвентар:

Спільний інвентар дозволяє продавати товарний запас у різних форматах. Наприклад, ви можете продати ту саму приватну кімнату, що й декілька різних типів кімнат. Вся мета цього інструменту - продати ту саму приватну кімнату, що й декілька різних типів кімнати. Ви не завжди знаєте, що вимагає кімната з двома королевами або королем або номер люкс. Функція спільного інвентарю у вашій інформаційній системі готелю. Допоможе задовольнити пропозиції, які шукає ринок.

7. База даних клієнтів:

Деякі з найкращих онлайн-систем бронювання готелів також будуть виступати як основна система управління клієнтами. Ваша попередня інформація про відвідувачів може бути чудовим скарбом, яким багато властивостей нехтувати. Ви можете постійно підтримувати зв'язок зі своїм попереднім гостем електронною поштою та інформувати їх про місцеві події, знижки та іншу відповідну інформацію.

Якщо ваша власність достатньо хороша, ця техніка, безумовно, допоможе вам отримати значну кількість гостей, що повернулися. Наявність профілів клієнтів допоможе відслідковувати їх особисту інформацію, що зробить їхнє перебування ще кращим.

8. Керує спілкуванням із гостями:

У вашому програмному забезпеченні для управління готелем повинні бути варіанти, щоб ви могли структурувати спілкування з гостями. Це повинно дозволяти вам надсилати електронні листи до і після прибуття, а також має настроювати шаблони, що відображають ідентичність вашого бренду. Власники готелів під час спілкування з гостем повинні намагатися бути максимально професійними.

Управління готелем вимагає постійної уваги до дрібниць. Ось певні особливості програмного забезпечення для готелів, які дозволять вам отримати контроль над фінансами та загальним управлінням безперебійного робочого процесу.

9. Бронювання готелів та підтримка рецепції:

Програмне забезпечення для реєстрації в готелях у цей час простіше при реєстрації та виїзді з готелів, оскільки це програмне забезпечення дає короткий посібник щодо наявності номерів та інших деталей, пов'язаних із платою. Крім того, загальні відомості про готельні послуги можна побачити в один раз за допомогою програмного забезпечення управління готелем.

10. Шлюз оплати :

Готельні програмні рішення допомагають менеджерам приймати оплату від клієнтів за допомогою кредитних / дебетових карт. Це автоматично оновлюється до їх фінансової бази даних, а також інформації про гостей.

Готелі генерують рахунки та рахунки-фактури, які додають до нього багато деталей. Найкраща система програмного забезпечення для готелів, яка забезпечує перевірку використання готелю зручностей та поєднує їх у рахунку-фактурі, стає ідеальною. Крім того, якщо програмне забезпечення допомагає у фінансових аспектах готелів, це робить користь для керівників готелів.

11. Подарункові купони / сертифікати:

Клієнти можуть викупити свої подарункові купони / сертифікати легко, не вимагаючи нарахування витрат. Програмне забезпечення управління гостинністю негайно визнає вартість талонів і відповідно вносить корективи. Програмне забезпечення також допомагає робити подарункові купони / сертифікати для цінних клієнтів

12. Деталі працівника та менеджера:

Після того, як дані про співробітників будуть введені в програмне забезпечення, це може бути корисним для кількох цілей. Керівник може призначити обов'язки відповідно до переліку, вони можуть міняти обов'язки за необхідності і навіть отримати контактні дані працівника (працівників) у разі надзвичайних ситуацій. Системи програмного забезпечення готелів також допомагають робити особисті посвідчення працівників. Заробітну плату працівників можна періодично перевіряти та навіть оновлювати. Надходження від виплат заробітної плати також отримуються від програмного забезпечення.

Як і у випадку зі співробітниками, навіть деталі керівників оновлюються. Керівникам також видається посвідчення особи.

13. Підтримка роздрібною торгівлі:

Деякі готелі також потребують роздрібних товарно-матеріальних цінностей, щоб вести інвентар господарських товарів та продуктів харчування. Більшість систем програмного забезпечення готелів мають інтегровану роздрібну торговельну інвентаризацію, що значно спрощує процес управління готельними системами.

14. Управління рестораном:

Готелі, які мають власний ресторан, потребують конкретних даних про ресторан. Це включає рахунки-фактури та виставлення рахунків і навіть інвентаризацію. Програмне забезпечення для управління готелями, яке має окрему область в рамках свого програмного пакету, може принести величезну користь готелям.

1.3.4 Переваги програмного забезпечення готелів

Програмне забезпечення для готельних рахунків надає такі переваги для організації готелю:

1. Централізоване управління.

Програмне забезпечення готелю забезпечує централізований контроль. Менеджер готелю може легко регулювати всі невеликі модулі програмного забезпечення готелю з одного інтерфейсу. Це допомагає йому керувати всім щоденним функціонуванням готелю. Щоденне, що відбувається в кожній одиниці готелю, можна спостерігати в режимі реального часу та покращувати, якщо потрібно. Більшість програмного забезпечення для управління готелями мають функцію налаштування, що дозволяє користувачам вносити зміни або додавати більше функцій відповідно до своїх вимог.

2. Надійна система.

Ще одна основна причина використання невеликого програмного забезпечення для управління готелями - це те, що він забезпечує надійну систему, яка усуває людські вади. Глюки, такі як подвійні бронювання, можуть спричинити серйозні проблеми, якщо резервування записується на папері. З системою програмного забезпечення для бронювання готелів це вже не проблема! Усі бронювання миттєво та цифрово задокументовані за допомогою програмного забезпечення управління бронюванням готелів.

3. Додано зручності.

Програмне забезпечення може зробити процедуру бронювання набагато більш придатною для людей, які шукають кімнати. Вони можуть перевірити, які номери є, а потім зробити бронювання. Програмне забезпечення допомагає людям вибирати приміщення з необхідними для них послугами. Це також дозволяє глобальним гостям здійснити бронювання, не стягуючи плату за транснаціональний телефонний дзвінок.

4. Генерація рахунків-фактур та звітів без стресу.

Хмарне програмне забезпечення готелів записує бронювання, гостей, транзакції, рахунки тощо. Це дозволяє зберігати записи та рахунки, тим самим полегшуючи процес створення звітів. Готельне програмне забезпечення легко виробляє рахунки-фактури. Створення точних та звітів у режимі реального часу - це легкий досвід із цим програмним забезпеченням.

5. Простий у використанні.

Програмне забезпечення надзвичайно просте у використанні, без необхідності навчання. Інтерфейс інстинктивний, із системами он-лайн бронювання, які круті для відвідувачів. Також персонал готелю може без проблем керувати внутрішніми програмними системами. Багатокористувацькі параметри роблять програмне забезпечення для готельних платежів зручним для користувачів та забезпечують координацію між різними відділами чи командами. Хмарне програмне забезпечення для управління готелями допомагає працівникам готелю працювати більш професійно з меншим стресом і одночасно підвищує їхню продуктивність.

У будь-якій галузі найважливіше - мати правильні інструменти, щоб допомогти цій галузі безперебійно працювати. Те саме стосується готельного господарства та індустрії гостинності. Кожен аспект, що стосується програмних рішень для гостинності, може зайняти багато часу та неефективний, якщо потрібні інструменти та програмне забезпечення не доступні. І програмне забезпечення для управління готелем на основі хмар - це ідеальний «інструмент». Програмне забезпечення допоможе грамотно та успішно працювати в готелі. Настільки, що гості продовжуватимуть повертатися у ваш готель, щоб насолодитися приємною гостинністю та блискучим обслуговуванням клієнтів.

1.3.5 Модулі програмного забезпечення

Інформаційне програмне забезпечення готелів має модулі, які охоплюють декілька аспектів управління готелем. В основному це включає відстеження бронювання, бронювання готелів, обслуговування номерів,

ведення господарства, управління запасами та облік. Хороша технологія управління готелем дозволить заощадити багато часу та коштів за рахунок впорядкування цих процесів. Нижче пояснено деякі модулі програмного забезпечення:

1. Адміністрація.

Цей модуль програмного забезпечення для готельних рахунків дозволяє налаштувати та облаштувати готель. Ось що все знаходиться під навісом модуля адміністрування:

- Управління типами номерів та можливостями за допомогою програмного забезпечення для управління готельними номерами
- Виділення статусу дозволу
- Встановити дати
- Вибір оплати
- Встановлення плати за типом номера
- Додавання, редагування та відключення персоналу
- Керування товарними запасами
- Встановити ставки податку

2. Ведення господарства.

Модуль побутового обслуговування або реєстрація програмного забезпечення управління готелем - це місце, де відбувається максимально щоденна робота. Наступні завдання є частиною цього модуля:

- Управління бронюванням
- Облік фінансових операцій
- Бронювання
- Доступ до звіту гостей для перевірки бронювання
- Роздрукувати картки для реєстрації гостей
- Заїзд гостей, коли вони приїжджають
- Огляньте гостей
- Рахунки

3. Системи бронювання готелів.

Для будь-якого готелю найважливішою особливістю програмного забезпечення бронювання готелів є управління бронюванням. Хороша інформаційна система готелю показує таку інформацію:

- Які номери доступні для бронювання?
- Коли клієнти прибувають і виїжджають?
- Доступні ціни

Програмне забезпечення управління готелем забезпечує точність бронювання, щоб персонал адміністрації міг уникнути незручних проскакувань та перерахунків.

4. Вклади.

З застереженнями приходьте на необхідність депозитів. Якщо готель загрожує депозитом, щоб забезпечити бронювання, це має бути імітовано на рахунку гостей. Саме тут відображається програмне забезпечення для виставлення рахунків у готелі. Він проводить перевірку резервацій із депозитами, а також незатверджених. Це дозволяє адміністративному персоналу точно контролювати фінансові операції та операції.

5. POS.

Програмне забезпечення для готелів в Інтернеті також постачається з вбудованою системою POS (точка продажу). Ця система відстежує продажі, які відбуваються в готелі. Наприклад, клієнт купує речі в магазині подарунків, розташованому всередині готелю, це програмне забезпечення відстежує його. Тоді вони формуватимуть квитанції та допоможуть адміністративному персоналу в успішному виконанні рахунків.

6. Звітність.

Найкраще програмне забезпечення управління готелем також легко обробляє функцію Reporting. Це включає:

- Майбутні бронювання
- Приміщення в користуванні
- Виплачені депозити
- Рівень заповнення

- Продажі наявного місяця

Ці звіти є не просто корисними для звітів про прибутки та збитки, але також дозволяють персоналу з побутової роботи при визначенні номерів, що виписуються, та номерів з гостями. Це гарантує, що всі кімнати прибрано прибрано без зайвих клопотів.

1.3.6 Речі, які слід враховувати під час вибору програмного забезпечення для управління готелями

1. Простота використання.

Інтерфейс інформаційної системи готелю не повинен бути надто складним у використанні та інтеграції. Легкість користувальницького інтерфейсу визначатиме криву навчання ваших співробітників. Більш складний та неінтуїтивний інтерфейс означає, що буде витрачено більше часу на навчання персоналу та перехід.

2. Підтримка клієнтів.

З програмним забезпеченням для управління готелем все може статися будь-коли. Система може зависнути, або відбудеться поломка сервера, всі ці проблеми не повинні впливати на обслуговування готелю. Переконайтеся, що у вас є технічна підтримка 24 * 7. Перш ніж здійснити остаточний дзвінок, перевірте відгуки та спробуйте зв'язатися з власниками готелів, які вже використовували програмне забезпечення в минулому.

3. Настроювання.

Залежно від розміру та типу власності, вимоги системи онлайн-управління готелем можуть відрізнятися. Можливо, вам може знадобитися система управління на ліжку замість кімнати в модулі переднього офісу. Ось чому важливо переконатися, що постачальник програмного забезпечення для готельних програм надає нам такий тип функціональності.

4. Варіанти інтеграції.

Якщо ви вже використовуєте систему для свого готелю або плануєте інтегрувати будь-яке додаткове програмне забезпечення, переконайтеся, що

ваш постачальник надає всі необхідні API та готовий надавати послуги інтеграції. Скажімо, наприклад, якщо ви використовуєте програмне забезпечення CRM, ніж його інтегрувати з вашим програмним забезпеченням для управління готелем, слід легко.

1.4 Висновок до першого розділу

У першому розділі був здійснений огляд автоматизованих систем управління. Було описано впровадження автоматизованих систем в готельному бізнесі та інших сферах людської діяльності. Узагальнення всіх переваг та недоліків вже готових програмних продуктів, які є на ринку. Також було описано найбільш потрібні функції та модулі для розробки актуального програмного забезпечення. В кінці розділу було детально розглянуто речі, які слід враховувати під час вибору програмного забезпечення для управління готелями.

2 ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ

2.1 Огляд технологій

Для відображення матеріалів у роботі використовується мова гіпертекстової розмітки HTML. Вона складає основу технології гіпертексту. Гіпертекстовий документ містить так звані гіперпосилання.

HTML або гіпертекстова мова розмітки - це глобально прийнята мова програмування для форматування веб-сторінок. У сучасному світі зазвичай використовується разом із JavaScript та каскадними таблицями стилів (CSS) для надання веб-сторінкам вигляду та відчуття, якого ми бажаємо. За допомогою HTML можна формувати зовнішній вигляд зображень, посилань, заголовків, тексту, макета сторінки та майже кожного елемента веб-сторінки. Хоча сьогодні існують інші всесвітньо визнані мови веб-програмування та такі інструменти, як системи управління вмістом, HTML продовжує залишатися переважаючою мовою програмування для створення веб-сторінок. Він також є найоптимальнішим для більшості малих і зростаючих підприємств, які не дуже потребують розширеної функціональності на своєму веб-сайті. Нижче наведено деякі переваги використання HTML під час створення вашого веб-сайту:

1. HTML простий у використанні та розумінні.

Якщо в будь-який момент вам потрібно найняти послуги іншої фірми з веб-дизайну або професіонала для внесення змін або оновлень на ваш веб-сайт, було б відносно легко знайти економічно ефективних та доступних постачальників рішень, які зможуть внести потрібні зміни на ваш веб-сайт.

2. Усі браузерери підтримують HTML.

Практично - якщо не всі - браузерери підтримують HTML. Звичайно, більше браузерів підтримує HTML, ніж будь-яка інша мова веб-

програмування. Як результат, коли ви створюєте веб-сайт за допомогою HTML, він відобразатиметься в більшості браузерів по всьому світу, якщо програміст піклується про оптимізацію веб-сайту для найбільш часто використовуваних браузерів. Оптимізація веб-сайту на базі HTML для сумісності браузера не є ні складним, ні складним.

3. Синтаксис HTML і XML дуже схожий.

Сьогодні XML все частіше використовується для зберігання даних. Подібність синтаксису між HTML і XML означає, що це легше і безпроблемно працювати між двома платформами.

4. HTML безкоштовний.

Головною перевагою HTML є те, що він безкоштовний. Вам не потрібне програмне забезпечення для HTML, додатки не потрібні, і це означає, що ви можете значно заощадити на вартості розробки веб-сайту. Навіть із системами управління вмістом з відкритим кодом, всі плагіни, які вам можуть знадобитися, не завжди є безкоштовними.

5. Більшість інструментів розробки підтримують HTML.

Будь то FrontPage, DreamWeaver або будь-який інший інструмент програмування, є більше інструментів веб-розробки, які дозволяють створювати веб-сайти на базі HTML, ніж будь-яка інша мова веб-програмування.

6. HTML є найбільш зручним для пошукових систем.

З усіх мов веб-програмування HTML є найбільш зручним для пошукових систем. Створення веб-сайтів, сумісних з SEO, за допомогою HTML значно простіше, ніж будь-яка інша мова програмування. HTML викликає найменші ускладнення в SEO та забезпечує найбільшу гнучкість при спробі створення веб-сайту, сумісного з SEO. Поки ви подбали про те, щоб ваш HTML-код був чистим та підтвердженим, веб-сайт HTML простіше читати та отримувати доступ до сканерів пошукових систем. Це скорочує час сканування та покращує час завантаження сторінки, допомагаючи веб-сайту краще працювати[6].

При виборі мови програмування для написання програмного продукту було зупинено вибір на PHP. PHP означає Hypertext Preprocessor і є мовою програмування на стороні сервера.

Існує багато причин використовувати PHP для програмування на сервері, по-перше, це вільна мова без ліцензійних платежів, тому вартість її використання мінімальна.

Хорошою перевагою використання PHP є те, що він може взаємодіяти з багатьма різними мовами баз даних, включаючи MySQL. І PHP, і MySQL сумісні з сервером Apache, який також вільно ліцензувати. PHP також може працювати на серверах Windows, Linux та Unix.

Завдяки тому, що всі ці мови є безкоштовними, це легко та просто налаштувати та створити веб-сайт за допомогою PHP.

PHP також має дуже гарну онлайн-документацію з хорошими рамками функцій. Це робить мову відносно простою для вивчення та дуже добре підтримується в Інтернеті. Існує незліченна кількість форумів та навчальних посібників з різних методів та проблем PHP, тому, як правило, знайти допомогу дуже просто.

Є три основні області, де використовуються скрипти PHP.

- Сценарії на стороні сервера. Це найбільш традиційне і основне цільове поле для PHP. Для цього вам потрібно три речі: PHP-аналізатор (CGI або серверний модуль), веб-сервер та веб-браузер. Вам потрібно запустити веб-сервер із підключеною установкою PHP. Ви можете отримати доступ до виходу програми PHP за допомогою веб-браузера, переглядаючи сторінку PHP через сервер. Все це може працювати на вашій домашній машині, якщо ви просто експериментуєте з програмуванням PHP

- Сценарій командного рядка. Ви можете зробити сценарій PHP, щоб запустити його без будь-якого сервера чи браузера. Вам потрібен лише аналізатор PHP, щоб використовувати його таким чином. Цей тип використання ідеально підходить для сценаріїв, які регулярно виконуються за

допомогою cron (у * nix або Linux) або планувальника завдань (у Windows). Ці сценарії також можна використовувати для простих завдань з обробки тексту.

- Написання настільних програм. PHP - це, мабуть, не найкраща мова для створення настільного додатку з графічним інтерфейсом користувача, але якщо ви дуже добре знаєте PHP і хотіли б використовувати деякі розширені функції PHP у своїх додатках на клієнті, ви також можете використовувати PHP-GTK для писати такі програми. Ви також маєте можливість писати кросплатформенні програми таким чином. PHP-GTK - це розширення до PHP, недоступне в основному дистрибутиві.

PHP можна використовувати у всіх основних операційних системах, включаючи Linux, у багатьох варіантах Unix (включаючи HP-UX, Solaris і OpenBSD), Microsoft Windows, macOS, RISC OS та, ймовірно, інших. PHP також підтримує більшість веб-серверів сьогодні. Сюди входять Apache, IIS та багато інших. І це включає в себе будь-який веб-сервер, який може використовувати бінарні файли FastCGI PHP, такі як lighttpd і nginx. PHP працює як модуль, або як процесор CGI.

Тож із PHP ви маєте свободу вибору операційної системи та веб-сервера. Крім того, у вас також є можливість використовувати процедурне програмування або об'єктно-орієнтоване програмування (ООП) або їх суміш.

За допомогою PHP ви не обмежені у виведенні HTML. Можливості PHP включають в себе виведення зображень, PDF-файлів і навіть флеш-фільмів (з використанням libswf та Ming), що генеруються на льоту. Ви також можете легко виводити будь-який текст, наприклад XHTML та будь-який інший XML-файл. PHP може автоматично генерувати ці файли та зберігати їх у файловій системі замість того, щоб роздруковувати їх, утворюючи кеш-сервер на динамічному вмісті.

Однією з найсильніших і найбільш значущих особливостей PHP є підтримка широкого спектру баз даних. Написати веб-сторінку з підтримкою бази даних неймовірно просто, використовуючи одне з розширень для бази даних (наприклад, для mysql) або використовуючи рівень абстракції, як PDO,

або підключитися до будь-якої бази даних, що підтримує стандарт Open Data Connection Connection через розширення ODBC . Інші бази даних можуть використовувати CURL або сокети , як CouchDB.

PHP також підтримує розмову з іншими службами, використовуючи протоколи, такі як LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (у Windows) та незліченну кількість інших. Ви також можете відкривати необроблені мережеві розетки та взаємодіяти за допомогою будь-якого іншого протоколу. PHP підтримує комплексний обмін даними WDDX між практично всіма мовами веб-програмування. Якщо говорити про взаємозв'язок, PHP підтримує інстанціювання об'єктів Java та прозоре їх використання в якості PHP-об'єктів[7].

PHP має корисні функції обробки тексту , що включає в себе Perl сумісні регулярні вирази (PCRE), а також багато розширень та інструментів для розбору та доступу до XML-документів . PHP стандартизує всі розширення XML на твердій основі libxml2 та розширює набір функцій, додаючи підтримку SimpleXML , XMLReader та XMLWriter .

Для зберігання усієї нормативної документації у роботі необхідно використати реляційну СУБД.

SQL - це стандартизована мова, що використовується для доступу до бази даних.

У такому випадку за допомогою спеціальних функцій PHP, можна буде реалізувати обмін даними із базою даних. Після детального огляду було вибрано найпопулярнішу і поширену СУБД - MySQL.

MySQL - це система управління базами даних, яка дозволяє керувати реляційними базами даних. Це програмне забезпечення з відкритим кодом, що підтримується Oracle. Це означає, що ви можете використовувати MySQL, не платячи ні копійки. Також, якщо ви хочете, ви можете змінити його вихідний код відповідно до ваших потреб.

Незважаючи на те, що MySQL є програмним забезпеченням з відкритим кодом, ви можете придбати комерційну ліцензійну версію від Oracle, щоб отримати послуги преміум-підтримки.

MySQL досить легко освоїти порівняно з іншим програмним забезпеченням баз даних, таким як Oracle Database або Microsoft SQL Server.

MySQL може працювати на різних платформах UNIX, Linux, Windows тощо. Ви можете встановити його на сервері або навіть на робочому столі. Крім того, MySQL надійний, масштабований та швидкий[8].

2.2 Програмне забезпечення, що використовується для розробки

При розробці програмного забезпечення важливим є вибір інтегрованого середовища розробки. Вони зазвичай складаються з редактора сирцевого коду, компілятора чи/або інтерпретатора, засобів автоматизації збірки, та зневаджувача. IDE містить редактор, в якому ви можете редагувати код, налагоджувати код, переглядати код у браузері (часто вбудованому), а також перевіряти та виходити з контролю вихідного коду. Щоб підтримати цю функціональність, IDE має набір функцій, яких ви не можете знайти в базовому редакторі, наприклад, Блокнот або Vim. Знову ж таки, ви можете розширити редакторів, щоб зробити багато з цих речей, але IDE мають всю цю функціональність в одному акуратному пакеті - як правило, заздалегідь налаштовано:

- Проекти. Ключовою відмінною особливістю IDE є те, що він розглядає додаток PHP як проект, а не просто набір файлів. Концепція, що містить проект - підтримує додаткову інформацію, таку як конфігурація управління вихідним кодом, налаштування бази даних для налагодження та розташування певних ключових каталогів.

- Налагодження. Ще одна зручна функція - це інтегрована налагодження. За допомогою цієї функціональності ви можете встановити точки перерви в редакторі, на якому інтерпретатор PHP зупиняється під час

виконання сценарію. Звідти ви можете перевірити значення локальних змінних для діагностики проблем у кодї. Це корисна альтернатива розповсюдженню echoоператорів через ваш код для перевірки значень або використання журналу помилок для отримання значень змінних.

- Кодова розвідка. PHP - це дуже регулярна мова програмування, а це означає, що вона дотримується простих зразків. Ці шаблони не тільки дозволяють легко писати, вони полегшують IDE перевірити код у вашому проєкті. Крім того, вони допомагають вам під час написання, показуючи результати їх перевірки. Наприклад, якщо ви визначите клас з назвою MyClassy вашому проєкті, тоді IDE надає спливаюче вікно, яке містить MyClassopцію, як тільки ви введете ключове слово new. Під час використання об'єкта цього типу IDE показує його доступні методи та змінні екземпляра. Коли ви починаєте вводити виклик функції, IDE допомагає вам, показуючи доступні аргументи. Чесно кажучи, це причина №1, що вам слід використовувати IDE, а не текстовий редактор. Цей тип кодової розвідки може економити години помилкових імен класів, назв методів та неправильних аргументів.

- Перегляд класу. Побічним ефектом наявності механізму розвідки коду в IDE є те, що IDE може створювати класний вид проєкту. Замість того, щоб показувати файли, система може показувати вам різні класи, які ви визначили, незалежно від файлу, в якому вони перебувають. Коли ви клацаєте по класах, ваш редактор переноситься до цього файлу і вибір, розміщений у класі, методі, або змінної примірника. Це набагато приємніший спосіб навігації навколо великих проєктів.

- Підтримка декількох мов. Кожна тут IDE підтримує не лише PHP, а колекцію відповідних мов: JavaScript, структуровану мову запитів (SQL), мову розмітки гіпертексту (HTML) та каскадні таблиці стилів (CSS). Підтримка HTML та CSS часто найкраща, тому що це простіше. Підтримка JavaScript часто зводиться до виділення синтаксису, але щось краще, ніж нічого.

- Контроль вихідного коду. Усі оцінені тут IDE підтримують деяке підключення до системи управління вихідним кодом, що дозволяє підтримувати версії файлів у вашому проекті протягом часу. Ви можете позначити окремі версії файлів як випуск, а потім повернутись до них, коли потрібно виконати внесені вами зміни. У командних середовищах дуже важливо використовувати систему контролю вихідного коду, але для людей важливо також використовувати її. Хороша система управління вихідним кодом може врятувати вас, коли диск підірветься або коли клієнт раптом захоче версію до останньої, а не те, що у вас є сьогодні. Більшість IDE підтримують систему одночасних версій (CVS) та Subversion, які є системами управління відкритим кодом. Одна IDE підтримує Perforce, комерційну систему управління вихідним кодом.

- Інтеграція FTP / SFTP. Пов'язане з контролем вихідного коду - це можливість використовувати FTP для останнього коду на сервері. Це набагато простіше, ніж використовувати клієнт FTP або самостійно упакувати файли, відправляти їх на сервер та розпаковувати їх.

- Навігація по базі даних. Корисна, але не суттєва особливість - навігація по базі даних. За допомогою цієї функції ви можете переглядати базу даних, з якою розмовляє ваша програма, дізнаватися назви таблиць і полів та виконувати запити. Деякі системи навіть автоматизують написання деякого коду доступу до бази даних для вас.

- Вбудований веб-браузер. Деякі з IDE підтримують інтегрований веб-браузер, який може переходити безпосередньо до сторінки, яку ви редагуєте, за допомогою додаткових аргументів, які ви вказуєте, веб-переглядач розміщується в IDE або викликається зовні. Якщо чесно, я не є великим шанувальником інтегрованого браузера, тому що я не проти перемикатися між редагуванням коду та переглядом результату у двох окремих програмах. Але я бачу атракціон, і не потрібно, щоб ти ним користувався.

- Сніпсети. Останньою особливістю, яку я знайшов у всіх цих IDE, була підтримка як консервованих, так і спеціальних фрагментів

коду. Фрагменти - це невеликі фрагменти коду, які виконують невеликі завдання, такі як виконання регулярного виразу на деякому вході, підключення до бази даних та запит до бази даних[9].

Безпосередньо для написання коду проекту була обрана IDE Eclipse.

Eclipse - це платформа, яка була створена з нуля для створення інтегрованих інструментів для веб-розробок та додатків. За задумом платформа сама по собі не забезпечує багато функцій кінцевого користувача. Цінність платформи - це те, що її заохочує: швидкий розвиток інтегрованих функцій на основі моделі плагінів.

Eclipse забезпечує загальну модель користувальницького інтерфейсу (UI) для роботи з інструментами. Він розроблений для роботи в декількох операційних системах, забезпечуючи надійну інтеграцію з кожною базовою ОС. Плагіни можуть запрограмувати портативні API Eclipse та працювати без змін у будь-якій підтримуваній операційній системі.

В основі Eclipse лежить архітектура для динамічного виявлення, завантаження та запуску плагінів. Платформа обробляє логістику пошуку та запуску правильного коду. Користувальницький інтерфейс платформи забезпечує стандартну модель навігації користувача. Кожен плагін може потім зосередитись на тому, щоб добре виконати невелику кількість завдань[10].

2.3 Висновок до другого розділу

У другому розділі був здійснений огляд технологій для здійснення розробки програмного продукту. У першій частині розділу було здійснено вибір мови програмування для написання програмного продукту – PHP, для відображення матеріалів у роботі вибрано мову гіпертекстової розмітки HTML та систему управління базами даних, яка дозволяє керувати реляційними базами даних – MySQL. У другій частині розділу розглянуто програмне забезпечення, що використовується для розробки. Безпосередньо для написання коду проекту була обрана IDE Eclipse

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Загальні відомості

Найменування завдання: Створення системи керування готелем.
Умовне позначення системи: СКГ “easyHotel”. Також в документі використовується наступні найменування: система, система керування готелем (СКГ).

3.2 Призначення та цілі розробки

Призначення СКГ «easyHotel» є засобом керування процесами в готелі та заохоченням до електронного бронювання номерів, а також доступу до інформації пов’язаної з ними.

Цілі створення системи: СКГ «easyHotel» створюється з метою впровадження технологій електронного зберігання, керування та доступу до інформації готелю.

Створення СКГ «easyHotel» має метою забезпечення:

- швидкого доступу до даних про запити на бронювання;
- легкого керування інформацією про вільні та зайняті номери готелю;
- та автоматичне бронювання номерів системою після підтвердження даних клієнта адміністратором;

3.3 Вимоги до розробки

В СКГ «easyHotel» необхідно забезпечити розділення доступу до системи на адміністративну і користувацьку частини.

Користувацька частина СКГ «easyHotel» повинна виконувати наступні функції:

- перегляд сайту готелю;

- можливість заповнення форми бронювання номерів готелю;

Адміністративна частина системи повинна забезпечувати:

- перегляд списку записів на бронювання;
- бронювання номеру з підтвердженням адміністратора;
- автоматичний пошук кімнати;
- надсилання e-mail клієнту про успішне бронювання ;
- перегляд списку усіх номерів;

Система повинна мати інтуїтивно зрозумілий інтерфейс.

Користувацька частина має мати не «важкий» дизайн, щоб не завантажувати користувача зайвими елементами. Інтерфейсу адміністративної частини важливо відповідати сучасним тенденціям, із використанням стандартів і можливостей HTML5 і CSS3.

Програмні засоби СКГ «easyHotel» повинні забезпечувати справне цілодобове її функціонування в нормальних умовах. Доступ до даних повинен мати лише адміністратор.

3.4 Вибір методу багатокритерійного автоматизованого підбору номерів

Метод автоматизованого багатокритерійного підбору - це методологія, яка рекомендує найбільш підходящий багатокритерійний метод прийняття рішень (MCDM) з підмножини кандидатських методів, коли передбачається ризик та невизначеність. Структурований підхід створений на основі аналізу проблем та характеристик методів MCDM. Результати цього аналізу дають керівникам рішень запропоновану групу кандидатських методів для їх вирішення. Аналіз чутливості застосовується до запропонованої групи кандидатних методів для аналізу надійності результатів, коли передбачається ризик та невизначеність.

Метод MCDM вибирається автоматично, він дає найбільш надійний результат - це чітко визначити проблему або дилему, яка ставиться перед

людиною, яка приймає рішення. Це може містити: ситуація, яка вимагає одноразового рішення або рішення, яке вимагає багаторазового, постійного вирішення.

Кожен день ми справляємося з багатьма багатокритеріальними проблемами які вирішуються інтуїтивно. Тож коли приймається багатокритеріальне рішення, як гарантувати побудову моделі та використання комп'ютерного підходу?

Моделний підхід MCDA підходить тоді, коли інтуїтивного підходу немає. Наприклад, коли той, хто приймає рішення, вважає, що це рішення є занадто велике і складне, щоб керуватись інтуїтивно, тому що він передбачає ряд суперечливих цілей або залучає чимало зацікавлених сторін із різноманітними поглядами.

Часто виникає бажання формальної процедури, щоб прийняття рішення стало процесом відкритим і прозорим і вважалось справедливим. Якщо проблема - така, яку можна структурувати як залучення до колекції альтернатив, які можна перевірити за кількома критеріями, тоді MCDA підхід може бути підходящим.

Такі альтернативи можуть бути альтернативними варіантами, діями, стратегіями або альтернативними підрозділами.

Неадекватна або погано визначена проблема часто призводить до неякісного прийнятого та неінформовного рішення. З іншого боку, добре структурована і чітко визначена проблема може призвести до більш високого рівня розуміння та безліч нової інформації.

Прийняття рішення - це процес, в якому альтернативи оцінюються для вибору або способи дій для досягнення бажаних цілей і завдань. Відповідний процес прийняття рішень може мати істотне значення для успіху в організації. Проблеми та інформація, необхідна для прийняття рішення, можуть бути невиразними та невизначеними. Багато дослідників наголошували на необхідності врахувати невизначеність у прийнятті рішень однак це не часто розглядалося на практиці.

Для різних проблем у реальному житті використовуються різні методи прийняття рішень, і не існує однаково-стандартних методів. Було констатовано, що хоча різні методи зважування часто розглядають однакові теоретичні алгоритми, вони часто призводять до різних результатів.

Оскільки прийняття суджень в умовах високого ризику, нечіткості та невизначеності робить рішення більш вразливими до спотворення, залучення більш складних наукових методів прийняття рішень може допомогти. Більшість людей здатні вирішувати одночасно лише невелику кількість критеріїв. Для ефективнішого керування проблемами, що мають багато критеріїв, керівники підприємств, як правило, використовують методи MCDM. Але методи MCDM можуть мати як переваги, так і недоліки у пошуку відповідного кінцевого результату.

MCDM - це поле оперативних досліджень, де альтернативи оцінюються для вибору найбільш підходящої альтернативи, яка відповідає бажаній цілі стосовно набору численних і часто суперечливих критеріїв. MCDM є важливою частиною теорії прийняття рішень та оперативних досліджень. Його часто вважають надійним. MCDM - це сукупність методів та процедур, за допомогою яких в процес прийняття рішення можуть бути включені багато суперечливих критеріїв. Більше того, MCDM можна розглядати як систематичний процес аналізу та вибору між альтернативами. Він спрямований на те, щоб розділити проблему на менші частини, проаналізувавши та оцінивши кожную частину, а потім об'єднати всі частини для вибору найкращої можливої альтернативи з набору альтернатив, використовуючи заздалегідь заданий набір критеріїв. MCDM має на меті дати можливість особам, які приймають рішення, вирішувати суперечливі кількісні та / або якісні багатокритеріальні проблеми, а також знаходити найкращі альтернативи з набору альтернатив у певних, невизначених, нечітких або ризикованих середовищах.

Всі багатокритеріальні методи покращують процес прийняття рішень, розкладаючи загальну оцінку альтернатив оцінці ряду часто суперечливих критеріїв.

Послідовність порівнянь є важливою, оскільки вона показує надійність результатів, а також, що невідповідність може бути на порядок меншим значенням, ніж консистенція, або 10% від загальної стурбованості при послідовному вимірюванні. Якщо невідповідність було більше 10%, то це може порушити процес прийняття рішень.

Для осіб, які приймають рішення, важливо зрозуміти природу невизначеності, щоб підвищити їх здатність приймати рішення та знизити рівень ризику, пов'язаний з їх рішеннями. Більше того, розуміння керівниками рішень природи невизначеності може знизити показники невідповідності та забезпечити більш надійне та надійне представлення ваг та заходів щодо ефективності

Усі процеси прийняття рішень включають елемент ризику. Ризик - це непевна подія або умова, яка, якщо вона має місце, має позитивний чи негативний вплив хоча б на одну мету. Оскільки ризики походять від невизначеності, ризик з вірогідністю одного можна вважати фактом. Результати процесу прийняття рішень суттєво формуються внутрішніми та зовнішніми чинниками. Деякі з цих факторів відомі, а інші невідомі тим, хто приймає рішення.

Ризики неминучі в проблемах реального світу. Внутрішні ризики включають зміни часу, витрат та команди, зовнішні ризики включають зміни в регулюванні, зміну ринку, технічні проблеми та непередбачуваний ризик. Дослідження свідчать, що 90% проблем можна було б уникнути за допомогою кращого управління ризиками. Ті, хто приймає рішення, повинні використовувати, посилювати та охоплювати позитивний ризик (можливість) та уникати, пом'якшувати та переносити негативний ризик (загрози).

3.4.1 Розробка моделі для прийняття багатокритерійних рішень

Конкретні кроки є важливими для вирішення багатокритеріальної проблеми прийняття рішень. На рисунку 3.1 представлений розроблений підхід.

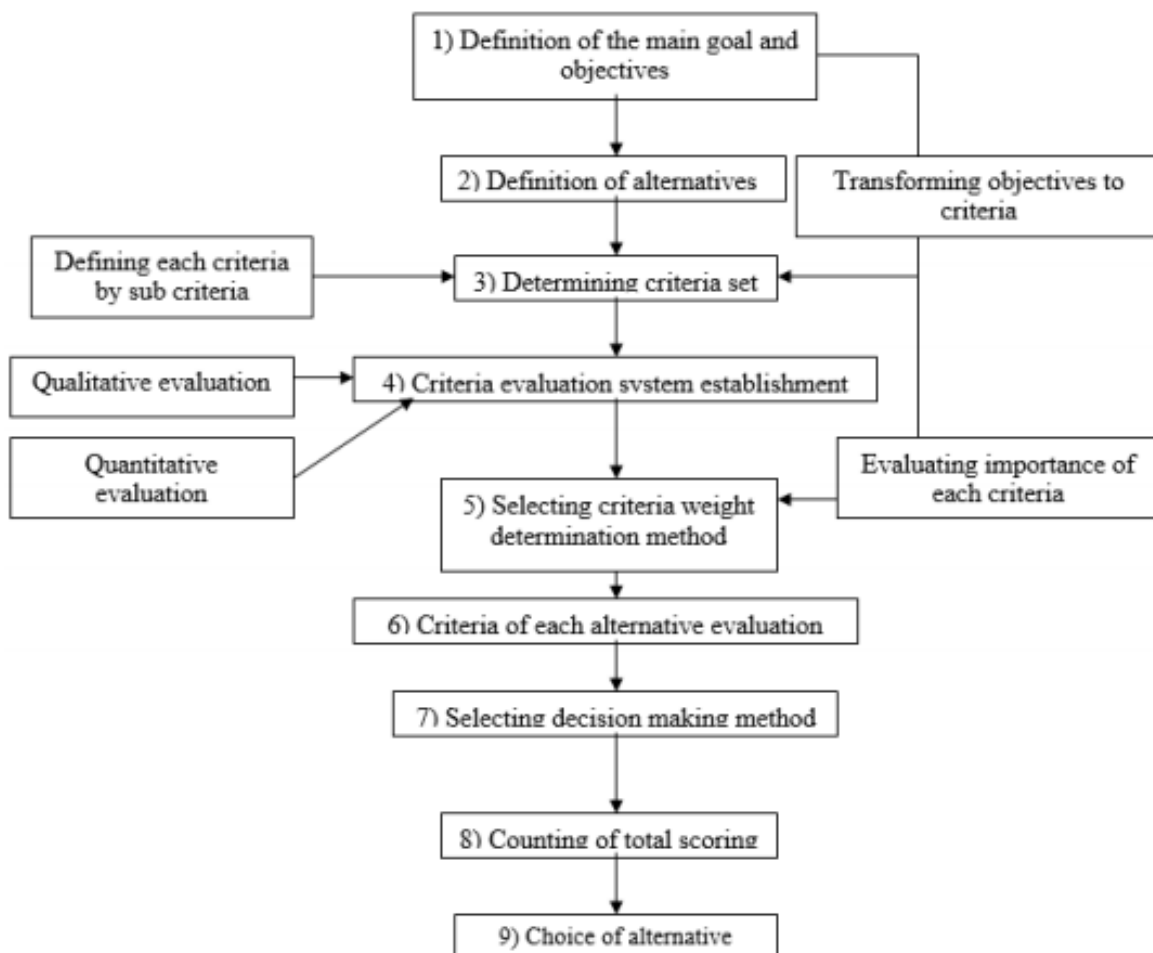


Рисунок 3.1 - Модель для прийняття багатокритерійних рішень

По-перше, особи, які приймають рішення, визначають усі можливі альтернативи вирішення проблеми.

По-друге, вони повинні встановити критерії альтернатив. Ті, хто приймає рішення, визначають фактори, які мають суттєво вплинути на вибір підрядника і є важливим для нього.

По-третє, учасник визначає цілі та набір критеріїв.

По-четверте, особи, які приймають рішення, визначають значення критеріїв, оскільки кожен критерій має своє значення для іншого конкретного вибору

По-п'яте, необхідно встановити ваги критеріїв, за якими можна визначити більш-менш істотні критерії. Більш критичним критеріям надається більша вага.

Потім особи, які приймають рішення, оцінюють альтернативи.

Нарешті, найкращий варіант вибирається за допомогою обчислень комп'ютерного програмного забезпечення

3.5 Застосування методу багатокритеріального підбору

Під час написання роботи виникло питання про найбільш оптимальний метод підбору номерів у готелі для АСУ. Свій вибір я зупинила на багатокритеріальному методі підбору номерів, адже саме цей метод, може приймати рішення по обранні, при необмеженій кількості критеріїв відбору.

Наведемо більш детальний опис, вибору номерів в розробці проекту-системи керування готелем.

Найкращим результатом багатокритеріального підбору номерів є те, що протягом часу подачі заявки від клієнта, вона швидко обробляється і на виході ми отримуємо на адміністративній частині додатку у вкладці запити на бронювання - результат найкращого підібраного, з допустимо можливих номерів готелю. Можна зменшувати чи збільшувати кількість критеріїв підбору, опираючись на вимоги та побажання конкретного замовника.

Розглянемо розв'язок задачі багатокритерійної оптимізації у роботі «The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs»[29] та «Optimization of software architecture selection for the system under design and reengineering»[30], де описано метод оцінки архітектур «For reasonable selection of decision in the method SAAM/ATAM selected architectures are analyzed on the subject of costs and benefits with

applying of method CBAM. This method ensures economic analysis of software system, which are based on the selected on previous methods variants of architecture and scenarios of simulation.»[29]. Проте у даній роботі задача була поставлена простіша, тому розглянемо наступний алгоритм виконання.

Розглянемо стандартну форму вимог до підбору номерів. Для прикладу оберемо самі критерії. Це буде: кількість гостей, сніданок включений у вартість, наявність міні сейфу в номері, наявність безкоштовного WI-fi та вартість.

Припустимо у нас є база даних з 10 номерів, наведено в таблиці 3.1.

Таблиця 3.1 – Критерії оцінки номерів

№	Кількість гостей	Сніданок, включений у вартість	Наявність міні-сейфу в номері	Наявність безкоштовного wi-fi	Вартість
1	3	Так	Так	Так	3000
2	2	Так	Ні	Так	1500
3	4	Ні	Ні	Так	3500
4	1	Ні	Так	Ні	1500
5	1	Так	Так	Так	2000
6	3	Так	Ні	Так	2500
7	4	Так	Так	Так	4300
8	2	Так	Ні	Так	1900
9	2	Так	Ні	Так	2300
10	3	Так	Ні	Так	3000

Оберемо головним критерієм підбору саме вартість, яку ми хочемо мінімізувати, щоб привабити клієнтів.

На інші критерії поставимо також довільні показники які міг ввести гість при поданні заявки. Для прикладу, наведено в таблиці 3.1, кількість гостей – 3, сніданок – так, наявність міні сейфу – ні, наявність безкоштовного WI-fi – так.

Почнемо підбір конкретного номеру з бази номерів. В таблиці 3.1 ми виділили клітинки які впливають на вибір. Таким чином на першому етапі ми з загальної бази номерів обираємо лише ті, що підходять нашому критерію «Кількість гостей» - 3. Таких критерії є лише 3, під номерами 1,6 та10. Вибір

наступного критерія буде проводитись вже з 3 обраних попереднім критерієм номерів. В наступному критерії змін не відбулось, оскільки 3 номери відповідають заданій вимозі. На 3 етапі, а саме «Наявність міні- сейфу в номері» залишається лише 2 номери 6 та 10. В наступній ітерації нічого не змінюється. На останньому ключовому етапі потрібно обрати вартість, але як вже було зазначено вище, нам потрібно її мінімізувати. Тому з двох варіантів обираємо 6 оскільки даний номер, який відповідає попереднім вимогам, є дешевшим.

3.6 Проектування СКГ

Структура даних проекту є схожою структурі даних популярних систем керування вмістом. Вона складається із двох основних компонент: бази даних системи, що містить усю необхідну для відображення та взаємодії з користувачами інформацію, а також системи класів, що забезпечують доступ, обробку і виведення інформації, що міститься в базі.

Згідно з технічним завданням, система повинна поділитись на дві частини користувацьку(клієнтську) і адміністративну. На рисунку 3.2 представлені актори системи.



Рисунок 3.2 - Актори системи.

Короткий опис акторів представлений в таблиці. 3.2.

Табл. 3.2. Виявлення акторів

<i>Актор</i>	<i>Короткий опис</i>
Клієнт	Особа яка має обмежений доступ до даних та можливість редагувати лише свої особисті дані. Може створювати запити на бронювання та тікети.
Адміністратор	Особа яка має доступ до всіх даних системи: запитів на бронювання, списку кімнат, бази бронювання, тікетів підтримки клієнтів. Та можливість редагувати наявні або додавати нові дані.

Виявлення варіантів використання зведене в таблицю 3.3.

Табл. 3.3. Виявлення варіантів використання

<i>Основний актор</i>	<i>Найменування</i>	<i>Формулювання</i>
Клієнт	Реєстрація в системі	Клієнт реєструється у системі залишивши запит на бронювання на головній сторінці, нового користувача системи буде створено якщо e-mail не зареєстровано в базі користувачів системи.
Клієнт, адміністратор	Авторизація в системі	Клієнт та адміністратор повинні авторизуватись в системі для отримання доступу до особистих кабінетів.
Клієнт	Залишити запит на бронювання	Клієнт має змогу залишити запит на бронювання з особистого.
Клієнт	Перегляд списку своїх запитів	Клієнт має змогу переглянути список своїх запитів та їх статус.
Клієнт	Зміна особистих даних	Клієнт має змогу змінити особисті дані: ім'я, номер телефону, емейл.
Клієнт	Зміна паролю	Клієнт має змогу змінити пароль до особистого кабінету.
Клієнт	Створити тікет	Клієнт може написати адміністратору за допомогою системи тікетів.

Клієнт, адміністратор	Перегляд тікетів	Клієнт та адміністратор можуть переглядати тікети з повідомленнями.
--------------------------	------------------	---

Продовження таблиці 3.3

Клієнт, адміністратор	Написання повідомлень у тікети	Клієнт та адміністратор мають змогу доповнювати тікети повідомленнями.
Адміністратор	Перегляд запитів на бронювання	Адміністратор переглядати список запитів на бронювання.
Адміністратор	Прийняти запит	Адміністратор може приймати запити на бронювання після чого створюється запис про бронювання для певної кімнати.
Адміністратор	Редагувати запит	Адміністратор може редагувати дані запиту на бронювання.
Адміністратор	Перегляд списку кімнат	Адміністратор може переглядати список кімнат та їхні статуси(вільна, занята) на момент перегляду.
Адміністратор	Додавання кімнати	Адміністратор може додати нову кімнату у базу даних.
Адміністратор	Додавання бронювань	Адміністратор може вручну додавати записи про бронювання вказаної кімнати.
Адміністратор	Перегляд статистики кімнати	Адміністратор може переглядати статистику кімнати: скільки днів кімната була зайнята протягом кожного місяця наявного роу, історію бронювань в радіусі місяця.
Адміністратор	Видалення бронювань	Адміністратор може видаляти запис про бронювання певної кімнати.

Всі варіанти використання приведені на рисунку 3.3.

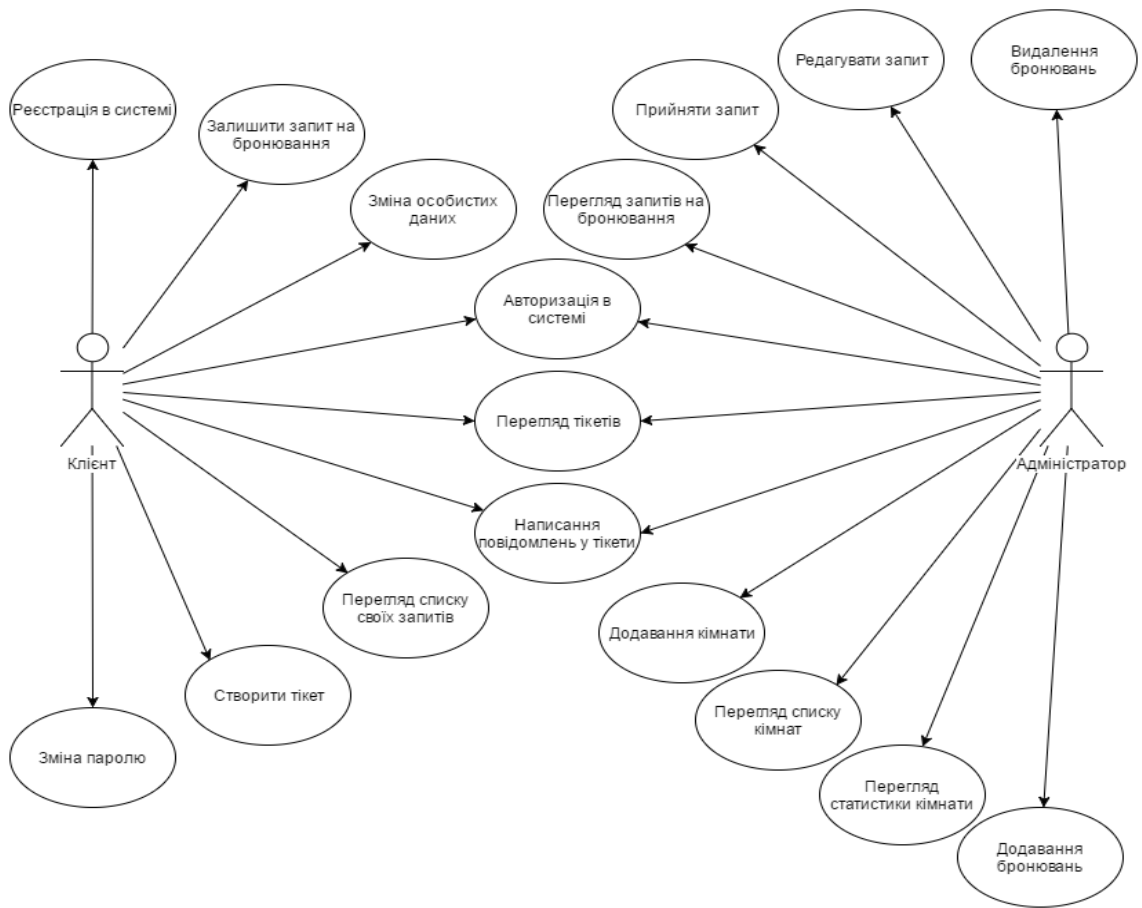


Рисунок 3.3 - Діаграма прецедентів системи.

На наступній діаграмі (рис. 3.4) ми можемо бачити 4 блоки класів, які в свою чергу мають перелік методів, що відповідають за певні процеси виконання. На рисунку ми бачимо методи типу `public` та `private`, вони блокують або дають доступ до використання атрибутів в інших класах.

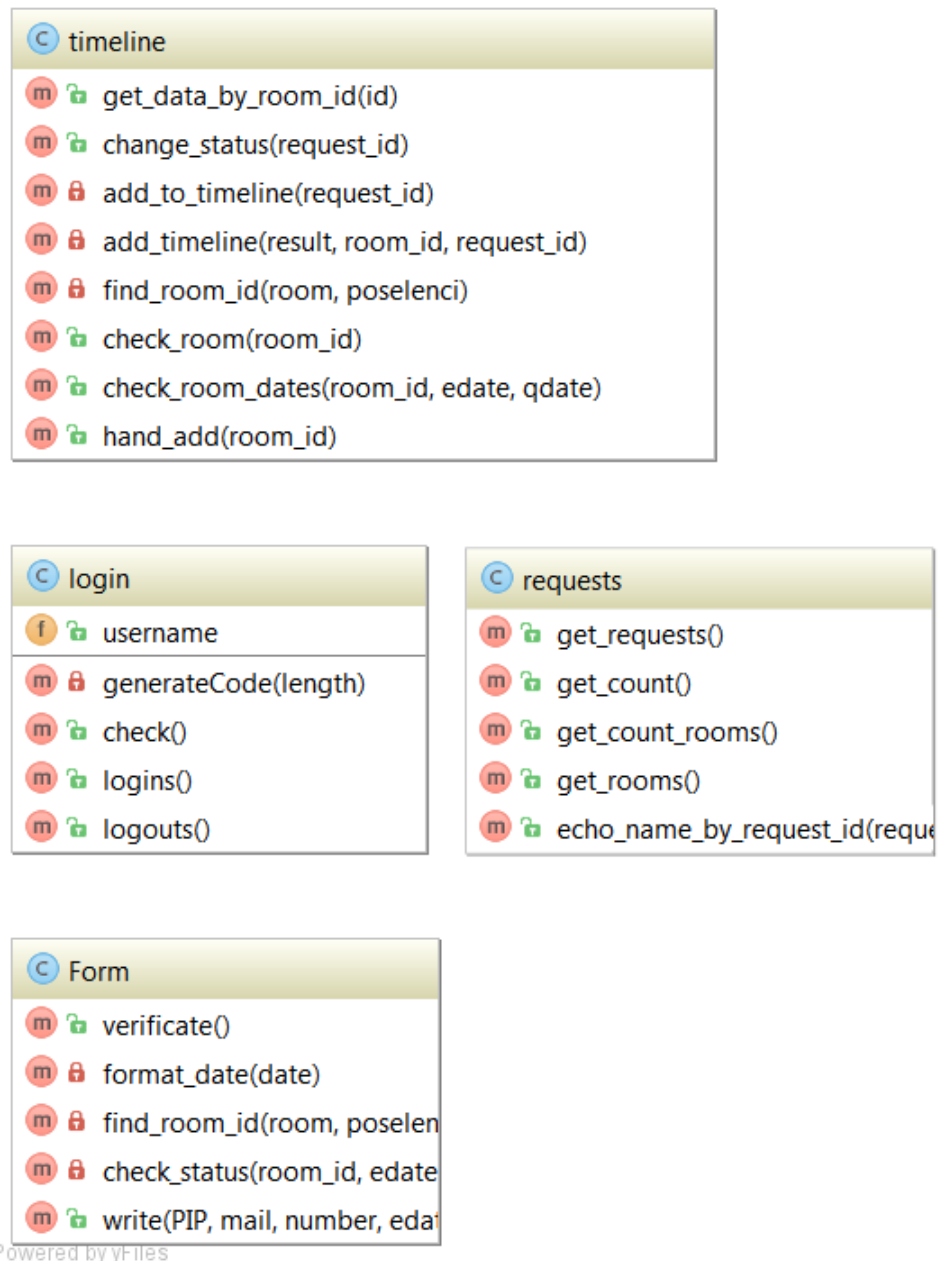


Рисунок 3.4 – Діаграма класів

Більш детально опишемо кожний клас. В класі timeline ми можемо бачити 8 методів, 3 з них є типу private, а інші 5 типу public. Клас request описує 5 public методів. В класі form описано 5 методів, 2 із них є типу public. Клас login має під собою 4 методи типу private, та 1 метод типу private.

3.7 Проектування бази даних

В ролі системи керування базами даних було обрано реляційну СКБД MySQL. Створення бази даних слід починати з її проектування (розробки). У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці.

Процес проектування можна здійснювати двома підходами. За першого підходу спочатку визначають основні задачі, для розв'язання яких створюється база, та потреби цих задач у даних. За другого підходу визначають предметну область (сферу), здійснюють аналіз її даних і встановлюють типові об'єкти предметної області. Найбільш раціональним підходом проектування бази даних є поєднання обох підходів.

3.7.1 Виявлення основних сутностей предметної області

Провівши аналіз предметної області, ми можемо виявити основні сутності предметної області. Оскільки мета роботи - це створення системи бронювання кімнат готелю, то перша сутність, яку можна виявити - це запит на бронювання (request) (рис. 3.5).

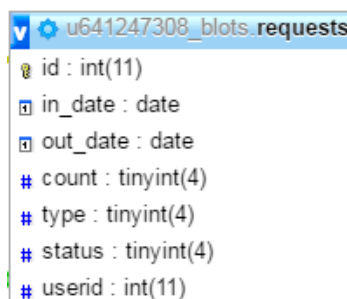


Рисунок 3.5 – Таблиця сутності request.

Таблиця складається лише з найнеобхіднішої інформації про запит: in_date - дати поселення, out_date - дати виселення, count - кількості місць у

кімнати, type - тип кімнати, status - статус запиту, та userid - ідентифікатора користувача який надіслав запит.

З цього можна дійти до висновку що потрібно створити сутність у якій буде зберігатись детальна інформація про кімнати – rooms (рис 3.6).

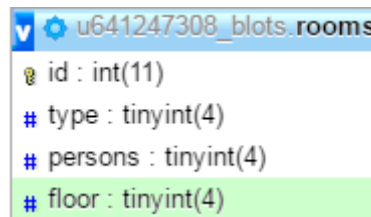


Рисунок 3.6 – Таблиця сутності rooms.

Таблиця складається з : type – типу кімнати, persons – місткісткості кімнати, floor – поверх на якій знаходиться кімната.

Для дотримання нормальних форм [2], тип кімнати був винесений у окрему сутність – room_type (рис. 3.7).



Рисунок 3.7 – Таблиця сутності room_type

Таблиця складається єдиного поля – name - назви типу кімнати.

Для відокремлення записів про бронювання та запитів про бронювання, а також для можливості ручного додавання інформації про зайнятість кімнат сутність була створенна – timeline (рис. 3.8).

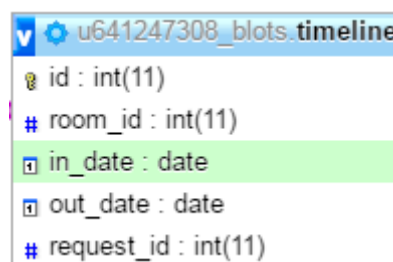


Рисунок 3.8 – Таблиця сутності timeline

Таблиця складається з: `room_id` – ідентифікатора кімнати, `in_date` – дати заселення, `out_date` – дати виселення, `request_id` – ідентифікатора запиту на бронювання.

Для зберігання даних про клієнтів та організації персональних кабінетів була створена сутність – `users` (рис. 3.9).

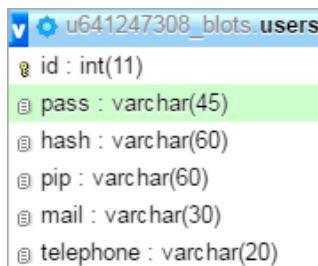


Рисунок 3.9 – Таблиця сутності `users`

Таблиця складається з персональних даних клієнта: `pip` – прізвище ім'я по-батькові, `telephone` – номер телефону. Та даних необхідних для авторизації у системі: `mail` – електронна пошта клієнта, `pass` – пароль, `hash` – поле необхідне для перевірки авторизації.

Для організації тикет-системи підтримки користувачів було створенно сутності – `tickets` (рис. 3.10) та `ticket_messages` (рис. 3.11).



Рисунок 3.10 – Таблиця сутності `tickets`

Таблиця `tickets` складається з: `status` – відображає статус тикета, `theme` – тема тикета, `userid` – ідентифікатор користувача який створив тикет.

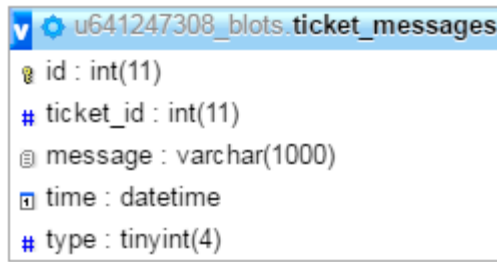


Рисунок 3.11 – Таблиця сутності ticket_messages

Таблиця ticket_messages створена для зберігання повідомлень і складається з полів: ticket_id – ідентифікатор тикета, message – повідомлення, time – дата та час відправлення повідомлення, type – який визначає ким було надіслано повідомлення(адміністратором чи клієнтом).

Для організації авторизації в адмін панель слугує сутність – admin (рис. 3.12).

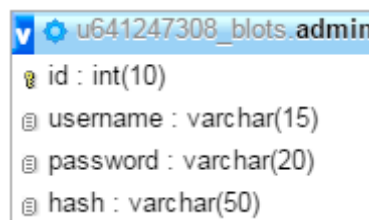


Рисунок 3.12 – Таблиця сутності admin

Таблиця складається з полів: username – логін адміністратора, password – пароль адміністратора, hash - поле необхідне для перевірки авторизації.

3.7.2 Побудова схеми реляційної бази даних

Під час аналізу предметної області були виявлені наступні сутності із такими відношеннями:

Сутність request має відношення з user.

Оскільки одному клієнту може належати кілька запитів на бронювання, але водночас один запит може мати лише одного власника(клієнта) то відношення між сутностями було визначено як «один до багатьох» (рис. 3.13).

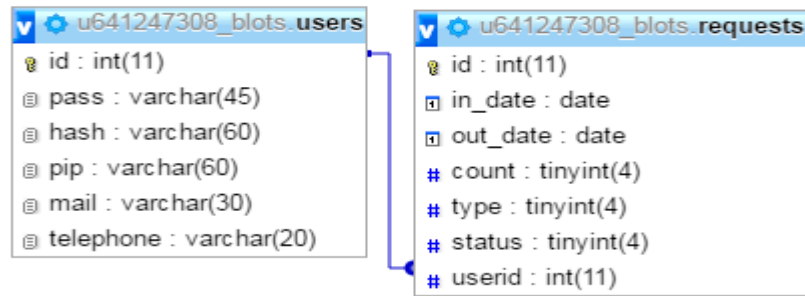


Рисунок 3.13 – Відношення між сутностями request та user.

Для реалізації даного відношення у таблиці request поле userid було позначено як зовнішній ключ для таблиці users.

Сутність timeline має відношення з сутністю requests.

Оскільки один запис бронювання може мати тільки один запит про бронювання, а один запит може мати тільки один запис то відношення між ними було встановлено як «один до одного» (рис 3.14).

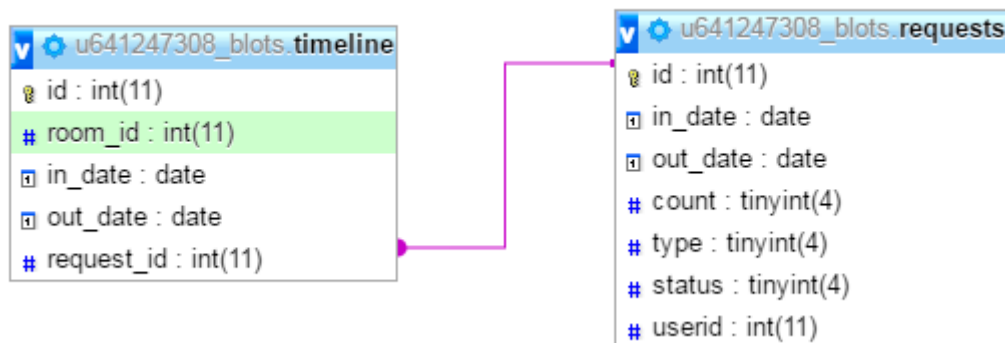


Рисунок 3.14 – Схема відношень таблиць timeline та request.

Для реалізації даного відношення у таблиці timeline поле request_id було позначено як зовнішній ключ для таблиці requests.

Сутність rooms має відношення з timeline.

Оскільки запис бронювання може відноситись до однієї кімнати, а кімната може мати багато записів бронювання, то відношення визначено як «один до багатьох» (рис. 3.15).

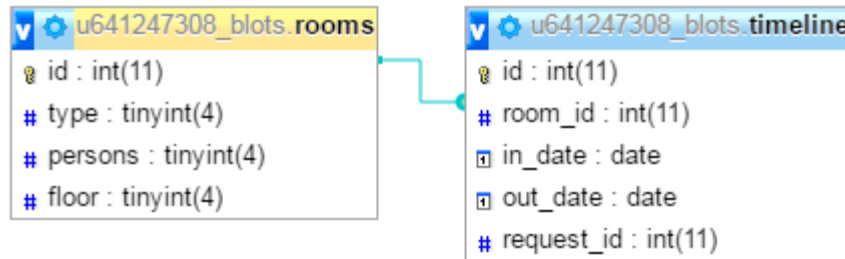


Рисунок 3.15 –Відношення між сутностями rooms та timeline.

Для реалізації такого відношення у таблиці timeline буде зберігатись зовнішній ключ(room_id) таблиці rooms.

Сутності rooms та requests мають відношення з room_types.

Оскільки тип кімнати може мати декілька кімнат та запитів на бронювання, а кімната та запит на бронювання можуть мати тільки один тип кімнати, то ці два відношення визначені як «один до багатьох» (рис. 3.16).

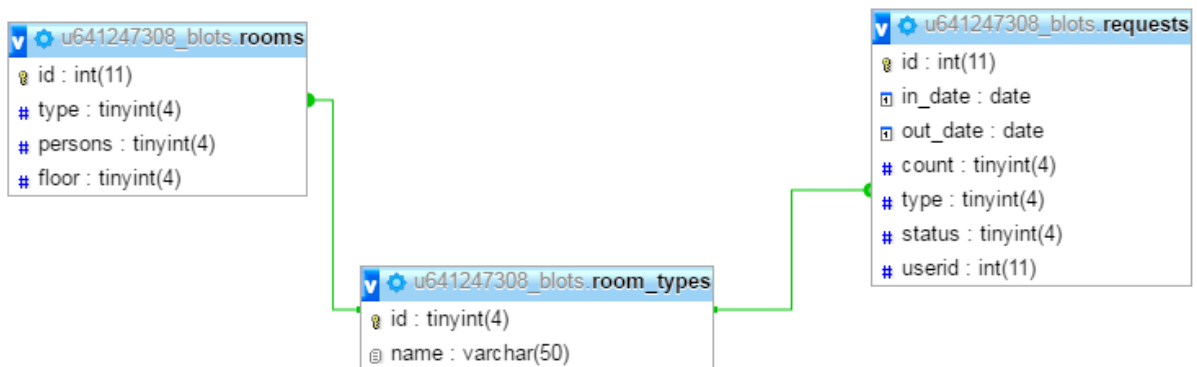


Рисунок 3.16 – Відношення між сутностями rooms та room_types і requests та room_types.

Для реалізації такого відношення у таблицях timeline та rooms буде зберігатись зовнішній ключ(type) таблиці room_types.

Сутність tickets має відношення з users.

Оскільки тiкет може мати лише одного власника, а клієнт може мати багато тiкетiв, то відношення визначено як «один до багатьох» (рис. 3.17).

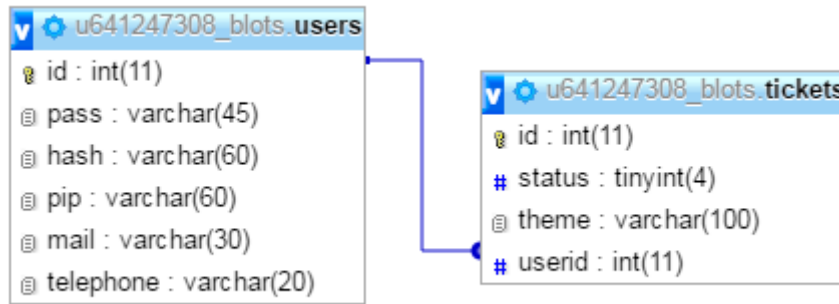


Рисунок 3.17 – Відношення між сутностями tickets та users.

Для реалізації такого відношення у таблиці tickets буде зберігатись зовнішній ключ(userid) таблиці users.

Сутність tickets має відношення з ticket_messages.

Оскільки повідомлення може належати лише одному тикету, а тикет може мати багато повідомлень, то відношення визначено як «один до багатьох» (рис. 3.18).

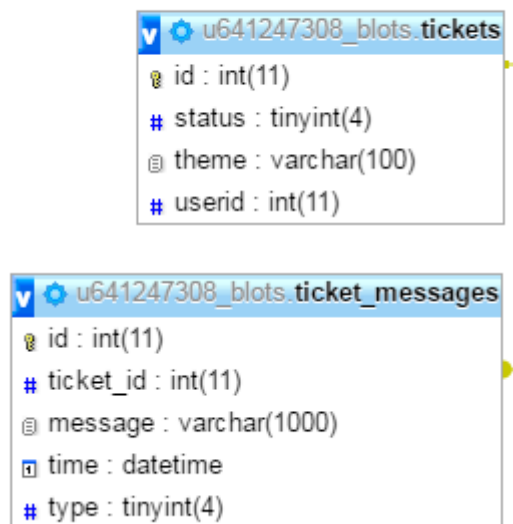


Рисунок 3.18 – Відношення між сутностями tickets та tickets_messages.

Для реалізації такого відношення у таблиці tickets буде зберігатись зовнішній ключ(userid) таблиці users.

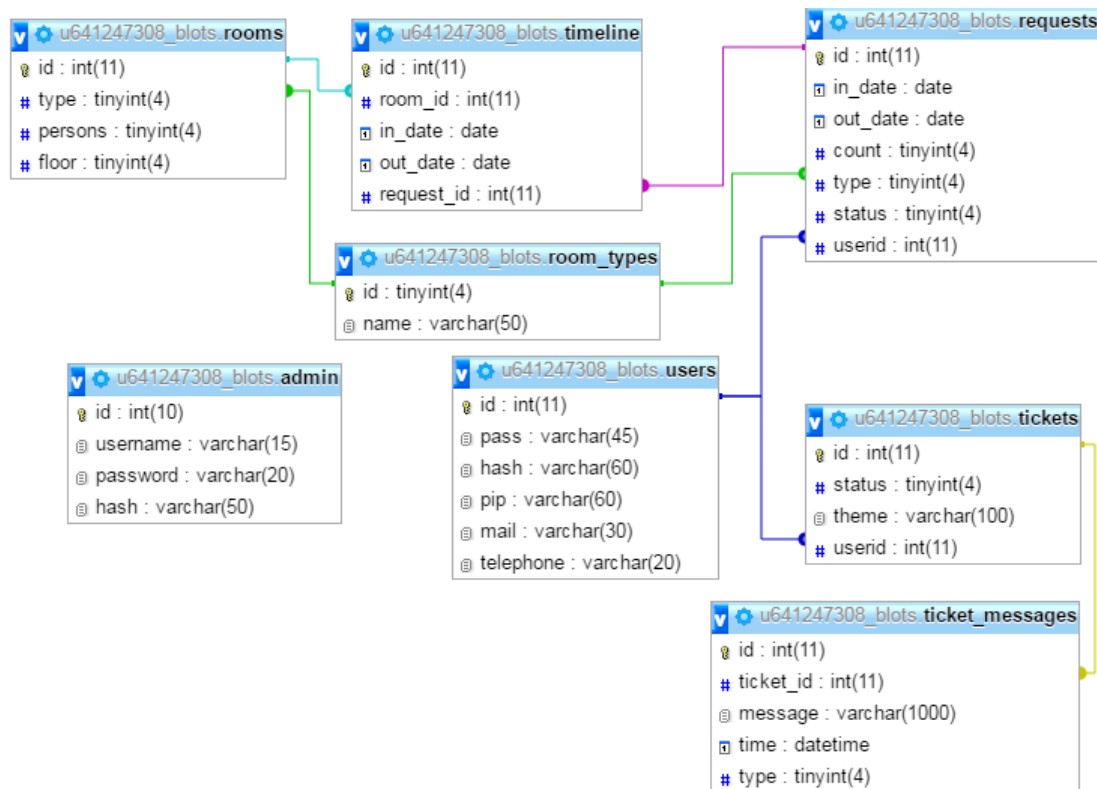


Рисунок 3.19 – Повна схема сутностей та їх зв'язків.

Після виявлення усіх сутностей і зв'язків між ними, можна побудувати загальну схему бази даних (рис. 3.19).

3.8 Опис концептуальної моделі бази даних

Для пришвидшення виконання певних елементів програми, було прийнято рішення помістити частину логіки на сторону сервера. На сервері було розміщено дві вбудовані процедури та один тригер.

Вбудовані процедури мають мету пришвидшити та полегшити виконання певних алгоритмів пов'язаних з маніпуляціями над даними з бази даних, тому вони виконують операції, які користувач програми, в теорії, буде використовувати досить часто.

Тригери мають мету забезпечити цілісність даних, які записуються у базу даних та, при необхідності, певним чином редагують записи при додаванні їх у базу даних.

Вбудована процедура `getDataByRoomID(id)` – забезпечує отримання інформації про бронювання кімнати.

Вбудована процедура `getRequest` – забезпечує отримання списку усіх запитів на бронювання.

Тригер `user_add_trigger` – тригер у таблиці `users`, який викликається перед спробою додавання даних в таблицю. Контролює коректність імені, що додається, при спробі додавання порожнього імені, тригер змінює його назву на «аноним».

3.9 Опис програмної реалізації

В процесі розробки було вирішено що робота з базою даних не буде вноситись в окремий клас, оскільки більшість функцій системи потребують зв'язку з базою даних, це б призвело до нагромодження великої кількості методів у даному класі і суттєво зменшило кількість методів в інших класах. Нижче наведено основні лістинги коду.

У лістингу 3.1 показано метод класу `Form` який відповідає за реєстрацію нового користувача.

Лістинг 3.1 – Реалізація методу `create_user`

```
private function create_user($PIP, $mail, $number){
    $conn = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
    $db    = mysql_select_db(DB_NAME);
    if (!mysql_set_charset('utf8', $conn)) {
        echo "Error: Unable to set the character set.\n";
        exit;
    }
    $pass=intval( rand(1,9) . rand(0,9) . rand(0,9) . rand(0,9)
    . rand(0,9) . rand(1,9) . rand(0,9) . rand(0,9));
    $pass1=md5($pass);
    $an= "INSERT INTO users (pass,pip,mail,telephone)
    VALUES ('".$pass1."','".$PIP."','".$mail."','".$number."')";
    mysql_query($an);
```

```
$message = 'Dear, '.$PIP.', your request have been sent to
administrator. And you were registered in our Booking system.'
```

```

Please use this data to login'."\n".'Login:
'.'.$mail."\n".'Password: '.$pass;
$headers = 'From: service@paix.hotel' . "\r\n" .
'Reply-To: service@paix.hotel' . "\r\n" .
'X-Mailer: PHP/' . phpversion();
mail($mail, 'Request obtained', $message, $headers);

$tn= "SELECT `id` FROM `users` WHERE mail = ' ".$mail."'";
$cn = mysql_query($tn);
$result = mysql_fetch_array($cn);

mysql_close($conn);
return $result['id'];
}

```

У лістингу 3.2 показано метод який відповідає за додавання запитів на бронювання у таблицю requests. Метод find_room_id – шукає кімнату яка підходить за заданими характеристиками(тип, місткість). Метод check_status перевіряє чи є бронювання можливим у даний період часу. Метод get_user шукає користувача у базі за його email адресою, якщо користувача з такою адресою не знайдено то викликається метод create_user, який було описано вище. Якщо ж такий користувач існує то повертається його id.

Лістинг 3.2 – Запис запиту на бронювання у базу даних

```

public function write($PIP, $mail, $number, $edate, $qdate,
$room, $poselenci){

    $room_id=$this->find_room_id($room, $poselenci);
    $status=$this->check_status($room_id, $edate, $qdate);
    $usid = $this->get_user($PIP, $mail, $number);

    $conn = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);

    $db = mysql_select_db(DB_NAME);

    if (!mysql_set_charset('utf8', $conn)) {
        echo "Error: Unable to set the character set.\n";
        exit;
    }

    $an= "INSERT INTO requests
(in_date,out_date,count,type,status,userid)
VALUES
(' ".$edate."', ' ".$qdate."', ' ".$poselenci."', ' ".$room."', ' ".$stat
us."', ' ".$usid."' )";

```



```

mysql_query($an);

mysql_close($conn);
}

```

Метод `get_rooms()` використовується для отримання списку усіх кімнат з їхніми типами які записані у таблиці `room_types` (лістинг 3.3).

Лістинг 3.3 – Отримання списку усіх кімнат.

```

public function get_rooms(){
    $conn = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
    $db    = mysql_select_db(DB_NAME);
    if (!mysql_set_charset('utf8', $conn)) {
        echo "Error: Unable to set the character set.\n";
        exit;
    }
    $an= "SELECT rooms.*, room_types.name FROM rooms
INNER JOIN room_types ON rooms.type=room_types.id";
    $qn = mysql_query($an);
    $i=0;
    while($row = mysql_fetch_array($qn)){
        $nrow[$i]['id'] = $row['id'];
        $nrow[$i]['name'] = $row['name'];
        $nrow[$i]['persons'] = $row['persons'];
        $nrow[$i]['floor'] = $row['floor'];
        $i++;
    }
    mysql_close($conn);
    return $nrow;
}

```

У лістингу 3.4 показано процес додавання нового запису бронювання у таблицю `timeline`, метод `check_another_requests_status` перевіряє решту запитів статус яких міг змінитись на «неможливо прийняти».

Лістинг 3.4 – Додавання бронювання в базу даних

```

private function add_timeline($result,$room_id,
$request_id){
    $conn = mysql_connect(DB_HOST, DB_USER,
DB_PASSWORD);
    $db    = mysql_select_db(DB_NAME);
    if (!mysql_set_charset('utf8', $conn)) {
        echo "Error: Unable to set the character
set.\n";
        exit;
    }
}

```

```

    }
    $qr= "INSERT INTO
timeline(room_id,in_date,out_date,request_id)
VALUES ('".$room_id."','".$result['in_date']."',
'".$result['out_date']."','".$request_id."')";

mysql_query($qr);

mysql_close($conn);

$message = 'Шановний(a),'.$result['pip'].' , повідомляємо вас що
заявку на бронювання готелю успішно підтверджено адміністратором.
Очікуємо на вас '.$result['in_date'].' після 12:00. ID заявки
'.$result['id'].'.';

mail($result['mail'], 'Бронювання підтверджено
адміністратором', $message);

$this->
check_another_requests_status($result['count'],$result['type']);
}

```

У лістингу 3.5 зображено як для зручності використання бази даних усі дані для підключення(сервер, користувач, пароль, назва бази даних), було винесено в окремий файл.

Лістинг 3.5 – Файл config.php

```

<?
define("DB_HOST", "mysql.hostinger.com.ua");
define("DB_NAME", "u641247308_blots");
define("DB_USER", "u641247308_blots");
define("DB_PASSWORD", "312543343");
?>

```

Лістинг 4.6 – Реалізація вбудованої процедури getDataByRoomID

```

CREATE PROCEDURE `getDataByRoomID` ( IN `id` INT( 11 ) )
SELECT timeline.id, timeline.room_id, timeline.in_date,
timeline.out_date, users.pip
FROM `timeline`
INNER JOIN `requests` ON timeline.request_id = requests.id
INNER JOIN `users` ON requests.userid = users.id
WHERE timeline.room_id = id
GROUP BY timeline.in_date DESC

```

У лістингу 3.6 продемонстрована робота вбудованої процедури `getDataByRoomID(id)` та алгоритму який вона виконує.

3.10 Опис класів

Клас **login** відповідає за авторизацію у систему. Атрибути та методи класу `login`: Атрибут `username` відповідає за відображення логіну на кожній сторінці системи. Метод `generateCode(length)` відповідає за генерацію унікального коду для авторизації. Метод `check()` викликається на кожній сторінці системи та перевіряє чи пройдено авторизацію, порівнює дані з кук з полем `hash` у таблиці `admin`. Метод `logins()` безпосередньо відповідає за авторизацію в системі, перевіряє дані з форми входу та викликає метод `generateCode(length)` для генерації коду який потім записується у куки та у базу. Метод `logouts` відповідає за вихід з системи, очищує поле `hash` у таблиці `admin`.

Клас **Form** відповідає за перевірку форми та створення запису у таблиці `requests`. Метод `verificate()` виконує перевірку даних введених в форму метод `format_date(date)` відповідає за приведення дати з формату форми до стандартів PHP5. Метод `find_room_id(room, poselenci)` відповідає за пошук кімнати з заданими критеріями а саме Тип кімнати і Кількість людей. Метод `check_status(room_id, edate, qdate)` перевіряє чи кімната вільна на даному проміжку часу. Метод `write()` відповідає за запис даних у таблицю `requests`.

Клас **requests** відповідає за вивід у адміністративну частину запитів користувачів та списоку кімнат. Метод `get_requests()` виконує вибірку записів з таблиці `requests`. Метод `get_count()` повертає кількість запитів з таблиці `requests`. Метод `get_rooms()` виконує вибірку кімнат з бази. Метод `get_count_rooms()` отримує кількість кімнат у таблиці `rooms`. Метод `echo_name_by_request_id(request_id)` повертає ім'я людини яка подала запит.

Клас **timeline** відповідає за процеси перевірки кімнат та запис бронювань. Метод `get_data_by_room_id(id)` виконує вибірку даних з таблиці

timeline про певну кімнату слугує для формування діаграм Ганта. Метод `change_status(request_id)` змінює статус запиту в таблиці `requests`. Метод `add_to_timeline(request_id)` виконує вибірку даних з таблиці `requests` викликає методи `find_room_id(room,poselenci)` та `add_timeline(result,room_id,request_id)`. Метод `find_room_id(room,poselenci)` відповідає за пошук кімнати з заданими критеріями а саме Тип кімнати і Кількість людей. Метод `add_timeline(result,room_id,request_id)` записує запис з бронюванням кімнати. Метод `check_room(room_id)` метод перевіряє чи вільна кімната на даний момент. Метод `check_room_dates(room_id, edate, qdate)` перевіряє чи кімната вільна на заданому проміжку. Метод `hand_add()` використовується для ручного додавання бронювань викликає метод `check_room_dates(room_id,edate,qdate)`.

3.11 Тестування системи

Для тестування буде використовуватись уже розгорнута на сервері система.

Тестування програми проводилося у браузерях Google Chrome та Opera. Система тестувалася на коректне відображення компонентів та правильну роботу ключових елементів.

Надалі буде продемонстровано тестування в браузері Google Chrome.

Виконавши вхід на сайт з системою ми бачимо форму для створення нового запиту на бронювання, яка в одночас слугує і реєстраційною формою для нових клієнтів (рис. 3.20).

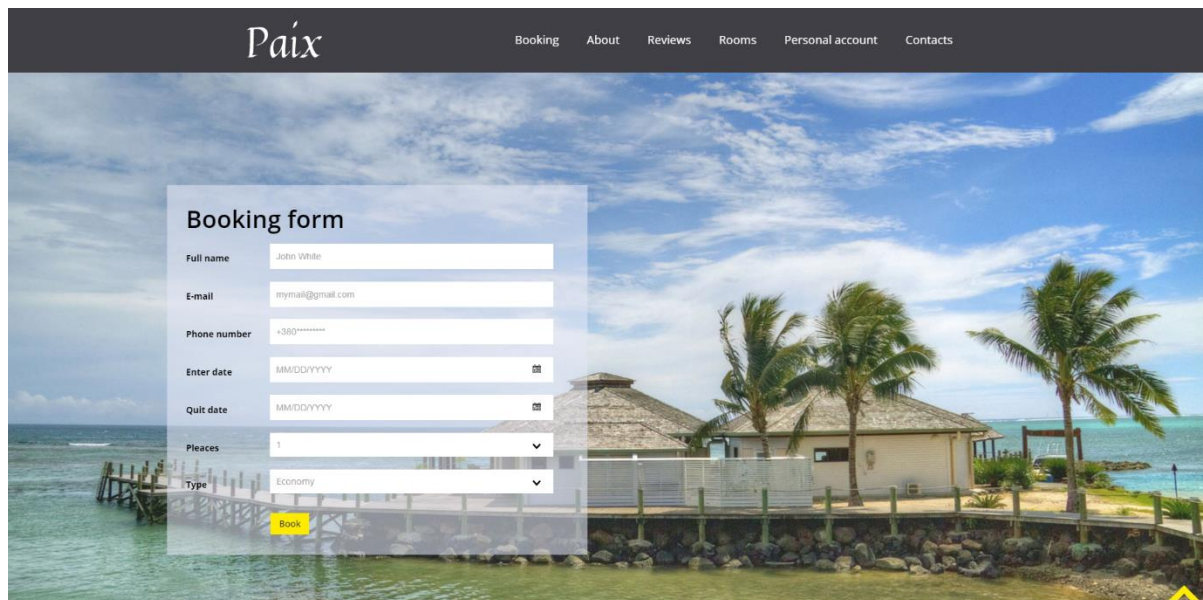


Рисунок 3.20 - Початкове вікно програми

Ввівши дані та натиснувши на кнопку «Book» ми отримаємо сповіщення про вдале надсилання даних з проханням перевірити електронну пошту(рис. 3.21).

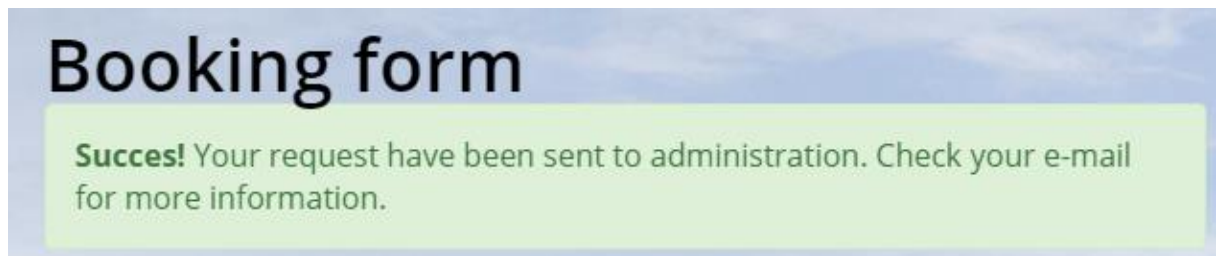


Рисунок 3.21 – Сповіщення

На електронну пошту надсилається пароль доступу до персонального кабінету, а для ідентифікації клієнта використовується email. Попастись у форму входу в систему можна натиснувши на кнопку «Personal account» у верхньому меню. Ввівши дані необхідні для авторизації(email та пароль). Клієнт попадає на головну сторінку особистого кабінету(рис. 3.22).

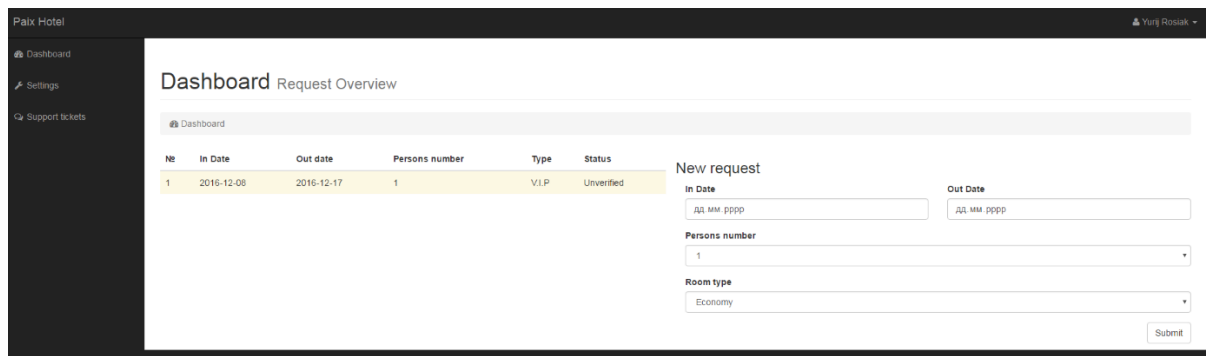


Рисунок 3.22 – Головна сторінка особистого кабінету

Натиснувши на своє ім'я(верхній правий кут екрану) клієнту відкриється випадаюче меню з посиланнями:

- «Profile» – відкриває сторінку зміни персональної інформації(ім'я, номер телефону, email).
- «Settings» – відкриває сторінку зміни паролю.
- «Logout» – переводить клієнта на сторінку авторизації.

Натиснувши на кнопку «Support tickets» клієнт буде переведений на сторінку тікет-ситсеми підтримки, де натиснувши кнопку «New ticket» форма для створення нового тікета (рис.3.23).

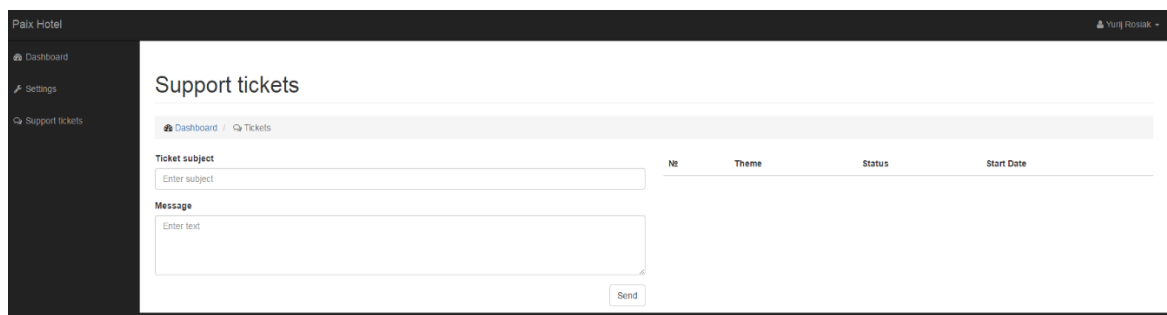


Рисунок 3.23 – Сторінка створення нового тікета

Заповнивши форму та надіславши її клієнт попадає на сторінку тікета, де він бачить усі повідомлення даного тікета(рис. 3.24).

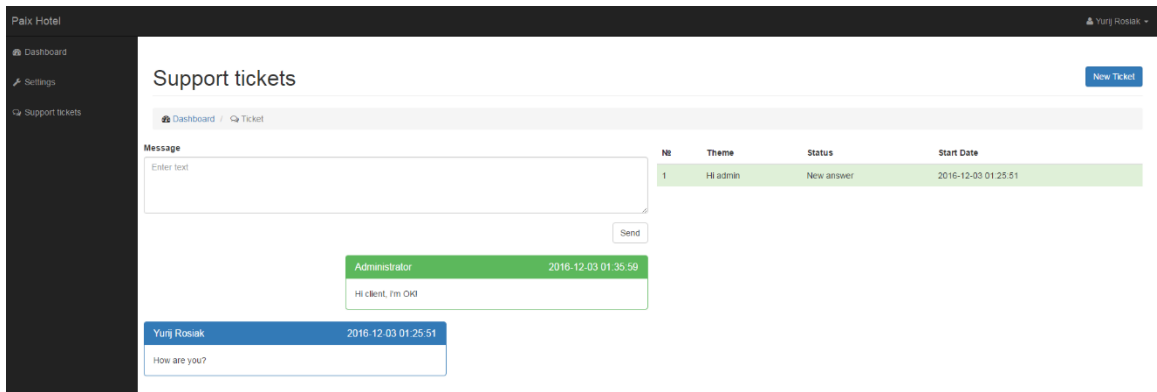


Рисунок 3.24 – Сторінка з повідомленнями тикета

Щоб увійти у адміністративну частину потрібно ввести в адресну стрічку браузера <domain_name>/admin. Ввівши дані у форму ми попадаємо на головну сторінку адміністративної частини(рис. 3.25).

На головній сторінці виводяться дані про запити на бронювання з інформацією про користувачів. Колір фону запису означає його статус: зелений – прийняті, жовтий – нерозглянуті, червоний – неможливі. Натиснувши на галочку запит буде прийнято. Натиснувши на гайковий ключ ми попадем на сторінку редагування запиту. Також зправа є вікно яке показує тикети без відповіді.

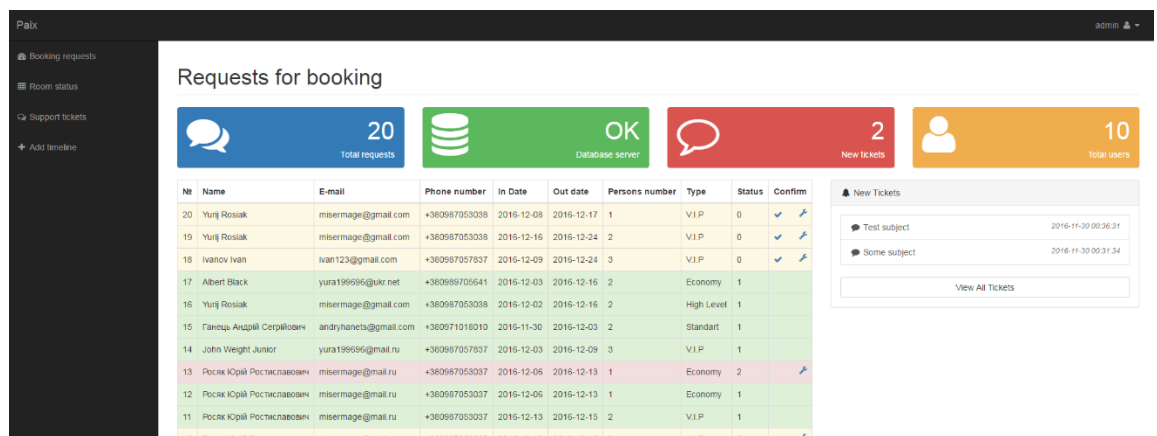


Рисунок 3.25 – Головна сторінка адміністративної частини

Натиснувши на кнопку «Room status» нам відкриється сторінка з списком усіх кімнат (рис. 3.26), колір «кімнати» зображає її статус на момент перегляду: зелений – вільна, червоний - зайнята. У верхньому правому куті є кнопка «Add

room» натиснувши на неї відкриється сторінка з формою додавання нової кімнати.

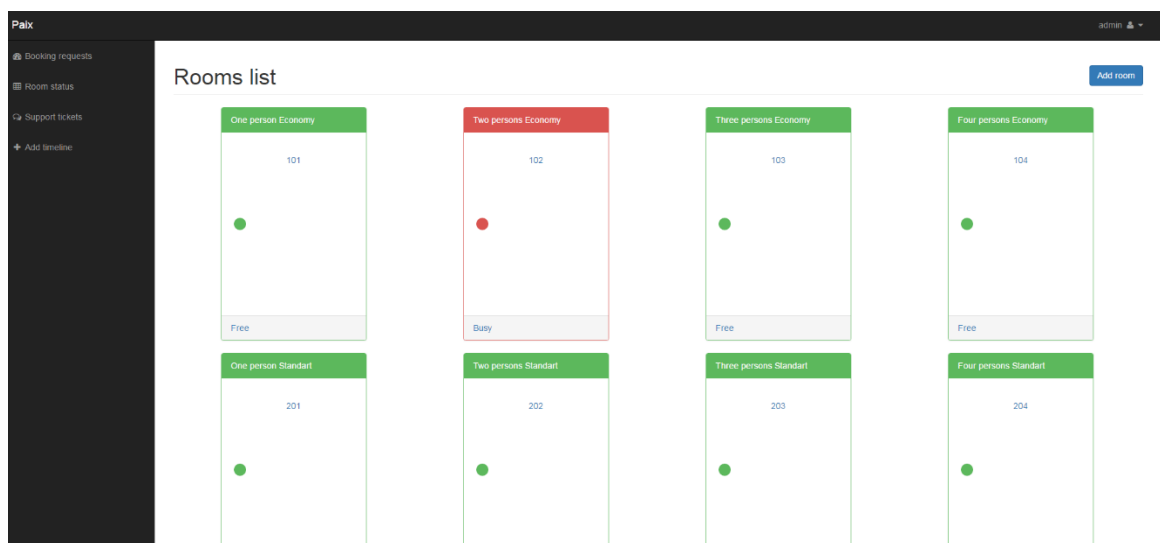


Рисунок 3.26 – Сторінка «Rooms status»

Натиснувши на будь-яку кімнату нам відкриється її коротка статистика. Та усі її записи бронювань (рис. 3.27).

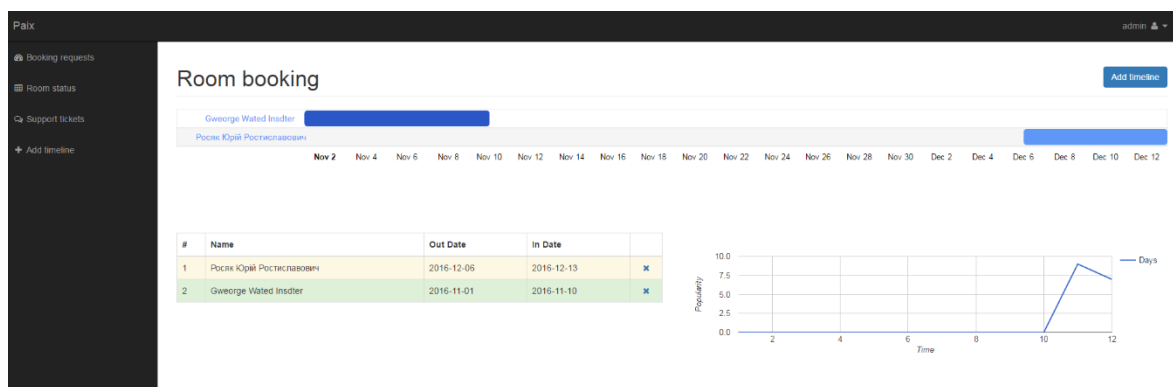


Рисунок 3.27 – Сторінка «Room booking»

Натиснувши на кнопку «Add timeline» нам відкриється форма додавання запису бронювання. Діаграма Ганта зображує в які дні забронював кімнату той чи інший клієнт. Графік що знаходиться нижче показує скільки днів кожного місяця даного року була зайнята кімната. У таблиці зліва виведено записи про бронювання, натиснувши на хрестик ми видаляємо запис.

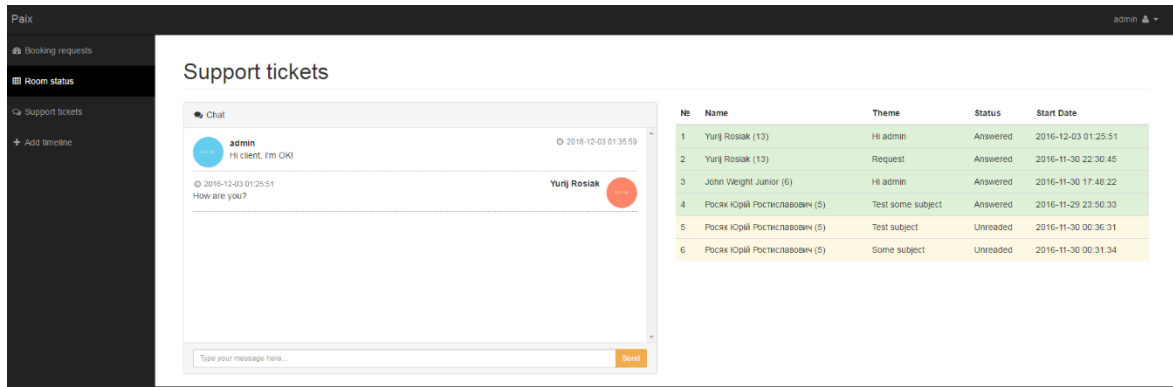


Рисунок 3.28 – Сторінка «Support tickets»

Перейшовши на сторінку «Support tickets» ми бачимо усі тікети з іменами користувачів які їх надіслали. Натиснувши на рядок таблиці нам відкриється сторінка тікету з усіма його повідомленнями (рис. 3.28).

3.12 Висновок до третього розділу

У третьому розділі були визначені цілі та вимоги розробки, здійсненне проектування системи керування готелем, де описана діаграма класів, прецедентів, баз даних та сама структура програми. В основній частині розділу був здійснений огляд коду декількох основних модулів продукту, а також зроблений аналіз методу багатокритеріального автоматизованого підбору номерів та його реалізація. В кінці розділу детально описується тестування програми з наведеними ілюстраціями.

4 СПЕЦІАЛЬНА ЧАСТИНА

Для розробки магістерської роботи – системи керування готелем, було використано стандарти написання коду. PHP Standards Recommendations [24] – це набір рекомендацій для розробників на PHP. Ставлення до PSR різне: від повного неприйняття, то фанатичною відданості. Сам по собі PSR з'явився як копіювання Java Community Process. Основне призначення PSR в тому, щоб надати PHP-розробникам якісь загальні концепції, які вже були перевірені і відпрацьовані. Стандартів PSR налічується дуже багато, але для написання магістерської роботи по системі курування готелем було використано лише 5 основних, опис яких наведений нижче.

4.1 PSR-0 – стандарт автозавантаження

Нижче представлені вимоги, обов'язкові до виконання в цілях забезпечення сумісності механізмів автозавантаження [25]:

- Повністю певний простір імен і ім'я класу повинні мати наступну структуру: `\ <Vendor Name> \ (<Namespace> \) * <Class Name>`.
- Кожне простір імен повинно починатися з простору імен вищого рівня, що вказує на розробника коду («ім'я виробника»).
- Кожне простір імен може включати в себе необмежену кількість вкладених підпросторів імен.
- Кожен роздільник простору імен при зверненні до файлової системи перетворюється в роздільник_імен_каталогів.
- Кожен символ `_` («знак підкреслення») в імені_класу перетворюється в роздільник_імен_каталогів. При цьому символ `_` («знак підкреслення») не володіє ніяким особливим значенням в імені простору імен (і не зазнає перетворень).
- При зверненні до файлової системи повністю певний простір імен і ім'я класу доповнюються суфіксом `.php`.

- В імені виробника, імені простору імен і імені класу допускається використання буквених символів в будь-яких комбінаціях нижнього і верхнього регістрів.

Нижче представлений приклад функції, що ілюструє, як описані вище вимоги впливають на процес автозавантаження в лістингу 4.1.

Лістинг 4.1 – Приклад стандарту автозавантаження

```
<?php
function autoload($className)
{
    $className = ltrim($className, '\\');
    $fileName  = '';
    $namespace = '';
    if ($lastNsPos = strrpos($className, '\\')) {
        $namespace = substr($className, 0, $lastNsPos);
        $className = substr($className, $lastNsPos + 1);
        $fileName  = str_replace('\\', DIRECTORY_SEPARATOR,
$namespace) . DIRECTORY_SEPARATOR;
    }
    $fileName .= str_replace('_', DIRECTORY_SEPARATOR,
$className) . '.php';

    require $fileName;
}
```

Представлені тут стандарти повинні сприйматися як мінімально необхідний набір правил для забезпечення сумісності автозавантажувача. Ви можете перевірити, наскільки ви прямуєте зазначених вимог, скориставшись наступним прикладом реалізації `SplClassLoader` (орієнтований на завантаження класів PHP 5.3) [24].

4.2 PSR-1 – базовий стандарт оформлення коду

Даний розділ описує стандартні елементи, які є істотними для забезпечення високої технічної сумісності коду, створеного і / або підтримуваного різними розробниками [26].

Слова «необхідно» / «повинно» («must»), «неприпустимо» («must not»), «потрібно» («required»), «потрібно» («shall»), «не дозволяйте» («shall not»), «слід» («should»), «не слід» («should not»), «рекомендується» («recommended»), «може» / «можливо» («may») і «необов'язково» («optional») в цьому документі слід розуміти так, як це описано в RFC 2119 (і його перекладі).

Загальні положення:

- У файлах необхідно використовувати тільки теги `<? Php` і `<? = .`
- Файли необхідно представляти тільки в кодуванні UTF-8 без BOM-байта.
- У файлах слід або оголошувати структури (класи, функції, константи і т.п.), або генерувати побічні ефекти (виконувати дії) (наприклад: передавати дані в вихідний потік, модифікувати настройки і т.п.), але не слід робити одночасно і те і інше.
- Імена просторів імен і імена класів повинні слідувати стандарту PSR-0.
- Імена класів повинні бути оголошені з використанням т.зв. « StudlyCaps » (кожне слово починається з великої літери, між словами немає роздільників).
- Константи класів повинні бути оголошені виключно в верхньому регістрі з використанням символу підкреслення для поділу слів.
- Імена методів повинні бути оголошені з використанням т.зв. « CamelCase » (перше слово пишеться в нижньому регістрі, далі кожне слово починається з великої літери, а між словами немає роздільників).
- PHP-код обов'язково слід укладати в повну версію (`<? Php?>`) тегів, вкорочену (скорочену запис echo) версію (`<? =?>`) тегів і неприпустимо укладати ні в які інші різновиди тегів.
- PHP-код повинен бути представлений тільки в кодуванні UTF-8 без BOM-байта.

- У файлах слід або оголошувати структури (класи, функції, константи і т.п.) і не створювати побічних ефектів (наприклад: передавати дані в вихідний потік, модифікувати настройки і т.п.), або реалізовувати логіку, яка породжує побічні ефекти, але не слід робити одночасно і те, і інше.

Під «побічними ефектами» розуміється реалізація логіки, не пов'язаної з оголошенням класів, функцій, констант і т.п. - навіть підключення зовнішнього файлу вже є «побічним ефектом».

«Побічні ефекти» включають (але не обмежуються цим переліком): передачу даних в вихідний потік, явне використання `require` або `include`, зміна налаштувань, генерування помилкових ситуацій або породження винятків, зміна глобальних або локальних змінних, читання з файлу або запис в файл і т.д [24].

4.3 PSR-2 – рекомендації по оформленні коду

Дані рекомендації розширюють і доповнюють базовий стандарт оформлення коду PSR-1.

Мета даних рекомендацій - зниження складності сприйняття коду, написаного різними авторами; вона досягається шляхом розгляду серії правил і очікувань щодо форматування PHP-коду.

Стилістичні правила, представлені тут, отримані шляхом узагальнення досвіду різних проектів. Співпраця багатьох авторів з багатьох проектів дозволяє виробити єдиний набір принципів і використовувати його в цих проектах. Таким чином, користь представлених рекомендацій - не стільки в самих рекомендаціях, скільки в їх розповсюдженні [26].

Загальні положення:

- Код повинен бути оформлений відповідно до стандарту PSR-1.
- Для оформлення відступів повинні використовуватися чотири пробіли (але не знак табуляції).

- Неприпустимо жорстко обмежувати довжину рядка; м'яке обмеження повинно складати 120 символів; слід намагатися, щоб довжина рядка становила 80 символів або менше.

- Після визначення простору імен (namespace) і після блоку імпорту просторів імен (use) повинна бути один порожній рядок.

- Відкриває фігурна дужка у визначенні класу повинна розташовуватися на новій рядку, а закриває фігурна дужка повинна розташовуватися на наступному рядку після тіла класу.

- Відкриває фігурна дужка у визначенні методу повинна розташовуватися на новій рядку, а закриває фігурна дужка повинна розташовуватися на наступному рядку після тіла методу.

- Область видимості повинна бути вказана явно для всіх властивостей і методів; модифікатори abstract і final повинні розташовуватися перед модифікаторами області видимості; модифікатор static повинен розташовуватися після модифікаторів області видимості.

- Після ключових слів в керуючих конструкціях повинен розташовуватися один пробіл, а після викликів функцій і методів - не має.

- Відкриває фігурна дужка в керуючих конструкціях повинна розташовуватися в тому ж рядку, що і сама конструкція, а закриває фігурна дужка повинна розташовуватися на наступному рядку після тіла конструкції.

- Після відкриває круглої дужки і перед закриває круглою дужкою в керуючих конструкціях не повинно бути пробілу.

Приклад оформлення за вищеперерахованими стандартами наведений в лістингу 4.2.

Лістинг 4.2 – Стандарти оформлення коду

```
<?php  
  
namespace Vendor\Package;  
  
use FooInterface;  
use BarClass as Bar;  
use OtherVendor\OtherPackage\BazClass;
```

```

class Foo extends Bar implements FooInterface
{
    public function sampleFunction($a, $b = null)
    {
        if ($a === $b) {
            bar();
        } elseif ($a > $b) {
            $foo->bar($arg1);
        } else {
            BazClass::bar($arg2, $arg3);
        }
    }

    final public static function bar()
    {
        // тело метода
    }
}

```

У майбутньому дані рекомендації можуть бути переглянуті і розширені, щоб охопити ті чи інші елементи коду і практики оформлення [24].

4.4 PSR-3 – інтерфейс протоколювання

Даний документ описує загальний інтерфейс бібліотек протоколювання.

Основна мета даного документа – дозволити бібліотекам отримувати об'єкт `Psr \ Log \ LoggerInterface` і використовувати його простим і універсальним чином для реалізації протоколювання. У разі, якщо якийсь фреймворк або CMS потребує розширеному функціоналі, можна розширювати даний інтерфейс, але слід зберігати сумісність з описаними в даному документі правилами. Це дозволить стороннім бібліотекам, застосовуваним при розробці програми, використовувати централізовану систему протоколювання.

Слово «розробник» (implementor) в даному документі слід розуміти як «автор, який реалізує інтерфейс `LoggerInterface` в деяких бібліотеці або

фреймворку, пов'язаних з протоколюванням». Користувачі систем протоколювання згадуються як просто «користувачі» [27].

4.5 PSR-4 – покращення автозавантаження

Даний стандарт описує специфікацію автозавантаження класів на основі шляхів до файлів. Він повністю сумісний (і може використовуватися як доповнення) з будь-якої іншої специфікацією автозавантаження, включаючи PSR-0. Даний стандарт також описує правила розміщення файлів, призначених для автозавантаження [28].

Загальні положення:

- Тут під «класом» слід розуміти також інтерфейси (interface), домішки (trait) і інші подібні конструкції.
- Повністю певний ім'я класу повинно мати наступну структуру: \ <Простір Імен> (\ <Підпростір Імен>) * \ <Ім'я Класу>.
- Повністю певний ім'я класу повинно починатися з простору імен вищого рівня, що вказує на розробника коду («ім'я виробника»).
- Повністю певний ім'я класу може включати в себе одне або більше підпросторів імен.
- Повністю певний ім'я класу повинно закінчуватися ім'ям класу.
- Символ _ («знак підкреслення») не володіє ніяким особливим значенням в повністю певному імені класу.
- У повністю певному імені класу можна використовувати літерні символи в будь-яких комбінаціях нижнього і верхнього регістрів.
- Всі імена класів повинні бути використані з дотриманням регістрочуттєвості.

Під час завантаження файлу, відповідного повністю певного імені класу, використовуються наступні правила:

- Послідовність з одного і більше просторів і підпросторів імен (не включаючи провідний роздільник просторів імен) в повністю певному імені

класу (т.зв. «префікс просторів імен») повинна відповідати хоча б одному «базовому каталогу».

- Послідовність підпросторів імен після «префікса простору імен» відповідає підкаталогу в «базовому каталозі», при цьому роздільники просторів імен \ відповідають розділювачам каталогів /. Ім'я підкаталогу і ім'я підпростору імен повинні збігатися аж до регістру символів.

- Ім'я класу, яким завершує собою повністю певне ім'я, відповідає імені файлу з розширенням .php. Файл та ім'я класу повинні збігатися аж до регістру символів.

У реалізації автозавантажувач неприпустимо породжувати виключення, помилкові ситуації будь-якого рівня і не слід повертати яке б то не було значення [24].

4.6 Висновок до четвертого розділу

В четвертому розділі було розглянуто набір рекомендацій для розробників на PHP – PHP Standards Recommendations, який слугує стандартизації концепцій програмування в PHP. В магістерській роботі «Система керування готелем з автоматизованим багатокритеріальним підбором номерів» було використано дані стандарти, які допомагають швидше розібратись в коді та систематизувати його.

Мета розділу полягає в тому, щоб забезпечити взаємодію компонентів та створити загальну технічну базу для впровадження перевірених концепцій оптимальних методів програмування та тестування для написання коду магістерської роботи.

5 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Головною метою розділу є обґрунтування економічної ефективності створення та впровадження автоматизованої системи керування готелем.

Щоб виконати оцінку економічної ефективності необхідно розрахувати трудомісткість реалізації проекту, витрати на оплату праці найманим працівникам, витрати апаратного і програмного забезпечення, амортизаційні відрахування, витрати енергоресурсів та інші витрати які є основними пунктами виконання обчислень, а також показники економічної ефективності розробки проекту.

5.1 Розрахунок норм часу на виконання науково-дослідної роботи

Реалізація автоматизованої системи керування готелем складається із підготовчого етапу, етапу технічної пропозиції, створення технічного завдання, проектування системи, практичної реалізації, тестування, верифікації та заключного етапу.

Кожен із етапів реалізації проекту характеризується метою та змістом, оцінкою часу виконання, кількістю та спеціалізацією виконавців, а також приблизною оцінкою вартості.

Для реалізації проекту системи керування готелем будуть найняті працівники, а саме проектний менеджер, інженер-програміст та тестувальник.

Норми часу на виконання розраховуватимуться на основі середнього часу виконання стадії в годинах, що наведені в таблиці 5.1 разом із інформацією про виконавців і сумарною кількістю затраченого часу.

Таблиця 5.1 – Операції технологічного процесу та їх час виконання

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Підготовча стадія	Проектний менеджер	10
		Інженер-програміст	5
2	Технічна пропозиція	Проектний менеджер	5
		Інженер-програміст	10
3	Створення технічного завдання	Проектний менеджер	10
		Інженер-програміст	10
4	Проектування системи	Інженер-програміст	60
5	Практична реалізація	Інженер-програміст	80
6	Тестування системи	Тестувальник	30
7	Верифікація системи	Тестувальник	10
		Інженер-програміст	10
		Проектний менеджер	10
8	Створення документації	Інженер-програміст	30
9	Заключна стадія	Проектний менеджер	10
Разом			290

В підсумку на реалізацію автоматизованої системи керування готелем з використанням інформаційних технологій розпізнавання образів необхідно 290 людино-годин, залучення трьох спеціалістів та виконання дев'яти різноманітних стадій реалізації проекту.

5.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Визначення витрат на оплату праці та відрахувань на соціальні заходи прямо залежить від кількості витраченого працівниками часу на роботу, ставки в годину чи місяць, кількість відрахувань на соціальні заходи встановлених в законному порядку на час розрахунку.

В результаті розрахунку потрібно визначити основну та додаткову заробітну плату, витрати на соціальні заходи та на основі цих даних визначити сумарні витрати на оплату праці.

Основна заробітна плата нараховується за виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час.

При розрахунку заробітної плати кількість робочих днів у місяці слід в середньому приймати – 24,5 дні/міс., або ж 196 год./міс. (тривалість робочого дня – 8 год.).

Наймані працівники для розробки інформаційної системи управління готелем працюють згідно контракту, в якому вказано їхню погодинну ставку. Тобто розрахунок заробітної плати працівників відбуватиметься на базі тарифної ставки та кількості відпрацьованих годин.

У штаті найманих працівників для розробки інформаційної системи залучено проектного менеджера, інженера-програміста і тестувальника.

Тарифні ставки учасників процесу розробки інформаційної системи управління готелем:

- Проектний менеджер – 160 грн./год.
- Інженер-програміст – 140 грн./год.
- Тестувальник – 100 грн./год.

Основна заробітна плата розраховується за формулою 5.1:

$$Z_{\text{осн.}} = T_c \cdot K_r, \quad (5.1)$$

де T_c – тарифна ставка, грн.; K_r – кількість відпрацьованих годин.

Оскільки всі види робіт в виконує три спеціаліста, то основна заробітна плата буде розраховуватись за даною формулою 5.1;

$$Z_{\text{осн.}} = 160 \cdot 45 + 140 \cdot 205 + 100 \cdot 40 = 39900 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 %% від суми основної заробітної плати й визначається за формулою 5.2.

Коефіцієнт додаткових виплат працівникам становить 0,1.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (5.2)$$

де $K_{\text{допл}}$ – коефіцієнт додаткових виплат працівникам

$$Z_{\text{дод.}} = 39900 \cdot 0,15 = 5985 \text{ грн.}$$

Звідси загальні витрати на оплату праці (фонд заробітної плати) визначаються за формулою 5.3:

$$B_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (5.3)$$

$$B_{\text{о.п.}} = 39900 + 5985 = 45885 \text{ грн.}$$

З цієї суми утримуються обов'язкові відрахування на заробітну плату:

- Єдиний соціальний внесок (ЄСВ), що становить 22%%;
- Військовий збір (ВЗ), що становить 1,5%%;

Сума відрахувань становить 23,5%% від фонду оплати праці та визначається за формулою 5.4:

$$B_{\text{с.з.}} = \Phi_{\text{оп.}} \cdot 0,235 \quad (5.4)$$

де $\Phi_{\text{оп}}$ – фонд оплати праці, грн.

$$B_{\text{с.з.}} = 45885 \cdot 0,235 = 10782,97$$

Усі витрати обчислено та наведено в таблиці 5.2 та обчислюються за формулою 5.5:

$$V_{\text{зп}} = \text{ФЗП} + \text{ФОП} \quad (5.5)$$

$$V_{\text{зп}} = 39900 + 10782,975 = 50682,97 \text{ грн.}$$

Таблиця 5.2 – Розрахунки витрат на оплату праці

№з/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на плату праці, грн. (6=3+4+5)
		Тарифна ставка, грн.	Кількість відпрацьованих год.	Фактично нарах. з/пл., грн.			
А	Б	1	2	3	4	5	6
1.	Проектний менеджер	160	45	7200	1080	-	-
2.	Інженер-програміст	140	205	28700	4305	-	-
3.	Тестувальник	100	40	4000	600	-	-
Разом		300	290	39900	5985	10782,97	50682,97

Згідно проведених розрахунків витрат на оплату праці можна зробити висновок, що загальні витрати становлять 50682,97 грн.

5.3 Розрахунок матеріальних витрат

Матеріальні витрати є невід'ємною частиною автоматизованої системи управління готелем та визначаються як добуток кількості витрачених матеріалів та їх ціни за формулою 5.6:

$$M_{ei} = q_i \cdot p_i, \quad (5.6)$$

де: q_i – кількість витраченого матеріалу i -го виду; p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити за формулою 5.7:

$$Z_{м.в.} = \sum M_{ei}. \quad (5.7)$$

Результати проведених розрахунків наведено у таблиці 5.3.

Таблиця 5.3 – Результати розрахунків матеріальних витрат.

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Фактично витрачено матеріалів	Ціна одиниці, грн.	Загальна сума витрат, грн.
1	Інтернет	шт.	2	150	300,00
2	Папір для друку	листів	500	0,20	100,00
3	Чорнила для принтера	шт.	1	110,00	110,00
Всього					510,00

Згідно проведених розрахунків, матеріальні витрати становлять 510,00 грн.

5.4 Розрахунок витрат на електроенергію

Однією із статей витрат є витрати на електроенергію під час проходження усіх етапів реалізації кінцевого продукту.

Затрати на електроенергію одиниці обладнання визначаються за формулою 5.8:

$$Z_e = W \cdot T \cdot S, \quad (5.8)$$

де W – необхідна потужність, кВт; T – кількість годин на реалізацію розробки; S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів. Отже, 1 кВт з ПДВ коштує 2,42 грн.

Потужність комп'ютерів для реалізації кінцевого продукту – 500 Вт, кількість годин роботи обладнання згідно таблиці 5.1 – 290 годин.

Визначимо витрати на електроенергію згідно формули 5.11:

$$Z_g = 0,5 \cdot 290 \cdot 2,80 = 390,9 \text{ грн.}$$

Згідно формули затрати на електроенергію становлять 350,9 грн.

5.5 Розрахунок суми амортизаційних відрахувань

Для будь якої діяльності характерною є властивість зношування на зниження якості властивостей інструментарію та фондів за допомогою яких ведеться діяльність.

Для вирішення проблеми із відновленням даних фондів використовується амортизація, що являє собою процес трансформації вартості основних фондів на вартість продукції, яка щойно була створена, задля повного відновлення основних фондів.

Для визначення амортизаційних відрахувань використовується формула 5.9:

$$A = \frac{B_B \cdot H_A}{100\%}$$

(5.9)

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 %% (квартальна – 15 %%).

Річний робочий фонд становитиме 2352 годин, так як робочий день становить 8 годин, а кількість робочих днів в місяці становить 24,5 годин.

Для даної розробки засобом розробки є комп'ютер. Його сума становить 25500 грн. Отже, амортизаційні відрахування будуть рівні:

$$A = 25500 \cdot 5\% / 100\% = 1275 \text{ грн.}$$

Згідно проведених обчислень амортизаційні відрахування становлять 1275 грн.

5.6 Обчислення накладних витрат

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_g = B_{o.n.} \cdot 0,2 \dots 0,6, \quad (5.10)$$

де H_g – накладні витрати.

Отже, накладні витрати становлять згідно формули 5.10:

$$H_g = 121325 \cdot 0,2 = 24265 \text{ грн.}$$

Накладні витрати згідно розрахунку формули, становить 24265 грн.

5.7 Складання кошторису витрат та визначення собівартості науково-дослідницької роботи

Результати проведених вище розрахунків наведено у таблиці 5.4.

Таблиця 5.4 – Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В %% до загальної суми
Витрати на оплату праці	50682,97	57,65
Відрахування на соціальні заходи	10782,97	12,2
Матеріальні витрати	510	0,57
Витрати на електроенергію	390,9	0,44
Амортизаційні відрахування	1275	1,45
Накладні витрати	24265	27,6
Собівартість	87906,84	100

Собівартість (C_6) програмного продукту розраховуємо за формулою:

$$C_6 = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_6 + A + H_6. \quad (5.11)$$

Отже, собівартість програмного продукту дорівнює:

$$C_6 = 50682,975 + 10782,975 + 510 + 390,9 + 1275 + 24265 = 87906,84 \text{ грн.}$$

Загальний кошторис витрат та визначення собівартості науково-дослідницької роботи становить 87906,84 грн.

5.8 Розрахунок ціни програмного продукту

Ціну науково-дослідної роботи можна визначити за формулою:

$$Ц = \frac{C_6 \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \quad (5.12)$$

де $P_{рен.}$ – рівень рентабельності, 30 %%; K – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та

мікропроцесорних систем); $B_{н.і}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту); $ПДВ$ – ставка податку на додану вартість, (20 %).

Оскільки розробка є прикладною то для розрахунку ціни не потрібно вказувати коефіцієнти K та V_i , оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ) \quad (5.13)$$

Звідси ціна на роботу складе:

$$Ц = 87906,84 \cdot (1 + 0,3) \cdot (1 + 0,2) = 137134,67 \text{ грн.}$$

Загальний розрахунок ціни програмного продукту становить 137134,67 грн.

5.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{\Pi}{C_B} \quad , \quad (5.14)$$

де Π – прибуток; C_B – собівартість.

Плановий прибуток ($\Pi_{пл}$) знаходимо за формулою:

$$\Pi_{пл} = Ц - C_v \quad (5.15)$$

Розраховуємо плановий прибуток:

$$P_{пл} = 137134,67 - 87906,84 = 49227,83 \text{ грн.}$$

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_p = \frac{P}{C_B} \quad (5.16)$$

Тоді,

$$E_p = 49227,83 / 87906,84 = 0,56.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p}, \quad (5.17)$$

Термін окупності дорівнює:

$$T_p = 1 / 0,56 = 1,78 \text{ р.}$$

Згідно формул плановий прибуток від розробки становить 49086,492 грн., економічна ефективність дорівнює 0,56, а термін окупності становить 1,78 роки що вважається доцільним та економічно вигідним.

5.10 Висновки до п'ятого розділу

В частині з обґрунтування економічної ефективності дипломної роботи освітнього рівня «магістр» було розраховано основні техніко-економічні показники розробки системи керування готелем (див. таблиця 5.5).

Таблиця 5.5 – Техніко-економічні показники науково-дослідної роботи

№ п/п	Показник	Значення
1.	Собівартість, грн.	87906,84
2.	Плановий прибуток, грн.	49227,83
3.	Ціна, грн.	137134,67
4.	Економічна ефективність	0,56
5.	Термін окупності, рік	1,78

Орієнтоване значення економічної ефективності становить 0,56 що є достатньо високим значенням.

Період окупності повинен варіюватися від 1 до 3 років, тоді розвиток вважається доцільним та економічно вигідним. Термін окупності даної роботи становить 1,78 років.

На основі проведених розрахунків можна зробити висновок, що створення автоматизованої системи управління готелем з використанням багатокритеріального підбору номерів є доцільною у зв'язку з невеликим терміном окупності та великим обсягом планового прибутку.

6 ЕКОЛОГІЯ

6.1 Аналіз сучасних програмних продуктів для обробки великих масивів екологічної інформації.

Оперативна, якісна і точна обробка великих масивів статистичної інформації може бути виконана лише з використанням сучасних засобів обчислювальної техніки. Наявність потужних, надійних і разом з тим простих в експлуатації програмних продуктів статистичного аналізу звільняє дослідника від рутинних операцій, розширює сферу застосування статистичних методів в різних галузях людської діяльності, сприяє появі якісно нових можливостей статистичного аналізу і моделювання даних. Використання пакетів прикладних програм - це єдиний реальний практичний інструмент розв'язування задач багатofакторного кореляційно-регресійного та аналізу в багатовимірному просторі.

Програмне забезпечення статистичних досліджень досить розвинуте. Сучасний ринок програмних продуктів пропонує різноманітні пакети програм для статистичної обробки даних - BMDP, SPSS, SAS, Systat, Minitab, S-Plus, Statgraphics, Statistica

Використання згаданих пакетів програм дає змогу автоматизувати процес статистичного дослідження в таких напрямках:

- створення файлів даних і таблиць;
- групування даних;
- графічний аналіз даних;
- розрахунок варіаційних характеристик вибірових сукупностей;
- побудова рядів розподілу;
- аналіз рядів динаміки і прогнозування їх майбутніх рівнів;
- кореляційно-регресійний аналіз;
- багатомірний аналіз.

З 1995 р. Світовим лідером на ринку статистичного програмного

забезпечення визнається інтегрована система 8шзііса для Statistica для Windows (версія 5.0), розроблена фірмою Stat Soft. Перша версія програми з'явилася у 1991р. для операційної системи MS-DOS і була новим напрямом розвитку статистичного програмного забезпечення. В ній реалізовано графічно-орієнтований підхід до статистичного аналізу даних, суть якого полягає в отриманні всебічного візуального представлення інформації на всіх етапах статистичної обробки даних.

Багатофункціональна, графічно орієнтована на обробку масових даних система Statistica відповідає основним стандартам Windows (динамічний обмін даними з іншими додатками, підтримка основних операцій з буфером обміну, робота в мережевому середовищі та інші).

Передусім це стандарти користувачького інтерфейсу — MDI, використання буфера-обміну, механізму динамічного зв'язку (DDE) з іншими додатками; система підтримує всі операції, реалізовані за допомогою методу Drag-and-Drop — «Перетягти та опустити», включаючи автозаповнення, інші.

Складніші процедури обробки даних у системі Statgraphucs виконує спеціалізований модуль Data Management — «Управління даними», а для обробки великих масивів даних або даних з довгими текстовими значеннями застосовують процедури Megafile Manager Data — «Менеджера мегафайлів».

Система Statgraphucs працює з чотирма типами документів. Це:

- електронна таблиця Spreadsheet, призначена для введення і перетворення первинних даних;
- електронна таблиця Scrollsheet — для виведення результатів аналізу;
- графік — для візуалізації результатів обробки та аналізу даних;
- звіт — файл у формі RTF (розширений текстовий формат), в якому зберігається текстова, числова і графічна інформація.

Система 8шШіса включає модуль Anova/Manova — «Дисперсійний аналіз», увесь арсенал методів багатовимірного аналізу (кластерний, дискримінантний, факторний аналіз, факторне шкалювання, канонічні кореляції).

Особливе місце посідає модуль Serpath — «Моделювання взаємозв'язків системами структурних рівнянь».

Зазначені модулі покривають практично весь спектр сучасних методів статистичного дослідження і моделювання. Запуск модуля здійснюється через перемикач модулів — Module Swither. У кожному модулі робота починається із «Стартової панелі», де відкривається файл первинних даних, вибирається процедура обробки даних і визначаються відповідні їй параметри.

У системі Statistica реалізовано принцип постійного логічного підказування. Якщо користувач не може визначитися щодо наступного кроку діалогу, через команду Епіг система сама спрямує до відповідного діалогового вікна. Якщо виникають складнощі з вибором параметрів обчислювальної процедури, вони задаються системою «за умовчанням».

Використання сучасних комп'ютерних технологій обробки даних, інтерактивний спосіб взаємодії з системою перетворюють статистичний аналіз, моделювання та прогнозування в захоплююче дослідження закономірностей навколишнього світу. Завдяки різноманітним формам організації діалогу, максимально простій із звичними для статистики термінами мові спілкування, наявності контекстно-залежної довідкової системи, мові програмування Statistica Basic пакет є ефективним інструментом проведення статистичного дослідження як для користувача-початківця, так і для професіонала[14].

6.2 Зниження енергоємності та енергозбереження.

Споживання енергії - одна з ключових проблем сучасних обчислень, будь то бездротові вбудовані клієнтські пристрої або високоефективні обчислювальні центри. Здатність розробляти енергоефективне програмне забезпечення має вирішальне значення, оскільки використання даних та обробка даних постійно зростає у всіх сферах суспільства. Необхідність енергоефективних обчислень обумовлена не лише впливом на навколишнє

середовище. Швидше, нам потрібні енергоефективні обчислення, щоб навіть виконати прогнозовані тенденції.

Проект "Надлишок", що фінансується ЄС, мав на меті підняття питання інтегрованого підходу для покриття всіх системних шарів від апаратного до програмного забезпечення для користувачів, а також обмеження, спричинені для експлуатація існуючих рішень та їх енергоефективність. Вони спочатку проаналізували, де витрачаються енергоефективність, і, виходячи з цих знань, вони розробили основу, яка повинна забезпечити швидкий розвиток енергоефективного виробництва програмного забезпечення.

Проект Excess здійснив основні кроки щодо надання набору інструментів та моделей розробникам програмного забезпечення та дизайнерам систем, щоб вони могли програмувати енергоефективним способом. Коробка інструментів охоплює принципово нові енергозберігаючі апаратні компоненти, такі як платформа Movidius Myriad, до складних ефективних бібліотек та алгоритмів.

Тести проводяться на великих агрегаціях потокової передачі даних, що є звичайною операцією, що використовується в аналітиці даних в реальному часі, показує вражаючі результати. Використовуючи систему Excess, програміст може запропонувати в 54 рази більш енергоефективне рішення порівняно зі стандартною реалізацією на високоякісних процесорах ПК. Цілісний підхід Excess спочатку представляє переваги обладнання, використовуючи вбудований процесор, а потім продовжує найкращий спосіб розділити обчислення всередині процесора, щоб ще більше підвищити продуктивність[13].

Операційна система комп'ютера має можливість проводити регулювання параметрів енергоспоживання або автоматичне відключення комп'ютеру, які можна адаптувати для свого робочого режиму. Крім цього, дуже важливо вимикати комп'ютер на той час, коли він не буде використовуватися. Це відноситься і до тих, хто звик ввечері вимикати комп'ютер не вимикаючи дисплея.

Європейська комісія Energy Star надає детальну інформацію про екологічне маркування, а також про споживання енергії сучасними офісними комп'ютерами на сайті www.eu-energystar.org. На цьому веб-сайті є калькулятор для підрахунку енергії, яку споживає комп'ютер, монітор і проекційне обладнання в компаніях та в громадському секторі, а також в кінцевих споживачів. Він дозволяє детально порівняти споживання енергії і загальні витрати, враховуючи багато факторів.

Приклад споживання електричної енергії комп'ютером можна побачити на рисунку 6.1.

	Офісний комп'ютер	Потужний ігровий комп'ютер
Сумарна електрична потужність компонентів комп'ютера та монітора	0,4 кВт	0,8 кВт
Споживання електричної енергії у годину	0,4 кВт*год	0,8 кВт*год
Витрати на роботу комп'ютера за годину	0,4 грн.	0,8 грн.
Споживання електроенергії за умови роботи 5 годин на добу	2 кВт*год	4 кВт*год
Споживання електроенергії за місяць	60 кВт*год	120 кВт*год
Витрати роботу комп'ютера за місяць	60 грн.	120 грн.

Рисунок 6.1 - Споживання електричної енергії комп'ютером

Згідно сайту «Вікіпедія» на планеті близько 3 885 567 619 користувачів інтернету, якщо кожен користувач заощадить 5 кВт*год електроенергії у місяць, це збереже понад 15 000 000 000 кВт*год електроенергії, що можна порівняти з річним виробництвом електроенергії у деяких державах[12].

6.3 Висновок до шостого розділу

У цьому розділі було розглянуто сучасних програмних продуктів для обробки великих масивів екологічної інформації та зниження енергоємності та енергозбереження в комп'ютерних технологіях. В результаті опрацювання розділу було зроблено висновок ,якщо кожен власник комп'ютерної техніки буде відключати свій комп'ютер або переводити його у режим сну на час коли він йому не потрібен, це приведе до величезної економії електроенергії.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз нового в законодавстві з питань охорони праці станом на 2019р.

НПАОП — це правила, норми, регламенти, положення, стандарти, інструкції та інші документи, обов'язкові для виконання. Реєстр нормативно-правових актів з охорони праці – це банк даних, який складається і ведеться з метою забезпечення єдиного обліку та формування відповідного інформаційного фонду цих актів. У Реєстрі документи згруповані за видами економічної діяльності. Дані про затвердження та введення в дію нормативно-правових актів з охорони праці оформлюють Показчиком.

Саме цим Показчиком мають користуватися підприємства, коли формують базу документів з охорони праці. Показчик оновлюють один раз на півріччя. Показчик НПАОП затверджений відповідно до: Положення про Державну службу України з питань праці, затвердженого постановою Кабміну 11 лютого 2015 року № 96; Положення про Державний реєстр нормативно-правових актів з охорони праці, затвердженого наказом Державного комітету України з нагляду за охороною праці від 8 червня 2004 року № 151[22].

Рівень промислового травматизму в Україні вражає. Саме тому Уряд звернув увагу на цю проблему, затвердивши Концепцію реформування системи управління охороною праці в Україні. Передбачено протягом 2019-2020 років сформувати нову систему запобігання виробничим ризикам шляхом впровадження ризикоорієнтованого підходу.[24] Стверджується, що нинішня система орієнтована на реагування, а не на запобігання ризикам. Нова редакція Закону «Про охорону праці» розроблена з метою гармонізації національного законодавства з охорони праці до рамкової Директиви Ради ЄС. Робоча назва законопроекту — «Про безпеку праці та здоров'я працівників», який за планами Уряду має бути прийнятий у 2020 році.

Проект розроблено Мінсоцполітики за участю Держпраці та менеджерів проекту ЄС-МОП «Зміцнення адміністрації праці з метою покращення умов праці і подолання незадекларованої праці». Закон поширюватиметься не лише на працівників і роботодавців, а й на осіб, які надають послуги за цивільно-правовим договором, та практикантів.

Також і високий показник тіньової зайнятості негативно позначається на економіці країни. В Україні декілька мільйонів громадян працюють без офіційного оформлення трудових відносин з роботодавцем. «Станом на сьогодні ми можемо говорити про декілька мільйонів наших громадян, які працюють поза межами офіційних трудових відносин», — таким чином громадяни погоджуються на відсутність соціальних гарантій і прав повноцінних членів суспільства.

Говорячи про результати проекту, сотні тисяч оформлених відповідно до законодавства працівників отримали соціальний пакет, а також право об'єктивного розслідування нещасних випадків. При цьому Україна завдяки проекту отримала комплекс рекомендацій щодо впровадження стандартів Євросоюзу та Міжнародної організації праці, а понад дві тисячі експертів пройшли навчання та підвищили свої знання з цих тем.

Новий трирічний проект ЄС-МОП «На шляху до безпечної, здорової та задекларованої праці в Україні» передбачає допомогу у впровадженні європейських стандартів безпеки та здоров'я на роботі, трудових відносин, реалізації практичних заходів для зниження рівня незадекларованої праці.

У ст. 265 КЗпП з'являться нові штрафи. Так, за неподання або подання завідомо недостовірних відомостей, що визначені законодавством про безпеку праці та здоров'я працівників, роботодавцю доведеться сплатити штраф у розмірі 1% від середньомісячного фонду оплати праці юридичної чи фізичної особи, з якої стягується штраф, за попередній рік, але не менше трикратного розміру мінімальної заробітної плати. Робота в умовах підвищеної небезпеки, експлуатації машин, механізмів, устаткування підвищеної небезпеки без дозволу держави спричинить штраф у розмірі 3% від середньомісячного

фонду оплати праці юридичної чи фізичної особи, за попередній рік, але не менше ніж 5 МЗП. За невиконання приписів інспекторів праці порушники сплачуватимуть 5% від середньомісячного фонду оплати праці за попередній рік, але не менше 10 МЗП. А ось за завдання шкоди життю або здоров'ю працівників внаслідок недоліків на споруді, в обладнанні або методах роботи штраф становитиме 25% від середньомісячного фонду оплати праці за попередній рік, але не менше 50 МЗП. У 2019 році штрафи на роботодавців за ст. 265 КЗпП сягають 30 МЗП за неоформлених працівників, а за перешкоджання перевірці, мета якої виявити неоформлених працівників, роботодавці сплачують 100 МЗП.

До основних правових принципів у сфері охорони праці в Україні належать:

- пріоритет охорони життя та здоров'я працівників перед економічними інтересами; повна відповідальність роботодавця за створення належних, безпечних і здорових умов праці;
- соціальний захист працівників, повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
- компетентність діяльності щодо забезпечення охорони праці;
- двосторонній зобов'язуючий (і працівника, і роботодавця) характер діяльності щодо додержання норм з охорони праці;
- встановлення єдиних вимог з охорони праці для всіх підприємств та суб'єктів підприємницької діяльності незалежно від форм власності та видів діяльності;
- встановлення та гарантування заходів самозахисту прав працівників на здорові та безпечні умови праці;
- обов'язковість відшкодування шкоди, заподіяної працівникові у зв'язку з виконанням трудових обов'язків та інші[23].

7.2 Вплив ультра- та інфразвуку, на здоров'я та працездатність людини

Ультразвук широко використовуються в багатьох галузях промисловості. Джерелами таких коливань є генератори, які працюють в діапазоні частот від 12 до 22 кГц для обробки рідких розплавів, очищення відливочок, в апаратах для очищення газів. В гальванічних цехах ультразвук виникає під час роботи очищувальних та знежирювальних ванн. Його вплив спостерігається на віддалі 25-50 м від обладнання.

Механізми дії ультразвуку на живі організми вкрай різноманітний. Він викликає функціональні порушення нервової системи, головний біль, зміни кров'яного тиску та складу і властивостей крові, зумовлює втрату слухової чутливості, підвищену втомлюваність. Ультразвук впливає на людину через повітря а також через рідке і тверде середовище. Ультразвукові коливання поширюються у всіх згаданих вище середовищах з частотою понад 16000 Гц[16].

Вищезазначені прояви дії ультразвуку широко використовується в багатьох галузях промисловості для інтенсифікації процесів хімічного травлення, нанесення металевого покриття, очищення, змивання та знежирення деталей і виробів, дефектоскопії (оцінка якості зварних швів, структури сплаву) та інші.

За способом передачі від джерела до людини ультразвук поділяють на: повітряний (передається через повітря) та контактний (передається на руки людини, що працює через тверде чи рідинне середовище).

За спектром ультразвук поділяють на: низькочастотний (коливання частотою від $1,2 \times 10^4$ до $1,0 \times 10^6$ Гц, що передаються людині повітряним чи контактним шляхом) та високочастотний (коливання частотою від $1,0 \times 10^5$ до $1,0 \times 10^9$ Гц, що передаються людині тільки контактним шляхом)[17].

Дія звуків низькочастотних ультразвукових установок ($1,0 \times 10^4$ - $1,0 \times 10^5$ Гц) призводить до змін функцій центральної нервової системи, серцево-судинної й ендокринної систем, слухового і вестибулярного аналізаторів. В операторів на ультразвукових установках спостерігається астенія, судинна гіпотонія, знижена електрична активність серця, мозку та скелетних м'язів.

Високочастотний ультразвук ($1,0 \times 10^5$ - $1,0 \times 10^9$ Гц), не пошкоджуючи барабанної перетинки і середнього вуха, руйнує кортієв орган, глибокі структури органів і тканин людини, викликає порушення капілярного кровотоку, знижує відчуття болю.

Професійні захворювання зареєстровані лише при контактній передачі ультразвуку на руки – вегетосенсорна і сенсомоторна поліневропатія рук. При цьому небезпечна дія ультразвуку полягає в тому, що на руки і тіло працівника, при дотику до рідких і твердих середовищ діють коливання високої інтенсивності, створені в цих середовищах при роботі ультразвукових машин. Така дія викликає нагрівання тіла і призводить до змін у тканинах організму людини[15].

Допустимі рівні ультразвуку в місцях контакту частин тіла оператора з робочими органами машин не повинні перевищувати 110 дБ. За умови сумарної дії ультразвуку від 1 до 4 год. за зміну нормативне значення допускається збільшити на 6 дБ, при впливі від 1/4 до 1 год. - на 12 дБ, від 5 до 15 хв. - на 18 дБ, від 1 до 5 хв. - на 24 дБ. При вимірюванні ультразвуку вимірювальну точку беруть на рівні голови людини на відстані 5 см. від вуха. Мікрофон повинен бути спрямований в сторону джерела ультразвуку і віддалений не менше, ніж на 0,5 м від людини, яка здійснює вимірювання.

Ультразвук, так само як і інфразвук, орган слуху людини не сприймає, однак він може спричиняти біль голови, загальну втому, розлади серцево-судинної та нервової систем. При клінічному обстеженні може бути виявлений астеничний синдром. У осіб, що тривалий час зайняті експериментальною роботою на ультразвукових установках, іноді спостерігаються діенцефальні

порушення (зниження ваги, різкий підйом вмісту цукру в крові з повільним зниженням до вихідного рівня, підвищення механічного збудження м'язів тощо). Можливі порушення периферичної нервової системи (оніміння, зниження чутливості, гіпергідроз) порушення вестибулярного апарата. Периферичні порушення обумовлені переважно контактним впливом ультразвукових коливань.

Робітники, працюючі в умовах впливу ультразвуку, згідно з наказом МОЗ № 246 від 21.05.2007. «Про затвердження порядку проведення медичних оглядів працівників певних категорій» підлягають щорічному періодичному медичному огляду з обов'язковим залученням до складу лікарняної комісії невропатолога, офтальмолога, хірурга та проведенням досліджень вібраційної чутливості (за показанням). Обмеженням при прийомі на роботу в умовах впливу ультразвуку (окрім загальних) є хронічні захворювання периферичної нервової системи , а також облітеруючі захворювання артерій, периферичний ангіоспазм[17].

Інфразвук - це коливання в повітрі, в рідкому або твердому середовищах з частотою менше 16 Гц. Всі механізми, котрі працюють при частотах обертання менше 20 об/с, випромінюють інфразвук. При русі автомобіля з швидкістю понад 100 км/год він є джерелом інфразвуку, який утворюється за рахунок зриву повітряного потоку з його поверхні. В машинобудівній галузі інфразвук виникає при роботі вентиляторів, компресорів, двигунів внутрішнього згорання, дизельних двигунів. Завдяки великій довжині інфразвук поширюється в атмосфері на великі відстані.

Дослідження довели, що звук, якого не чути, також шкідливо впливає на здоров'я людини. Так, інфразвуки особливий вплив роблять на психічну сферу людини: уражають усі види інтелектуальної діяльності, погіршують настрої, іноді з'являється відчуття розгубленості, тривоги, переляку, страху, а при високій інтенсивності - почуття слабкості, як після сильного нервового потрясіння. Навіть слабкі інфразвуки можуть робити на людину істотний вплив, особливо якщо вони носять тривалий характер. На думку вчених, саме

інфразвуками, що нечутно проникають крізь самі товсті стіни, викликається багато нервових захворювань жителів великих міст. Високий рівень інфразвуку викликає порушення функції вестибулярного апарату, зумовлюючи запаморочення, біль голови. Знижується увага, працездатність[16].

При дії інфразвуку на організм людини на рівні 110-150 дБ можуть мати місце неприємні суб'єктивні відчуття, порушення функцій нервової, серцево-судинної і дихальної систем, вестибулярного аналізатора; може з'явитись відчуття страху, сонливість тощо. Специфічна для дії інфразвуку реакція – порушення рівноваги. При дії інфразвуку на рівні 105 дБ спостерігаються психофізіологічні реакції підвищеної тривоги і невпевненості, емоційної нестійкості[15].

Практично неможливо зупинити інфразвук за допомогою будівельних конструкцій на шляху його поширення. Неefективні також засоби індивідуального захисту. Дієвим засобом захисту є зниження рівня інфразвуку в джерелі його випромінювання. Серед таких заходів можна виділити наступні: збільшення частот обертання валів до 20 і більше обертів на секунду; підвищення жорсткості коливних конструкцій великих розмірів; усунення низькочастотних вібрацій; внесення конструктивних змін в будову джерел, що дозволяє перейти з області інфразвукових коливань в область звукових; в цьому випадку їх зниження може бути досягнуте застосуванням звукоізоляції та звукопоглинання[16].

Гігієнічна регламентація інфразвуку проводиться згідно з санітарними нормами. На робочих місцях рівні інфразвуку не повинні перевищувати 105 дБ (октавні смуги від 2 до 16 Гц). Так як загальноприйняті методи боротьби з шумом, засновані на звукоізоляції та звукопоглинанні, є малоефективні щодо інфразвуку, оскільки інфразвук має значно вищу проникливість, тому необхідно домагатись усунення або зниження його рівня в джерелі, що його генерує[15].

Боротьбу з інфразвуком у джерелі виникнення слід вести шляхом зміни режиму роботи технологічного обладнання — збільшення його швидкості, щоб основна частота силових імпульсів лежала за межами інфразвукового діапазону.

У боротьбі мають використовуватися заходи по зниженню інтенсивності аеродинамічних процесів - обмеження швидкості руху транспорту, зниження швидкостей витікання рідин (авіаційні і реактивні двигуни, двигуни внутрішнього згоряння, системи викиду пари теплових електростанцій і т. ін.)

Працюючі в умовах впливу інфразвуку мають проходити попередні і періодичні медичні огляди в строки і обсяги встановлені наказом МОЗ від 21.05.07 р. №246. Рекомендуються також лікувальні та профілактичні процедури, що використовуються для працюючих шумних і вібраційних професій[18].

7.3 Планування заходів цивільного захисту на об'єкті у випадку надзвичайних ситуацій.

Планування цивільного захисту об'єкта — це розроблення сукупності документів, у яких визначені сили і засоби, порядок і послідовність дій з метою забезпечення захисту населення, виробництва, а також виконання завдань вищих органів, пов'язаних із поданням допомоги населенню інших об'єктів і міст.

Ці документи, розроблені з урахуванням реальних можливостей і умов об'єкта, є настановою для організованих дій як з метою підготовки об'єкта до захисту в надзвичайних умовах, так із метою ліквідації наслідків надзвичайних ситуацій (стихійних лих, виробничих аварій і вогнищ воєнних конфліктів).

На об'єкті мають бути розроблені два плани: на воєнний та мирний час.

План цивільного захисту на воєнний час — це документи, які визначають організацію і порядок переведення об'єкта з мирного на воєнний час і ведення цивільного захисту в початковий період війни.

План цивільного захисту на мирний час — це документи, які визначають організацію і порядок виконання заходів цивільного захисту з метою запобігання або зменшення можливих втрат від важких виробничих аварій, катастроф, і стихійних лих, а також ведення рятувальних та інших невідкладних робіт при їх виникненні.

Реальність розроблених планів ЦЗ буде залежати від повноти вихідних даних, наявності сил і засобів, правильного обліку всіх можливостей об'єкта. Плани ЦЗ об'єкта розробляють його керівники, спеціалісти і орган управління ЦЗ. Розробляючи заходи служб (формувань) ЦЗ об'єкта, ряд питань необхідно узгоджувати з відповідними районними службами ЦЗ, районним відділом з питань НС та цивільного захисту населення.

Об'єкт, що знаходиться на території одного або кількох населених пунктів, є основним об'єктом, який складає єдиний план разом з адміністрацією населеного пункту.

Розробка плану відбувається у три етапи в певній послідовності. Перший етап — підготовчий, протягом якого визначається склад виконавців і затвердження їх, підготовка виконавців до роботи, доведення до них директив, рекомендацій та інших документів, узагальнення й аналіз вихідних даних, необхідних для розробки плану ЦЗ, визначення обсягу робіт і розподіл обов'язків між виконавцями та закріплення відповідальних за розділами плану.

Для планування, підготовки і проведення заходів евакуації має бути інформація, щоб забезпечити відповіді на такі запитання: чисельність працюючих відвідувачів, обслуговуючого персоналу на даному об'єкті, всього населення в населеному пункті; час доби, коли буває найбільше скупчення людей у приміщеннях; розміщення людей у приміщеннях; стан входів, аварійних виходів; наявність і стан входів для пожежників, міліції, поліції, внесення технічних засобів; труднощі, які треба враховувати під час евакуації

людей (вузькі проходи, сходи, непрацюючі ліфти та ін.); забезпеченість будівельними матеріалами, матеріалами для огорожування небезпечних місць, захищення аварійної або цінної апаратури; забезпеченість тимчасовими робочими місцями та ін.; устаткування, прилади, апаратура, документи, які необхідно евакуювати і перелік тих, що можна залишити; можливість переведення виробництва, переведення установ на скорочений режим роботи; вирішення питань зупинення виробництва, установ на скорочений режим роботи; вирішення питань зупинення роботи технологічних ліній, припинення чи скорочення виробництва продукції; забезпечення засобами індивідуального захисту, оповіщення і зв'язку; підготовленість пунктів збору, транспорту для перевезення людей і цінностей, наявність поблизу загрозливих об'єктів (пожежо- і вибухонебезпечні будівлі й матеріали, столярні цехи, приміщення складів, комор, трансформаторні приміщення, хімічні підприємства чи склади)[20].

Планування евакуації має передбачати виникнення найбільш несприятливих ситуацій під час підготовки і проведення евакуації: відсутність відповідних керівників, транспорту, електрозабезпечення, погані погодні умови, аварія на дорозі, паніка серед людей та ін.

Другий етап — практична розробка, оформлення документів. Заходи, які плануються в документах плану, мають бути спрямовані на виконання завдань ЦЗ в надзвичайних ситуаціях.

У документах плану визначають заходи, які потрібно виконати в мирний час, при загрозі виникнення надзвичайних ситуацій, несподіваному нападі противника, стихійних лихах, виробничих аваріях, катастрофах і при ліквідації наслідків надзвичайних ситуацій, проведенні рятувальних та інших невідкладних робіт, а також характер і порядок дій формувань, зміст і обсяг робіт, строки виконання заходів з урахуванням конкретних умов і можливостей даного об'єкта.

Планування забезпечення особового складу формувань ЦЗ засобами індивідуального захисту та іншими матеріально-технічними засобами

проводиться за нормами, затвердженими Начальником ЦЗ України, узгодженими з Кабінетом Міністрів, відповідним міністерством, відомством.

Третій етап — узгодження розроблених планів із відділом ЦЗ району, з районним агропромисловим управлінням, адміністрацією населеного пункту, службами ЦЗ району, після цього затвердження документів плану ЦЗ. Документи плану ЦЗ підписує керівник — ЦЗ об'єкта, деякі (план евакуації, прийому і розміщення евакуйованого) підписує і начальник ЦЗ голова адміністрації населеного пункту. Зміст плану ЦЗ об'єкту узгоджується з вимогами плану ЦЗ району, що підтверджує начальник відділу з питань цивільного захисту населення району, після чого план ЦЗ затверджує керівник ЦЗ об'єкта.

Після затвердження плану об'єкта організується вивчення документів усім керівним складом об'єкта[19].

7.4 Вплив електромагнітного імпульсу (ЕМІ) ядерного вибуху на елементи виробництва та заходи захисту.

Для того, щоб зрозуміти всю складність проблем загрози ЕМІ і заходів щодо захисту від неї, необхідно коротко розглянути історію вивчення цього фізичного явища і сучасний стан знань у цій області.

Те, що ядерний вибух буде обов'язково супроводжуватися електромагнітним випромінюванням, було ясно фізикам-теоретикам ще до першого випробування ядерного пристрою в 1945 році. Під час проведених в кінці 50-х - початку 60-х років ядерних вибухів в атмосфері і космічному просторі наявність ЕМІ було зафіксовано експериментально. Однак кількісні характеристики імпульсу вимірювалися в недостатній мірі, по-перше, тому що була відсутня контрольно-вимірювальна апаратура, здатна реєструвати надзвичайно потужне електромагнітне випромінювання, що існує надзвичайно короткий час (мільйонні частки секунду), по-друге, тому що в ті роки в радіоелектронній апаратурі використовувалися виключно

електровакуумні прилади, які мало схильні до впливу ЕМІ, що знижувало інтерес до його вивчення.

Створення напівпровідникових приладів, а потім і інтегральних схем, особливо пристроїв цифрової техніки на їх основі, і широке впровадження засобів в радіоелектронну військову апаратуру примусили військових фахівців оцінити загрозу ЕМІ. З 1970 року питання захисту зброї і військової техніки від ЕМІ стали розглядатися міністерством оборони США як такі що мають вищу пріоритетність.

Основною причиною виникнення ЕМІ тривалістю менше 1 с вважають взаємодію гамма-променів і нейтронів ядерного вибуху з атомами газів повітря, внаслідок чого з них вибиваються електрони (ефект Комптона) і хаотично розлітаються в середовищі позитивно заряджених атомів газів. Важливе значення має також виникнення асиметрії в розподілі просторових електричних зарядів, пов'язаних з особливостями поширення гамма-променів і утворення електронів.

Гамма-промені, які випускаються із зони вибуху в напрямі поверхні землі, поглинаються в більш щільних шарах атмосфери, вибиваючи з атомів повітря швидкі електрони, які летять у напрямку гамма-променів зі швидкістю світла, а позитивні іони (залишки атомів) залишаються на місці. У результаті поділу і переміщення позитивних і негативних зарядів у цій області й у зоні вибуху, а також при взаємодії зарядів з геомагнітним полем Землі утворюються елементарні й результуючі електричні та магнітні поля ЕМІ, які досягають поверхні землі в зоні радіусом кількох сотень кілометрів. Виникають сильні поперечні струми і утворюється подібність великої "плоскої антени", яка випромінює потужний ЕМІ з часом наростання порядку 10 нс і тривалістю більше 230 нс; зі смугою частот від 10 кГц до 100 МГц. Залежно від висоти ядерного вибуху за інших однакових умов змінюються характер, інтенсивність ЕМІ і дальність його поширення.

При наземному і низькому повітряному вибуху вражаюча дія НМІ спостерігається на відстані кількох кілометрів від центру вибуху. Під час

ядерного вибуху на висотах від 3 до 25 км утворюється симетричне джерело генерації, але радіус поширення ЕМІ залишається обмеженим внаслідок сильного поглинання гамма-випромінювання в щільних шарах атмосфери.

Найбільшу вражаючу дію має ЕМІ, що виникає при екзоатмосферному вибуху (більше 40 км). Зі збільшенням висоти вибуху збільшується і район джерела генерації ЕМІ, досягаючи в діаметрі тисячі кілометрів і товщини 20--40 км. Так, під час вибуху на висоті 80 км. ЕМІ буде поширюватися на площі радіусом 960 км, а під час вибуху на висоті 160 км -- на площі радіусом 1400 км. Екзоатмосферний ЕМІ характеризується дуже малим часом наростання (декілька сот наносекунд), високою інтенсивністю електричного поля (більше 50 кВ/хв) і магнітного поля (близько 130 А/хв). Розряд блискавки порівняно з ЕМІ має значно більшу тривалість зростання і спаду (5--300 мкс), створює дуже потужні поля (близько 100 кВ/хв), несе значно більшу енергію, але спектр частот становить близько 10 МГц, тоді як для ЕМІ він більше -- 100 МГц. Пікове значення ЕМІ може досягти 50 000 В/хв, що дорівнює всій енергії яка випромінюється в радіочастотній частині спектра.

Вражаюча дія ЕМІ обумовлена виникненням напруги і струмів у провідниках різної довжини, розміщених у повітрі, землі.

ЕМІ захвачують спектр частот від десятків до кількох сотень мегагерц, тобто діапазон, в якому працюють установки електропостачання, зв'язку і радіолокації.

Напруженість електромагнітного поля, створюваного ЕМІ, досягає 50 000 В/м, тоді як у радіолокації вона не перевищує 200 В/м, а у зв'язку -- 10 В/м.

ЕМІ генеруються і при інших видах ядерних вибухів - повітряному і наземному. Теоретично встановлено, що в цих випадках його інтенсивність залежить від ступеня асиметричності просторових параметрів вибуху. Тому повітряний вибух з точки зору генерації ЕМІ найменш ефективний. ЕМІ наземного вибуху буде мати високу інтенсивність, проте вона швидко зменшується в міру віддалення від епіцентру.

Час наростання ЕМІ до максимального становить кілька мільярдних частинок секунди, що значно менше часу спрацьовування відомих електронних систем захисту. Це значить, що в момент приходу ЕМІ чутливе електронне обладнання одержить дуже велике перевантаження, протистояти якому воно не зможе. Параметри ЕМІ залежать від потужності й висоти вибуху, а також відстані від епіцентру вибуху. При вибухах над атмосферою на висоті понад 100 км мегатонного діапазону створюються ЕМІ, які охоплюють своєю дією велику територію, багато тисяч квадратних кілометрів.

ЕМІ уражає радіоелектронну і радіотехнічну апаратуру. В провідниках індукуються високі напруги і струми, які можуть призвести до постійних або тимчасових пошкоджень ізоляції кабелів, відключення реле і переривників, пошкодження елементів зв'язку, магнітних запам'ятовуючих пристроїв у ЕОМ і системах передачі даних тощо. Найбільш уразливими елементами обладнання є напівпровідникові прилади -- транзистори, діоди, кремневі випрямлячі, інтегруючі ланцюги, цифрові процесори, управляючі й контрольні прилади. Чутливі до пошкодження ЕМІ транзистори звукової частоти, перемикаючі транзистори, інтегруючі ланцюги та ін[21].

7.5 Висновок до сьомого розділу

У шостому розділі було розглянуто нове законодавство з питань охорони праці станом на 2019р та зміни у ньому, вплив ультра- та інфразвуку на здоров'я та працездатність людини, планування заходів цивільного захисту на об'єкті у випадку надзвичайних ситуацій та вплив електромагнітного імпульсу (ЕМІ) ядерного вибуху на елементи виробництва та заходи захисту.

ЗАГАЛЬНІ ВИСНОВКИ ДО ДИПЛОМНІ РОБОТИ

Результатом розробки даної магістерської роботи стало отримання повноцінного програмного продукту «Система керування готелем» з автоматизованим багатокритеріальним методом підбору номерів.

Перевагами даного програмного продукту є легкий доступ до необхідної інформації для будь-якого рядового користувача, а також зручне керування матеріалами для адміністраторів ресурсу. Даний програмний продукт можна використовувати на будь-якому сервері де встановлене необхідне програмне забезпечення, а саме будь-який веб-сервер, побудований на скриптовій мові програмування PHP5, та встановленим сервером баз даних – MySQL.

Дане програмне забезпечення є вдалим проектом, оскільки виконує усі необхідні завдання, працює без помилок та достатньо швидко з малим навантаженням на веб-сервер.

Подальший розвиток передбачає створення повноцінного функціоналу для забезпечення для виправлення вищезгаданих недоліків.

Основні результатом магістерської роботи є застосування методу автоматизованого багатокритеріального підбору номерів на прикладі системи керування готелем. Результати роботи можуть бути впроваджені в процес розробки будь якої системи керування.

В спеціальній частині описано набір рекомендацій для розробників на PHP – PHP Standards Recommendations, який слугує стандартизації концепцій програмування в PHP. В магістерській роботі «Система керування готелем з автоматизованим багатокритеріальним підбором номерів» було використано дані стандарти, які допомагають швидше розібратись в коді та систематизувати його.

Мета розділу полягає в тому, щоб забезпечити взаємодію компонентів та створити загальну технічну базу для впровадження перевірених концепцій

оптимальних методів програмування та тестування для написання коду магістерської роботи.

В розділі «Обґрунтування економічної ефективності» дипломної роботи освітнього рівня «магістр» було розраховано основні техніко-економічні показники розробки системи керування готелем та зроблено висновок, що створення автоматизованої системи управління готелем з використанням багатокритеріального підбору номерів є доцільною у зв'язку з невеликим терміном окупності та великим обсягом планового прибутку.

В розділі «Охорона праці та безпека в надзвичайних ситуаціях» було розглянуто нове законодавство з питань охорони праці станом на 2019р та зміни у ньому, вплив ультра- та інфразвуку на здоров'я та працездатність людини, планування заходів цивільного захисту на об'єкті у випадку надзвичайних ситуацій та вплив електромагнітного імпульсу (ЕМІ) ядерного вибуху на елементи виробництва та заходи захисту.

В розділі «Екологія» було розглянуто сучасні програмні продукти для обробки великих масивів екологічної інформації та зниження енергоємності та енергозбереження в комп'ютерних технологіях. В результаті опрацювання розділу було зроблено висновок, якщо кожен власник комп'ютерної техніки буде відключати свій комп'ютер або переводити його у режим сну на час коли він йому не потрібен, це приведе до величезної економії електроенергії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Servio Hotel Management System – Режим доступу: <https://www.intellect.co.ua/avtomatizatsiya-gotelyu/> – Дата доступу: 20.11.2019
2. Теорія прийняття рішень: підручник для студентів / Л.С.Файнзільберг, О.А. Жуковська, В.С. Якимчук. – Київ: Освіта України, 2018. – 246 с.
3. A Methodology for the Selection of Multi-Criteria Decision Analysis Methods in Real Estate and Land Management Processes / Maria Rosaria Guarini , Fabrizio Battisti and Anthea Chiovitti. - Department of Architecture and Design, Sapienza University of Rome, 2018
4. Автоматизовані системи управління готелями – Режим доступу: [http://tourlib.net/statti_ukr/gudzovata.htm /](http://tourlib.net/statti_ukr/gudzovata.htm/) – Дата доступу: 22.11.2019
5. Загальна характеристика автоматизованих систем управління – Режим доступу: http://www.d-learn.pu.if.ua/data/users/9645/Lecture1_ISU.pdf – Дата доступу: 18.11.2019
6. Advantages of an HTML-based website. – Режим доступу: [http://www.vtech-seo.com/web-design-articles/advantages-of-html.html /](http://www.vtech-seo.com/web-design-articles/advantages-of-html.html) – Дата доступу: 20.11.2019
7. What can PHP do? – Режим доступу: <https://www.php.net/manual/en/intro-whatcando.php> – Дата доступу: 15.11.2019
8. What Is MySQL – Режим доступу: <http://www.mysqltutorial.org/what-is-mysql/> – Дата доступу: 29.11.2019
9. Seven great PHP IDEs compared – Режим доступу: <https://www.ibm.com/developerworks/library/os-php-ide/index.html> – Дата доступу: 21.11.2019
10. Eclipse documentation - Archived Release – Режим доступу: https://help.eclipse.org/kepler/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm – Дата доступу: 21.11.2019

11. Накази Держпраці 2019 рік – Режим доступу:<http://dsp.gov.ua/26463-2/> – Дата доступу: 21.11.2019
12. ЖИТЛОВО-КОМУНАЛЬНЕ ГОСПОДАРСТВО – Режим доступу: <http://www.volnovrda.gov.ua/zhytlovo-komunalne-hospodarstvo/1953-yak-zaoshchaditi-elektrichnu-energiyu-za-dopomogoyu-komp-yuternoji-tekhniki> – Дата доступу: 20.11.2019
13. New approach to computing boosts energy efficiency – Режим доступу: <https://www.sciencedaily.com/releases/2016/08/160826083950.htm> – Дата доступу: 20.11.2019
14. Тарасова В.В. Екологічна статистика.[Текст]/В.В.Тарасова.- Київ: «Центр учбово\ї літератури», 2008 ро.-391с.
15. Заходи та засоби захисту від шуму – Режим доступу: <https://studfile.net/preview/5252766/page:23/> – Дата доступу: 02.12.2019
16. Інфра- та ультразвук. Параметри інфра- та ультразвукових коливань – Режим доступу: <http://studies.in.ua/bjd-atamachuk/1068-335-nfra-ta-ultrazvuk-parametri-nfra-ta-ultrazvukovih-kolivan.html> – Дата доступу: 03.12.2019
17. Шкідливий вплив ультразвуку на здоров'я працюючих та його профілактика – Режим доступу: <http://oppb.com.ua/news/shkidlyvyy-vplyv-ultrazvuku-na-zdorovya-pracyuyuchyh-ta-yogo-profilaktyka> – Дата доступу: 03.12.2019
18. Охорона праці / Що таке інфразвук? – Режим доступу: https://pidruchniki.com/1180091238963/bzhd/scho_take_infrazvuk – Дата доступу: 05.12.2019
19. Планування з питань цивільного захисту – Режим доступу: <https://studfile.net/preview/5705643/page:7/> – Дата доступу: 02.12.2019
20. Цивільна оборона та цивільний захист / План цивільного захисту на мирний час – Режим доступу: https://pidruchniki.com/19110522/bzhd/plan_tsivilnogo_zahistu_mirniy_chas – Дата доступу: 03.12.2019

21. Цивільна оборона та цивільний захист/ Електромагнітний імпульс – Режим доступу: http://pidruchniki.com/17910211/bzhd/elektromagnitniy_impuls – Дата доступу: 04.12.2019
22. Новий Показчик нормативно-правових актів з охорони праці – Режим доступу: https://www.sop.com.ua/article/616-noviy-pokajchik-normativno-pravovih-aktiv-z-ohoroni-prats?from=statbloc_topmenu_link – Дата доступу: 04.12.2019
23. Федерація профспілок працівників малого та середнього підприємництва України/ закон України «про охорону праці» № 2694-12 – Режим доступу: <http://federation.org.ua/zakonodavstvo/zakon-ukraini-pro-okhoronu-pratsi-2694-12.html> – Дата доступу: 04.12.2019
24. Хроніки детерминированности – Режим доступу: https://svyatoslav.biz/misc/psr_translation/ – Дата доступу: 05.12.2019
25. PSR-0 – Стандарт автозагрузки – Режим доступу: <https://webhamster.ru/mytetrashare/index/mtb0/1492091791auq91c7j> – Дата доступу: 04.12.2019
26. Стандарты оформления кода PSR-1 и PSR-2 – Режим доступу: <https://dmitriygnatenko.ru/article/psr> – Дата доступу: 04.12.2019
27. PSR-3: Logger Interface – Режим доступу: <https://www.php-fig.org/psr/psr-3/> – Дата доступу: 15.11.2019
28. PSR-4: автозагрузчик – Режим доступу: <https://riptutorial.com/ru/php/example/32582/psr-4> – Дата доступу: 14.11.2019
29. The method for comparative evaluation of software architecture with accounting of trade-offs / A Kharchenko, I Bodnarchuk, V Yatsyshyn - American Journal of Information Systems, 2014 – Режим доступу: https://scholar.google.com/citations?user=pJCS3A4AAAAJ&hl=uk#d=gs_md_cita0 – Дата доступу: 02.12.2019
30. Optimization of software architecture selection for the system under design and reengineering / O Kharchenko, I Raichev, I Bodnarchuk, N Zagorodna - 2018 14th International Conference on Advanced ..., 2018 – Режим доступу:

https://scholar.google.com/citations?user=pJCS3A4AAAAAJ&hl=uk#d=gs_md_cita-120 – Дата доступу: 02.12.2019

31. Лаура Томсон. Разработка Web-приложений на PHP и MySQL. / Лаура Томсон, Люк Веллинг; пер. с англ. А. Моргунов. – М.:Издательский дом "Вильямс", 2010. – 672с. – ISBN 978-5-8459-1574-0

32. CSS Tutorial // W3Schools Online Web Tutorials – Режим доступу: <http://www.w3schools.com/css/default.asp>. – Дата доступу: 05.11.2019

33. PHP: Documentation // PHP: Hypertext Preprocessor – Режим доступу: <http://php.net/docs.php>. – Дата доступу: 22.11.2019

34. Документация по MySQL // MySQL.RU – Режим доступу: <http://www.mysql.ru/docs/>. – Дата доступу: 15.11.2019

35. Современный учебник JavaScript // Учебник JavaScript с примерами скриптов – Режим доступу: <http://javascript.ru> – Дата доступу: 25.11.2019

36. Що таке PHP – Режим доступу: <http://www.lib.mdpu.org.ua/e-book/web/lec1.htm> – Дата доступу: 22.11.2019

37. Знайомство з PHP – Режим доступу: <https://wiki.cusru.edu.ua/index.php> – Дата доступу: 20.11.2019

38. Можливості PHP – Режим доступу: <http://web.if.ua/14-mozhlivost-php.html> – Дата доступу: 21.11.2019

39. Інструкція до лабораторної роботи з курсу «Автоматизовані системи обробки інформації з обмеженим доступом» для студентів спеціальності: 7.17010201 «Системи технічного захисту інформації, автоматизація її обробки» / Укл.: В. Б. Дудикевич, В. М. Максимович, Г. В. Микитин, Ю. Р. Гарасим. – Львів: НУ «ЛП», 2012. – 17 с.

40. Karimi, A.P.; Mehrdadi, N.; Hashemian, S.J.; Bidhendi, G.R.; Moghaddam, R.T. Using the fuzzy TOPSIS and fuzzy AHP methods for wastewater treatment process selection. Int. J. Acad. Res. 2011, 3, 737–745.

41. Escrig-Olmedo, E.; Rivera-Lirio, J.M.; Muñoz-Torres, M.J.; Fernández-Izquierdo, M.Á. Integrating multiple ESG investors' preferences into

sustainable investment: A fuzzy multicriteria methodological approach. J. Clean. Prod. 2017, 162, 1334–1345.

42. Khoshnava, S.M.; Rostami, R.; Valipour, A.; Ismail, M.; Rahmat, A.R. Rank of green building material criteria based on the three pillars of sustainability using the hybrid multi criteria decision making method. J. Clean. Prod. 2018, 173, 82–99.

43. Чибісов, Ю. В. Формування раціональних потоків поїздів на мережі залізниць [Текст] / Ю. В. Чибісов // Вісник Нац.о техн. ун-ту «Харківського політехнічного інституту». – Харків: Вид-во техн. ун-ту ХПІ, 2013. – Вип. 56. – С. 66–76.

44. Божанова, Т. А. Про узагальнені розв’язки однієї задачі векторної оптимізації на транспортних мережах [Електр. ресур] / Т. А. Божанова, П. І. Когут // Динамические системы: зб. наук. праць. – 2010. – Вип. 28. – С. 48–62. – Режим доступа : http://www.dynsys.crimea.edu/issue/28/dynsys_28_bozhanova.pdf – Дата доступу: 15.11.2019

45. D’Apice, C. Efficient Controls for Traffic Flow on Networks / C. D’Apice, P. I. Kogut, R. Manzo // Dynamical and Control Systems. – 16(2010). – № 3. – P. 407–437. 7

46. Тарасова В.В. Екологічна статистика. [Текст] / В. В. Тарасова. – Київ: «Центр учбової літератури», 2008 р. – 391 с.

47. Микитюк О.М., Злотін О.З., Бровдій В.М. Екологія людини: Підручник // 3-є вид., випр. і доп. Харків: ОВС, 2004. 256 с.

48. Экология города /под ред. Ф. В. Стольберга Киев: Знания, 2000.

49. Білявський Г. О. Основи екології [Текст]: підручник для студ. вищих навч. закладів / Г. О. Білявський, Р. С. Фурдуй, І. Ю. Костіков. К. : Либідь, 2004. – 408 с. ISBN 966-06-0289-8.

50. Нормативно-правові акти у сфері пожежної та техногенної безпеки. – Режим доступу: <https://km.dsns.gov.ua/ua/Normativna-baza-.html>. – Назва з екрану. Дата доступу: 15.11.2019